

Ngenea Hub

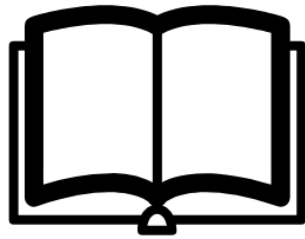
Ngenea Hub2 provides a management and control interface for data and systems across a global pixstor estate.

Using Hub2 users can create Spaces for datasets across multiple pixstors, enable data synchronisation, perform data manipulation through workflows and control system settings for all pixstors from one administrative interface.

Move data based on business needs

With ngenea, you can quickly and securely transport data to and from globally distributed cloud, object storage, traditional NAS files and tape resources - automatically moving data into the 'right cost' resource according to value and usage as your work teams and business needs demand.

Pick your area of interest:



[User Guide](#)



[Administration Guide](#)

User Guide

Overview

Ngenea Hub provides a management and control interface for data and systems across a global pixstor estate.

Using Ngenea Hub, users can create Spaces for datasets across multiple pixstors, enable data synchronisation, perform data manipulation through workflows and control system settings for all pixstors from one administrative interface.

Accessing Hub

To login to Hub enter the URL of the hub into the browser address bar. E.G. <https://myhub>

On successful connection to the Hub a login page is presented.

The image shows a login interface for NG-Hub. At the top center is the NG-Hub logo, which consists of the text "NG-Hub" in a white sans-serif font, followed by a trademark symbol (TM). To the right of the text is a circular graphic composed of many small dots in shades of blue and green. Below the logo, there are two white input fields. The first field is labeled "Username" in a small, white, sans-serif font. The second field is labeled "Password" in the same font. Below these fields is a green rectangular button with rounded corners. Inside the button, there is a white right-pointing arrow followed by the word "Login" in a white sans-serif font. The entire login form is set against a dark teal background.

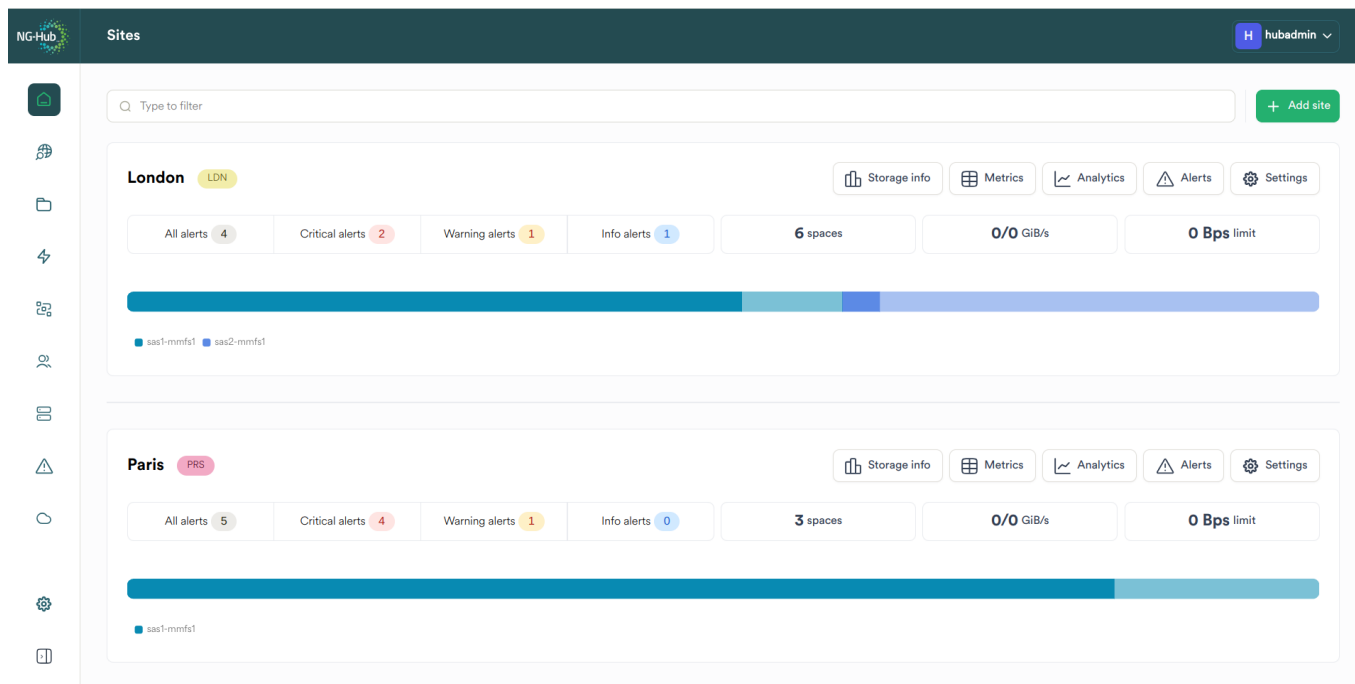
Enter your Username and Password to authenticate with Hub and login.

Logging Out Of Hub

After successful login, from the User Profile control in the top right of the screen select the Log out option.

The Main Screen

The main Hub screen is comprised of several areas.



Menu Bar

The menu bar is positioned on the far left of the Hub screen.

Menu Buttons

The menu buttons perform the following functions when clicked:

Home



Navigates to the page first presented after Login

Spaces



Navigates to the Spaces screen

Jobs



Navigates to the Jobs screen

Groups & Users



Navigates to the Groups & Users screen

Sites



Navigates to the Sites screen

Alerts



Navigates to the Global Alerts screen

Targets



Navigates to the Targets screen

Global Settings



Navigates to the Global Settings screen

Enlarge/Reduce



Clicking the toggle widens the menu bar to show:

- Menu button descriptions

- Hub version

Clicking the toggle button again shrinks the menu bar.

Page Location

The page location is positioned to the top of the Hub screen.

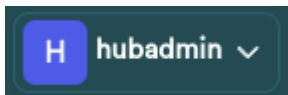
Main page content

The main page content is positioned to the center of the Hub screen.

User Profile

The User Profile control is positioned to the top right of the Hub screen.

The User Profile control displays the name of the user logged into Hub using the current browser session.



Use this control to Log out of Hub and update your User settings.

User Profile

Selecting the Profile option from the drop down menu presents the User settings dialog of the logged in user.

Clicking the Save button at the bottom of the User settings dialog saves any changes made.

hubadmin

Basic information

Password

Type the password

Leave empty to keep the password unchanged.

First name

Hub✕

Last name

Admin✕

Email

hubadmin@mycompany.com✕

Groups

1 group selected▼

Administrators

NAS User Settings

Also create this user on all pixstors

☐

API Keys

Save

Password

Enter a new Password to change your Password.

First name

Enter a new first name to change your First name.

Last name

Enter a new Last name to change your Last name.

Email

Enter a new valid email address to change your email address.

Groups

A list of groups your user account belongs to is displayed.

API Keys

Provides the capability to add and remove API keys.

An API key is typically used for by 3rd party software to connect to Hub to perform automation.

Adding an API Key

+ Add API key

To add a new API key, click the Add API key button.

Enter a name for the API Key .

Name

MyAPIKey



Tip: It is not possible to change the name of an API key once the key has been created.

Add additional API keys as required.

Upon pressing Save the API Key dialog is raised displaying the API Keys created.

API keys created

These are the created API keys. You won't be able to display them again. Please copy them before you close this window.



```
{
  "root": [ 1 item
    {
      "name": "MyAPIKey"
      "key": "9tbdetnY.jGbTyHBg8f5SHKQHQxwU7VFSLwBTWn7H"
    }
  ]
}
```

Close

Ensure to save the generated keys safely as once the API Key dialog is closed the keys cannot be viewed again.

Removing an API key

Created API Keys are listed

API Keys

Key 1

Name

MyAPIKey

+ Add API key



To remove an API key, click the dustbin icon next to the API Key.

Tip: Deleted API Keys are non-recoverable. If an API Key has been inadvertently removed, do not press the Save button, instead click off the User settings dialog to the main area of the screen.

Clicking the Save button at the bottom of the User settings dialog saves any changes made.

Search

Hub provides the ability to Search for assets across all Hub managed Sites.

Where proxies are enabled for Search supported asset types, thumbnails and proxies are displayed.

Asset locations, Spaces, metadata and Tags are searchable providing rich criteria to locate assets.

Located assets can be processed using the Jobs Panel to perform Workflow actions.

Searching

Enter a term in the Search bar to start a Search. E.G.: football.

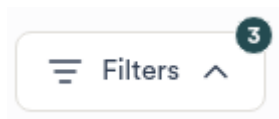
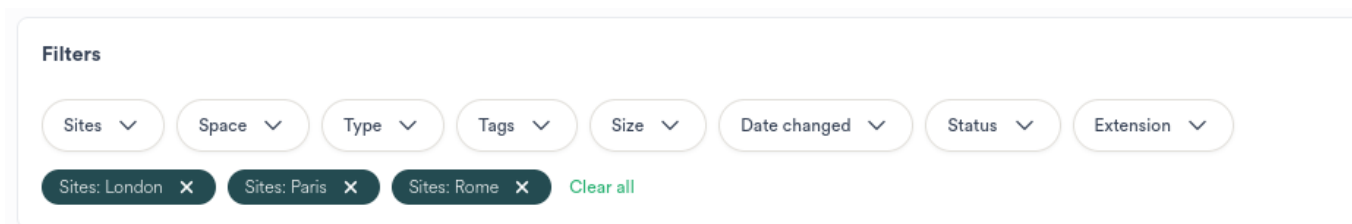
Terms are automatically wildcarded to enable wide Searching. E.G.: *football*.

Global search

Q football

Filters

Search provides several high level filters each of which apply additional criteria to narrow the Search to return only the required assets.



Click the filter button to display the available filters. The number of applied filters is displayed on the filter button.

Sidebar



Clicking the toggle displays the side bar to show:

- Item Info
- Tags
- Metadata

Clicking the toggle button again hides the side bar.



Item info

Name	football hd001.png
Path	/mmfs1/data/datasync/footbal...
Size	3.2 MB

Tags

Empty (add tags)

Metadata (core)

accesstime	24.01.2024, 10:25 GMT
blocksize	8388608
changetime	24.01.2024, 10:25 GMT
deviceid	250609814
directory	/mmfs1/data/datasync/footbal...

Reordering

Re-arrange items in top to bottom order by dragging vertically up or down.

Side bar organizer



Preview

Item preview



Item info

Item details



Tags

Tags



Metadata (core)

Metadata (core)



Metadata (gpfs)

Metadata (gpfs)

Save

Click Save to store the sequence which is reflected in the Job side bar.

Results

Search matches assets to the Search input returning collated results to screen.

Searches are processed asynchronously, therefore some results may take longer to return to screen than others.

Where a Site is unavailable or takes too long to return results, information messages appear at the top of the screen allowing for deselection of the Site or Site(s).

Filters

342 results

football

2 instances

football hd000.png

24.01.2024, 10:25 GMT | 3.2 MB

ROM / Space: datasync / football / png

football hd001.png

24.01.2024, 10:25 GMT | 3.2 MB

ROM / Space: datasync / football / png

football hd002.png

24.01.2024, 10:25 GMT | 3.2 MB

ROM / Space: datasync / football / png

football hd003.png

24.01.2024, 10:25 GMT | 3.2 MB

ROM / Space: datasync / football / png

football hd004.png

24.01.2024, 10:25 GMT | 3.2 MB

PAR / Space: datasync / football / png

football hd005.png

24.01.2024, 10:25 GMT | 3.2 MB

Result Card

Returned assets appear as a Result Card.

football hd015.png

24.01.2024, 10:25 GMT | 3.2 MB

PAR / Space: datasync / football / png

Where multiple instances of the same asset are present on more than one Site, the results are collated.

2 instances

Open the instances drop down to display all the assets from all Sites

The Result Card displays:

Item	Description
Thumbnail	The generated thumbnail or a default icon where no thumbnail exists
Name	The name of the asset
Date and Size	The date and time of last modification, the size of the file on the pixstor file system
Site chip	The site on which the asset is found

Item	Description
Space Location	The location of the asset in the Space
	The filebrowser navigation button

Tags

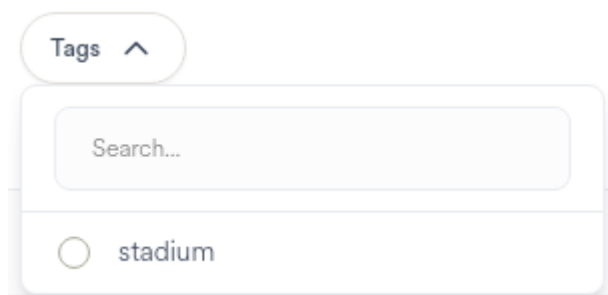
Search provides the ability to Tag assets.

Apply Search Tags to assets in order to collate assets into collections or curate as required.

Tip: Applied Tags are added to Site Search services asynchronously. Dependent on workload there may be a small delay between the addition of the Tag in the UI and the ability to Search for the asset by the Tag.

Searching By Tag

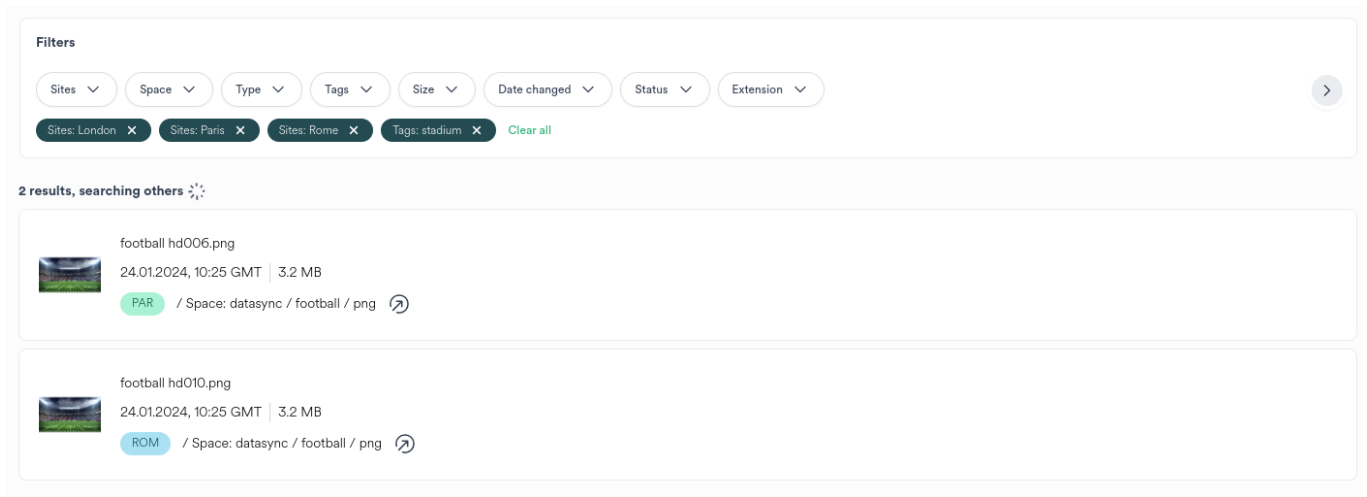
- To Search for an asset by Tag, open the Search filter bar and select the Tag from the drop down menu
- Entering text into the Tag filter matches available Tags to provide a shorter selection list



Applied Tags are denoted with a chip below the filter bar. Click the x to remove the Tag from the filter.

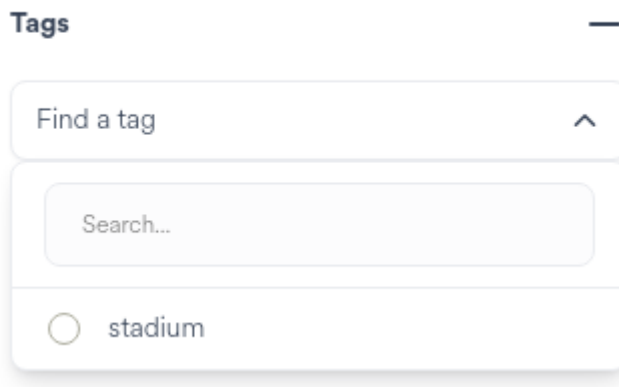


Example of Searching for assets where the Tag stadium is applied:

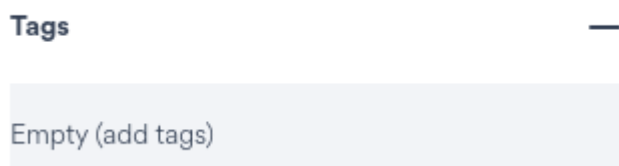


Tagging An Asset

To add a Tag to an asset, open the side bar and locate the Tags section in the bar.



Where no Tags are set on the asset, a prompt to add Tags is shown. Click the entry box to start.



To apply an existing Tag to the asset, select the Tag.



Applied Tags are denoted with a chip below the filter bar. Click the x to remove the Tag from the filter.

Tags

Find a tag

stadium

If no Tags exist, enter a Tag (case sensitive) in the Tag entry box and press Enter to create and apply the new Tag.

Tags

Find a tag

NFL

No results found (Create new)

Actions

Post Search, assets can be actioned by Workflows using the Job Panel.

To enable asset selection first select an individual Space from the Search filters.

Space

Selecting an individual Space enables asset selection.

- Click the checkbox to the left of an asset to add the asset to the Job Panel selection list



football hd006.png

24.01.2024, 10:25 GMT | 3.2 MB

PAR

/ Space: datasync / football / png



football hd010.png

24.01.2024, 10:25 GMT | 3.2 MB

ROM

/ Space: datasync / football / png



Assets in the Job Panel are processed per Space, per Site. Selecting assets from additional Sites raises a confirmation dialog.

- Choose whether to retain the assets from the current Site or to clear all the current selections in order to add new selections from other Site(s)

Different sources

X

You have files selected from a different source site/space in the selection list. You can either clear all selections, or go back to source site/space to continue adding from the same source.

Clear selections

Go back

When the Job Panel selection contains selected items, navigating to the Space on the selected Site retains the selections in the Job Panel selection list.

Click the filebrowser navigation button on the Space card to navigate to the selected Space, filtered by Site. Any assets already selected remain in the Job Panel selection list.

Example of asset selections after navigating to the Space on the Site with assets selected from the Search results:

NGHub

Spaces

hubadmin

datasync > football > png

Paris

Type to filter

Filters

	Name	Date modified	Size	File status	Number of files
☐	football hd000.png	Wed, 24 Jan 2024, 10:25:00 GMT	3.2 MB	ROM	-
☐	football hd001.png	Wed, 24 Jan 2024, 10:25:01 GMT	3.2 MB	ROM	-
☐	football hd002.png	Wed, 24 Jan 2024, 10:25:00 GMT	3.2 MB	ROM	-
☐	football hd003.png	Wed, 24 Jan 2024, 10:25:02 GMT	3.2 MB	ROM	-
☐	football hd004.png	Wed, 24 Jan 2024, 10:25:01 GMT	3.2 MB	ROM	-
☐	football hd005.png	Wed, 24 Jan 2024, 10:25:02 GMT	3.2 MB	ROM	-
☒	football hd006.png	Wed, 24 Jan 2024, 10:25:01 GMT	3.2 MB	ROM	-
☐	football hd007.png	Wed, 24 Jan 2024, 10:25:00 GMT	3.2 MB	ROM	-
☐	football hd008.png				
☐	football hd009.png				
☐	football hd010.png	Wed, 24 Jan 2024, 10:25:02 GMT	3.2 MB	ROM	-

Job type: Hydrate

4 items

Source Paris

View selection list

Clear selections

Next

Spaces

A Space is an area of storage present on or across one or more pixstor which comprises the following capabilities:

- name
- location
- size
- data protection
- performance
- file share

Viewing Spaces

To view available spaces, click the Spaces menu button.



Navigates to the Spaces screen

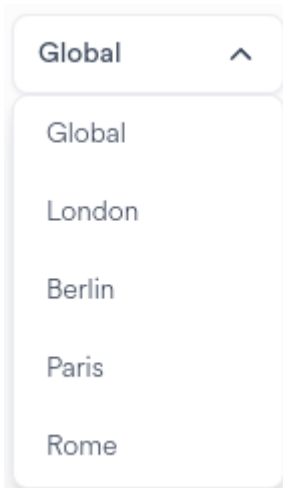
The ability to view spaces is restricted by group membership.

- Administrators can view and administrate all Spaces across all Sites
- Users can view and use all Spaces across all Sites
- Restricted users can view and use a defined subset of Spaces on associated Sites

Hub provides two views of Spaces - Global and Local.

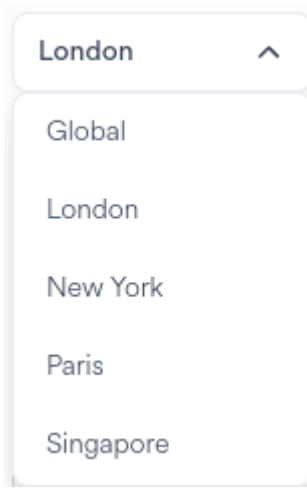
- Global displays all Spaces on all Sites
- Local displays the Spaces on a specific site

The default view of Spaces is Global.



To switch to a site-centric view, select the specific site from the Spaces drop-down menu.

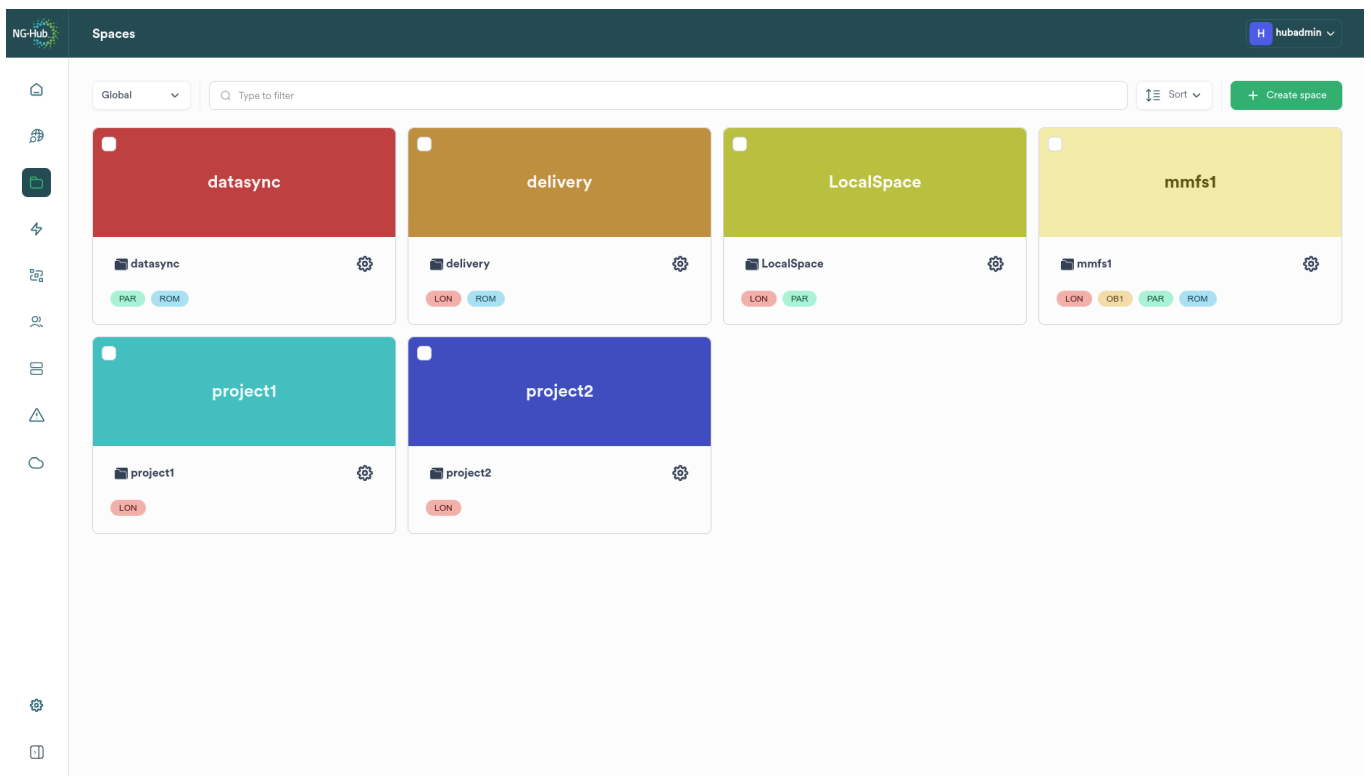
To switch to Global view, select Global from the Spaces drop-down menu.



Global View

Global view displays all Spaces on all Sites.

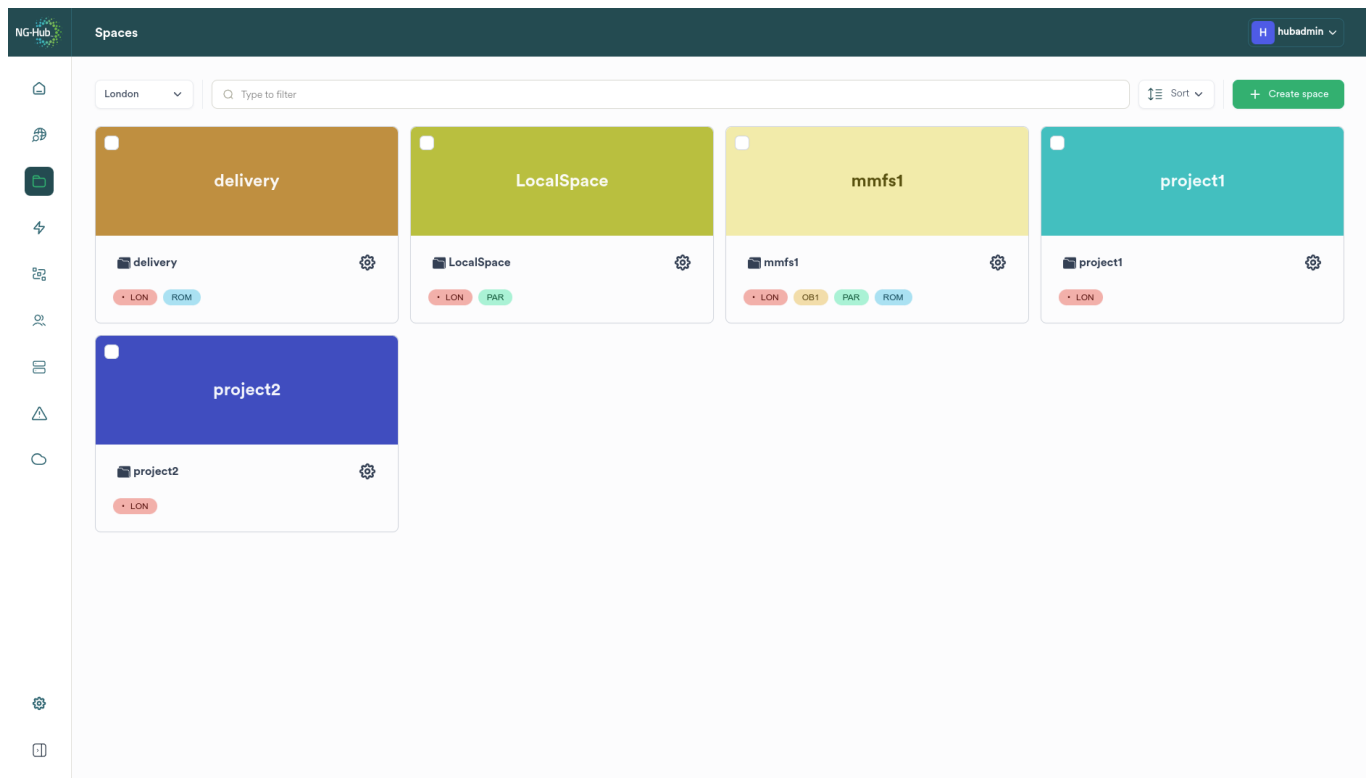
The example below displays all the Spaces across all Sites.



Local View

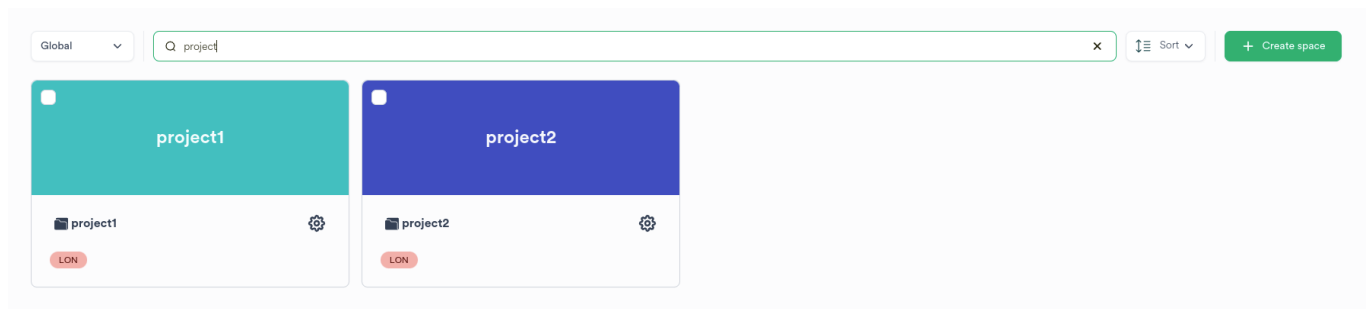
Local view displays the Spaces on a specific Site.

The example below displays less Spaces than Global as the site is participating in fewer Spaces.



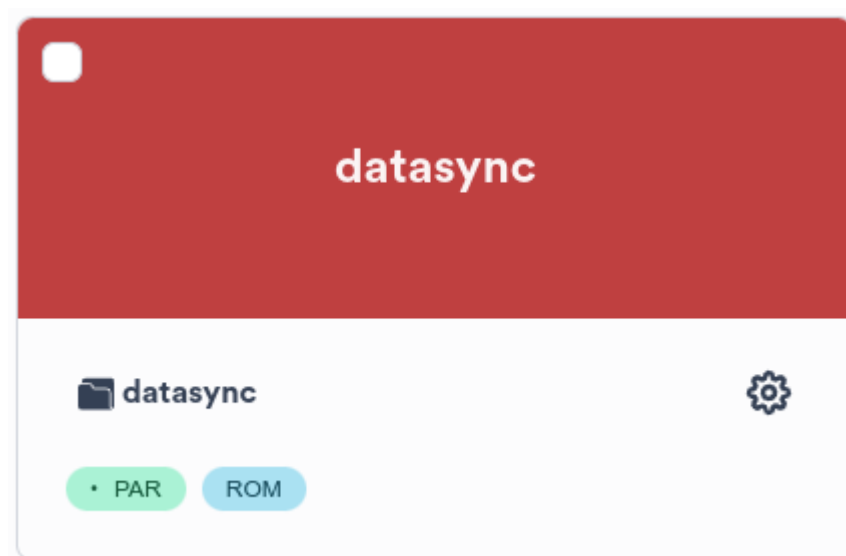
Filtering the Space View

To display Spaces matching keywords, enter the keywords in the filter bar.

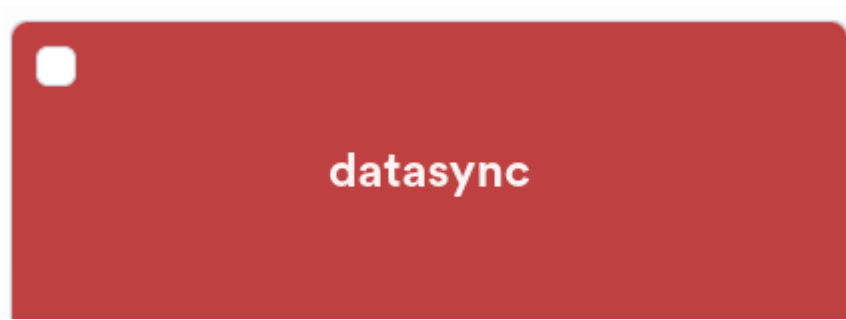


The Space Card

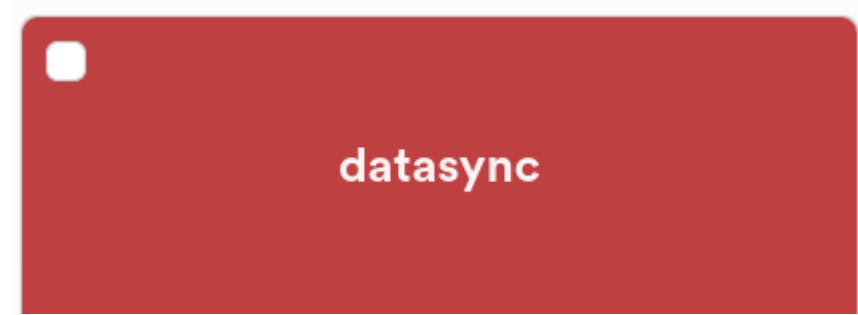
A Space is displayed as a card in the Space view.



A Space Card comprises:



Space Card Header



Denotes the name of the Space.

Click the Header to view the Space File Browser.

Space Name



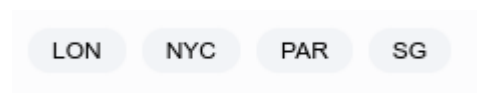
Denotes the name of the Space.
Click the Space Name to view the Space File Browser.

For more information on the Space File Browser, see [Browsing Spaces](#)

Space Site Chips

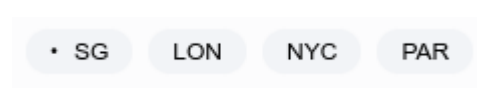
Site Chips denote the sites on which a Space is present.

Click a Site Chip to change the view to the Local View of the [Space File Browser](#) for a specific Site.



In Global view all sites are listed in alphabetical order.

In Local View, the selected site is identified with a dot.



All other sites are listed in alphabetical order after the local site.

Space Selector Checkbox



Selecting the Space Selector Checkbox raises the Job Creator Panel to perform data operations on the entire Space.

Tip: To perform more granular operations on data inside the Space, refer to [Selecting files and folders](#)

The Job Creator Panel allows data operations on the Space as a whole.

Workflow:

Hydrate

View selection list

 |

Clear selections

1

items

Source

london

Queue

default

Next

 >

For further information refer to [Using the Job Creator Panel](#).

Space Settings



Clicking the Settings button on the Space card raises the Settings dialog for the Space.

Basic Space options

datasync

Delete space 

Job schedules

Basic Space options

Shares

Space backup

Size

Size

120

✕

TB ▾

Space colour



Choose a colour to identify the space

Snapshot config

Pick how long you want to store the snapshots for and how often you want snapshots to be taken

Use snapshots



Configure frequency, time & storage duration for snapshots

Job schedules

🔍 Type to filter



Name

Controls



schedule-datasync



Edit



Delete

Save

Important: Adding or Configuring a Space can only be performed by a Hub Administrator or User with Space Administration rights granted through group management.

Browsing Spaces

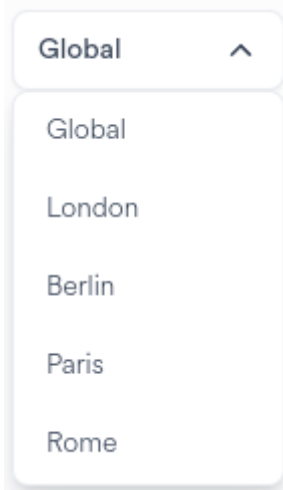
The Spaces File Browser displays the file and folder contents of a Space across multiple or specific sites including additional contextual information such as file counts, size, metadata and status.

The Space File Browser provides two views of Spaces file and folder content - Global and Local.

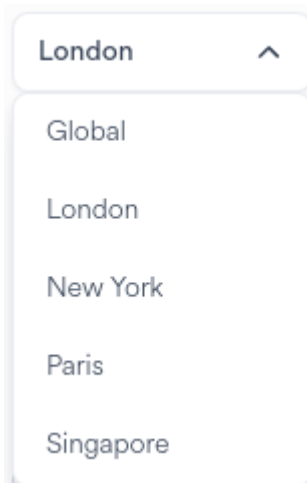
- Global displays the file and folder content for a Space across all Spaces on all associated Sites

- Local displays the file and folder content for a Space on a specific Site

The default view of the Spaces File Browser is Global.



To switch to a site-centric view, select the specific site from the Spaces drop-down menu.

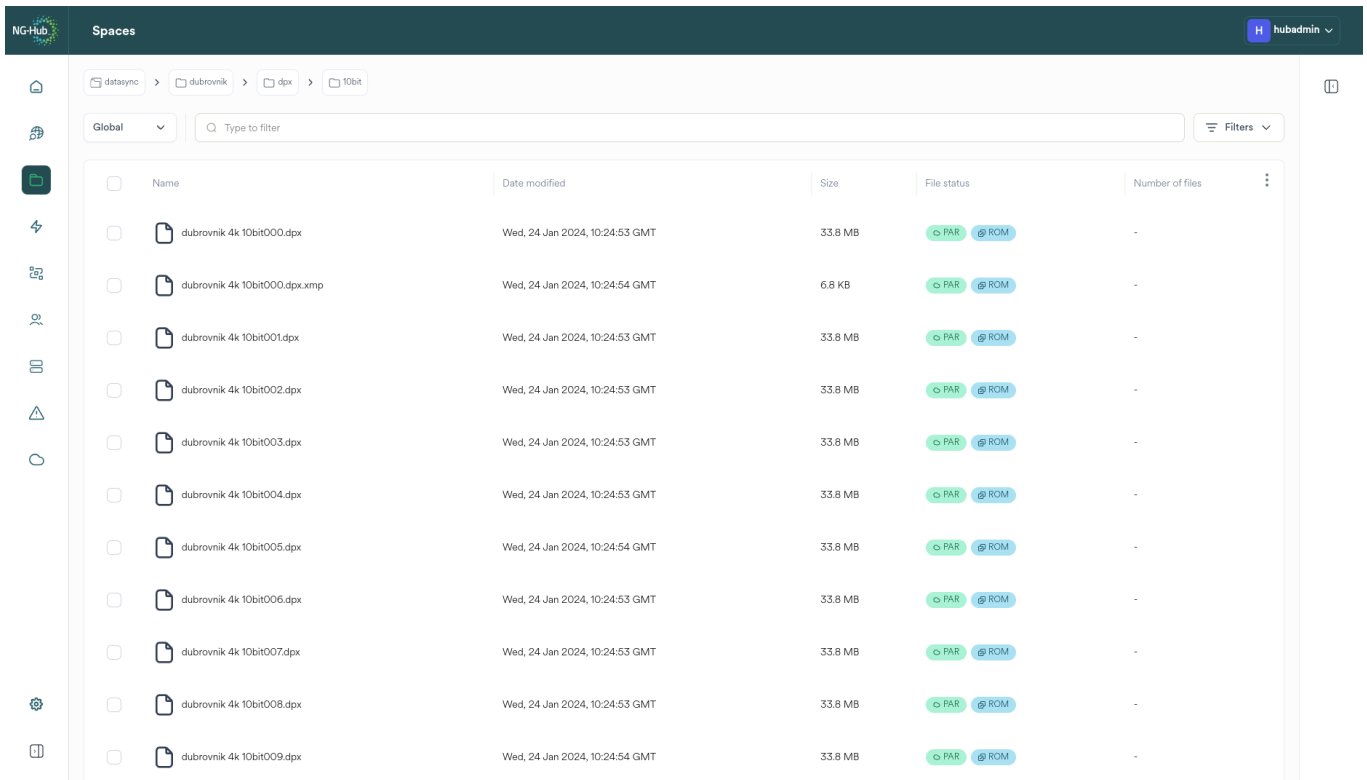


To switch to Global view, select Global from the Spaces drop-down menu.

Spaces File Browser Global View

Global view displays the file and folder content for a Space across all Spaces on all associated Sites

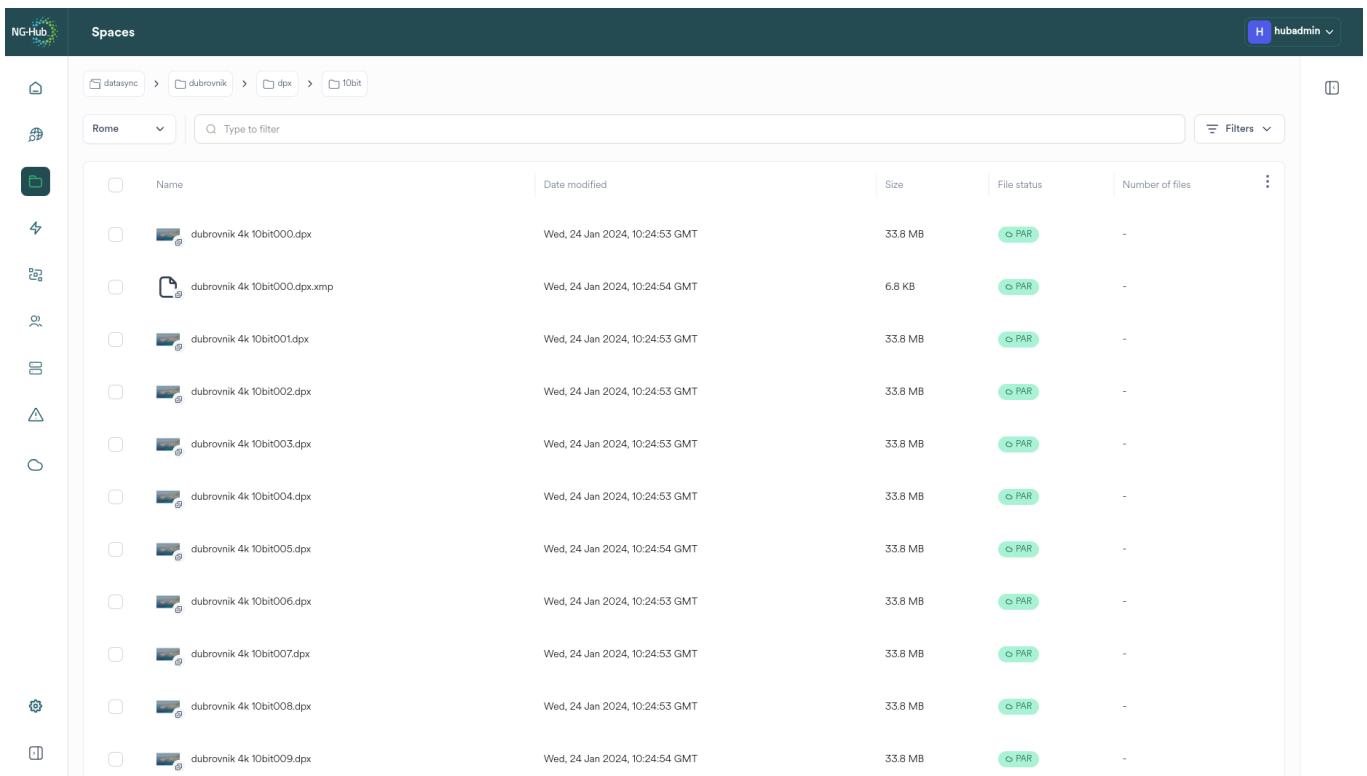
The example below displays the file and folder content for the Spaces across all Sites.



Spaces File Browser Local View

Local view displays the file and folder content for a Space on a specific Site

The example below displays the file and folder content for a Spaces across all Sites with a site-centric view.

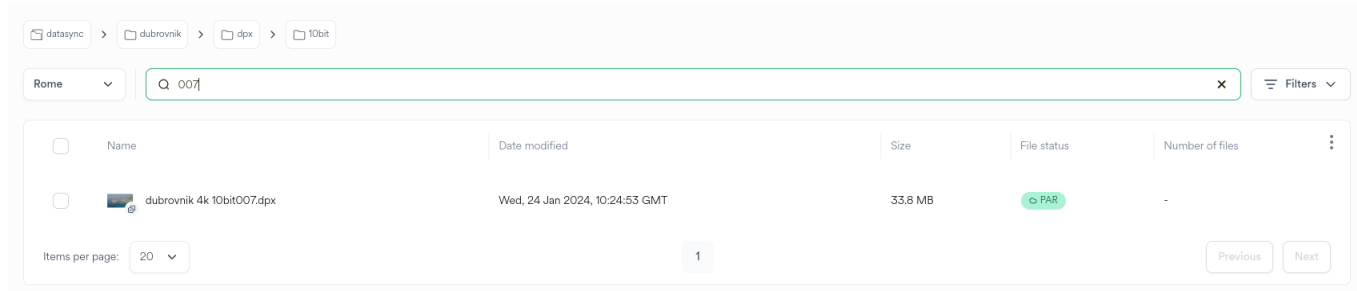


The Space File Browser Screen

The Spaces File Browser screen is comprised of several areas.

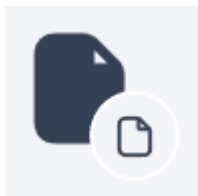
Filtering the Space File Browser

To display files or folders matching keywords, enter the keywords in the filter bar.

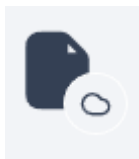


Space File Browser Icon Statuses

In Local View the Spaces File Browser displays additional information to designate the status of the file on the Site's pixstor file system.



Spaces File Browser Local View displays the default file icon if the file or folder is unmanaged. An unmanaged file has not yet been processed by Ngenea operations.



Spaces File Browser Local View displays the cloud icon if the file or folder is Pre-Staged. A pre-staged file is present on the Site's pixstor file system and has an identical copy in the Ngenea target (E.G. AWS cloud bucket).



Spaces File Browser Local View displays the cloud icon if the file or folder is dehydrated. A dehydrated file has only metadata present on the Site's pixstor file system and has a fully hydrated identical copy in the Ngenea target (E.G. AWS cloud bucket).

Space File Browser File Statuses

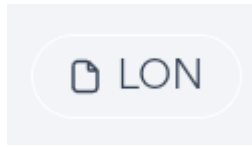
Site Chips denote the sites on which a Space file or folder is present and their status.

In Global view all sites are listed in alphabetical order.

In Local View, the selected site is identified with a dot.

All other sites are listed in alphabetical order after the local site.

Spaces File Browser Local View displays the default file icon if the file or folder is unmanaged. An unmanaged file has not



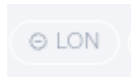
yet been processed by Ngenea operations.



Spaces File Browser Local View displays the cloud icon if the file or folder is pre-staged. A pre-staged file is present on the Site's pixstor file system and has an identical copy in the Ngenea target (E.G. AWS cloud bucket).



Spaces File Browser Local View displays the cloud icon if the file or folder is dehydrated. A dehydrated file has only metadata present on the Site's pixstor file system and has a fully hydrated identical copy in the Ngenea target (E.G. AWS cloud bucket).



Spaces File Browser Local View the file or folder displays a greyed out site with a circled dash where the file is not present on the associated site.

Space File Browser File Attributes

The Space File Browser displays information regarding the files and folders within the Space, globally on all Sites or local to a Site.

Option	Description
Name	The file or directory name
Date Modified	The date and time of last modification
Size	The size of the file on the pixstor file system
File Status	The file status per-site
Number of Files	If the item is a directory, displays the total file count within the directory tree thereunder

Viewing File or Folder Metadata

Metadata Panel

Click a file or directory name to view the associated metadata:



Item info

Name	dubrovnik 4k 10bit000.dpx
Type	file
Path	/mmfs1/data/datasync/dubrov...
Site names	Rome
Size	33.8 MB
Date changed	24.01.2024, 10:24 GMT
Date accessed	24.01.2024, 10:24 GMT
Date modified	24.01.2024, 10:24 GMT

Hydration status

Rome	Premigrated
Paris	Migrated

The metadata panel provides extended metadata, including a summary of the file status on each site participating within a Space.

Where an asset has been ingested via Search thumbnails are displayed for supported file types.

Using the Job Creator Panel

Explain panel areas, buttons and operations

Workflow:

Hydrate

View selection list

 |

Clear selections

1

items

Source

london

Queue

default

Next

 >

Selecting files and directories

Selecting an individual file or directory populates the selection list with the item. Selecting a directory populates all items within the chosen directory tree.

Selections of the entire Space or a directory are each counted as 1 item.

Select the 'select all' checkbox next to the Name field in order to select the entire Space.

—

Name

✓

001.file

✓

002.file

✓

003.file

As selections are added or removed, the selected item count is updated:

5

items

View selection list

Click the View Selection List button to show the files and directories selected to be processed by the chosen workflow

✕ Clear selections

Click the Clear selections to remove all files and directories currently selected from the selection list

Next >

Click the Next button to enact the Job

Sites and queues

Source

london



Select the site where the job will run.

Queue

default



Select the queue the job will be assigned to.

If the workflow runs across multiple sites, such as "Send to Site", select a source site and queue, and a destination site and queue.

Workflow:

➔ Send to Site (hydrated)



View selection list | ✕ Clear selections

1

items

Source

London



Queue

highpriority



Destination

Paris



Queue

default



Submit



Workflow fields

Jobs with Workflows which require additional user decisions prior to enacting the Job raise the Configure workflow fields dialog:

Configure workflow fields



Sync Policy

Newest



Snapdiff rotate on error

Always rotate on error



Sync dirs acl

Sync folder ACL changes



Send

After entering the field information [if required], select the button at the bottom of the dialog to submit the Job.

Send

Click the button at the bottom of the dialog to submit the Job. The button label is changed dependent on the workflow chosen.

Job Types (Workflows)

Workflow: Hydrate



View selection list | Clear selections

1

items

Source

london



Queue

default



Next >

Default Workflows

Dehydrate

The Dehydrate workflow transfers the file metadata and data to an Ngenea target (E.G. an AWS cloud bucket). After dehydration the file appears to be normally present alike any other file on the pixstor file system, but consumes no space. Reading the file automatically hydrates the file with data content allowing the user to read the file as normal.

Pre-Stage

The Pre-Stage workflow transfers the file metadata and data to an Ngenea target (E.G. an AWS cloud bucket). After migration the file on Ngenea target is an identical instance of the file on the pixstor file system. The file on the pixstor file system is not dehydrated. Reading the file allows the user to read the file as normal. Pre-Staging can reduce the total time to Dehydrate the same data in future.

Hydrate

The Hydrate workflow retrieves the file data from an Ngenea target (E.G. an AWS cloud bucket). After successful transfer the metadata and data content of the file is present on the destination site. Reading the file allows the user to read the file as normal.

Sync Space to Site

The Sync Space to Site uses file system snapshots to discover changes between the last file system snapshot and when the workflow was run. Changes are applied by sending newly created or recently modified files and directories, including deleting or moving files or directories in place on the target site as necessary to match the source site.

Send to Site (hydrated)

The Send to Site (hydrated) workflow transfers data from a source site to a destination site. After successful transfer the metadata and data content of the file is present on the destination site. Reading the file allows the user to read the file as normal.

Send to Site (dehydrated)

The Send to Site (dehydrated) workflow transfers data from a source site to a destination site. After successful transfer the metadata of the file is present on the destination site. To a user the file appears to be normally present alike any other file on the pixstor file system. Reading the file automatically hydrates the file with data content allowing the user to read the file as normal.

Additional Custom Workflows

Hub can support additional Custom Workflows providing custom operations to data through multiple task steps. Custom workflows are configured and provisioned by a Hub Administrator using the Hub CLI. When provisioned Hub Custom Workflows will appear in the list of workflows available for users to use with their data.

Managing Spaces

Create a Space Wizard

Click the Create Space button to display a dialog to configure the selected space.

+ Create space

Important: This function can only be performed by a Hub Administrator

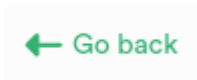
Navigating the Wizard



Click the close button to exit the wizard. Changes are not saved.



Click the Next button to advance to the next page of the wizard. The Next button is disabled until all required page elements are completed.



Click the Go Back button to return to the previous wizard page.



Click the Finish & Create button to apply the changes displayed on the wizard summary page.

Basic Space Options

Spaces are created on pixstor file systems at `/mmfs1/data/[Space]` or at an alternative custom location.

Spaces can be restricted to a limited amount of data.

Spaces are defined to a specific file system.

Name & Size

Name

Type the space name

Name is required

Size

0

TB ▾

Filesystem

Choose a filesystem ▾

Space colour



Choose a colour to identify the space

- Enter a name of the Space. The name is case-sensitive.
- Choose the data limit for the Space or unrestricted (Size = 0)
- Select the file system the Space will reside on across all pixstors
- Select a colour for the Space Card

Snapshots create safety copies of data on the pixstor file system.

Snapshots are not backups and should not be used as protection against media failures

Snapshot config

Enable automatic snapshots 

Frequency

daily 

Time

00  : 50  UTC+0

Snapshot retention

1 week 

- Choose whether snapshots are required for the Space
- Choose the frequency of the snapshot
- If the frequency is daily or longer, select the time of day in UTC+0 on the Site to snapshot the Space
- Choose how long to retain the snapshots

Select the Sites to create this Space on

A Space can exist across multiple Sites.

By default data does not move across multiple Sites. Scheduled workflows must be set up to enable cross-site data movement.

Hub provides the ability to synchronise a Space with two Sites - a bi-directional sync.

☐	Name	Tier	Free/Total	
☒	London	nmve	73 / 240 TB	
☐	ob1	Select an option	No data available	
☒	Paris	nearline	49.4 TB / 50 TB	
☒	Rome	nmve	216 TB / 1.43 PB	

Items per page: 20

- Select the Sites on which to create the Space
- Select the pixstor file system tier on which to host the files for the Space. The tier is limited to those available for the file system selected in the prior wizard screen.

If two Sites are selected the Wizard will prompt to setup a bi-directional sync.

Location

Use default location
☒

Create the space at /mmfs1/data/myspace

- To select an alternative custom location, deactivate the Use default location slider

[Pick location on filesystem ↗](#)

Click the Pick location on filesystem button to open the location selector

Select location



mmfs1



data

London



Type to filter

Name

Size

Number of files

Date created



delivery

512 B

-

Tue, 23 Jan 2024, 15:50:49 GMT



LocalSpace

512 B

-

Wed, 24 Jan 2024, 15:18:54 GMT



project1

4.8 GB

504

Tue, 23 Jan 2024, 15:50:49 GMT



project2

5.5 GB

281

Thu, 29 Feb 2024, 22:00:49 GMT

Items per page:

20



1

Previous

Next

Confirm

- Select a Site to browse. Parent directories not present on another Site will be added when a Space is created.
- Navigate to the directory under which to create the Space
- Click the Confirm button to return to the Space creation wizard

The chosen location is displayed and will be used for all Sites on which the Space is created.

Set up bi-directional sync

If two Sites are selected on the prior wizard page, Hub prompts to setup a bi-directional sync.

Only one bi-directional sync can run at a time.

This page can be skipped if no synchronisation is required.

Source

London

highpriority

Destination

Paris

default

- Select the source Site to launch the first sync from
- Select the other Site as the destination site
- Select the Queue(s) to assign bi-directional sync to

Schedule

Determine the required frequency of the synchronisation.

Schedule

Frequency

☒ Mins or Hours
 ☐ Daily or Weekly

Interval

6

×

hours

▼

Choosing Mins or Hours will ensure that the schedule will run on the next interval set.

E.G: * 1 hour: The synchronisation will run on the next hour (12.00, 13.00) * 15 mins: The synchronisation will run on the next 15 minute interval past the hour (15, 30, 45, 00)

Schedule

Frequency

- ☐ Mins or Hours
- ☒ Daily or Weekly

Every

☒ Mon ☐ Tue ☒ Wed ☒ Thu ☐ Fri ☒ Sat ☒ Sun

At : UTC+0

Choosing Daily or Weekly allows the synchronisation to be scheduled once per chosen day at a specific time of day.

Hint: The schedule time is in UTC+0. You may need to account for any timezone offset of the site when scheduling.

Shares

Prior to Space creation Shares can be assigned.

See [Shares](#) for more details

Spaces backup

Space backup

Configure backups of this space

Site(s) to backup

Select sites

Global exclusion rules

Exclude filetypes

Type the file extension and hit Return key

- Select the Sites which will perform backups for this space

- Define file extensions which must be excluded from backups across all Sites.
E.G. *.tmp

Choosing Site(s) to backup presents further backup options.

For more information refer to [Backups](#).

Important: Multiple Sites are not permitted to backup to the same Ngenea Target.

Deleting a Space



Clicking the Settings button on the Space card raises the Settings dialog for the Space.

Space settings



Basic Space options

myproject

Delete space

Job schedules

Basic Space options

Shares

Space backup

Size

Size

120



TB

Space colour



Choose a colour to identify the space

Delete space

Clicking the Delete Space button raises a confirmation dialog

Delete the space

×

Deleting the Space will:

Remove this Space from Sites: **london, paris**

Remove SMB shares: **myproject**

Remove where this is the sole associated Space:

Policies: **archive_over_30days**

Where a Space on a Site contains data or has Snapshots, Space deletion will:

- fail-safe leaving data and Snapshots intact
- remove the Space from NG-Hub management

Confirm you wish to delete the space **"myproject"**

This process cannot be undone.

To confirm, enter the name of the space below and press the confirm button.

Confirm

Cancel

The delete confirmation dialog describes what will happen when the space is deleted, including what associated objects will also be removed.

Type the name of the space into the text input to activate the Confirm button.

Confirm you wish to delete the space **"myproject"**

This process cannot be undone.

To confirm, enter the name of the space below and press the confirm button.

myproject

×

Confirm

Cancel

Cancel

Click the
Cancel
button to

close the confirmation dialog. The space will not be deleted.

Click the Confirm button to delete the space.



Shares

Prior to Space creation Shares can be assigned.

Shares are created across all Sites where the Space is provisioned.

As the Space is not yet created any Share created is assigned to the whole Space. To create Shares within the Space, edit the Space after creation and add the required Shares.



To create shares for directories within a Space can be created via the Edit Space button after successful Space creation. Shares are created identically across all Sites participating in a Space.

☐ Read only

☒ Enable multi-threaded reads

☒ Enable multi-threaded writes

☒ Locking for root share

☒ HSM Support

☐ Guest OK

Select the available extended SMB options as appropriate:

Option	Description
Read only	If enabled then users of the Share may not create or modify files in the Space.
Enable multi-threaded writes	Enable asynchronous reads and writes
Locking for root share	Enforce file locking
HSM support	Support Ngenea operations to data within the Space
Guest OK	Access will be permitted as the default guest user

SMB Custom Options

On occasion Administrators may find it necessary to further tailor SMB Shares in order to provide additional capabilities E.G. compatibility or performance tunings.

Pixstor supports additional SAMBA Custom Options per-share. Custom Options defined for a Space's Share are applied across all Sites where the Space is provisioned.

Important: SMB Custom Options are applied verbatim to the underlying SAMBA configuration. Incorrect entries can result in service outage.

Add a new Custom Option as follows:

+ Add new key

Click the Add new key to define a new custom option

 **Keyword**
.....

Enter the name of the custom option setting in the Keyword field

String

Numeric

Boolean

Select the type of custom option from the drop down menu. Choose or enter the value for the custom option.

Example of added Custom Options:

 **fullaudit**
.....



String



success



 **level2 oplocks**
.....



Boolean



False



+ Add new key

To modify an existing custom option change the keyword, type or values as appropriate.

To delete a custom option select the bin icon next to the item.



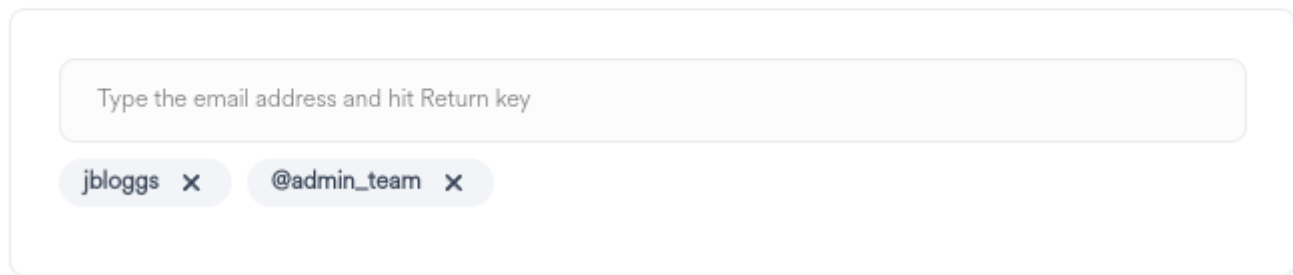
Click the delete button to remove a custom option

Admin Users

Granted full permission to data in the Space to a specific set of users or groups:

Admin users

Users (username) or groups (@groupname) which will always have full control of all files.

A screenshot of a user input interface. At the top is a text box with the placeholder text "Type the email address and hit Return key". Below the text box are two tags: "jbloggs" followed by a close icon (X), and "@admin_team" followed by a close icon (X).

- Add users by username
- Add groups by prefixing the group name with an @ character

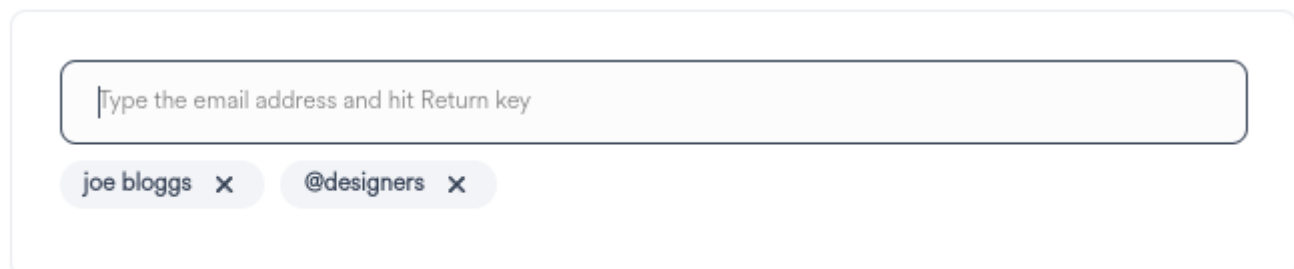
Hint: For users or groups containing spaces or symbols, etc. use quotes. E.G. @"Domain Admins"

Allowed Users

Restrict the access to the share to a specific set of users or groups:

Allowed users

Users (username) or groups (@groupname) which are permitted to connect to the SMB share.

A screenshot of a user input interface. At the top is a text box with the placeholder text "Type the email address and hit Return key". Below the text box are two tags: "joe bloggs" followed by a close icon (X), and "@designers" followed by a close icon (X).

- Add users by username
- Add groups by prefixing the group name with an @ character

Hint: For users or groups containing spaces or symbols, etc. use quotes. E.G. @"Domain Admins"

Force permissions

In some scenarios it may be desirable to force the file and directory permissions to specific values in order to create a consistent known permission model. Typically this is observed where systems are not connected to Identity Management services such as Active Directory, LDAP or similar.

Enable force permissions



Type	Read	Write	Execute	
User	☒	☒	☒	
Group	☒	☐	☐	
Other	☒	☐	☐	

- Set the permission overrides as required

Host access control

pixstor provides the ability to limit connectivity to specific network clients.

1. If no allow or deny options are defined pixstor will allow connections from any system.
2. If only a hosts allow option is defined for a share, only the network clients listed will be allowed to use the share. All others will be denied.
3. If only a hosts deny option is defined for a share, any network client which is not listed will be able to use the share.
4. If both a hosts allow and hosts deny option are defined, a network client must appear in the hosts allow and not appear in the hosts deny to access the share.

Hosts allow

Hosts to allow

10.20.30.41

×

🗑

27.33.111.62

×

🗑

+

Add host

- Add the required hosts by specifying the IP address

+ Add host

Click the Add host button to add additional hosts



Click the delete button to remove a host

Hosts deny

Hosts to deny

126.211.200.89



86.22.11.72



+ Add host

- Add the required hosts by specifying the IP address

+ Add host

Click the Add host button to add additional hosts



Click the delete button to remove a host

NFS Advanced

Extended NFS options

Modifying the extended NFS options controls the Share capabilities:

NFS Advanced settings



Extended NFS options

Extended NFS options

ID mapping

Additional NFS share options

NFS network
restrictions

- ☐ Read only
- ☐ Asynchronous
- ☒ Write delay
- ☒ Secure ports
- ☒ Subtree check

ID mapping

Save

Select the available extended SMB options as appropriate:

Option	Description
Read only	If enabled then users of the Share may not create or modify files in the Space.
Asynchronous	Enable asynchronous reads and writes
Write delay	Reply to I/O requests only after the changes have been committed to stable storage at a cost of performance reduction.
Secure ports	Requires that NFS requests originate from a TCP/IP port from 1-1024
Subtree check	Check the accessed file is in the appropriate filesystem and also within the Share

ID Mapping

All Squash

All Squash maps all User IDs (UIDs) and group IDs (GIDs) to the anonymous user. This is useful for NFS-exported public FTP directories, news spool directories

☒ All squash (map all uids and gids)

User ID

65534 ×

Group ID

65534 ×

Root Squash

Root squash allows the root user on the client to both access and create files on the NFS server as root. This is conceptually equivalent to the Administrator in Windows.

Root Squash is needed if you are hosting root filesystems on the NFS server (E.G. for diskless clients). You should not use `no_root_squash` unless you are aware of the underlying implications.

☒ Root squash (map requests from uid/gid 0 only)

User ID

65534 ×

Group ID

65534 ×

No Squash

No Squash allows the root user on the NFS client host to access the NFS-mounted directory with the same rights and privileges that the superuser would normally have.

☒ No squash (no uid/gid mapping)

NFS network restrictions

pixstor provides the ability to limit connectivity to specific network clients.

☒ Available to all clients

- Select the Available to all clients option for no restrictions

Restrict to hosts

☒ Restrict to hosts

Type in IP address or hostname

+ Add host

- Add required hosts to restrict by specifying the IP address or the FQDN hostname

Restrict to network range

☒ Restrict to network range

Range 1

IP address

CIDR

10.176.34.0

x

24

x

Range 2

IP address

CIDR

179.63.16.8

x

32

x

+ Add IP range

- Add the required IP address or network range and specify a valid CIDR mask to apply the restriction

+ Add IP range

Click the Add IP range button to add additional restrictions

Click the delete button to remove a host

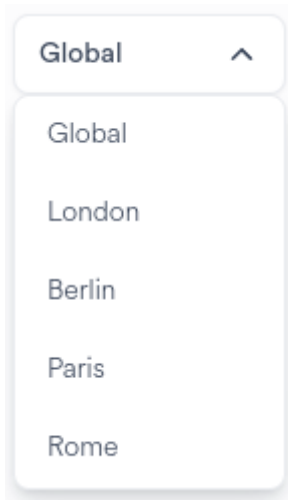
Jobs

A job comprises one or more tasks, each of which perform an action.

Tasks can be data orientated (E.G. hydrate, dehydrate, SendToSite) or can be management or configuration tasks of pixstor sites and/or services.

Viewing Jobs

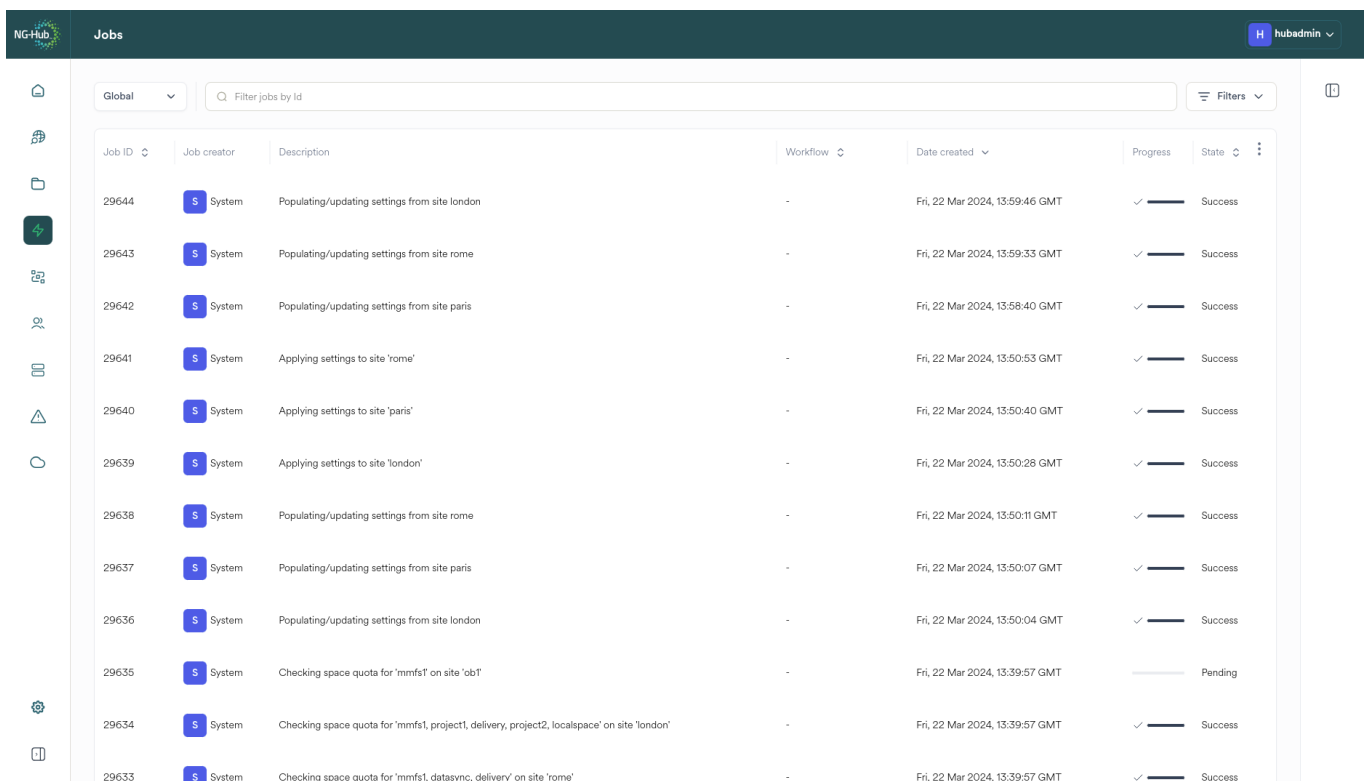
To view Jobs executed on a specific Site, select the Site from the Sites drop down menu or choose Global to display all Jobs from all Sites.



Click and select the site to show the jobs from the selected site.

After selection the Jobs from the selected Site are displayed including high level information for each Job.

Click a Job ID to view in depth information for the job.



Job ID	Job creator	Description	Workflow	Date created	Progress	State
29644	System	Populating/updating settings from site london	-	Fri, 22 Mar 2024, 13:59:46 GMT	✓	Success
29643	System	Populating/updating settings from site rome	-	Fri, 22 Mar 2024, 13:59:33 GMT	✓	Success
29642	System	Populating/updating settings from site paris	-	Fri, 22 Mar 2024, 13:58:40 GMT	✓	Success
29641	System	Applying settings to site 'rome'	-	Fri, 22 Mar 2024, 13:50:53 GMT	✓	Success
29640	System	Applying settings to site 'paris'	-	Fri, 22 Mar 2024, 13:50:40 GMT	✓	Success
29639	System	Applying settings to site 'london'	-	Fri, 22 Mar 2024, 13:50:28 GMT	✓	Success
29638	System	Populating/updating settings from site rome	-	Fri, 22 Mar 2024, 13:50:11 GMT	✓	Success
29637	System	Populating/updating settings from site paris	-	Fri, 22 Mar 2024, 13:50:07 GMT	✓	Success
29636	System	Populating/updating settings from site london	-	Fri, 22 Mar 2024, 13:50:04 GMT	✓	Success
29635	System	Checking space quota for 'mmfs1' on site 'obf'	-	Fri, 22 Mar 2024, 13:39:57 GMT		Pending
29634	System	Checking space quota for 'mmfs1, project1, delivery, project2, localspace' on site 'london'	-	Fri, 22 Mar 2024, 13:39:57 GMT	✓	Success
29633	System	Checking space quota for 'mmfs1, datasync, delivery' on site 'rome'	-	Fri, 22 Mar 2024, 13:39:57 GMT	✓	Success

Filtering the Jobs View

To display a specific Job ID, enter the ID in the filter bar.

Global
Filter jobs by Id
Filters
1

Filters
Hide noop
Date created
State
Queue
Workflow
Job creator
Workflow: Bi-directional Sp...
Clear all

Job ID	Job creator	Site	Description	Workflow	Date created	Progress	Queue	State
87468	schedule-sync-test	London	Bi-directional space sync of '/mmfs1/data/sync-...	Bi-directional space sync	Thu, 22 Aug 2024, 17:14:00 GMT+1	✓	-	Success
87467	schedule-sync-test	London	Bi-directional space sync of '/mmfs1/data/sync-...	Bi-directional space sync	Thu, 22 Aug 2024, 17:13:00 GMT+1	✓	-	Success

Jobs for a site can be additionally filtered through selection of various criteria.

Filters
2

Hide noop

Date created

Click the filter button to display the available filters. The number of applied filters is displayed on the Jobs filter button.

Select the Hide noop filter option to hide any job which did not perform any action as the job determined that no actions were required.

Click the Date Created button to filter for any job which was created between two date ranges.

Clicking the Date Created button raises the date selection dialog. Choose the criteria required and click the Save button to apply the date range.

Choose a date range
X

Today
Yesterday
This Week
Last Week
This Month
Last Month
- days up to today
- days starting today

Mar 5, 2024
Mar 16, 2024

March
2024

Mar 2024

Apr 2024

Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat
25	26	27	28	29	1	2	31	1	2	3	4	5	6
3	4	5	6	7	8	9	7	8	9	10	11	12	13
10	11	12	13	14	15	16	14	15	16	17	18	19	20
17	18	19	20	21	22	23	21	22	23	24	25	26	27
24	25	26	27	28	29	30	28	29	30	1	2	3	4
31	1	2	3	4	5	6							

Save

51

State ▼

Click the Job State button to display jobs matching the status. Available statuses are:

Status	Description
New	A job has been created
Pending	The job is waiting to run
Started	The job is running
Succeeded	The job finished successfully
Error	The job finished with one or more error conditions
Failure	The job finished with one or more failure conditions
Skipped	The job was skipped as the work assigned to the job was not required to be undertaken - no change would have occurred if the job had run.
Cancelled	The job was cancelled
Cancelling	The job is in the process of cancelling
Pausing	The job is in the process of Pausing
Unknown	The job experienced a result which could not be matched to a Status

Queue ▼

Click the Queue button to filter for jobs matching the Queue.

Workflow ▼

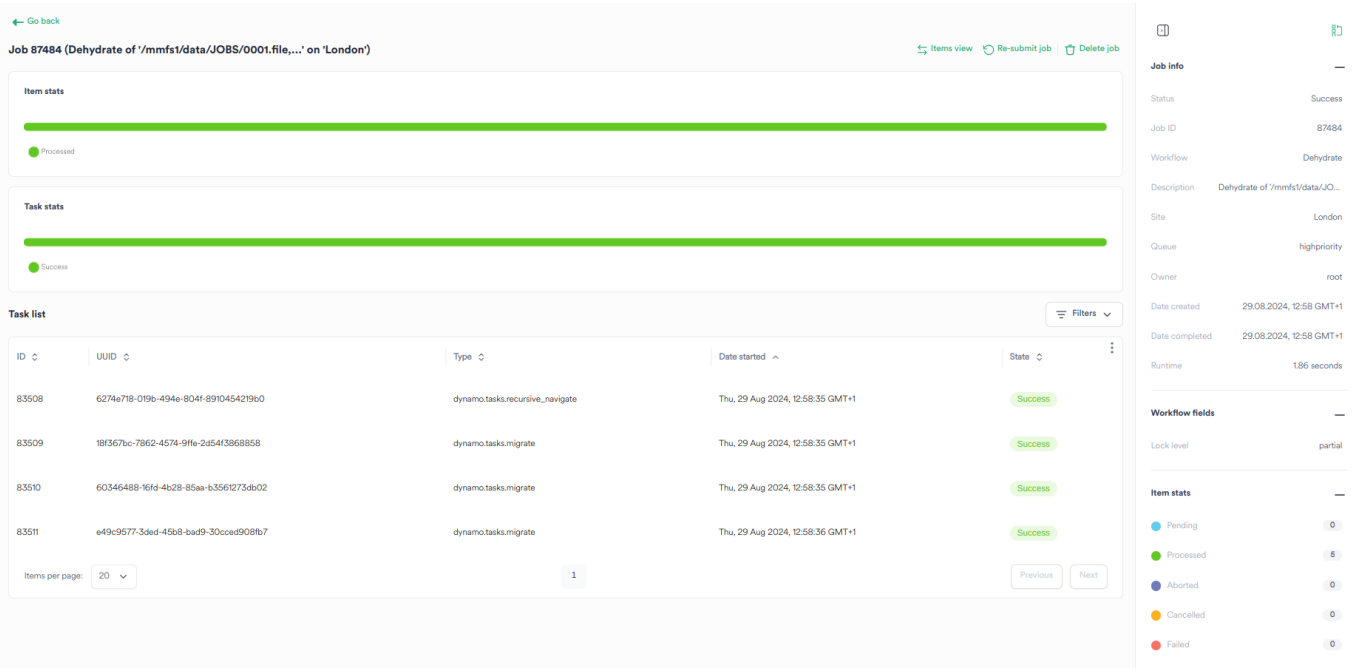
Click the Workflow button to filter for jobs matching the Workflow. Available workflows are:

Type	Description
Bi-Directional Sync	Data synchronisation between two sites
Delete File	Deletion of a designated data
Dehydrate	Dehydration of data
Pre-Stage	Staging of data to an Ngenea target. Subsequent dehydrations save time as there is no requirement to stage data prior to dehydration.
Hydrate	Hydration of data
Reverse Stub	Creation of dehydrated files not prior existing which reference data in an Ngenea target
Send	Deliver data from a source to a destination site
Send to Site (hydrated)	Deliver data from a source to destination site as hydrated files
Send to Site (dehydrated)	Deliver data from a source to destination site as dehydrated files

Type	Description
Sync Space to Site	Synchronising data from a source to a destination site
Transparent Recall	User or application initiated hydration of data on reading
Job creator ▾	Click the Job creator button and select the name of one or more creator(s) to view jobs by the selected creator(s)

Viewing a Job

Clicking a Job ID in the main Job screen displays the in depth information for the job.



The Job View comprises:

Task stats

The total count of types operation result per task is represented by the horizontal bar segments.

Hovering over the bar provides the count of each task status for the tasks processed.



File/Item stats

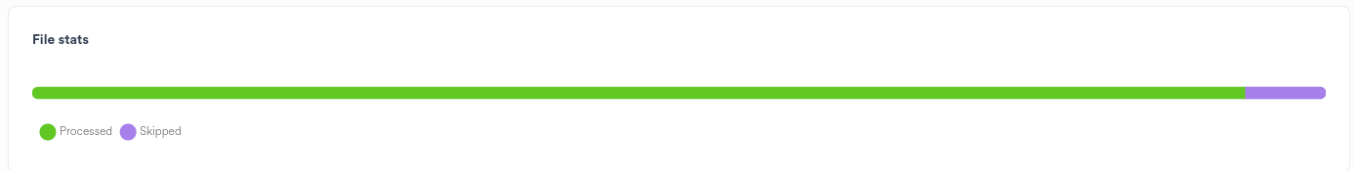
The total count of types operation result per item is represented by the horizontal bar segments.

Items may be files, directories or Objects.



Hovering over the bar provides the count of each type processed.

Bi-Directional Sync and Site Sync workflows only provide File stats.



The Job Side bar

Enlarge/Reduce



Clicking the toggle displays the Job Side bar to show:

- Job info
- Task stats
- File Stats (Item stats)

Clicking the toggle button again hides the Job Side bar.

Reordering

Re-arrange items in top to bottom order by dragging vertically up or down.

Side bar organizer



Job info

Job details



Task stats

Job task statistics



File stats

Job file statistics

Save

Job Controls

Jobs provide the following controls where supported by the Workflow:

 Pause job

If supported by the Workflow, click the Pause job button to pause the Job. Tasks which are not running will be paused.

 Resume job

If supported by the Workflow, click the Resume job button to resume a paused Job.

 Cancel job

Click the Cancel Job button to cancel any future tasks from running and no longer proceed with the Job

 Re-submit job

Click the Re-submit job button to launch an identical Job on the Job queue

 Delete job

Click the Delete Job button to remove the records of the Job and associated tasks from the Hub database

Job Info Summary

Provides an overview of high level information of the Job.

Job info

Status	Success
Job ID	87468
Workflow	Bi-directional space sync
Description	Bi-directional space sync of '/...
Site	London
Queue	highpriority
Owner	schedule-datasync
Date created	22.08.2024, 17:14 GMT+1
Date completed	22.08.2024, 17:14 GMT+1
Runtime	15.9 seconds

Job Workflow Fields

Provides a listing of any workflow fields the job was executed with.

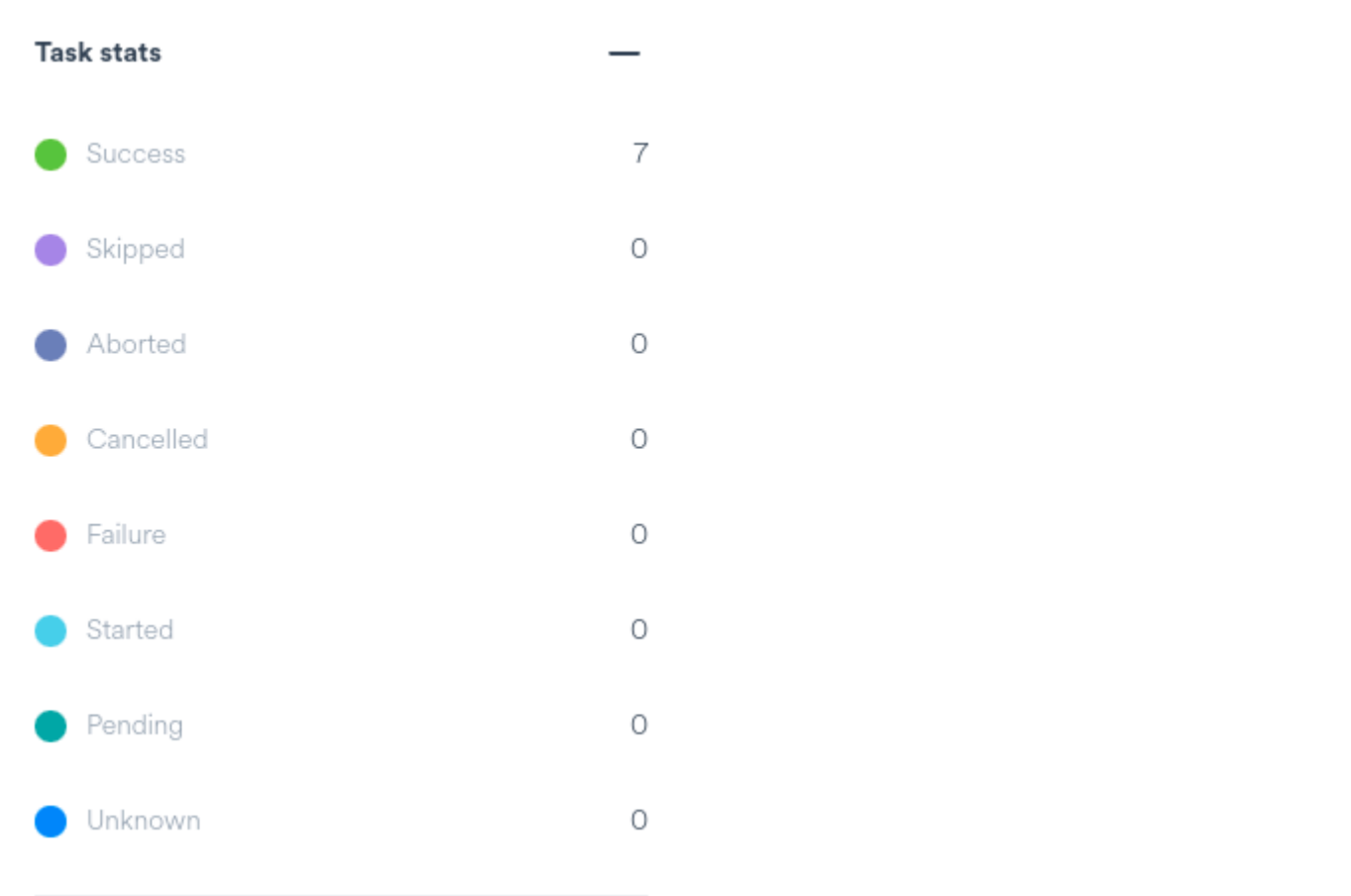
Workflow fields

Hydrate	
Destinationsite	paris
Sync preference	newest
Snapdiff rotate on error	

Job Info Task Stats

Jobs undertake tasks to process one or more actions, files, directories or objects as defined by the Workflow.

The Task stats displays the total count of types operation result per task.



Item stats

The total count of types operation result per item is represented by the horizontal bar segments.

Items may be files, directories or Objects.

Hovering over the bar provides the count of each operation type for the files processed.

Item stats	
 Pending	0
 Processed	240
 Aborted	0
 Cancelled	0
 Failed	0

File stats

Bi-Directional Sync and Site Sync workflows only provide File stats.

The File stats displays the total count of types operation result per file or directory.

File stats	
 All	1234
 Pending	0
 Processed	1234
 Skipped	0
 Aborted	0
 Cancelled	0
 Failed	1

The Job Task List

The Job details page is viewed with a `task-based view` or with a `item-based view`.

Switching between views can be done by clicking `Tasks view` or `Items view` on the top right corner of Job Tasks page.

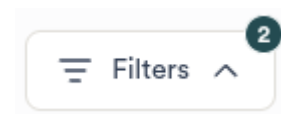
Tasks View

The Task view displays all tasks comprising a job, their ID, type, start time and status.

ID	Type	Date started	State
c9db871a-154c-467e-9136-fb56cc8677f0	dynamo.tasks.recursive_action	Thu, 17 Aug 2023, 12:16:03 GMT+1	Success
f2e1e161-816f-403f-84ff-fb789498db7d	dynamo.tasks.reverse_stub	Thu, 17 Aug 2023, 12:16:08 GMT+1	Success
99d4e4c4-c84a-4121-90ef-4cd58bb75fe1	dynamo.tasks.migrate	Thu, 17 Aug 2023, 12:16:08 GMT+1	Success
c9b294c6-5a0b-4ebf-8fcc-f695b4cb0913	dynamo.tasks.remove_location_xattrs_for_moved	Thu, 17 Aug 2023, 12:16:08 GMT+1	Success
flb5c1b1-2bcd-426b-93ed-69131757b355	dynamo.tasks.recursive_action	Thu, 17 Aug 2023, 12:16:08 GMT+1	Success

Filtering the Job Task List

Tasks for a job can be additionally filtered through selection of various criteria.



Click the filter button to display the available filters. The number of applied filters is displayed on the Task list filter button.



Click the State button to display jobs matching the status. Available statuses are:

Status	Description
New	A task has been created
Pending	The task is waiting to run
Started	The task is running
Succeeded	The task finished successfully
Error	The task finished with one or more error conditions
Failure	The task finished with one or more failure conditions
Skipped	The task was skipped as the work assigned to the job was not required to be undertaken - no change would have occurred if the job had run.
Cancelled	The task was cancelled
Cancelling	The task is in the process of cancelling
Aborted	

Status	Description
	Whilst running, the task was unable to proceed correctly and aborted the in progress action
Paused	The task is paused
Pausing	The task is in the process of pausing
Unknown	The task experienced a result which could not be matched to a Status

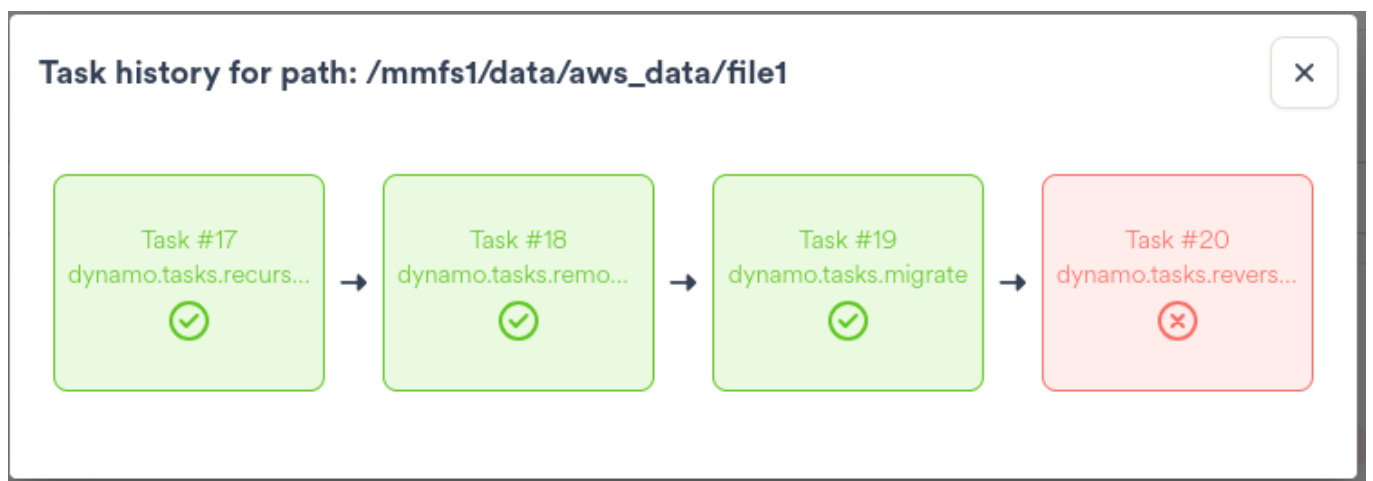
Click the Task type button to filter for a specific task type which comprises the Job. The task types are dynamic therefore the displayed types may differ per job type.

Task type ▼

Items View

The Item view displays the object paths for all tasks across the job. When the **Failed items** tab is selected, the error message is displayed for failed items.

Clicking on a path displays the task history across the job.



Clicking on each task in task history displays the task details for the chosen task ID.

Output for task #18

```
"root" : { 13 items
  "url" : "http://10.201.2.211:8000/api/tasks/18/"
  "id" : 18
  "task_id" : "20f366b8-9ddf-400a-a720-35be4f27d065"
  "tasktype" : "dynamo.tasks.remove_location_xattrs_for_moved"
  "state" : "SUCCESS"
  "started" : "2024-04-19T13:29:26.603025Z"
  "completed" : "2024-04-19T13:29:26.655133Z"
  "runtime" : 0.052108
  "job" : 9
  "site" : "site1"
  "paths" : "http://10.201.2.211:8000/api/tasks/18/files/"
  "result" : { 7 items
    "log" : [] 0 items
```

Filtering the Items List

To search for a specified item, enter the path in the filter bar. It will only search paths based on the state of the active tab out of the Queued/Failed/Completed options.

Queued itemsFailed itemsCompleted items

Q Type to filter by path

Path	Message	
/mmfs1/data/aws_data/file2	-	
/mmfs1/data/aws_data/file1	-	
/mmfs1/data/aws_data/file3	-	

Items per page: 20 1PreviousNext

Task information Dialog

Clicking a task ID in the Job Task List displays the information for the chosen task.



```

"root" : { 18 items
  "url" : "http://hub:8000/api/tasks/c9db871a-154c-467e-9136-fb56cc8677f0/"
  "taskid" : "c9db871a-154c-467e-9136-fb56cc8677f0"
  "tasktype" : "dynamo.tasks.recursive_action"
  "state" : "SUCCESS"
  "started" : "2023-08-17T11:16:03.726630Z"
  "completed" : "2023-08-17T11:16:07.809412Z"
  "runtime" : 4.082782
  "numfiles" : 10
  "numprocessedfiles" : 0
  "numskippedfiles" : 0
  "numabortedfiles" : 0
  "numcancelledfiles" : 0
  "numfailedfiles" : 0

```

Dependent on the number of operations and the quantity of inputs to the task, the information displayed can range from short to extensive.

Optionally select an action button to more easily view the Task Information.

	Click the Full Screen button to display the Task Information in a larger view
	Click the Copy button to copy the Task Information output to the clipboard.
	Click the Download button to download the Task Information locally
	Click the Close button to close the Task Information dialog

Scheduling

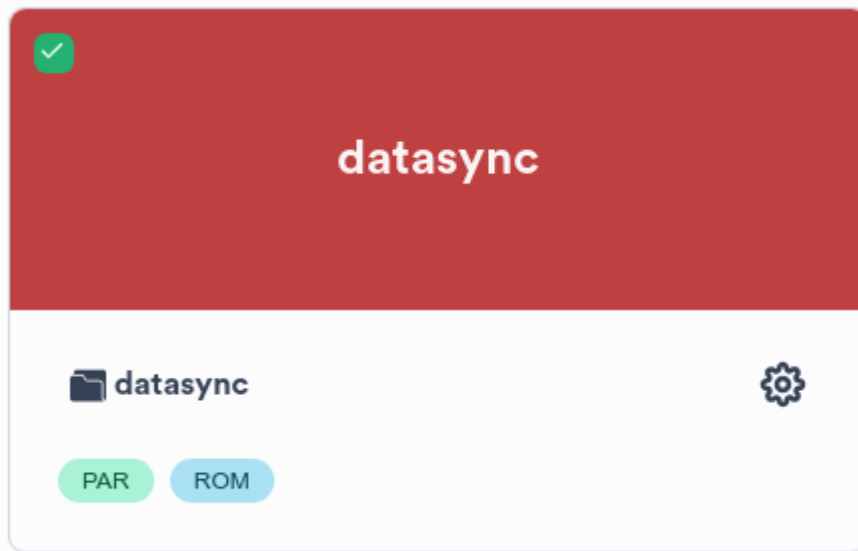
Hub can regularly perform the same action to a Space via Job Schedules.

Spaces which are deployed with Bi-Directional Sync automatically have a Schedule added.

Schedules can be added to the entirety of a Space. It is not possible to schedule more granular activities within a Space (E.G. regular Hydration of an individual file or directory).

Adding a Schedule

To add a Schedule to a Space, select the Space checkbox at the top left of the Space Card.



Selecting the checkbox shows the Job Creator Panel.

A screenshot of the 'Job Creator Panel'. It features a 'Workflow:' dropdown menu with 'Hydrate' selected. To the right are links for 'View selection list' and 'Clear selections'. Below the workflow menu, there's a section with '1 items' on the left. To its right are 'Source' and 'Queue' dropdown menus, with 'london' and 'default' selected respectively. A green 'Next >' button is on the far right.

For further information refer to [Using the Job Creator Panel](#).

After selecting any chosen requirements for the Job Creator Panel, the Configure workflow fields panel is raised.

If the job is to be immediately run, click the larger titled button related to the chosen Workdlow (E.G. Hydrate). Alternately the Job can be Scheduled to run later.

To Schedule the Job to run later, click the Schedule button.



Click the Schedule button to Schedule the Job for later

The Schedule button raises the Schedule Job dialog.

Schedule job



Recurring job name

myschedulename



Frequency

☒ Mins or Hours

☐ Daily or Weekly

Interval

1



mins



 Schedule "Hydrate"

- Enter the name of the Schedule
- Select the desired Schedule settings
- Click the Schedule button to create the Schedule

Editing Schedules

Schedule settings can be modified as required.



Clicking the Settings button on the Space card raises the Settings dialog for the Space.

Important: Adding or Configuring a Space can only be performed by a Hub Administrator or User with Space Administration rights granted through group management.

Click the **Job Schedules** menu item on the Space settings dialog.

Basic Space options

Job schedules**Job schedules**

Shares

Space backup

Q Type to filter



Name

Controls



schedule-datasync



Edit



Delete

Items per page:

20 ▾

1

Previous

Next

The Schedules for the Space are viewable.



Edit

Click the Edit button for the Schedule to be modified

Clicking the Edit button raises the Update job schedule dialog.

Update job schedule



Recurring job name

schedule-datasync



Frequency

☒ Mins or Hours

☐ Daily or Weekly

Interval

2



mins



Queue

site1



highpriority



Save

- Modify the desired Schedule settings
- Click the Save button to store the modified Schedule settings

Deleting Schedules

Schedules no longer required can be removed from Hub.



Clicking the Settings button on the Space card raises the Settings dialog for the Space.

Important: Adding or Configuring a Space can only be performed by a Hub Administrator or User with Space Administration rights granted through group management.

Click the **Job Schedules** menu item on the Space settings dialog.

Job schedules

Basic Space options

Job schedules

Shares

Space backup

Q Type to filter



Name

Controls



schedule-datasync



Edit



Delete

Items per page:

20 ▾

1

Previous

Next

The Schedules for the Space are viewable.



Delete

Click the Delete button for the Schedule to be deleted

Tip: Deleted Schedules are non-recoverable. If a Schedule has been inadvertently removed, do not press the Save button, instead click off the Space settings dialog to the main area of the screen.

Clicking the Save button at the bottom of the Space settings dialog saves any changes made.

Policies

Hub provides the capability to perform actions to data via Policies.

Policies comprise criteria, conditions and triggers which are applied to data within one or more Spaces residing on a Site.

Viewing Policies

To view available Policies, click the Policies menu button.



Navigates to the Policies screen

NGHub

Policies

hubadmin

Global

Type to filter

Create policy

Status	Policy name	Policy type	Action	Date created	Spaces	Controls
Enabled	migrate_images	Migration	Run	Thu, 06 Jun 2024, 22:42:01 GMT+1	2 Spaces	Edit Delete
Enabled	migrate-nearline	Migration	Run	Wed, 28 Feb 2024, 13:17:30 GMT+1	1 Spaces	Edit Delete
Enabled	migrate-nvme1	Migration	Run	Wed, 28 Feb 2024, 13:16:49 GMT+1	1 Spaces	Edit Delete
Enabled	project_to_cloud	Migration	Run	Thu, 29 Feb 2024, 16:03:27 GMT+1	1 Spaces	Edit Delete
Enabled	tier_down	Migration	Run	Wed, 28 Feb 2024, 13:18:17 GMT+1	1 Spaces	Edit Delete

Items per page: 20

1

Previous Next

Filtering the Policies View

To filter the list of Policies, type the Policy name or part of a Policy name in the filter bar.

Global

Q migrate

Create policy

Status	Policy name	Policy type	Action	Date created	Spaces	Controls
Enabled	migrate_images	Migration	Run	Thu, 06 Jun 2024, 22:42:01 GMT+1	2 Spaces	Edit Delete
Enabled	migrate-nearline	Migration	Run	Wed, 28 Feb 2024, 13:17:30 GMT+1	1 Spaces	Edit Delete
Enabled	migrate-nvme1	Migration	Run	Wed, 28 Feb 2024, 13:16:49 GMT+1	1 Spaces	Edit Delete

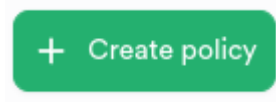
Items per page: 20

1

Previous Next

Adding a Policy

Click the Create Policy button to display a dialog to configure the Policy.



Important: This function can only be performed by a Hub Administrator

Creating a Policy Wizard

Navigating the Wizard



Click the close button to exit the wizard. Changes are not saved.

	Click the Next button to advance to the next page of the wizard. The Next button is disabled until all required page elements are completed.
	Click the Go Back button to return to the previous wizard page.
	Click the Finish & Create button to apply the changes displayed on the wizard summary page.

Name

Name

Name and type of policy

Policy name

Use Policies to

Tier data down to slower performance tiers or to promote data up to higher performance tiers.

Free space on a site's PixStor file system by migrating data to an Ngenea target.

Policies can be automatically enacted on a schedule or run as required manually.

Use one or more conditions to select distinct data to process.

- Provide a descriptive name for the Policy

Scope

The same Policy can run against one or more Spaces concurrently for a Site.

Policies are specific to a Site.

Scope

Spaces to include when applying the policy

Select site

London



Select spaces

☐	Name ^	Used space	Free space	Total space	
☐	 LocalSpace	0 B	0 B	0 B	
☐	 delivery	0 B	0 B	0 B	
☐	 mmfs1	0 B	0 B	0 B	
☒	 project1	0 B	0 B	0 B	
☒	 project2	0 B	0 B	0 B	
Items per page: 20		1	Previous	Next	

- Select the Space or Spaces to run the Policy against on the Site.

Automation

To setup a regular scheduled Policy run, enable the Automate Schedule slider.

Determine the required frequency of the Policy run.

Automate schedule

Frequency

☐ Mins or Hours

☒ Daily or Weekly

Every

☒ Mon
☒ Tue
☒ Wed
☒ Thu
☒ Fri
☒ Sat
☒ Sun

At

07

:

00

UTC+0

Determine the required frequency of the Policy run.

Choosing Mins or Hours will ensure that the Policy will run on the next interval set.

E.G: * 1 hour: The Policy will run on the next hour (12.00, 13.00) * 15 mins: The Policy will run on the next 15 minute interval past the hour (15, 30, 45, 00)

Schedule

Frequency

☐ Mins or Hours

☒ Daily or Weekly

Every

☒ Mon
☐ Tue
☒ Wed
☒ Thu
☐ Fri
☒ Sat
☒ Sun

At

12

:

05

UTC+0

Choosing Daily or Weekly allows the Policy to be scheduled once per chosen day at a specific time of day.

Hint: The schedule time is in UTC+0. You may need to account for any timezone offset of the site when scheduling.

Choose the number of Threads to assign to the policy run.

Performance

Choose the number of policy threads to use. More threads deliver faster processing, but take care to balance the total workload of all policies running on the site versus the available resources.

Threads Level

4

X

Hint: Best practice is to start small and increase after assessing the impact of the policy to the Site.

Pool triggers

Setup a pool trigger to define the flow of data.

Examples of supported data flows are:

Pool to Pool

Data in pool `nvme` is tiered to pool `nearline` according to the Policy conditions.

Pool trigger

Storage pool 1

nvme

▼

☐ Maximum utilisation

0

%

↓

Storage pool 2

nearline

▼

☐ Maximum utilisation

0

%

Add another pool to this hierarchy +

Pool with Utilisation Limit

Data in pool `nvme` is tiered to pool `nearline` according to the Policy conditions.

Assuming a full `nvme` pool, data over which matches Policy conditions and is filling the pool over the 40% maximum utilisation is targeted for tiering to pool `nearline`. Data will be tiered until `nearline` reaches 80% utilisation or the `nvme` reaches 40% utilisation. If neither can be achieved, a best fit results.

Storage pool 1

nvme

Maximum utilisation

40

%

↓

Storage pool 2

nearline

Maximum utilisation

80

%

Hint: Ngenea source/destinations are only supported with one additional pool. Specifying more than two pools, including Ngenea will result in no data orchestration to/from Ngenea Targets.

Pool to Ngenea

Data in pool `nvme` is dehydrated to an Ngenea Target according to the Policy conditions.

Pool trigger

Storage pool 1

nmve

Maximum utilisation

0

%

↓

Storage pool 2

Ngenea

When total pool storage usage exceeds

0

%

Migrate files until storage usage falls to

0

%

Pre-emptively pre-migrate an extra

0

%

Add another pool to this hierarchy +

Ngenea to Pool

Data in an Ngenea Target is rehydrated to pool `nvme` according to the Policy conditions.

Pool trigger

Storage pool 1

Ngenea

☐ When total pool storage usage exceeds

0

%

☐ Migrate files until storage usage falls to

0

%

☐ Pre-emptively pre-migrate an extra

0

%

Storage pool 2

nmve

☐ Maximum utilisation

0

%

Add another pool to this hierarchy +

Ngenea Watermarks

Utilise Ngenea watermarks to provide finite control over data states during Policy processing.

The following example provides (assuming 93% utilisation):

- Data will only be de-hydrated when the Storage Pool 1 (E.G. `nmve` in the above example) is utilised over 90%
- 13% of data residing on the pool will be de-hydrated, reducing utilisation to 80%
- A further 10% of data residing on the pool will be Pre-Staged, ensuring that future de-hydrations process the first 10% of data faster as the data is already present in the Ngenea Target

Ngenea

☒ When total pool storage usage exceeds

90

X

%

☒ Migrate files until storage usage falls to

80

X

%

☒ Pre-emptively pre-migrate an extra

70

X

%

Conditions

Select the processing order of the data.

For more finite control of the data targeted, add a Condition group.

+ Add condition group

Click the Add condition group button to add a Condition group

- Specify the criteria for the condition group as necessary
- It is not required to specify all selections of the Condition group

Condition groups

Oldest images

Selection by date

Include

All files accessed at a time older than

365

days

Selection by filetype

Include

Type a filetype and hit Return key

*.png

*.jpg

Selection by size

Include

Files less than

20

Byte

Minimize

Perform optional actions:

Condition group

Click the Condition Group Edit button to name the Condition group

Click the delete icon to remove the Condition group

+ Add condition group

Click the Add condition group button to add further Condition groups

Summary

Upon completing the wizard steps a summary is presented:

Summary

Review the policy definition below.

Policy name

migrate_images

← Go back and edit

Policy type

↩ Migrate

← Go back and edit

Scope

project1

project2

← Go back and edit

Threads level

4

← Go back and edit

Migration hierarchy

nmve

→

Ngenea

← Go back and edit

Processing order & conditions

Biggest files first - 1 condition

← Go back and edit

Finish & Create >

Click the Finish & Create button to apply the changes displayed on the wizard summary page.

Alternatively **Go back** and change the proposed configuration as required or **close** the wizard to cancel the creation of the Policy.

Editing a Policy

Important: This function can only be performed by a Hub Administrator.

Clicking the global settings button in the main menu bar displays the list of Ngenea targets:

Policies

hubadmin

Global

Type to filter

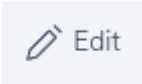
Create policy

Status	Policy name	Policy type	Action	Date created	Spaces	Controls
Enabled	migrate_images	Migration	Run	Thu, 06 Jun 2024, 22:42:01 GMT+1	2 Spaces	Edit Delete
Enabled	migrate-nearline	Migration	Run	Wed, 28 Feb 2024, 13:17:30 GMT+1	1 Spaces	Edit Delete
Enabled	migrate-mvme1	Migration	Run	Wed, 28 Feb 2024, 13:16:49 GMT+1	1 Spaces	Edit Delete
Enabled	project_to_cloud	Migration	Run	Thu, 29 Feb 2024, 16:03:27 GMT+1	1 Spaces	Edit Delete
Enabled	tier_down	Migration	Run	Wed, 28 Feb 2024, 13:18:17 GMT+1	1 Spaces	Edit Delete

Items per page: 20

1

Previous Next



Click the edit icon on the required Policy row to edit the Policy

- Modify the Policy settings as required. Refer to [Adding a Policy](#) for settings guidance.

Name

Configure the policy scope

Automation

Pool triggers

Conditions

migrate_images

Enabled

Name

Policy name

migrate_images

×

Use Policies to

Tier data down to slower performance tiers or to promote data up to higher performance tiers.
Free space on a site's PixStor file system by migrating data to an Ngenea target.
Policies can be automatically enacted on a schedule or run as required manually.
Use one or more conditions to select distinct data to process.

Configure the policy scope

Select site

London

▼

Select spaces

	Name ^	Used space	Free space	Total space	
	LocalSpace	0 B	0 B	0 B	
	delivery	0 B	0 B	0 B	
	mmfs1	0 B	0 B	0 B	

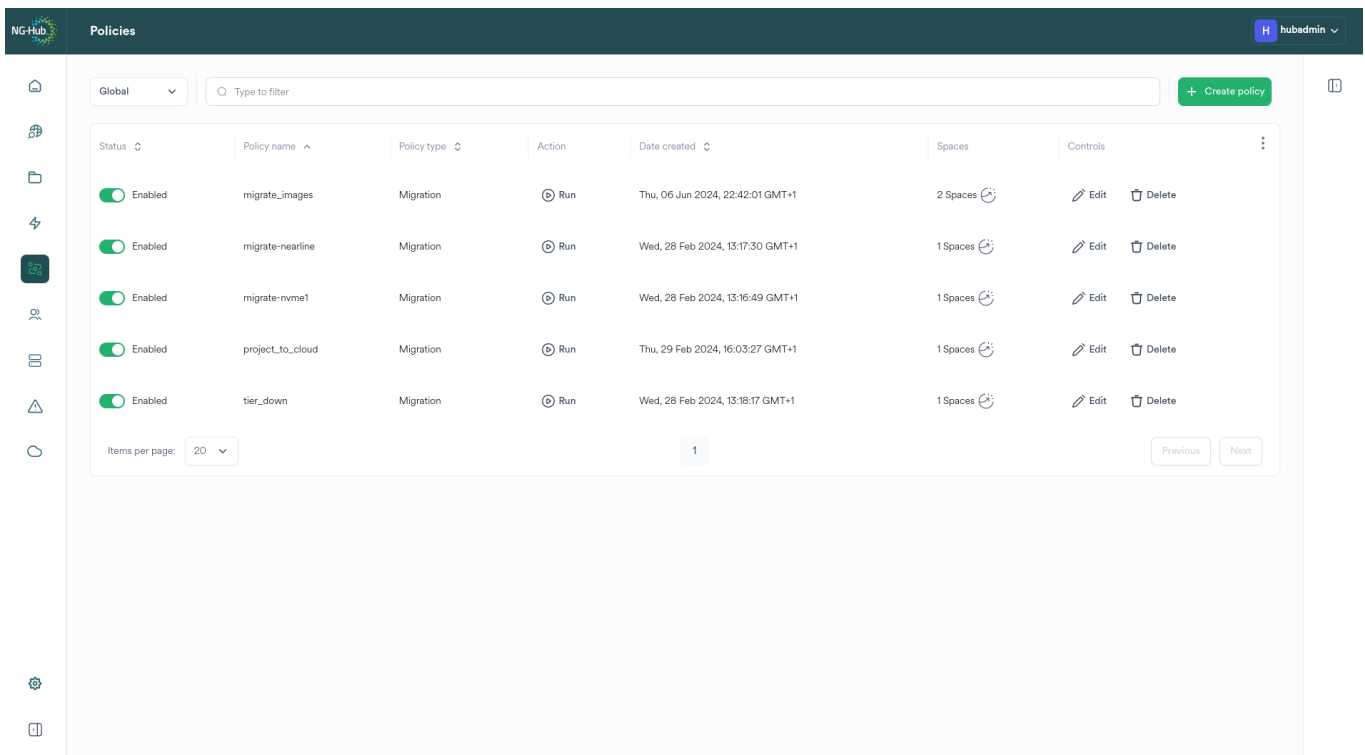
Save

Deleting a Policy

Important: This function can only be performed by a Hub Administrator.

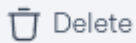
Clicking the Policies button in the main menu bar displays the list of Policies:

79



The screenshot shows the 'Policies' page in the NGHub interface. At the top, there's a 'Global' filter dropdown and a search bar with the placeholder 'Type to filter'. A green '+ Create policy' button is in the top right. The table below lists five policies, all with a status of 'Enabled'. Each row includes a toggle switch, the policy name, type, action (a 'Run' button), date created, number of spaces, and control icons (edit and delete). The policies are: 'migrate_images' (2 Spaces), 'migrate-nearline' (1 Space), 'migrate-nvme1' (1 Space), 'project_to_cloud' (1 Space), and 'tier_down' (1 Space). At the bottom of the table, there's a pagination section showing 'Items per page: 20' and a single page indicator '1'.

Status	Policy name	Policy type	Action	Date created	Spaces	Controls
Enabled	migrate_images	Migration	Run	Thu, 06 Jun 2024, 22:42:01 GMT+1	2 Spaces	Edit Delete
Enabled	migrate-nearline	Migration	Run	Wed, 28 Feb 2024, 13:17:30 GMT+1	1 Spaces	Edit Delete
Enabled	migrate-nvme1	Migration	Run	Wed, 28 Feb 2024, 13:16:49 GMT+1	1 Spaces	Edit Delete
Enabled	project_to_cloud	Migration	Run	Thu, 29 Feb 2024, 16:03:27 GMT+1	1 Spaces	Edit Delete
Enabled	tier_down	Migration	Run	Wed, 28 Feb 2024, 13:18:17 GMT+1	1 Spaces	Edit Delete



Delete

Click the delete icon on the required Policy row to delete the Policy

A confirmation dialog is raised:

Delete policy



Do you wish to delete policy 'safsf'?

Yes

No

- Click Yes to delete the Policy. This action is irreversible.
- Alternately click no, or close the confirmation dialog.

Running a Policy

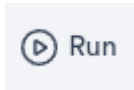
Policies are run either on a Schedule, as specified in the Wizard on creation or in the Policy settings.

Alternatively, policies can be run on demand by clicking the Run button in the relevant row of the Action column on the Policies page.

Important:

- Policies cannot be cancelled from the Hub UI once started
- Progress feedback is not visible in the UI

- Ensure the Policy is appropriate prior to running or scheduling



Click the Run button to manually start the Policy run

A confirmation dialog is raised:

A confirmation dialog titled 'Run policy' with a close button (X) in the top right corner. The question is 'Do you wish to run policy 'migrate-nearline'?'. There are two buttons: a green 'Yes' button and a white 'No' button.

Run policy

Do you wish to run policy 'migrate-nearline'?

Yes

No

- Click Yes to run the Policy.
- Alternately click no, or close the confirmation dialog.

Running Policies are shown on the Jobs page, subject to the above constraints. I.E. no controls are available for Policies on the Jobs page.

Policies provide specific Policy information when the Job is selected in the Jobs Page.

	
Policy info	
Policy name	migrate_images
Policy type	Migration
Site	London
Status	Enabled
Date created	06.06.2024, 22:42 GMT+1
Threads	4
Spaces	2
Conditions	1
Triggers	2
Filesystem	mmfs1
Order	Biggest files first

Refer to doc:jobs for further Jobs guidance.

Cancelling a Policy

Presently there is no capability to cancel a running Policy via the Hub UI.

Running policies can be terminated by issuing a `kill -9` to the `mmapplypolicy` process running on the relevant Ngenea node on the Site.

Upon terminating the running policy via the CLI, the Hub job will also terminate. The resultant Job state is (correctly) Failure - as the running policy was forcibly terminated.

Groups & Users

Hub provides management of Groups and Users internally.

Users may be authenticated by external directory services such as LDAP or Active Directory.

Two default groups are provided:

- Members of Administrators can configure and manage all Sites and Spaces
- Members of Users have read-only access to all Spaces

Additional groups can be deployed to provide restricted access for users to Sites and Spaces.

Viewing Groups

To view the available Groups managed in Hub, select the Group tab.

Important: Group management can only be performed by a Hub Administrator.

The screenshot shows the 'Groups & Users' management interface in NG-Hub. The top navigation bar includes the NG-Hub logo, the title 'Groups & Users', and a user profile for 'hubadmin'. Below the navigation bar, there are tabs for 'Groups' and 'Users'. A search bar with the placeholder 'Type to filter by group name' and a '+ Create group' button are present. The main content area displays a list of groups, each with an 'Edit' button and a 'Delete' button. The groups are categorized into four sections: Administrators, Local SSH Admins, OB Software, and ShotMgmt. Each group entry includes a username, a date joined, a last active time, and a controls menu.

Group Name	Username	Date joined	Last active	Controls
Administrators	hubadmin [Hub Admin] hubadmin@mycompany.com	Tue, 23 Jan 2024, 10:40:58 GMT	Thu, 29 Feb 2024, 22:33:26 GMT	...
Local SSH Admins	localsshadmin [Local SSH Admin] sshadmin@mycompany.com	Wed, 24 Jan 2024, 14:23:18 GMT	-	Delete
OB Software	obsoftware [OB Software] obsoftware@mycompany.com	Wed, 24 Jan 2024, 11:28:21 GMT	-	Delete
ShotMgmt	localuser [Local User] shotmgmt@mycompany.com	Wed, 24 Jan 2024, 11:50:35 GMT	-	Delete

The Groups tab provides the following actions:

- Group creation
- Group deletion
- Group member management

Filtering the Groups View

To display Groups matching keywords, enter the keywords in the filter bar.

Q admin

×

+

Create group

Administrators

Default group for users with administrative access

✎

Edit

Username ^	Date joined ↕	Last active ↕	Controls
H hubadmin	Fri, 30 Aug 2024, 10:56:01 GMT+1	Mon, 02 Sep 2024, 13:35:09 GMT+1	
LS localsshadmin [Local SSH Admin] sshadmin@mycompany.com	Mon, 02 Sep 2024, 14:34:23 GMT+1	-	🗑 Delete
R root	Fri, 10 Feb 2023, 15:06:31 GMT+1	Thu, 29 Aug 2024, 15:20:52 GMT+1	🗑 Delete

LocalSSHAdmins

Local SSH Admins

✎

Edit

🗑

Delete

Username ^	Date joined ↕	Last active ↕	Controls
LS localsshadmin [Local SSH Admin] sshadmin@mycompany.com	Mon, 02 Sep 2024, 14:34:23 GMT+1	-	🗑 Delete

Group Members

The Groups View will only display up to four members per group.

View all members

Click the View all members button to view all the members of the group

Clicking the View all members button raises the Group Membership dialog.

vfx_grading members

LS

localsshadmin [Local SSH Admin]
sshadmin@mycompany.com

X Remove

LU

localuser [Local User]
shotmgmt@mycompany.com

X Remove

OS

obsoftware [OB Software]
obsoftware@mycompany.com

X Remove

RO

readonly [Read Only]
myuser@mycompany.com

X Remove

R2

resolve2 [resolve 2]
resolve2@mycompany.com

X Remove

Save

To filter for a User within the list of group members enter a keyword in the Filter for...

Q Type to filter by username

✕ Remove

Click the Remove button to remove a User from the Group

Tip: If a User has been inadvertently removed, do not press the Save button, instead click off the Group members dialog to the main area of the screen.

Clicking the Save button at the bottom of the Group members dialog saves any changes made.

Creating Groups

Click the Create Group button to display a dialog to configure a new group.

+ Create group

Important: This function can only be performed by a Hub Administrator

Basic Group Options

Basic settings

Group name

Type the name of the group

Description

Type a description for the group

Users

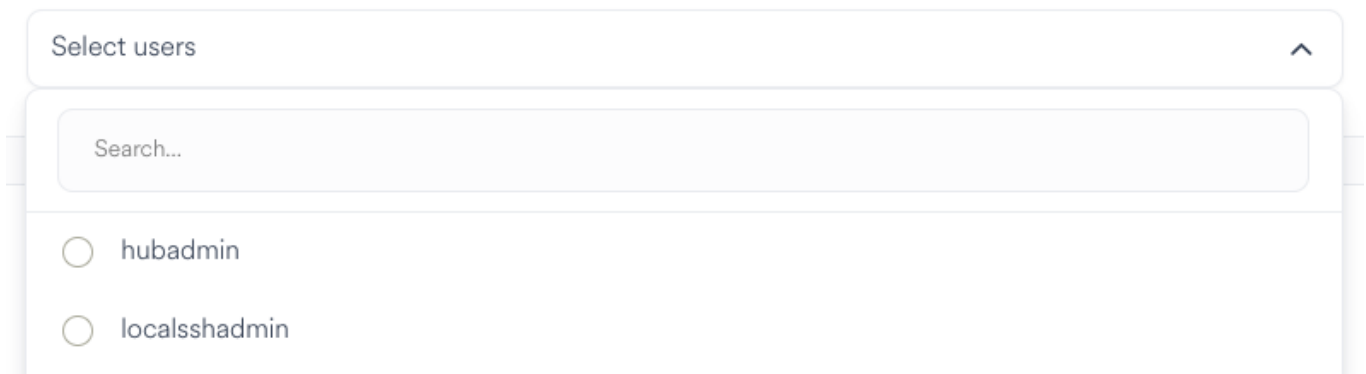
Select users



- Enter a name for the group. The name is case-sensitive and may not contain whitespace
- Enter a human-readable description for the group.

Select zero or more users to assign to the group

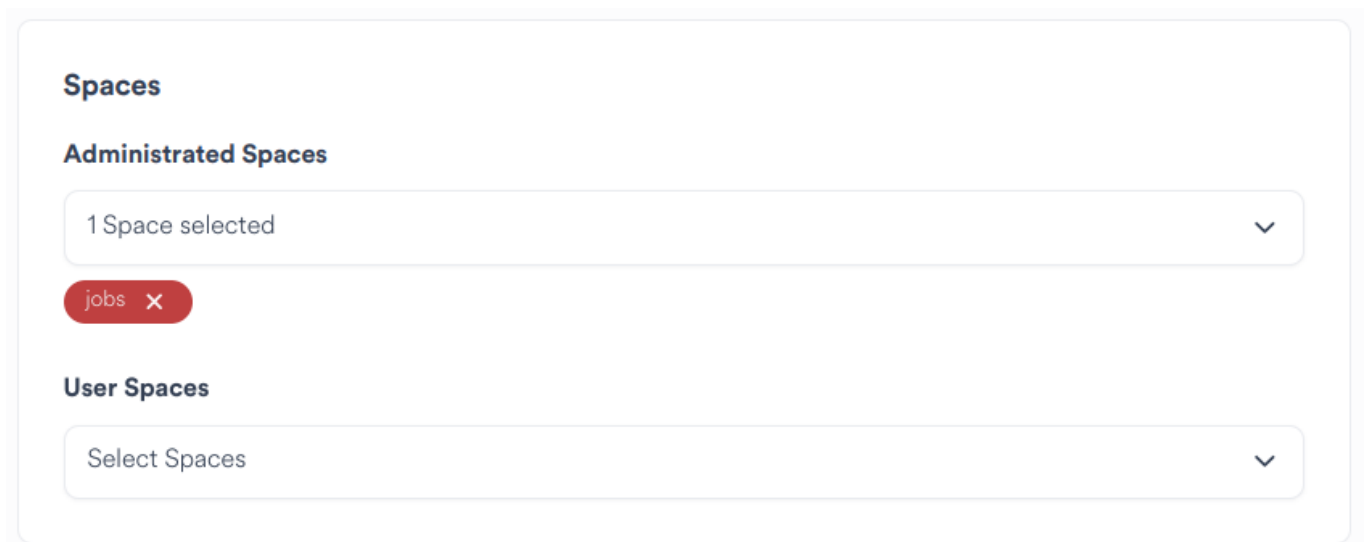
Users



- Click a username to select it
- Click the username again to deselect it

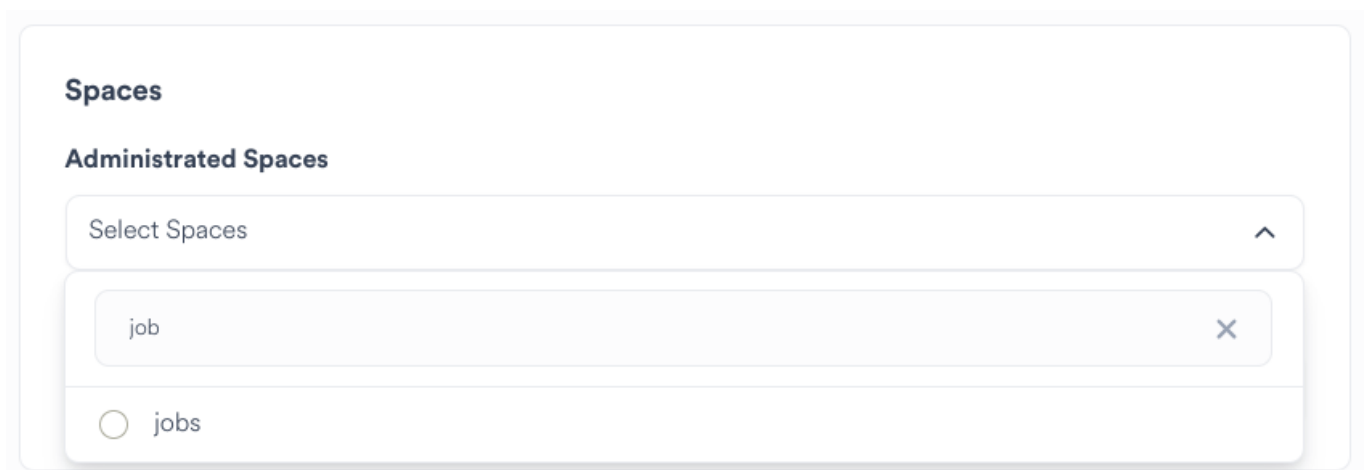
Group Space Options

Select which Spaces can be viewed and administered by users in this group.



- Administered Spaces can be viewed and browsed, and can have their settings changed
- User Spaces can be viewed and browsed, but cannot have their settings changed

To filter for a Space within the list of spaces enter a keyword to filter for



Selected spaces are shown as chips. Clicking the 'X' on a chip deselects the space.

Management Settings

Select whether users in the group have the right to administer Ngenea Hub

Management

Administer Sites/Hub

Whether this group has rights to administer sites and the Hub



This includes the right to:

- Configure site settings
- Manage and run policies
- Manage Groups and Users
- View Alerts
- Manage, browse, and import from Targets
- Manage Global Settings

NAS Group Settings

Select whether the group should be created as a local NAS group.

NAS Group Settings

Also create this group on all pixstors



This will create a unix group on PixStor for all sites.

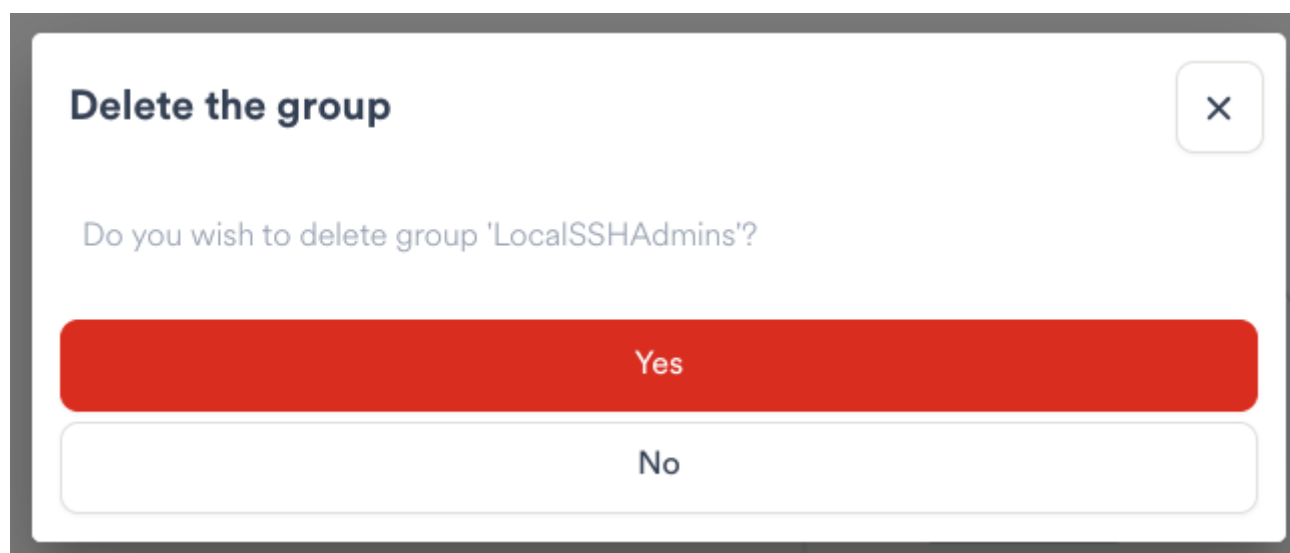
Deleting Groups

LocalSSHAdmins

Local SSH Admins



A confirmation dialog will be displayed



- Click "No" to close the confirmation dialog. The group will not be deleted.
- Click "Yes" to confirm and remove the group.

The default groups `Administrators` and `Users` cannot be deleted.

Viewing Users

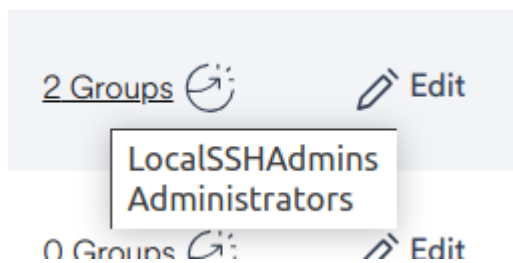
To view the available Users managed in Hub, select the Users tab.

Important: User management can only be performed by a Hub Administrator.

Username	Email	Date joined	Groups	Controls
HA hubadmin [Hub Admin]	hubadmin@mycompany.com	Tue, 23 Jan 2024, 10:40:58 GMT	1 Groups	Edit
LS localsshadmin [Local SSH Admin]	sshadmin@mycompany.com	Wed, 24 Jan 2024, 14:23:18 GMT	3 Groups	Edit Delete
LU localuser [Local User]	shotmgmt@mycompany.com	Wed, 24 Jan 2024, 11:50:35 GMT	2 Groups	Edit Delete
LU localuser2 [Local User]	swrev@mycompany.com	Wed, 24 Jan 2024, 11:57:42 GMT	2 Groups	Edit Delete
OS obsoftware [OB Software]	obsoftware@mycompany.com	Wed, 24 Jan 2024, 11:28:21 GMT	2 Groups	Edit Delete
RO readonly [Read Only]	myuser@mycompany.com	Thu, 08 Feb 2024, 11:52:36 GMT	2 Groups	Edit Delete
R2 resolve2 [resolve 2]	resolve2@mycompany.com	Tue, 23 Jan 2024, 16:43:39 GMT	1 Groups	Edit Delete
R root		Tue, 23 Jan 2024, 17:35:42 GMT	0 Groups	Edit Delete

The Users tab provides the following actions:

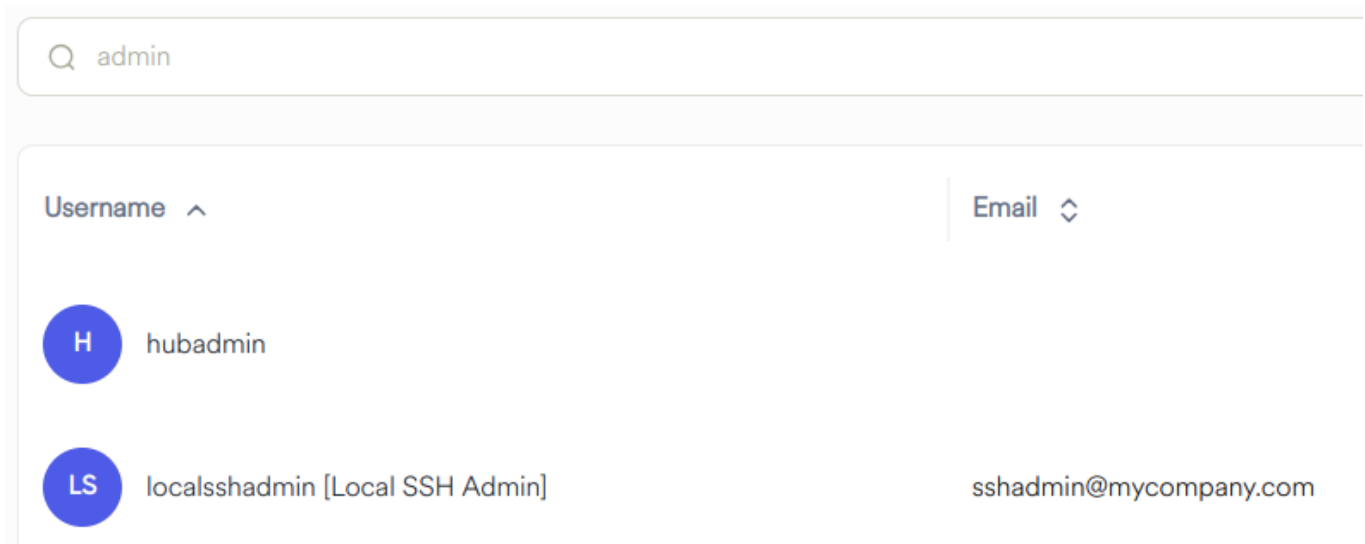
- User creation
- User deletion
- Group member management



Hover over the Groups column to see what groups a user belongs to

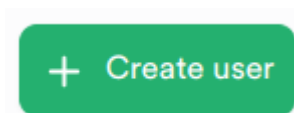
Filtering the Users View

To display Users matching keywords, enter the keywords in the filter bar.



Creating Users

Click the Create User button to display a dialog to configure a new user.



Important: This function can only be performed by a Hub Administrator

Basic User Options

User details

Configure the basic user information

Basic information

Username

Password

First name

Last name

Email

Groups



- Enter a name for the user. The name is case-sensitive and may not contain whitespace
- Enter a password
- Enter First and Last name for the user
- Enter an email address for the user

The password must meet all the requirements which are listed below the password box

Password



Password must be at least 8 characters long.
Password must contain at least one digit.
Password must contain at least one special character: ? - + # etc..
Password must contain at least one uppercase letter.

Select zero or more groups to assign the user to

Groups

Search...

- ☒ Administrators
- ☐ Hub2 Group With Spaces
- ☐ mynasgroup
- ☐ NGHUB-DEV-Administrators
- ☐ space-admin
- ☐ Users

- Click a group name to select it
- Click the group name again to deselect it

NAS User Settings

Select whether the user should be created as a local NAS user.

NAS User Settings

Also create this user on all pixstors



Grant administrative command line access to all sites



Primary Group

Select a primary group



This will create a unix user on PixStor for all sites.

Select whether to grant command line access to the user. This means the user can connect to PixStor nodes via SSH.

Select the Primary Group for the user.

Primary Group

Select a primary group



<Create new user group>

LocalSSHAdmins

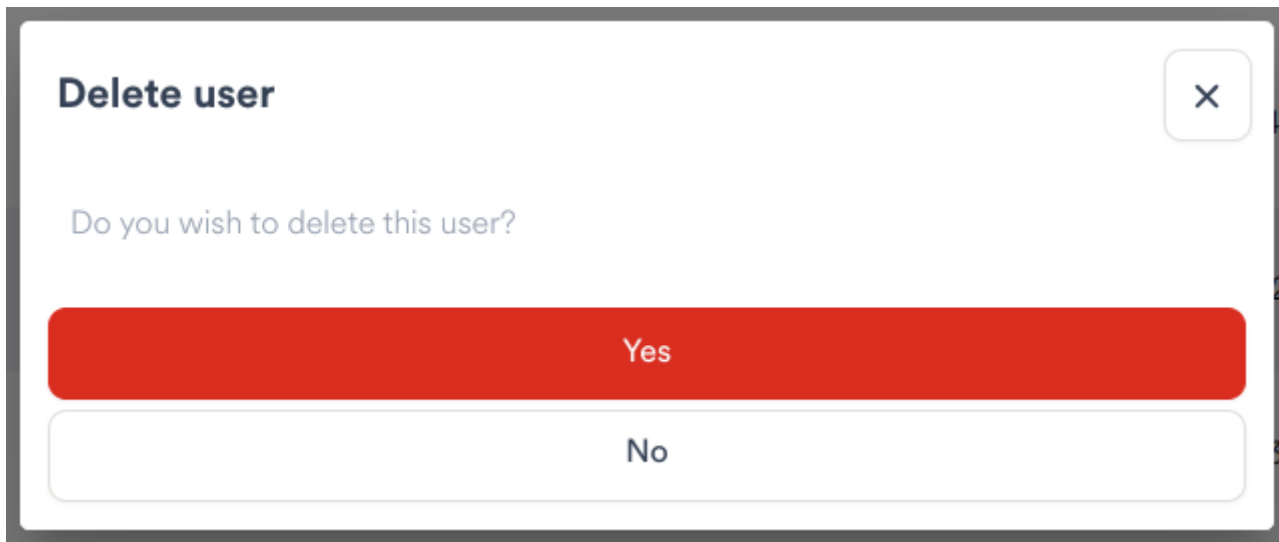
The dropdown will list any existing NAS groups. Alternatively, "Create a new user group" will create a new NAS-enabled Hub group with the same name as this user.

Deleting Users

To delete a user, click the delete "bin" icon in the row of that user.



A confirmation dialog will be displayed



- Click "No" to close the confirmation dialog. The user will not be deleted.
- Click "Yes" to confirm and remove the user.

You cannot delete the user you are currently logged in with.

Restricting Users from Spaces

All users in the group Users have read-only access to all Spaces.

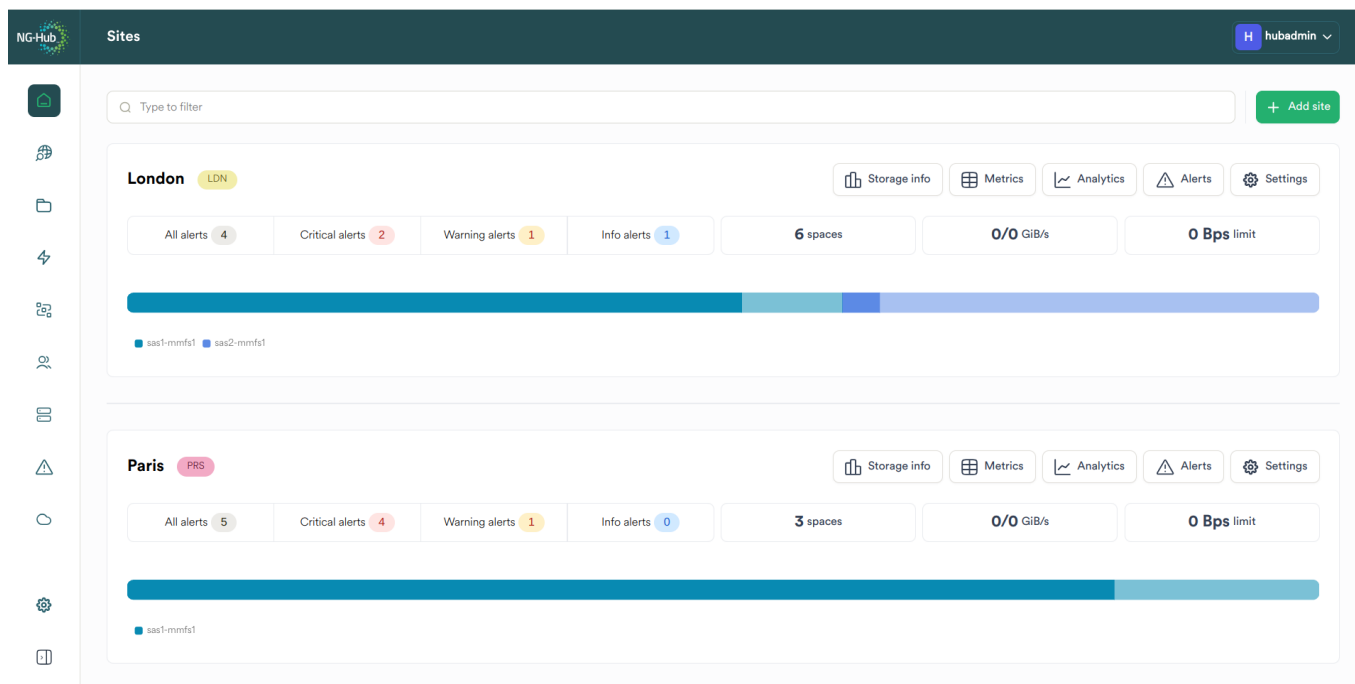
Should it be required to restrict a user from accessing specific spaces Spaces this can be achieved by:

1. Creating a new user group
2. Ensuring that only the specific spaces are assigned to the group
3. Add specific users to the group
4. Ensure the specific users are removed from the group Users

Caution: Adding a Space to a group's Administered Spaces and Used spaces will allow assigned Users to change the settings for a Space. If administrative operations are not required, do not assign Spaces to Administered Spaces - create an additional group to allow specific users to administrate specific spaces.

Sites

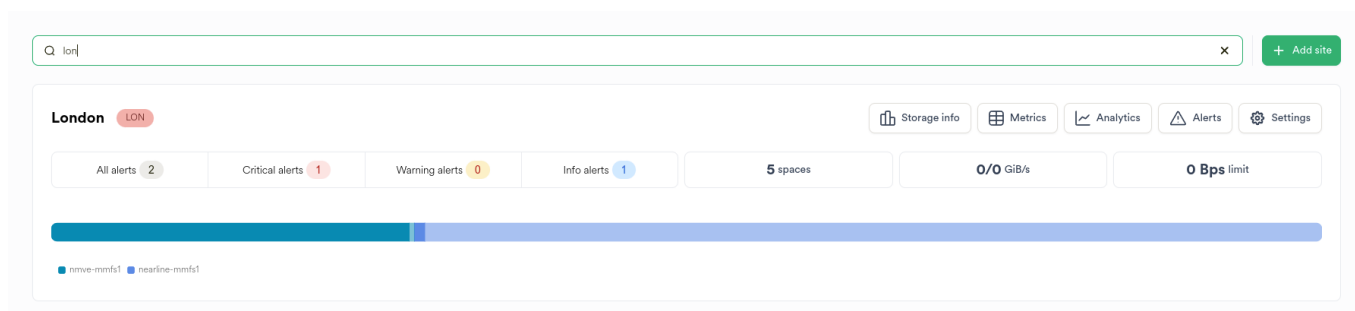
A site is a physical or cloud based pixstor server managed by Hub.



Filtering Sites

To filter for a site within the list of sites enter a keyword in the Search for...

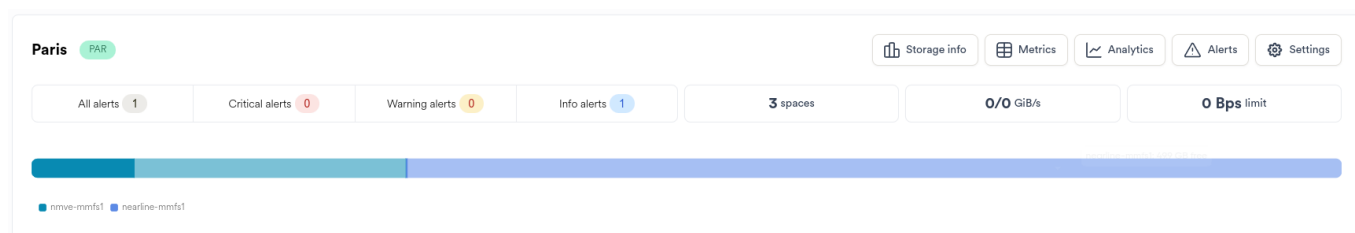
The displayed sites is limited to the sites which match the keyword(s).



The Site Card

Each site is displayed as a site card.

A Site is displayed as a card in the Sites view.



A Site Card comprises:

Site Name

London

LON

Displays the designated friendly name of the site with the Ngenea site chip

Site Summary

The site summary displays high level totals for the number of files and folders present, the Ngenea hydrated and dehydrated states and the number of Spaces the site hosts.

All alerts 2

Critical alerts 1

Warning alerts 0

Info alerts 1

5 spaces

0/0 GiB/s

0 Bps limit

Alert Types

Numbered filter buttons display the count of each type of Alert for the Site.

Select a button to open the Alerts for the Site, filtered for the specific Alert type.

All alerts 6

Click the All alerts button to view all alerts

Critical alerts 3

Click the Critical alerts button to filter for all Critical alerts

Warning alerts 0

Click the Warning alerts button to filter for all Warning alerts

Info alerts 3

Click the Info alerts button to filter for all Info alerts

Storage Info



Storage info

Click the Storage info button to open the Storage Info browser for the Site.

Site Metrics



Metrics

Click the Metrics button to open the pixstor nexus site metrics in a new browser tab.

Site Analytics



Analytics

Click the Analytics button to open the pixstor nexus site analytics in a new browser tab.

Site Alerts



Alerts

Click the Alerts button to open the Alerts for the Site.

Bandwidth Control

If bandwidth control has been enabled by a Hub Administrator with CLI access, the bandwidth of a site can be limited to a defined value. The current value is observed on the bandwidth limit button.

0 Bps limit

Click the bandwidth limit button to display the bandwidth control dialog.

The bandwidth control dialog allows limiting the bandwidth of a site to a defined value. Enter the limit in Megabits per second (Mbps) and press Save to apply the limit.

Limit traffic for Paris



Bandwidth in Mbps

10000



Save

Hint: If the bandwidth for a site has been inadvertently set do not press the Save button, instead click off the Bandwidth limit dialog to the main area of the screen.

Important: This function can only be performed by a Hub Administrator.

Settings

Important: This function can only be performed by a Hub Administrator.



Settings

Click the site's Settings button to display a dialog to configure the selected site.

Name

London

Site: london

Network settings

Name

Backup

Email SMTP settings

Identity management

Site name & shortcode

London ✕

LON ✕

Type of site



On premise



Cloud

Public URL

https://10.60.0.167 ✕

Site colour



Choose a colour to identify the site

Network settings

NAS Interfaces



Group



Range 1



IP range start

IP range end

CIDR

Gateway

10.70.0.167 ✕

10.70.0.170 ✕

24 ✕

10.70.0.1 ✕

Save

- Modify the Site settings as required. Refer to [Adding a Site](#) for settings guidance.

Pool Space

One or more pixstor storage pools which comprise the pixstor file system are represented.

Hovering over the pool percentage bar provides the remaining capacity for the pool.



■ nmve-mmfs1 ■ nearline-mmfs1

Adding a Site

Add Site Wizard

Hub allows remote configuration of all participating pixstor sites.

New sites are automatically joined to Hub awaiting optional configuration via the Site Wizard.

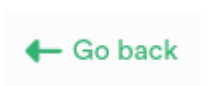
Navigating the Wizard



Click the close button to exit the wizard. Changes are not saved.



Click the Next button to advance to the next page of the wizard. The Next button is disabled until all required page elements are completed.



Click the Go Back button to return to the previous wizard page.



Click the Finish & Create button to apply the changes displayed on the wizard summary page.

Add Site



Click the Add site button to display a dialog to configure the selected site.

Important: This function can only be performed by a Hub Administrator.

Pick site to configure

Sites which have been automatically registered to Hub but not yet configured are shown:

Pick site to configure

Site ID: london

Configure this new site >

Site ID: new_york

Configure this new site >

Site ID: paris

Configure this new site >

Site ID: singapore

Configure this new site >

- Select a site to configure

Configure this new site >

Click the Configure this new site button to proceed

Site Name

Name

Site name & shortcode

London

×

LON

×

Type of site

☒ On premise ☐ Cloud

Public URL

https://10.60.0.167

×

Site colour



Choose a colour to identify the site

- Provide a friendly name for the Site
- Provide a 3 character short code for the site. The shortcode is displayed as the label on the Site's chip. E.G. LDN for London

- Specify whether the site is on-premise or a pixstor cloud deployment. Each site type provides different Network Setting options.
- Provide the URL IP or FQDN which refers to the pixstor management node of the Site
- Select a colour for the Site chip

Network Settings (On premise)

- Add the required IP address or network range and specify a valid CIDR mask to apply the restriction
- Specify a gateway, if required
- Select the server(s) and interface(s) of the server where the IP range will be configured

+ Add IP range

Click the Add IP range button to add additional restrictions

+ Create another group

Click Create another group to add additional IP range to interface mapping groups



Click the delete button to remove an IP range or Interface group

Network Settings (Cloud & General)

pixstor cloud systems use predefined network architectures.

Unlike on-premise pixstor systems there is no requirement to create IP ranges or interface groups. IP addressing is externally managed by the cloud / virtual environment.

Both on-premise and cloud systems share common network configuration for DNS, Timezone and NTP.

The screenshot displays a configuration interface with four distinct sections, each enclosed in a light gray rounded rectangle. The first section, titled 'DNS Servers', contains a single button labeled '+ Add DNS server'. The second section, titled 'DNS search domains', contains a single button labeled '+ Add DNS search domains'. The third section, titled 'Timezone', features a dropdown menu currently set to 'Europe/London' with a downward arrow icon on the right. The fourth section, titled 'NTP Servers', includes a text input field with the placeholder text 'Type in NTP' and a trash can icon to its right, followed by a button labeled '+ Add NTP server'.

- Specify the IP address or FQDN hostname of one or more DNS servers
- Specify one or more DNS search domains
- Specify the Timezone in which the server resides, or will participate in
- Specify the IP address or FQDN hostname of one or more DNS servers

Hint: If the pixstor site will be joined to an external Identify Mapping service such as Active Directory or LDAP, best practice is to ensure that the DNS and NTP servers match those of the service, or point at the service hosted DNS and NTP if it provides such capabilities. Should the pixstor become out of time sync with the Identify Mapping service login failures can occur.

Backup

pixstor provides the capability to backup data within a Space on a per-Site basis to specific Ngenea Targets.

Hub enables configuration to be set for the Ngenea Backup service running on pixstor sites.

If the Site is enabled to participate in backups, each Space requires additional configuration to enable the per-Space backup.

For more information refer to [Backups](#).

Backup

Enable backup

Do you want this site to participate in backups?

Frequency

☐ Mins or Hours

☒ Daily or Weekly

Every

✓

Mon

✓

Tue

✓

Wed

✓

Thu

✓

Fri

✓

Sat

✓

Sun

At

08

▼

:

00

▼

UTC+0

Determine the required frequency of the backup.

Schedule

Frequency

☒ Mins or Hours

☐ Daily or Weekly

Interval

6

×

hours

▼

Choosing Mins or Hours will ensure that the schedule will run on the next interval set.

E.G: * 1 hour: The backup will run on the next hour (12.00, 13.00) * 15 mins: The backup will run on the next 15 minute interval past the hour (15, 30, 45, 00)

Schedule

Frequency

☐ Mins or Hours

☒ Daily or Weekly

Every

☒ Mon

☐ Tue

☒ Wed

☒ Thu

☐ Fri

☒ Sat

☒ Sun

At

12 ▾

:

05 ▾

UTC+0

Choosing Daily or Weekly allows the backup to be scheduled once per chosen day at a specific time of day.

Hint: The schedule time is in UTC+0. You may need to account for any timezone offset of the site when scheduling.

Optional tuning can be set for the backup operation in the Advanced Configuration.

Refer to the Ngenea Backup documentation prior to applying any parameters.

Advanced configuration

Leave fields blank to use defaults

Number of threads

Hashing threshold (MB)

Timeout (seconds)

Connection timeout (seconds)

Number of connection retries

Email SMTP Settings

pixstor provides the capability to notify an inbox if service issues arise.

Email SMTP settings

Configure the email SMTP settings

Server

Port

Username

Password

Alert recipients
Add the emails to send notifications about this site.

- Specify the SMTP configuration of an email server to which to send notification emails
- Specify one or more valid email addresses to receive the notification emails

Mode

Identity Management

- Specify the Identity Management mode as appropriate:

Mode	Description
Active Directory	pixstor uses RFC2307 compliant identity mapping with Active Directory
Standalone	pixstor generates local UIDs and GIDs mapped to Active Directory SIDs

Domain details

Domain name

MYDOMAIN

Machine account name

MYPIXSTOR

ID mapping range

2000

200000

Domain Servers

Use DNS to determine AD servers

- Specify the Domain to join
- Specify the Machine account name
- Specify the ID range to map to
- Specify whether to use DNS to locate an Active Directory Domain Controller or alternatively specify an IP address or hostname

Domain join username

Administrator

Domain join password

.....

- Specify a valid username and password with domain join capability

Summary

Upon completing the wizard steps a summary is presented:

Summary

Here are the summary of your selections

You have named the site as **london**

↩ Go back and edit

You added **1 NAS group**

↩ Go back and edit

You have added **1 DNS server**

↩ Go back and edit

You have added **1 NTP server**

↩ Go back and edit

You want to send notifications to **2 emails**

↩ Go back and edit

You configured **1 domain**

↩ Go back and edit

Finish & Create >

Click the Finish & Create button to apply the changes displayed on the wizard summary page.

Alternatively **Go back** and change the proposed configuration as required or **close** the wizard to cancel the creation of the Ngenea target.

Alerts

Hub provides a view of alerts across all Sites.

- Only active alerts are displayed
- Muted alerts are not displayed



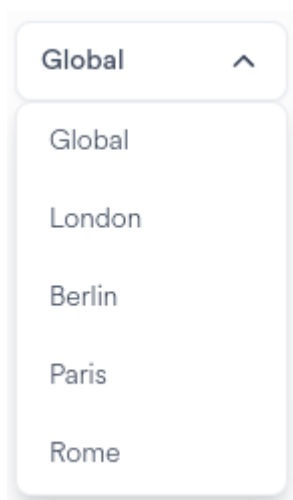
Click the alerts button to navigate to the Global Alerts screen

Alerting since	Alert name	Site	Node	Severity	Summary
Fri, 01 Mar 2024, 10:40:37 GMT	PixStorPoolData	London	hw-dev-pixit-01	Critical	PixStor pool data capacity alert
Thu, 22 Feb 2024, 10:52:21 GMT	ServiceFailed	London	hw-dev-pixit-01	Info	NetworkManager-wait-online.service has failed to ...
Tue, 13 Feb 2024, 16:39:31 GMT	ServiceFailed	Paris	hw-dev-pixit-02	Info	NetworkManager-wait-online.service has failed to ...
Tue, 13 Feb 2024, 16:39:21 GMT	ServiceFailed	Rome	hw-dev-pixit-03	Info	NetworkManager-wait-online.service has failed to ...
Tue, 30 Jan 2024, 09:48:08 GMT	SiteHealth	ob1	-	Critical	All workers are offline
Tue, 30 Jan 2024, 09:48:08 GMT	WorkerHealth	ob1	devpx6-jucker-21ga-demo-77-1	Critical	Worker on devpx6-jucker-21ga-demo-77-1 is offline

Hub provides two views of Alerts - Global and Local.

- Global displays all Alerts on all Sites
- Local displays the Alerts on a specific Site

The default view of Alerts is Global.



To switch to a site-centric view, select the specific site from the Alerts drop-down menu.

To switch to Global view, select Global from the Alerts drop-down menu.

London

Global

London

New York

Paris

Singapore

Choosing a specific Site displays Alerts only from the chosen Site:

London

Type to filter on alert name

Alerting since	Alert name	Site	Node	Severity	Summary
Fri, 01 Mar 2024, 10:40:37 GMT	PixStorPoolData	London	hw-dev-pixit-01	Critical	PixStor pool data capacity alert
Thu, 22 Feb 2024, 10:52:21 GMT	ServiceFailed	London	hw-dev-pixit-01	Info	NetworkManager-wait-online.service has failed to ...

Items per page: 20

1

PreviousNext

Alert Types

Above the Alert table, numbered filter buttons display the count of each type of Alert.

Select a button to filter for the specific Alert type.

All alerts 6	Click the All alerts button to view all alerts
Critical alerts 3	Click the Critical alerts button to filter for all Critical alerts
Warning alerts 0	Click the Warning alerts button to filter for all Warning alerts
Info alerts 3	Click the Info alerts button to filter for all Info alerts

Categories

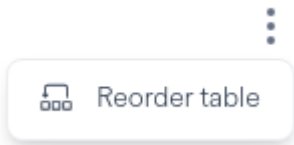
To filter for an Alert category, enter text into the filter bar.

Alerts are filtered where the Alert name matches the text in whole or part.

Global	Q health					X
Alerting since	Alert name	Site	Node	Severity	Summary	
Tue, 30 Jan 2024, 09:48:08 GMT	SiteHealth	ob1	-	Critical	All workers are offline	
Tue, 30 Jan 2024, 09:48:08 GMT	WorkerHealth	ob1	devpx6-jtucker-2lga-demo-77-1	Critical	Worker on devpx6-jtucker-2lga-demo-77-1 is offline	
Items per page: 20					1	Previous Next

Reordering

The alerts table columns can be reorganised for user preference.



Click the vertical dots on the right hand side of the table and select the Reorder table button to open the table organizer

Re-arrange items in top to bottom order by dragging vertically up or down.

Side bar organizer

X

Job info

Job details

Task stats

Job task statistics

File stats

Job file statistics

Save

Targets

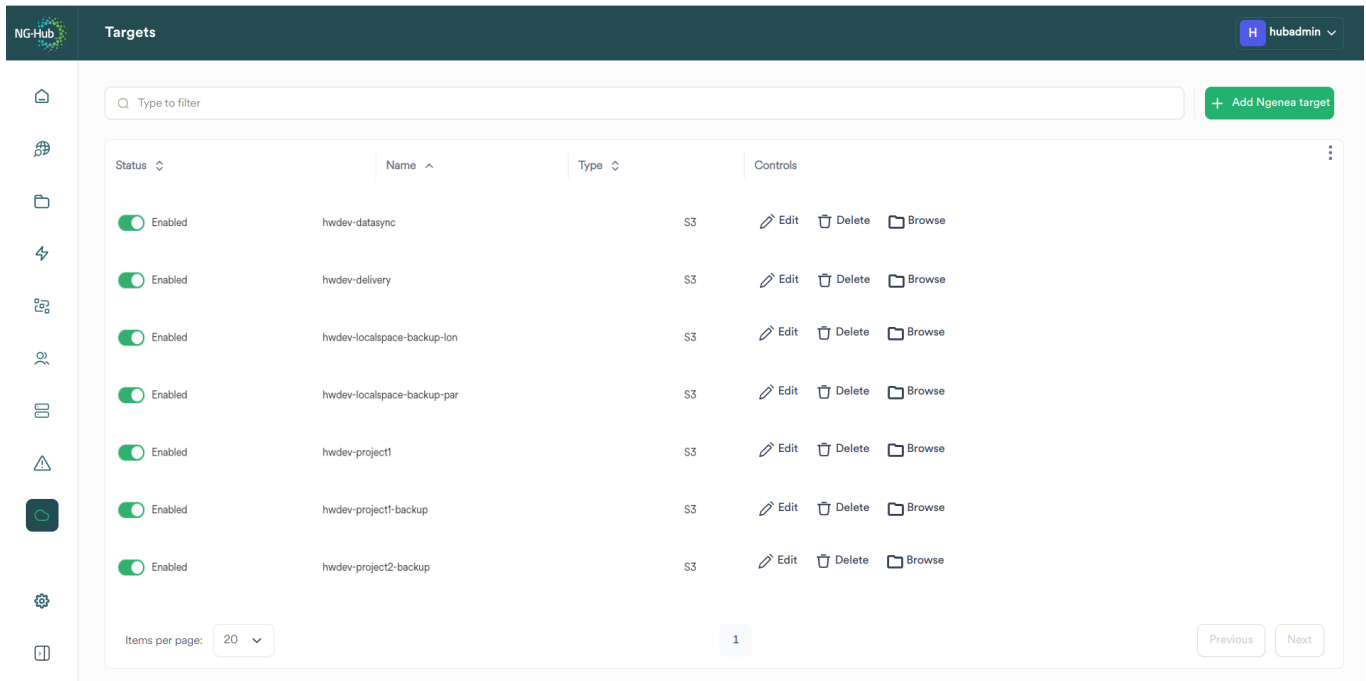
Ngenea provides the capability to off-site data, import data, or send data between pixstor sites. The destination for the data is known as a target.

Ngenea targets provide the association between the pixstor file system and the target.

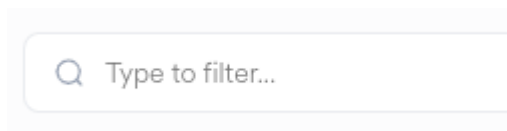
Typically a target is mapped to the location of a Space on the pixstor file system.

Viewing Targets

Clicking the Targets button in the main menu bar displays the list of Ngenea targets:



Filtering Targets



To filter the list of Ngenea targets, type the target or part of a target name in the filter bar.

Target Wizard

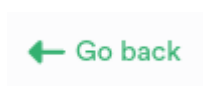
Navigating the Wizard



Click the close button to exit the wizard. Changes are not saved.



Click the Next button to advance to the next page of the wizard. The Next button is disabled until all required page elements are completed.



Click the Go Back button to return to the previous wizard page.



Click the Finish & Create button to apply the changes displayed on the wizard summary page.

Adding a Target

Important: This function can only be performed by a Hub Administrator.

+ Add Ngenea target

Click the Add Ngenea target button to start the Ngenea Target Wizard

Type & reference

Ngenea supports the following storage target types:

Storage Target Type	Description
S3	AWS S3 and S3 compatible targets
Microsoft Azure	Azure Blob Storage
SpectraLogic BlackPearl	Spectralogic DS3 targets
Google Object Storage	Google Cloud Storage
POSIX file system mount	Locally mounted targets, such as NFS

Select the required target type from the drop down menu.

Type & reference

Configure Ngenea target basic settings

Storage target type

S3

Learn about [StorageTarget](#)

Ngenea Target reference

myproject

Reference key for the Ngenea Target, e.g myBucket

Available on sites

2 sites selected

London

Paris

Enter a 'friendly name' for the Ngenea target reference.

E.G.:

- myproject (such as 'deepspace')
- mydepartment (such as 'graphics')
- a descriptive phrase (such as 'referencematerial')

Select the sites to create the target on.

Depending on the Storage target type selected, a Type-specific configuration page will be shown after selecting Next.

Type-specific config AWS S3 or compatible

Type-specific config

Configure the Ngenea target according to the selected target type

Bucket

Type the bucket

[Learn about Bucket](#)

Access key ID

Type the access key ID

[Learn about AccessKeyId](#)

Secret access key

Type the secret access key

[Learn about SecretAccessKey](#)

Region

Type the region

[Learn about Region](#)

Storage server address

s3.amazonaws.com

X

Host name of the storage address. [Learn about Endpoint](#)

Storage server port

443

X

Port of the storage service. [Learn about Port](#)

Scheme

HTTPS

V

Connection protocol. [Learn about Scheme](#)

Verify SSL

Validate SSL certificates of the target

Enter the settings as required, which must match those set in the S3 object storage provider:

Setting	Description
Bucket	The name of the storage bucket as specified at the object storage service.
Access key ID	The unique access key for the AWS user account performing data transfers. This will be stored encrypted.
Secret access key	The unique security key for the AWS user account performing data transfers. This will be stored encrypted.
Region	The AWS (or S3 compliant provider) region hosting the S3 Cloud Storage
Storage server address	Not used for Amazon S3 Cloud Storage. For services which reside at specific IPs, such as AWS Snowball, MinIO or LocalStack, specific the host or IP address to connect to.
Storage server port	The TCP/IP port used to communicate
Scheme	HTTP or HTTPS transfer. HTTPS is recommended. Data integrity cannot be guaranteed over HTTP transfer schemes.
Verify SSL	Whether to verify the SSL connection of the target. Disabling the SSL verification allows connections to storage targets which do not provide valid SSL certificates. Connecting to invalid SSL certificates is insecure.

Type-specific config Microsoft Azure

Type-specific config

Configure the Ngenea target according to the selected target type

Access key

Type the access key

Learn about [AccessKey](#)

Scheme

HTTPS

Connection protocol. Learn about [Scheme](#)

Container

Type the container name

Learn about [Container](#)

Storage account

Type the storage account

Learn about [StorageAccount](#)

Enter the settings as required, which must match those set in the Azure object storage provider:

Setting	Description
Access key ID	The unique access key for the Azure user account performing data transfers. This will be stored encrypted.
Scheme	HTTP or HTTPS transfer. HTTPS is recommended. Data integrity cannot be guaranteed over HTTP transfer schemes.
Container	The storage container for the blob data.
Storage account.	The Azure namespace containing the Container

Type-specific config Spectra Logic BlackPearl

Type-specific config

Configure the Ngenea target according to the selected target type

Bucket

Type the bucket

Learn about [Bucket](#)

Access key ID

Type the access key ID

Learn about [AccessKeyId](#)

Secret access key

Type the secret access key

Learn about [SecretAccessKey](#)

Storage server address

Type the storage server address

Host name of the storage address. Learn about [Endpoint](#)

Enter the settings as required, which must match those set in the BlackPearl DS3 object storage provider:

Setting	Description
Bucket	The name of the storage bucket as specified at the object storage service.
Access key ID	The unique access key for the BlackPearl user account performing data transfers. This will be stored encrypted.
Secret access key	The unique security key for the BlackPearl user account performing data transfers. This will be stored encrypted.
Storage server address	The FQDN hostname of the BlackPearl.

Type-specific config Google Object Storage

Type-specific config

Configure the Ngenea target according to the selected target type

Bucket

Type the bucket

[Learn about Bucket](#)

Google authentication JSON

{ }

X

[Learn about CredentialsJSON](#)

Enter the settings as required, which must match those set in the Google Cloud Storage object storage provider:

Setting	Description
Bucket	The name of the storage bucket as specified at the object storage service.
Google authentication JSON	Enter the contents of the JSON key for the user or service account granted permission to transfer data to Cloud Storage bucket. This will be stored encrypted. For more information refer to Google documentation

Important: If the Ngenea Target is to be used as a backup only target, do not define JSON information in the Google Authentication JSON. Instead define keyword `CredentialsFile` with the path to the file containing JSON credentials on pixstor. For more information refer to [Backup-Only Targets](#).

Type-specific config Filesystem Mount

Type-specific config

Configure the Ngenea target according to the selected target type

Target mount point

Type the target mount point

Mount point of the storage target.
Maps to StoreObjectName, RetrieveObjectName and EnsureMountPoint.
[Learn about MigrationTargetFolder](#)

Enter the settings as required, which must match the location of the NAS mount point on the pixstor:

Important: Target mounts points are site-specific however Hub will create the target across all selected sites.

116

Setting	Description
Target mount point	The location of the POSIX compliant mounted storage. E.G. If the external POSIX storage is mounted on <code>/mmfs1/myexternalnas</code> enter <code>/mmfs1/myexternalnas</code>

Migration

Migration

Configure the migration target

Default migration target

Whether this is the default migration target ☐

File match

Type the regex to match files

Regex filter to match files for migration to this target. Learn more about [LocalFileRegex](#)

Delete on recall

Delete files from the external storage on recall ☐

Learn more about [DeleteOnRecall](#)

Configure the migration settings as required to handle data migration accordingly:

Setting	Description
File match	Refer to the RegEx Filters example table below
Delete on recall	Determines whether to delete the recalled data from the external storage target after the data has been successfully recalled.
Backup-only target	Determines whether the target will be used to hold backup, not live data. Required to backup a Space. Refer to Backup Only brand_backend_name Targets

Example regex filters where a space named myspace is present on the pixstor filesystem at location `/mmfs1/data/myspace`:

RegEx Filter Examples	Outcome
<code>/mmfs1/data/myspace/(.*)</code>	Data with the myspace folder is eligible for Ngenea operations. Dehydrated files within the myspace folder are present at the root of the storage target.
<code>/mmfs1/data/(myspace/.*)</code>	The myspace directory and data within is eligible for Ngenea operations. The

RegEx Filter Examples	Outcome
	myspace folder is present at the root of the storage target.
<code>/mmfs1/data/myspace/subdirectory/(.*)</code>	Data with the subdirectory of the myspace directory is eligible for Ngenea operations. Dehydrated files within the subdirectory of the myspace directory are present at the root of the storage target. The myspace directory is not present. Data immediately within the myspace directory (other than that within the subdirectory) is not eligible for Ngenea operations.
<code>/mmfs1/data/myspace/(subdirectory/.*)</code>	The subdirectory of the myspace directory and data within is eligible for Ngenea operations. The subdirectory of the myspace directory is present at the root of the storage target. The myspace directory is not present. Data immediately within the myspace directory (other than that within the subdirectory) is not eligible for Ngenea operations.

Advanced configuration

Advanced configuration

Define a custom configuration setting for the ngenea target

 **Keyword** 

String ▼

Type the value

+ Add new key

To add configuration settings to a target to control specific behaviour during Ngenea data operations.

+ Add new key

Click the Add new key to define a new configuration setting

 **Keyword**
.....

Enter the name of the configuration setting in the Keyword field

Select the type of configuration setting from the drop down menu. Choose or enter the value for the configuration setting.

Boolean

Example of added Advanced configuration settings:

+ Add new key

Summary

Upon completing the wizard steps a summary is presented:

[← Go back and edit](#)

Click the Finish & Create button to apply the changes displayed on the wizard summary page.

Alternatively **Go back** and change the proposed configuration as required or **close** the wizard to cancel the creation of the Ngenea target.

Editing a Target

Important: This function can only be performed by a Hub Administrator.

Clicking the Targets button in the main menu bar displays the list of Ngenea targets:

NG-Hub

Targets

H hubadmin

Type to filter

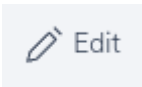
+ Add Ngenea target

Status	Name	Type	Controls
Enabled	hwdev-datasync	S3	EditDeleteBrowse
Enabled	hwdev-delivery	S3	EditDeleteBrowse
Enabled	hwdev-localspace-backup-lon	S3	EditDeleteBrowse
Enabled	hwdev-localspace-backup-par	S3	EditDeleteBrowse
Enabled	hwdev-project1	S3	EditDeleteBrowse
Enabled	hwdev-project1-backup	S3	EditDeleteBrowse
Enabled	hwdev-project2-backup	S3	EditDeleteBrowse

Items per page: 20

1

PreviousNext



Click the edit icon on the required Ngenea target row to edit the Ngenea target

- Modify the Ngenea target settings as required. Refer to [Adding a Target](#) for settings guidance.



Type & reference

hwdev-project1

☒ Enabled

Type-specific config

Type & reference

Migration

Key setup

Storage target type

S3

▼

Learn about [StorageTarget](#)

Type-specific config

Bucket

hwdev-project1

×

Learn about [Bucket](#)

Access key ID

Type the access key ID

Leave empty to keep this unchanged.

Secret access key

Type the secret access key

Leave empty to keep this unchanged.

Region

eu-west-1

×

Learn about [Region](#)

Storage server address

Save

Deleting a Target

Important: This function can only be performed by a Hub Administrator.

Clicking the Targets button in the main menu bar displays the list of Ngenea targets:

Targets

hubadmin

Type to filter

Add Ngenea target

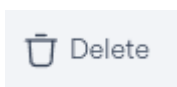
Status	Name	Type	Controls
Enabled	hwdev-datasync	S3	
Enabled	hwdev-delivery	S3	
Enabled	hwdev-localspace-backup-ion	S3	
Enabled	hwdev-localspace-backup-par	S3	
Enabled	hwdev-project1	S3	
Enabled	hwdev-project1-backup	S3	
Enabled	hwdev-project2-backup	S3	

Items per page: 20

1

Previous

Next



Click the delete icon on the required Ngenea target row to delete the Ngenea target

A confirmation dialog is raised:

Delete Ngenea target



Do you wish to delete this Ngenea target?

Yes

No

- Click Yes to delete the Ngenea target. This action is irreversible.
- Alternately click no, or close the confirmation dialog.

Browse a Target

Important: This function can only be performed by a Hub Administrator.

Clicking the Targets button in the main menu bar displays the list of Ngenea targets:

Targets

hubadmin

+ Add Ngenea target

Status	Name	Type	Controls
Enabled	hwdev-datasync	S3	
Enabled	hwdev-delivery	S3	
Enabled	hwdev-localspace-backup-ion	S3	
Enabled	hwdev-localspace-backup-par	S3	
Enabled	hwdev-project1	S3	
Enabled	hwdev-project1-backup	S3	
Enabled	hwdev-project2-backup	S3	

Items per page: 20

1

Previous
Next



Browse

Click the browse icon on the Ngenea target row to browse the Ngenea target

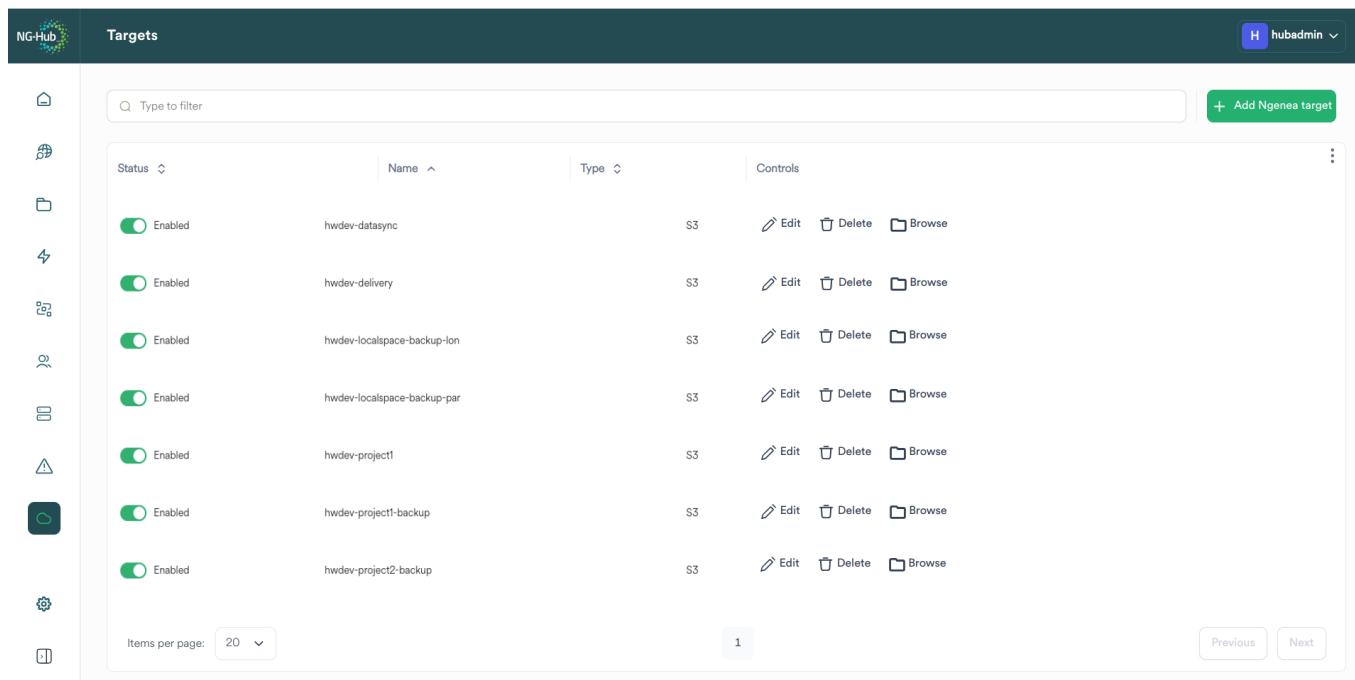
For more information on browsing and importing from an Ngenea target, see [Target Import](#)

Target Import

An Ngenea target can be browsed, and files can be imported to the local PixStor file system.

Important: This function can only be performed by a Hub Administrator.

Clicking the Targets button in the main menu bar displays the list of Ngenea targets:

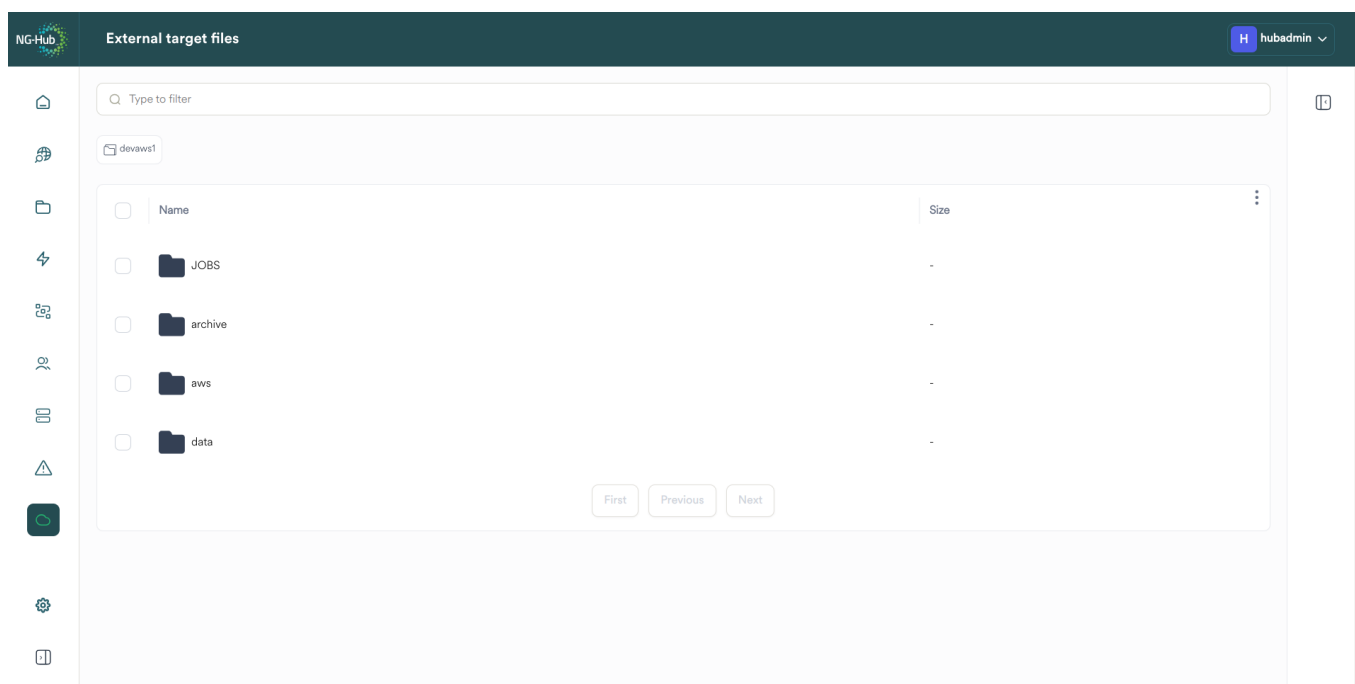


Click the browse icon on the Ngenea target row to browse the Ngenea target

Browsing a Target

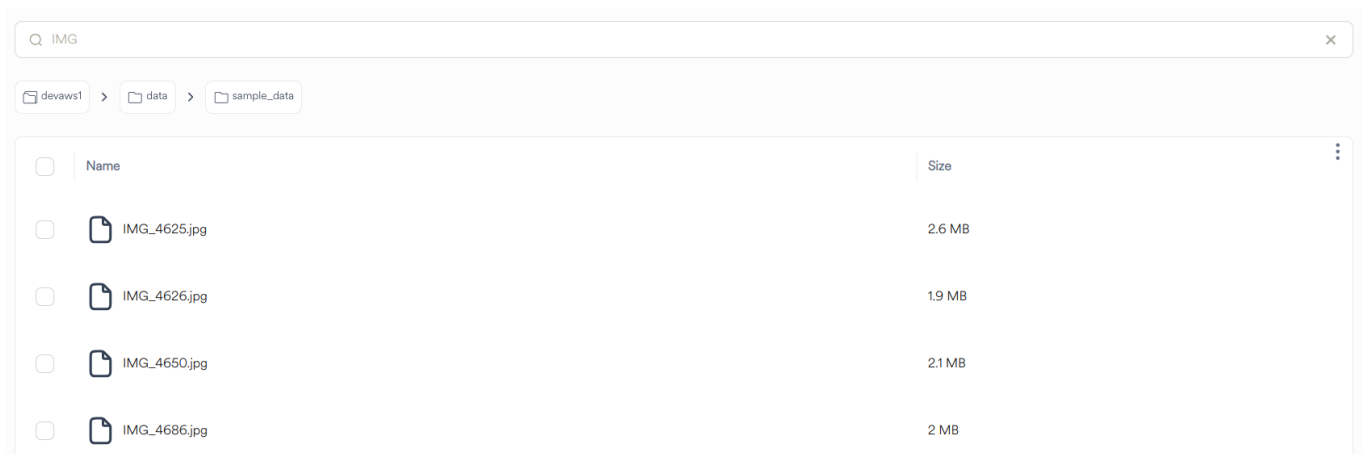
The target browser displays the file and folder contents of a target including additional contextual information such as size.

The target browser screen is comprised of several areas.



Filtering the Target Browser

To display files or folders matching keywords, enter the keywords in the filter bar.



Target Browser File Attributes

The target browser displays information regarding the files and folders within the Target.

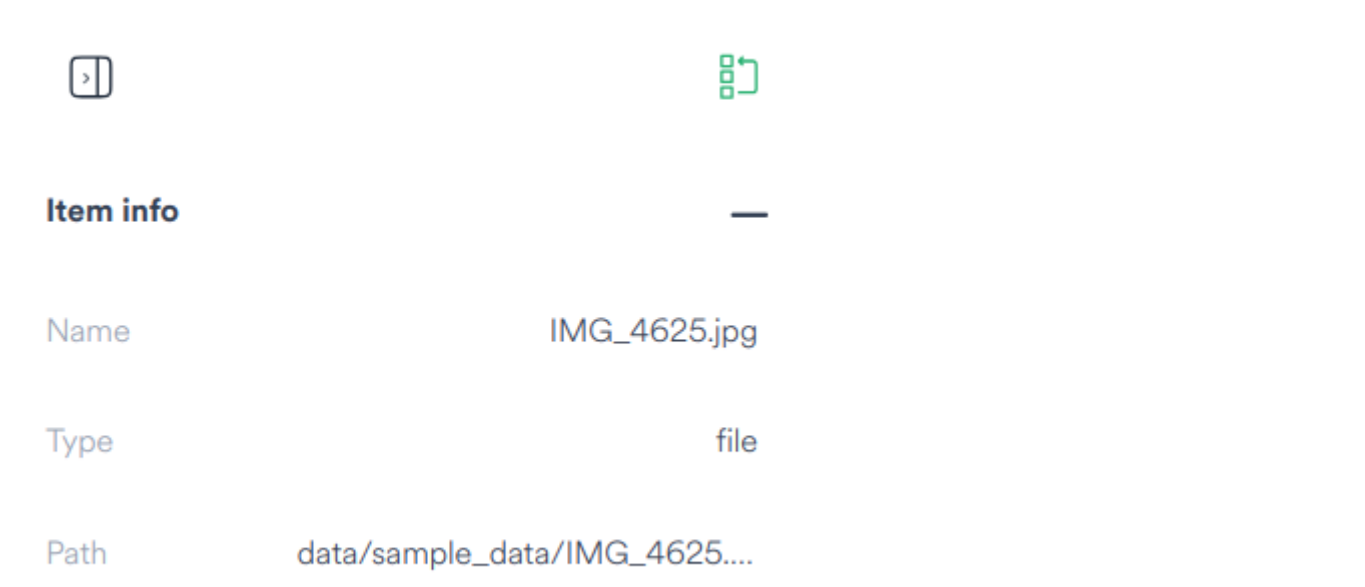
Available information may vary between target types.

Option	Description
Name	The file or directory name
Size	The size of the file on the target

Viewing File or Folder Metadata

Metadata Panel

Click a file or directory name to view the associated metadata:



The metadata panel provides a summary of metadata.

Using the Job Creator Panel

Explain panel areas, buttons and operations

Workflow: 🔄 Import to Site View selection list ✕ Clear selections

Source: London Queue: default

Location: /mmfs1/data/JOBS Change location ↗ Next >

1 items

Selecting files and directories

Selecting an individual file or directory populates the selection list with the item. Selecting a directory populates all items within the chosen directory tree.

Selections of the entire Target or a directory are each counted as 1 item.

Select the 'select all' checkbox next to the Name field in order to select the entire Target.

☐	Name
☒	 001.file
☒	 002.file
☒	 003.file

As selections are added or removed, the selected item count is updated:

5
items

 [View selection list](#)

Click the View Selection List button to show the files and directories selected to be processed by the chosen workflow

✕ [Clear selections](#)

Click the Clear selections to remove all files and directories currently selected from the selection list

[Choose location ↗](#)

Select location

LDN > jobs

London

Type to filter

Name	Size
sub	512 B

Items per page: 20

1

2

3

4

...

101

Confirm

Note: Target files can only be imported to Locations in Spaces which match the target's "File match" pattern.

Next >

Click the Next button to enact the Job

Jobs with Workflows which require additional user decisions prior to enacting the Job raise the Configure workflow fields dialog:

127

Configure workflow fields

Hydrate

Hydrate

Submit

After entering the field information [if required], select the button at the bottom of the dialog to submit the Job.

Click the button at the bottom of the dialog to submit the Job.

Submit

Job #87498 is created.
Click [this link](#) to see the job.

Upon submitting the workflow, a notification indicates the job was created with a link to view details of that job

Job Types (Workflows)

Workflow:

Import to Site

View selection list

Clear selections

Source

London

Queue

default

1 items

Location

/mmfs1/data/JOBS

Change location

Next >

Default Workflows

Import to Site

The Import to Site workflow transfers the file metadata and data from the Ngenea target (E.G. an AWS cloud bucket). After successful transfer the metadata and data content of the file is present on the chosen site, within the chosen Location.

Global Settings

The global settings page controls settings which are applied across all pixstor sites participating in hub management.



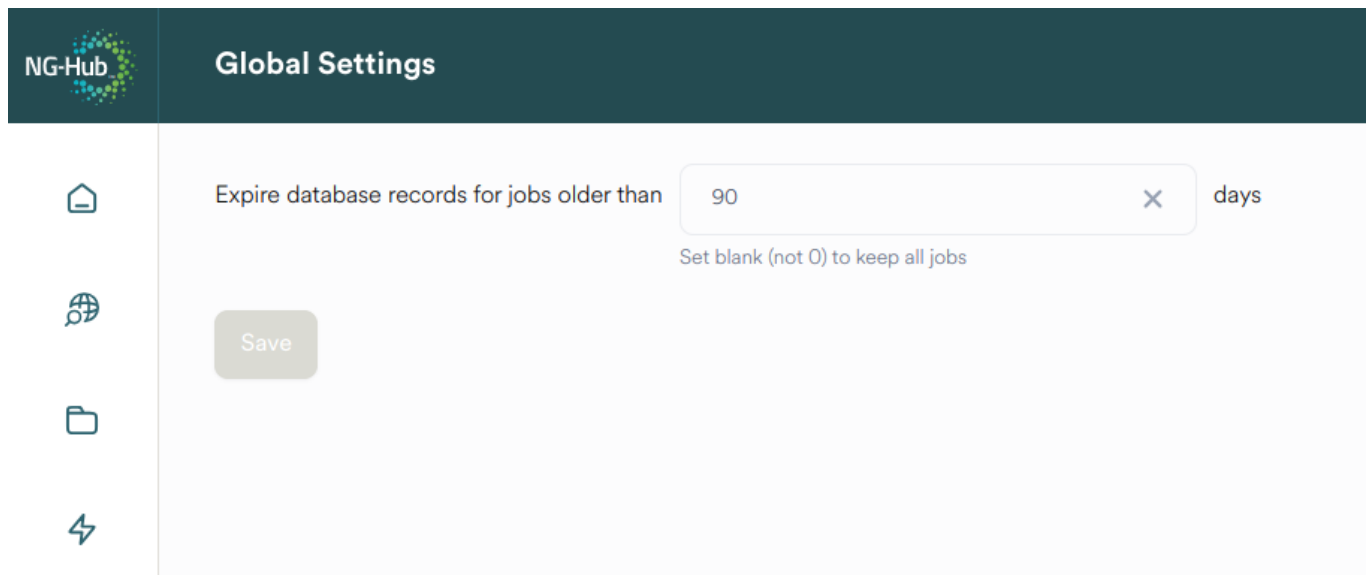
Click the settings button to navigate to the Global Settings screen

Database

Old job data can be retained in the database for a specific number of days so that the hub database does not fill up the system.

Database expiry can be configured by setting `jobs_ttl` in global settings page in UI.

Clicking the global settings button displays a form for updating the `jobs_ttl` in days. The `Save` button will be enabled once the user changes the value in the field.

A screenshot of the Global Settings page in the NG-Hub UI. The page has a dark teal header with the NG-Hub logo and the title "Global Settings". On the left is a sidebar with icons for Home, Global Settings (active), Jobs, and Settings. The main content area shows a form for "Expire database records for jobs older than". It features a text input field with the value "90", a clear button (X), and the unit "days". Below the input field is a hint: "Set blank (not 0) to keep all jobs". A "Save" button is positioned below the input field.

Backups

PixStor provides the capability to backup data within a Space on a per-Site basis to specific Ngenea Targets.

Hub enables configuration to be set for the Ngenea Backup service running on pixstor sites.

Important: Best practice is to treat backup data separately from live data. Hub will only configure a backup to an Ngenea target set as a `backup-only` target type.

Configuring Backups

To perform a backup for a Space on a Site, the following actions must be undertaken:

- An Ngenea Target is provisioned with Backup-Only enabled
- The Site from which to backup is enabled to participate in backups
- An appropriate backup schedule is set
- The Space is enabled for backup, selecting the Site and the Ngenea Target

Backup-Only Targets

An Ngenea Target can be utilised for backups of a Space.

Where a Space exists across multiple Sites, optionally, each site can perform its own backup of the Space.

Important: Multiple Sites are not permitted to backup to the same Ngenea Target.

Backup-only target

Whether this target is for backups

Enabling Site for Backup

pixstor provides the capability to backup data within a Space on a per-Site basis to specific Ngenea Targets.

Hub enables configuration to be set for the Ngenea Backup service running on pixstor sites.

If the Site is enabled to participate in backups, each Space requires additional configuration to enable the per-Space backup.

Backup

Enable backup 🔴

Do you want this site to participate in backups?

Frequency

☐ Mins or Hours

☒ Daily or Weekly

Every ✓ Mon ✓ Tue ✓ Wed ✓ Thu ✓ Fri ✓ Sat ✓ Sun

At 08 ▾ : 00 ▾ **UTC+0**

Determine the required frequency of the backup.

Schedule

Frequency

☒ Mins or Hours

☐ Daily or Weekly

Interval 6 × hours ▾

Choosing Mins or Hours will ensure that the schedule will run on the next interval set.

E.G: * 1 hour: The backup will run on the next hour (12.00, 13.00) * 15 mins: The backup will run on the next 15 minute interval past the hour (15, 30, 45, 00)

Schedule

Frequency

☐ Mins or Hours

☒ Daily or Weekly

Every

☒ Mon

☐ Tue

☒ Wed

☒ Thu

☐ Fri

☒ Sat

☒ Sun

At

12 ▾

:

05 ▾

UTC+0

Choosing Daily or Weekly allows the backup to be scheduled once per chosen day at a specific time of day.

Hint: The schedule time is in UTC+0. You may need to account for any timezone offset of the site when scheduling.

Optional tuning can be set for the backup operation in the Advanced Configuration.

Refer to the Ngenea Backup documentation prior to applying any parameters.

Advanced configuration

Leave fields blank to use defaults

Number of threads

Type a number

Hashing threshold (MB)

Type a number

Timeout (seconds)

Type a number

Connection timeout (seconds)

Type a number

Number of connection retries

Type a number

Enabling Space Backup

Space backup

Site(s) to backup

2 sites selected



LON x

ROM x

Global exclusion rules

Exclude folders

Pick folders to exclude ↗

Exclude filetypes

Type the file extension and hit Return key

- Select the Sites which will perform backups for this space
- Define any Global Exclusions

Global Exclusion Rules

Global exclusions apply to all Site backup configurations for the Space unless overridden on a per-Site basis.

Pick folders to exclude ↗

Click the Pick folders to exclude button to raise the folder browser dialog

Use the folder browser to unselect items which will not be included (hence excluded) from backups.

Unselect items to exclude ✕

📁 datasync > 📁 dubrovnik

Paris ▾

🔍 Type to filter

Name	Size	Number of files	Date created	⋮
☒ 📁 dpx	8.2 GB	251	Wed, 24 Jan 2024, 10:24:17 GMT	
☒ 📁 png	2.8 GB	250	Wed, 24 Jan 2024, 10:24:18 GMT	

Items per page: 20 ▾

1

Previous

Next

Confirm

Exclude filetypes

Type the file extension and hit Return key

Define file extensions which must be excluded from backups across all Sites.
E.G. *.tmp

Hint: If multiple Sites are selected, backup configuration is presented per-Site. Optionally each site can specify its own settings.

For each Site participating in Space backup a configuration panel is displayed:

PAR

paris backup settings

^

Exclusion rules

Do you want to use the global exclusion rules from above for this site?

Ngenea target

Choose where to store the backed up files

Number of versions

2

Inactive threshold (days)

Type a number

If left blank or 0, automatic tier change of inactive files is disabled.

ACL save

Do you want ACL entries to be saved?

Shadow folder metadata save

Do you want shadow folder metadata to be saved?

Review the backup settings for each site:

Setting	Description
Exclusion Rules	By default the Site inherits the Global exclusion rules. To override the rules, deselect the slider and define any exclusions using the same methods as described above.
Ngenea Target	Choose an available Ngenea Target. Only targets configured as Backup-Only target and not currently in use for other Sites are available for selection.
Number of versions	Defines the maximum number of all file versions within a remote storage location for the targeted Fileset before Ngenea Backup prunes excess versions. If the number of versions of a file has exceeds the defined threshold, the oldest version will be deleted.
Inactive threshold(days)	

Setting	Description
	Defines the threshold of days for a defined CRITERIA (E.G.: modification) after which each file within the monitored Independent Fileset is validated when INACTIVE_THRESHOLD is enabled. Those files where their defined CRITERIA exceeds this value will be re-copied from the local PixStor filesystem to storage container into an in-active tier. All instances of the file in typical standard storage classes are transitioned to older file versions.
ACL save	If enabled, all of the related NSFv4 ACL entries for an ingested file will be included in the remote object's metadata when a file is backed up. This allows Ngenea to apply those NSFv4 ACL entries when recalled or stubbed.
Shadow folder metadata save	If enabled, additional data objects will be created within the remote storage target as this data is stored as an object within the cloud provider. If enabled, all the directory modification, creation and deletion operations between subsequent backup runs will be stored in remote storage. This allows the restoration or ngrecall of directories with NFSv4 ACL and POSIX permission support. This data will also be version controlled similarly to files. This operation will occur if the permissions or ACL entries are edited on any directories meaning that each change to ACLs will be updated with regular runs of backup.

Administration Guide

Ngenea Hub harnesses the power of [Ngenea](#) to provide global workflows, enabling your data to be where you need it, when you need it.

Installation

Notice

Hub versions 1.18+ have additional requirements

If an upgrade is being performed from a version < 1.18 please note changes in the Installation and Upgrade guides

A deployment of Ngenea Hub comprises two main components:

- **Ngenea Hub** - the central management point, from which all tasks are managed
- **Ngenea Worker** - worker agents, installed on individual Ngenea servers, that execute the Ngenea tasks

Ngenea Worker are installed on one or more nodes in a Site, which typically represents a single Ngenea cluster or location.

Ngenea Hub needs to be accessible from all Ngenea Workers on the following ports:

- 6379/tcp
- 5672/tcp
- 8000/tcp

8000/tcp is required for Hub versions >= 1.18

Installing Ngenea Hub

Ngenea Hub can be installed in a number of ways, use one of the methods described below.

CentOS / Redhat - Online Installation

Configure Docker Authentication

Note: This step is not required on PixStor systems

Configure Docker authentication for `eurepo.arcapix.com`.

```
docker login eurepo.arcapix.com
```

Installing Ngenea Hub

Transfer the `ngenea-hub` rpm to the target system.

Install the `ngenea-hub` package via yum.

```
yum install ngenea-hub-<version>.rpm
```

Optionally, create an initial Ngenea Hub configuration file at `/etc/sysconfig/ngeneahub`. This file contains the credentials which will be required for deploying workers, and can also be edited to use external queue systems prior to starting the Ngenea Hub service. This file will be created automatically if it does not exist when the service is started.

```
ngeneahubctl createconfig
```

Enable and start the Ngenea Hub service.

```
systemctl enable --now ngeneahub
```

Enable and start the Ngenea Hub frontend service.

```
systemctl enable --now ngeneahub-frontend.path
```

Check the status of the service with:

```
ngeneahubctl status  
systemctl status ngeneahub-frontend.path
```

CentOS / Redhat - Offline installation

The `ngeneahub` service will attempt to pull the required docker images from the Ngenea software repository servers. In situations where this is not possible (due to network restrictions, for instance), the containers can be installed via additional RPM: `ngenea-hub-images`, available at the same location as the main RPM.

Once the RPMs are transferred to the target system, they can be installed using `rpm`.

```
rpm -ivh ngenea-hub-<version>.rpm ngenea-hub-images-<version>.rpm
```

Cloud Deployment

Coming soon: image-based deployment of Ngenea Hub in the cloud.

Container Native Deployment

It is possible to deploy Ngenea Hub using standard container management tools and processes.

Please [contact us](#) to discuss.

Ngenea Worker

Pre-requisite: Ngenea Server software installed and configured

Pre-requisite: File systems accessible on nodes where `ngenea-worker` is installed are configured with `auto-inode-limit`.

This can be achieved via: `mmchfs <file system name> --auto-inode-limit`

Install the `ngenea-worker` package via `rpm`.

```
yum install ngenea-worker
```

Add the appropriate worker [configuration](#).

Note: The following section is new for version 1.18+

Run the join command to ensure TLS communications. Provide the username and password of a registered hub admin user to successfully authenticate. Refer to [Hub Initial Configuration](#) to add a hub admin user.

```
ngenea-worker join --user <USERNAME>
```

Example of ngenea-worker join with worker configuration option `api_secure=false`:

```
root@myserver:/root ngenea-worker join --user hubadmin
hubadmin's password:
Attempting to auth the hub using the provided credentials...
Authenticated as user onprem-worker.
Storing the client certificates for the worker...
Client certificate stored at /etc/pki/tls/certs/onprem.crt
Client private key stored at /etc/pki/tls/private/onprem.key
Storing the root CA for NgeneaHub...
Root CA stored at /etc/pki/tls/certs/ng-hub-ca.crt
```

Enable and start the `ngenea-worker` systemd unit.

```
systemctl enable --now ngenea-worker
```

Product Interoperability Matrix

Ngenea Hub requires specific versions of Ngenea.

- All Ngenea Hub managed sites must use the same version of Ngenea
- Mixed versions are unsupported

Ngenea

The following table defines the supported Ngenea versions for each version of Ngenea Hub.

Ngenea Hub version	Ngenea version	minimum	Ngenea version	maximum
2.5.4	1.30.1-1		1.30.1-1	
2.5.3	1.30.1-1		1.30.1-1	
2.5.2	1.30.1-1		1.30.1-1	
2.5.1	1.30.1-1		1.30.1-1	
2.5.0	1.30.1-1		1.30.1-1	
2.4.3	1.30.1-1		1.30.1-1	
2.4.2	1.30.1-1		1.30.1-1	
2.4.1	1.30.1-1		1.30.1-1	

Ngenea Hub version	Ngenea version	minimum	Ngenea version	maximum
2.4.0	1.30.1-1		1.30.1-1	
2.3.1	1.29.0-2		1.29.0-2	
2.3	1.29.0-2		1.29.0-2	
2.2.5	1.28.0		1.28.0	
2.2.4	1.28.0		1.28.0	
2.2.3	1.28.0		1.28.0	
2.2.2	1.28.0		1.28.0	
2.2.1	1.28.0		1.28.0	
2.2	1.28.0		1.28.0	
2.1	1.27.0		1.27.0	
2.0.2	1.25.0		1.25.0	
1.30.0	1.27.0		1.27.0	
1.29.0	1.27.0		1.27.0	
1.28.0	1.26.1		1.26.1	
1.27.0	1.25.0		1.25.0	
1.26.0	1.24.1		1.24.1	
1.25.0	1.24.1		1.24.1	
1.24.0	1.21.0		1.22.0	
1.23.0	1.21.0		1.22.0	
1.22.0	1.21.0		1.22.0	
1.21.0	1.21.0		1.21.0	
1.20.0	1.21.0		1.21.0	
1.19.0	1.21.0		1.21.0	
1.17.3	1.20.1		1.20.1	
1.17.0	1.19.0		1.19.0	
1.13.0-1.16.0	1.16.0		1.19.0	
1.10.0-1.12.0	1.15.0		1.16.0	
1.9.0	1.14.0		1.14.0	
1.8.0	1.13.0		1.14.0	
1.7.0	1.12.0		1.12.0	
1.6.0	1.12.0		1.12.0	
1.5.0	1.12.0		1.12.0	
<=0.6.0	1.9.0		1.11.0	

Event Based Orchestration

The following table defines the supported Event Based Orchestration versions for each version of Ngenea Hub.

Ngenea Hub version	Event Based Orchestration version
1.30.0	1.0.0

Once installed, additional features such as Search may be set up as described in [Feature Set-up](#)

Ngenea Hub software must align with Ngenea software. Ensure to review the [Product Interoperability Matrix](#) and align software versions accordingly.

Upgrade

Note: Hub versions 1.18+ have additional requirements

If an upgrade is being performed from a version < 1.18 please note changes in the Installation and Upgrade guides

Ensure to review the [Product Interoperability Matrix](#) and align software versions accordingly.

Note: Hub versions 2.3+ have additional requirements

If an upgrade is being performed from a version < 2.30 please note changes in the Installation and Upgrade guides

Prior LDAP connections require to be re-authenticated, or preferably via Kerberos. Ensure to review the [LDAP / Active Directory Login](#) LDAP and Kerberos sections applying the preferred setup accordingly.

Before you start

Before upgrading, you must wait for any pending or active jobs to complete, otherwise they may be lost or present an incorrect future state.

Scheduled workflows must be temporarily disabled prior to upgrading to prevent new jobs being submitted during the upgrade process.

Backup Workflows

Ngenea Hub ships with some default workflows. New releases may make changes to these workflows, so any customisations to them may be lost during upgrade. For safety, workflows should be backed-up before upgrading.

The easiest way to backup workflows is using [ngclient](#) or by using Ngenea Hub backup management command [Backing Up HUB Static Configurations](#)

```
ngclient workflows list > workflows_backup.json
```

See [NGCLIENT-WORKFLOWS](#) for more information.

Stop Services

Note: Hub versions 2.1+ have additional requirements

When upgrading Hub 2.1+, the additional service `ngenea-worker-frontend.path` is required to be actioned

First, shutdown Ngenea Worker on all nodes

```
systemctl stop ngenea-worker
```

Note that any Ngenea Worker packages pre-1.12.0 will use the older syntax:

```
systemctl stop ngenea-worker@SITENAME
```

Then shutdown Ngenea Hub

```
systemctl stop ngeneahub  
systemctl stop ngeneahub-frontend.path
```

Warning: Not stopping the workers before upgrading may result in jobs being stuck as `PENDING`. If this happens, restarting the workers using `systemctl restart ngenea-worker` will allow jobs to start running again.

Upgrade Packages

Download the latest RPMs from the Download page.

Upgrade Ngenea Hub

```
yum upgrade ngenea-hub-<version>.rpm
```

As with [Installation](#), for offline situations, the Ngenea Hub base and image rpms can be upgraded with

```
rpm -Uvh ngenea-hub-<version>.rpm ngenea-hub-images-<version>.rpm
```

Upgrade Ngenea Worker on all nodes

```
yum upgrade ngenea-worker-<version>.rpm
```

Upgrade Configurations

Warning: If upgrading from Ngenea Hub version 1.17 or older, you must convert worker configuration to 1.18+ format. Please refer to the appropriate worker configuration and worker join steps as described in [Ngenea Worker](#). Installation of the worker is not required as this has been achieved in the prior Upgrade Packages step.

Ngenea Hub versions 2.3+ have additional requirements. If an upgrade is being performed from a version < 2.30 please note changes in the Installation and Upgrade guides Prior LDAP connections require to be re-authenticated, or

preferably via Kerberos. Ensure to review the [LDAP / Active Directory Login](#) LDAP and Kerberos sections applying the preferred setup accordingly.

Restart Services

First, startup Ngenea Hub

```
systemctl start ngeneahub  
systemctl start ngeneahub-frontend.path
```

Then start Ngenea Worker on all nodes

```
systemctl start ngenea-worker
```

If any scheduled workflows were disabled, they can now safely be re-enabled.

Validation

Check the status of the Ngenea Hub service with:

```
ngeneahubctl status  
systemctl status ngeneahub-frontend.path
```

Check the status of Ngenea Worker service with

```
systemctl status ngenea-worker
```

Restoring Workflows

Check the workflows post-update. If there are any issue or inconsistencies with the upgraded workflows, they can be restored from the backup created pre-upgrade. This is also done using `ngclient`.

Any workflow which is missing can be re-imported using

```
ngclient workflows import <workflow_file>
```

Any workflow which has changed can be restored using

```
ngclient workflows update <id> <workflow_file>
```

Note, `ngclient` only allows importing or updating single workflows at a time. The `workflow_file` passed to the above commands must only contain a single workflow definition.

Configuration

Hub Initial Configuration

Once all services are up, create an admin user with:

```
ngeneahubctl adduser
```

Register an initial site (replacing `SITENAME`) (see [Worker Installation](#) for enabling a site's worker agents).

```
ngeneahubctl addsite SITENAME
```

Log into the UI at <http://server.address:8000>.

Hub Configuration

Settings

The main configuration file for Ngenea Hub is at `/etc/sysconfig/ngeneahub`. This is an environment file which holds the information required for connecting to the various backend services.

All of these mandatory settings are defaulted by Ngenea Hub if not present.

Mandatory Settings

Setting	Description
DJANGO_SECRET	Secret string used secure signed data within django
POSTGRES_DB	Internal database name
POSTGRES_USER	Internal database username
POSTGRES_PASSWORD	Internal database password
WORKER_THREADS	The default number of celery concurrency threads to deal with multiple queued tasks for all celery based containers. This can be overwritten by container specific settings. Defaults to 2.
DAG_THREADS	The number of celery-dag container concurrency threads to deal with multiple queued DAG based tasks. Defaults to 7. More threads result in faster task resolutions for all job types and more frequent job updates.
CELERY_BROKER	Choice of broker must be either RABBITMQ or REDIS, defaults to REDIS
RABBITMQ_USER	Username for the rabbitmq broker, must be set regardless of CELERY_BROKER, defaults to ngeneahub
RABBITMQ_PASSWORD	

Setting	Description
	Username for the rabbitmq broker, must be set regardless of CELERY_BROKER, defaults to ngeneahub
RABBITMQ_VHOST	Virtual hostname to be used for the rabbitmq broker, must be set regardless of CELERY_BROKER, defaults to nghub
TASK_DAEMON_BATCH_SIZE	Controls the amount of tasks to update the state of STARTED to once a DAG related job has been started. Defaults to 100
JWT_PRIVATE_KEY	The private key to use for ALL JWTs used within the hub for authenticating requests, generates one on first startup
JWT_PUBLIC_KEY	The public key to use for ALL JWTs used within the hub for authenticating requests, generates one on first startup
JWT_EXPOSED_JSON	An exposed json containing details for JWT encryption, generates one on first startup
GF_SECURITY_ADMIN_PASSWORD	The grafana security password for the internal grafana, defaults on first startup
HUB_PORT	User configurable hub port, defaults to 8000
WEB_BIND_IP	User configurable web bind IP, defaults to 0.0.0.0
SHARED_SECRET	Secret key shared internally within the hub services for TLS operations, generated on first startup
DJANGO_ENCRYPTED_FIELDS_KEY	Defines the key for encrypting keys within the database, generated on first startup
DJANGO_EJF_CRYPTER_KEY	Additional cryptography key for obfuscating django responses, generated on first startup

Optional settings

Setting	Description
REDIS_HOST	Address of the Redis queue results store. Defaults to the container service address.
WORKERS	The number of gunicorn workers to spawn for serving API requests. Default to 8.
API_TIMEOUT	The timeout for requests against the API server, in seconds. Default: 600
GATEWAY_TIMEOUT	The timeout for requests going through nginx, in seconds. This should be set to greater or equal than API_TIMEOUT. Default: 600
CONSUMER_TIMEOUT	

Setting	Description
	The timeout for rabbitmq consumer delivery acknowledgement in seconds. Default: 10800000 (3 hours)
PUBLIC_URL	User configurable base url for the hub stack to be served from, must not end in a trailing slash.
CELERY_THREADS	The number of main celery container concurrency threads to deal with multiple queued internal tasks. More threads means that more internal tasks and system tasks can run in parallel.
EVENT_THREADS	The number of celery-events container concurrency threads to deal with multiple queued event reporting tasks. More threads ensure that snapdiff events can be reported to the hub faster.
RESULT_THREADS	The number of celery-results container concurrency threads to deal with multiple queued result reporting tasks. More threads mean streamed task results can report the the hub faster.
HEARTBEAT	(bool) Key for Disabling/Enabling celery heartbeats, default: true (enabled)
GOSSIP	(bool) Key for Disabling/Enabling celery gossip, default: false (disabled)
MINGLE	(bool) Key for Disabling/Enabling celery mingle, default: false (disabled)
REDIS_HEALTH_CHECK_INTERVAL	(int) The Redis backend supports health checks. This value must be set as an integer whose value is the number of seconds between health checks. default: 60
REDIS_TCP_BACKLOG	(int) In high requests-per-second environments you need a high backlog in order to avoid slow client connections issues to redis. Default: 511
REDIS_SOCKET_TIMEOUT	(int) When there are network issues redis backend connection sockets can become stale, this

Setting	Description
	timeout setting will reset the socket connection after this value in seconds after becoming idle and resume operation. Default: 60
CELERY_SOCKET_TIMEOUT	(int) When there are network issues redis broker sockets can become stale, this timeout setting will re-acquire the socket after becoming idle for this value in seconds and resume operation. Default: 60
CELERY_CONNECTION_TIMEOUT	(int) When there are network issues redis broker connection via the acquired sockets can become stale, this timeout setting will reset the connection after becoming idle for this value in seconds and resume operation. Default: 60
EXPIRE_OLD_JOBS_INTERVAL	(cron) schedule for when old job expiration will be run. When the task runs, jobs older than the configured jobs_ttl will be expired. Default: 0 0 * * * (minutes can be random from 0-59)
REMOVE_OLD_SEARCH_RESULTS_INTERVAL	(cron) schedule for when search result removal will run. When the task runs, search results older than the configured search_result_ttl will be expired. Default: 0 0 * * * (minutes can be random from 0-59)
INVALIDATE_CANCELLED_JOB_TASKS_INTERVAL	(cron) schedule for when cancelled jobs are revoked. When the task runs, any tasks still active in a cancelled job will be automatically cancelled. Default: 0 * * * * (minutes can be random from 0-59)
CLEANUP_OLD_EVENTS_INTERVAL	(cron) schedule for when old snapdiff events will be cleaned up. When the task runs, events for all but the 2 most recent completed snapdiff jobs per workflow will be deleted. Default: 0 * * * * (minutes can be random from 0-59)

Setting	Description
INACTIVE_TASKS_INTERVAL	(cron) schedule for when inactive tasks will be invalidated. When the task runs, any STARTED task which is not actually running in a worker will be marked as FAILED. Default: 0 * * * * (minutes can be random from 0-59)
SYNC_SITE_SETTINGS_INTERVAL	(cron) schedule for when sync of the site settings will be run. When the task runs, the site settings will be created or updated in the DB. Default: 0 * * * * (minutes can be random from 0-59)
SYNC_GLOBAL_SETTINGS_INTERVAL	(cron) schedule for when sync of the global settings will be run. When the task runs, the global settings will be sent to sites to be in sync. Default: 0 * * * * (minutes can be random from 0-59)
REFRESH_SITE_ANALYTICS_INTERVAL	(cron) schedule for when refresh of the site analytics will be run. When the task runs, refresh of site analytics will be triggered. Default: 37 */12 * * *
SYNC_STORAGE_POOLS_INTERVAL	(cron) schedule for when sync of the storage pools will be run. When the task runs, storage pools will be synced. Default: */30 * * * *
SYNC_REMOTE_SERVERS_INTERVAL	(cron) schedule for when sync of the remote servers will be run. When the task runs, remote servers will be synced. Default: 0 0 * * * (minutes can be random from 0-59)
SYNC_SPACES_QUOTA_INTERVAL	(cron) schedule for when sync of the spaces' quotas will be run. When the task runs, spaces' quotas will be synced. Default: */30 * * * *
SYNC_SPACES_INTERVAL	(cron) schedule for when sync of spaces will be run. When the task runs, spaces will be synced. Default: 0 * * * * (minutes can be random from 0-59)
SYNC_ALERTS_INTERVAL	(cron) schedule for when sync of alerts will be run. When the task

Setting	Description
	runs, alerts will be fetched from all sites. Default: * * * * *
EXPIRE_OLD_FSOBJECTS_INTERVAL	(cron) schedule for when old fsubjects expiration will be run. When the task runs, fsubjects older than the configured fsubjects_ttl will be expired. Default: 0 0 * * * (minutes can be random from 0-59)
REMOVED_QUEUE_CLEANUP_INTERVAL	schedule for when removed queues are expired and can be recreated or re-used. Default: 24 * 60

Enabling LDAP/Active Directory Authentication

See [LDAP / Active Directory Login](#) for configuration details.

Broker Settings

See [Hub Messaging Queue Configuration](#) for more broker details.

Server Configurations

Some settings are stored in the Ngenea Hub DB.

They can be viewed and changed via the REST API `/api/configurations/` endpoint.

See [Configuration](#) for more details.

Docker Compose configuration

The `docker-compose` file is stored in `/usr/share/ngeneahub/docker/docker-compose.yml`.

This can be extended by creating an override file at `/usr/share/ngeneahub/docker/docker-compose.override.yml`.

Worker Configuration

Note: This configuration is new for versions 1.18+. Best practice is to backup any pre-1.18 worker configuration prior to conversion to the 1.18+ format.

The Ngenea Worker configuration should be added to `/etc/ngenea/ngenea-worker.conf`. The configuration is in ini format. For example:

```
[settings]
site = site1
```

```
api_host = localhost
api_port = 8000
api_secure = true
redis_port = 6379
gpfs_nodes = ["localhost"]
```

Note: `api_secure=true` requires a valid SSL certificate and is not compatible with self-signed SSL certificates.

ngenea-worker defaults to `api_secure=true` if the `api_secure` setting is not specified.

Refer to: [External SSL](#) for provisioning SSL certificates

Alternately, specify `api_secure=false` to disable use of SSL.

The following is a list of available settings:

Option	Type	Default	Required	Description
site	string		Yes	The name of the queue to listen to
api_host	string		Yes	The base url for Ngenea Hub, this will be the url without the https protocol or port.
api_port	string		Yes	The port that the Ngenea Hub is being hosted on, by default within Ngenea Hub it is 8000.
api_secure	string		No	If the API is behind a secure HTTPS connection, by default this is true. Refer to: External SSL for provisioning SSL certificates if <code>api_secure=true</code> .
api_secure_verify	string		No	If the API requests should verify certificates, by default this is true.
threads	int	10	No	The number of concurrent tasks that can be run.
heartbeat	bool	true	No	Key for Disabling/Enabling celery heartbeats, by default Enabled.

Option	Type	Default	Required	Description
gossip	bool	false	No	Key for Disabling/Enabling celery gossip, by default Disabled.
mingle	bool	false	No	Key for Disabling/Enabling celery mingle, by default Disabled.
redis_health_check_interval	int	60	No	The Redis backend supports health checks. This value must be set as an integer whose value is the number of seconds between health checks.
redis_socket_timeout	int	60	No	The Redis results backend supports a socket connection timeout, this value must be set as an integer whose value is the number of seconds.
celery_socket_timeout	int	60	No	The Redis celery broker supports a socket timeout, this value must be set as an integer whose value is the number of seconds.
celery_connection_timeout	int	60	No	The Redis celery broker supports a socket connection timeout, this value must be set as an integer whose value is the number of seconds.
gpfs_nodes	list		No	A list of nodes to run the snapdiff policy scan on as a list of hostnames
enable_plugins	bool	false	No	Key for Disabling/Enabling worker plugin behaviour (currently in alpha), by default Disabled.
loglevel	str	INFO	No	Key for setting the worker loglevel (more specific to worker main process). valid choices are INFO, DEBUG, WARNING, CRITICAL,

Option	Type	Default	Required	Description
				ERROR (by default INFO).

Note: The initial access credentials can be found in the `/etc/sysconfig/ngeneahub` configuration file on the Ngenea Hub server. By default, the `BROKER_PASSWORD` is used for both the `broker_url` and the `result_backend` passwords, and the `BROKER_USER` is used for the `broker_url` username.

Controlling Functions

Ngenea Worker contains multiple internal function queues, each of these are configurable and can be disabled. For details on what these functions are see the [queue function](#).

The following are the configuration values to control them on both a custom or default queue:

For a detailed list of workflow steps and the functions that run them see [Workflow Steps](#)

Custom Queues

Custom job queues can be added to the worker by adding Queue sections to the worker config.

```
[Queue highpriority]
threads = 20
```

For more information on queues, see [Queues](#)

Passing Custom Config File to Ngenea Worker

The custom config file will be present in `/etc/ngenea/ngenea_config_arg.conf` file to run the systemd services. The format of the file is given below.

```
CONFIG=/etc/ngenea/ngenea-worker-custom.conf
```

Note: By default, this file will have the `CONFIG` commented to allow the default config to pass through if custom config file is not specified.

Uncomment the `CONFIG` in `/etc/ngenea/ngenea_config_arg.conf` file. Create a new worker config file (as `ngenea-worker.conf`) with different site name. Also site should be added to hub and worker using the commands listed below.

```
ngeneahubctl addsite <sitename>
ngenea-worker join --user <username> --site <sitename>
```

To Run `ngenea-worker` with multiple sites

To run services with two sites,

1. Create new service file same as `/usr/lib/systemd/system/ngenea-worker.service` with different filename (Eg: `ngenea-worker-site1.service`).
2. Create new custom config file same as `/etc/ngenea/ngenea_config_arg.conf` with different filename (Eg: `ngenea_config_arg1.conf`).
3. Create new worker config file same as `/etc/ngenea/ngenea-worker.conf` with different filename and give the filename in `ngenea_config_arg1.conf`.
4. In `ngenea-worker-site1.service`, change the value of `EnvironmentFile=...` to new custom config file created (Eg: `EnvironmentFile=/etc/ngenea/ngenea_config_arg1.conf`).
5. Now run `systemctl start ngenea-worker.service` and `systemctl start ngenea-worker-site1.service`.

Setting Ngenea Worker debug level in systemd service script

In `/usr/lib/systemd/system/ngenea-worker.service` file, set the value of `Environment=DYNAMO_DEBUG=..` to `true` as below. It will be `false` by default. This debug level is more specific to tasks logging.

```
Environment=DYNAMO_DEBUG=true
```

Hub Messaging Queue Configuration

Selecting Messaging Queue

The messaging between workers and Ngenea Hub is configurable between two different supported brokers. This is controlled with variables in `/etc/sysconfig/ngeneahub` along with additional configuration files.

The following are all broker related configuration values:

Setting	Default	Description
CELERY_BROKER	redis	This controls the messaging queue <code>redis rabbitmq</code>
RABBITMQ_USER	ngeneahub	User for the rabbitmq user for use with Ngenea Hub
RABBITMQ_PASSWORD	Randomly generated password	Password used for the rabbitmq user for use with Ngenea Hub
RABBITMQ_VHOST	nghub	The virtual host used within the rabbitmq instance within Ngenea Hub
REDIS_HOST	redis	Address of the Redis queue results store. Defaults to the container service address.
REDIS_HEALTH_CHECK_INTERVAL	60	The Redis backend supports health checks. This value must

Setting	Default	Description
		be set as an integer whose value is the number of seconds between health checks.
REDIS_TCP_BACKLOG	511	In high requests-per-second environments you need a high backlog in order to avoid slow client connections issues to redis.
REDIS_SOCKET_TIMEOUT	60	When there are network issues redis backend connection sockets can become stale, this timeout setting will reset the socket connection after this value in seconds after becoming idle and resume operation.

For more details on general configuration see [Hub Configuration](#) for more details.

Redis Setting Files

All Ngenea Hub installs come packaged with a default configuration file for redis within `/etc/ngenea/redis/redis.conf`, any edits made to this file will persist with software upgrades. This file can be configured using any values supported by the [Redis configuration](#) page.

Any changes to configuration files for Redis will need a restart of the service.

RabbitMQ Setting Files

If Ngenea Hub is configured to use RabbitMQ, the service can have additional configuration files provided within `/etc/ngenea/rabbitmq/conf.d`. This can be populated with custom configuration files for the wider [RabbitMQ configuration](#).

The provided default configuration files that are within the `/etc/ngenea/rabbitmq/conf.d` are defined and updated in `/usr/share/ngeneahub/rabbitmq`. These will be replaced on upgrade so any changes to the configuration should be done within a custom configuration file.

Any changes to configuration files or additional configuration files for RabbitMQ will need a restart of the service.

Admin Interface

By default, using `rabbitmq` as your messaging queue does not provide any usable management tools, there is however an admin panel that can be enabled by running the following command in the Ngenea Hub host's command line:

This will allow navigation to port `15672` on the Ngenea Hub instance which uses the login details within `/etc/sysconfig/ngeneahub` being `RABBITMQ_USER` and

RABBITMQ_PASSWORD. This interface allows users to manage, view and remove any queues or messages.

Limitations

Currently there is no support for RabbitMQ within the in built grafana support for the hub at /hubmetrics. This means that to see the state of the queues, workers and throughput, the RabbitMQ admin panel

Feature Set-up

To use certain features in Ngenea Hub, additional set-up is required as described in this section.

Search

The search feature provides the ability to search for files across one or more sites.

This page describes the steps required to set up the search feature.

For information on how to use the search feature, see [Search](#)

Prerequisites

The search feature is only supported on workers running on a PixStor.

There are two backends which can provide search functionality to Ngenea Hub: **PixStor Search** and **PixStor Analytics**.

This document assumes that the desired backend has already been deployed, as described in the PixStor deployment guide. Be aware that neither backend will have been deployed by default.

Configuration

Search Backend

By default, Ngenea Hub will use the Analytics backend. If you want to use the PixStor Search backend instead, change the `search_backend` to `pixstor_search`, as described in [Global Configurations](#)

```
$ curl -s -X PATCH 'http://example.com/api/configurations/' -H
'Accept: application/json' -H "Authorization: Bearer
$JWT_ACCESS_TOKEN" -H 'Content-Type: application/json' -d
'{"search_backend": "pixstor_search"}'
```

All sites **must** use the same backend - all analytics or all PixStor Search.

Elasticsearch URL

If you are using the Analytics backend and the worker is running on **PixStor 6**, you will need to change the `elasticsearch_url` for the site to `http://localhost:19200`, as described in [Site-specific Configurations](#)

```
$ curl -s -X PATCH 'http://example.com/api/sites/1/' -H 'Accept: application/json' -H "Authorization: Api-Key $APIKEY" -H 'Content-Type: application/json' -d '{"elasticsearch_url": "localhost:19200"}'
```

If not set, the `elasticsearch_url` will default to `localhost:9200`, which is correct for workers running on PixStor 5.

Different sites may configure a different `elasticsearch_url`, for example if one is PixStor 5 and another is PixStor 6.

Search UI

Once configured, search can be used via the REST API as described in [Search](#)

To use search via the Ngenea Hub UI, you must first enable the `searchui` feature flag, as described in [Feature Flags](#)

For example, using `ngclient`

```
ngclient features enable searchui
```

Cloud Functions

Cloud to Hub is a function intended to be running in the cloud. It runs in response to cloud storage events, triggering a job in Ngenea Hub to reflect the change onto another system (e.g. PixStor). This allows for keeping multiple systems in sync, using cloud storage as the source of truth.

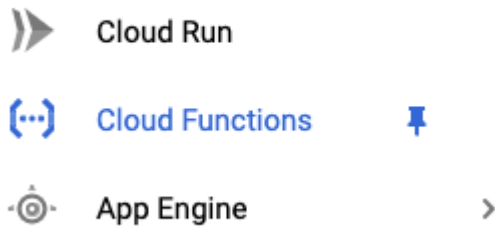
The code is designed to work with any of the supported platforms (see above), and any event (create, delete, ...) with only changes to the `config.json` file, as described below.

Currently supports:

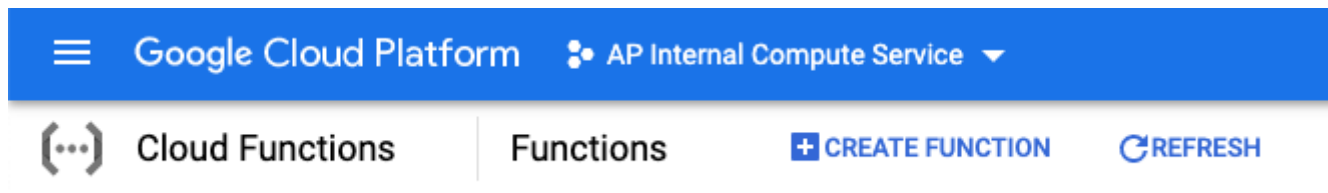
GCP Cloud Function

In the GCP console menu under Serverless, select **Cloud Functions**

SERVERLESS



Choose **Create Function**



In the first Configuration page:

1. Give the function a name
2. Select a region to run from. Usually best to be in the same region as the bucket we'll be using
3. Under trigger select **Cloud Storage** type
4. And **Finalise/Create** for the event type
5. Select the bucket to monitor

Then press **Save**

1 Configuration — 2 Code

Basics

Function name *
ngeneahub-function ?

Region
europe-west2 ▼ ?

Trigger

Cloud Storage

Trigger type
Cloud Storage ▼

Event type *
Finalise/Create ▼ ?

Bucket *
pixcloud-reports BROWSE

☐ Retry on failure ?

SAVE CANCEL

If you wish to create a new service account for this function, use the following `gcloud` command to create a new service account and assign it the role `storage.objectViewer`

```
PROJECT_ID='GCP-PROJECT-1'
SERVICE_ACCOUNT_ID='ngeneahub-function'
ROLE_NAME='roles/storage.objectViewer'

gcloud iam service-accounts create $SERVICE_ACCOUNT_ID \
  --description='A service account to give the {{ brand_name }} \
function read access to GCS buckets' \
  --display-name=$SERVICE_ACCOUNT_ID

gcloud projects add-iam-policy-binding $PROJECT_ID \
  --member="serviceAccount:"
```

```
$SERVICE_ACCOUNT_ID@$PROJECT_ID.iam.gserviceaccount.com" \
--role=$ROLE_NAME
```

Open up the **RUNTIME, BUILD AND CONNECTIONS SETTINGS** section

Under the **RUNTIME** tab at the bottom, select a **Runtime** service account that has the following permissions as a minimum (or the newly created service account from above):

- `storage.objectViewer`

Select **Next** to continue

Runtime, build, connections and security settings ^

RUNTIME

BUILD

CONNECTIONS

SECURITY

Memory allocated *

256 MB

▼

Timeout *

60

seconds ?

Runtime service account ?

Runtime service account

ngeneahub-function

▼

Auto-scaling ?

Minimum number of instances

0

Maximum number of instances

3000

If the Ngenea Hub doesn't have an external IP to connect to, you'll need a **VPC Connector** for the function to be able to access the Ngenea Hub private IP.

The creation of the **VPC Connector** is out of scope of these docs.

To select an existing **VPC Connector**, under the **RUNTIME, BUILD AND CONNECTIONS SETTINGS** section, select **Connections**.

From the VPC Connector drop down menu, select an existing connector and check the **Only route requests to private IPs through the VPC connector** radiobox.

Runtime, build, connections and security settings



RUNTIME

BUILD

CONNECTIONS

SECURITY

Ingress settings ?

- ☒ Allow all traffic
- ☐ Allow internal traffic only
Only traffic from VPC networks in the same project or the same VPC SC perimeter is allowed.
- ☐ Allow internal traffic and traffic from Cloud Load Balancing
Traffic from VPC networks in the same project, the same VPC SC perimeter or from Cloud Load Balancing is allowed.

Egress settings ?

By default, your function can send requests to the Internet, but not to resources in VPC networks. To send requests to resources in your VPC network, create or select a VPC connector already created in the same region as the function.

VPC Connector

ngeneahubtestfunctions

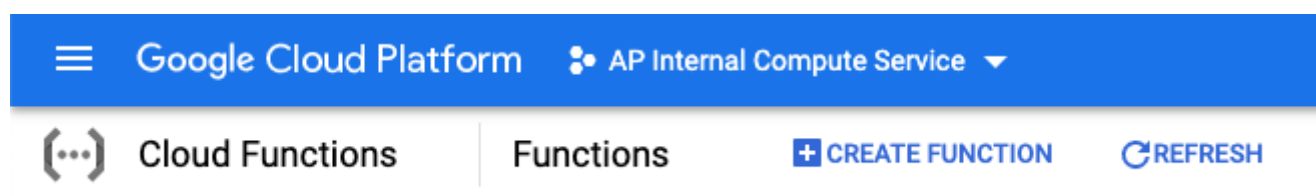


[Create a serverless VPC connector](#)

- ☒ Only route requests to private IPs through the VPC connector
- ☐ Route all traffic through the VPC connector

In the Code config section

1. Change the **Runtime** to **Python 3.9**
2. The **Entry Point** is **main**
3. Select **ZIP upload** in the **Source code**
4. Choose the GCP zip previously downloaded from the ../../download page
5. Select a **Stage bucket** for use while deploying. You can use the bucket we'll be monitoring
6. Select **Next** to build the **Cloud Function**

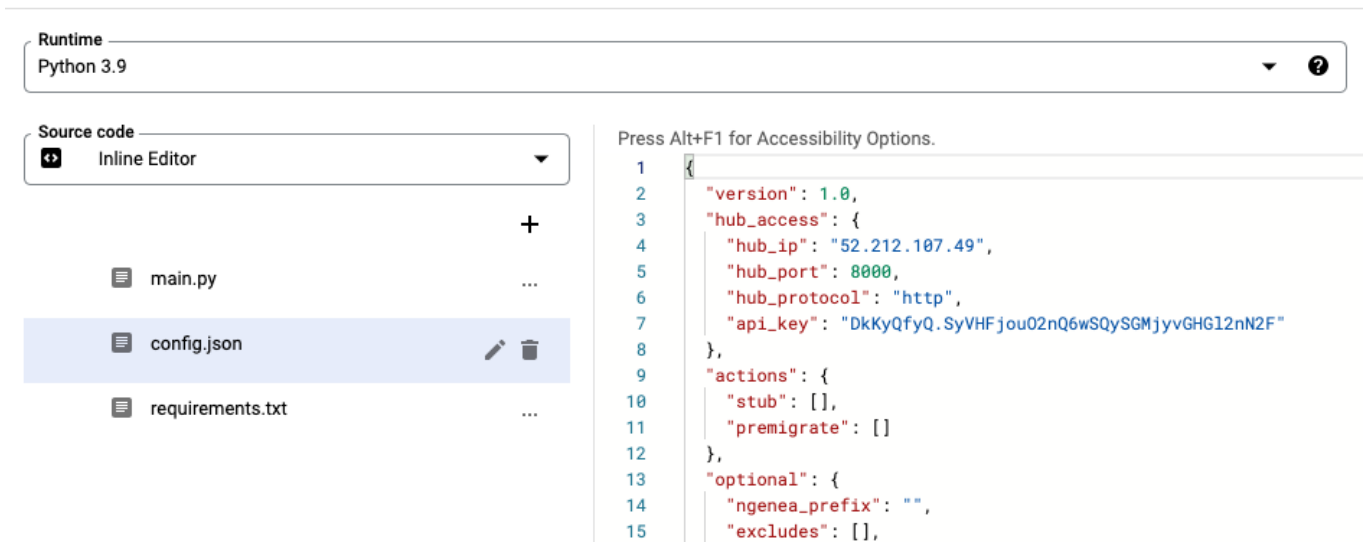


Once built you need to edit the default `config.json` file

Choose your new function and click **EDIT**

Select Next to get to the code edit section

Select the `config.json` file to edit



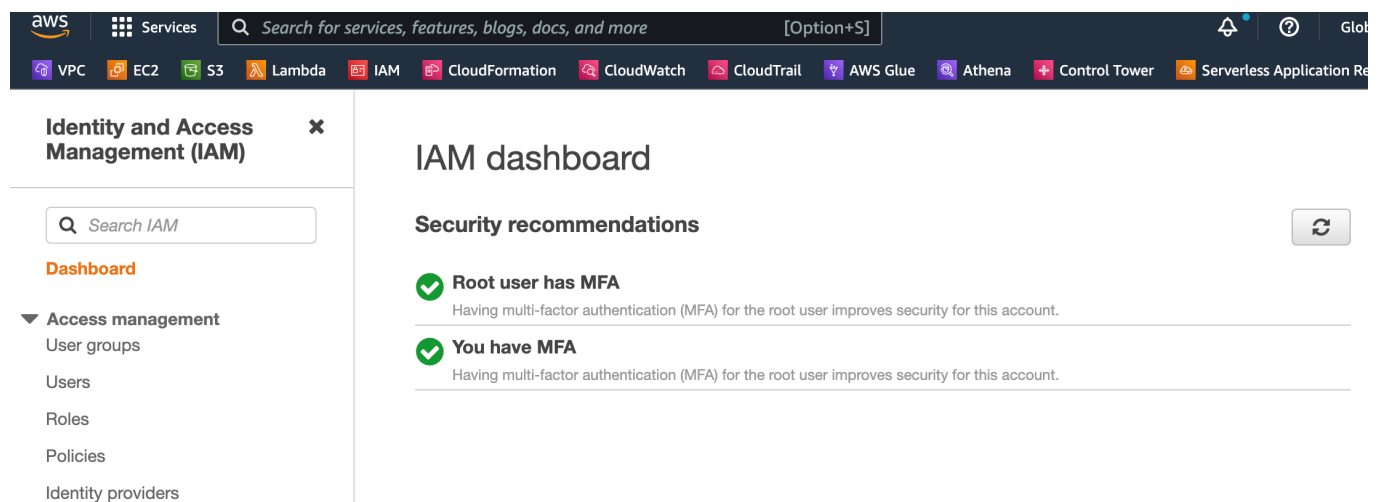
Edit the config file based on the docs from [Cloud Functions](#)

Select Deploy to save the changes. This can take 1-2 mins to update

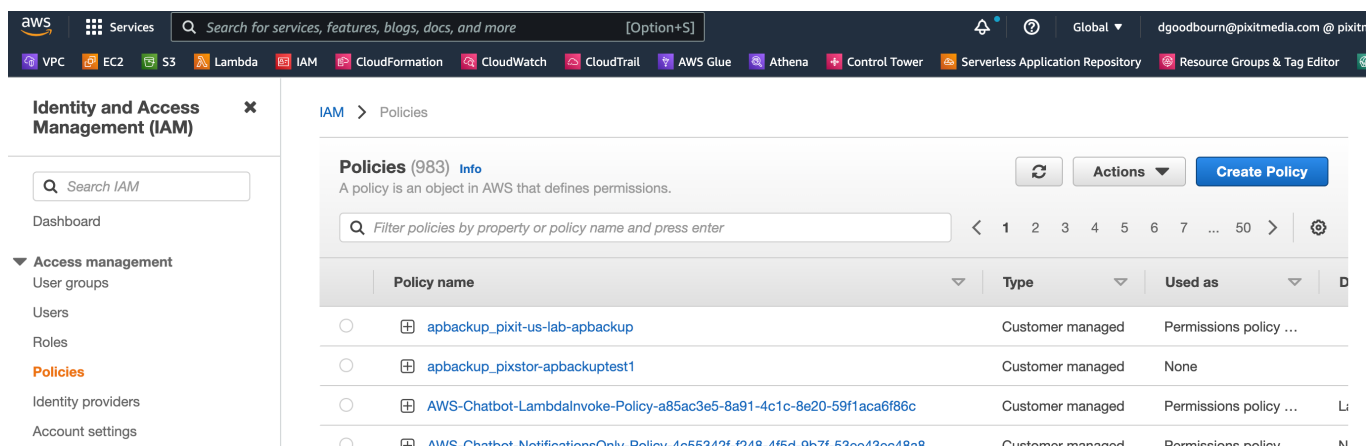
AWS Lambda Function

Create the IAM policy and role

In the AWS console go to the IAM service



Select the **Policies** then **Create Policy**



Move to the **JSON** tab and replace the text with the following:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "logs:CreateLogGroup",
      "Resource": "arn:aws:logs:*:<<AWS_ACCOUNT_ID>>:*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "logs:CreateLogStream",
        "logs:PutLogEvents"
      ],
      "Resource": [
        "arn:aws:logs:*:<<AWS_ACCOUNT_ID>>:log-group:/aws/lambda/<<LAMBDA_NAME>>:*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": "iam:GetUser",
      "Resource": "*"
    }
  ]
}
```

Replace **<<AWS_ACCOUNT_ID>>** with your account id. This must be the Id without the - in the name, i.e. 123456789012, not 1234-5678-9012

Replace **<<LAMBDA_NAME>>** with the name of your Lambda function

Select **Next**

If you require a tag, add it now, otherwise just select **Review**

Name the policy **ngeneahub_lambda_policy**

Then select **Create Policy**

Review policy

Name*

Use alphanumeric and '+,=, @, _' characters. Maximum 128 characters.

Description

Maximum 1000 characters. Use alphanumeric and '+,=, @, _' characters.

Summary

Service ▾	Access level	Resource	Request condition
Allow (2 of 316 services) Show remaining 314			
CloudWatch Logs	Limited: Write	Multiple	None
IAM	Limited: Read	All resources	None

Next create a new **Role**

Identity and Access Management (IAM) ✕

Dashboard
▼ Access management
User groups
Users
Roles
Policies
Identity providers
Account settings

[IAM](#) > [Roles](#)
Roles (92) [Info](#)
An IAM role is an identity you can create that has specific permissions with credentials that are valid for short durations. Roles can be assumed by entities that you trust.

☐ **Role name** ▾ **Trusted entities** **Last acti...** ▾

☐	aaa-del-role-566vn76r	AWS Service: lambda	-
☐	AmazonNimbleStudioAdminRole	AWS Service: identity.nimble	285 days ago
☐	AmazonNimbleStudioUserRole	AWS Service: identity.nimble	285 days ago

Select the **AWS service** for the trusted entity and a **Lambda** for the use case

Select trusted entity

Trusted entity type

☒ **AWS service**
Allow AWS services like EC2, Lambda, or others to perform actions in this account.

☐ **AWS account**
Allow entities in other AWS accounts belonging to you or a 3rd party to perform actions in this account.

☐ **Web identity**
Allows users federated by the specified external web identity provider to assume this role to perform actions in this account.

☐ **SAML 2.0 federation**
Allow users federated with SAML 2.0 from a corporate directory to perform actions in this account.

☐ **Custom trust policy**
Create a custom trust policy to enable others to perform actions in this account.

Use case

Allow an AWS service like EC2, Lambda, or others to perform actions in this account.

Common use cases

- ☐ **EC2**
Allows EC2 instances to call AWS services on your behalf.
- ☒ **Lambda**
Allows Lambda functions to call AWS services on your behalf.

Search for the policy we just created above and select it, to attach the policy to this role. Then select **Next**

IAM > Roles > Create role

Step 1
Select trusted entity

Step 2
Add permissions

Step 3
Name, review, and create

Add permissions

Permissions policies (Selected 1/791)
Choose one or more policies to attach to your new role.

1 match

Clear filters

☒	Policy name	Type	Description
☒	ngeneahub_lambda_policy	Custom...	

► **Set permissions boundary - optional**

Set a permissions boundary to control the maximum permissions this role can have. This is not a common setting, but you can use it to delegate permission management to others.

Cancel

Previous

Next

Name the role **ngeneahub_lambda_role** and select **Create Role**

Step 1
[Select trusted entity](#)

Step 2
[Add permissions](#)

Step 3
Name, review, and create

Name, review, and create

Role details

Role name

Enter a meaningful name to identify this role.

Maximum 128 characters. Use alphanumeric and '+=, @-_' characters.

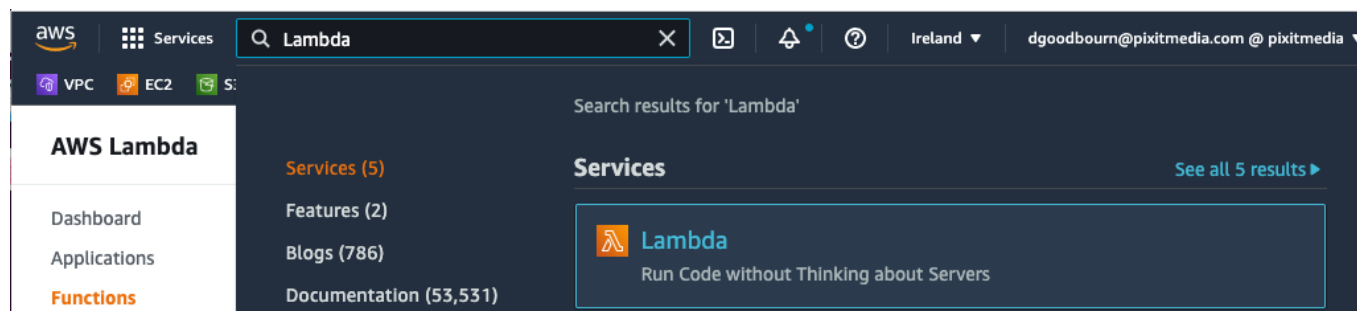
Description

Add a short explanation for this policy.

Maximum 1000 characters. Use alphanumeric and '+=, @-_' characters.

Create the Lambda function

In the AWS console go to the **Lambda** service



Click the **Create function** button

- Create a new function with the **Author from scratch** option selected
- Name the function
- And select **Python 3.7** from the **Runtime** list

Create function [Info](#)

AWS Serverless Application Repository applications have moved to [Create application](#).

☒ **Author from scratch**
Start with a simple Hello World example.

☐ **Use a blueprint**
Build a Lambda application from sample code and configuration presets for common use cases.

☐ **Container image**
Select a container image to deploy for your function.

Basic information

Function name

Enter a name that describes the purpose of your function.

ngeneahub_connection

Use only letters, numbers, hyphens, or underscores with no spaces.

Runtime [Info](#)

Choose the language to use to write your function. Note that the console code editor supports only Node.js, Python, and Ruby.

Python 3.7

Under **Change default execution role**, select **Use an existing role** and find the role we created above

Permissions [Info](#)

By default, Lambda will create an execution role with permissions to upload logs to Amazon CloudWatch Logs. You can customize this default role later when adding triggers.

▼ Change default execution role

Execution role

Choose a role that defines the permissions of your function. To create a custom role, go to the [IAM console](#).

- ☐ Create a new role with basic Lambda permissions
- ☒ Use an existing role
- ☐ Create a new role from AWS policy templates

Existing role

Choose an existing role that you've created to be used with this Lambda function. The role must have permission to upload logs to Amazon CloudWatch Logs.

ngeneahub_lambda_role

[View the ngeneahub_lambda_role role on the IAM console.](#)

If your Ngenea Hub doesn't have an external IP to connect to, you'll need to select the **Enable Network** option under the **Advanced settings**

- Select the **VPC** the Ngenea Hub is located in
- Choose the **Subnet** it's in
- And select a **Security group** that gives access to the Ngenea Hub over port 8000

▼ Advanced settings

☐ Enable Code signing [Info](#)

Use code signing configurations to ensure that the code has been signed by an approved source and has not been altered since signing.

☒ Enable Network [Info](#)

To provide network access for your Lambda function, specify a virtual private cloud (VPC), VPC subnets, and VPC security groups. VPC configuration is optional unless your user permissions require you to configure a VPC.

VPC

Choose a VPC for your function to access.

vpc-0bd7136b621f84a87 (10.10.0.0/24) ▼



Subnets

Select the VPC subnets for Lambda to use to set up your VPC configuration.

Choose subnets ▼



subnet-01c2a9d8719207552 (10.10.0.0/24) eu-west-1c ✕

Name: pixstor-demo

We recommend that you choose at least 2 subnets for Lambda to run your functions in high availability mode.

Security groups

Choose the VPC security groups for Lambda to use to set up your VPC configuration. The table below shows the inbound and outbound rules for the security groups that you choose.

Choose security groups ▼



sg-032d65f704d480aa5 (pixstor-demo-002-SG-FRONTEND) ✕

Pixcloud Security Group

aws:cloudformation:logical-id: pixCloudSG aws:cloudformation:stack-name: pixstor-demo-002

aws:cloudformation:stack-id: arn:aws:cloudformation:eu-west-1:918992847240:stack/pixstor-demo-002/af457060-d040-11eb-9b92-06df0d66bfc3

Name: pixstor-demo-002-SG-FRONTEND

A policy will also have to be created to permit the function to traverse the VPC's networking, this can be done by adjusting the previously created IAM policy:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "logs:CreateLogGroup",
      "Resource": "arn:aws:logs:*:<<AWS_ACCOUNT_ID>>:*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "logs:CreateLogStream",
        "logs:PutLogEvents"
      ],
      "Resource": [
        "arn:aws:logs:*:<<AWS_ACCOUNT_ID>>:log-group:/aws/lambda/<<LAMBDA_NAME>>:*"
      ]
    },
    {
      "Effect": "Allow",
```

```

    "Action": "iam:GetUser",
    "Resource": "*"
  },
  {
    "Effect": "Allow",
    "Action": [
      "iam:GetUser",
      "ec2:DescribeNetworkInterfaces",
      "ec2:CreateNetworkInterface",
      "ec2:DeleteNetworkInterface"
    ],
    "Resource": "*"
  }
]
}

```

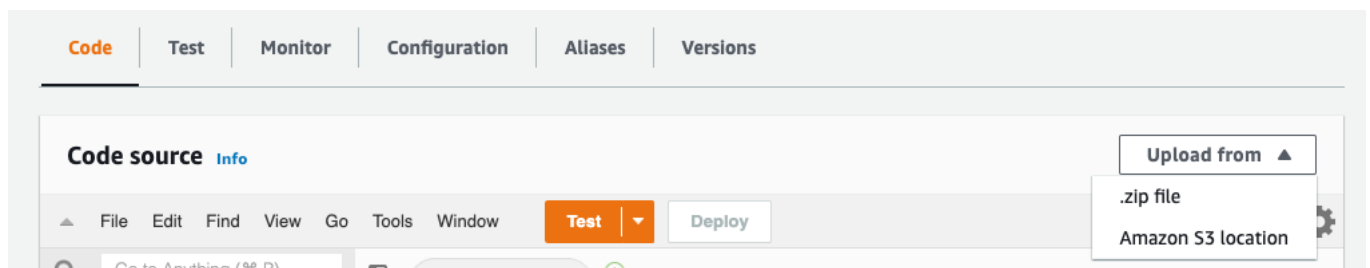
By adding this additional section, the function can traverse correctly.

With this configured, we can now create our function by selecting **Create function**

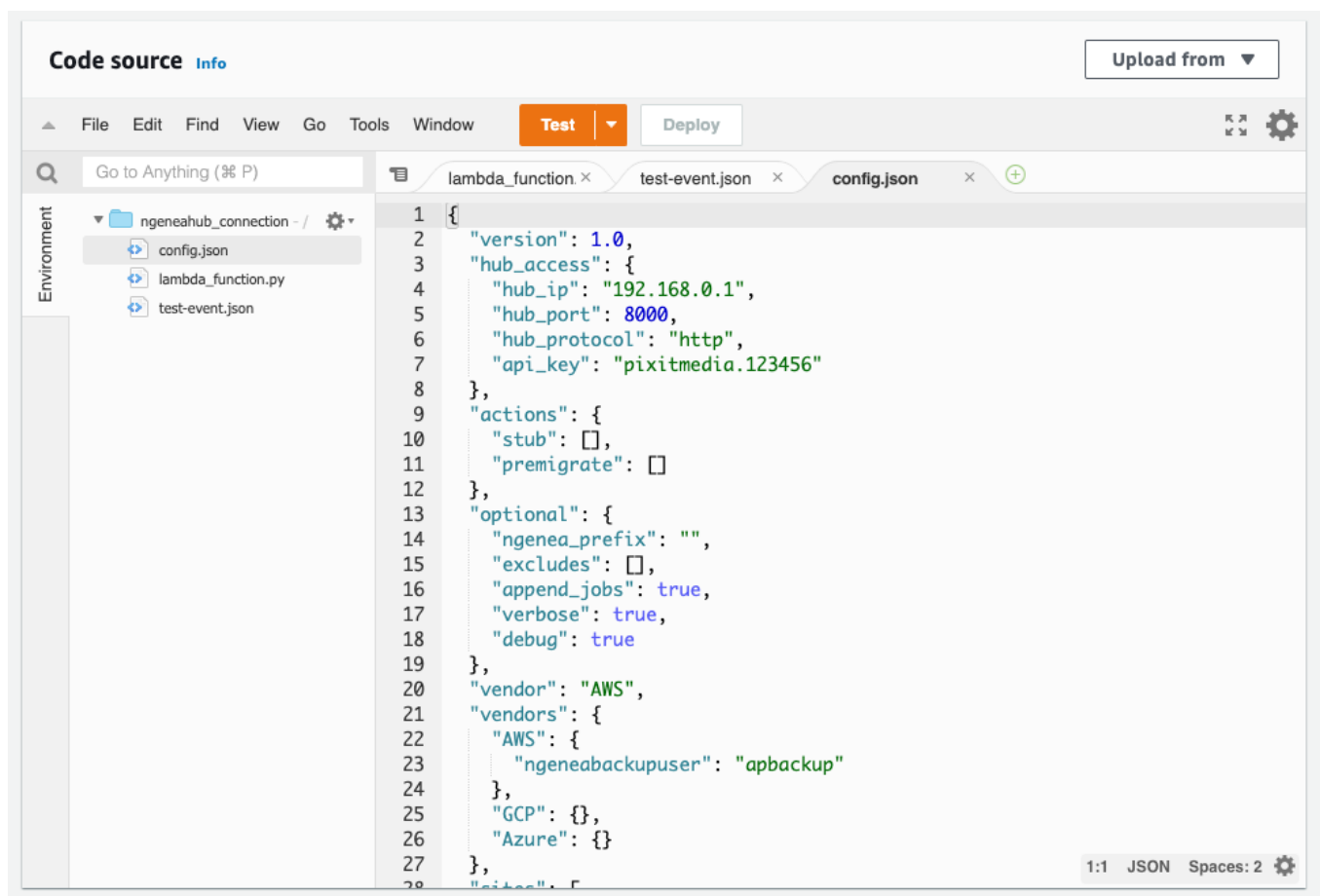
Once the function is created - This can take a few minutes if you've chosen to create the **Network** connection.

Under the **Code** tab, you'll need to upload the zip file containing all the code

Choose the AWS zip previously downloaded from the ../../download page

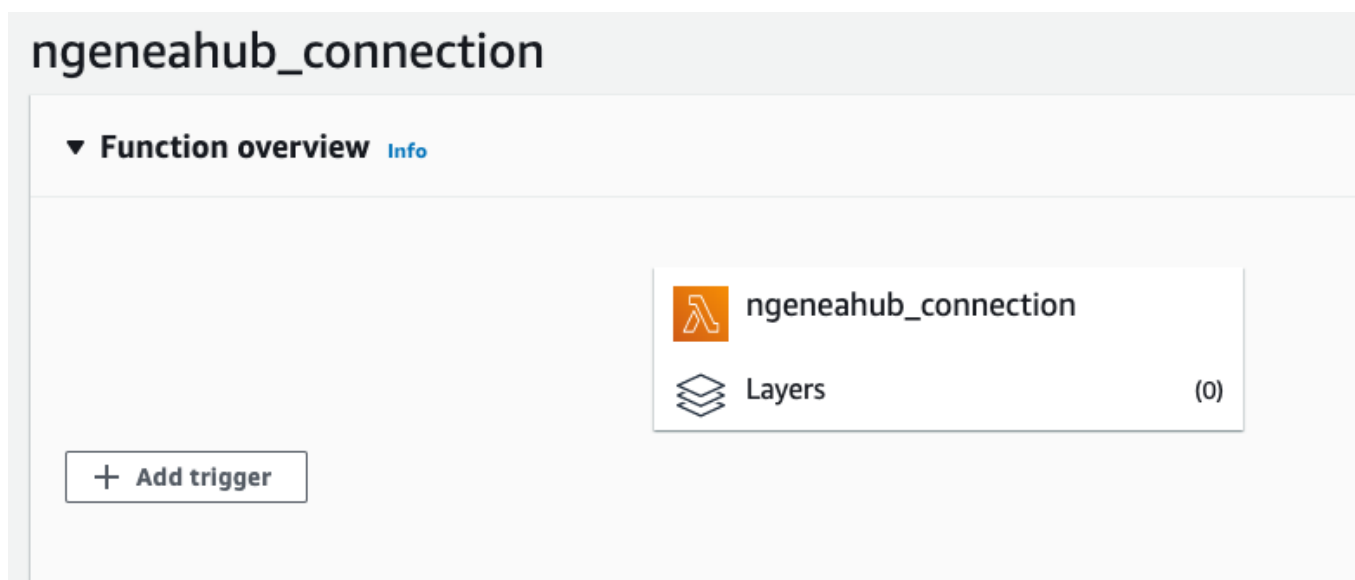


Once the code is uploaded, you need to edit the `config.json` file with all the relevant details based on the docs from [Cloud Functions](#)



Once you've updated the `config.json` file, save it by selecting the **Deploy** button

Now the Lambda function is created, we need to assign it to a bucket event. Select **Add trigger** under the **Function overview**



Choose:

- **S3** as the **Trigger**
- Select your bucket
- **All object create events** for the **Event type**
- Check the **Recursive Invocation** checkbox warning

Then just press **Add**

Add trigger

Trigger configuration



S3
aws storage



Bucket

Please select the S3 bucket that serves as the event source. The bucket must be in the same region as the function.

dgoodbourn-playground2



Event type

Select the events that you want to have trigger the Lambda function. You can optionally set up a prefix or suffix for an event. However, for each bucket, individual events cannot have multiple configurations with overlapping prefixes or suffixes that could match the same object key.

All object create events



Prefix - optional

Enter a single optional prefix to limit the notifications to objects with keys that start with matching characters.

e.g. images/

Suffix - optional

Enter a single optional suffix to limit the notifications to objects with keys that end with matching characters.

e.g. .jpg

Lambda will add the necessary permissions for Amazon S3 to invoke your Lambda function from this trigger. [Learn more](#) about the Lambda permissions model.



Recursive invocation

If your function writes objects to an S3 bucket, ensure that you are using different S3 buckets for input and output. Writing to the same bucket increases the risk of creating a recursive invocation, which can result in increased Lambda usage and increased costs. [Learn more](#)

- ☒ I acknowledge that using the same S3 bucket for both input and output is not recommended and that this configuration can cause recursive invocations, increased Lambda usage, and increased costs.

Cancel

Add

You now have a trigger assigned to the Lambda function

To monitor the Lambda function, under the **Monitor** tab you can see all different types of monitoring

Using the **View logs in CloudWatch** is the best place to see any out from the Lambda function

Code
Test
Monitor
Configuration
Aliases
Versions

Metrics
Logs
Traces

View logs in CloudWatch
View X-Ray traces in ServiceLens
View Lambda Insights
View profiles in CodeGuru

CloudWatch Logs Insights

Lambda logs all requests handled by your function and automatically stores logs generated by your code through Amazon CloudWatch Logs. To validate your code, instrument it with custom logging statements. The following tables list the most recent and most expensive function invocations across all function activity. To view logs for a specific function version or alias, visit the **Monitor** section at that level.

1h
3h
12h
1d
3d
1w
Custom

Add to dashboard

Coming soon:

- Azure

Configuration

The configuration file is in JSON format.

version

Indicates the config format version. Currently, only `1.0` is supported.

hub_access

Defines setting for interacting with Ngenea Hub

- **hub_ip**: IP addresss of the Ngenea Hub REST API to use
- **hub_port**: port for the Ngenea Hub REST API, typically `8000`
- **hub_protocol**: either `http` or `https`
- **api_key**: API 'client key' to authenticate Ngenea Hub with
- **workflow**: name of the workflow to submit for the event file, e.g. `reverse_stub`
- **workflow_flags**: mapping of settings to pass to the workflow, e.g. `{"hydrate": true}`

For new files, one would typically use **workflow** `reverse_stub` or `recall`, which ship with Ngenea Hub by default. There is no default 'delete' workflow in Ngenea Hub, so one must be created - e.g.

```
{
  "name": "delete_file",
  "label": "delete_file",
  "icon_classes": [
    "fa fa-cloud fa-stack-2x text-success",
    "fa fa-angle-up fa-stack-2x text-light"
  ],
  "discovery": null,
  "enabled": true,
  "visible": true,
  "fields": [],
  "filter_rules": [
    {
      "type": "all",
      "state": "all",

```

```

    "action": [
      {
        "name": "dynamo.tasks.delete_paths_from_gpfs",
        "recursive": false
      }
    ],
    "description": "Delete the file at a given path"
  }
]
}

```

sites

List of sites to reflect the events to.

- **site**: name of the site, as registered in Ngenea Hub
- **default**: (optional) default mode for 'recall' type workflows. One of `stub`, `premigrate`
- **skip_from_ngenea**: (optional) if True then any file that was created via ngenea will be skipped for this site

default is used if the event path doesn't match any **action** (see below), and if `hydrate` isn't explicitly set in **workflow_flags** (see above)

skip_from_ngenea is useful when e.g. Ngenea Hub is being used to sync files between sites. Files which are being transmitted via the cloud bucket don't need to be automatically recalled onto either site. On the other hand, files uploaded directly to the cloud still need to be recalled.

For GCP, we may not be able to determine the source of a delete event. In that case, deletes will always be reflected to all sites.

actions

Mapping of actions -- `stub` or `premigrate` -- to path prefixes.

This is used to determine whether a file should be hydrated by a 'recall' type workflow. If a path matches multiple actions, the longest match wins. For example, using

```

{
  "stub": ["data"],
  "premigrate": ["data/cats"]
}

```

the path `data/cats/cat-01.jpg` would be premigrated, while `data/cats-02.jpg` would be stubbed.

If not specified, the site-specific **default** (see above) will be used, unless missing or unless `hydrate` is explicitly set in **workflow_flags**. If none of these settings are configured, the default behaviour is to stub.

optional

Optional settings

- **ngenea_prefix**: prefix which maps a cloud path to a local path, e.g. with prefix `/mfs1` the cloud path `data/cats-01.jpg` is mapped to `/mfs1/data/cats-01.jpg`. Default: `''`
- **excludes**: list of strings used to exclude paths. The strings are treated as sub-strings which can match anywhere in the (cloud) path string. Default: `[]`
- **append_jobs**: if `true`, tasks will be grouped under the same job id (per hour). If `false`, each task will get its own job. Default: `false`
- **verbose**: set logging output to `info` level. Default: `false`
- **debug**: set logging output to `debug` level. Takes precedence over **verbose**. Default: `false`

vendor

The vendor that the function is being run on.

Currently supported values: `AWS`, `GCP`

vendors

Vendor specific settings. Currently only used by `AWS`

AWS

- **ngeneabackupuser**: name of the user ngenea uses for AWS. Used to identify whether a file came from ngenea for **skip_from_ngenea**

Complete Example

```
{
  "version": 1.0,
  "hub_access": {
    "hub_ip": "192.168.0.1",
    "hub_port": 8000,
    "hub_protocol": "http",
    "api_key": "pixitmedia.123456",
    "workflow": "reverse_stub",
    "workflow_flags": {
      "hydrate": false,
      "overwrite": true
    }
  },
  "sites": [
    {
      "site": "uk",
      "default": "stub"
    }
  ],
  "actions": {
    "stub": [],
    "premigrate": []
  },
  "optional": {
```

```

    "ngeneia_prefix": "",
    "excludes": [],
    "append_jobs": true,
    "verbose": true,
    "debug": true
  },
  "vendor": "AWS",
  "vendors": {
    "AWS": {
      "ngeneiabackupuser": ""
    },
    "GCP": {},
    "Azure": {}
  }
}

```

Ngeneia Worker Plugins

Warning: This feature is currently in alpha

When constructing workflows to run on Ngeneia Hub, there may not be a task that covers the behaviour required for a given workflow. To alleviate this, there is now support for custom plugin tasks to be added to any instance of Ngeneia Worker that can cover that missing behaviour.

These are defined as custom tasks for the celery <https://docs.celeryq.dev/en/stable/> task distribution framework.

Creating your plugin

Warning: This feature is currently in alpha.

A plugin for the hub is a python module that contains one or multiple celery tasks that can be discovered by the Ngeneia Worker, these tasks can then be used in workflows in Ngeneia Hub.

Example plugin creation

Using the Ngeneia Hub CLI a template plugin can be created in the plugin directory at `/var/lib/ngeneia-worker/plugins`. All plugins will persist in this location and will need to be in this directory to be correctly installed.

For the following example, we will be creating a plugin named `echo_args` that will return the expected key word arguments provided to the task itself.

To begin, create a plugin template for the example `echo_args` using:

```
ngeneia-worker plugins create echo_args
```

This will create the template plugin project at `/var/lib/ngenea-worker/plugins/echo_args`. To implement custom logic, navigate to `/var/lib/ngenea-worker/plugins/echo_args` in which contains the python module file `echo_args/echo_args.py`. Within this file there is the basic example task `example_task`:

```
from arcapix.dynamo.server.status import FileActionStatus
from arcapix.dynamo.server.queue import get_current_task_queue_name

@shared_task(bind=True, name="dynamo.custom.example_task")
def example_task(self, *args, paths: List = None, jobid: int = None,
**kwargs):
    print(paths)
    print(jobid)
    status = FileActionStatus(self.name, paths, jobid,
get_current_task_queue_name())
    for path in paths:
        status.add("processed", path["path"])
    return status.dump()
```

Warning: The task return must adhere to the format exemplified above otherwise the job will result in an indefinite `Pending` state in the UI/API and the Ngenea Hub logs will present a traceback in the task call back. Refer to section: `Custom Tasks/Return Payload` for the specific return format of task results.

For more information on the supported file states, please refer to the section `supported file states` in the `Custom Tasks` page.

The logic within this task can be replaced along with the name of the function to our `echo` task:

Note: The name provided to `shared_task` will be the task name that is used to call the task within a workflow.

```
from arcapix.dynamo.server.status import FileActionStatus
from arcapix.dynamo.server.queue import get_current_task_queue_name

@shared_task(bind=True, name="dynamo.custom.echo_args")
def echo_args(self, *args, paths: List = None, jobid: int = None,
**kwargs):
    status = FileActionStatus(self.name, paths, jobid,
get_current_task_queue_name())
    for path in paths:
        status.add("processed", path["path"])
    return status.dump()
```

With the new task in place, the entry point for the plugin will have to be adjusted in `setup.py`.

Note: Ensure that all custom task entry points are defined under `worker_plugin`

Initial `setup.py` entry points:

```
...
    "worker_plugin": [
        "echo_args=echo_args.echo_args:example_task",
    ]
...
```

In this case, the changes are as follows:

```
...
    "worker_plugin": [
        "echo_args=echo_args.echo_args:echo_args",
    ]
...
```

External Dependencies

If the logic in the plugin requires an external dependency, it can be installed into the worker using the `setup.py` within the template plugin generated via `ngenea-worker plugins create`. To add additional dependencies, they can be added to the `install_requires` section:

Note: Ensure that the celery package and version is un-edited within the plugin requirements, otherwise it may have unexpected effects on the functionality of Ngenea Worker.

```
install_requires=[
    "celery<=5.4",
    "example-module==0.1.0"
]
```

Installing custom plugins

To install the newly created plugin refer to the [plugin installation page](#).

Managing plugins

Warning: This feature is currently in alpha.

Enabling your plugin

To enable worker plugin behaviour, the `enable_plugins` setting needs to be set to `true` in the `/etc/ngenea/ngenea-worker.conf` file

```
enable_plugins=true
```

By default, this setting will be `false`

After plugins have been created, they can be managed using the following scripts.

Installing your plugin

All the plugins that have been created within `/var/lib/ngenea-/worker/plugins` can be installed through running:

```
ngenea-worker plugins install
```

This will install all of the plugin packages within the plugins directory. These packages will not update within the Ngenea Worker unless it has had a version number increase in the `setup.py`.

Single plugins can also be installed by providing the name of the package explicitly:

```
ngenea-worker plugins install PACKAGE_NAME
```

After installing the desired plugins, the Ngenea Worker service will need to be restarted:

```
systemctl restart ngenea-worker
```

Uninstalling your plugin

Uninstalling any of the plugins can be done through the uninstall script:

Warning: This can remove any modules within the environment that Ngenea Worker runs within, use caution when uninstalling any modules that aren't directly within `/var/lib/ngenea-worker/plugins` as it could cause Ngenea Worker to not function correctly.

```
ngenea-worker plugins uninstall <PACKAGE_NAME>
```

To uninstall all of the modules in the plugin directory it can be done through:

```
pushd /var/lib/ngenea-worker/plugins/  
ngenea-worker plugins uninstall * -y  
popd
```

Custom Tasks

Warning: This feature is currently in alpha

In addition to the [predefined tasks](#), Ngenea Worker plugins allow custom tasks to be defined.

Once [created](#), custom tasks can be included in [Custom Workflows](#).

Custom tasks must accept the same standard arguments, and return a payload in the same format, as predefined tasks do.

The Structure of a Task

Arguments

Tasks must accept the following keyword arguments:

name	description	example	default
jobid	ID of the job the task is associated with.	100	None
paths	List of paths (may be files, folders or symlinks) to be processed. Each path is given as a dict with a "path" key, and can also have other keys such as "size" and "message".	[{"path": "/mmfs1/data/my-fileset/file1", "size": 264321}, {"path": "/mmfs1/data/my-fileset/file2", "size": 15}]	None

To pass specific keyword arguments to a task, include these in the workflow definition as hardcoded values or [Runtime fields](#).

In addition, tasks should accept arbitrary other keyword arguments via a `**kwargs` parameter, but do not need to process them.

Here is an example function signature, from the definition of the predefined task `dynamo.tasks.migrate`:

```
def migrate(self, *args, jobid=None, paths=None, skip_modified_during_migrate=False, **kwargs)
```

Return Payload

Ngeneia Hub expects task results to be returned in a specific format.

There is a class `FileActionStatus` that the worker code uses as a convenience for constructing the return payload. This may be used as an alternative to constructing the return payload from scratch.

Using the FileActionStatus class

The class is imported like this:

```
from arcapix.dynamo.server.status import FileActionStatus
```

A status object is instantiated like this:

```
status = FileActionStatus(taskname, input_paths, jobid, queue_name)
```

- `taskname` is the name of the current task
- `input_paths` is the list of path objects as passed to the task in the `paths` parameter

- `jobid` is the job's numerical id. This is provided as an input to the task.
- `queue_name` is the queue the task is currently running on. This can be retrieved with `get_current_task_queue_name()` from `arcapix.dynamo.server.queue`

For each path operated on by the task, call the `.add(key, path)` method on the status object, where `key` is the state of the file:

```
status.add("processed", "/mmfs1/data/my-fileset/file1")
```

The status object keeps track of files as they are added. At the completion of the task, return the results like this:

```
return status.dump()
```

Constructing a return payload from scratch

Tasks return a dict containing at least these 3 keys:

```
{
  "jobid": <job_id>,
  "paths": <list of path objects that were processed or skipped>,
  "status": {
    "task": <task_name>,
    "input_paths": <list of path objects input to the task>,
    "input_total": <number of paths input to the task>,
    "summary": <dict giving number of files keyed by state>,
    "details": <dict giving list of file objects keyed by state>,
    "started": <timezone-aware datetime for when the task
started>
  }
}
```

The `paths` key lists the paths to be passed to the next task in the workflow.

Here is an example payload:

```
{
  "jobid": 100,
  "paths": [{"path": "/mmfs1/data/my-fileset/file1"}],
  "status": {
    "task": "my-plugin",
    "input_paths": [{"path": "/mmfs1/data/my-fileset/file1"},
{"path": "/mmfs1/data/my-fileset/file2"}],
    "input_total": 2,
    "summary": {
      "failures": 1,
      "processed": 1,
      "skipped": 0
    },
    "details": {
      "failures": [{"path": "/mmfs1/data/my-fileset/file2",
"message": ["Reason xyz for the failure"]}],
      "processed": [{"path": "/mmfs1/data/my-fileset/file1"}],
```

```

        "skipped": []
    },
    "started": "2023-04-25T14:56:31.253043",
}
}

```

Supported file states

These are the supported file states for tasks that perform operations on files:

- "processed": file was successfully processed
- "skipped": file was not processed because they were already in the desired state
- "aborted": file was not processed and should not be processed by subsequent tasks in the workflow
- "failures": file could not be processed because of an error
- "inprogress": file is still being processed (this state is rarely used)

Generally, the files that should be passed to the next task are those that were processed or skipped.

Streaming task results

This applies to delete and move handlers in a snapdiff workflow.

Rather than returning, results for delete and move handler in snapdiff workflows must be streamed back to Ngenea Hub

In this case Ngenea Hub will pass keyword argument `stream_results=True` to the task to indicate that it expects streamed results

Important: If your task will be used as a move or delete handler in a snapdiff workflow, it must accept `stream_results` either as an explicit parameter or via `**kwargs`

A `StreamTaskResults` class is provided to facilitate this.

```

from arcapix.dynamo.server.streamstatus import StreamTaskResults

```

The class behaves the same as a `FileActionStatus`

```

if stream_results:
    status = StreamTaskResults(taskname, input_paths, jobid,
                               queue_name, taskid)
else:
    status = FileActionStatus(taskname, input_paths, jobid,
                              queue_name)

```

The parameters are the same as `FileActionStatus`, described above, with the addition on `taskid`. The task must be defined with `bind=True`. The `taskid` can then be retrieved with `self.request.id`

For each path operated on by the task, call the `.add(key, path)` method on the status object, where `key` is the state of the file:

```
status.add("processed", "/mmfs1/data/my-fileset/file1")
```

The status object keeps track of files as they are added, and in the background sends them to Ngenea Hub in batches.

At the completion of the task, return the results like this:

```
return status.dump()
```

Calling `status.dump()` also flushes any outstanding files to Ngenea Hub.

Support for Streaming Paths from API

This applies to delete and move handlers in a snapdiff workflow.

To fetch the streaming paths, the function `expand_paths` from the module `arcapix.dynamo.server.utils.inputs` has to be used like below.

```
from arcapix.dynamo.server.utils.inputs import expand_paths
```

Warning: `expand_paths` provides a generator function fetching pages of paths on demand. Developers must ensure that memory starvation of Hub is negated by following the principles of generator functions thereby avoiding storing large number of values in memory concurrently.

Queues

There are various functions available to get information about the celery queue a custom task is running on

Note: The way queues are handled, and the way celery queue names are formatted, has changed in Ngenea Hub 2.4.0

For more information on Ngenea Hub queues see [Queues](#)

```
from arcapix.dynamo.server.queue import ...
```

- `get_current_task_queue_name()` - returns the name of the celery queue the current task is running on
- `get_current_site()` - returns the name of the site the current task is running on
- `get_formatted_queue_name(site, queue, function)` - returns the celery queue name for a given `site`, `queue`, and `function`
- `ParsedQueueName` - a class representing a celery queue name, parsed into `site`, `queue`, and `function`

```

parsed_queue = ParsedQueueName.current_task_queue()

print(parsed_queue.site) # london
print(parsed_queue.queue) # highpriority
print(parsed_queue.function) # custom

print(str(parsed_queue)) # london.highpriority#custom

```

Important: The format of celery queue names is subject to change. You should always use the above functions, rather than manually parsing and formatting queue names.

Example Plugins

Warning: This feature is currently in alpha

Here are a couple examples of functionality that can be added to the hub, these can be dropped into any template project.

Email notification task

The intent of this plugin is to email a list of staff members at the end of a workflow to ensure that they are informed of its completion, it has extra key word arguments that allow the editing of the subject:

```

import sys

from email.mime.multipart import MIMEMultipart
from email.mime.text import MIMEText
from email.utils import formatdate
from smtplib import SMTP
from typing import List

from celery import shared_task
from arcapix.dynamo.server.status import FileActionStatus
from arcapix.dynamo.server.queue import get_current_task_queue_name

def send_email(email_to: List[str], email_from: str, subject: str,
message: str, server: str = "localhost"):
    """
    Sends an email through an already configured SMTP server

    :param email_to: List of email recipients
    :param email_from: Address that email derives from
    :param subject: Email subject
    :param message: Message to send
    :param server: Target SMTP server address
    :return: None
    """
    smtp_message = MIMEMultipart()

```

```

smtp_message['From'] = email_from
smtp_message['To'] = ', '.join(email_to)
smtp_message['Date'] = formatdate(localtime=True)
smtp_message['Subject'] = subject

smtp_message.attach(MIMEText(message))

try:
    smtp = SMTP(server)
    smtp.sendmail(email_from, email_to, smtp_message.as_string())
    smtp.close()
except OSError as error:
    # All SMTP errors are derived from OSError and catching all
    # SMTP errors from the base exception is not allowed
    print('Error sending notification email: %s', error)

@shared_task(bind=True, name="dynamo.custom.email_staff")
def email_staff(
    self,
    *args,
    paths: List = None,
    jobid: int = None,
    staff_members: List = None,
    message: str = None,
    subject: str = None,
    server: str = None,
    from_address: str = None,
    **kwargs
):
    if not message:
        message = f"Job
{jobid} has completed successfully processing {len(paths)} paths"

    if not from_address:
        from_address = "ngenea-worker@pixitmedia.com"

    if not subject:
        subject = "Ngenea Worker Job completed successfully"

    if not server:
        server = "localhost"

    status = FileActionStatus(self.name, paths, jobid,
get_current_task_queue_name())

    send_email(email_to=staff_members, email_from=from_address,
subject=subject, message=message, server=server)

    return status.dump()

if __name__ == "__main__":

```

```
sys.exit(0) # pragma: no cover
```

The `setup.py` will need editing to make the entry point the name of the function `email_staff` in the module that had been created for this example plugin.

To make use of this new task in the hub, here is an example workflow that will email staff members after all the data has been migrated with a custom subject and receiving email address:

```
{
  "name": "migrate_notif",
  "label": "Migrate with notif",
  "icon_classes": [
    "fa fa-cloud fa-stack-2x text-primary",
    "fa fa-refresh fa-stack-1x text-light"
  ],
  "discovery": null,
  "enabled": true,
  "visible": true,
  "fields": [],
  "filter_rules": [
    {
      "type": "all",
      "state": "all",
      "action": [
        {
          "name": "dynamo.tasks.migrate"
        },
        {
          "name": "dynamo.custom.email_staff",
          "staff_members": [
            "johnsmith@organisation.com",
            "admin@organisation.com"
          ],
          "subject": "Project number 7 job complete",
          "from_address": "notifications@organisation.com"
        }
      ]
    }
  ]
}
```

Running additional script

After running a set of tasks, you may need to execute a script on the host machine of the Ngenea Worker instance. The following plugin allows the execution of an arbitrary script located at `/opt/cloud_script.sh`:

Note: Any script run through this plugin will be run with root access

```

import sys

from subprocess import run as run_script
from subprocess import TimeoutExpired, SubprocessError, PIPE, STDOUT

from typing import List

from celery import shared_task

from arcapix.dynamo.server.status import FileActionStatus
from arcapix.dynamo.server.queue import get_current_task_queue_name

DEFAULT_SCRIPT = "/opt/cloud_script.sh"

@shared_task(bind=True, name="dynamo.custom.run_cloud_script")
def run_cloud_script(
    self,
    *args,
    paths: List = None,
    jobid: int = None,
    script_location: str = None,
    timeout: int = 600,
    additional_args: List = None,
    use_paths: bool = False,
    **kwargs
):
    status = FileActionStatus(self.name, paths, jobid,
get_current_task_queue_name())

    try:
        args = [script_location if script_location else
DEFAULT_SCRIPT]
        if additional_args:
            args = args + additional_args

        if use_paths:
            # Appends the paths to the arguments
            args = args + [path["path"] for path in paths]

        result = run_script(
            args,
            stdout=PIPE,
            stderr=STDOUT,
            check=True,
            timeout=timeout,
        )

        status.add_log(str(result.stdout))

        if result.returncode == 0:
            for path in paths:
                status.add("processed", path["path"])

```

```

except TimeoutExpired:
    status.add_log("Called script timed out")
except SubprocessError as sp_err:
    status.add_log(str(sp_err))

return status.dump()

if __name__ == "__main__":
    sys.exit(0) # pragma: no cover

```

The `setup.py` will need editing to make the entry point the name of the function `run_cloud_script` in the module that had been created for this example plugin.

To make use of this new task in the hub, here is an example workflow that will run the default cloud script after all of the data has been migrated:

```

{
  "name": "migrate_cloud_script",
  "label": "Migrate with notif",
  "icon_classes": [
    "fa fa-cloud fa-stack-2x text-primary",
    "fa fa-refresh fa-stack-1x text-light"
  ],
  "discovery": null,
  "enabled": true,
  "visible": true,
  "fields": [],
  "filter_rules": [
    {
      "type": "all",
      "state": "all",
      "action": [
        {
          "name": "dynamo.tasks.migrate"
        },
        {
          "name": "dynamo.custom.run_cloud_script"
        }
      ]
    }
  ]
}

```

SubDAG Helpers

Warning: This feature is currently in alpha

Typically, workflows have a fixed definition. However, certain use cases require modifying the workflow at runtime. For instance, a task might need to dynamically

batch paths for child tasks to ensure load balancing, or submit different child tasks based on the results of the current task.

Sub-DAGs enable the dynamic definition of workflows. A sub-DAG can consist of a single task or a chain/graph of multiple tasks. Once submitted, the sub-DAG is automatically integrated into the parent job for reporting and job control.

The worker repository provides a helper function called `task_template` which allows for defining a single task DAG graph for use with dynamic DAGs.

The function is defined in `arcapix.dynamo.server.dags.util` module. An example for defining a task using the DAG helper module is shown below:

```
from arcapix.dynamo.server.dags.util import task_template,
submit_subdag

def submit_stub_task(self, paths, source_site, dest_site, **kwargs):
    stub_template = task_template(
        "dynamo.tasks.reverse_stub",
        kwargs={
            "hydrate": False,
            "overwrite": True,
        },
        site=dest_site,
    )

    submit_subdag(
        paths=paths,
        task_id=self.request.id,
        "dynamocore.tasks.dag.run_subdag",
        subgraph_template=stub_template
    )
```

For complex sub-DAGs, the worker repository provides a helper module in `arcapix.dynamo.server.dag.helper`. This module allows for defining a chain of tasks using the helper functions like `add_task` and `map` for use with dynamic DAGs.

An example for defining a chain of tasks using the DAG helper module is shown below:

```
from arcapix.dynamo.server.dags.helper import Graph, add_task
from arcapix.dynamo.server.dags.util import submit_subdag

def submit_chain_of_dag_tasks(self, paths, source_site, dest_site,
**kwargs):
    graph = Graph()

    xattr = graph.add_task(
        "dynamo.tasks.remove_location_xattrs_for_moved",
        kwargs={"jobid": jobid},
        site=source_site,
    )
```

```

recall = graph.add_task(
    "dynamo.tasks.reverse_stub",
    parents=[xattr.id],
    kwargs={
        "jobid": jobid,
        "site": dest_site,
        "endpoint": endpoint,
        "hydrate": True,
        "overwrite": True,
        "skip_hash": True,
        "extra_flags": ["--skip-check-uuid"],
        "batch_size": REVERSE_STUB_BATCH_SIZE,
    },
    site=dest_site,
)

submit_subdag(
    paths=[{"path": i} for i in paths],
    task_id=self.request.id,
    "dynamocore.tasks.dag.run_subdag",
    subgraph_template=graph.to_dict()
)

```

This example demonstrates how to use the `Graph` and `add_task` helpers to create a chain of tasks and submit them as a sub-DAG. The `submit_subdag` function is used to submit the sub-DAG with the specified paths and task ID, using the dynamically generated task graph template.

Disaster Recovery / Cold Failover

It's possible to configure Ngenea Hub to be able to cold-failover to another node if it's running on a PixStor.

Setup

Configure datastore

Configure Ngenea Hub to store it's persistent data on the GPFS filesystem so it can be read by multiple nodes. This is done by settings the following setting in `/etc/sysconfig/ngeneahub`

```
DATA_DIR=/mmfs1/.arcapix/ngeneahub/data
```

Configure Networking

It's strongly recommended to configure a floating IP that can be used for the Ngenea Worker to connect to. This will allow cold failover without having to reconfigure workers.

This can be done by setting the following settings in `/etc/sysconfig/ngeneahub`:

- `SERVICE_CIDR`. Set this to the IP and netmask of the IP you want to be managed by ngeneahub. e.g. `192.168.2.3/24` for the IP `192.168.2.3` on a network with a netmask of `255.255.255.0`
- `SERVICE_INTERFACE`. Set this to the name of the interface the IP address should be added to. e.g. `man0`

Configure the workers to use this IP by editing `/etc/ngenea/ngenea-worker.conf` on each worker node and modifying `broker_url` and `result_backend`

Install Ngenea Hub

Install Ngenea Hub on multiple nodes as usual. Make sure `/etc/sysconfig/ngeneahub` are in sync across these nodes. Enable and start the service on one node only. Leave the service disabled and stopped on the other nodes.

Performing failover

In the case of a node failure, after confirming the services are no longer running on the other node, the following steps can be performed to bring the service up on another node:

important You must be certain the service is not running anywhere else before continuing, otherwise data loss can occur.

- Remove the lock file from `${DATA_DIR}/.lock`.
- Start the Ngenea Hub service

Migration from local datastore

After setting `DATA_DIR` in `/etc/sysconfig/ngeneahub` and restarting the service, data will automatically be migrated. This is a one-way operation.

External SSL

This document provides information on

- NGINX Installation on vanilla centos 7
- Configuration for SSL Termination and Reverse proxies to ngeneahub

Installing NGINX

Adding the EPEL Software Repository

```
sudo yum install epel-release
```

Installing NGINX

```
sudo yum install nginx
```

Starting Nginx service

```
sudo systemctl start nginx
```

Check Nginx service Status

```
sudo systemctl status nginx
```

Nginx status output should look like this

Output

```
● nginx.service - The nginx HTTP and reverse proxy server
   Loaded: loaded (/usr/lib/systemd/system/nginx.service; disabled;
  vendor preset: disabled)
   Active: active (running) since Mon 2022-01-24 20:14:24 UTC; 5s ago
     Process: 1898 ExecStart=/usr/sbin/nginx (code=exited, status=0/
  SUCCESS)
     Process: 1896 ExecStartPre=/usr/sbin/nginx -t (code=exited,
  status=0/SUCCESS)
     Process: 1895 ExecStartPre=/usr/bin/rm -f /run/nginx.pid
  (code=exited, status=0/SUCCESS)
    Main PID: 1900 (nginx)
      CGroup: /system.slice/nginx.service
              └─1900 nginx: master process /usr/sbin/nginx
                 └─1901 nginx: worker process

Jan 24 20:14:24 centos-updates systemd[1]: Starting The nginx HTTP
and reverse proxy server...
Jan 24 20:14:24 centos-updates nginx[1896]: nginx: the configuration
file /etc/nginx/nginx.conf syntax is ok
Jan 24 20:14:24 centos-updates nginx[1896]: nginx: configuration
file /etc/nginx/nginx.conf test is successful
Jan 24 20:14:24 centos-updates systemd[1]: Started The nginx HTTP and
reverse proxy server.
```

The service should be **active**

To stop the Nginx service

```
sudo systemctl stop nginx
```

To disable the Nginx service

```
sudo systemctl disable nginx
```

To Enable the Nginx service

```
sudo systemctl enable nginx
```

Configuration for SSL Termination and Reverse Proxy using OpenSSL

Create Self-Signed Certificates for Nginx

Create the Certificate Configuration file named `localhost.conf`

```
[req]
default_bits      = 2048
default_keyfile   = localhost.key
distinguished_name = req_distinguished_name
req_extensions    = req_ext
x509_extensions  = v3_ca

[req_distinguished_name]
countryName          = Country Name (2 letter code)
countryName_default  = US
stateOrProvinceName  = State or Province Name (full name)
stateOrProvinceName_default = New York
localityName         = Locality Name (eg, city)
localityName_default = Rochester
organizationName     = Organization Name (eg, company)
organizationName_default = localhost
organizationalUnitName = organizationalunit
organizationalUnitName_default = Development
commonName           = Common Name (e.g. server FQDN or YOUR name)
commonName_default   = localhost
commonName_max       = 64

[req_ext]
subjectAltName = @alt_names

[v3_ca]
subjectAltName = @alt_names

[alt_names]
DNS.1 = example.com
DNS.2 = 127.0.0.1
```

Create the Certificate using OpenSSL using below command

```
sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout
localhost.key -out localhost.crt -config localhost.conf
```

Copy the Certificate Key Pair to the Certificates folder `/etc/ssl/certs`

```
sudo cp localhost.crt /etc/ssl/certs/localhost.crt
sudo cp localhost.key /etc/ssl/private/localhost.key
```

Creating configuration file for Nginx

Create a configuration file in `/etc/nginx/conf.d/proxy.conf` and Update the Nginx Configuration File to Load the Certificate Key Pair

```
server {

    listen 80;
```

```

listen 443 ssl http2;
listen [::]:443 ssl http2; #for IPv6

server_name example.com;

#specify the certificate files to use
ssl_certificate /etc/ssl/certs/localhost.crt;
ssl_certificate_key /etc/ssl/private/localhost.key;

ssl_protocols TLSv1.2 TLSv1.1 TLSv1;

root /usr/share/nginx/html;

#Serving index.html file when requesting /
index index.html;

#Reverse proxy for requests
location / {
    proxy_set_header    Host $host:8000;
    proxy_set_header    X-Real-IP $remote_addr;
    proxy_set_header    X-Forwarded-For
$proxy_add_x_forwarded_for;
    proxy_set_header    X-Forwarded-Proto $scheme;

    proxy_pass http://localhost:8000/;#if running outside
docker you can use 127.0.0.1 or localhost instead of
host.docker.internal, or with docker with network_mode: "host"
    proxy_redirect off;
}
}

```

Reload the Nginx service after you have made some configuration changes

```
sudo systemctl reload nginx
```

Configuration for SSL Termination and Reverse Proxy using Certbot and LetsEncrypt (Another method)

Add trusted SSL Certificates from Letsencrypt

We need to redirect all unencrypted HTTP connections to HTTPS. This is done with certbot and letsencrypt certificates. The certbot will obtain free certificates and also handle the renewal process automatically. To do that we will install certbot and also a plugin for our NGINX server.

```
sudo yum install certbot python3-certbot-nginx
```

Once we have installed those packages, we can obtain our certificates.

```
sudo certbot --nginx -d example.com
```

It will ask you if you want to redirect all traffic from HTTP to HTTPS. Select yes (2). This automatically makes some changes to our NGINX default configuration.

```
server {  
    server_name example.com;  
  
    location / {  
        proxy_pass http://127.0.0.1:8000;  
    }  
  
    listen [::]:443 ssl ipv6only=on; # managed by Certbot  
    listen 443 ssl; # managed by Certbot  
    ssl_certificate /etc/letsencrypt/live/example.com/fullchain.pem;  
# managed by Certbot  
    ssl_certificate_key /etc/letsencrypt/live/example.com/  
privkey.pem; # managed by Certbot  
    include /etc/letsencrypt/options-ssl-nginx.conf; # managed by  
Certbot  
    ssl_dhparam /etc/letsencrypt/ssl-dhparams.pem; # managed by  
Certbot  
}  
server {  
    if ($host = example.com) {  
        return 301 https://$host$request_uri;  
    } # managed by Certbot  
  
    listen 80 default_server;  
    listen [::]:80 default_server;  
  
    server_name example.com;  
    return 404; # managed by Certbot  
}
```

HTTPS Configuration for NgeneaHub

To configure NgeneaHub for HTTPS, a configuration file named 'nghub.conf' should be created under etc/nginx/conf.d/pixstor/ folder. It should contain the configuration below:

```
location /ngeneahub/ {  
    proxy_pass http://localhost:8000;  
    proxy_http_version 1.1;  
    proxy_set_header Host $host;  
    proxy_set_header X-Forwarded-Proto $scheme;  
    proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;  
}  
location /ngeneahub/ws {  
    proxy_pass http://localhost:8000;  
    proxy_http_version 1.1;  
    proxy_set_header Host $host;
```

```
proxy_set_header Upgrade $http_upgrade;
proxy_set_header Connection "Upgrade";
proxy_set_header X-Forwarded-Proto $scheme;
proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
}
```

Reload the Nginx service after you have made some configuration changes

```
sudo systemctl reload nginx
```

To auto-renew the certificates, Run

```
certbot renew
```

LDAP / Active Directory Login

This section describes the LDAP / Active Directory integration for Ngenea Hub.

A user can log in to Ngenea Hub using their username and password stored in the LDAP / Active Directory server rather than a manually created account in Ngenea Hub.

LDAP services are required to implement [RFC2307](#) or RFC2307bis for provisioning of the LDAP schema.

As such, the username corresponds to the identity store user's `sAMAccountName`, etc.

Upon successful login, a Ngenea Hub user account will be automatically created for that user.

Where `LDAP_MIRROR_GROUPS` is configured, new groups are created in Ngenea Hub if not existing when the first user login is a member of the groups.

Administrators can assign Hub groups and permissions prior to subsequent logins for ease of use. Refer to `LDAP_MIRROR_GROUPS` in Managing users and groups from AD in HUB

If `LDAP_USER_SEARCH` and/or `LDAP_GROUP_SEARCH` is not set, all LDAP / Active Directory users can authenticate via the setting of `LDAP_DOMAIN`.

When LDAP is enabled, internal Ngenea Hub users can still be created and internally authenticated.

Configuration

The following settings control LDAP configuration.

The following are set in the main configuration file for Ngenea Hub at `/etc/sysconfig/ngeneahub`. Any setting which doesn't specify a default is required when `LDAP_ENABLED` is true.

After changing settings in `/etc/sysconfig/ngeneahub` a Hub restart is required to reflect the changes applied.

Example Configurations

No whitespace must be present in the below configurations.

LDAP

```
LDAP_ENABLED=True
LDAP_DOMAIN=ldap.example.com
LDAP_HOSTNAME=ldap://ldap.example.com:389
```

LDAP enumerating specific User and Group membership

```
LDAP_ENABLED=True
LDAP_DOMAIN=my.ldap.example.com
LDAP_HOSTNAME=ldap://ldap.example.com:389
LDAP_USER_SEARCH=cn=Users,dc=hubusers,dc=example,dc=com
LDAP_GROUP_SEARCH=cn=Groups,dc=hubgroup,dc=example,dc=com
```

LDAPS

```
LDAP_ENABLED=True
LDAP_DOMAIN=ldap.example.com
LDAP_HOSTNAME=ldaps://ldap.example.com:636
LDAP_GLOBAL_OPTIONS=OPT_X_TLS_REQUIRE_CERT:OPT_X_TLS_ALLOW
```

LDAP + STARTTLS

```
LDAP_ENABLED=True
LDAP_DOMAIN=ldap.example.com
LDAP_HOSTNAME=ldap://ldap.example.com:389
LDAP_START_TLS=true
LDAP_GLOBAL_OPTIONS=OPT_X_TLS_REQUIRE_CERT:OPT_X_TLS_ALLOW
```

Managing users and groups from AD in HUB

The user account which is generated for an AD user behaves the same as any other Ngenea Hub user. This means it can be assigned to Ngenea Hub groups, and will gain the permissions from those groups.

By default, a new AD user will not belong to any groups, and therefore will not have any permissions. A privileged user will need to assign the user to any appropriate Ngenea Hub groups.

If `LDAP_MIRROR_GROUPS` is enabled, then when a user logs in to Ngenea Hub, groups will be automatically be created for any AD groups the user belongs to (if the group doesn't already exist), and the user will be assigned to those groups.

Only groups belonging to the `LDAP_GROUP_SEARCH` domain will be populated. If `LDAP_USER_SEARCH` is not set, we use `LDAP_DOMAIN` to work it out, then all AD groups

that authenticated users belong to would be created. This is usually not what you want. Example of explicit search terms:

```
LDAP_USER_SEARCH=cn=Users,dc=hubusers,dc=example,dc=com
```

```
LDAP_GROUP_SEARCH=cn=Groups,dc=hubgroup,dc=example,dc=com
```

NOTE: When either `LDAP_USER_SEARCH` or `LDAP_GROUP_SEARCH` is not set, the search is constructed from domain e.g. `LDAP_DOMAIN=ldap.example.com` the search string will be `dc=ldap,dc=example,dc=com`.

Mirrored AD groups behave the same as any other Ngenea Hub group, meaning permissions can be assigned to them to apply role-based access controls (RBAC). By default, mirrored AD groups will have no permissions assigned.

Any user can be assigned to a Hub mirrored AD group. Assigning a user to a Hub mirrored group does not change group membership in AD.

How can I manage my Hub users and groups using AD?

1. What settings are required?

```
LDAP_MIRROR_GROUPS=True
```

This will create an AD group in Hub for every group the user is a member of.

2. If a user is added or removed from a group in AD, when is this reflected in Hub?

AD changes are not immediately reflected in Hub.

If a user is added to an AD group, the group membership for the individual user is updated on next authentication (API or UI login).

If a user is removed from an AD group, the group membership for the individual user is updated on next authentication (API or UI login).

Example:

- If user1 has been added to a group in AD, the Hub state will not reflect the AD changes viewed by user2 until user1 logs in and user2 refreshes their view of users and groups.
- If user1 has been removed from a group in AD, the Hub state will not reflect the AD changes viewed by user2 until user1 logs in and user2 refreshes their view of users and groups.

3. How can I limit the groups created in Hub when `LDAP_MIRROR_GROUPS=True`

Create a Hub specific OU in AD.

Create groups to mirror to Hub in the OU.

Add the required users in the `ou=Hub/<groups>`

Only the groups in the `ou=Hub` will be mirrored to Hub on login.

4. How do I set up the groups in readiness for future user logins?

Create an AD user.

Log the user in.

Initially the AD user will not be granted permissions.

As the Hub admin, set the permissions on the groups.

Log out the user AD user and on next login they will be assigned to the groups and pick up the set permissions.

When future users log in, they will be assigned to the groups and pick up the set permissions on login.

5. If a user or group is deleted from AD how is this reflected in Hub?

Users deleted from AD are not automatically deleted from Hub.

The correlating Hub user account is unable to authenticate with Hub.

Groups deleted from AD are not automatically deleted from Hub.

Users who are currently logged in will retain security permissions until the group is deleted from Hub.

Systems administrators must ensure to delete users and groups from Hub after their removal from AD.

6. Does changing the password for a user affect the user hub?

Hub does not store credentials for AD users.

AD users who log in to Hub are authenticated against AD prior to allowing the user to login.

If Hub cannot contact AD to authenticate the user, the user is not allowed to login.

7. Does user and group management in Hub affect AD?

No, Hub 'mirrors' the group membership on login, but the mirror is only from AD to Hub.

There are no updates pushed from Hub to AD.

Queues

Custom job queues can be created with dedicated settings, and jobs can be assigned to run on those queues.

For example:

- a `highpriority` queue with a high number of threads, allowing for faster task processing
- a `lowpriority` queue with a low number of threads for jobs which do not need to complete promptly
- a dedicated queue for transparent recall jobs, so that those jobs are never blocked by other workflows

In addition to custom queues, there is a `default` queue which always exists for all workers. The default queue is used when a job is submitted without specifying a target queue.

Configuration

Queues are configured in the worker configuration file located at `/etc/ngenea/ngenea-worker.conf`

A queue is defined by adding a new `Queue` section to the config file. The following creates a queue named `highpriority` with 20 threads.

```
[Queue highpriority]
threads = 20
```

Queue names may contain letters, numbers, underscores, and hyphens. Queue names may not contain whitespace.

Queues support the following settings:

- `threads`: how many tasks can run in parallel on the queue

If no settings are provided under the queue section, the queue will inherit the global settings.

If no `threads` count is set in the global settings section, then a default of 10 is used.

For example, in the following

```
[settings]
...
threads = 4

[Queue transparent_recall]

[Queue highpriority]
threads = 10
```

The `highpriority` explicitly sets the number of threads to 10.

The `transparent_recall` queue does not explicitly set a number of threads, so it inherits `threads = 4` from global settings.

Custom job queues have three functions

- `worker` where core workflow tasks run
- `discovery` where discovery tasks (recursive, snapdiff) run
- `custom` where `custom plugin` tasks run

The `default` queue, which always exists, has those same three function plus two additional functions

- `interactive` where internal tasks such as file browsing are run
- `settings` where settings tasks, such as space creation and policies, are run

These two additional functions are not available on custom queues.

By default, each function gets the same number of threads. The number of threads can be configured per function

```
[Queue highpriority]
threads = 20
custom = {"threads": 10}
```

In this example, the `worker` and `discovery` functions will get 20 threads, but the `custom` function will only get 10 threads.

Per-function settings under the main `[settings]` section will only apply to functions of the `default` queue; they are not inherited.

Disabling Functions

Functions can be disabled for a queue by setting the function to `false`

For example, to disable the `custom` function, meaning custom plugin tasks cannot run on that queue

```
[Queue no-custom]
custom = false
```

In the same way, `default` queue functions can be disabled.

For example, in a cluster, you may set the management node to only run settings tasks

```
# management node
[settings]
discovery = false
worker = false
custom = false
```

And, you set ngenea nodes to only run workflow tasks, and not run settings tasks

```
# ngenea node
[settings]
settings = false
```

Warning: Each default queue function must be enabled on at least one node within a cluster, or else Hub will not function correctly.

Queue discovery

When a new queue is added to the worker config file, the change will be detected and the new queue will start up automatically. When a queue is removed from the worker config, the change will be detected and the queue will be shutdown automatically.

If this does not occur, or does not occur fast enough, a reload can be forced with

```
systemctl reload ngenea-worker
```

This will reload the worker config and start up/shutdown queues according to the worker config, without interrupting any queues which already exist and have not been changed.

Ngenea Hub uses worker heartbeats to automatically discover and register newly started queues. New queues will be visible and ready for use in Ngenea Hub workflows shortly after being added to the worker config.

When a queue is removed and shutdown, it will not be automatically de-registered from Ngenea Hub.

Removed queues **must** be manually de-registered using

```
ngeneahubctl manage remove_queue <site> <queue>
```

To re-use the same queue name of a previously removed queue, the `--offline-only` flag can be used. This must be preceded by a complete worker restart.

```
systemctl restart ngenea-worker
```

`--offline-only` will remove the queue only if the queue is offline. Note: If the queue is still alive, it will be recreated. If you want to disable the queue and ensure that it is then removed entirely from the hub, use `--offline-only`

Warning: If you submit a job to a queue which has been shutdown and not de-registered, the job will not be processed unless the queue is configured and started up again.

Memory considerations

Queues use roughly 40MB of RAM per thread when no tasks are running.

Since each queue has three functions, if all three functions in a given queue are configured with the same number of threads, then the memory requirement is roughly `threads * 125MB`.

For example, creating a high priority queue with 20 threads per function (60 threads total) will use ~2.5GB of memory when no tasks are running.

Similarly, creating a low priority queue with 5 threads per function (15 threads total) will use ~625MB of memory when no tasks are running.

Task execution will consume additional memory on top of this baseline memory usage. Since more threads mean more tasks can run at the same time, a queue with a large number of threads can lead to significant memory usage at peak.

Workflows

A queue can be passed when submitting a workflow. If no queue is passed, the `default` queue will be used.

If a workflow contains tasks which run on multiple sites, such as `send` or `sync` workflows, the same queue will be used for both sites. If the named queue doesn't exist on one of the sites, the `default` queue will be used instead.

A different queue can be specified for each task in a workflow by providing the queue as a parameter in the task steps, for example

```
{
  "name": "dynamo.tasks.migrate",
  "queue": "highpriority"
}
```

Workflow 'fields' can be used to allow selecting per-task queues at runtime.

For example, the `built-in` `send` workflow provides a `destinationqueue` field, which can be passed at runtime in a similar way to `destinationsite`

```
{
  "paths": [
    "/mmfs1/data/project_one",
  ],
  "site": "london",
  "queue": "highpriority",
  "workflow": "send_to_site",
  "fields": {
    "destinationsite": "dublin",
    "destinationqueue": "lowpriority"
  }
}
```

For more information on constructing and running workflows, see [Custom Workflows](#)

Usage

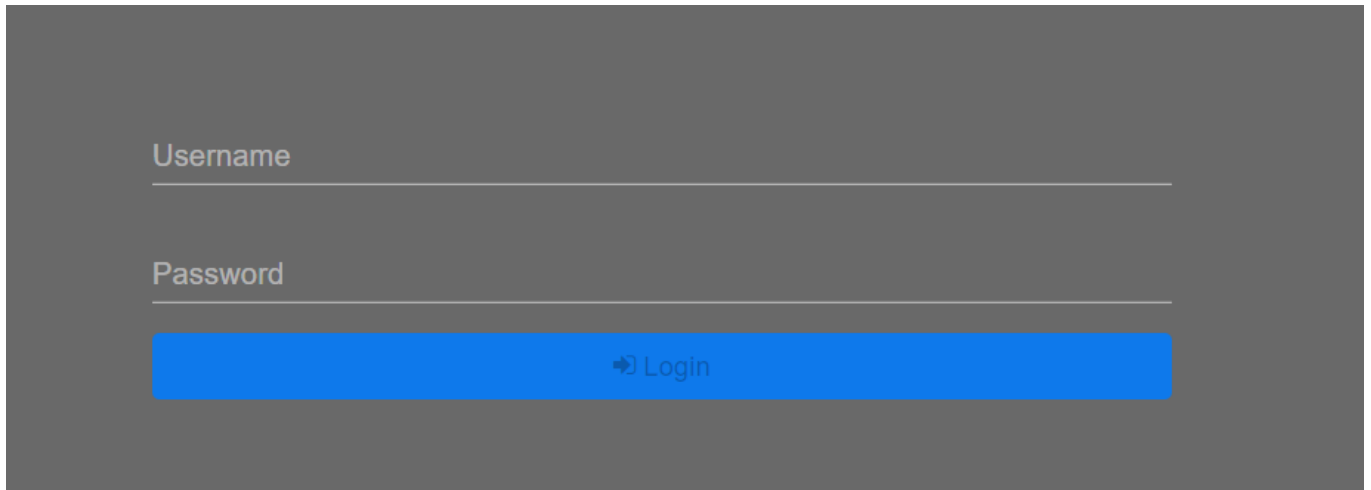
Web Interface

To access Ngenea Hub, go to <http://example.com:8000/>.

Authentication

Login

Upon navigating to the Ngenea Hub UI a login screen is presented.

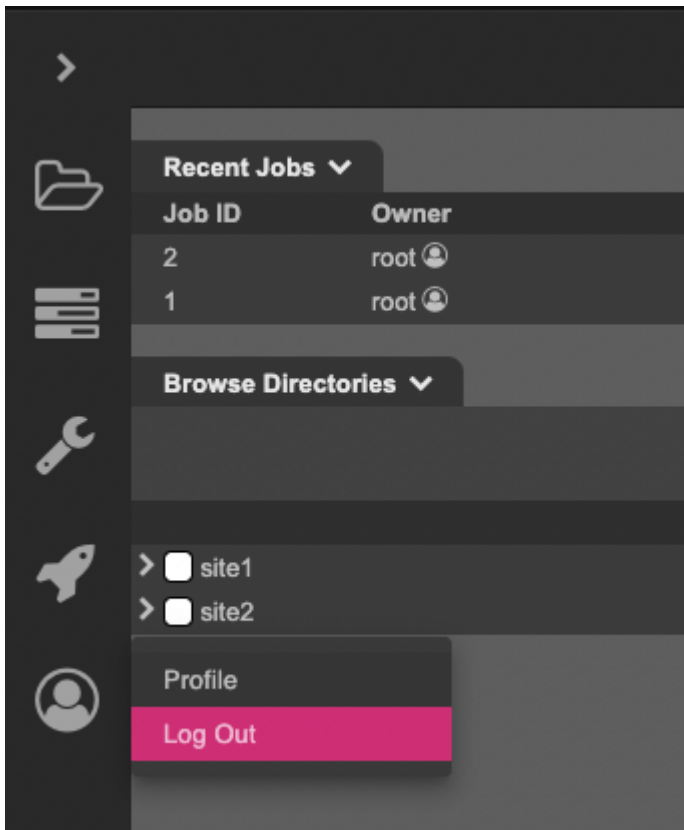


The screenshot shows a login interface on a dark gray background. There are two text input fields, one labeled 'Username' and one labeled 'Password'. Below these fields is a large blue button with the text 'Login' and a small icon to its left.

Enter a valid username and password before pressing the **Login** button to authenticate.

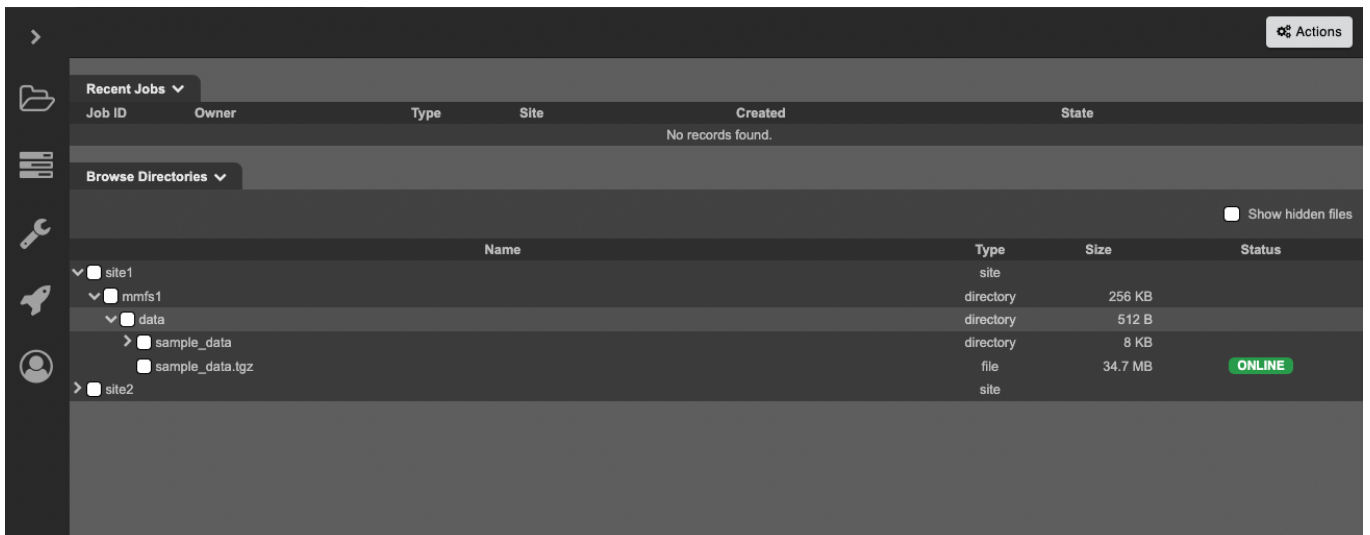
Logout

To end your session, select Logout from the **Man Icon** at the bottom left hand corner of the UI.



Browser

Upon logging in, select the directory icon on the left hand side of the UI. This takes to the Browser page.



The Browser page allows users to see a list of recent jobs as well as browse all available directories across all Sites for the purpose of either migrating, pre-migrating, recalling, or sending files.

Recent Jobs

The Recent Jobs section shows a list of the 5 most recent jobs that were initiated via the UI.

The view presents several columns:

Option	Description
Job ID	Shows the identification number associated with the job
Owner	Name of the user who created the job
Type	Shows the job's type, i.e. Migrate, Recall, Premigrate, or Send
Site	Name of the Site where files were migrated from
Created	Shows the job's creation time
State	Shows the job's state, i.e. success or failure

Search

Note: Before you can use the search feature, additional set-up is necessary, as described in the [search feature](#) page.

Search section contains a search bar to discover the contents of the configured Sites.

The view presents a search bar, a button for managing the search filters and another button for starting and stopping the search. The view also presents (?) button to get more instructions about starting a search.

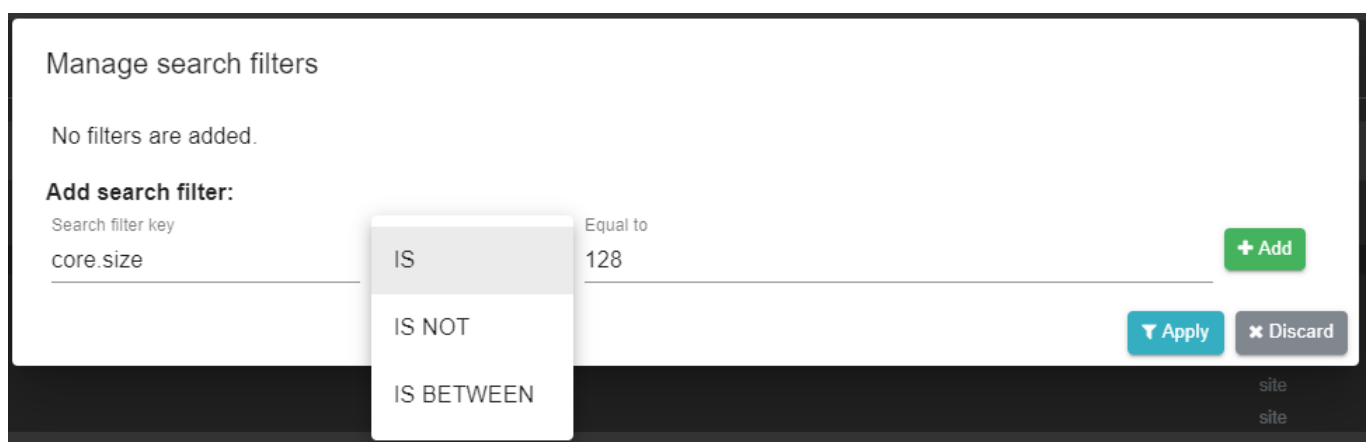


To search for a term, type the search term in the search bar and click **Search** button (or hit **Enter** key) to start a search. The default search filter is "core.filename".

Managing the search filters from the dialog

For more complex search operations, add more filters by clicking **Apply Filters** button and use the dialog opened after that. Type the search filter key in the input box with the label **Search filter key** and then pick an operator from the select box with the label **Operator**.

For the operator **IS**, the dialog is seen as in the image below:



For the operator **IS NOT**, the dialog is seen as in the image below:

Manage search filters

No filters are added.

Add search filter:

Search filter key	Operator	Equal to	
core.size	IS NOT	128	+ Add

▼ Apply ✕ Discard

For the operator **IS BETWEEN**, the dialog is seen as in the image below:

Manage search filters

No filters are added.

Add search filter:

Search filter key	Operator	Greater than or equal to	Less than
core.size	IS BETWEEN	36	187

+ Add

▼ Apply ✕ Discard

Click **Add** button to add the search filter. The dialog is seen as in the image below, when "core.size" is selected to be greater than or equal to 36 and less than 187, and "core.group.name" is selected to be "root".

Manage search filters

core.size: {"gte":36,"lt":187} ✕

core.group.name: root ✕

Add search filter:

Search filter key	Operator	Equal to	
	IS		+ Add

▼ Apply ✕ Discard

Once the selection is finished, click **Apply** button to save & close the dialog (this action does not submit the search). See that filters are shown as a query in the search bar and they are shown with badges under the search bar. The search section is seen as in the image below, after the search filters are selected:

Search Items ▼

Enter a search term

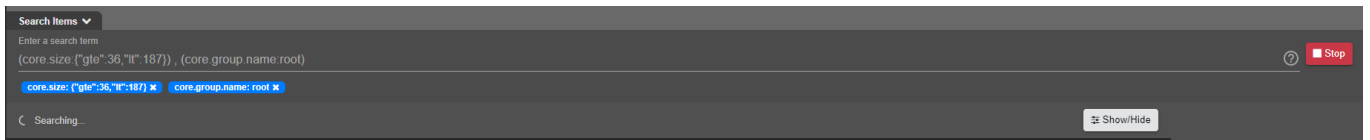
(core.size:{"gte":36,"lt":187}) , (core.group.name:root)

core.size:{"gte":36,"lt":187} ✕ core.group.name:root ✕

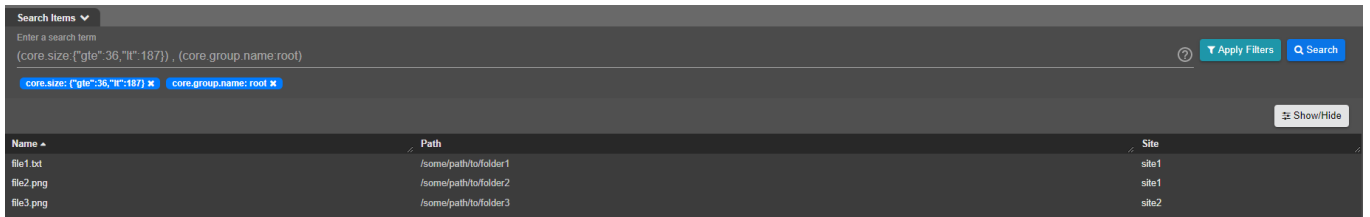
▼ Apply Filters 🔍 Search

Submitting the search and viewing search results

Click **Search** button, or hit **Enter** key to start the search. The search section will contain search results from each configured site, matching with the search filters. To stop the search, click **Stop** button. The search section is seen as in the image below after the search is in progress.



The search section is seen as in the image below after some results are found.



Advanced usage: Managing the search filters from the search bar

You can also make complex search operations by typing the search term in the format below:

```
(<key1>:<value1>) , (<key2>:<value2>) , (<key3>:<value3>) ...
```

For multiple search filters, you need to separate them by paranthesis. For using various comparison operators, here are the filter formats:

Operation	Format
Equal to	(<key1>: <value1>)
Less than	(<key2>: {"lt": <value2> })
Greater than or equal to	(<key3>: {"gte": <value3> })
Combined	(<key4>: {"gte": <value4>, "lt": <value5> })

Browse Directories

The Browse Directories section contains a list of configured Sites and correspondent directories under the Sites' filesystems.

The view presents several columns:

Option	Description
Name	Shows the name of the Site, directories, and files
Type	Shows whether the listed item is a Site, directory, or file
Size	Shows the directories and files size
Status	Shows whether files are online or offline

User can select one or multiple directories, or one or multiple files to migrate, premigrate, recall them or send them to a different Site.

Migrate

To Migrate a directory or file, expand the Site containing said directory and file. Select the directory or file you wish to Migrate by ticking their relevant boxes.

Click the "Actions" button at the top right hand side of the page and select "Migrate".

A new Job is created and it is shown at the top of the "Recent Jobs" list. Job's State will display a progress bar until completion.

Once job is complete, the State will either show as Success or Failed.

Premigrate

To Premigrate a directory or file, expand the Site containing said directory and file. Select the directory or file you wish to Premigrate by ticking their relevant boxes.

Click the "Actions" button at the top right hand side of the page and select "Premigrate".

A new Job is created and it is shown at the top of the "Recent Jobs" list. Job's State will display a progress bar until completion.

Once job is complete, the State will either show as Success or Failed.

Recall

To Recall a directory or file, expand the Site containing said directory and file. Select the directory or file you wish to Recall by ticking their relevant boxes.

Click the "Actions" button at the top right hand side of the page and select "Recall".

A new Job is created and it is shown at the top of the "Recent Jobs" list. Job's State will display a progress bar until completion.

Once job is complete, the State will either show as Success or Failed.

Send

Premigrate behaviour change: Prior to Ngenea Hub 1.8.0, the send workflow would migrate data on the source site. This has been changed to pre-migrate.

To Send a directory or file from one Site to another, expand the Site containing said directory and file. Select the directory or file you wish to Send by ticking their relevant boxes.

Click the "Actions" button at the top right hand side of the page and select "Send".

Select the Site you wish to send the directory and/or files to. Tick the "Hydrate files on destination" if required, and click "Confirm".

Confirm running workflow

Please confirm you wish to run workflow 'Send' for the following:

Site:

hci-lab

Folders (including all contents and subfolders):

/mmfs1/data/vfx/apps/autodesk/plugins

Items:

-

Additional fields:

Destination Site

☐ Hydrate files on destination

Confirm

Cancel

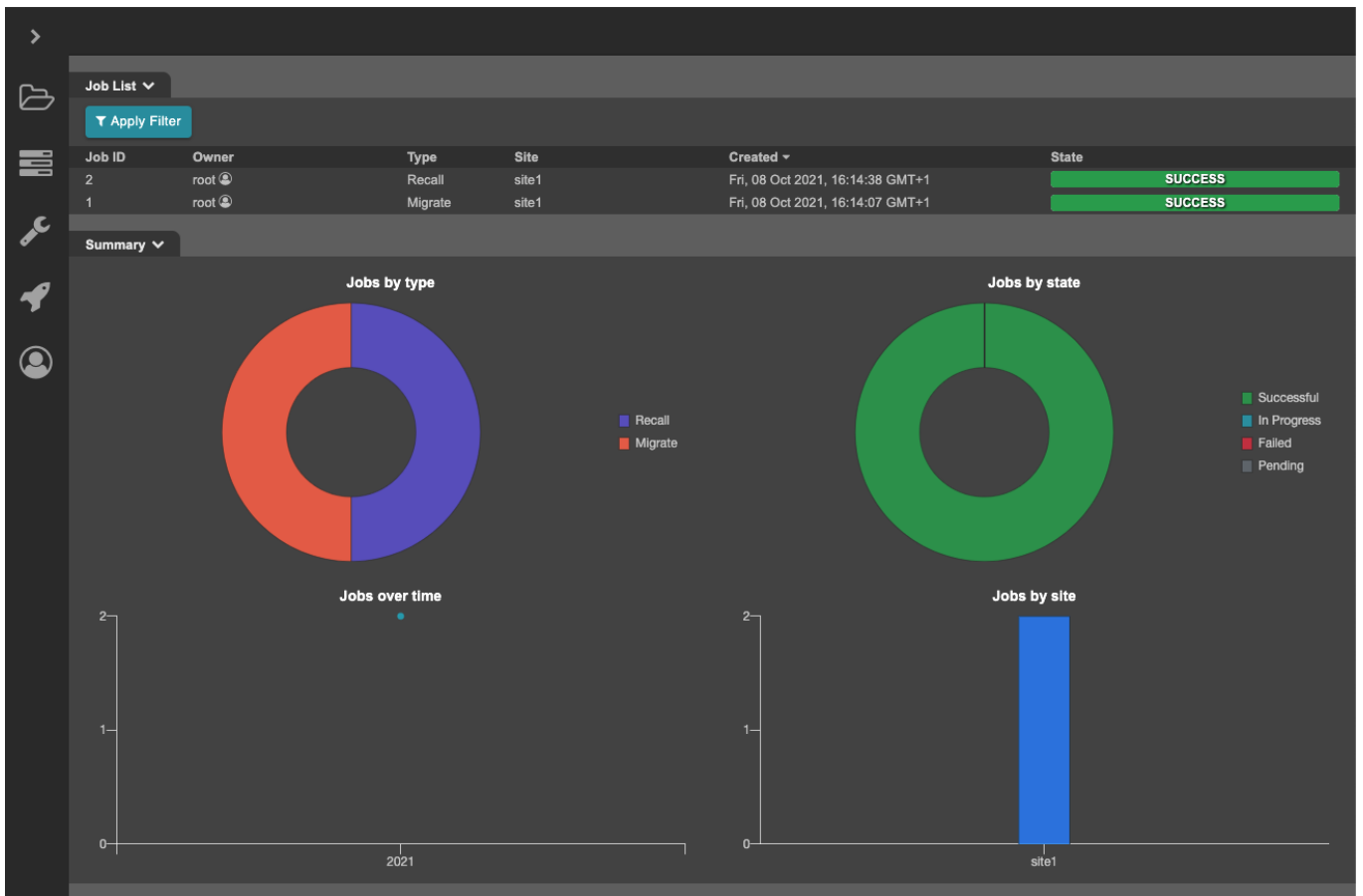
A new Job is created and it is shown at the top of the "Recent Jobs" list. Job's State will display a progress bar until completion.

Once job is complete, the State will either show as Success or Failed.

Expanding the receiving Site's Directories now shows the path that was replicated from the sending Site.

Jobs

The Jobs page shows a list of all the jobs that were initiated via the UI.



The view presents several columns:

Option	Description
Job ID	Shows the identification number associated with the job
Owner	Name of the user who created the job
Type	Shows the job's type, i.e. Migrate, Recall, Premigrate, or Send
Site	Name of the Site where files were migrated from
Created	Shows the job's creation time
State	Shows the job's state, i.e. success or failure

Each column can be sorted in ascending and descending order.

Pagination

To select whether to view 20, 50, or 100 Jobs at the time, choose the relevant option in the "Items per Page" dropdown.

Clicking on the right and left arrow next to "Items per Page" will take you to the next/previous pages.

Apply Filter

Jobs list can be filtered by time period, job type, and job state.

To filter the list, select the "Apply Filter" button on the top left hand side of the UI.

Select one or a combination of filters, and click "Apply".

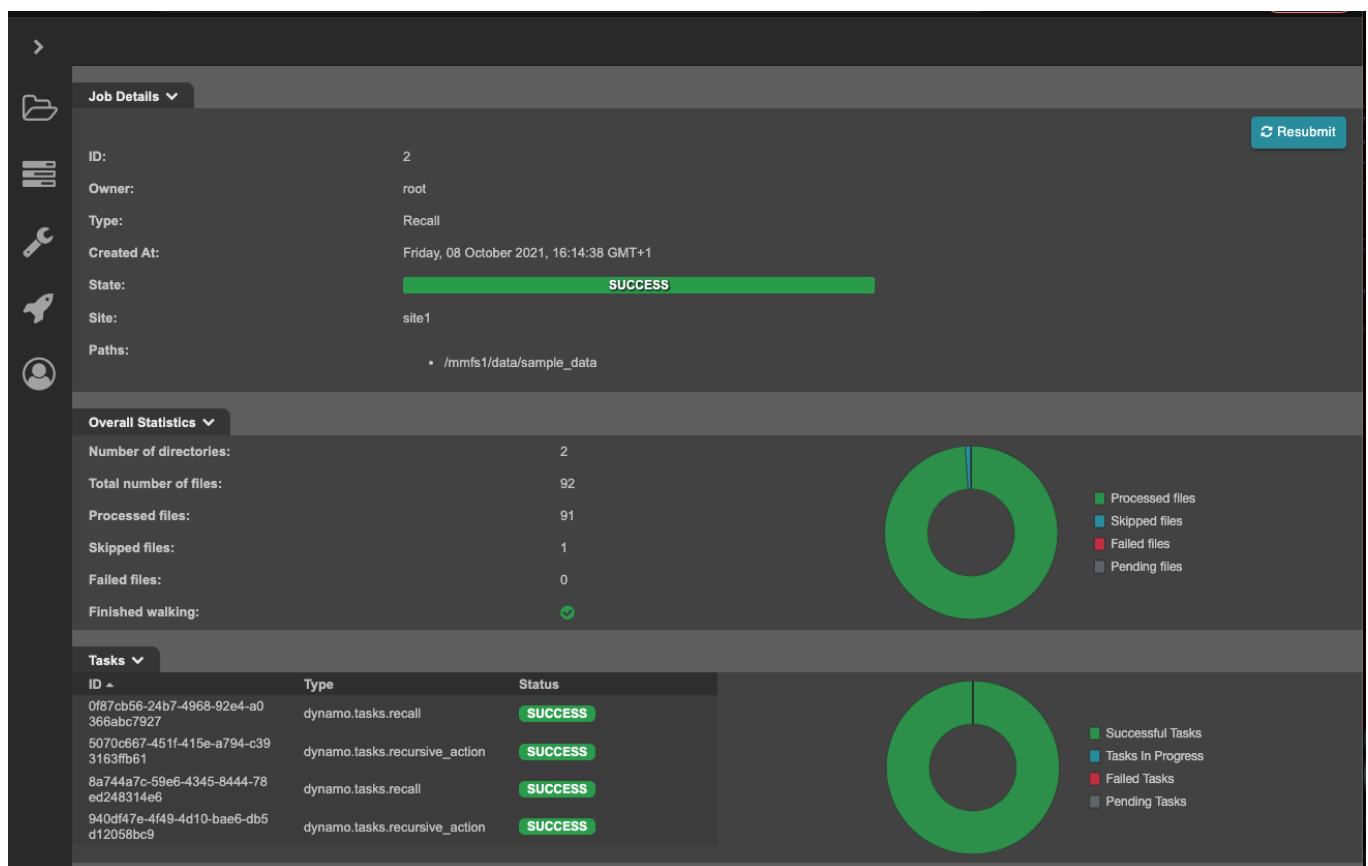
Job are now filtered as per your selection.

To remove a filter, simply select the "x" next to the applied filter.

Job ID

On the Jobs page, click on a Job ID to see additional information regarding the Job.

The Job Details, Overall Statistics, and Tasks tab are displayed.



Each tab shows specific Job details, some of which are clickable:

- Overall Statistics --> Total number of files, processed files, skipped files, and failed files.
- Tasks --> ID

Selecting any of the clickable items opens a dialogue showing the relevant output.

Jobs can also be resubmitted by clicking the "Resubmit" button at the top right hand side of the page.

Owner

On the Jobs page, click on any Owner to see additional information regarding the user who initiated the Job.

Selecting an Owner takes to a page that shows details about the user, as well as a list of Jobs initiated by said user.

User Details

Username: root

First Name: -

Last Name: -

Email: -

Date Joined: Friday, 08 October 2021, 15:54:35 GMT+1

Last Login: Friday, 08 October 2021, 15:54:58 GMT+1

Active: ✔

[Update Profile](#)

Job List

Job ID	Owner	Type	Site	Created	State
2	root	Recall	site1	Fri, 08 Oct 2021, 16:14:38 GMT+1	SUCCESS
1	root	Migrate	site1	Fri, 08 Oct 2021, 16:14:07 GMT+1	SUCCESS

If you selected your own user, you will see an "Update Profile" on the top right hand side of the page.

This takes you to the "Update User" page where you can change your own password, email, first name and last name.

Update user: root

Password

Confirm Password

Email

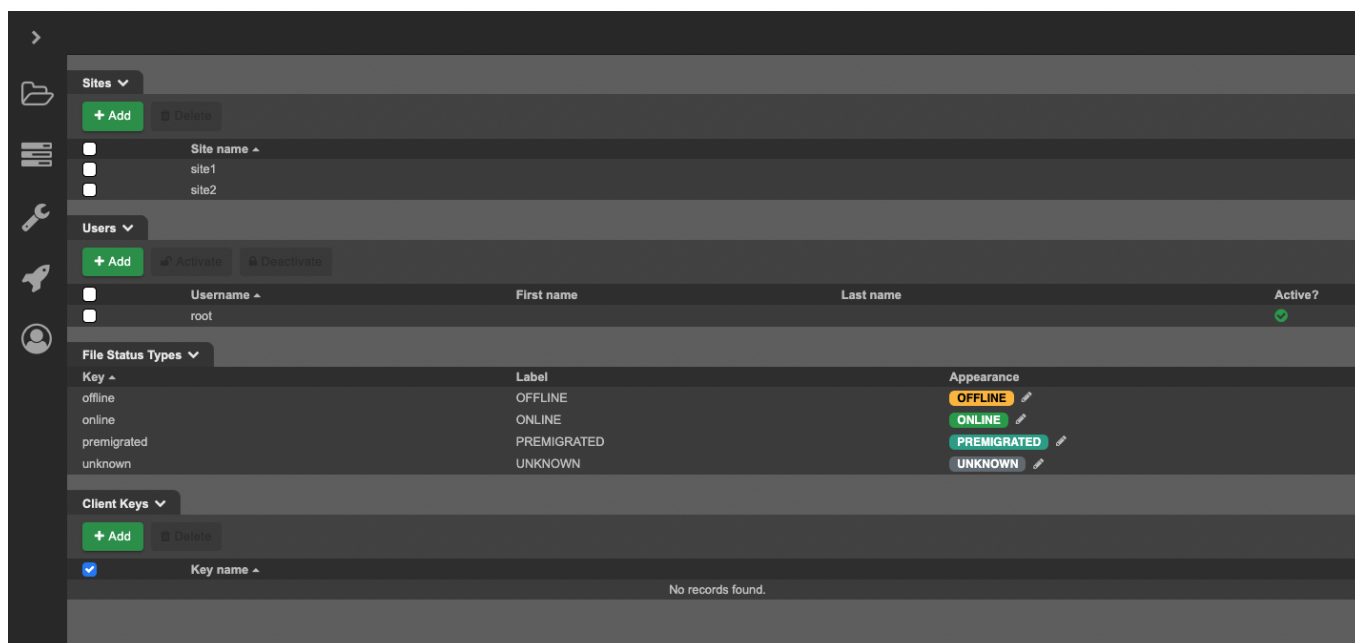
First Name Last Name

[Save](#) [Cancel](#)

The Job List's layout is the same as the one shown in Jobs page and has the same functionalities.

Administration

The Administration page allows you to add and delete Sites, add, activate and deactivate Users, as well as changing the name and colours of the labels found in the Browse Directories section.



Sites

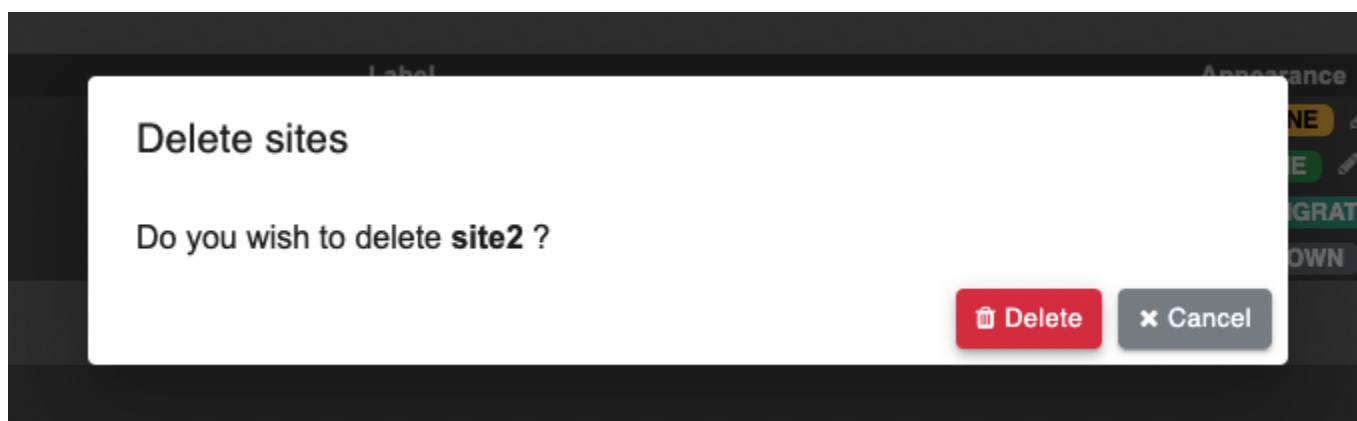
The Sites tab contains the list of Sites that have been configured. This list can be sorted by Site name in ascending and descending order.

Add Site

To add a Site, select the "Add" button, enter a Site name, and confirm by clicking "Add"

Delete Site

To delete a Site, select one or multiple Sites by ticking their correspondent boxes. Click "Delete" and then confirm deletion once the "Delete Site" dialogue is presented.

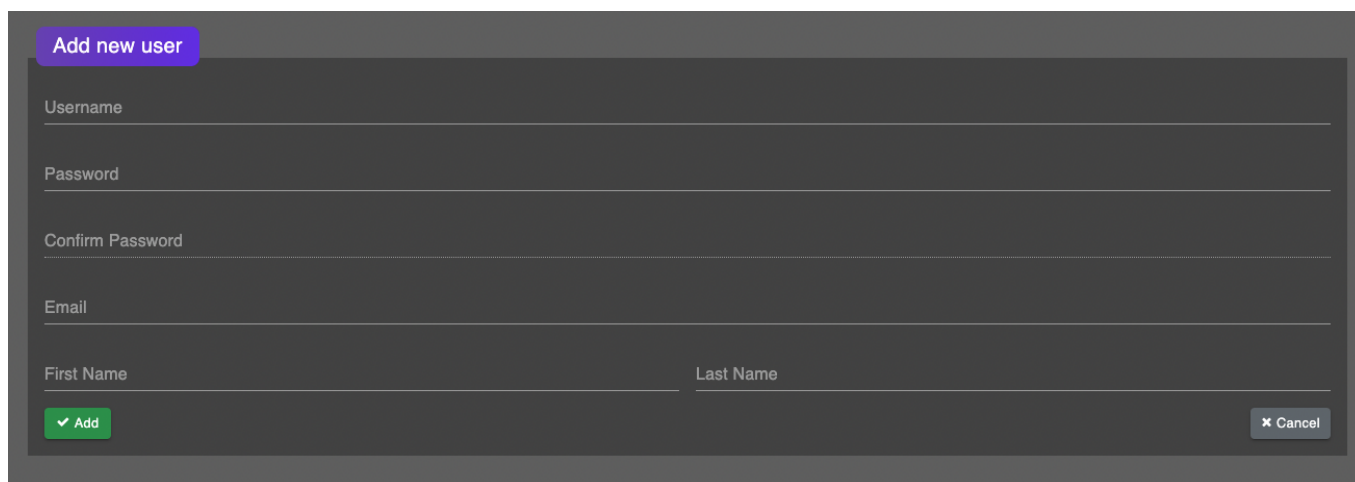


Users

The Users tab contains the list of Users who are allowed to use the UI. This list can be sorted by Username in ascending and descending order.

Add User

To add a new User, select the "Add" button, then enter a Username, Password, Email, First Name, and Last Name.

A dark-themed form titled "Add new user" in a purple button. The form contains input fields for "Username", "Password", "Confirm Password", "Email", "First Name", and "Last Name". At the bottom left is a green "Add" button with a checkmark, and at the bottom right is a grey "Cancel" button with an 'x' icon.

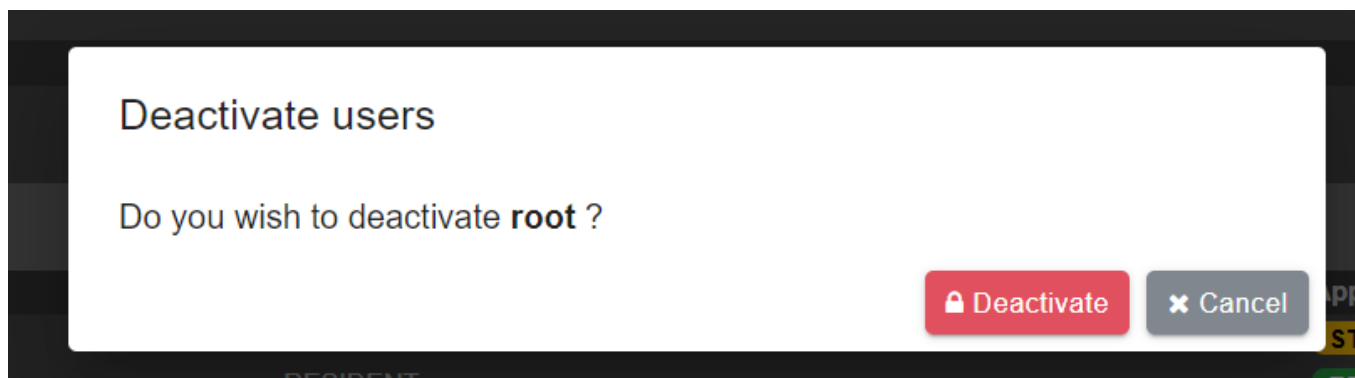
Confirm by clicking "Add".

Activate User

To activate an existing User, select the "Activate" button, and then confirm activation once the "Activate user" dialog is presented.

Deactivate User

To deactivate a User, select one or multiple Users by ticking their correspondent boxes. Click "Deactivate" and then confirm deactivation once the "Deactivate user" dialogue is presented.



User will no longer have access to the UI but still exists and can be reactivated at any time.

File Status Types

The File Status Types tab lists the labels that are used to indicate a file's status.

The labels are customisable as both colour and label name can be changed.

To change the appearance of a label, select the pencil symbol next to the label that needs updating.

"Update File Status Type" page is displayed:

Update file status type: offline

Label
STUBBED

Background Color Text Color

STUBBED

Save Cancel

Change label name, background colour, or text colour as per your preference.

Click "Save" to confirm the update.

API

Reference

The API reference can be found at your Ngenea Hub install at [http\(s\)://myhub/api/docs/](http(s)://myhub/api/docs/). This section does not attempt to duplicate the reference, but instead provide some usage examples.

Swagger
https://myhub/api/docs/?format=openapi Explore

Ngenea Hub API v1
[Base URL: myhub2/api]
<https://myhub2/api/docs/?format=openapi>
This API allows users use Ngenea Hub functionalities via REST calls.

Schemes
HTTPS Authorize

Filter by tag

actions

- GET /actions/
- POST /actions/
- GET /actions/{id}/

alerts

- GET /alerts/
- GET /alerts/{id}/

analytics

- GET /analytics/

auth

- GET /auth/clientkeys/

Authentication

Authentication to the API can be performed in 2 ways

- JWT Authentication
- Client Keys

The first one is for interacting with the API interactively and is therefore most likely not suitable for building automated workflows. On the other hand, client keys are valid until they are revoked and are more suitable for automation.

JWT Authentication

To use the API directly, authentication tokens should be generated to prevent sending the username and password repeatedly. You need to generate these tokens by sending your username-password pair to the login endpoint: `/auth/token/`

```
curl -s -X POST 'http://localhost:8000/api/auth/token/' -H 'Accept: application/json' -H 'Content-Type: application/json' -d '{"username": "dfoster", "password": "*****"}' | jq -r '{
  "access": <access_token>,
  "refresh": <refresh_token>,
}
```

There are 2 types of authentication tokens:

- Access token
- Refresh token

Access tokens are used for doing API requests. You need to include the token in the Authorization header to use any other endpoint:

```
curl -s -X GET 'http://localhost:8000/api/jobs/' -H 'Accept: application/json' -H 'Content-Type: application/json' -H "Authorization: Bearer $JWT_ACCESS_TOKEN" | jq '{
  "count": 0,
  "next": null,
  "previous": null,
  "results": [],
  "stats": {
    "type": {
      "migrate": 0,
      "premigrate": 0,
      "recall": 0
    },
    "state": {
      "SUCCESS": 0,
      "FAILURE": 0,
      "STARTED": 0,
      "PENDING": 0,
      "ERROR": 0
    },
    "created": {},
    "site": {}
  }
}
```

On the other hand, refresh tokens are used for refreshing the access token. For security purposes, access tokens expire in 1 hour and refresh tokens expire in 1 day. When an expired token is used, one of HTTP 401 Unauthorized and HTTP 403 Forbidden errors is received. In that case, you need to refresh the access token with `/api/token/refresh/` endpoint:

```
curl -s -X POST 'http://localhost:8000/api/auth/token/refresh/' -H
'Accept: application/json' -H 'Content-Type: application/json' -d
'{"refresh": "<refresh_token>"}' | jq -r
{
  "access": <new_access_token>,
}
```

Refresh tokens can also be expired too. In that case, you need to send your credentials (username and password) again to obtain new token pair.

Client Keys

Creating Client Keys

Note: The UI does not currently support creating client keys and therefore have to be done via the API directly

Before we can authenticate using the client key, we need to temporarily authenticate using JWT to be able to create a client key.

To get a valid JWT access token using curl and jq:

```
export JWT_TOKEN=$(curl -s -X POST 'http://localhost:8000/api/auth/
token/' -H 'Accept: application/json' -H 'Content-Type:
application/json' -d '{"username": "dfoster", "password": "*****"}'
| jq -r .access)
echo $JWT_TOKEN
<token>
```

This can now be used to create a client key:

```
curl -s -X POST 'http://localhost:8000/api/auth/clientkeys/' -H
'Accept: application/json' -H 'Content-Type: application/json' -H
"Authorization: Bearer $JWT_TOKEN" -d '{"name": "my_automation_key"}'
| jq '.'
{
  "url": "http://localhost:8000/api/auth/clientkeys/1/",
  "id": 1,
  "name": "my_automation_key",
  "api_key": "YOUR_API_KEY"
}
```

{warning} This **is** the only time the client key will be visible, make sure it **is** recorded.

Using Client Keys

The key created in the previous section can now be used by setting the header Authorization: Api-Key YOUR_API_KEY against an API endpoint. For example:

```
export API_KEY=YOUR_API_KEY

curl -s -X GET 'http://localhost:8000/api/jobs/' -H 'Accept:
```

```

application/json' -H 'Content-Type: application/json' -H
"Authorization: Api-Key $API_KEY" | jq
{
  "count": 0,
  "next": null,
  "previous": null,
  "results": [],
  "stats": {
    "type": {
      "migrate": 0,
      "premigrate": 0,
      "recall": 0
    },
    "state": {
      "SUCCESS": 0,
      "FAILURE": 0,
      "STARTED": 0,
      "PENDING": 0,
      "ERROR": 0
    },
    "created": {},
    "site": {}
  }
}

```

Submitting Workflow

To submit a workflow, the following parameters are required:

name	description
workflow	The name of the workflow to submit
paths	A list of paths to execute the workflow on
site	The name of the site where the workflow should be started from. Steps within a workflow may run on different sites.

In addition, the following optional parameters may be provided:

name	description
discovery	Name of the file discovery technique to use. Currently the only supported discovery is <code>recursive</code> . If no discovery is specified, <code>recursive</code> will be used as the default. If explicitly set to <code>null</code> , no discovery will be performed and the provided paths will be used 'as is'.
fields	Additional parameter for the workflow, typically used by custom workflows.

Migrate

Using a Client Key stored in a environment variable `TOKEN`, the following is an example of migrating a file using `curl`.

```
curl -s -X POST 'http://example.com/api/file/workflow/' -H 'Accept: application/json' -H 'Content-Type: application/json' -H "Authorization: Api-Key $TOKEN" -d '{"paths": ["/mmfs1/data/sample_data.tgz"], "site": "dfoster1", "workflow": "migrate", "discovery": null}'
```

Note: Since we're only migrating a single file, we don't need recursive discovery, so discovery has been set to `null` to disable it.

Trailing Slashes

In general, the endpoint used with `POST` requests must have a trailing slash.

For example, a `POST` request made to `http://example.com/api/file/workflow` would error, but if made to `http://example.com/api/file/workflow/` it would work.

Monitoring and Management

systemd service

Ngenea Hub is controlled via the `ngeneahub` `systemd` service

ngeneahubctl cli tool

The `ngeneahubctl` tool can be used to manually stop/start the services, outside of `systemd`, for debugging

Docker containers

Ngenea Hub uses a collection of docker containers, which can be managed by standard Docker monitoring/management tools and processes:

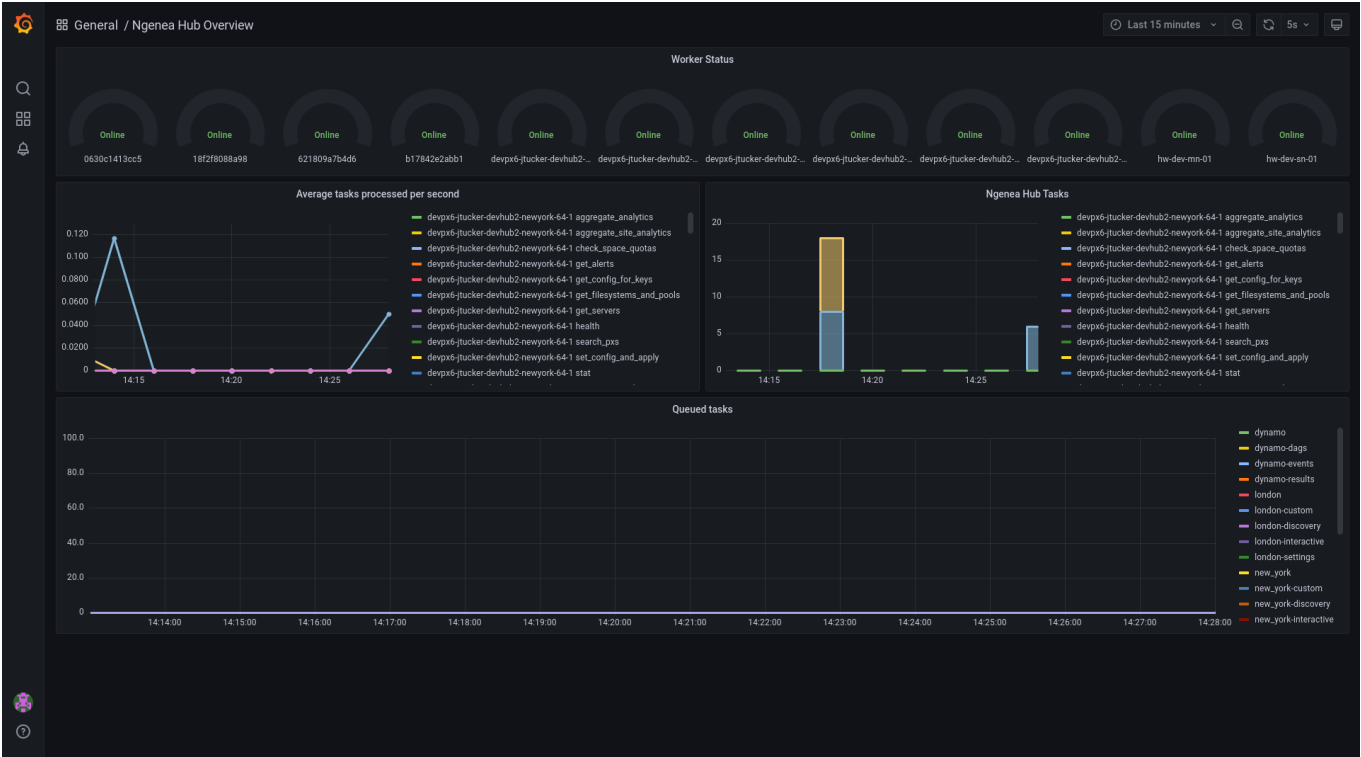
Container Name	Description
<code>ngeneahub_app_1</code>	Web application
<code>ngeneahub_jobrefresh_1</code>	Maintenance task controller
<code>ngeneahub_db_1</code>	Application database (Postgres)
<code>ngeneahub_redis_1</code>	Task results backend and celery broker

Hub Task Metrics

Hub provides a Grafana panel of the Tasks in the Hub queue and the status of the Workers.

To view the Metrics open the URL as follows, substituting the location of your hub.
`http(s)://myhub/hubmetrics`

It is required to be authenticated to Hub in order to view the metrics.



Health endpoints

To view the state of all sites and related nodes within known to Ngenea Hub, a GET request can be performed to `/api/health` to view all of the sites, nodes and the hub service itself. The states are currently based on how many nodes are online for each site using the following states:

State	Description
ok	All nodes are functional
warning	Some nodes are offline within a site
critical	One or more sites are completely offline

An example output of the health endpoint can be seen below:

```
```json {
 "overall_health": "ok", "hub_status": {
 "health": "ok"
 }, "site_status": [
 {
 "site": "site1", "health": "ok", "nodes": [
 {
```

```

 "name": "pixstor-east-ng-test", "health": "ok",
 "online": true
 }
]
}, {
 "site": "site2", "health": "ok", "nodes": [
 {
 "name": "pixstor-west-ng-test", "health":
 "ok", "online": true
 }
]
}
]
}

```

A request can also be performed to specific sites using `/api/sites/ID/health/` to view the site specific health status:

```

```json {
    "site": "site1", "health": "ok", "nodes": [
        {
            "name": "pixstor-east-ng-test", "health": "ok", "online": true
        }
    ]
}

```

Custom Workflows

Defining workflows

It's possible to define custom workflows which use pre-defined rules as building blocks to create your workflow.

Note: Custom workflows are not currently exposed via the UI. Use the API `/api/workflows/` endpoint to create custom workflows

A workflow definition requires the following parameters:

Name	Description
<code>name</code>	The unique name for this workflow. For easy of submission again the API, this should not contain spaces.
<code>label</code>	The human readable name for this workflow, can contain spaces.
<code>icon_classes</code>	List of icon classes to represent the workflow in the UI. Font Awesome is useful here.
<code>filter_rules</code>	A list of rules to apply to provided files that match defined states. Described in more detail below.
<code>fields</code>	A list of runtime fields. Described in more detail below.

Additionally, you can optionally provide:

Name	Description
<code>discovery</code>	Which discovery task the workflow should be used by default, this can be either recursive or snapdiff.
<code>discovery_options</code>	A json containing any additional options to pass to the workflow default discovery. Described in more detail below.

Filter Rules

Filter rules are defined in JSON. They are a list of individual rules in a mapping format that will be performed on each matching file result when a discovery task is complete. If called through the API with no discovery task provided, rules will be applied to any states provided in the workflow input.

Steps are defined in JSON. Steps is a list of individual steps that will be performed serially. Each rule must contain the following:

Name	Description	Required
<code>state</code>	The state of a result provided by the discovery task with any given path, an example of that could be "processed" or "modified" more details about this are in the discovery section.	Yes
<code>type</code>	The type of result the rule will apply to, the only valid types are: <code>file</code> <code>directory</code> <code>symlink</code> <code>all</code>	Yes
<code>action</code>		Yes

Name	Description	Required
	A list of tasks to perform on files that match the state and type	
<code>include</code>	A list of globs to apply to provided files to limit actions to just them.	No
<code>exclude</code>	A list of globs to apply to provided files. Described in more detail below.	No
<code>ignore_site_includes</code>	Whether to ignore any global includes defined on the site the workflow will run on	No
<code>ignore_site_excludes</code>	Whether to ignore any global excludes defined on the site the workflow will run on	No

These rules control which actions will be performed on certain files based on their given state that they have been given following specific discovery tasks such as `snappdiff` or provided in the initial input of a workflow. These states can allow direct control of workflows performed on files provided, allowing multiple workflow paths within the same job by utilizing multiple rules controlling specific states with additional control with include and exclude path rules.

Alongside rules bound to a state, there are two special states that rules can be used, these being `default` and `all`. Rule sets cannot have both `default` and `all` rules within them, but it is possible to have multiple of one type with different sets of exclude and include rules to allow for more granular control.

Rules defined with `default` as their `state` and `type` will perform their action on paths that have not been captured by all other rules within a given rule set. This means that if there are specific file states that need to be actioned differently, paths that do not match any other rules actioned against without ignoring those non-matching paths.

The other special rule type is rules with the `state` and `type` of `all`. This rule will perform its action on all paths regardless of their provided type and state. This is an additional operation so if another rule has an explicit rule provided it will perform multiple actions on the same path, for each matching rule in rule set. Simple workflows are typically composed of a single rule with the state and type of `all` as this will simply process all paths provided to it.

Within each rule, there must be a list of actions to perform on the resulting file provided within the `action` key. These actions will be performed serially. Each action must be a mapping that contain the following in each entry:

Name	Description	Required
<code>name</code>		Yes

Name	Description	Required
	The name of the task to run, e.g. <code>dynamo.tasks.migrate</code>	
site	The name of the site to run against, if this is not provided it will use the site provided within the workflow call.	No
queue	The name of the queue the task should run on. The queue must exist on the site the task will run against. If not provided it will use the queue provided in the workflow call, or the default queue if one is not provided	No

If steps have optional arguments, these can be passed as additional key:value pairs in these step definition mapping to pass those optional arguments.

As an example we can define a generic rule that captures every type of file and state and sends it to a second site, this would be useful for a bulk move using the `recursive` discovery task to cover all types of files in directories provided to the task:

Example 1 - Send to london

```
{
  "state": "all",
  "type": "all",
  "action": [
    {
      "name": "dynamo.tasks.migrate",
      "queue": "highpriority"
    },
    {
      "name": "dynamo.tasks.reverse_stub",
      "site": "london"
    }
  ]
}
```

Runtime fields

A workflow needs to be able to accept parameters as it submitted. Taking example #1 above, "london" doesn't want to be hardcoded as the destination site, as that would mean a new workflow would need to be defined for each possible destination.

Instead, fields can be defined, that in turn will need to be provided at workflow submission time. Fields are defined as a mapping with the following keys:

Name	Description	Required
name	The name of the field.	Yes
label	The friendly name for this field, used for presenting in the UI	Yes
type	The type of the field, valid options are: - string - a free text field - int - a free text field that will be validated a integer - bool - a checkbox - choices - A dropdown box representing a list of choices, populated from choices list of objects. - enum[enum_type] - A dropdown box representing a choice of option, populated from enum_type. enum_type can be one of the following - site - A list of all the sites Ngenea Hub has defined - queue - A list of all queues available on the selected site - list - a list of values of any scalar type	Yes
default	The default value for runtime fields	optional

The following is an example of a custom field definition for providing a site to an action step:

Example 2 - Custom field definition

```
[
  {
    "name": "target_site",
    "label": "Site to migrate to",
    "type": "enum[site]"
  }
]
```

```

    }
  ]
}

```

Custom field definition with default value

```

[
  {
    "name": "target_site",
    "label": "site to migrate to",
    "type": "enum[site]",
    "default": "london"
  }
]

```

If default value is specified in runtime fields, it will take the default value for fields while running workflow if the user input is not given otherwise it will always use the user input.

Back in the definition of an action step, any value that is prefixed with a `*` will be used as a field name and the value replaced instead of a literal string.

The following example, modifies example #1 to use the custom field as defined in example #3:

Example 3 - Updated rule now using custom fields

```

{
  "state": "all",
  "type": "all",
  "action": [
    {
      "name": "dynamo.tasks.migrate"
    },
    {
      "name": "dynamo.tasks.reverse_stub",
      "site": "*target_site"
    }
  ]
}

```

So, a complete request to create a workflow that will process all file and state types with a dynamic "site" field will look like:

Example 4 - Full workflow request

```

{
  "name": "send_file",
  "label": "Send files from one site to another",
  "icon_classes": ["fa fa-cloud fa-stack-2x text-primary", "fa fa-refresh fa-stack-1x text-light"],
  "filter_rules": [
    {
      "state": "all",
      "type": "all",
      "action": [

```

```

        {
            "name": "dynamo.tasks.migrate"
        },
        {
            "name": "dynamo.tasks.reverse_stub",
            "site": "*target_site"
        }
    ]
},
],
"fields": [
    {
        "name": "target_site",
        "label": "Site to migrate to",
        "type": "enum[site]"
    }
]
}

```

The following is an example of a custom field definition for providing a choices to an action step:

Example 5 - Custom field definition

```

[
    {
        "name": "sync_policy",
        "label": "sync_policy",
        "type": "choices",
        "choices": [
            {
                "label": "Newest",
                "value": "newest"
            },
            {
                "label": "Sourcesite",
                "value": "sourcesite"
            }
        ]
    }
]

```

choices support both string and integer type values.

Back in the definition of an action step, any value that is prefixed with a `*` will be used as a field name and the value replaced instead of a literal string.

The following example, uses the custom field in action:

Example 6 - Updated rule now using custom fields

```

{
    "state": "all",
    "type": "all",
    "action": [

```

```

    {
      "name": "dynamo.tasks.migrate"
    },
    {
      "name": "dynamo.tasks.reverse_stub",
      "sync_policy": "*sync_policy"
    }
  ]
}

```

So, a complete request to create a workflow that will process all file and state types with a static choices field will look like:

Example 7 - Full workflow request

```

{
  "name": "send_file",
  "label": "Send files from one site to another",
  "icon":
    "<span class='fa-stack'><i class='fa fa-cloud fa-stack-2x text-
    primary'></i><i class='fa fa-angle-right fa-stack-2x text-light'></
    i></span>"
  "filter_rules": [
    {
      "state": "all",
      "type": "all",
      "action": [
        {
          "name": "dynamo.tasks.migrate"
        },
        {
          "name": "dynamo.tasks.reverse_stub",
          "sync_policy": "*sync_policy"
        }
      ]
    }
  ],
  "fields": [
    {
      "name": "sync_policy",
      "label": "sync_policy",
      "type": "choices",
      "choices": [
        {
          "label": "Newest",
          "value": "newest"
        },
        {
          "label": "Sourcesite",
          "value": "sourcesite"
        }
      ]
    }
  ]
}

```

```
]
}
```

Running Workflows

Once a workflow has been defined, it can be performed through the file browser by selecting files and directories and clicking the actions button. It is then possible to select the workflow you wish to call, this workflow call will not use a discovery task unless a directory is selected, in that case it will make use of the recursive discovery step.

This can also be performed via a POST request to `/api/file/workflow`. When called through the API, you have the option to provide a discovery step, these steps can expand the initial paths provided to them to either recursively perform actions or perform something like a file difference scan.

Name	Description	Type	Required
paths	A list of paths to perform the workflows against, these can be just strings of file absolute file paths or can be JSON with the keys of "path" and "state", detailed example in example 7	JSON List	Yes
site	The site to perform the workflow against	String	Yes
queue	The queue to run workflow tasks on. The default queue will be used if not provided.	String	None
fields	The runtime fields for a workflow	String	Yes
discovery	The discovery phase to use for this workflow run, this will override any defaults	String	No
job	The ID of a job that this workflow should be run within	Integer	No

Following the example workflow defined above, you can call the workflow to recursively send all files within any paths provided using the following POST to `/api/file/workflow`:

Example 8 - Calling example workflow

```
{
  "paths": [
    "/mmfs1/data/project_one",
    "/mmfs1/data/project_two"
  ],
  "site": "london",
  "queue": "highpriority",
  "workflow": "send_file",
  "discovery": "recursive",
  "fields": {
    "target_site": "dublin",
  }
}
```

This will now migrate all files within `/mmfs1/data/project_one` and `/mmfs1/data/project_two` and then recall them at the site defined as `dublin`.

If there is a more complex workflow that have been defined that includes rules for specific states, the input paths can include this state information. This behaviour can be only be used when no discovery state is provided, an example of a custom rule set using could be:

Example 9 - Calling workflow with state data

```
{
  "name": "migrate_state",
  "label": "Stateful file migration",
  "filter_rules": [
    {
      "type": "all",
      "state": "modified",
      "action": {
        "name": "dynamo.tasks.migrate"
      }
    },
    {
      "type": "all",
      "state": "moved",
      "action": {
        "name": "dynamo.tasks.delete_paths_from_gpfs"
      }
    }
  ],
  "discovery": null,
  "fields": []
}
```

Here is a simple rule set that will migrate all paths provided with the state modified and will delete all paths provided with the state moved. With this example workflow provided you can perform a POST to `/api/file/workflow` with the following JSON:

Example 10 - Calling workflow with state data

```
{
  "paths": [
    {
      "path": "/mmfs1/data/project_one",
      "state": "modified"
    },
    {
      "path": "/mmfs1/data/project_two",
      "state": "moved"
    }
  ],
  "site": "london",
  "workflow": "migrate_state",
  "discovery": null,
  "fields": {}
}
```

Using multiple state based rules with different include and exclude path filters, you could achieve more complex behaviour in workflow calls for more finite control.

Discovery Steps

Discovery steps can make complex large bulk operations much more manageable to call, allowing you to provide a single path that expands to cover all the contents of a path, or to see time based differences for a given path.

Note: If a workflow is submitted without a discovery task explicitly provided, it will default to using the discovery task defined as the default during the workflow's creation, visible via the workflow's "discovery" attribute. To avoid this, it is possible to explicitly pass `null` as the discovery task via the API to skip any discovery phase and additional processing on the paths provided and instead process the actions specified using the rules, without any additional checks.

Name	Description	Supported states
recursive	Performs a recursive expansion of the initial provided paths. This allows paths to be expanded to cover all sub file and directories, it will then perform the defined action for all the generic rules in a workflow against all resulting files.	all
snapdiff	Performs a time based file scan on an independant fileset between the last	created updated moved deleted all

Name	Description	Supported states
	time a scan was performed. It will retrieve all file differences between those moments in time and the state of that file.	

For more complex discovery steps such as `snapdiff`, there are defined states that files, directories and links can be in once it has completed its scan. This allows more explicit control of file and state control within a single call to a workflow. If for example you want all results with the type of file that have the state `created` to be sent to another site without any temporary files, a rule to cover that could be:

Example 11 - Custom rule for filtering snapdiff discovery results

```
{
  "state": "created",
  "type": "file",
  "exclude": ["*.temp"],
  "action": [
    {
      "name": "dynamo.tasks.migrate"
    },
    {
      "name": "dynamo.tasks.reverse_stub",
      "site": "*target_site"
    }
  ]
}
```

Discovery Options

Additional options can be passed to the discovery task by setting `discovery_options` on the workflow. The supported options are those described in [Discovery Steps](#).

Example 12 - Passing options to the discovery

```
{
  "discovery": "recursive",
  "discovery_options": {
    "skip_missing": false
  }
}
```

Discovery options will be used with the workflow default discovery only. If a different discovery is set at runtime, the discovery options will be ignored. Discovery options cannot currently be overwritten at runtime.

Includes and Excludes

Includes and excludes can be used to select paths that individual filter rules should apply to, or generally limit which paths should be handled during a workflow run.

Include/exclude patterns behave like unix shell pattern matching ('globbing'). The 'wildcard' asterisk character `*` will match any characters within a string. Patterns must match whole paths; partial matches are not supported, except through the use of wildcards.

Includes and excludes are combined as "a path matching any includes and not any excludes". For example `{"include": ["/mmfs1/data/*"], "exclude": ["*.tmp"]}` would match only files in `/mmfs1/data`, but not files in that directory with the `.tmp` extension. If no includes are defined, then all files are considered included (unless explicitly excluded).

There are three places where path include and exclude patterns can be defined:

- on a site
- within a filter rule
- at runtime, when a workflow is submitted

Site patterns can be used to apply includes and excludes globally, to all workflows and workflow steps. If defined, these will be appended to any patterns defined within a filter rule or at runtime.

For example, if a rule defines `{"exclude": ["*.tmp"]}` and the site defines `{"exclude": ["*.cache"]}`, then the combined excludes for that rule would be `{"exclude": ["*.tmp", "*.cache"]}`

If not desired, this behaviour can be overridden by specifying `ignore_site_includes` / `ignore_site_excludes` either on a per-rule basis, or for all rules by passing those parameters when submitting a workflow run.

For workflows involving multiple sites, such as `send` and `sync`, only the primary (source) site patterns will be considered.

If includes and excludes are passed when submitting a workflow, they will be applied to all filter rules within the workflow, replacing any patterns already defined within the workflow rules. Any site patterns will be appended to the runtime patterns, unless 'ignore' is specified.

Jobs

Housekeeping

Job details are kept for a time after job completion. During this time, they can be viewed in the UI.

But older jobs are periodically culled from the database, to keep it to a manageable size. The housekeeping process runs once a day, and removes results for any job

that completed more than 90 days ago (by default). A different 'time-to-live' (TTL) can be set using the `jobs_ttl` configuration - see [Configuration](#) for more information.

Site Sync

Ngeneia Hub provides the facility for syncing data from one site to another.

Site Sync applies changes in one direction. See also [Bidirectional Site Sync](#) for applying changes in both directions.

Synchronisation is achieved by utilising a scheduled workflow which periodically discovers file and directory changes within an Independent Fileset on a source site and applying those changes to a target site. Changes are applied by sending newly created or recently modified files and directories, including deleting or moving files or directories in place on the target site as necessary to match the source site.

The following walkthrough details how to set up a sync between two sites ('Site Sync') using the Ngeneia Hub UI.

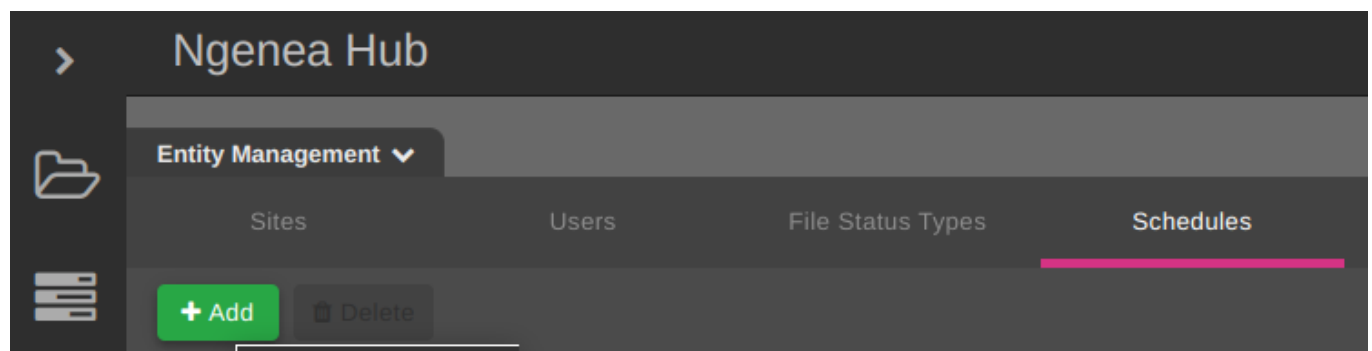
The [REST API](#) can also be utilised to achieve the same setup.

Schedule

A schedule defines when and how often a sync occurs.

In the Ngeneia Hub UI, navigate to the `Administration` page, and select the `Schedules` tab.

To create a new schedule, click the `Add` button.



Configure the schedule by entering appropriate details into the `Add new schedule` dialog:

The screenshot shows the 'Add new schedule' form. It has a purple header button 'Add new schedule'. The form fields are: Name (site-from-site1), Site (site1), Paths (a list with one item: /mmfs1/data/test), Discovery (snapdiff), Day of week (*), Day of month (*), Month of year (*), Hour (*), and Minute (*/*). There is a checkbox for 'Enable the schedule' and 'Add' and 'Cancel' buttons at the bottom.

1. Enter a descriptive **Name** for the schedule, E.G. "sync-to-site2"
2. Select the **source** site from the dropdown menu
3. Set the **Discovery** to **snapdiff**. Snapdiff discovery uses snapshots to track changes over time within an Independent Fileset.
4. Set the **Path** to the top level directory (path) of the Independent Fileset to be synchronised
5. Set the time criteria
6. Click the **Add** button to add the new schedule

Time criteria

A schedule utilises **cron** syntax. By default, all the fields are populated with ***** - this means the schedule will run every minute. This is likely too frequent as the calculation of file changes could take longer than the time between the schedule running every minute. Sizing for an ideal setup will balance the need to sync files quickly versus how long the sync snapdiff stage takes to run. This can be most effectively achieved through observation.

Examples:

- To enact the schedule every 5 minutes, specify the **Minute** field as follows: ***/5**.
- To enact the schedule once an hour, set the **Minute** field to a number of minutes past the hour. Setting **Minute** to **5** will run the sync once an hour at 5 minutes past the hour.
- To enact the schedule at set times throughout the hour, change the **Minute** field separating entries by commas (E.G.) **7,23,46**

Workflow

Schedules enact a workflow. A workflow performs one or more series of tasks to files and/or directories.

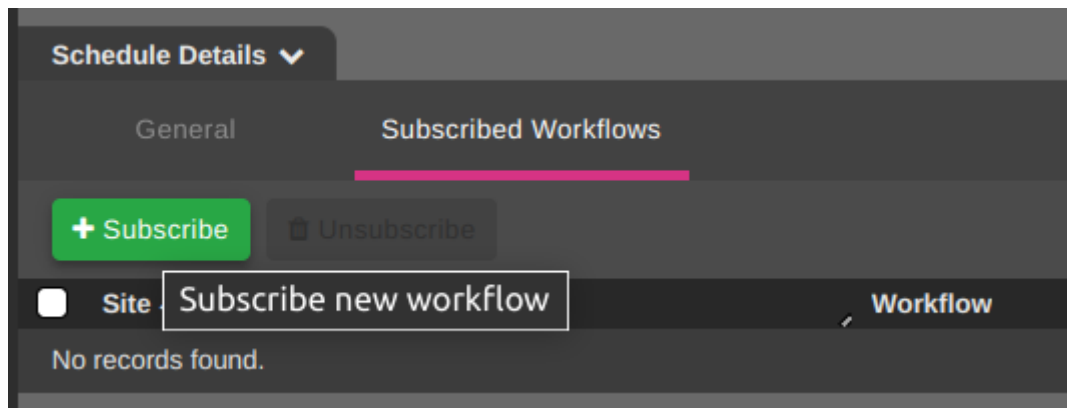
In the Ngenea Hub UI, navigate to the **Administration** page, and select the Schedules tab.

Select the prior created schedule in the **Schedules** tab, which then displays the **Schedule Details** page.

Subscribing a workflow

To associate a workflow with a schedule undertake the following actions:

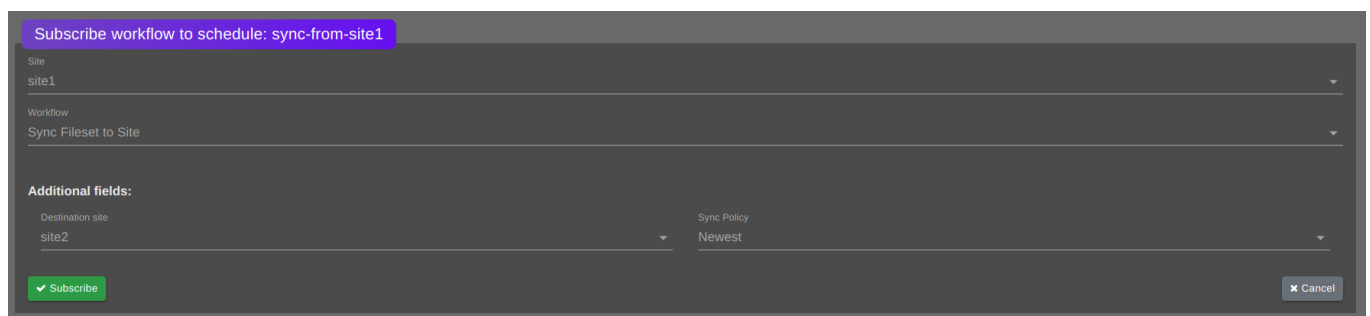
1. Click **Subscribe** and in the create dialogue set **Site** to the source site from which to synchronise.



Ngene Hub provides a built-in workflow for synchronising changes from an Independent Fileset on a source site to a target site.

1. Select **Sync Fileset to Site** from the **Workflow** dropdown menu.

Configure the workflow subscription settings by entering appropriate details into the **Subscribe workflow to schedule** dialog:



1. Select the target site in the **Destination site**
2. Select an appropriate **Sync Policy**
3. Click **Subscribe** to associate the workflow to the schedule

Sync Policy

The **Sync Policy** setting controls how conflicts are resolved. A conflict may occur if (E.G.) different versions of the same file exist on both source and target sites. By default, the newest version of the file will be retained.

Outcome

Now associated (subscribed), files and/or directories will synchronise at the next scheduled execution time. When executing the scheduled workflow, two jobs appear within the schedule details page. One for the snapdiff discovery, and one for the subscribed sync workflow.

The first enactment of the sync will synchronise all files and directories in the Independent Fileset from the source site. Further enactments will synchronise files

and/or directories created, modified, deleted or moved since the last enactment and the time of the most recent enactment.

Additional Options

Associating more than one workflow to the same schedule can be utilised to synchronise from the same source site to another target site. Additional workflows cause creation of an additional job, and run simultaneously with other syncs enacted from the same schedule.

Troubleshooting

If you encounter issues with sync, refer to the [sync troubleshooting guide](#) page for guidance.

Bidirectional Site Sync

Ngenea Hub provides the facility for syncing data between sites.

Bidirectional Site Sync is similar to [Site Sync](#), but applies changes in both directions. If a Bidirectional Site Sync runs between sites A and B, changes on site A will be applied on site B, followed by changes on site B being applied on site A.

Synchronisation is achieved by utilising a scheduled workflow which periodically discovers file and directory changes within an Independent Fileset on each of the sites and applies those changes to the other site. Changes are applied by sending newly created or recently modified files and directories, including deleting or moving files or directories in place on the target site as necessary to match the other site.

A prerequisite for a working Bidirectional Site Sync is that the Independent Fileset has been created on both sites.

The principles of setting up Bidirectional Site Sync are similar (but with some differences explained below) to setting up a Site Sync, so please refer to the walkthrough on the [Site Sync page](#) for detailed instructions. It can be done using the Ngenea Hub UI or the [REST API](#).

Schedule

Discovery

Create a new schedule. Unlike with Site Sync, for Bidirectional Site Sync set the schedule's `Discovery` to `bidirectional_snapdiff`.

This discovery type operates identically to the `snapdiff` discovery type, with snapshots used to track changes over time within an Independent Fileset.

A separation is enforced between `bidirectional_snapdiff` schedules and schedules with other discovery types. Whereas the `Discovery` of other schedules may be changed, a change from or to the `bidirectional_snapdiff` discovery type

is not allowed. This is to ensure consistency of the specific workflow rules that apply to Bidirectional Site Sync.

Time criteria

Unlike with Site Sync, a Bidirectional Site Sync will not be triggered if the previous sync is still running. This means that some syncs will be skipped if scheduled closer together than the time it takes to complete a sync.

Workflow

Subscribing a workflow

Select the `Bi-directional Site Sync` workflow.

This is the only workflow that may be subscribed to a `bidirectional_snapdiff` schedule, and it cannot be subscribed to other types of schedule.

A `bidirectional_snapdiff` schedule may have at most one subscribed workflow.

Outcome

When executing the scheduled workflow, only one job is created and appears within the schedule details page. This job includes the tasks for both the snapdiff discoveries (one on each site), as well as the tasks that enact the sync in each direction.

Troubleshooting

If you encounter issues with sync, refer to the [sync troubleshooting guide](#) page for guidance.

Search

Note: Before you can use the search feature, additional set-up is necessary, as described in the [search feature](#) page.

The search endpoint provides the ability to search for files across multiple sites, and aggregate the results.

Search is performed in two steps - submitting a query, and retrieving the results.

Submitting a query

Search performs a query by submitting asynchronous tasks to each requested site. The sites then perform the actual search and return results as available.

A search is initiated by `POST`ing a query to the search endpoint

```
curl -s -X POST 'http://example.com/api/search/' -H 'Accept: application/json' -H 'Content-Type: application/json' -H
```

```
"Authorization: Api-Key $TOKEN" -d
'{"path": "/mmfs1/data", "sites": ["site1"], "recursive": true,
"filters": {"hsm.status": "migrated"}}'
```

The search request payload is made of

name	description
path	Directory to query
sites	List of one or more sites to search. Default: all sites
recursive	Whether to search the path recursively. Default: false, only immediate children of path will be returned.
filters	A collection of filters against arbitrary metadata, see below. Default: None
metadata_fields	A list of metadata fields to include in search results. This can include specific field names (e.g. hsm.status), or namespace wildcards (e.g. core.*) to select all fields in a given namespace. Default: all available fields.
merge	If the same file exists on multiple site, this will cause them to be merged in the results (see below). Default: false

Upon successful submission, the request will return status 201 (Created), and a response body which includes the url for retrieving search results (see below)

```
{"id":1,"url":"http://example.com/api/search/1/"}
```

Filters

Filters are a collection of filters to apply to arbitrary file metadata.

The specific metadata available to be filtered on depends on the search backend being used. The fields in the following examples may not be available for all backends.

At a minimum, one can expect to be able to filter on core.filename, the file basename. For example to filter only jpeg files, {"core.filename": "*.jpg"}

Possible filter types are

type	description	example
exact match	match a value exactly	{"core.filename": "cats-01.jpg"}, {"core.size": 0}
match list	match any of the values in the list (value1 OR value2 OR ...)	{"core.group.name": ["editor", "admin"]}
wildcard		{"core.filename", "*.jpg"}

type	description	example
	any string value containing an asterisk (*) is treated as a wildcard	
range	numerical or date range, using any combination of less-than (lt), less-than-equal (lte), greater-than (gt), greater-than-equal (gte)	<code>{"core.modificationtime": {"gte": "2021-01-01", "lt": "2021-02-01"}}, {"core.size": {"gt": 100000, "lte": 200000}}</code>
negation	exclude anything matching a given filter	<code>{"not": {"core.filename": ".DS_Store"}}</code>

Filters are combined as AND, e.g. `{"core.extension": ".jpg", "hsm.status": "migrated"}` matches .jpg files which are HSM migrated.

Retrieving results

When search results are read, they can be retrieved using the url returned when the query was submitted.

```
$ curl 'http://example.com/api/search/1/' -H "Authorization: Api-Key $TOKEN"
{
  "count": 1,
  "next": null,
  "previous": null,
  "items": [
    {
      "href": "http://example.com/api/file/?path=%2Fmmfs1%2Fdata%2Fhello.txt&site=site1",
      "site": "site1",
      "path": "/mmfs1/data",
      "name": "hello.txt",
      "metadata": {
        "core.accesstime": "2021-10-12T16:27:28",
        "core.changetime": "2021-10-12T16:28:45",
        "core.directory": "/mmfs1/data",
        "core.extension": ".txt",
        "core.filename": "hello.txt",
        "core.group.id": 0,
        "core.group.name": "root",
        "core.hash.sha512": "db3974a97...94d2434a593",
        "core.modificationtime": "2021-10-12T16:28:45",
        "core.pathname": "/mmfs1/data/hello.txt",
        "core.size": 12,
        "core.user.id": 0,
        "core.user.name": "root",
        "gpfs.filesetname": "root",
        "gpfs.filesystem": "mmfs1",

```

```

        "gpfs.kballocated" : 0,
        "gpfs.poolname" : "sas1",
        "hsm.status" : "migrated"
        "ngenea.pathname" : "data/hello.txt",
        "ngenea.size" : 12,
        "ngenea.target" : "awss3",
        "ngenea.uuid": "acf1a307-5b6a-43b0-8fb2-d2b366e88008",
    },
    "proxies": {
        "_thumbnail": "/media/123/456/789/123456789012345.png"
    }
},
"metadata_fields": ["core.accesstime", ...],
"complete": true,
"errors": {"site2": "Search backend is offline"}
}

```

Results from different sites may not arrive at the same time. The `complete` field indicates whether all sites what returned their results. This includes when a site returns with an error.

Results from different sites are 'concatenated', meaning if the same file exists on multiple sites, there will be separate result items for the file for each site.

The `metadata` field on each item contains arbitrary file metadata. The specific metadata will vary depending on the search backend being used. In the case of the PixStor Search backend, the available fields will vary depending on file type, and which plugins were used when the files were ingested.

If `metadata_fields` was specified when the query was submitted, the `metadata_fields` entry in the response will match, with any wildcards expanded to list the available fields which match those wildcards. Otherwise, the `metadata_fields` entry will list all the available metadata fields which could be returned from the search backend. Individual files may not have all the listed fields.

All search backends format results to be namespaced, similar to PixStor Search, for consistency.

If an error occurs while performing the search on any of the sites, the `errors` entries will provide a mapping of site names and error messages.

Proxies

The `proxies` field on each item contains links to proxies of the file, such as thumbnails and previews.

These will only be available if the backend is `pixstor_search`, and PixStor Search itself has been configured and ingested in 'search plus' mode.

The links are relative urls, which must be combined with the `public_url` from the corresponding `/sites` endpoint to fetch the actual proxy files, e.g. `https://pixstor-sitel/media/....`

Typical proxies may include:

- `_thumbnail` - a 150x150 px thumbnail image
- `image.preview` - a 400x300 px preview image
- `video.preview` - a 400x300 px, lower resolution version of a video
- `audio.preview` - a downscaled version of an audio file

Proxies may not be available for all file types, depending on which search plugins have been enabled and ingested.

Parameters

Search results are paginated. The following parameters can be used to control what results are returned

name	description
<code>page</code>	Numbered page of results to fetch. Default: 1
<code>page_size</code>	Maximum number of results to return per page. Default: 20
<code>sort</code>	One or more fields to sort results on, separated by commas, e.g. <code>?sort=name,site</code> . Field names can be prefixed with <code>-</code> to reverse order. For fields in <code>metadata</code> , the field name is specified as is, e.g. <code>?sort=-core.accesstime</code> . Default: arbitrary order.
<code>group_by_name</code>	When this url parameter is 'true' results are merged based on matching name, e.g. <code>?group_by_name=true</code>

Merged results

When a search is submitted with `"merge": true`, the search results will be 'merged'.

This means that entries for matching files from different sites will be combine. An entry is considered to be matching if it has the same full path.

```
$ curl 'http://example.com/api/search/2/' -H "Authorization: Api-Key $TOKEN"
{
  "count": 1,
  "next": null,
  "previous": null,
  "items": [
    {
      "path": "/mmfs1/data",
      "name": "hello.txt",
      "metadata": {
        "core.accesstime": "2021-10-12T16:27:28",
```

```

    "core.changetime" : "2021-10-12T16:28:45",
    "core.directory" : "/mmfs1/data",
    "core.extension" : ".txt",
    "core.filename" : "hello.txt",
    "core.group.id" : 0,
    "core.group.name" : "root",
    "core.hash.sha512" : "db3974a97...94d2434a593",
    "core.modificationtime" : "2021-10-12T16:28:45",
    "core.pathname" : "/mmfs1/data/hello.txt",
    "core.size" : 12,
    "core.user.id" : 0,
    "core.user.name" : "root",
    "gpfs.filesetname" : "root",
    "gpfs.filesystem" : "mmfs1",
    "gpfs.kballocated" : 0,
    "gpfs.poolname" : "sas1",
    "hsm.status" : "migrated"
    "ngenea.pathname" : "data/hello.txt",
    "ngenea.size" : 12,
    "ngenea.target" : "awss3",
    "ngenea.uuid" : "acfla307-5b6a-43b0-8fb2-d2b366e88008",
  },
  "status": {
    "site1": true,
    "site2": false
  }
},
],
"metadata_fields": ["core.accesstime", ...],
"complete": true
}

```

Merged results no longer have the `site` and `href` fields. In their place is a `status` field, which maps sites to whether the file is 'resident' on that site.

A file is considered resident if the file is not migrated, or is premigrated ('hydrated'). A file is considered not resident if the file is migrated (stubbed), or not present at all.

Results grouped by name

To group search results by name, add the url parameter `group_by_name=true`.

This will return results from any of the requested sites. Results within a group may belong to different directories.

```

$ curl 'http://example.com/api/search/1/?group_by_name=true' -H
"Authorization: Api-Key $TOKEN"
{
  "count": 60,
  "next": "http://testserver/api/search/1/?
group_by_name=true&page=2",
  "previous": "http://testserver/api/search/1/?group_by_name=true",
  "more_results": false,

```

```

"items": [
  {"test-10": [
    {
      "site": "testsite",
      "path": "/mmfs1/data",
      "metadata": {...}
    },
    {
      "site": "anothersite",
      "metadata": {...}
    }
  ]
},
{"test-11": [
  {
    "site": "testsite",
    "path": "/mmfs1/data",
    "metadata": {...}
  },
  {
    "site": "anothersite",
    "path": "/mmfs1/archive",
    "metadata": {...}
  }
]
},
{"test-19": [
  {
    "site": "testsite",
    "path": "/mmfs1/data",
    "metadata": {...}
  },
  {
    "site": "anothersite",
    "path": "/mmfs1/data",
    "metadata": {...}
  },
  ...
]
},
"complete": true,
"metadata_fields": null,
"errors": {}
}

```

Max Results

There is a hard limit on the number of results returned, per site. By default, each site will return, at most, 200 results.

Fetching a lot of results makes queries slower and, since results are stored in the DB, storing more results uses more space. On the other hand, the limiting may lead to some matches not being returned.

The maximum number of results per site is controlled by the `search_max_results` configuration - see [Configuration](#) for more info.

Result limiting is applied when the search query is submitted, not when results are retrieved. If you change `search_max_results`, you will need to resubmit your query to fetch any additional matches.

Note, some backends have a hard limit of 10,000 results.

Housekeeping

The results from a query are stored, so they can be retrieved multiple times without performing a new query.

However, over time, the files on each site will change, and the stored results may no longer accurately reflect the active file system.

Therefore, old results are periodically culled. The housekeeping process runs once a day, and removes results for any search which was submitted more than a week ago (by default). A different 'time-to-live' (TTL) can be set using the `search_result_ttl` configuration - see [Configuration](#) for more information.

Results can also be manually removed by performing a DELETE request against the given search result endpoint

```
curl -X DELETE 'http://example.com/api/search/1/' -H "Authorization: Api-Key $TOKEN"
```

Controlling Bandwidth Usage

Ngenea Hub can control the amount of traffic speed for each node within each site that processes files through Ngenea. This will limit both outgoing and incoming traffic with defined cloud services.

Enabling the bandwidth feature in the UI

Regardless of enabling this feature, it is possible to change the bandwidth via REST and will enable changing of bandwidth when the Site instance has its `bandwidth` attribute changed.

This can be enabled by enabling the feature flag, details on how to do this can be found on the [Feature Flags](#) page.

This will enable the UI within the Site details page and represents the bandwidth limit in Mb/s.

Checking Node status

Each site within Ngenea Hub is the collection of all nodes running an instance of Ngenea Worker using the name of a given site. These are nodes within a cluster that are collectively listening to the same queue for a given site. Each time any worker comes online on a new node or a known node, this is tracked within Ngenea Hub using `Node` objects. These are automatically created when first starting up a worker, after creation their online status can be monitored.

You can view the nodes for each site within `/api/nodes/` this will be a complete list of all nodes known to Ngenea Hub.

For bandwidth control, there will need to be existing nodes known to Ngenea Hub, otherwise the bandwidth rules cannot be applied. If your worker coming online was not tracked due to timing issues, you can manually scan for existing nodes using one of the [Actions](#).

Registering Datastores

In order to control the bandwidth for all nodes under a site, the site will need to have the Ngenea policy targets defined as `Datastore` instances within Ngenea Hub.

The following example is for defining an AWS S3 target for Ngenea within Ngenea Hub:

```
{
  "name": "site1_amazon",
  "type": "S3",
  "bucket": "bucket01",
  "secretaccesskey": "secret-key",
  "accesskey": "access-key"
}
```

With this established datastore, all that is left to do is link the created datastore to a site so that when a change in bandwidth is applied, the site knows what service to limit the traffic to.

Cloud IPAddresses

Each datastore that points to a cloud target will have access to a list of all known IP addresses that are associated with that specific service. This will be used to limit all traffic between those IP ranges and the nodes currently running a worker instance when a bandwidth limit is applied.

These IP ranges are updated internally once a day in a scheduled task.

Using manual IP addresses

Manual IP addresses can be used to override the list of cloud related IPs, this can be useful to control bandwidth to custom endpoint targets such as services like minio making use of `POST api/ipaddresses/`. Using the following example to add an address to the list of IPAddresses:

Note: This will disable the use of all cloud addresses for a datastore and will instead only use the custom IP addresses.

```
{
  "ipaddr": "208.65.153.237",
  "datastore": 1
}
```

This will ensure that all traffic between those nodes and the defined IP addresses will be limited to the max bandwidth limit

Linking a Datastore to a Site

With our example site `site1` creating a `SiteLink` between `site1` and the datastore `site1_amazon` allows the bandwidth to be applied to all S3 targets on all nodes that are running the Ngenea Worker service.

For this example all local data for this Ngenea is located in `/mmfs1/data/aws_data` with data being placed in the bucket `bucket01` under `aws_data/`.

The following `SiteLink` can be used to represent this:

```
{
  "site": "site1",
  "datastore": "site1_amazon",
  "site_path": "/mmfs1/data/aws_data/",
  "datastore_path": "aws_data/"
}
```

This will ensure that each `Node` under `site1` will limit its traffic to all related addresses.

Applying the bandwidth

Note: This will effect all traffic on each node running a worker to the cloud services defined with the sites linked datastores.

With the `SiteLink` in place, changing the bandwidth attribute to the Mb/s desired on the site instance via the API route `PATCH /api/site/{id}`:

```
{
  "bandwidth": 1000
}
```

Using this or the UI, this will cause the hub to signal the worker to limit traffic using the IP ranges defined for the datastores.

This total bandwidth will be divided between all nodes for any given site, so if a site has a bandwidth of 1Gb/s then both nodes will be limited to 500mb/s.

Configuration

Global Configurations

Some configurations are stored in the Ngenea Hub configuration file, as described in [Hub Configuration](#). These are generally static, or sensitive settings. Changes to these settings require a service restart.

In addition, there are configurations which can be changed on-the-fly, typically to change Ngenea Hub behaviour. These settings can be viewed and changed via the REST API, as described below.

Available Settings

Name	Description	Default
jobs_ttl	How long job details should be stored after job completion, in days.	90
search_backend	Backend to use when performing searches. Currently supported backends: <code>analytics</code> , <code>pixstor_search</code> .	<code>pixstor_se</code>
search_result_ttl	How long search results should be stored, in days.	7
search_max_results	Maximum number of search results to fetch from the search backend, per site. Fetching more results will make queries slower and will require more storage space. Fetching fewer results may lead to some files being missing. Note, some backends have a hard limit of 10,000 results.	200
snapshot_create_delete_retry_timeout	Time in seconds after which workers will give up retrying snapshot create and delete operations. These occur in the <code>dynamo.tasks.snapdiff</code> and <code>dynamo.tasks.rotate_snapshot</code> tasks. The default 1000s will give 4 or more retries. Set to 0 to disable retries.	1000
stat_timeout	How long to wait for results from the <code>/api/file/</code> endpoint, in seconds.	10
stat_refresh_period	How long a files details will be retained as a cache within the file browser, in seconds.	10
task_invalidation_timeout	How long in minutes for a task in the STARTED state will wait before being invalidated.	360
snapdiff_stream_timeout		10

Name	Description	Default
	Idle timeout when waiting for delete and move tasks to complete during snapdiff workflows, in minutes.	
force_local_managed_users	Allow local NAS users to be managed on AD joined sites.	False
custom_statistics_tasks	A list containing the names of custom tasks that add their number of reported files to the job total, it will do so regardless of when these tasks are executed within a workflow.	["dynamo.c "dynamo.cu
maximum_external_results	Maximum number of items to retrieve per page from an external target scan.	100
node_health_last_heartbeat_interval	The maximum allowed time (in seconds) between the last recorded heartbeat and the current time for a node to be considered online. If the node's last heartbeat occurred more than this interval ago, the node will be marked as offline.	120
initial_uid	Minimum uid value for the nas user that can be configured	100000
initial_gid	Minimum gid value for the nas group that can be configured	100000

Note: If the file details are not returned for large fileset, increase the default value of `stat_refresh_period` in the configuration tab of hub admin page <http://admin> or do a patch on configuration API.

REST API

Configurations can be listed and set via the Ngenea Hub REST API.

Note: The configurations endpoint does not support client key authentication. You must use [JWT Authentication](#).

To list the current configuration settings,

```
$ curl -s 'http://example.com/api/configurations/' -H 'Accept: application/json' -H "Authorization: Bearer $JWT_ACCESS_TOKEN"
{
  "search_backend": "analytics",
  "search_max_results": 200,
  "search_result_ttl": 7,
  ...
}
```

To change one or more configuration settings, make a `PATCH` request the same endpoint

```
$ curl -s -X PATCH 'http://example.com/api/configurations/' -H
'Accept: application/json' -H "Authorization: Bearer
$JWT_ACCESS_TOKEN" -H 'Content-Type: application/json' -d
'{"search_max_results": 500}'
{
  "search_max_results": 500,
  ...
}
```

Settings Migration

In versions 1.9.0 and earlier, some of the above settings were configured via the Ngenea Hub config file ([Hub Configuration](#)).

Upon updating to version 1.10.0 or above, any values currently set in that config file will be captured. Thereafter, any changes to those settings within the config file will be ignored.

Site-specific Configurations

Some configuration options can be set on a per-site level, and may differ between sites.

These can be viewed and changed via the REST API, as described below. They can also be viewed and changed in the Ngenea Hub UI, from the 'Sites' tab on the Administration page.

Available Settings

Name	Description	Default
bandwidth	Limit the bandwidth for the site (In MB/s). In the UI, it is hidden behind the <code>bandwidth_controls</code> feature flag (see Feature Flags)	not set (unlimited)
elasticsearch_url	URL used to interact with Elasticsearch when <code>search_backend</code> is set to <code>analytics</code> (see Global Configurations above). The URL is evaluated on the node(s) on which the site worker is running.	localhost: 9200
pixstor_search_url	URL used to interact with PixStor Search when <code>search_backend</code> is set to <code>pixstor_search</code> (see Global Configurations above). The URL is evaluated on the node(s) on which the site worker is running.	https:// localhost/
public_url	Public URL that can be used to reach this site. Typically this will be the	not set

Name	Description	Default
	hostname or external IP address of the site management node.	
file_batch_gb	Limit the total size of file data in a batch, in gigabytes. See File 1 Batching below	
file_batch_size	Limit the total number of files in a batch. See File Batching below	40
enable_auto_file_batch_sizing	If Dynamic file batching should be enabled for this site. See Dynamic File Batching below	True
lock_threshold	The snapdiff discovery uses locking to prevent multiple snapdiff running against the same fileset at once. To prevent stale locks, lock are considered 'expired' after the lock_threshold, given in seconds.	86400 (one day)
include	A list of include glob patterns which will apply to all workflows run against this site	not set
exclude	A list of exclude glob patterns which will apply to all workflows run against this site	not set

File Batching

The file list generated by [discovery tasks](#) may be broken into smaller batches before passing them to workflow steps.

This makes the overall job execution more granular. Individual tasks will be smaller and faster. This also makes it easier to cancel a job, given that only PENDING tasks can be cancelled.

On the other hand, if the batching is too small, the large number of tasks generated may saturate the job queue, blocking out tasks from other jobs.

File batching is based on both `file_batch_gb` and `file_batch_size`. Whichever limit results in a smaller batch is the one which is used. For example, given 100 files of 500MB each, a `file_batch_gb` of 1 and `file_batch_size` of 10 will result in 50 batches of 2 files each (1GB total per batch), because 1GB (2 files) is smaller than 10 files (5GB).

Dynamic File Batching

The Dynamic File Batching feature is designed to improve the processing of tasks with large sets of files by adjusting batch size based on the total number of files being processed from using a predefined set of ranges.

By enabling the auto file batch sizing configuration option, the system automatically adjusts file batch sizes to optimize resource utilization and enhance performance.

REST API

Configurations can be listed and set via the Ngenea Hub REST API.

The sites endpoint supports client key authentication

To list the current site configuration settings,

```
$ curl -s 'http://example.com/api/sites/1/' -H 'Accept: application/json' -H "Authorization: Api-Key $APIKEY"
{
  "name": "site1",
  "elasticsearch_url": "localhost:19200",
  "file_batch_size": 100,
  ...
}
```

Note - configurations are only included when fetching a specific site, not when listing all sites.

To change one or more configuration settings, make a `PATCH` request the same endpoint

```
$ curl -s -X PATCH 'http://example.com/api/sites/1/' -H 'Accept: application/json' -H "Authorization: Api-Key $APIKEY" -H 'Content-Type: application/json' -d '{"file_batch_gb": 5}'
{
  "file_batch_gb": 5,
  ...
}
```

Certificate Lifetime

The certificate lifetime refers to the duration for which a certificate is valid. You can view and modify this duration through the `/api/pki/lifetime/1/` endpoint.

Viewing the Certificate Lifetime

To view the current certificate lifetime, you can send a `GET` request to the `/api/pki/lifetime/1/` endpoint. This will return a JSON response with the current lifetime in days.

```
{
  "days": 2147483647,
  "updated_at": "2024-04-12T09:19:43.732Z",
  "updated_by": 0
}
```

Modifying the Certificate Lifetime

To modify the certificate lifetime, you can send a PATCH request to the `/api/pki/lifetime/1/` endpoint. The request body should be a JSON object with a `days` field specifying the new lifetime in days.

Please note that only administrators and users with the `filebrowser.change_certificatelifetime` permission can modify the certificate lifetime.

Example:

```
curl -X PATCH -H "Content-Type: application/json" -d '{"days": 365}'  
https://your-domain.com/api/pki/lifetime/1/
```

In this example, we are setting the certificate lifetime to 365 days.

Remember to replace the `days` value with the desired lifetime in days.

Summary

The `/api/pki/lifetime/1/` endpoint allows you to view and modify the certificate lifetime. Remember that to use the changed lifetime in the certificates, then Ngenea Hub needs to be restarted.

Feature Flags

Feature flags control whether selected pre-release features are enabled.

Certain features may be included in a release which aren't yet fully implemented, or fully tested. By default, these features are disabled and 'hidden', so should not affect normal functionality.

However, these features may be enabled on a 'preview' basis, on the understanding that they may be incomplete or unstable.

Warning: Do not enable preview features unless you are willing to accept the potential risks.

Once a feature is finalised and stable, it will be released officially, and the corresponding feature flag will be removed.

Available Features

The following features are currently available.

name	description	stability	default
<code>searchui</code>	Enable search features in the Ngenea Hub UI	stable	True

name	description	stability	default
bandwidth_controls	Enable bandwidth controls in the stable Ngenea Hub UI		True

REST API

Features can be listed, enabled, or disabled via the Ngenea Hub REST API.

To list the available features, and whether they're currently enabled

```
$ curl -s 'http://example.com/api/features/' -H 'Accept: application/json' -H "Authorization: Api-Key $TOKEN"
{
  "count": 1,
  "next": null,
  "results": [
    {
      "name": "searchui",
      "description": "Enable search features in the Ngenea Hub UI",
      "enabled": false
    },
    {
      "name": "bandwidth_controls",
      "description": "Enable bandwidth controls in the Ngenea Hub UI",
      "enabled": false,
    }
  ]
}
```

Individual features are keyed by their name, e.g. <http://example.com/api/features/searchui/>

To enable a feature, make a PATCH request against the desired feature

```
$ curl -s -X PATCH 'http://example.com/api/features/searchui/' -H 'Accept: application/json' -H "Authorization: Api-Key $TOKEN" -H 'Content-Type: application/json' -d '{"enabled": true}'
[
  {
    "name": "searchui",
    "description": "Enable search features in the Ngenea Hub UI",
    "enabled": true
  },
  ...
]
```

And similarly, to disable a feature

```
curl -s -X PATCH 'http://example.com/api/features/searchui/' -H
'Accept: application/json' -H "Authorization: Api-Key $TOKEN" -H
'Content-Type: application/json' -d '{"enabled": false}'
```

Note: It may be necessary to restart the Ngenea Hub service for a feature change to take effect.

ngclient

Alternatively, feature flags can be interacted with using [ngclient](#).

To list available features and whether they're currently enabled

```
$ ngclient features list
[X] searchui           Enable search features in the UI
[ ] bandwidth_controls Enable bandwidth controls in the UI
```

To enable a feature

```
ngclient features enable bandwidth_controls
```

And to disable a feature

```
ngclient features disable bandwidth_controls
```

See [ngclient features](#) for more information.

Actions

Actions allow administrative users to perform certain operations, as described below.

Available Actions

`discover_nodes`

Ngenea Hub monitors for when new nodes come online.

The `discover_nodes` action can be used to manually scan for nodes. This may be necessary if a node was previously manually removed from Ngenea Hub

This action takes no arguments.

REST API

Actions are submitted via the `/api/actions/` endpoint, and typically execute asynchronously. The state of the action can be viewed via the same endpoint.

Note: The actions endpoint does not support client key authentication. You must use [JWT Authentication](#).

To submit an action, make a `POST` request against the `/api/actions/` endpoint

```
$ curl -s -X POST 'http://example.com/api/actions/' -H 'Accept: application/json' -H "Authorization: Bearer $JWT_ACCESS_TOKEN" -H 'Content-Type: application/json' -d '{"action": "discover_nodes"}'
{
  "id": 12345,
  ...
}
```

This returns a unique id, which can be used to check the status and results of the action

```
$ curl -s 'http://example.com/api/actions/12345/' -H 'Accept: application/json' -H "Authorization: Bearer $JWT_ACCESS_TOKEN"
{
  "action": "discover_nodes",
  "user": "myuser",
  "state": "SUCCESS"
  "results": {...}
}
```

Limitations

Site Sync

Site Sync must only be used in one direction

The intent of `site_sync` is for one source site to synchronise all required changes to any amount of destination sites and not consider the state of the destination site(s). Site Sync must only be utilised when data is required to be synchronised without concern for any active changes on destination site(s).

Site Sync only considers changes within an applicable time window for synchronisation

Synchronisation is not 'event driven'. Changes are collated within a window of time (defined by the associated schedule), and sent as a group.

The order in which certain events occurred cannot be determined. For example; delete determination; where no data point exists to determine the 'change time' [ctime] of a file or directory due to deletion prior to the sync time window.

During bidirectional synchronisation, when conflicting create/modify and delete events occur for files, the create/modify event takes precedence over the delete event to prevent data loss. In such scenarios, a newer version of the file which was prior deleted will be present after synchronisation. Directory behaviour is not affected.

Site Sync supports Independent Filesets

Site Sync methodology is incompatible with any requirement to synchronise an entire file system, nor is use of Dependent Filesets, or arbitrary directory trees supported.

Destination site Independent Filesets must exist prior to synchronisation

Site Sync does not create Independent Filesets on destination site(s) prior to synchronisation. Destination Independent Fileset creation is an administrative function and must be undertaken prior to configuration and operation of a Site Sync methodology to the destination site.

Directory deletion is not supported

Site sync does not support directory deletion. Deletion of a large file tree structure on a source site will delete files within the directory structure, resulting in an empty directory tree on the destination site.

Bidirectional Site Sync

Bidirectional Site Sync implements eventual consistency

Site Sync adheres to the principle of eventual consistency whereby one or more subsequent Site Sync jobs or tasks are required to be enacted for source and destination sites to be in sync . Prior to all required Site Sync jobs enacting the synchronisation status is viewed as partially synchronised. Each subsequent job increases the totality of synchronisation.

Data which has failed to be synchronised in prior synchronisations is placed into subsequent synchronisation runs, leading to eventual consistency.

Bidirectional Site Sync is only supported via schedules

Ref: Schedules

A bidirectional Site Sync is created via defining a schedule using the `bidirectional_snapdiff` discovery and subscribing the `bidirectional_sync` workflow to the schedule.

A path may only be managed by one schedule per site, and at most one workflow may be subscribed to a `bidirectional_snapdiff` schedule.

Hub does not support multiple bidirectional synchronisation with the same source site for the same Independent Fileset (E.G. between site1 and site2, and between site1 and site3).

The `bidirectional_snapdiff` discovery causes all iterative Independent Fileset changes to be tracked. Identified changes are compared to across iterative runs. When both sets of changes have been evaluated, only appropriately valid changes are synchronised to the destination site.

Bidirectional synchronisation is sequential

When setting up a bidirectional site sync 'site1' is the site which is configured when creating the schedule, and 'site2' is the site configured as the destinationsite when subscribing the workflow to the schedule.

A bidirectional site sync will first synchronise changes from site1 to site2, and then from site2 to site1.

A failure while synchronising site1 to site2 will not block the reverse direction sync from enacting.

The sequential nature of synchronisation ensures conflict situations where the synchronisation from site2 to site1 wins by virtue of the last write [most recent write at any site] of a file taking precedence.

Swapping files / Last write wins

Synchronisation of a set of file moves whereby the actions include synchronisation determination of the paths of two files being swapped is complex and can result in conflicts. Where identical file paths exist at the destination site, file moves will fail rather than overwriting the existing files.

Manual intervention is required before a synchronisation will again succeed. N.B.: the replaying of the swap of the files on the destination site is not sufficient to resolve the conflict.

Renaming or deleting files and directories causes re-sending of data

Where a file is moved on site1 and the same file is deleted on site2 during an active synchronisation, the moved file from site1 will be resent to site2. This behaviour will be observed even if the delete event occurred later chronologically.

Renaming or files and directories on both bidirectional Site Sync sites causes duplication of data

This behaviour is observed when a file or directory is moved on both sites to different locations during an active synchronisation. E.G.:

- /mmfs1/data/path1 is moved to /mmfs1/data/path2 on site1
- /mmfs1/data/path1 is moved to /mmfs1/data/path3 on site2

This scenario results in duplicate data at both sites in both /mmfs1/data/path2 and /mmfs1/data/path3.

Creation or deletion of empty directories on site 1 does not synchronise to site 2

Site Sync does not perform deletions of empty directories on a destination site and does not create empty directories on a destination site.

Troubleshooting

This section outlines steps for troubleshooting issues with Ngenea Hub

Service Status

To check the status of Ngenea Hub and its individual services

```
ngeneahubctl status
```

To check the status of Ngenea Worker

```
systemctl status ngenea-worker
```

Service Logs

The full logs for Ngenea Hub can be viewed with

```
journalctl -u ngeneahub
```

To view Ngenea Worker logs

```
journalctl -u ngenea-worker
```

Specific Features

Site Sync

Site sync - both one-way and bidirectional - may fail due to conflicts which cannot be automatically resolved. This page outlines options for intervening to resolve such conflicts.

Snapshot Rotation

By default, snapdiff snapshots will always rotate, even if an error occurs during sync.

Traditionally, snapshots are rolled-back on error. However, this can lead to changes being replayed inappropriately, leading to further errors and conflicts. Because of this, the default behaviour was changed to always rotate.

The downside to this approach is that some changes may be missed by sync. For example, if a network issue prevents a file from being synced, that file will not be re-synced unless or until it changes again. Note, however, there are retries within a sync run to mitigate such temporary issues.

If a sync job does fail, it will be necessary to determine the source of the error and manually resolve any issues. The job details will report on which paths failed overall. Looking at the individual task details will give more information on where and why a specific path failed.

To disable this behaviour, so that snapshots will rollback on error, set the runtime field `snapdiff_rotate_on_error` to `False`

Note: If the sync is run within a schedule, it will use the first subscribed workflow's `snapdiff_rotate_on_error` value. This does not apply to `bidirectional_sync` as it can only have one subscribed workflow.

Manual Resolution

In some cases it is possible to resolve conflicts by manually applying changes.

For example, if a file is moved on site A and deleted on site B, sync will fail because there is no file to move (or delete, depending on the sync direction) on the target site. In this case, manually deleting the file on site A, or re-sending the file (in its new location) onto site B will resolve the conflict. Thereafter, sync will be able to run without issue.

Re-sync all

Another option is to re-sync everything from scratch. This is the safest option, as it ensures that no file changes are lost.

Note that sync will skip any files which are already in the correct state, so a re-sync won't take as long as syncing to a brand new target.

Snapdiff-based sync uses filesystem snapshots to track file changes over time. The snapdiff discovery uses a 'last snap' file to record the last snapshot which was successfully synced.

This last snap file is located at `/mmfs1/.rotate/ngenea-worker.lastsnap.name.<fileset_name>.id.<fileset_id>`

By removing the last snap file, the next sync run will behave as if it has not been run before, and so will sync everything. Once sync has run successfully, you can safely delete the snapshot which was previously recorded in the last snap file (before that file was deleted).

Force rotate

The riskiest option is to force a snapshot rotation. This effectively says that you don't care about the current failure and just want the sync to move on.

As discussed above, this is the default behaviour. Note that this may result in some file changes not being synced. For a safer option, see **Re-sync all** above.

The following steps can be used as a one-off when the default rotate-on-error function has been disabled.

To force a rotate, you should first temporarily disable sync. This can be done by setting the sync schedule to disabled in the Ngenea Hub UI.

Next, create a new snapshot of the fileset. The name is expected to be of the form `ngenea-worker.snapdiff.<timestamp>`.

Update the 'last snap' file (described above), replacing the currently recorded snapshot name with the name of the new snapshot you just created.

This last snap file is located at `/mmfs1/.rotate/ngeneaworker.lastsnap.name.<fileset_name>.id.<fileset_id>`

Finally, re-enable sync. Sync will now pick up new file changes starting from the point at which the new snapshot was created.

Once sync has run successfully, you can safely delete the snapshot which was previously recorded in the last snap file (before that file was updated).

Locking Errors

Lock files are used to ensure that a sync for a given fileset will not run if one is already running.

In this case, any new sync job will fail, and under the snapdiff task details, you will see an error like

```
SnapdiffLockError: Could not perform snapdiff as one is currently running for provided fileset
```

Under rare circumstances, a lock may not be correctly cleaned up, preventing syncs from running, even though there are none currently active.

In that case, the lock file can be removed manually. First, ensure there aren't any syncs running. For extra safety, temporarily disable any scheduled syncs.

The lock file is located at `/mmfs1/.rotate/snapdiff.name.<fileset_name>.id.<fileset_id>.lock`

Any snapdiff lock file will automatically expire after 24 hours by default. The lifetime can be changed using the `lock_threshold` [site setting](#)

Reference

Default Workflows

This section documents all the workflows which come installed in Ngenea Hub by default. See [Custom Workflows](#) for guidance on how to create your own workflows.

migrate

Migrate one or more files from a site.

discovery: `recursive`

steps:

- `dynamo.tasks.migrate`

fields:

- `lock_level` (choice): Filesystem locking level for ngenea operations. One of: `partial`, `implicit`, `none`. Default=`partial`

premigrate

Premigrate one or more files from a site.

discovery: `recursive`

steps:

- `dynamo.tasks.migrate`
 - `premigrate`: `True`

fields:

- `lock_level` (choice): Filesystem locking level for ngenea operations. One of: `partial`, `implicit`, `none`. Default=`implicit`

recall

Recall one or more files on to a site.

discovery: `recursive`

steps:

- `dynamo.tasks.recall`

fields:

- `lock_level` (choice): Filesystem locking level for ngenea operations. One of: `partial`, `implicit`, `none`. Default=`implicit`

send

Send files from one site to another via cloud storage.

discovery: `recursive`

steps:

- `dynamo.tasks.remove_location_xattrs_for_moved`
- `dynamo.tasks.migrate`
 - `premigrate`: `True`
- `dynamo.tasks.reverse_stub`

fields:

- `destination` (string): site to send files to
- `destinationqueue` (string): queue on the destination site to run the task on

- `hydrate` (bool): hydrate files on the destination site

site_sync

Sync a fileset from one site to another via cloud storage.

The `snapdiff` discovery looks for changes within the fileset on the source site since the last time the workflow was invoked. These changes are then synced to the destination site.

The workflow should be invoked with a single path which is the link point of the independent fileset to be synced.

discovery: `snapdiff`

fields:

- `destinationsite` (string): site to sync changes to
- `destinationqueue` (string): queue on the destination site to run the task on
- `sync_preference` (string): determines how conflicts should be resolved on the remote site. One of: newest, local, ignore

created

Files with state `created` are sent to the destination site, subject to `sync_preference`

steps:

- `dynamo.tasks.remove_location_xattrs_for_moved`
- `dynamo.tasks.migrate`
 - `premigrate`: True
 - `overwrite`: True
 - `abort_missing`: True
- `dynamo.tasks.reverse_stub`
 - `overwrite`: True

updated

Files with state `updated` are sent to the destination site, subject to `sync_preference`

steps:

- `dynamo.tasks.check_sync_state`
- `dynamo.tasks.remove_location_xattrs_for_moved`
- `dynamo.tasks.migrate`
 - `premigrate`: True
 - `overwrite`: True
 - `abort_missing`: True
- `dynamo.tasks.reverse_stub`
 - `overwrite`: True

moved

Files with state `moved` are moved 'in-place' on the destination site

steps:

- `dynamo.tasks.move_paths_on_gpfs`

deleted

Files with state `deleted` are removed on the destination site

steps:

- `dynamo.tasks.delete_paths_from_gpfs`

`import_to_site_hydrated`

This workflow is designed to import fully hydrated remote objects from external ngenea targets to the local filesystem.

discovery: `null`

steps:

- `dynamo.tasks.import_files_from_external_target`
 - `hydrate: True`

fields:

- `endpoint` (string): Name of the ngenea target configuration
- `location` (string): An absolute path for where to place the file/folder in the local filesystem, this can be a different path than the remote location. Defaults to `None`
- `lock_level` (string): Filesystem locking level for ngenea operations. One of: `partial`, `implicit`, `none`. Default=`implicit`

`import_to_site_dehydrated`

This workflow is designed to import stubbed remote objects from external ngenea targets to the local filesystem.

discovery: `null`

steps:

- `dynamo.tasks.import_files_from_external_target`
 - `hydrate: False`

fields:

- `endpoint` (string): Name of the ngenea target configuration
- `location` (string): An absolute path for where to place the file/folder in the local filesystem, this can be a different path than the remote location. Defaults to `None`
- `lock_level` (string): Filesystem locking level for ngenea operations. One of: `partial`, `implicit`, `none`. Default=`implicit`

Hidden Workflows

This section documents all the workflows which come installed in Ngenea Hub by default that are not provided in the UI. Some of these are intended for advanced users making use of the API, and some are used for GPFS policy automation.

transparent_recall

This is the workflow that will be performed on dehydrated files on the system when accessed, if the policy is configured to make use of Ngenea Hub within its Transparent Recall policy.

Typically, all of these Transparent Recalls are placed into a single job within Ngenea Hub.

steps:

- `dynamo.tasks.recall`
 - `lock_level`: "partial"

reverse_stub

This workflow is designed to create stubs files or fully hydrated files using remote content on a given site without the need for an existing file on the site for a given path. Due to this requiring the data to be within the cloud and not present on the filesystem, it requires the paths to target cloud targets through the API.

steps:

- `dynamo.tasks.reverse_stub`

fields:

- `hydrate` (bool): If the file should be fully hydrated during the recall operation
- `overwrite` (bool): If the operation should overwrite a local file that already exists

delete_file

This workflow deletes one or more file/folder paths from the filesystem.

steps:

- `dynamo.tasks.delete_paths_from_gpfs`

fields:

- `recursive` (bool): If non-empty directories should be deleted. if recursive is `false` additional jobs will be needed to delete empty directories. Defaults to `false`

Discovery Steps

This section documents all the currently supported discovery steps in Ngenea Hub. See [Custom Workflows](#) for guidance on how to use these in your own workflows.

`dynamo.tasks.recursive_navigate` - recursive - Function: discovery

Navigates down any folder tree provided using the initial paths as the base of this navigation. It will process every discovered item and can have its behaviour defined using type rules using `file|link|folder` or action on all items with `all` as its type and state.

Upon discovering non-empty directories, it will create another instance of this task to navigate down that directory while still processing the current directory. Once a batch of items has been found it will action the appropriate rule on them.

Argument	Type	Default	Description
<code>skip_missing</code>	<code>bool</code>	<code>False</code>	Allows the processing of any files directly provided to the discovery step regardless of it being on the filesystem.

`dynamo.tasks.recursive_action` - recursive(deprecated) - Function: discovery

Navigates down any folder tree provided to it and actions any rule defined with `all` as its type and state on all found files.

Argument	Type	Default	Description
<code>skip_missing</code>	<code>bool</code>	<code>False</code>	Allows the processing of any files directly provided to the discovery step regardless of it being on the filesystem.

`dynamo.tasks.snapdiff` - snapdiff - Function: discovery

For use only with a GPFS Independent Fileset, it will create a GPFS snapshot and it will then process the list of differences between the time of the initial run and subsequent runs. On the first run of this, it will ingest all the files within that Fileset.

Argument	Type	Default	Description
<code>skip_old_ctimes</code>	<code>bool</code>	<code>False</code>	Skips any files within the snapdiff difference list if the ctime of the file is older than the oldest snapshot.
<code>condense_moves</code>	<code>bool</code>	<code>True</code>	If a directory is moved, this will condense all the move operations into a single operation for move based tasks, otherwise it will action against every effected file.

Workflow Steps

The following are all the currently supported steps in Ngenea Hub. All of these steps are ran on the Ngenea Worker provided site on its respective function queue when running a workflow. See [Custom Workflows](#) for guidance on how to use these steps in your own workflows.

Each task alongside its name has its function queue detailed, these functions can be controlled through the Ngenea Worker [configuration](#)

`dynamo.tasks.migrate` - Function: Worker

Migrates a list of files to a pre-defined remote target using Ngenea.

Argument	Type	Default	Description
<code>premigrate</code>	<code>bool</code>	<code>False</code>	retain the content of every migrated file and do not set the OFFLINE flag for the file.migrating.
<code>stub_size</code>	<code>int</code>	<code>0</code>	retain a segment of every migrated file starting from its beginning and having a specified approximate length in bytes.
<code>overwrite</code>	<code>bool</code>	<code>False</code>	overwrite remote objects if they already exist--do not create remote object instances with various UUID suffixes
<code>fail_on_mismatch</code>	<code>bool</code>	<code>False</code>	fail a file migration if a remote object exists but has different hash or metadata. In that case, the task errors
<code>lock_level</code>	<code>string</code>	<code>implicit</code>	Defined the locking mode that ngenea will use when performing the migrate
<code>endpoint</code>	<code>string</code>		specify the endpoint to migrate
<code>abort_missing</code>	<code>bool</code>	<code>False</code>	allow the migrate task to make any missing files end up in the "aborted" state instead of "failed"
<code>migrate_offline_files</code>	<code>bool</code>	<code>False</code>	If set to True, the migrate task will process offline files in the same manner as regular online files. If set to False, the migrate task will process the offline files using <code>--sync-metadata</code>

`dynamo.tasks.recall` - Function: Worker

Recalls a list of files to a pre-defined remote target using Ngenea.

Argument	Type	Default	Description
skip_hash	bool	False	If the recall should skip checking the hash of the file
endpoint	string		specify which endpoint(site) to recall from
lock_level	string	partial	Defines the locking level ngenea will use during the recall
default_uid	string		When a file is recalled, it uses this UID if one is not set on the remote object
default_gid	string		When a file is recalled, it uses this GID if one is not set on the remote object
update_atime	bool	false	When a files is recalled, update its access time (atime) to 'now'
update_mtime	bool	false	When a files is recalled, update its modification time (mtime) to 'now'
delete_remote	bool	false	If set, when a file is recalled, it deletes the file in the remote location

dynamo.tasks.reverse_stub - Function: Worker

Recalls a list of files to a pre-defined remote target using Ngenea.

Argument	Type	Default	Description
hydrate	bool	False	If the file should be premigrated instead of a regular stub
stub_size	int	0	The max file size before files will be stubbed for this task
skip_hash	bool	False	If the recall should skip checking the hash of the file
overwrite	bool	False	Overwrite local files if they already exist, except files with only metadata changes.
endpoint	string		specify which endpoint(site) to recall from.
retry_stale	string	None	Controls if the worker should attempt to retry file failures due to stale file handles. This string can be either stub for only removing reverse stubbed files or all.
lock_level	string	implicit	Defines the locking level ngenea will use during the recall
default_uid	string		When a file is recalled, it uses this UID if one is not set on the remote object
default_gid	string		When a file is recalled, it uses this GID if one is not set on the remote object
update_atime	bool	false	When a files is recalled, update its access time (atime) to 'now'
update_mtime	bool	false	

Argument	Type	Default	Description
			When a files is recalled, update its modification time (mtime) to 'now'
<code>conflict_preference</code>	<code>string</code>	<code>None</code>	Dictates what state the local file should be to pass the check. Options are "newest" which passes if the local file is the latest version of the file on either site, "local" which accepts the local file version regardless of the check and "ignore" which always uses the other sites file version.

`dynamo.tasks.ngenea_sync_metadata` - Function: Worker

Sync the local ngenea metadata on a file with the remote target using Ngenea.

Argument	Type	Default	Description
<code>skip_hash</code>	<code>bool</code>	<code>False</code>	If the sync should skip checking the hash of the file
<code>endpoint</code>	<code>string</code>		Specify which endpoint(site) to recall from
<code>default_uid</code>	<code>string</code>		When a file is synced, it uses this UID if one is not set on the remote object
<code>default_gid</code>	<code>string</code>		When a file is synced, it uses this GID if one is not set on the remote object

`dynamo.tasks.delete_paths_from_gpfs` - Function: Worker

Removes a list of files from a GPFS filesystem.

Argument	Type	Default	Description
<code>recursive</code>	<code>bool</code>	<code>False</code>	If any directory path is provided and this is set, it will remove the entire file tree, otherwise it will only remove empty directories. It is important to note that the recursive behaviour of removing the entire directory tree will not apply to filesets or if the target directory contains files or directories that are restricted from being deleted such as <code>snapshots</code> , in such cases the task will silently ignore those files or directories and report the task as successful.

`dynamo.tasks.check_sync_state` - Function: Worker

Checks a provided site against the calling sites to ensure that the local file is in a specified state compared to another site. Using this task will also perform `dynamo.tasks.stat_paths` on the provided site before execution.

Argument	Type	Default	Description
<code>sync_preference</code>	string	ignore	Dictates what state the local file should be to pass the check. Options are "newest" which passes if the local file is the latest version of the file on either site, "local" which accepts the local file version regardless of the check and "ignore" which always uses the other sites file version.
<code>site</code>	string		The target site to compare
<code>abort_outdated</code>	bool	False	If this bool is set files that do not need to be executed will be marked as aborted as opposed to skipped
<code>hash_includes_acl</code>	bool	False	If this bool is set the metadata comparison for directories will also compare Access Control Lists

`dynamo.tasks.move_paths_on_gpfs` - Function: Worker

Moves files on the filesystem using provided paths with a `source` key.

This task moves files in two steps (via an intermediate temporary location), to avoid move conflicts (for example, this task correctly handles cases where files are 'swapped': `fileA` moved to `fileB`, and `fileB` moved to `fileA`).

Argument	Type	Default	Description
<code>delete_remote_xattrs</code>	bool	False	If set, after a file has been moved all remote location xattrs will be removed
<code>source_missing_signature</code>	json	null	A signature to send any paths to where the source is missing. If this isn't set, a missing source is treated as a failure.
<code>target_max_age</code>	int	null	If the target file exists, it is overwritten by default. If this optional parameter is set, the target has to have a ctime $\leq$ this timestamp for the move to proceed. Otherwise, the source file is removed. Given in seconds since epoch. Primarily used for sync workflows.

`dynamo.tasks.one_step_move_paths_on_gpfs` - Function: Worker

Moves files on the filesystem using provided paths with a `source` key.

This task is not used in the default Ngenea Hub workflows. It is less robust than `dynamo.tasks.move_paths_on_gpfs`, because it moves files directly from source to their new location (in one step). So for example, trying to 'swap' files (`fileA` moved

to fileB, and fileB moved to fileA) with this task will effectively result in one of these files being deleted on the target site.

However, this task is for cases where doing moves in two steps is not an acceptable option.

Argument	Type	Default	Description
<code>delete_remote_xattrs</code>	<code>bool</code>	<code>False</code>	If set, after a file has been moved all remote location xattrs will be removed
<code>source_missing_signature</code>	<code>json</code>	<code>null</code>	A signature to send any paths to where the source is missing. If this isn't set, a missing source is treated as a failure.
<code>target_max_age</code>	<code>int</code>	<code>null</code>	If the target file exists, it is overwritten by default. If this optional parameter is set, the target has to have a ctime $\leq$ this timestamp for the move to proceed. Otherwise, the source file is removed. Given in seconds since epoch. Primarily used for sync workflows.

`dynamo.tasks.remove_location_xattrs_for_moved` - Function: Worker

This task removes all remote location xattrs on all provided paths.

This step takes no additional arguments.

`dynamo.tasks.move_in_cloud` - Function: Worker

Moves a file on the filesystem's related cloud storage platform using provided paths with a `source` key.

This step takes no additional arguments.

`dynamo.tasks.remove_from_cloud` - Function: Worker

Deletes a file on the filesystem's related cloud storage platform using provided paths.

This step takes no additional arguments.

`dynamo.tasks.ensure_cloud_file_exists` - Function: Worker

Ensures all files provided to the task exist on the filesystem's related cloud storage platform. If some do not, it will attempt to retry this check an additional two more times before failing.

This step takes no additional arguments.

`dynamo.tasks.import_files_from_external_target` - Function: Worker

Recalls a list of files from an external target to a predefined local object using Ngenea. The input files must be a list of remote object paths not local file paths. (eg, projects/)

Argument	Type	Default	Description
<code>endpoint</code>	<code>string</code>		Specify which endpoint(ngenea target name) to recall from
<code>lock_level</code>	<code>string</code>	<code>implicit</code>	Defines the locking level ngenea will use during the recall
<code>location</code>	<code>string</code>	<code>null</code>	Specify an absolute path for where to place the file/folder in the local filesystem
<code>hydrate</code>	<code>bool</code>	<code>false</code>	If the remote object should be premigrated instead of a regular stub
<code>extra_flags</code>	<code>list</code>		List of extra arguments that can be passed to recall command
<code>skip_hash</code>	<code>bool</code>	<code>false</code>	If the recall should skip checking the hash of the file
<code>default_uid</code>	<code>string</code>		When a file is recalled, it uses this UID if one is not set on the remote object
<code>default_gid</code>	<code>string</code>		When a file is recalled, it uses this GID if one is not set on the remote object
<code>update_atime</code>	<code>bool</code>	<code>false</code>	When a files is recalled, update its access time (atime) to 'now'
<code>update_mtime</code>	<code>bool</code>	<code>false</code>	When a files is recalled, update its modification time (mtime) to 'now'

`dynamo.tasks.copy_files_from_external_target`

Copies a list of files from an external target to a predefined local location using Ngenea. The input files should be provided as a list of remote object paths, not local file paths (e.g., projects/). Once copied, these files are no longer associated with the target.

Argument	Type	Default	Description
<code>endpoint</code>	<code>string</code>		Specify which endpoint(ngenea target name) to recall from
<code>lock_level</code>	<code>string</code>	<code>implicit</code>	Defines the locking level ngenea will use during the recall
<code>location</code>	<code>string</code>	<code>null</code>	Specify an absolute path for where to place the file/folder in the local filesystem
<code>default_uid</code>	<code>string</code>		

Argument	Type	Default	Description
		requires ignore_metadata to be: True, else ignored	When a file is recalled, it uses this UID if one is not set on the remote object
default_gid	string	requires ignore_metadata to be: True, else ignored	When a file is recalled, it uses this GID if one is not set on the remote object
extra_flags	list		List of extra arguments that can be passed to recall command
ignore_metadata	bool	false	Do not read user, group and timestamps etc, from data to be copied

Automated Tasks

All detailed tasks below are run on automated schedules defined by the hub itself, some of these can have their schedules adjusted and thresholds adjusted.

Every minute

`alerts.tasks.fetch_alerts`

Submits the task `dynamo.tasks.get_alerts` to retrieve and sync the alerts on all sites with the Ngenea Hub's internal database. This is run on the `discovery` function of the . The returning task that stores the alerts in the database runs within the internal `celery` queue within Ngenea Hub.

The schedule of this task can be controlled by using the `SYNC_ALERTS_INTERVAL` config setting, any provided value must comply with cron syntax.

Every 30 minutes

`dynamocore.tasks.admin.load_storagepools`

Fetches the GPFS storage pools from all active sites registered within Ngenea Hub and stores them within the database. Additionally, it will also create the default filesystem root spaces if they don't exist for a site.

The schedule of this task can be controlled by using the `REFRESH_SITE_ANALYTICS_INTERVAL` config setting, any provided value must comply with cron syntax.

`dynamocore.tasks.space_site.send_check_space_site_quota`

Sends a task to the Ngenea Hub's internal `celery` queue to collect all spaces within every site named `dynamocore.tasks.space_site.send_check_space_site_quota` which will then queue the worker task `dynamo.tasks.check_space_quotas` on every site's `worker` function queue. It will then send a task back to the hub `celery` queue to store the quotas in the database.

The schedule of this task can be controlled by using the [SYNC_SPACES_QUOTA_INTERVAL](#) config setting, any provided value must comply with cron syntax.

Hourly

`dynamocore.tasks.jobs.update_state_as_failed_for_inactive_tasks`

This task will assess all tasks currently that have been started in Ngenea Hub and will ensure that they are not lost due to a communication error.

Any tasks found to not be actively running within any of the defined queues will be marked as invalid after the configurable time threshold `task_invalidation_timeout` in the [configuration API](#).

The schedule of this task can be controlled by using the [INACTIVE_TASKS_INTERVAL](#) config setting, any provided value must comply with cron syntax.

`dynamocore.tasks.jobs.invalidate_cancelled_job_tasks`

After jobs are cancelled within the Ngenea Hub some tasks can be left in an un-cancelled state if the service un-expectedly exits leaving tasks in a non-cancelled state indefinitely. This task will automatically clean up these tasks that are in incorrect states.

The schedule of this task can be controlled by using the [INVALIDATE_CANCELLED_JOB_TASKS_INTERVAL](#) config setting, any provided value must comply with cron syntax.

`dynamocore.tasks.network.ensure_consistent_tc_rules`

With bandwidth controls enabled for any sites this task ensures that these rules are enacted on each site. Sites that are already compliant with their networking rules will not see any effect.

`dynamocore.tasks.events.cleanup_old_events`

This task automatically removes bidirectional sync events within the database. It will clear all objects for all but the latest 2 jobs along with any jobs that have been cancelled and failed

The schedule of this task can be controlled by using the [CLEANUP_OLD_EVENTS_INTERVAL](#) config setting, any provided value must comply with cron syntax.

`dynamocore.tasks.browsing.browsing_task_clear`

Removes all cached task results for API requests for file details for the file browser in the UI. It will remove all entries that are older than an hour along with any that received errors when retrieving results.

`dynamocore.tasks.system_settings.create_or_update_site_settings_auto`

This task starts up a task in the main `celery` queue within Ngenea Hub for updating or creating outstanding settings across all active sites within the management of Ngenea Hub. This starts by executing `dynamocore.tasks.site.active_sites` followed immediately by `dynamocore.tasks.system_settings.create_or_update_site_settings_for_sites`. This will cause Ngenea Hub to retrieve the settings for every site using `dynamo.tasks.get_config_for_keys` and update the hub database accordingly using an additional task on the main `celery` queue within Ngenea Hub.

This call uses the main `worker` function queue with the , more details on queue functions are available in the [function configuration](#).

The schedule of this task can be controlled by using the `SYNC_SITE_SETTINGS_INTERVAL` config setting, any provided value must comply with cron syntax.

```
dynamocore.tasks.system_settings.sync_global_settings_auto
```

Starts up a task in the main `celery` within Ngenea Hub that creates a collection of tasks to retrieve the global settings and external Ngenea targets within the hub database and apply them to every . It achieves this with the Ngenea Worker task `dynamo.tasks.set_config_and_apply` this is run on the main `worker` function.

The schedule of this task can be controlled by using the `SYNC_GLOBAL_SETTINGS_INTERVAL` config setting, any provided value must comply with cron syntax.

```
dynamocore.tasks.spaces.sync_spaces
```

When a space is created or updated and the job fails, this automated task schedules a retry attempt on making the space. If this is already being processed, it will not update the space within this interval.

The schedule of this task can be controlled by using the `SYNC_SPACES_INTERVAL` config setting, any provided value must comply with cron syntax.

Every 12 Hours

```
dynamocore.tasks.site.fetch_site_analytics
```

.. note:: This has an offset of 37 minutes to prevent overlapping system tasks.

Retrieves the site-wide analytics from all sites and then stores them within the Ngenea Hub database. The timeout for this operation is controlled via the `analytics_timeout` in the [configuration API](#). This analytics task is run on the `discovery` function within the Ngenea Hub.

The schedule of this task can be controlled by using the `REFRESH_SITE_ANALYTICS_INTERVAL` config setting, any provided value must comply with cron syntax.

`dynamocore.tasks.search.remove_old_search_results`

This automated task removes all the old search result caches for searches submitted by the UI. The time period for this lifetime is controlled by the `search_result_ttl` in the [configuration API](#), the default for this threshold is 7 days.

The schedule of this task can be controlled by using the [REMOVE_OLD_SEARCH_RESULTS_INTERVAL](#) config setting, any provided value must comply with cron syntax.

`dynamocore.tasks.jobs.expire_old_jobs`

This automated task removes all the old jobs (settings, workflows, scheduled) that have been submitted and processed within Ngenea Hub.

The threshold for this can be controlled by the `jobs_ttl` in the [configuration API](#), the default for this is 30 days.

The schedule of this task can be controlled by using the [EXPIRE_OLD_JOBS_INTERVAL](#) config setting, any provided value must comply with cron syntax.

`dynamocore.tasks.network.update_cloud_ips_for_s3`

Once per day, Ngenea Hub collects the IP addresses published by Amazon for their S3 service for uses in controlling the bandwidth of any S3 related operations going through Ngenea Worker.

If there are any changes in the address listings, this will be reported to the Ngenea Worker which will adjust any network interface it may be managing.

If bandwidth controls are not enabled, the changes will have no effect on any existing Ngenea Worker instances.

`dynamocore.tasks.network.update_cloud_ips_for_azure`

Once per day, Ngenea Hub collects the IP addresses published by Microsoft for their Azure Blob service for uses in controlling the bandwidth of any Azure related operations going through Ngenea Worker.

If there are any changes in the address listings, this will be reported to the Ngenea Worker which will adjust any network interface it may be managing.

If bandwidth controls are not enabled, the changes will have no effect on any existing Ngenea Worker instances.

`dynamocore.tasks.network.update_cloud_ips_for_google`

Once per day, Ngenea Hub collects the IP addresses published by Google for their GCS service for uses in controlling the bandwidth of any GCS related operations going through Ngenea Worker.

If there are any changes in the address listings, this will be reported to the Ngenea Worker which will adjust any network interface it may be managing.

If bandwidth controls are not enabled, the changes will have no effect on any existing Ngenea Worker instances.

`dynamocore.tasks.server.get_remote_servers`

Once per day Ngenea Hub has to retrieve the currently configured CTDB servers on each Ngenea Worker. It achieves this through the task `dynamo.tasks.get_servers` on the worker function of each site. This is then stored through the main Ngenea Hub `celery` queue through the `dynamocore.tasks.server.store_server_callback` task.

The schedule of this task can be controlled by using the `SYNC_REMOTE_SERVERS_INTERVAL` config setting, any provided value must comply with cron syntax.

`dynamocore.tasks.jobs.expire_old_fsobjects`

Removes the stored long term file browser caching entries in the database and will expire any entries that have not been accessed within 7 days.

The schedule of this task can be controlled by using the `EXPIRE_OLD_FSOBJECTS_INTERVAL` config setting, any provided value must comply with cron syntax.

`dynamocore.tasks.refresh.delete_removed_queues`

Once queues are marked for deletion after being removed from a sites config file and the queue removal tool has been run, it will collect these queues and remove them along with and alerts linked to them.

The schedule of this task can be controlled by using the `REMOVED_QUEUE_CLEANUP_INTERVAL` config setting, any provided value must comply with cron syntax.

Job States

When jobs are created on the call of a workflow, they can end up in specific state that

State	Description
Pending	The job is being populated with tasks through its discovery task and will begin when paths have been collected
Started	Some tasks within the job have started to be processed
Success	All tasks in a job have completed successfully
Failure	There was an error or failure when attempting a task within a job, meaning the job could not complete
Skipped	Based on the output of the provided discovery task, no tasks needed to be created so there is no work to

State	Description
Cancelled	The job has been manually closed via request and all remaining task have been cancelled

Task States

When jobs create tasks, after performing their action on the provided files they can end up in specific state as seen below

State	Description
Pending	This task is has been created but has not yet been picked up by a site
Started	This task is now running on site
Success	This task has completed successfully
Failure	There was a unexpected error when attempting a task within a job, meaning the job could not complete and could not provide structured output
Error	There was a captured error when attempting a task within a job, meaning the job will not have processed all paths but has structured output of what has been completed. Any paths which were successfully processed will be handled by subsequent tasks in a task chain.
Skipped	This task has will have no work to perform so it has been automatically skipped by another task
Cancelled	Either a previous task has failed or the job has been manually cancelled, causing this task to no longer run

Note: Job and task states are updated asynchronously. There may be, for example, a short delay between a job/task completing and its state being reported as such.

File States

As a file is processed by a workflow, each task reports whether the state of the file following the task. The containing job reports the state of the file after all workflow steps have completed.

A file can have any of the following states

Processed

The file was successfully processed by the workflow step without issues

Skipped

The files was not processed because it was already in the expected state, e.g. a delete was skipped because the file wasn't found, a migration was skipped because the file was already offline.

Because it is in the expected state, skipped files can be processed by subsequent tasks in the workflow.

Failed

An error occurred meaning the file couldn't be processed. Because of this error, the file will not be processed by later tasks in the workflow.

The overall job will be marked as failed if it contains any failed files. Workflows which use the `snappdiff` discovery will be 'rolled back' in this case.

Failed files typically indicate an issue that needs to be manually resolved, such as network issues.

Aborted

An error occurred meaning the file couldn't be processed. Because the file was aborted, it will not be processed by later tasks in the workflow.

The overall job is **not** marked as failed if it contains aborted files. Workflows which use the `snappdiff` discovery will 'rotate' in this case.

The intended use case for this state is in scheduled (recurring) jobs, where the any aborted files are expected to be successfully processed in a later run.

For example, if a file changes while it is being sent, the send will be aborted to prevent data corruption. But because the file was modified, it will be identified as modified in the next `snappdiff` scan and processed again; effectively retried.

API

GET /actions/

Query Parameters:

- **page** (integer) -- A page number within the paginated result set. When not given, first page is retrieved by default.
- **page_size** (integer) -- Number of results to return per page. Page size parameter can be a number between 20 and 100. For disabling pagination and retrieving all results, 0 should be given. When page size parameter is empty or <20, 20 results are returned by default. When page size parameter >100, 100 results are returned by default.

Status Codes:

- 200 OK --

Response JSON Object:

- **count** (integer) -- (required)
- **next** (string) --
- **previous** (string) --
- **results[].action** (string) -- (required)
- **results[].id** (integer) -- (read only)
- **results[].state** (string) --

POST /actions/

Request JSON Object:

- **action** (string) -- (required)

Status Codes:

- 201 Created --

Response JSON Object:

- **action** (string) -- (required)

GET /actions/{id}/

Parameters:

- **id** (string) --

Status Codes:

- 200 OK --

Response JSON Object:

- **action** (string) -- (required)
- **id** (integer) -- (read only)
- **results** (string) -- (read only)
- **state** (string) --
- **user** (integer) --

GET /alerts/

Query Parameters:

- **name** (string) -- name
- **node** (string) -- node
- **site** (string) -- site
- **severity** (string) -- severity
- **summary** (string) -- summary
- **timestamp** (string) -- timestamp
- **suppressed** (string) -- suppressed

Status Codes:

- 200 OK --

Response JSON Object:

- **[].created** (string) -- Time that the alert was recorded in Hub
- **[].id** (integer) -- (read only)
- **[].name** (string) -- Name of of the alert (required)
- **[].node** (string) -- Hostname of the node the alert originated from
- **[].queue** (string) -- Name of the queue that the alert relates to
- **[].severity** (string) -- (required)
- **[].site** (string) -- (read only)
- **[].summary** (string) -- Summary description of the alert
- **[].suppressed** (boolean) -- Whether the alert is suppressed
- **[].timestamp** (string) -- Time when the alert was triggered (required)

GET /alerts/{id}/

Parameters:

- **id** (integer) -- A unique integer value identifying this alert.

Status Codes:

- 200 OK --

Response JSON Object:

- **created** (string) -- Time that the alert was recorded in Hub

- **id** (integer) -- (read only)
- **name** (string) -- Name of the alert (required)
- **node** (string) -- Hostname of the node the alert originated from
- **queue** (string) -- Name of the queue that the alert relates to
- **severity** (string) -- (required)
- **site** (string) -- (read only)
- **summary** (string) -- Summary description of the alert
- **suppressed** (boolean) -- Whether the alert is suppressed
- **timestamp** (string) -- Time when the alert was triggered (required)

GET /analytics/

Retrieves list of files under given path for given site.

Query Parameters:

- **page** (integer) -- A page number within the paginated result set. When not given, first page is retrieved by default.
- **page_size** (integer) -- Number of results to return per page. Page size parameter can be a number between 20 and 100. For disabling pagination and retrieving all results, 0 should be given. When page size parameter is empty or <20, 20 results are returned by default. When page size parameter >100, 100 results are returned by default.
- **path** (string) -- Target directory path
- **site** (string) -- Site name
- **children** (boolean) -- Include children directories
- **cache_ttl** (integer) -- How long the cache will last for the target path
- **custom_timeout** (integer) -- Timeout for fetching analytics
- **download** (string) -- Download details as file e.g. CSV ?download=csv.

Status Codes:

- 200 OK --

GET /auth/clientkeys/

API endpoint for managing client keys

Query Parameters:

- **page** (integer) -- A page number within the paginated result set. When not given, first page is retrieved by default.
- **page_size** (integer) -- Number of results to return per page. Page size parameter can be a number between 20 and 100. For disabling pagination and retrieving all results, 0 should be given. When page size parameter is empty or <20, 20 results are returned by default. When page size parameter >100, 100 results are returned by default.

Status Codes:

- 200 OK --

Response JSON Object:

- **count** (integer) -- (required)
- **next** (string) --
- **previous** (string) --
- **results[].id** (integer) -- (read only)
- **results[].name** (string) -- Name of the client key (required)
- **results[].url** (string) -- (read only)

- **results[].user** (string) -- (required)

POST /auth/clientkeys/

API endpoint for managing client keys

Request JSON Object:

- **api_key** (string) -- (read only)
- **id** (integer) -- (read only)
- **name** (string) -- Name of the client key (required)
- **url** (string) -- (read only)

Status Codes:

- [201 Created](#) --

Response JSON Object:

- **api_key** (string) -- (read only)
- **id** (integer) -- (read only)
- **name** (string) -- Name of the client key (required)
- **url** (string) -- (read only)

GET /auth/clientkeys/{id}/

API endpoint for managing client keys

Parameters:

- **id** (string) --

Status Codes:

- [200 OK](#) --

Response JSON Object:

- **id** (integer) -- (read only)
- **name** (string) -- Name of the client key (required)
- **url** (string) -- (read only)
- **user** (string) -- (required)

PATCH /auth/clientkeys/{id}/

API endpoint for managing client keys

Parameters:

- **id** (string) --

Request JSON Object:

- **id** (integer) -- (read only)
- **name** (string) -- Name of the client key (required)
- **url** (string) -- (read only)
- **user** (string) -- (read only)

Status Codes:

- [200 OK](#) --

Response JSON Object:

- **id** (integer) -- (read only)
- **name** (string) -- Name of the client key (required)
- **url** (string) -- (read only)
- **user** (string) -- (read only)

DELETE /auth/clientkeys/{id}/

API endpoint for managing client keys

Parameters:

- **id** (string) --

Status Codes:

- [204 No Content](#) --

POST /auth/token/**Request JSON Object:**

- **password** (string) -- (required)
- **username** (string) -- (required)

Status Codes:

- [201 Created](#) --

Response JSON Object:

- **password** (string) -- (required)
- **username** (string) -- (required)

GET /auth/token/publickey/

API endpoint for retrieving public key that is used for token verification.

Status Codes:

- [200 OK](#) --

POST /auth/token/refresh/

Takes a refresh type JSON web token and returns an access type JSON web token if the refresh token is valid.

Request JSON Object:

- **access** (string) -- (read only)
- **refresh** (string) -- (required)

Status Codes:

- [201 Created](#) --

Response JSON Object:

- **access** (string) -- (read only)
- **refresh** (string) -- (required)

POST /auth/token/verify/

Verifies that the token is not expired AND the token owner exists in the database AND the token owner is an active user.

Request JSON Object:

- **token** (string) -- (required)
- **type** (string) -- Token type e.g: access or refresh (required)

Status Codes:

- [201 Created](#) --

Response JSON Object:

- **token** (string) -- (required)
- **type** (string) -- Token type e.g: access or refresh (required)

GET /configurations/

API endpoint for viewing and setting configurations.

Status Codes:

- 200 OK --

Response JSON Object:

- **analytics_timeout** (integer) -- Maximum time to wait for results from analytics in seconds
- **custom_statistics_tasks** (object) -- A list of custom tasks to be considered discovery tasks
- **force_local_managed_users** (boolean) -- Allow NAS users to be managed on AD joined sites
- **initial_gid** (integer) -- Minimum GID to be configured
- **initial_uid** (integer) -- Minimum UID to be configured
- **jobs_ttl** (integer) -- Time to store job details after the job completes, in days
- **maximum_external_results** (integer) -- Maximum number of items to retrieve per page from an external target scan
- **node_health_last_heartbeat_interval** (integer) -- The time since the last health heartbeat was recorded in seconds
- **search_backend** (string) -- Search backend
- **search_max_results** (integer) -- Maximum search results
- **search_result_ttl** (integer) -- Maximum time to store search results in days
- **snapdiff_stream_timeout** (integer) -- Maximum amount of minutes to wait for task results
- **snapshot_create_delete_retry_timeout** (integer) -- Maximum time for workers to retry snapshot create and delete operations, in seconds
- **stat_refresh_period** (integer) -- Maximum time to wait for results from stat in seconds
- **stat_timeout** (integer) -- Maximum time to wait for results from stat in seconds
- **task_invalidation_timeout** (integer) -- Maximum amount of minutes before a task in the STARTED state is considered invalid

PATCH /configurations/

API endpoint for viewing and setting configurations.

Request JSON Object:

- **analytics_timeout** (integer) -- Maximum time to wait for results from analytics in seconds
- **custom_statistics_tasks** (object) -- A list of custom tasks to be considered discovery tasks
- **force_local_managed_users** (boolean) -- Allow NAS users to be managed on AD joined sites
- **initial_gid** (integer) -- Minimum GID to be configured
- **initial_uid** (integer) -- Minimum UID to be configured
- **jobs_ttl** (integer) -- Time to store job details after the job completes, in days
- **maximum_external_results** (integer) -- Maximum number of items to retrieve per page from an external target scan
- **node_health_last_heartbeat_interval** (integer) -- The time since the last health heartbeat was recorded in seconds
- **search_backend** (string) -- Search backend
- **search_max_results** (integer) -- Maximum search results
- **search_result_ttl** (integer) -- Maximum time to store search results in days

- **snappdiff_stream_timeout** (integer) -- Maximum amount of minutes to wait for task results
- **snapshot_create_delete_retry_timeout** (integer) -- Maximum time for workers to retry snapshot create and delete operations, in seconds
- **stat_refresh_period** (integer) -- Maximum time to wait for results from stat in seconds
- **stat_timeout** (integer) -- Maximum time to wait for results from stat in seconds
- **task_invalidation_timeout** (integer) -- Maximum amount of minutes before a task in the STARTED state is considered invalid

Status Codes:

- 200 OK --

Response JSON Object:

- **analytics_timeout** (integer) -- Maximum time to wait for results from analytics in seconds
- **custom_statistics_tasks** (object) -- A list of custom tasks to be considered discovery tasks
- **force_local_managed_users** (boolean) -- Allow NAS users to be managed on AD joined sites
- **initial_gid** (integer) -- Minimum GID to be configured
- **initial_uid** (integer) -- Minimum UID to be configured
- **jobs_ttl** (integer) -- Time to store job details after the job completes, in days
- **maximum_external_results** (integer) -- Maximum number of items to retrieve per page from an external target scan
- **node_health_last_heartbeat_interval** (integer) -- The time since the last health heartbeat was recorded in seconds
- **search_backend** (string) -- Search backend
- **search_max_results** (integer) -- Maximum search results
- **search_result_ttl** (integer) -- Maximum time to store search results in days
- **snappdiff_stream_timeout** (integer) -- Maximum amount of minutes to wait for task results
- **snapshot_create_delete_retry_timeout** (integer) -- Maximum time for workers to retry snapshot create and delete operations, in seconds
- **stat_refresh_period** (integer) -- Maximum time to wait for results from stat in seconds
- **stat_timeout** (integer) -- Maximum time to wait for results from stat in seconds
- **task_invalidation_timeout** (integer) -- Maximum amount of minutes before a task in the STARTED state is considered invalid

GET /datastores/

API endpoint for managing DataStores.

Query Parameters:

- **page** (integer) -- A page number within the paginated result set. When not given, first page is retrieved by default.
- **page_size** (integer) -- Number of results to return per page. Page size parameter can be a number between 20 and 100. For disabling pagination and retrieving all results, 0 should be given. When page size parameter is

empty or <20, 20 results are returned by default. When page size parameter >100, 100 results are returned by default.

Status Codes:

- 200 OK --

Response JSON Object:

- **count** (integer) -- (required)
- **next** (string) --
- **previous** (string) --
- **results[].accesskey** (string) --
- **results[].accesskeyid** (string) --
- **results[].bucket** (string) --
- **results[].container** (string) --
- **results[].credentialsfile** (string) --
- **results[].endpoint** (string) --
- **results[].id** (integer) -- (read only)
- **results[].name** (string) -- DataStore Name (required)
- **results[].region** (string) --
- **results[].secretaccesskey** (string) --
- **results[].storageaccount** (string) --
- **results[].type** (string) -- Site Type (required)
- **results[].url** (string) -- (read only)

POST /datastores/

API endpoint for managing DataStores.

Request JSON Object:

- **accesskey** (string) --
- **accesskeyid** (string) --
- **bucket** (string) --
- **container** (string) --
- **credentialsfile** (string) --
- **endpoint** (string) --
- **id** (integer) -- (read only)
- **name** (string) -- DataStore Name (required)
- **region** (string) --
- **secretaccesskey** (string) --
- **storageaccount** (string) --
- **type** (string) -- Site Type (required)
- **url** (string) -- (read only)

Status Codes:

- 201 Created --

Response JSON Object:

- **accesskey** (string) --
- **accesskeyid** (string) --
- **bucket** (string) --
- **container** (string) --
- **credentialsfile** (string) --
- **endpoint** (string) --
- **id** (integer) -- (read only)
- **name** (string) -- DataStore Name (required)

- **region** (string) --
- **secretaccesskey** (string) --
- **storageaccount** (string) --
- **type** (string) -- Site Type (required)
- **url** (string) -- (read only)

GET /datastores/{id}/

API endpoint for managing DataStores.

Parameters:

- **id** (string) --

Status Codes:

- 200 OK --

Response JSON Object:

- **accesskey** (string) --
- **accesskeyid** (string) --
- **bucket** (string) --
- **container** (string) --
- **credentialsfile** (string) --
- **endpoint** (string) --
- **id** (integer) -- (read only)
- **name** (string) -- DataStore Name (required)
- **region** (string) --
- **secretaccesskey** (string) --
- **storageaccount** (string) --
- **type** (string) -- Site Type (required)
- **url** (string) -- (read only)

PATCH /datastores/{id}/

API endpoint for managing DataStores.

Parameters:

- **id** (string) --

Request JSON Object:

- **accesskey** (string) --
- **accesskeyid** (string) --
- **bucket** (string) --
- **container** (string) --
- **credentialsfile** (string) --
- **endpoint** (string) --
- **id** (integer) -- (read only)
- **name** (string) -- DataStore Name (required)
- **region** (string) --
- **secretaccesskey** (string) --
- **storageaccount** (string) --
- **type** (string) -- Site Type (required)
- **url** (string) -- (read only)

Status Codes:

- 200 OK --

Response JSON Object:

- **accesskey** (string) --
- **accesskeyid** (string) --
- **bucket** (string) --
- **container** (string) --
- **credentialsfile** (string) --
- **endpoint** (string) --
- **id** (integer) -- (read only)
- **name** (string) -- DataStore Name (required)
- **region** (string) --
- **secretaccesskey** (string) --
- **storageaccount** (string) --
- **type** (string) -- Site Type (required)
- **url** (string) -- (read only)

DELETE /datastores/{id}/

API endpoint for managing DataStores.

Parameters:

- **id** (string) --

Status Codes:

- [204 No Content](#) --

GET /external_targets/

Model View Set for filebrowser's ExternalTarget model.

Query Parameters:

- **page** (integer) -- A page number within the paginated result set. When not given, first page is retrieved by default.
- **page_size** (integer) -- Number of results to return per page. Page size parameter can be a number between `20` and `100`. For disabling pagination and retrieving all results, `0` should be given. When page size parameter is empty or `<20`, `20` results are returned by default. When page size parameter `>100`, `100` results are returned by default.

Status Codes:

- [200 OK](#) --

Response JSON Object:

- **count** (integer) -- (required)
- **next** (string) --
- **previous** (string) --
- **results[].access_key** (string) -- (read only)
- **results[].access_key_id** (string) -- (read only)
- **results[].advanced_settings** (object) --
- **results[].available_on_sites[]** (integer) --
- **results[].backup_only** (boolean) --
- **results[].backup_site** (integer) -- Site which claimed this target for apbackup
- **results[].bucket** (string) -- Cloud storage bucket name
- **results[].container** (string) -- Azure storage container
- **results[].credentials_file** (string) -- JSON file of Google cloud credentials

- **results[].credentials_json** (string) -- (read only)
- **results[].delete_on_recall** (boolean) -- Delete files from the external storage on recall
- **results[].enabled** (boolean) -- Whether this ngenea target is enabled. Default is true
- **results[].endpoint** (string) -- Hostname of the target storage service
- **results[].id** (integer) -- (read only)
- **results[].is_ready** (string) -- (read only)
- **results[].local_file_regex** (string) -- Regex filter to match files for migration to this target (required)
- **results[].migration_target_folder** (string) -- Filepath of migration target for FS storage type
- **results[].name** (string) -- Ngenea target name (required)
- **results[].port** (integer) -- Port of the target storage service
- **results[].region** (string) --
- **results[].scheme** (string) -- Connection protocol
- **results[].secret_access_key** (string) -- (read only)
- **results[].ssl_verify** (boolean) --
- **results[].storage_account** (string) -- Azure storage account
- **results[].storage_type** (string) -- (required)

POST /external_targets/

Model View Set for filebrowser's ExternalTarget model.

Request JSON Object:

- **access_key** (string) -- Azure-specific access key
- **access_key_id** (string) --
- **advanced_settings** (object) --
- **available_on_sites[]** (integer) --
- **backup_only** (boolean) --
- **backup_site** (integer) -- Site which claimed this target for apbackup
- **bucket** (string) -- Cloud storage bucket name
- **container** (string) -- Azure storage container
- **credentials_file** (string) -- JSON file of Google cloud credentials
- **credentials_json** (object) -- JSON blob of Google Cloud credentials
- **delete_on_recall** (boolean) -- Delete files from the external storage on recall
- **enabled** (boolean) -- Whether this ngenea target is enabled. Default is true
- **endpoint** (string) -- Hostname of the target storage service
- **id** (integer) -- (read only)
- **local_file_regex** (string) -- Regex filter to match files for migration to this target (required)
- **migration_target_folder** (string) -- Filepath of migration target for FS storage type
- **name** (string) -- Ngenea target name (required)
- **port** (integer) -- Port of the target storage service
- **region** (string) --
- **scheme** (string) -- Connection protocol
- **secret_access_key** (string) --
- **ssl_verify** (boolean) --
- **storage_account** (string) -- Azure storage account

- **storage_type** (string) -- (required)

Status Codes:

- 201 Created --

Response JSON Object:

- **access_key** (string) -- Azure-specific access key
- **access_key_id** (string) --
- **advanced_settings** (object) --
- **available_on_sites[]** (integer) --
- **backup_only** (boolean) --
- **backup_site** (integer) -- Site which claimed this target for apbackup
- **bucket** (string) -- Cloud storage bucket name
- **container** (string) -- Azure storage container
- **credentials_file** (string) -- JSON file of Google cloud credentials
- **credentials_json** (object) -- JSON blob of Google Cloud credentials
- **delete_on_recall** (boolean) -- Delete files from the external storage on recall
- **enabled** (boolean) -- Whether this ngenea target is enabled. Default is true
- **endpoint** (string) -- Hostname of the target storage service
- **id** (integer) -- (read only)
- **local_file_regex** (string) -- Regex filter to match files for migration to this target (required)
- **migration_target_folder** (string) -- Filepath of migration target for FS storage type
- **name** (string) -- Ngenea target name (required)
- **port** (integer) -- Port of the target storage service
- **region** (string) --
- **scheme** (string) -- Connection protocol
- **secret_access_key** (string) --
- **ssl_verify** (boolean) --
- **storage_account** (string) -- Azure storage account
- **storage_type** (string) -- (required)

GET /external_targets/{id}/

Model View Set for filebrowser's ExternalTarget model.

Parameters:

- **id** (string) --

Status Codes:

- 200 OK --

Response JSON Object:

- **access_key** (string) -- (read only)
- **access_key_id** (string) -- (read only)
- **advanced_settings** (object) --
- **available_on_sites[]** (integer) --
- **backup_only** (boolean) --
- **backup_site** (integer) -- Site which claimed this target for apbackup
- **bucket** (string) -- Cloud storage bucket name
- **container** (string) -- Azure storage container
- **credentials_file** (string) -- JSON file of Google cloud credentials
- **credentials_json** (string) -- (read only)
- **delete_on_recall** (boolean) -- Delete files from the external storage on recall

- **enabled** (boolean) -- Whether this ngenea target is enabled. Default is true
- **endpoint** (string) -- Hostname of the target storage service
- **id** (integer) -- (read only)
- **is_ready** (string) -- (read only)
- **local_file_regex** (string) -- Regex filter to match files for migration to this target (required)
- **migration_target_folder** (string) -- Filepath of migration target for FS storage type
- **name** (string) -- Ngenea target name (required)
- **port** (integer) -- Port of the target storage service
- **region** (string) --
- **scheme** (string) -- Connection protocol
- **secret_access_key** (string) -- (read only)
- **ssl_verify** (boolean) --
- **storage_account** (string) -- Azure storage account
- **storage_type** (string) -- (required)

PATCH /external_targets/{id}/

Model View Set for filebrowser's ExternalTarget model.

Parameters:

- **id** (string) --

Request JSON Object:

- **access_key** (string) -- Azure-specific access key
- **access_key_id** (string) --
- **advanced_settings** (object) --
- **available_on_sites[]** (integer) --
- **backup_only** (boolean) --
- **backup_site** (integer) -- Site which claimed this target for apbackup
- **bucket** (string) -- Cloud storage bucket name
- **container** (string) -- Azure storage container
- **credentials_file** (string) -- JSON file of Google cloud credentials
- **credentials_json** (object) -- JSON blob of Google Cloud credentials
- **delete_on_recall** (boolean) -- Delete files from the external storage on recall
- **enabled** (boolean) -- Whether this ngenea target is enabled. Default is true
- **endpoint** (string) -- Hostname of the target storage service
- **id** (integer) -- (read only)
- **local_file_regex** (string) -- Regex filter to match files for migration to this target (required)
- **migration_target_folder** (string) -- Filepath of migration target for FS storage type
- **name** (string) -- Ngenea target name (required)
- **port** (integer) -- Port of the target storage service
- **region** (string) --
- **scheme** (string) -- Connection protocol
- **secret_access_key** (string) --
- **ssl_verify** (boolean) --
- **storage_account** (string) -- Azure storage account
- **storage_type** (string) -- (required)

Status Codes:

- 200 OK --

Response JSON Object:

- **access_key** (string) -- Azure-specific access key
- **access_key_id** (string) --
- **advanced_settings** (object) --
- **available_on_sites[]** (integer) --
- **backup_only** (boolean) --
- **backup_site** (integer) -- Site which claimed this target for apbackup
- **bucket** (string) -- Cloud storage bucket name
- **container** (string) -- Azure storage container
- **credentials_file** (string) -- JSON file of Google cloud credentials
- **credentials_json** (object) -- JSON blob of Google Cloud credentials
- **delete_on_recall** (boolean) -- Delete files from the external storage on recall
- **enabled** (boolean) -- Whether this ngenea target is enabled. Default is true
- **endpoint** (string) -- Hostname of the target storage service
- **id** (integer) -- (read only)
- **local_file_regex** (string) -- Regex filter to match files for migration to this target (required)
- **migration_target_folder** (string) -- Filepath of migration target for FS storage type
- **name** (string) -- Ngenea target name (required)
- **port** (integer) -- Port of the target storage service
- **region** (string) --
- **scheme** (string) -- Connection protocol
- **secret_access_key** (string) --
- **ssl_verify** (boolean) --
- **storage_account** (string) -- Azure storage account
- **storage_type** (string) -- (required)

DELETE /external_targets/{id}/

Model View Set for filebrowser's ExternalTarget model.

Parameters:

- **id** (string) --

Status Codes:

- 204 No Content --

GET /external_targets/{id}/files/

Model View Set for filebrowser's ExternalTarget model.

Parameters:

- **id** (string) --

Query Parameters:

- **page** (integer) -- A page number within the paginated result set. When not given, first page is retrieved by default.
- **page_size** (integer) -- Number of results to return per page. Page size parameter can be a number between 20 and 100. For disabling pagination and retrieving all results, 0 should be given. When page size parameter is

empty or <20, 20 results are returned by default. When page size parameter >100, 100 results are returned by default.

- **path** (string) -- Target remote path, value can be a glob eg 'aws/cats*.jpg'.
NOTE: For globs the scan isn't recursive
- **max_items** (integer) -- Maximum number of items to retrieve per page
- **marker** (string) -- Last pagination response marker

Status Codes:

- 200 OK --

Response JSON Object:

- **directories[].name** (string) -- (read only)
- **directories[].path** (string) -- (required)
- **directories[].size** (string) -- (read only)
- **directories[].type** (string) -- (required)
- **files[].name** (string) -- (read only)
- **files[].path** (string) -- (required)
- **files[].size** (string) -- (read only)
- **files[].type** (string) -- (required)
- **marker** (string) -- (required)
- **next_marker** (string) -- (required)
- **path** (string) -- (required)
- **summary** (object) -- (required)

GET /features/

API endpoint for managing feature flags.

Query Parameters:

- **page** (integer) -- A page number within the paginated result set. When not given, first page is retrieved by default.
- **page_size** (integer) -- Number of results to return per page. Page size parameter can be a number between 20 and 100. For disabling pagination and retrieving all results, 0 should be given. When page size parameter is empty or <20, 20 results are returned by default. When page size parameter >100, 100 results are returned by default.

Status Codes:

- 200 OK --

Response JSON Object:

- **count** (integer) -- (required)
- **next** (string) --
- **previous** (string) --
- **results[].description** (string) -- Description of what the feature does (read only)
- **results[].enabled** (boolean) -- Whether the feature has been enabled
- **results[].name** (string) -- Name of the feature (read only)

GET /features/{name}/

API endpoint for managing feature flags.

Parameters:

- **name** (string) --

Status Codes:

- 200 OK --

Response JSON Object:

- **description** (string) -- Description of what the feature does (read only)
- **enabled** (boolean) -- Whether the feature has been enabled
- **name** (string) -- Name of the feature (read only)

PATCH /features/{name}/

API endpoint for managing feature flags.

Parameters:

- **name** (string) --

Request JSON Object:

- **description** (string) -- Description of what the feature does (read only)
- **enabled** (boolean) -- Whether the feature has been enabled
- **name** (string) -- Name of the feature (read only)

Status Codes:

- 200 OK --

Response JSON Object:

- **description** (string) -- Description of what the feature does (read only)
- **enabled** (boolean) -- Whether the feature has been enabled
- **name** (string) -- Name of the feature (read only)

GET /file/

Retrieves list of files under given path for given site.

Query Parameters:

- **page** (integer) -- A page number within the paginated result set. When not given, first page is retrieved by default.
- **page_size** (integer) -- Number of results to return per page. Page size parameter can be a number between 20 and 100. For disabling pagination and retrieving all results, 0 should be given. When page size parameter is empty or <20, 20 results are returned by default. When page size parameter >100, 100 results are returned by default.
- **path** (string) -- Target directory path
- **site** (string) -- Site name
- **details** (boolean) -- Show details of children objects
- **restricted** (boolean) -- Include restricted objects
- **cache_ttl** (integer) -- How long the cache will last for the target path
- **type** (string) -- item type

Status Codes:

- 200 OK --

POST /file/workflow/

Performs a workflow on a list of files

Request JSON Object:

- **discovery** (string) -- Discovery name
- **exclude[]** (string) --
- **fields** (object) --
- **ignore_site_excludes** (boolean) --

- **ignore_site_includes** (boolean) --
- **include[]** (string) --
- **job** (integer) -- Job ID
- **paths[]** (object) --
- **queue** (string) -- Queue name
- **site** (string) -- Site name (required)
- **workflow** (string) -- Workflow name (required)

Status Codes:

- [201 Created](#) --

Response JSON Object:

- **discovery** (string) -- Discovery name
- **exclude[]** (string) --
- **fields** (object) --
- **ignore_site_excludes** (boolean) --
- **ignore_site_includes** (boolean) --
- **include[]** (string) --
- **job** (integer) -- Job ID
- **paths[]** (object) --
- **queue** (string) -- Queue name
- **site** (string) -- Site name (required)
- **workflow** (string) -- Workflow name (required)

GET /filesets/

Retrieve list of filesets on a given site.

Query Parameters:

- **site** (string) -- Site name

Status Codes:

- [200 OK](#) --

GET /filestatustypes/

API endpoint for managing file status types.

Query Parameters:

- **page** (integer) -- A page number within the paginated result set. When not given, first page is retrieved by default.
- **page_size** (integer) -- Number of results to return per page. Page size parameter can be a number between [20](#) and [100](#). For disabling pagination and retrieving all results, [0](#) should be given. When page size parameter is empty or [<20](#), [20](#) results are returned by default. When page size parameter [>100](#), [100](#) results are returned by default.

Status Codes:

- [200 OK](#) --

Response JSON Object:

- **count** (integer) -- (required)
- **next** (string) --
- **previous** (string) --
- **results[].background_color** (string) -- (required)
- **results[].key** (string) -- (required)
- **results[].label** (string) -- (required)

- **results[].text_color** (string) -- (required)
- **results[].url** (string) -- (read only)

GET /filestatustypes/{key}/

API endpoint for managing file status types.

Parameters:

- **key** (string) --

Status Codes:

- **200 OK** --

Response JSON Object:

- **background_color** (string) -- (required)
- **key** (string) -- (required)
- **label** (string) -- (required)
- **text_color** (string) -- (required)
- **url** (string) -- (read only)

PATCH /filestatustypes/{key}/

API endpoint for managing file status types.

Parameters:

- **key** (string) --

Request JSON Object:

- **background_color** (string) -- (required)
- **label** (string) -- (required)
- **text_color** (string) -- (required)

Status Codes:

- **200 OK** --

Response JSON Object:

- **background_color** (string) -- (required)
- **label** (string) -- (required)
- **text_color** (string) -- (required)

GET /filesystems/

Query Parameters:

- **page** (integer) -- A page number within the paginated result set. When not given, first page is retrieved by default.
- **page_size** (integer) -- Number of results to return per page. Page size parameter can be a number between 20 and 100. For disabling pagination and retrieving all results, 0 should be given. When page size parameter is empty or <20, 20 results are returned by default. When page size parameter >100, 100 results are returned by default.

Status Codes:

- **200 OK** --

Response JSON Object:

- **count** (integer) -- (required)
- **next** (string) --
- **previous** (string) --
- **results[].mountpoint** (string) -- Filesystem mount point (required)
- **results[].name** (string) -- Name of the filesystem (required)
- **results[].site** (string) -- Site the filesystem belongs to (required)

GET /filesystems/{id}/

Parameters:

- **id** (string) --

Status Codes:

- 200 OK --

Response JSON Object:

- **mountpoint** (string) -- Filesystem mount point (required)
- **name** (string) -- Name of the filesystem (required)
- **site** (string) -- Site the filesystem belongs to (required)

GET /groups/

API endpoint for managing groups.

Query Parameters:

- **page** (integer) -- A page number within the paginated result set. When not given, first page is retrieved by default.
- **page_size** (integer) -- Number of results to return per page. Page size parameter can be a number between 20 and 100. For disabling pagination and retrieving all results, 0 should be given. When page size parameter is empty or <20, 20 results are returned by default. When page size parameter >100, 100 results are returned by default.

Status Codes:

- 200 OK --

Response JSON Object:

- **count** (integer) -- (required)
- **next** (string) --
- **previous** (string) --
- **results[].description** (string) -- (read only)
- **results[].id** (integer) -- (read only)
- **results[].name** (string) -- (required)
- **results[].nas_group.gid** (integer) -- GID number for the user. This is automatically generated and cannot be changed. (read only)
- **results[].object_permissions[].model** (string) -- (required)
- **results[].object_permissions[].object_pk** (string) -- (required)
- **results[].object_permissions[].permission** (integer) -- (required)
- **results[].permissions[]** (integer) --
- **results[].users[].date_joined** (string) --
- **results[].users[].email** (string) --
- **results[].users[].first_name** (string) --
- **results[].users[].last_login** (string) --
- **results[].users[].last_name** (string) --
- **results[].users[].username** (string) -- Required. 150 characters or fewer. Letters, digits and @/./+/-/_ only. (required)

POST /groups/

API endpoint for managing groups.

Request JSON Object:

- **description** (string) --
- **id** (integer) -- (read only)

- **name** (string) -- (required)
- **nas_group** (boolean) --
- **object_permissions[].model** (string) -- (required)
- **object_permissions[].object_pk** (string) -- (required)
- **object_permissions[].permission** (integer) -- (required)
- **permissions[]** (integer) --
- **users[]** (string) --

Status Codes:

- [201 Created](#) --

Response JSON Object:

- **description** (string) --
- **id** (integer) -- (read only)
- **name** (string) -- (required)
- **nas_group** (boolean) --
- **object_permissions[].model** (string) -- (required)
- **object_permissions[].object_pk** (string) -- (required)
- **object_permissions[].permission** (integer) -- (required)
- **permissions[]** (integer) --
- **users[]** (string) --

GET /groups/{id}/

API endpoint for managing groups.

Parameters:

- **id** (string) --

Status Codes:

- [200 OK](#) --

Response JSON Object:

- **description** (string) -- (read only)
- **id** (integer) -- (read only)
- **name** (string) -- (required)
- **nas_group.gid** (integer) -- GID number for the user. This is automatically generated and cannot be changed. (read only)
- **object_permissions[].model** (string) -- (required)
- **object_permissions[].object_pk** (string) -- (required)
- **object_permissions[].permission** (integer) -- (required)
- **permissions[]** (integer) --
- **users[].date_joined** (string) --
- **users[].email** (string) --
- **users[].first_name** (string) --
- **users[].last_login** (string) --
- **users[].last_name** (string) --
- **users[].username** (string) -- Required. 150 characters or fewer. Letters, digits and @/./+/-/_ only. (required)

PATCH /groups/{id}/

API endpoint for managing groups.

Parameters:

- **id** (string) --

Request JSON Object:

- **description** (string) --
- **id** (integer) -- (read only)
- **name** (string) -- (required)
- **nas_group** (boolean) --
- **object_permissions[].model** (string) -- (required)
- **object_permissions[].object_pk** (string) -- (required)
- **object_permissions[].permission** (integer) -- (required)
- **permissions[]** (integer) --
- **users[]** (string) --

Status Codes:

- 200 OK --

Response JSON Object:

- **description** (string) --
- **id** (integer) -- (read only)
- **name** (string) -- (required)
- **nas_group** (boolean) --
- **object_permissions[].model** (string) -- (required)
- **object_permissions[].object_pk** (string) -- (required)
- **object_permissions[].permission** (integer) -- (required)
- **permissions[]** (integer) --
- **users[]** (string) --

DELETE /groups/{id}/

API endpoint for managing groups.

Parameters:

- **id** (string) --

Status Codes:

- 204 No Content --

GET /health/

Status Codes:

- 200 OK --

GET /ipaddresses/

API endpoint for managing IP addresses.

Query Parameters:

- **page** (integer) -- A page number within the paginated result set. When not given, first page is retrieved by default.
- **page_size** (integer) -- Number of results to return per page. Page size parameter can be a number between 20 and 100. For disabling pagination and retrieving all results, 0 should be given. When page size parameter is empty or <20, 20 results are returned by default. When page size parameter >100, 100 results are returned by default.
- **datastore_id** (integer) -- Data store ID that the IP is assigned to

Status Codes:

- 200 OK --

Response JSON Object:

- **count** (integer) -- (required)

- **next** (string) --
- **previous** (string) --
- **results[].datastore.id** (integer) -- (read only)
- **results[].datastore.name** (string) -- DataStore Name (required)
- **results[].datastore.url** (string) -- (read only)
- **results[].id** (integer) -- (read only)
- **results[].ipaddr** (string) -- IP Address (IPv4) (required)
- **results[].url** (string) -- (read only)

POST /ipaddresses/

API endpoint for managing IP addresses.

Request JSON Object:

- **datastore** (integer) -- (required)
- **id** (integer) -- (read only)
- **ipaddr** (string) -- (required)
- **url** (string) -- (read only)

Status Codes:

- [201 Created](#) --

Response JSON Object:

- **datastore** (integer) -- (required)
- **id** (integer) -- (read only)
- **ipaddr** (string) -- (required)
- **url** (string) -- (read only)

GET /ipaddresses/{id}/

API endpoint for managing IP addresses.

Parameters:

- **id** (string) --

Status Codes:

- [200 OK](#) --

Response JSON Object:

- **datastore.id** (integer) -- (read only)
- **datastore.name** (string) -- DataStore Name (required)
- **datastore.url** (string) -- (read only)
- **id** (integer) -- (read only)
- **ipaddr** (string) -- IP Address (IPv4) (required)
- **url** (string) -- (read only)

PATCH /ipaddresses/{id}/

API endpoint for managing IP addresses.

Parameters:

- **id** (string) --

Request JSON Object:

- **datastore** (integer) -- (required)
- **id** (integer) -- (read only)
- **ipaddr** (string) -- (required)
- **url** (string) -- (read only)

Status Codes:

- 200 OK --

Response JSON Object:

- **datastore** (integer) -- (required)
- **id** (integer) -- (read only)
- **ipaddr** (string) -- (required)
- **url** (string) -- (read only)

DELETE /ipaddresses/{id}/

API endpoint for managing IP addresses.

Parameters:

- **id** (string) --

Status Codes:

- 204 No Content --

GET /jobs/

API endpoint for managing jobs.

Query Parameters:

- **page** (integer) -- A page number within the paginated result set. When not given, first page is retrieved by default.
- **page_size** (integer) -- Number of results to return per page. Page size parameter can be a number between 20 and 100. For disabling pagination and retrieving all results, 0 should be given. When page size parameter is empty or <20, 20 results are returned by default. When page size parameter >100, 100 results are returned by default.
- **created** (string) -- Time period string for filtering jobs by time. Leave null for displaying jobs in all times.
- **created_time_from** (string) -- Start time for filtering jobs by creation time in UTC. Discarded when created parameter is given.
- **created_time_to** (string) -- End time for filtering jobs by creation time in UTC. Discarded when created parameter is given.
- **completed_time_from** (string) -- Start time for filtering jobs by completion time in UTC.
- **completed_time_to** (string) -- End time for filtering jobs by completion time in UTC.
- **workflow** (string) -- Job workflow type
- **state** (array) -- Job states
- **owner_ids** (array) -- Job owner user IDs. Send -1 for the Unknown owner. Send -2 for the System jobs.
- **clientkey_ids** (array) -- Job clientkey IDs
- **schedule_ids** (array) -- Job schedule IDs
- **site** (string) -- Job site name
- **site_id** (integer) -- Job site ID
- **input_paths_prefix** (string) -- Path prefix for the job paths
- **search_keyword** (string) -- Keyword to filter by
- **hide_noop** (boolean) -- Hide successful jobs with no processed files
- **queue** (string) -- Job queue name
- **queue_id** (integer) -- Job queue ID

Status Codes:

- 200 OK --

POST /jobs/

API endpoint for managing jobs.

Status Codes:

- 201 Created --

GET /jobs/jobtype_choices/

Get all custom jobtypes.

Query Parameters:

- **page** (integer) -- A page number within the paginated result set. When not given, first page is retrieved by default.
- **page_size** (integer) -- Number of results to return per page. Page size parameter can be a number between 20 and 100. For disabling pagination and retrieving all results, 0 should be given. When page size parameter is empty or <20, 20 results are returned by default. When page size parameter >100, 100 results are returned by default.

Status Codes:

- 200 OK --

GET /jobs/recent/

Retrieves last N jobs as recent jobs. N = 5 by default (defined in dynamohub/settings/base.py).

Query Parameters:

- **page** (integer) -- A page number within the paginated result set. When not given, first page is retrieved by default.
- **page_size** (integer) -- Number of results to return per page. Page size parameter can be a number between 20 and 100. For disabling pagination and retrieving all results, 0 should be given. When page size parameter is empty or <20, 20 results are returned by default. When page size parameter >100, 100 results are returned by default.

Status Codes:

- 200 OK --

GET /jobs/stats/

API endpoint for managing jobs.

Query Parameters:

- **page** (integer) -- A page number within the paginated result set. When not given, first page is retrieved by default.
- **page_size** (integer) -- Number of results to return per page. Page size parameter can be a number between 20 and 100. For disabling pagination and retrieving all results, 0 should be given. When page size parameter is empty or <20, 20 results are returned by default. When page size parameter >100, 100 results are returned by default.
- **created** (string) -- Time period string for filtering jobs by time. Leave null for displaying jobs in all times.

- **created_time_from** (string) -- Start time for filtering jobs by creation time in UTC. Discarded when `created` parameter is given.
- **created_time_to** (string) -- End time for filtering jobs by creation time in UTC. Discarded when `created` parameter is given.
- **completed_time_from** (string) -- Start time for filtering jobs by completion time in UTC.
- **completed_time_to** (string) -- End time for filtering jobs by completion time in UTC.
- **workflow** (string) -- Job workflow type
- **state** (array) -- Job states
- **owner_ids** (array) -- Job owner user IDs. Send `-1` for the `Unknown` owner. Send `-2` for the `System` jobs.
- **clientkey_ids** (array) -- Job clientkey IDs
- **schedule_ids** (array) -- Job schedule IDs
- **site** (string) -- Job site name
- **site_id** (integer) -- Job site ID
- **input_paths_prefix** (string) -- Path prefix for the job paths
- **search_keyword** (string) -- Keyword to filter by
- **hide_noop** (boolean) -- Hide successful jobs with no processed files
- **queue** (string) -- Job queue name
- **queue_id** (integer) -- Job queue ID

Status Codes:

- `200 OK` --

GET /jobs/{id}/

API endpoint for managing jobs.

Parameters:

- **id** (string) --

Status Codes:

- `200 OK` --

PATCH /jobs/{id}/

API endpoint for managing jobs.

Parameters:

- **id** (string) --

Status Codes:

- `200 OK` --

DELETE /jobs/{id}/

API endpoint for managing jobs.

Parameters:

- **id** (string) --

Status Codes:

- `204 No Content` --

POST /jobs/{id}/cancel/

Cancels the pending and started tasks currently on the MQ for the given ID's job.

Parameters:

- **id** (string) --

Status Codes:

- 201 Created --

GET /jobs/{id}/files/

Retrieves the files related with a job, with their execution status.

Parameters:

- **id** (string) --

Query Parameters:

- **page** (integer) -- A page number within the paginated result set. When not given, first page is retrieved by default.
- **page_size** (integer) -- Number of results to return per page. Page size parameter can be a number between 20 and 100. For disabling pagination and retrieving all results, 0 should be given. When page size parameter is empty or <20, 20 results are returned by default. When page size parameter >100, 100 results are returned by default.
- **category** (string) --

Status Codes:

- 200 OK --

POST /jobs/{id}/pause/

API endpoint to pause the active dag job

Parameters:

- **id** (string) --

Status Codes:

- 201 Created --

GET /jobs/{id}/progress/

Returns the job completion percentage for the given job id

Parameters:

- **id** (string) --

Status Codes:

- 200 OK --

POST /jobs/{id}/resubmit/

Resubmits the job with given id. If the job is not finished yet, this action will not have an effect.

Parameters:

- **id** (string) --

Status Codes:

- 201 Created --

POST /jobs/{id}/resume/

API endpoint to resume the paused dag job

Parameters:

- **id** (string) --

Status Codes:

- 201 Created --

GET /jobs/{id}/statistics/

Returns the statistics of all files for the given job id

Parameters:

- **id** (string) --

Status Codes:

- 200 OK --

GET /jobstats/

API endpoint for managing jobstat.

Query Parameters:

- **page** (integer) -- A page number within the paginated result set. When not given, first page is retrieved by default.
- **page_size** (integer) -- Number of results to return per page. Page size parameter can be a number between 20 and 100. For disabling pagination and retrieving all results, 0 should be given. When page size parameter is empty or <20, 20 results are returned by default. When page size parameter >100, 100 results are returned by default.
- **state** (array) -- Job states
- **job** (integer) -- Job ID
- **path** (string) -- Target path
- **history** (boolean) -- Returns the task history along with job stat

Status Codes:

- 200 OK --

Response JSON Object:

- **count** (integer) -- (required)
- **next** (string) --
- **previous** (string) --
- **results[].code** (string) --
- **results[].id** (integer) -- (read only)
- **results[].job** (integer) -- Related job id (required)
- **results[].message** (string) --
- **results[].path** (string) -- Path to object on filesystem
- **results[].state** (string) --
- **results[].task** (integer) -- Latest related dag task (required)
- **results[].url** (string) -- (read only)

GET /jobstats/{id}/

API endpoint for managing jobstat.

Parameters:

- **id** (string) --

Status Codes:

- 200 OK --

Response JSON Object:

- **code** (string) --
- **id** (integer) -- (read only)
- **job** (integer) -- Related job id (required)
- **message** (string) --
- **path** (string) -- Path to object on filesystem
- **state** (string) --
- **task** (integer) -- Latest related dag task (required)

GET /jobstats/{id}/history/

Returns the task history for the job stat ID

Parameters:

- **id** (string) --

Query Parameters:

- **state** (array) -- Job states
- **job** (integer) -- Job ID
- **path** (string) -- Target path
- **history** (boolean) -- Returns the task history along with job stat

Status Codes:

- 200 OK --

Response JSON Object:

- **code** (string) --
- **id** (integer) -- (read only)
- **job** (integer) -- Related job id (required)
- **message** (string) --
- **path** (string) -- Path to object on filesystem
- **state** (string) --
- **task** (integer) -- Latest related dag task (required)

GET /my-permissions/

API endpoint for viewing the requesting user permissions.

Status Codes:

- 200 OK --

Response JSON Object:

- **[].group_object_permissions.app_label** (string) -- (required)
- **[].group_object_permissions.codename** (string) -- (required)
- **[].group_object_permissions.id** (integer) -- (required)
- **[].group_object_permissions.model** (string) -- (required)
- **[].group_object_permissions.name** (string) -- (required)
- **[].group_object_permissions.object_pk** (string) -- (required)
- **[].group_permissions.app_label** (string) -- (required)
- **[].group_permissions.codename** (string) -- (required)
- **[].group_permissions.id** (integer) -- (read only)
- **[].group_permissions.model** (string) -- (required)
- **[].group_permissions.name** (string) -- (required)
- **[].user_object_permissions.app_label** (string) -- (required)
- **[].user_object_permissions.codename** (string) -- (required)
- **[].user_object_permissions.id** (integer) -- (required)

- **[].user_object_permissions.model** (string) -- (required)
- **[].user_object_permissions.name** (string) -- (required)
- **[].user_object_permissions.object_pk** (string) -- (required)
- **[].user_permissions.app_label** (string) -- (required)
- **[].user_permissions.codename** (string) -- (required)
- **[].user_permissions.id** (integer) -- (read only)
- **[].user_permissions.model** (string) -- (required)
- **[].user_permissions.name** (string) -- (required)

GET /nodes/

Query Parameters:

- **page** (integer) -- A page number within the paginated result set. When not given, first page is retrieved by default.
- **page_size** (integer) -- Number of results to return per page. Page size parameter can be a number between 20 and 100. For disabling pagination and retrieving all results, 0 should be given. When page size parameter is empty or <20, 20 results are returned by default. When page size parameter >100, 100 results are returned by default.

Status Codes:

- 200 OK --

Response JSON Object:

- **count** (integer) -- (required)
- **next** (string) --
- **previous** (string) --
- **results[].has_default_queue** (string) -- (read only)
- **results[].id** (integer) -- (read only)
- **results[].last_heartbeat** (string) -- Time the node sent its last heartbeat event
- **results[].name** (string) -- Hostname for a given worker node (required)
- **results[].online** (string) -- (read only)
- **results[].site** (string) -- (required)
- **results[].url** (string) -- (read only)

GET /nodes/{id}/

Parameters:

- **id** (string) --

Status Codes:

- 200 OK --

Response JSON Object:

- **has_default_queue** (string) -- (read only)
- **id** (integer) -- (read only)
- **last_heartbeat** (string) -- Time the node sent its last heartbeat event
- **name** (string) -- Hostname for a given worker node (required)
- **online** (string) -- (read only)
- **site** (string) -- (required)
- **url** (string) -- (read only)

PATCH /nodes/{id}/

Parameters:

- **id** (string) --

Request JSON Object:

- **has_default_queue** (string) -- (read only)
- **id** (integer) -- (read only)
- **last_heartbeat** (string) -- Time the node sent its last heartbeat event
- **name** (string) -- Hostname for a given worker node (required)
- **online** (string) -- (read only)
- **site** (string) -- (required)
- **url** (string) -- (read only)

Status Codes:

- 200 OK --

Response JSON Object:

- **has_default_queue** (string) -- (read only)
- **id** (integer) -- (read only)
- **last_heartbeat** (string) -- Time the node sent its last heartbeat event
- **name** (string) -- Hostname for a given worker node (required)
- **online** (string) -- (read only)
- **site** (string) -- (required)
- **url** (string) -- (read only)

DELETE /nodes/{id}/

Parameters:

- **id** (string) --

Status Codes:

- 204 No Content --

GET /permissions/

API endpoint for viewing permissions.

Query Parameters:

- **page** (integer) -- A page number within the paginated result set. When not given, first page is retrieved by default.
- **page_size** (integer) -- Number of results to return per page. Page size parameter can be a number between 20 and 100. For disabling pagination and retrieving all results, 0 should be given. When page size parameter is empty or <20, 20 results are returned by default. When page size parameter >100, 100 results are returned by default.

Status Codes:

- 200 OK --

Response JSON Object:

- **count** (integer) -- (required)
- **next** (string) --
- **previous** (string) --
- **results[].app_label** (string) -- (required)
- **results[].codename** (string) -- (required)
- **results[].id** (integer) -- (read only)
- **results[].model** (string) -- (required)
- **results[].name** (string) -- (required)

GET /permissions/{id}/

API endpoint for viewing permissions.

Parameters:

- **id** (string) --

Status Codes:

- 200 OK --

Response JSON Object:

- **app_label** (string) -- (required)
- **codename** (string) -- (required)
- **id** (integer) -- (read only)
- **model** (string) -- (required)
- **name** (string) -- (required)

GET /pki/ca/pub/

Status Codes:

- 200 OK --

GET /pki/lifetime/

Status Codes:

- 200 OK --

Response JSON Object:

- **[].days** (integer) -- The number of days the certificate is valid for.
- **[].updated_at** (string) -- (read only)
- **[].updated_by** (integer) -- (read only)

GET /pki/lifetime/{id}/

Parameters:

- **id** (integer) -- A unique integer value identifying this certificate lifetime.

Status Codes:

- 200 OK --

Response JSON Object:

- **days** (integer) -- The number of days the certificate is valid for.
- **updated_at** (string) -- (read only)
- **updated_by** (integer) -- (read only)

PATCH /pki/lifetime/{id}/

Parameters:

- **id** (integer) -- A unique integer value identifying this certificate lifetime.

Request JSON Object:

- **days** (integer) -- The number of days the certificate is valid for.
- **updated_at** (string) -- (read only)
- **updated_by** (integer) -- (read only)

Status Codes:

- 200 OK --

Response JSON Object:

- **days** (integer) -- The number of days the certificate is valid for.
- **updated_at** (string) -- (read only)
- **updated_by** (integer) -- (read only)

GET /pki/service/private/

Status Codes:

- 200 OK --

GET /pki/service/pub/

Status Codes:

- 200 OK --

POST /pki/site/

Status Codes:

- 201 Created --

GET /policies/

Query Parameters:

- **page** (integer) -- A page number within the paginated result set. When not given, first page is retrieved by default.
- **page_size** (integer) -- Number of results to return per page. Page size parameter can be a number between 20 and 100. For disabling pagination and retrieving all results, 0 should be given. When page size parameter is empty or <20, 20 results are returned by default. When page size parameter >100, 100 results are returned by default.

Status Codes:

- 200 OK --

Response JSON Object:

- **count** (integer) -- (required)
- **next** (string) --
- **previous** (string) --
- **results[].condition_groups[].date_conditions[].age** (string) --
- **results[].condition_groups[].date_conditions[].condition_group** (integer) -- (required)
- **results[].condition_groups[].date_conditions[].date_field** (string) --
- **results[].condition_groups[].date_conditions[].id** (integer) -- (read only)
- **results[].condition_groups[].date_conditions[].include** (boolean) --
- **results[].condition_groups[].date_conditions[].older_or_newer** (string) --
- **results[].condition_groups[].filesize_conditions[].condition_group** (integer) -- (required)
- **results[].condition_groups[].filesize_conditions[].greater_or_less** (string) -- (required)
- **results[].condition_groups[].filesize_conditions[].id** (integer) -- (read only)
- **results[].condition_groups[].filesize_conditions[].include** (boolean) --
- **results[].condition_groups[].filesize_conditions[].size** (integer) -- (required)
- **results[].condition_groups[].filetype_conditions[].condition_group** (integer) -- (required)
- **results[].condition_groups[].filetype_conditions[].filetypes[]** (string) --
- **results[].condition_groups[].filetype_conditions[].id** (integer) -- (read only)
- **results[].condition_groups[].filetype_conditions[].include** (boolean) --
- **results[].condition_groups[].id** (integer) -- (read only)
- **results[].condition_groups[].name** (string) -- (required)
- **results[].condition_groups[].policy** (integer) -- (required)

- **results[].created** (string) -- (read only)
- **results[].enabled** (boolean) --
- **results[].filesystem** (string) --
- **results[].id** (integer) -- (read only)
- **results[].name** (string) -- (required)
- **results[].order.by** (string) --
- **results[].order.id** (integer) -- (read only)
- **results[].order.reverse** (boolean) --
- **results[].policy_type** (string) --
- **results[].schedule.day_of_month** (string) -- Cron Days Of The Month to Run. Use "*" for "all". (Example: "1,15")
- **results[].schedule.day_of_week** (string) -- Cron Days Of The Week to Run. Use "*" for "all". (Example: "0,5")
- **results[].schedule.first_occur** (boolean) -- Whether the first occurrence will be at the stipulated time.
- **results[].schedule.hour** (string) -- Cron Hours to Run. Use "*" for "all". (Example: "8,20")
- **results[].schedule.id** (integer) -- (read only)
- **results[].schedule.minute** (string) -- Cron Minutes to Run. Use "*" for "all". (Example: "0,30")
- **results[].schedule.month_of_year** (string) -- Cron Months Of The Year to Run. Use "*" for "all". (Example: "0,6")
- **results[].schedule.start_date_time** (string) -- Start date time based on the time provided, timezone and if self.first_occur is true.
- **results[].schedule.time** (string) -- Time to start running on the policy's SITE. REMEMBER: The Ngenea hub and the site may be running in different timezones
- **results[].schedule.timezone** (string) -- (required)
- **results[].site** (integer) --
- **results[].spaces[]** (integer) --
- **results[].threads** (integer) -- Number of threads to run policy on
- **results[].triggers** (string) -- (read only)

POST /policies/

Request JSON Object:

- **created** (string) -- (read only)
- **enabled** (boolean) --
- **filesystem** (string) --
- **id** (integer) -- (read only)
- **name** (string) -- (required)
- **order.by** (string) --
- **order.id** (integer) -- (read only)
- **order.reverse** (boolean) --
- **policy_type** (string) --
- **schedule.day_of_month** (string) -- Cron Days Of The Month to Run. Use "*" for "all". (Example: "1,15")
- **schedule.day_of_week** (string) -- Cron Days Of The Week to Run. Use "*" for "all". (Example: "0,5")
- **schedule.first_occur** (boolean) -- Whether the first occurrence will be at the stipulated time.

- **schedule.hour** (string) -- Cron Hours to Run. Use "*" for "all". (Example: "8,20")
- **schedule.id** (integer) -- (read only)
- **schedule.minute** (string) -- Cron Minutes to Run. Use "*" for "all". (Example: "0,30")
- **schedule.month_of_year** (string) -- Cron Months Of The Year to Run. Use "*" for "all". (Example: "0,6")
- **schedule.time** (string) -- Time to start running on the policy's SITE. REMEMBER: The Ngenea hub and the site may be running in different timezones
- **site** (integer) --
- **spaces[]** (integer) --
- **threads** (integer) -- Number of threads to run policy on
- **triggers[].is_cloud** (boolean) --
- **triggers[].lower_threshold** (number) --
- **triggers[].max_utilisation** (number) --
- **triggers[].pool_name** (string) -- (required)
- **triggers[].premigrate_threshold** (number) --
- **triggers[].upper_threshold** (number) --

Status Codes:

- [201 Created](#) --

Response JSON Object:

- **created** (string) -- (read only)
- **enabled** (boolean) --
- **filesystem** (string) --
- **id** (integer) -- (read only)
- **name** (string) -- (required)
- **order** (integer) --
- **policy_type** (string) --
- **schedule.day_of_month** (string) -- Cron Days Of The Month to Run. Use "*" for "all". (Example: "1,15")
- **schedule.day_of_week** (string) -- Cron Days Of The Week to Run. Use "*" for "all". (Example: "0,5")
- **schedule.first_occur** (boolean) -- Whether the first occurrence will be at the stipulated time.
- **schedule.hour** (string) -- Cron Hours to Run. Use "*" for "all". (Example: "8,20")
- **schedule.id** (integer) -- (read only)
- **schedule.minute** (string) -- Cron Minutes to Run. Use "*" for "all". (Example: "0,30")
- **schedule.month_of_year** (string) -- Cron Months Of The Year to Run. Use "*" for "all". (Example: "0,6")
- **schedule.time** (string) -- Time to start running on the policy's SITE. REMEMBER: The Ngenea hub and the site may be running in different timezones
- **site** (integer) --
- **spaces[]** (integer) --
- **threads** (integer) -- Number of threads to run policy on
- **triggers[].id** (integer) -- (read only)
- **triggers[].is_cloud** (boolean) --

- **triggers[].lower_threshold** (number) --
- **triggers[].max_utilisation** (number) --
- **triggers[].pool_name** (string) -- (required)
- **triggers[].premigrate_threshold** (number) --
- **triggers[].upper_threshold** (number) --

GET /policies/{id}/

Parameters:

- **id** (string) --

Status Codes:

- **200 OK** --

Response JSON Object:

- **condition_groups[].date_conditions[].age** (string) --
- **condition_groups[].date_conditions[].condition_group** (integer) -- (required)
- **condition_groups[].date_conditions[].date_field** (string) --
- **condition_groups[].date_conditions[].id** (integer) -- (read only)
- **condition_groups[].date_conditions[].include** (boolean) --
- **condition_groups[].date_conditions[].older_or_newer** (string) --
- **condition_groups[].filesize_conditions[].condition_group** (integer) -- (required)
- **condition_groups[].filesize_conditions[].greater_or_less** (string) -- (required)
- **condition_groups[].filesize_conditions[].id** (integer) -- (read only)
- **condition_groups[].filesize_conditions[].include** (boolean) --
- **condition_groups[].filesize_conditions[].size** (integer) -- (required)
- **condition_groups[].filetype_conditions[].condition_group** (integer) -- (required)
- **condition_groups[].filetype_conditions[].filetypes[]** (string) --
- **condition_groups[].filetype_conditions[].id** (integer) -- (read only)
- **condition_groups[].filetype_conditions[].include** (boolean) --
- **condition_groups[].id** (integer) -- (read only)
- **condition_groups[].name** (string) -- (required)
- **condition_groups[].policy** (integer) -- (required)
- **created** (string) -- (read only)
- **enabled** (boolean) --
- **filesystem** (string) --
- **id** (integer) -- (read only)
- **name** (string) -- (required)
- **order.by** (string) --
- **order.id** (integer) -- (read only)
- **order.reverse** (boolean) --
- **policy_type** (string) --
- **schedule.day_of_month** (string) -- Cron Days Of The Month to Run. Use "*" for "all". (Example: "1,15")
- **schedule.day_of_week** (string) -- Cron Days Of The Week to Run. Use "*" for "all". (Example: "0,5")
- **schedule.first_occur** (boolean) -- Whether the first occurrence will be at the stipulated time.

- **schedule.hour** (string) -- Cron Hours to Run. Use "*" for "all". (Example: "8,20")
- **schedule.id** (integer) -- (read only)
- **schedule.minute** (string) -- Cron Minutes to Run. Use "*" for "all". (Example: "0,30")
- **schedule.month_of_year** (string) -- Cron Months Of The Year to Run. Use "*" for "all". (Example: "0,6")
- **schedule.start_date_time** (string) -- Start date time based on the time provided, timezone and if self.first_occur is true.
- **schedule.time** (string) -- Time to start running on the policy's SITE. REMEMBER: The Ngenea hub and the site may be running in different timezones
- **schedule.timezone** (string) -- (required)
- **site** (integer) --
- **spaces[]** (integer) --
- **threads** (integer) -- Number of threads to run policy on
- **triggers** (string) -- (read only)

PATCH /policies/{id}/

Parameters:

- **id** (string) --

Request JSON Object:

- **created** (string) -- (read only)
- **enabled** (boolean) --
- **filesystem** (string) --
- **id** (integer) -- (read only)
- **name** (string) -- (required)
- **order.by** (string) --
- **order.id** (integer) -- (read only)
- **order.reverse** (boolean) --
- **policy_type** (string) --
- **schedule.day_of_month** (string) -- Cron Days Of The Month to Run. Use "*" for "all". (Example: "1,15")
- **schedule.day_of_week** (string) -- Cron Days Of The Week to Run. Use "*" for "all". (Example: "0,5")
- **schedule.first_occur** (boolean) -- Whether the first occurrence will be at the stipulated time.
- **schedule.hour** (string) -- Cron Hours to Run. Use "*" for "all". (Example: "8,20")
- **schedule.id** (integer) -- (read only)
- **schedule.minute** (string) -- Cron Minutes to Run. Use "*" for "all". (Example: "0,30")
- **schedule.month_of_year** (string) -- Cron Months Of The Year to Run. Use "*" for "all". (Example: "0,6")
- **schedule.time** (string) -- Time to start running on the policy's SITE. REMEMBER: The Ngenea hub and the site may be running in different timezones
- **site** (integer) --
- **spaces[]** (integer) --
- **threads** (integer) -- Number of threads to run policy on

- **triggers[].is_cloud** (boolean) --
- **triggers[].lower_threshold** (number) --
- **triggers[].max_utilisation** (number) --
- **triggers[].pool_name** (string) -- (required)
- **triggers[].premigrate_threshold** (number) --
- **triggers[].upper_threshold** (number) --

Status Codes:

- [201 Created](#) --

Response JSON Object:

- **created** (string) -- (read only)
- **enabled** (boolean) --
- **filesystem** (string) --
- **id** (integer) -- (read only)
- **name** (string) -- (required)
- **order** (integer) --
- **policy_type** (string) --
- **schedule.day_of_month** (string) -- Cron Days Of The Month to Run. Use "*" for "all". (Example: "1,15")
- **schedule.day_of_week** (string) -- Cron Days Of The Week to Run. Use "*" for "all". (Example: "0,5")
- **schedule.first_occur** (boolean) -- Whether the first occurrence will be at the stipulated time.
- **schedule.hour** (string) -- Cron Hours to Run. Use "*" for "all". (Example: "8,20")
- **schedule.id** (integer) -- (read only)
- **schedule.minute** (string) -- Cron Minutes to Run. Use "*" for "all". (Example: "0,30")
- **schedule.month_of_year** (string) -- Cron Months Of The Year to Run. Use "*" for "all". (Example: "0,6")
- **schedule.time** (string) -- Time to start running on the policy's SITE. REMEMBER: The Ngenea hub and the site may be running in different timezones
- **site** (integer) --
- **spaces[]** (integer) --
- **threads** (integer) -- Number of threads to run policy on
- **triggers[].id** (integer) -- (read only)
- **triggers[].is_cloud** (boolean) --
- **triggers[].lower_threshold** (number) --
- **triggers[].max_utilisation** (number) --
- **triggers[].pool_name** (string) -- (required)
- **triggers[].premigrate_threshold** (number) --
- **triggers[].upper_threshold** (number) --

DELETE /policies/{id}/

Parameters:

- **id** (string) --

Status Codes:

- [204 No Content](#) --

POST /policies/{id}/run/

Parameters:

- **id** (string) --

Status Codes:

- [202 Accepted](#) -- Task id of the task fired for the associated request.

GET /queues/

Query Parameters:

- **page** (integer) -- A page number within the paginated result set. When not given, first page is retrieved by default.
- **page_size** (integer) -- Number of results to return per page. Page size parameter can be a number between [20](#) and [100](#). For disabling pagination and retrieving all results, [0](#) should be given. When page size parameter is empty or <20 , 20 results are returned by default. When page size parameter >100 , 100 results are returned by default.

Status Codes:

- [200 OK](#) --

Response JSON Object:

- **count** (integer) -- (required)
- **next** (string) --
- **previous** (string) --
- **results[].id** (integer) -- (read only)
- **results[].label** (string) -- UI facing label for the queue
- **results[].last_heartbeat** (string) --
- **results[].name** (string) -- Queue name. Must match `[a-zA-Z0-9-_]+` (required)
- **results[].nodes** (string) -- (read only)
- **results[].site** (string) -- (read only)

GET /queues/{id}/

Parameters:

- **id** (string) --

Status Codes:

- [200 OK](#) --

Response JSON Object:

- **id** (integer) -- (read only)
- **label** (string) -- UI facing label for the queue
- **last_heartbeat** (string) --
- **name** (string) -- Queue name. Must match `[a-zA-Z0-9-_]+` (required)
- **nodes** (string) -- (read only)
- **site** (string) -- (read only)

GET /schedules/

Query Parameters:

- **page** (integer) -- A page number within the paginated result set. When not given, first page is retrieved by default.
- **page_size** (integer) -- Number of results to return per page. Page size parameter can be a number between [20](#) and [100](#). For disabling pagination and retrieving all results, [0](#) should be given. When page size parameter is empty or <20 , 20 results are returned by default. When page size parameter >100 , 100 results are returned by default.

Status Codes:

- 200 OK --

Response JSON Object:

- **count** (integer) -- (required)
- **next** (string) --
- **previous** (string) --
- **results[].day_of_month** (string) -- The day setting for the cron schedule
- **results[].day_of_week** (string) -- The week setting for the cron schedule
- **results[].discovery** (string) --
- **results[].discovery_options** (object) --
- **results[].enabled** (boolean) -- If the schedule should be enabled
- **results[].hour** (string) -- The hour setting for the cron schedule
- **results[].id** (integer) -- (read only)
- **results[].managed_paths** (object) -- Path of managed filesystem elements
- **results[].minute** (string) -- The minute setting for the cron schedule
- **results[].month_of_year** (string) -- The month setting for the cron schedule
- **results[].name** (string) -- Schedule Name (required)
- **results[].site** (string) -- (required)
- **results[].space** (string) -- (required)
- **results[].url** (string) -- (read only)

POST /schedules/

Request JSON Object:

- **day_of_month** (string) -- The day setting for the cron schedule
- **day_of_week** (string) -- The week setting for the cron schedule
- **discovery** (string) --
- **discovery_options** (object) --
- **enabled** (boolean) -- If the schedule should be enabled
- **hour** (string) -- The hour setting for the cron schedule
- **id** (integer) -- (read only)
- **managed_paths** (object) -- Path of managed filesystem elements
- **minute** (string) -- The minute setting for the cron schedule
- **month_of_year** (string) -- The month setting for the cron schedule
- **name** (string) -- Schedule Name (required)
- **site** (string) -- (required)
- **url** (string) -- (read only)

Status Codes:

- 201 Created --

Response JSON Object:

- **day_of_month** (string) -- The day setting for the cron schedule
- **day_of_week** (string) -- The week setting for the cron schedule
- **discovery** (string) --
- **discovery_options** (object) --
- **enabled** (boolean) -- If the schedule should be enabled
- **hour** (string) -- The hour setting for the cron schedule
- **id** (integer) -- (read only)
- **managed_paths** (object) -- Path of managed filesystem elements
- **minute** (string) -- The minute setting for the cron schedule
- **month_of_year** (string) -- The month setting for the cron schedule
- **name** (string) -- Schedule Name (required)

- **site** (string) -- (required)
- **url** (string) -- (read only)

GET /schedules/{id}/

Parameters:

- **id** (string) --

Status Codes:

- 200 OK --

Response JSON Object:

- **day_of_month** (string) -- The day setting for the cron schedule
- **day_of_week** (string) -- The week setting for the cron schedule
- **discovery** (string) --
- **discovery_options** (object) --
- **enabled** (boolean) -- If the schedule should be enabled
- **hour** (string) -- The hour setting for the cron schedule
- **id** (integer) -- (read only)
- **managed_paths** (object) -- Path of managed filesystem elements
- **minute** (string) -- The minute setting for the cron schedule
- **month_of_year** (string) -- The month setting for the cron schedule
- **name** (string) -- Schedule Name (required)
- **site** (string) -- (required)
- **space** (string) -- (required)
- **url** (string) -- (read only)

PATCH /schedules/{id}/

Parameters:

- **id** (string) --

Request JSON Object:

- **day_of_month** (string) -- The day setting for the cron schedule
- **day_of_week** (string) -- The week setting for the cron schedule
- **discovery** (string) --
- **discovery_options** (object) --
- **enabled** (boolean) -- If the schedule should be enabled
- **hour** (string) -- The hour setting for the cron schedule
- **id** (integer) -- (read only)
- **managed_paths** (object) -- Path of managed filesystem elements
- **minute** (string) -- The minute setting for the cron schedule
- **month_of_year** (string) -- The month setting for the cron schedule
- **name** (string) -- Schedule Name (required)
- **site** (string) -- (required)

Status Codes:

- 200 OK --

Response JSON Object:

- **day_of_month** (string) -- The day setting for the cron schedule
- **day_of_week** (string) -- The week setting for the cron schedule
- **discovery** (string) --
- **discovery_options** (object) --
- **enabled** (boolean) -- If the schedule should be enabled
- **hour** (string) -- The hour setting for the cron schedule
- **id** (integer) -- (read only)

- **managed_paths** (object) -- Path of managed filesystem elements
- **minute** (string) -- The minute setting for the cron schedule
- **month_of_year** (string) -- The month setting for the cron schedule
- **name** (string) -- Schedule Name (required)
- **site** (string) -- (required)

DELETE /schedules/{id}/

Parameters:

- **id** (string) --

Status Codes:

- [204 No Content](#) --

GET /schedules/{parent_lookup_schedule}/workflows/

Parameters:

- **parent_lookup_schedule** (string) --

Query Parameters:

- **page** (integer) -- A page number within the paginated result set. When not given, first page is retrieved by default.
- **page_size** (integer) -- Number of results to return per page. Page size parameter can be a number between 20 and 100. For disabling pagination and retrieving all results, 0 should be given. When page size parameter is empty or <20, 20 results are returned by default. When page size parameter >100, 100 results are returned by default.

Status Codes:

- [200 OK](#) --

Response JSON Object:

- **count** (integer) -- (required)
- **next** (string) --
- **previous** (string) --
- **results[].fields** (object) -- Mapping of path to operation for task usage
- **results[].id** (integer) -- (read only)
- **results[].queue** (string) -- (required)
- **results[].site** (string) -- (required)
- **results[].url** (string) -- (read only)
- **results[].workflow** (string) -- (required)

POST /schedules/{parent_lookup_schedule}/workflows/

Parameters:

- **parent_lookup_schedule** (string) --

Request JSON Object:

- **fields** (object) -- Mapping of path to operation for task usage
- **id** (integer) -- (read only)
- **queue** (string) -- Queue name
- **site** (string) -- (required)
- **workflow** (string) -- (required)

Status Codes:

- [201 Created](#) --

Response JSON Object:

- **fields** (object) -- Mapping of path to operation for task usage
- **id** (integer) -- (read only)

- **queue** (string) -- Queue name
- **site** (string) -- (required)
- **workflow** (string) -- (required)

GET /schedules/{parent_lookup_schedule}/workflows/{id}/

Parameters:

- **parent_lookup_schedule** (string) --
- **id** (string) --

Status Codes:

- 200 OK --

Response JSON Object:

- **fields** (object) -- Mapping of path to operation for task usage
- **id** (integer) -- (read only)
- **queue** (string) -- (required)
- **site** (string) -- (required)
- **url** (string) -- (read only)
- **workflow** (string) -- (required)

PATCH /schedules/{parent_lookup_schedule}/workflows/{id}/

Parameters:

- **parent_lookup_schedule** (string) --
- **id** (string) --

Request JSON Object:

- **fields** (object) -- Mapping of path to operation for task usage
- **id** (integer) -- (read only)
- **queue** (string) -- (required)
- **site** (string) -- (required)
- **url** (string) -- (read only)
- **workflow** (string) -- (required)

Status Codes:

- 200 OK --

Response JSON Object:

- **fields** (object) -- Mapping of path to operation for task usage
- **id** (integer) -- (read only)
- **queue** (string) -- (required)
- **site** (string) -- (required)
- **url** (string) -- (read only)
- **workflow** (string) -- (required)

DELETE /schedules/{parent_lookup_schedule}/workflows/{id}/

Parameters:

- **parent_lookup_schedule** (string) --
- **id** (string) --

Status Codes:

- 204 No Content --

POST /search/

API endpoint for file search

Request JSON Object:

- **filters** (object) -- Metadata filters to apply to search
- **merge** (boolean) -- Whether matching files should be merged
- **metadata_fields** (object) -- Available metadata fields from this search
- **path** (string) -- Directory to search (required)
- **recursive** (boolean) -- Search the target path recursively
- **sites[]** (string) --

Status Codes:

- [201 Created](#) --

Response JSON Object:

- **filters** (object) -- Metadata filters to apply to search
- **merge** (boolean) -- Whether matching files should be merged
- **metadata_fields** (object) -- Available metadata fields from this search
- **path** (string) -- Directory to search (required)
- **recursive** (boolean) -- Search the target path recursively
- **sites[]** (string) --

PATCH /search/metadata/

API endpoint for file search

Request JSON Object:

- **id** (integer) -- (read only)
- **url** (string) -- (read only)

Status Codes:

- [200 OK](#) --

Response JSON Object:

- **id** (integer) -- (read only)
- **url** (string) -- (read only)

GET /search/metadata_fields/

API endpoint for file search

Query Parameters:

- **page** (integer) -- A page number within the paginated result set. When not given, first page is retrieved by default.
- **page_size** (integer) -- Number of results to return per page. Page size parameter can be a number between [20](#) and [100](#). For disabling pagination and retrieving all results, [0](#) should be given. When page size parameter is empty or <20 , 20 results are returned by default. When page size parameter >100 , 100 results are returned by default.

Status Codes:

- [200 OK](#) --

Response JSON Object:

- **count** (integer) -- (required)
- **next** (string) --
- **previous** (string) --
- **results[].id** (integer) -- (read only)
- **results[].url** (string) -- (read only)

GET /search/{id}/

Get paginated results for a given search id.

Parameters:

- **id** (string) --

Query Parameters:

- **page** (integer) -- A page number within the paginated result set. When not given, first page is retrieved by default.
- **page_size** (integer) -- Number of results to return per page. Page size parameter can be a number between 20 and 100. For disabling pagination and retrieving all results, 0 should be given. When page size parameter is empty or <20, 20 results are returned by default. When page size parameter >100, 100 results are returned by default.
- **sort** (string) -- One or more fields to sort results by
- **group_by_name** (boolean) --
When this url parameter is 'True' results are merged based on matching file name. e.g.

```
[
  {
    data: [
      {site: site1, path: /mmfs1, ...},
      {site: site2, path: /mmfs1, ...}
    ],
  },
  {
    foo: [
      {site: site1, path: /mmfs1/data, ...}
      # not present on site2
    ]
  }
]
```

Status Codes:

- 200 OK --

Response JSON Object:

- **href** (string) -- (read only)
- **metadata** (object) -- File metadata
- **name** (string) -- Directory or file name (required)
- **path** (string) -- Directory or file path (required)
- **proxies** (object) -- File proxies (thumbnails, etc.)
- **site** (string) -- Site Name

DELETE /search/{id}/

API endpoint for file search

Parameters:

- **id** (string) --

Status Codes:

- 204 No Content --

GET /servers/

Query Parameters:

- **page** (integer) -- A page number within the paginated result set. When not given, first page is retrieved by default.
- **page_size** (integer) -- Number of results to return per page. Page size parameter can be a number between 20 and 100. For disabling pagination and retrieving all results, 0 should be given. When page size parameter is empty or <20, 20 results are returned by default. When page size parameter >100, 100 results are returned by default.

Status Codes:

- 200 OK --

Response JSON Object:

- **count** (integer) -- (required)
- **next** (string) --
- **previous** (string) --
- **results[].list_of_interfaces** (object) -- List of interfaces of this server
- **results[].name** (string) -- Server name (required)
- **results[].site** (integer) -- Site correlated with this server (required)

GET /servers/{id}/

Parameters:

- **id** (string) --

Status Codes:

- 200 OK --

Response JSON Object:

- **list_of_interfaces** (object) -- List of interfaces of this server
- **name** (string) -- Server name (required)
- **site** (integer) -- Site correlated with this server (required)

GET /settings/

API to get a list of global settings.

Status Codes:

- 200 OK --

Response JSON Object:

- **[].desc** (string) -- (read only)
- **[].id** (integer) -- (read only)
- **[].key** (string) -- (required)
- **[].value** (object) --

PATCH /settings/

API to set a setting or a group of global settings. A group of settings should be sent in key-value schema. Requested data will be stored in the database, and then sent to each site. Terminology Setting - An individual setting e.g. ngenea:realm

Notes on use:

- Non-global settings cannot be set on this endpoint either alone or with global settings
- Setting a key to None will delete it from the database and set null on all active sites

e.g. {"values": [{"key": "ngeneahsm_targets:devaws:Storage", "value": None}]}

Request JSON Object:

- **values[].desc** (string) -- (read only)
- **values[].id** (integer) -- (read only)
- **values[].key** (string) -- (required)
- **values[].value** (object) --

Status Codes:

- **202 Accepted** -- Task id of the task fired for the associated request per site.

POST /settings/sync/

API to sync global settings on each site Stored data will be sent to the given site or to each site if not given.

Query Parameters:

- **site** (string) -- Site to run the sync operation on if provided, otherwise all sites

Status Codes:

- **202 Accepted** -- Task id of the task fired for the associated request per site.

GET /settingstasks/

API endpoint for viewing tasks.

Query Parameters:

- **page** (integer) -- A page number within the paginated result set. When not given, first page is retrieved by default.
- **page_size** (integer) -- Number of results to return per page. Page size parameter can be a number between 20 and 100. For disabling pagination and retrieving all results, 0 should be given. When page size parameter is empty or <20, 20 results are returned by default. When page size parameter >100, 100 results are returned by default.
- **tasktype** (string) -- Task type
- **state** (array) -- Task states
- **job_id** (integer) -- Job ID

Status Codes:

- **200 OK** --

Response JSON Object:

- **count** (integer) -- (required)
- **next** (string) --
- **previous** (string) --
- **results[].id** (integer) -- (read only)
- **results[].job** (integer) --
- **results[].site** (string) -- (required)
- **results[].started** (string) -- Time that the task started running
- **results[].state** (string) --
- **results[].task_id** (string) -- (read only)
- **results[].tasktype** (string) -- (read only)
- **results[].url** (string) -- (read only)

GET /settingstasks/{id}/

API endpoint for viewing tasks.

Parameters:

- **id** (string) --

Status Codes:

- **200 OK** --

Response JSON Object:

- **completed** (string) -- Time of the task completion
- **friendly_name** (string) -- Text to be used to describe what this task is doing.
- **id** (integer) -- (read only)
- **job** (integer) --
- **request** (string) -- (read only)
- **results** (string) -- (read only)
- **runtime** (string) -- (read only)
- **site** (string) -- (required)
- **started** (string) -- Time that the task started running
- **state** (string) --
- **task_id** (string) -- (read only)
- **tasktype** (string) -- (read only)
- **url** (string) -- (read only)

GET /shares/nfs/

Status Codes:

- **200 OK** --

Response JSON Object:

- **[].clients[].advanced_settings** (object) --
- **[].clients[].anonymous_group_id** (integer) --
- **[].clients[].anonymous_user_id** (integer) --
- **[].clients[].asynchronous** (boolean) --
- **[].clients[].force_all_user_id** (boolean) --
- **[].clients[].force_root_user_id** (boolean) --
- **[].clients[].hosts** (string) -- (required)
- **[].clients[].id** (integer) -- (read only)
- **[].clients[].insecure_ports** (boolean) --
- **[].clients[].nfs_share** (integer) -- (required)
- **[].clients[].read_only** (boolean) --
- **[].clients[].subtree_check** (boolean) --
- **[].clients[].uuid** (string) -- (read only)
- **[].clients[].write_delay** (boolean) --
- **[].fsid** (integer) --
- **[].id** (integer) -- (read only)
- **[].name** (string) -- Note: This name is for identification only. NFSClients don't have names in remote representations
- **[].path** (string) --
- **[].space** (integer) -- (required)
- **[].uuid** (string) -- (read only)

POST /shares/nfs/

Request JSON Object:

- **clients[].advanced_settings** (object) --

- **clients[].anonymous_group_id** (integer) --
- **clients[].anonymous_user_id** (integer) --
- **clients[].asynchronous** (boolean) --
- **clients[].force_all_user_id** (boolean) --
- **clients[].force_root_user_id** (boolean) --
- **clients[].hosts** (string) -- (required)
- **clients[].id** (integer) -- (read only)
- **clients[].insecure_ports** (boolean) --
- **clients[].nfs_share** (integer) -- (required)
- **clients[].read_only** (boolean) --
- **clients[].subtree_check** (boolean) --
- **clients[].uuid** (string) -- (read only)
- **clients[].write_delay** (boolean) --
- **fsid** (integer) --
- **id** (integer) -- (read only)
- **name** (string) -- Note: This name is for identification only. NFSClients don't have names in remote representations
- **path** (string) --
- **space** (integer) -- (required)
- **uuid** (string) -- (read only)

Status Codes:

- [201 Created](#) --

Response JSON Object:

- **clients[].advanced_settings** (object) --
- **clients[].anonymous_group_id** (integer) --
- **clients[].anonymous_user_id** (integer) --
- **clients[].asynchronous** (boolean) --
- **clients[].force_all_user_id** (boolean) --
- **clients[].force_root_user_id** (boolean) --
- **clients[].hosts** (string) -- (required)
- **clients[].id** (integer) -- (read only)
- **clients[].insecure_ports** (boolean) --
- **clients[].nfs_share** (integer) -- (required)
- **clients[].read_only** (boolean) --
- **clients[].subtree_check** (boolean) --
- **clients[].uuid** (string) -- (read only)
- **clients[].write_delay** (boolean) --
- **fsid** (integer) --
- **id** (integer) -- (read only)
- **name** (string) -- Note: This name is for identification only. NFSClients don't have names in remote representations
- **path** (string) --
- **space** (integer) -- (required)
- **uuid** (string) -- (read only)

GET /shares/nfs/{id}/

Parameters:

- **id** (integer) -- A unique integer value identifying this nfs share.

Status Codes:

- [200 OK](#) --

Response JSON Object:

- **clients[].advanced_settings** (object) --
- **clients[].anonymous_group_id** (integer) --
- **clients[].anonymous_user_id** (integer) --
- **clients[].asynchronous** (boolean) --
- **clients[].force_all_user_id** (boolean) --
- **clients[].force_root_user_id** (boolean) --
- **clients[].hosts** (string) -- (required)
- **clients[].id** (integer) -- (read only)
- **clients[].insecure_ports** (boolean) --
- **clients[].nfs_share** (integer) -- (required)
- **clients[].read_only** (boolean) --
- **clients[].subtree_check** (boolean) --
- **clients[].uuid** (string) -- (read only)
- **clients[].write_delay** (boolean) --
- **fsid** (integer) --
- **id** (integer) -- (read only)
- **name** (string) -- Note: This name is for identification only. NFSClients don't have names in remote representations
- **path** (string) --
- **space** (integer) -- (required)
- **uuid** (string) -- (read only)

PATCH /shares/nfs/{id}/

Parameters:

- **id** (integer) -- A unique integer value identifying this nfs share.

Request JSON Object:

- **clients[].advanced_settings** (object) --
- **clients[].anonymous_group_id** (integer) --
- **clients[].anonymous_user_id** (integer) --
- **clients[].asynchronous** (boolean) --
- **clients[].force_all_user_id** (boolean) --
- **clients[].force_root_user_id** (boolean) --
- **clients[].hosts** (string) -- (required)
- **clients[].id** (integer) -- (read only)
- **clients[].insecure_ports** (boolean) --
- **clients[].nfs_share** (integer) -- (required)
- **clients[].read_only** (boolean) --
- **clients[].subtree_check** (boolean) --
- **clients[].uuid** (string) -- (read only)
- **clients[].write_delay** (boolean) --
- **fsid** (integer) --
- **id** (integer) -- (read only)
- **name** (string) -- Note: This name is for identification only. NFSClients don't have names in remote representations
- **path** (string) --
- **space** (integer) -- (required)
- **uuid** (string) -- (read only)

Status Codes:

- 200 OK --

Response JSON Object:

- **clients[].advanced_settings** (object) --
- **clients[].anonymous_group_id** (integer) --
- **clients[].anonymous_user_id** (integer) --
- **clients[].asynchronous** (boolean) --
- **clients[].force_all_user_id** (boolean) --
- **clients[].force_root_user_id** (boolean) --
- **clients[].hosts** (string) -- (required)
- **clients[].id** (integer) -- (read only)
- **clients[].insecure_ports** (boolean) --
- **clients[].nfs_share** (integer) -- (required)
- **clients[].read_only** (boolean) --
- **clients[].subtree_check** (boolean) --
- **clients[].uuid** (string) -- (read only)
- **clients[].write_delay** (boolean) --
- **fsid** (integer) --
- **id** (integer) -- (read only)
- **name** (string) -- Note: This name is for identification only. NFSClients don't have names in remote representations
- **path** (string) --
- **space** (integer) -- (required)
- **uuid** (string) -- (read only)

DELETE /shares/nfs/{id}/

Parameters:

- **id** (integer) -- A unique integer value identifying this nfs share.

Status Codes:

- [204 No Content](#) --

GET /shares/samba/

Status Codes:

- [200 OK](#) --

Response JSON Object:

- **[].admin_users** (object) --
- **[].advanced_settings** (object) --
- **[].allowed_users** (object) --
- **[].create_mask** (string) --
- **[].directory_mask** (string) --
- **[].force_create_mode** (string) --
- **[].force_directory_mode** (string) --
- **[].guest_ok** (boolean) --
- **[].hosts_allow** (object) --
- **[].hosts_deny** (object) --
- **[].hsm_support** (boolean) --
- **[].id** (integer) -- (read only)
- **[].multi_thread_reads** (boolean) --
- **[].multi_thread_writes** (boolean) --
- **[].name** (string) -- (required)
- **[].on_sites** (object) --
- **[].path** (string) --

- **[].read_only** (boolean) -- (required)
- **[].root_share_locking** (boolean) --
- **[].space** (integer) -- (required)
- **[].uuid** (string) -- (read only)
- **[].visible** (boolean) -- (required)

POST /shares/samba/

Request JSON Object:

- **admin_users** (object) --
- **advanced_settings** (object) --
- **allowed_users** (object) --
- **create_mask** (string) --
- **directory_mask** (string) --
- **force_create_mode** (string) --
- **force_directory_mode** (string) --
- **guest_ok** (boolean) --
- **hosts_allow** (object) --
- **hosts_deny** (object) --
- **hsm_support** (boolean) --
- **id** (integer) -- (read only)
- **multi_thread_reads** (boolean) --
- **multi_thread_writes** (boolean) --
- **name** (string) -- (required)
- **on_sites** (object) --
- **path** (string) --
- **read_only** (boolean) -- (required)
- **root_share_locking** (boolean) --
- **space** (integer) -- (required)
- **uuid** (string) -- (read only)
- **visible** (boolean) -- (required)

Status Codes:

- [201 Created](#) --

Response JSON Object:

- **admin_users** (object) --
- **advanced_settings** (object) --
- **allowed_users** (object) --
- **create_mask** (string) --
- **directory_mask** (string) --
- **force_create_mode** (string) --
- **force_directory_mode** (string) --
- **guest_ok** (boolean) --
- **hosts_allow** (object) --
- **hosts_deny** (object) --
- **hsm_support** (boolean) --
- **id** (integer) -- (read only)
- **multi_thread_reads** (boolean) --
- **multi_thread_writes** (boolean) --
- **name** (string) -- (required)
- **on_sites** (object) --
- **path** (string) --

- **read_only** (boolean) -- (required)
- **root_share_locking** (boolean) --
- **space** (integer) -- (required)
- **uuid** (string) -- (read only)
- **visible** (boolean) -- (required)

GET /shares/samba/{id}/

Parameters:

- **id** (integer) -- A unique integer value identifying this samba share.

Status Codes:

- **200 OK** --

Response JSON Object:

- **admin_users** (object) --
- **advanced_settings** (object) --
- **allowed_users** (object) --
- **create_mask** (string) --
- **directory_mask** (string) --
- **force_create_mode** (string) --
- **force_directory_mode** (string) --
- **guest_ok** (boolean) --
- **hosts_allow** (object) --
- **hosts_deny** (object) --
- **hsm_support** (boolean) --
- **id** (integer) -- (read only)
- **multi_thread_reads** (boolean) --
- **multi_thread_writes** (boolean) --
- **name** (string) -- (required)
- **on_sites** (object) --
- **path** (string) --
- **read_only** (boolean) -- (required)
- **root_share_locking** (boolean) --
- **space** (integer) -- (required)
- **uuid** (string) -- (read only)
- **visible** (boolean) -- (required)

PATCH /shares/samba/{id}/

Parameters:

- **id** (integer) -- A unique integer value identifying this samba share.

Request JSON Object:

- **admin_users** (object) --
- **advanced_settings** (object) --
- **allowed_users** (object) --
- **create_mask** (string) --
- **directory_mask** (string) --
- **force_create_mode** (string) --
- **force_directory_mode** (string) --
- **guest_ok** (boolean) --
- **hosts_allow** (object) --
- **hosts_deny** (object) --
- **hsm_support** (boolean) --

- **id** (integer) -- (read only)
- **multi_thread_reads** (boolean) --
- **multi_thread_writes** (boolean) --
- **name** (string) -- (required)
- **on_sites** (object) --
- **path** (string) --
- **read_only** (boolean) -- (required)
- **root_share_locking** (boolean) --
- **space** (integer) -- (required)
- **uuid** (string) -- (read only)
- **visible** (boolean) -- (required)

Status Codes:

- **200 OK** --

Response JSON Object:

- **admin_users** (object) --
- **advanced_settings** (object) --
- **allowed_users** (object) --
- **create_mask** (string) --
- **directory_mask** (string) --
- **force_create_mode** (string) --
- **force_directory_mode** (string) --
- **guest_ok** (boolean) --
- **hosts_allow** (object) --
- **hosts_deny** (object) --
- **hsm_support** (boolean) --
- **id** (integer) -- (read only)
- **multi_thread_reads** (boolean) --
- **multi_thread_writes** (boolean) --
- **name** (string) -- (required)
- **on_sites** (object) --
- **path** (string) --
- **read_only** (boolean) -- (required)
- **root_share_locking** (boolean) --
- **space** (integer) -- (required)
- **uuid** (string) -- (read only)
- **visible** (boolean) -- (required)

DELETE /shares/samba/{id}/

Parameters:

- **id** (integer) -- A unique integer value identifying this samba share.

Status Codes:

- **204 No Content** --

GET /sitelinks/

API endpoint for managing sitelinks.

Query Parameters:

- **page** (integer) -- A page number within the paginated result set. When not given, first page is retrieved by default.

- **page_size** (integer) -- Number of results to return per page. Page size parameter can be a number between 20 and 100. For disabling pagination and retrieving all results, 0 should be given. When page size parameter is empty or <20, 20 results are returned by default. When page size parameter >100, 100 results are returned by default.
- **site_id** (integer) -- Site ID
- **datastore_id** (integer) -- Data store ID

Status Codes:

- 200 OK --

Response JSON Object:

- **count** (integer) -- (required)
- **next** (string) --
- **previous** (string) --
- **results[].datastore** (string) -- (required)
- **results[].datastore_path** (string) -- (required)
- **results[].id** (integer) -- (read only)
- **results[].site** (string) -- (required)
- **results[].site_path** (string) -- (required)
- **results[].url** (string) -- (read only)

POST /sitelinks/

API endpoint for managing sitelinks.

Request JSON Object:

- **datastore** (string) -- (required)
- **datastore_path** (string) -- (required)
- **id** (integer) -- (read only)
- **site** (string) -- (required)
- **site_path** (string) -- (required)
- **url** (string) -- (read only)

Status Codes:

- 201 Created --

Response JSON Object:

- **datastore** (string) -- (required)
- **datastore_path** (string) -- (required)
- **id** (integer) -- (read only)
- **site** (string) -- (required)
- **site_path** (string) -- (required)
- **url** (string) -- (read only)

GET /sitelinks/{id}/

API endpoint for managing sitelinks.

Parameters:

- **id** (string) --

Status Codes:

- 200 OK --

Response JSON Object:

- **datastore** (string) -- (required)
- **datastore_path** (string) -- (required)

- **id** (integer) -- (read only)
- **site** (string) -- (required)
- **site_path** (string) -- (required)
- **url** (string) -- (read only)

PATCH /sitelinks/{id}/

API endpoint for managing sitelinks.

Parameters:

- **id** (string) --

Request JSON Object:

- **datastore** (string) -- (required)
- **datastore_path** (string) -- (required)
- **id** (integer) -- (read only)
- **site** (string) -- (required)
- **site_path** (string) -- (required)
- **url** (string) -- (read only)

Status Codes:

- **200 OK** --

Response JSON Object:

- **datastore** (string) -- (required)
- **datastore_path** (string) -- (required)
- **id** (integer) -- (read only)
- **site** (string) -- (required)
- **site_path** (string) -- (required)
- **url** (string) -- (read only)

DELETE /sitelinks/{id}/

API endpoint for managing sitelinks.

Parameters:

- **id** (string) --

Status Codes:

- **204 No Content** --

GET /sites/

API endpoint for managing sites.

Query Parameters:

- **page** (integer) -- A page number within the paginated result set. When not given, first page is retrieved by default.
- **page_size** (integer) -- Number of results to return per page. Page size parameter can be a number between 20 and 100. For disabling pagination and retrieving all results, 0 should be given. When page size parameter is empty or <20, 20 results are returned by default. When page size parameter >100, 100 results are returned by default.

Status Codes:

- **200 OK** --

Response JSON Object:

- **count** (integer) -- (required)
- **next** (string) --

- **previous** (string) --
- **results[].am_i_configured** (boolean) -- Boolean field determining if UI has been configured for the site
- **results[].bandwidth** (integer) -- speed for site (in Mb/s)
- **results[].color[]** (integer) --
- **results[].directories_count** (integer) -- How many directories this site has
- **results[].editable** (string) -- (read only)
- **results[].files_count** (integer) -- How many files this site has
- **results[].files_hydrated_count** (integer) -- How many files hydrated this site has
- **results[].files_stubbed_count** (integer) -- How many files stubbed this site has
- **results[].id** (integer) -- (read only)
- **results[].label** (string) -- Human readable name for this site
- **results[].name** (string) -- Site Name. Must match [a-zA-Z0-9-_]+ (required)
- **results[].public_url** (string) -- The base URL by which this site can be reached
- **results[].shortcode** (string) -- Shortcode
- **results[].spaces[]** (string) --
- **results[].type** (string) -- Type indicating where the site (pixstor) is deployed
- **results[].url** (string) -- (read only)

POST /sites/

API endpoint for managing sites.

Request JSON Object:

- **am_i_configured** (boolean) -- Boolean field determining if UI has been configured for the site
- **bandwidth** (integer) -- speed for site (in Mb/s)
- **color[]** (integer) --
- **directories_count** (integer) -- How many directories this site has
- **elasticsearch_url** (string) -- URL of the Elasticsearch server to use for the Analytics search backend on this site
- **enable_auto_file_batch_sizing** (boolean) -- If optimal batch size should be determined at runtime.
- **exclude** (object) -- Global workflow excludes for this site
- **file_batch_gb** (integer) -- File batch GB
- **file_batch_size** (integer) -- File batch size
- **files_count** (integer) -- How many files this site has
- **files_hydrated_count** (integer) -- How many files hydrated this site has
- **files_stubbed_count** (integer) -- How many files stubbed this site has
- **gpfs_iscan_buckets** (integer) -- Number of buckets to use for the gpfs snapdiff policy
- **gpfs_iscan_threads** (integer) -- Number of threads to use for the gpfs snapdiff policy
- **id** (integer) -- (read only)
- **include** (object) -- Global workflow includes for this site
- **label** (string) -- Human readable name for this site
- **lock_threshold** (integer) -- Threshold for soft locking snapshot rotations
- **name** (string) -- Site Name. Must match [a-zA-Z0-9-_]+ (required)

- **pixstor_search_url** (string) -- The base URL for querying the PixStor API
- **public_url** (string) -- The base URL by which this site can be reached
- **shortcode** (string) -- Shortcode (required)
- **spaces[]** (string) --
- **type** (string) -- Type indicating where the site (pixstor) is deployed

Status Codes:

- **201 Created** --

Response JSON Object:

- **am_i_configured** (boolean) -- Boolean field determining if UI has been configured for the site
- **bandwidth** (integer) -- speed for site (in Mb/s)
- **color[]** (integer) --
- **directories_count** (integer) -- How many directories this site has
- **editable** (string) -- (read only)
- **elasticsearch_url** (string) -- URL of the Elasticsearch server to use for the Analytics search backend on this site
- **enable_auto_file_batch_sizing** (boolean) -- If optimal batch size should be determined at runtime.
- **exclude** (object) -- Global workflow excludes for this site
- **file_batch_gb** (integer) -- File batch GB
- **file_batch_size** (integer) -- File batch size
- **files_count** (integer) -- How many files this site has
- **files_hydrated_count** (integer) -- How many files hydrated this site has
- **files_stubbed_count** (integer) -- How many files stubbed this site has
- **gpfs_iscan_buckets** (integer) -- Number of buckets to use for the gpfs snapdiff policy
- **gpfs_iscan_threads** (integer) -- Number of threads to use for the gpfs snapdiff policy
- **id** (integer) -- (read only)
- **include** (object) -- Global workflow includes for this site
- **label** (string) -- Human readable name for this site
- **lock_threshold** (integer) -- Threshold for soft locking snapshot rotations
- **name** (string) -- Site Name. Must match [a-zA-Z0-9-_]+ (required)
- **pixstor_search_url** (string) -- The base URL for querying the PixStor API
- **public_url** (string) -- The base URL by which this site can be reached
- **shortcode** (string) -- Shortcode
- **spaces[].id** (integer) -- (read only)
- **spaces[].name** (string) -- Space Name (required)
- **spaces[].url** (string) -- (read only)
- **type** (string) -- Type indicating where the site (pixstor) is deployed
- **url** (string) -- (read only)

GET /sites/{id}/

API endpoint for managing sites.

Parameters:

- **id** (string) --

Status Codes:

- **200 OK** --

Response JSON Object:

- **am_i_configured** (boolean) -- Boolean field determining if UI has been configured for the site
- **bandwidth** (integer) -- speed for site (in Mb/s)
- **color[]** (integer) --
- **directories_count** (integer) -- How many directories this site has
- **editable** (string) -- (read only)
- **elasticsearch_url** (string) -- URL of the Elasticsearch server to use for the Analytics search backend on this site
- **enable_auto_file_batch_sizing** (boolean) -- If optimal batch size should be determined at runtime.
- **exclude** (object) -- Global workflow excludes for this site
- **file_batch_gb** (integer) -- File batch GB
- **file_batch_size** (integer) -- File batch size
- **files_count** (integer) -- How many files this site has
- **files_hydrated_count** (integer) -- How many files hydrated this site has
- **files_stubbed_count** (integer) -- How many files stubbed this site has
- **gpfs_iscan_buckets** (integer) -- Number of buckets to use for the gpfs snapdiff policy
- **gpfs_iscan_threads** (integer) -- Number of threads to use for the gpfs snapdiff policy
- **id** (integer) -- (read only)
- **include** (object) -- Global workflow includes for this site
- **label** (string) -- Human readable name for this site
- **lock_threshold** (integer) -- Threshold for soft locking snapshot rotations
- **name** (string) -- Site Name. Must match [a-zA-Z0-9-_]+ (required)
- **pixstor_search_url** (string) -- The base URL for querying the PixStor API
- **public_url** (string) -- The base URL by which this site can be reached
- **shortcode** (string) -- Shortcode
- **spaces[].id** (integer) -- (read only)
- **spaces[].name** (string) -- Space Name (required)
- **spaces[].url** (string) -- (read only)
- **type** (string) -- Type indicating where the site (pixstor) is deployed
- **url** (string) -- (read only)

PATCH /sites/{id}/

API endpoint for managing sites.

Parameters:

- **id** (string) --

Request JSON Object:

- **am_i_configured** (boolean) -- Boolean field determining if UI has been configured for the site
- **bandwidth** (integer) -- speed for site (in Mb/s)
- **color[]** (integer) --
- **directories_count** (integer) -- How many directories this site has
- **elasticsearch_url** (string) -- URL of the Elasticsearch server to use for the Analytics search backend on this site
- **enable_auto_file_batch_sizing** (boolean) -- If optimal batch size should be determined at runtime.

- **exclude** (object) -- Global workflow excludes for this site
- **file_batch_gb** (integer) -- File batch GB
- **file_batch_size** (integer) -- File batch size
- **files_count** (integer) -- How many files this site has
- **files_hydrated_count** (integer) -- How many files hydrated this site has
- **files_stubbed_count** (integer) -- How many files stubbed this site has
- **gpfs_iscan_buckets** (integer) -- Number of buckets to use for the gpfs snapdiff policy
- **gpfs_iscan_threads** (integer) -- Number of threads to use for the gpfs snapdiff policy
- **id** (integer) -- (read only)
- **include** (object) -- Global workflow includes for this site
- **label** (string) -- Human readable name for this site
- **lock_threshold** (integer) -- Threshold for soft locking snapshot rotations
- **name** (string) -- Site Name. Must match [a-zA-Z0-9- _]+
- **pixstor_search_url** (string) -- The base URL for querying the PixStor API
- **public_url** (string) -- The base URL by which this site can be reached
- **shortcode** (string) -- Shortcode
- **spaces[]** (string) --
- **type** (string) -- Type indicating where the site (pixstor) is deployed

Status Codes:

- 200 OK --

Response JSON Object:

- **am_i_configured** (boolean) -- Boolean field determining if UI has been configured for the site
- **bandwidth** (integer) -- speed for site (in Mb/s)
- **color[]** (integer) --
- **directories_count** (integer) -- How many directories this site has
- **editable** (string) -- (read only)
- **elasticsearch_url** (string) -- URL of the Elasticsearch server to use for the Analytics search backend on this site
- **enable_auto_file_batch_sizing** (boolean) -- If optimal batch size should be determined at runtime.
- **exclude** (object) -- Global workflow excludes for this site
- **file_batch_gb** (integer) -- File batch GB
- **file_batch_size** (integer) -- File batch size
- **files_count** (integer) -- How many files this site has
- **files_hydrated_count** (integer) -- How many files hydrated this site has
- **files_stubbed_count** (integer) -- How many files stubbed this site has
- **gpfs_iscan_buckets** (integer) -- Number of buckets to use for the gpfs snapdiff policy
- **gpfs_iscan_threads** (integer) -- Number of threads to use for the gpfs snapdiff policy
- **id** (integer) -- (read only)
- **include** (object) -- Global workflow includes for this site
- **label** (string) -- Human readable name for this site
- **lock_threshold** (integer) -- Threshold for soft locking snapshot rotations
- **name** (string) -- Site Name. Must match [a-zA-Z0-9- _]+ (required)
- **pixstor_search_url** (string) -- The base URL for querying the PixStor API
- **public_url** (string) -- The base URL by which this site can be reached

- **shortcode** (string) -- Shortcode
- **spaces[].id** (integer) -- (read only)
- **spaces[].name** (string) -- Space Name (required)
- **spaces[].url** (string) -- (read only)
- **type** (string) -- Type indicating where the site (pixstor) is deployed
- **url** (string) -- (read only)

DELETE /sites/{id}/

API endpoint for managing sites.

Parameters:

- **id** (string) --

Status Codes:

- **204 No Content** --

GET /sites/{id}/health/

API endpoint for managing sites.

Parameters:

- **id** (string) --

Status Codes:

- **200 OK** --

Response JSON Object:

- **am_i_configured** (boolean) -- Boolean field determining if UI has been configured for the site
- **bandwidth** (integer) -- speed for site (in Mb/s)
- **color[]** (integer) --
- **directories_count** (integer) -- How many directories this site has
- **editable** (string) -- (read only)
- **files_count** (integer) -- How many files this site has
- **files_hydrated_count** (integer) -- How many files hydrated this site has
- **files_stubbed_count** (integer) -- How many files stubbed this site has
- **id** (integer) -- (read only)
- **label** (string) -- Human readable name for this site
- **name** (string) -- Site Name. Must match [a-zA-Z0-9-_]+ (required)
- **public_url** (string) -- The base URL by which this site can be reached
- **shortcode** (string) -- Shortcode
- **spaces[]** (string) --
- **type** (string) -- Type indicating where the site (pixstor) is deployed
- **url** (string) -- (read only)

POST /sites/{id}/refresh/

API endpoint for managing sites.

Parameters:

- **id** (string) --

Status Codes:

- **201 Created** --

Response JSON Object:

- **am_i_configured** (boolean) -- Boolean field determining if UI has been configured for the site

- **bandwidth** (integer) -- speed for site (in Mb/s)
- **color[]** (integer) --
- **directories_count** (integer) -- How many directories this site has
- **editable** (string) -- (read only)
- **files_count** (integer) -- How many files this site has
- **files_hydrated_count** (integer) -- How many files hydrated this site has
- **files_stubbed_count** (integer) -- How many files stubbed this site has
- **id** (integer) -- (read only)
- **label** (string) -- Human readable name for this site
- **name** (string) -- Site Name. Must match [a-zA-Z0-9-_]+ (required)
- **public_url** (string) -- The base URL by which this site can be reached
- **shortcode** (string) -- Shortcode
- **spaces[]** (string) --
- **type** (string) -- Type indicating where the site (pixstor) is deployed
- **url** (string) -- (read only)

GET /sites/{id}/settings/

API endpoint for managing sites.

Parameters:

- **id** (string) --

Query Parameters:

- **flat** (boolean) -- Return flat dict

Status Codes:

- **200 OK** --

Response JSON Object:

- **[].desc** (string) -- (read only)
- **[].id** (integer) -- (read only)
- **[].key** (string) -- (required)
- **[].value** (object) --

PATCH /sites/{id}/settings/

API to set a setting or a group of settings associated with a site. A group of settings should be sent in key-value schema. Terminology Setting - An individual setting e.g. sambda:realm

Parameters:

- **id** (string) --

Request JSON Object:

- **values[].desc** (string) -- (read only)
- **values[].id** (integer) -- (read only)
- **values[].key** (string) -- (required)
- **values[].value** (object) --

Status Codes:

- **202 Accepted** -- Task id of the task fired for the associated request.

POST /sites/{id}/settings/refresh/

Refresh settings for a given site under management. This API will also create settings in the hub for this site if they don't already exist.

Parameters:

- **id** (string) --

Status Codes:

- [202 Accepted](#) -- Task id of the task fired for the associated request.

GET /spaces/

Model View Set for filebrowser's Space model.

Query Parameters:

- **page** (integer) -- A page number within the paginated result set. When not given, first page is retrieved by default.
- **page_size** (integer) -- Number of results to return per page. Page size parameter can be a number between [20](#) and [100](#). For disabling pagination and retrieving all results, [0](#) should be given. When page size parameter is empty or [<20](#), [20](#) results are returned by default. When page size parameter [>100](#), [100](#) results are returned by default.

Status Codes:

- [200 OK](#) --

Response JSON Object:

- **count** (integer) -- (required)
- **next** (string) --
- **previous** (string) --
- **results[].color[]** (integer) --
- **results[].created** (string) -- Time of the job creation (read only)
- **results[].editable** (string) -- (read only)
- **results[].id** (integer) -- (read only)
- **results[].immutable** (boolean) -- Space that once created cannot be deleted or updated. Usually the default space.
- **results[].mountpoint** (string) -- Mount point (required)
- **results[].name** (string) -- Space Name (required)
- **results[].on_sites** (object) --
- **results[].permission_mode** (string) -- Ability to set the permission mode on a space
- **results[].relationships** (string) -- (read only)
- **results[].sites[].id** (string) -- (read only)
- **results[].sites[].label** (string) -- (required)
- **results[].sites[].name** (string) -- (required)
- **results[].sites[].pool** (string) -- (read only)
- **results[].sites[].usage[]** (integer) --
- **results[].size** (integer) -- Space size quota in bytes. No value specified, means as large as your system allows.
- **results[].snapshot_schedule.duration** (string) -- (required)
- **results[].snapshot_schedule.frequency** (string) -- (required)
- **results[].snapshot_schedule.id** (integer) -- (read only)
- **results[].snapshot_schedule.space** (integer) --
- **results[].snapshot_schedule.time** (string) -- (required)
- **results[].url** (string) -- (read only)
- **results[].uuid** (string) -- (read only)

POST /spaces/

Request JSON Object:

- **color[]** (integer) --
- **created** (string) -- Time of the job creation (read only)
- **mountpoint** (string) -- (required)
- **name** (string) -- (required)
- **permission_mode** (string) -- Ability to set the permission mode on a space
- **sites[].id** (string) -- (read only)
- **sites[].label** (string) -- (required)
- **sites[].name** (string) -- (required)
- **sites[].pool** (string) -- (read only)
- **sites[].usage[]** (integer) --
- **size** (integer) -- Space size quota in bytes. No value specified, means as large as your system allows.
- **snapshot_schedule.duration** (string) -- (required)
- **snapshot_schedule.frequency** (string) -- (required)
- **snapshot_schedule.id** (integer) -- (read only)
- **snapshot_schedule.space** (integer) --
- **snapshot_schedule.time** (string) -- (required)

Status Codes:

- [201 Created](#) --

Response JSON Object:

- **color[]** (integer) --
- **created** (string) -- Time of the job creation (read only)
- **editable** (string) -- (read only)
- **id** (integer) -- (read only)
- **immutable** (boolean) -- Space that once created cannot be deleted or updated. Usually the default space.
- **mountpoint** (string) -- Mount point (required)
- **name** (string) -- Space Name (required)
- **on_sites** (object) --
- **permission_mode** (string) -- Ability to set the permission mode on a space
- **relationships** (string) -- (read only)
- **sites[].id** (string) -- (read only)
- **sites[].label** (string) -- (required)
- **sites[].name** (string) -- (required)
- **sites[].pool** (string) -- (read only)
- **sites[].usage[]** (integer) --
- **size** (integer) -- Space size quota in bytes. No value specified, means as large as your system allows.
- **snapshot_schedule.duration** (string) -- (required)
- **snapshot_schedule.frequency** (string) -- (required)
- **snapshot_schedule.id** (integer) -- (read only)
- **snapshot_schedule.space** (integer) --
- **snapshot_schedule.time** (string) -- (required)
- **url** (string) -- (read only)
- **uuid** (string) -- (read only)

GET /spaces/{id}/

Model View Set for filebrowser's Space model.

Parameters:

- **id** (string) --

Status Codes:

- **200 OK** --

Response JSON Object:

- **color[]** (integer) --
- **created** (string) -- Time of the job creation (read only)
- **editable** (string) -- (read only)
- **id** (integer) -- (read only)
- **immutable** (boolean) -- Space that once created cannot be deleted or updated. Usually the default space.
- **mountpoint** (string) -- Mount point (required)
- **name** (string) -- Space Name (required)
- **on_sites** (object) --
- **permission_mode** (string) -- Ability to set the permission mode on a space
- **relationships** (string) -- (read only)
- **sites[].id** (string) -- (read only)
- **sites[].label** (string) -- (required)
- **sites[].name** (string) -- (required)
- **sites[].pool** (string) -- (read only)
- **sites[].usage[]** (integer) --
- **size** (integer) -- Space size quota in bytes. No value specified, means as large as your system allows.
- **snapshot_schedule.duration** (string) -- (required)
- **snapshot_schedule.frequency** (string) -- (required)
- **snapshot_schedule.id** (integer) -- (read only)
- **snapshot_schedule.space** (integer) --
- **snapshot_schedule.time** (string) -- (required)
- **url** (string) -- (read only)
- **uuid** (string) -- (read only)

PATCH /spaces/{id}/

Model View Set for filebrowser's Space model.

Parameters:

- **id** (string) --

Request JSON Object:

- **color[]** (integer) --
- **created** (string) -- Time of the job creation (read only)
- **mountpoint** (string) -- Mount point (read only)
- **name** (string) --
- **permission_mode** (string) -- Ability to set the permission mode on a space
- **sites[].id** (string) -- (read only)
- **sites[].label** (string) -- (required)
- **sites[].name** (string) -- (required)
- **sites[].pool** (string) -- (read only)
- **sites[].usage[]** (integer) --

- **size** (integer) -- Space size quota in bytes. No value specified, means as large as your system allows.
- **snapshot_schedule.duration** (string) -- (required)
- **snapshot_schedule.frequency** (string) -- (required)
- **snapshot_schedule.id** (integer) -- (read only)
- **snapshot_schedule.space** (integer) --
- **snapshot_schedule.time** (string) -- (required)

Status Codes:

- 200 OK --

Response JSON Object:

- **color[]** (integer) --
- **created** (string) -- Time of the job creation (read only)
- **editable** (string) -- (read only)
- **id** (integer) -- (read only)
- **immutable** (boolean) -- Space that once created cannot be deleted or updated. Usually the default space.
- **mountpoint** (string) -- Mount point (required)
- **name** (string) -- Space Name (required)
- **on_sites** (object) --
- **permission_mode** (string) -- Ability to set the permission mode on a space
- **relationships** (string) -- (read only)
- **sites[].id** (string) -- (read only)
- **sites[].label** (string) -- (required)
- **sites[].name** (string) -- (required)
- **sites[].pool** (string) -- (read only)
- **sites[].usage[]** (integer) --
- **size** (integer) -- Space size quota in bytes. No value specified, means as large as your system allows.
- **snapshot_schedule.duration** (string) -- (required)
- **snapshot_schedule.frequency** (string) -- (required)
- **snapshot_schedule.id** (integer) -- (read only)
- **snapshot_schedule.space** (integer) --
- **snapshot_schedule.time** (string) -- (required)
- **url** (string) -- (read only)
- **uuid** (string) -- (read only)

DELETE /spaces/{id}/

Model View Set for filebrowser's Space model.

Parameters:

- **id** (string) --

Status Codes:

- 204 No Content --

GET /spaces/{id}/files/

Model View Set for filebrowser's Space model.

Parameters:

- **id** (string) --

Query Parameters:

- **page** (integer) -- A page number within the paginated result set. When not given, first page is retrieved by default.
- **page_size** (integer) -- Number of results to return per page. Page size parameter can be a number between 20 and 100. For disabling pagination and retrieving all results, 0 should be given. When page size parameter is empty or <20, 20 results are returned by default. When page size parameter >100, 100 results are returned by default.
- **site** (string) -- Site to run the stat operation on if provided, otherwise all space's sites
- **path** (string) -- Path to retrieve stat
- **details** (boolean) -- Retrieve all the details of the children files
- **restricted** (boolean) -- Include restricted objects
- **check_available_sites** (boolean) -- Check only available sites
- **custom_refresh_period** (integer) -- Custom time in seconds to invalidate the DB cache
- **custom_timeout** (integer) -- Custom time in seconds to invalidate the task
- **raise_exception** (boolean) -- Raise exception if timeout
- **type** (string) -- item type

Status Codes:

- 200 OK --

Response JSON Object:

- **atime** (string) --
- **children** (string) -- (read only)
- **ctime** (string) --
- **data_size** (integer) --
- **deleted** (boolean) --
- **disk_usage** (integer) --
- **image_preview** (string) -- Relative URL of an image preview
- **metadata** (object) --
- **mtime** (string) --
- **name** (string) -- Name of of the fs object
- **path** (string) -- Path of the object (required)
- **path_depth** (integer) --
- **remote_locations** (object) --
- **restricted** (boolean) --
- **site** (string) -- (read only)
- **size** (integer) --
- **status** (string) -- Migration status
- **thumbnail** (string) -- Relative URL of an object thumbnail
- **time_refreshed** (string) --
- **total_files** (integer) -- Only applies to directories
- **type** (string) --
- **url** (string) -- (read only)
- **video_preview** (string) -- Relative URL of a video preview

GET /storagepools/

Query Parameters:

- **page** (integer) -- A page number within the paginated result set. When not given, first page is retrieved by default.

- **page_size** (integer) -- Number of results to return per page. Page size parameter can be a number between 20 and 100. For disabling pagination and retrieving all results, 0 should be given. When page size parameter is empty or <20, 20 results are returned by default. When page size parameter >100, 100 results are returned by default.

Status Codes:

- 200 OK --

Response JSON Object:

- **count** (integer) -- (required)
- **next** (string) --
- **previous** (string) --
- **results[].capacity** (integer) -- Capacity of storage pool (required)
- **results[].filesystem** (string) -- Storage pool filesystem (required)
- **results[].free_space** (integer) -- Free space of storage pool (required)
- **results[].is_default** (boolean) -- Check if default pool
- **results[].name** (string) -- Name storage pool (required)
- **results[].site** (integer) -- Site correlated with storage object (required)
- **results[].used_space** (integer) -- Used space of storage pool (required)

GET /storagepools/{id}/

Parameters:

- **id** (string) --

Status Codes:

- 200 OK --

Response JSON Object:

- **capacity** (integer) -- Capacity of storage pool (required)
- **filesystem** (string) -- Storage pool filesystem (required)
- **free_space** (integer) -- Free space of storage pool (required)
- **is_default** (boolean) -- Check if default pool
- **name** (string) -- Name storage pool (required)
- **site** (integer) -- Site correlated with storage object (required)
- **used_space** (integer) -- Used space of storage pool (required)

GET /tags/

API endpoint for search tags

Query Parameters:

- **filter** (string) -- substring that returned tags must contain

Status Codes:

- 200 OK --

Response JSON Object:

- **[].name** (string) -- (required)

GET /tasks/

API endpoint for viewing tasks.

Query Parameters:

- **page** (integer) -- A page number within the paginated result set. When not given, first page is retrieved by default.
- **page_size** (integer) -- Number of results to return per page. Page size parameter can be a number between 20 and 100. For disabling pagination

and retrieving all results, 0 should be given. When page size parameter is empty or <20, 20 results are returned by default. When page size parameter >100, 100 results are returned by default.

- **tasktype** (string) -- Task type
- **task_id** (string) -- Task Id
- **state** (array) -- Task states
- **job_id** (integer) -- Job ID
- **internal** (string) --

Status Codes:

- 200 OK --

Response JSON Object:

- **count** (integer) -- (required)
- **next** (string) --
- **previous** (string) --
- **results[].id** (integer) -- (read only)
- **results[].job** (integer) -- (required)
- **results[].site** (string) -- (required)
- **results[].started** (string) -- Time that the task started running
- **results[].state** (string) --
- **results[].task_id** (string) -- Celery task ID
- **results[].tasktype** (string) -- (required)
- **results[].url** (string) -- (read only)

GET /tasks/{id}/

API endpoint for viewing tasks.

Parameters:

- **id** (string) --

Status Codes:

- 200 OK --

Response JSON Object:

- **completed** (string) -- Time of the task completion
- **friendly_name** (string) -- Text to be used to describe what this task is doing.
- **id** (integer) -- (read only)
- **job** (integer) -- (required)
- **paths** (string) -- (read only)
- **result** (object) --
- **runtime** (string) -- (read only)
- **site** (string) -- (required)
- **started** (string) -- Time that the task started running
- **state** (string) --
- **task_id** (string) -- Celery task ID
- **tasktype** (string) -- (required)
- **url** (string) -- (read only)

GET /tasks/{id}/files/

API endpoint for viewing tasks.

Parameters:

- **id** (string) --

Query Parameters:

- **page** (integer) -- A page number within the paginated result set. When not given, first page is retrieved by default.
- **page_size** (integer) -- Number of results to return per page. Page size parameter can be a number between 20 and 100. For disabling pagination and retrieving all results, 0 should be given. When page size parameter is empty or <20, 20 results are returned by default. When page size parameter >100, 100 results are returned by default.
- **state** (string) --
- **type** (string) --
- **site** (string) --

Status Codes:

- 200 OK --

Response JSON Object:

- **completed** (string) -- Time of the task completion
- **friendly_name** (string) -- Text to be used to describe what this task is doing.
- **id** (integer) -- (read only)
- **job** (integer) -- (required)
- **paths** (string) -- (read only)
- **result** (object) --
- **runtime** (string) -- (read only)
- **site** (string) -- (required)
- **started** (string) -- Time that the task started running
- **state** (string) --
- **task_id** (string) -- Celery task ID
- **tasktype** (string) -- (required)
- **url** (string) -- (read only)

GET /users/

API endpoint for managing users.

Query Parameters:

- **page** (integer) -- A page number within the paginated result set. When not given, first page is retrieved by default.
- **page_size** (integer) -- Number of results to return per page. Page size parameter can be a number between 20 and 100. For disabling pagination and retrieving all results, 0 should be given. When page size parameter is empty or <20, 20 results are returned by default. When page size parameter >100, 100 results are returned by default.

Status Codes:

- 200 OK --

Response JSON Object:

- **count** (integer) -- (required)
- **next** (string) --
- **previous** (string) --
- **results[].date_joined** (string) --
- **results[].email** (string) --
- **results[].first_name** (string) --
- **results[].groups[].id** (integer) -- (read only)
- **results[].groups[].name** (string) -- (required)

- **results[].groups[].url** (string) -- (read only)
- **results[].id** (integer) -- (read only)
- **results[].is_active** (boolean) -- Designates whether this user should be treated as active. Unselect this instead of deleting accounts.
- **results[].last_login** (string) --
- **results[].last_name** (string) --
- **results[].nas_user.allow_ssh** (boolean) --
- **results[].nas_user.primary_group** (string) -- (required)
- **results[].nas_user.uid** (integer) -- UID number for the user. This is automatically generated. And cannot be changed (read only)
- **results[].profile.country** (string) -- User's country
- **results[].profile.default_site** (integer) --
- **results[].profile.department** (string) -- Name of the department user is working in
- **results[].profile.home_site** (integer) --
- **results[].profile.line_manager** (string) -- Name of the line manager of the user
- **results[].profile.phone** (string) -- User's phone number
- **results[].profile.timezone** (string) -- Timezone in a string format e.g. 'Europe/London'
- **results[].url** (string) -- (read only)
- **results[].username** (string) -- Required. 150 characters or fewer. Letters, digits and @/./+/-/_ only. (required)

POST /users/

API endpoint for managing users.

Request JSON Object:

- **email** (string) --
- **first_name** (string) --
- **groups[]** (string) --
- **last_name** (string) --
- **nas_user.allow_ssh** (boolean) --
- **nas_user.primary_group** (string) -- (required)
- **password** (string) -- (required)
- **profile.country** (string) -- User's country
- **profile.default_site** (integer) --
- **profile.department** (string) -- Name of the department user is working in
- **profile.home_site** (integer) --
- **profile.line_manager** (string) -- Name of the line manager of the user
- **profile.phone** (string) -- User's phone number
- **profile.timezone** (string) -- Timezone in a string format e.g. 'Europe/London'
- **username** (string) -- Required. 150 characters or fewer. Letters, digits and @/./+/-/_ only. (required)

Status Codes:

- [201 Created](#) --

Response JSON Object:

- **email** (string) --
- **first_name** (string) --
- **groups[]** (string) --

- **last_name** (string) --
- **nas_user.allow_ssh** (boolean) --
- **nas_user.primary_group** (string) -- (required)
- **password** (string) -- (required)
- **profile.country** (string) -- User's country
- **profile.default_site** (integer) --
- **profile.department** (string) -- Name of the department user is working in
- **profile.home_site** (integer) --
- **profile.line_manager** (string) -- Name of the line manager of the user
- **profile.phone** (string) -- User's phone number
- **profile.timezone** (string) -- Timezone in a string format e.g. 'Europe/London'
- **username** (string) -- Required. 150 characters or fewer. Letters, digits and @/./+/-/_ only. (required)

POST /users/change_password/

API endpoint for managing users.

Request JSON Object:

- **new_password** (string) -- (required)
- **old_password** (string) -- (required)

Status Codes:

- [201 Created](#) --

Response JSON Object:

- **new_password** (string) -- (required)
- **old_password** (string) -- (required)

GET /users/{username}/

API endpoint for managing users.

Parameters:

- **username** (string) --

Status Codes:

- [200 OK](#) --

Response JSON Object:

- **date_joined** (string) --
- **email** (string) --
- **first_name** (string) --
- **groups[].id** (integer) -- (read only)
- **groups[].name** (string) -- (required)
- **groups[].url** (string) -- (read only)
- **id** (integer) -- (read only)
- **is_active** (boolean) -- Designates whether this user should be treated as active. Unselect this instead of deleting accounts.
- **last_login** (string) --
- **last_name** (string) --
- **nas_user.allow_ssh** (boolean) --
- **nas_user.primary_group** (string) -- (required)
- **nas_user.uid** (integer) -- UID number for the user. This is automatically generated. And cannot be changed (read only)
- **profile.country** (string) -- User's country

- **profile.default_site** (integer) --
- **profile.department** (string) -- Name of the department user is working in
- **profile.home_site** (integer) --
- **profile.line_manager** (string) -- Name of the line manager of the user
- **profile.phone** (string) -- User's phone number
- **profile.timezone** (string) -- Timezone in a string format e.g. 'Europe/London'
- **url** (string) -- (read only)
- **username** (string) -- Required. 150 characters or fewer. Letters, digits and @/./+/-/_ only. (required)

PATCH /users/{username}/

API endpoint for managing users.

Parameters:

- **username** (string) --

Request JSON Object:

- **email** (string) --
- **first_name** (string) --
- **groups[]** (string) --
- **last_name** (string) --
- **nas_user.allow_ssh** (boolean) --
- **nas_user.primary_group** (string) -- (required)
- **password** (string) --
- **profile.country** (string) -- User's country
- **profile.default_site** (integer) --
- **profile.department** (string) -- Name of the department user is working in
- **profile.home_site** (integer) --
- **profile.line_manager** (string) -- Name of the line manager of the user
- **profile.phone** (string) -- User's phone number
- **profile.timezone** (string) -- Timezone in a string format e.g. 'Europe/London'

Status Codes:

- **200 OK** --

Response JSON Object:

- **email** (string) --
- **first_name** (string) --
- **groups[]** (string) --
- **last_name** (string) --
- **nas_user.allow_ssh** (boolean) --
- **nas_user.primary_group** (string) -- (required)
- **password** (string) --
- **profile.country** (string) -- User's country
- **profile.default_site** (integer) --
- **profile.department** (string) -- Name of the department user is working in
- **profile.home_site** (integer) --
- **profile.line_manager** (string) -- Name of the line manager of the user
- **profile.phone** (string) -- User's phone number
- **profile.timezone** (string) -- Timezone in a string format e.g. 'Europe/London'

DELETE /users/{username}/

API endpoint for managing users.

Parameters:

- **username** (string) --

Status Codes:

- 204 No Content --

POST /users/{username}/activate/

Activates user account with given username.

Parameters:

- **username** (string) --

Status Codes:

- 201 Created --

POST /users/{username}/deactivate/

Deactivates user account with given username.

Parameters:

- **username** (string) --

Status Codes:

- 201 Created --

GET /workflows/

API endpoint for viewing workflows.

Query Parameters:

- **page** (integer) -- A page number within the paginated result set. When not given, first page is retrieved by default.
- **page_size** (integer) -- Number of results to return per page. Page size parameter can be a number between 20 and 100. For disabling pagination and retrieving all results, 0 should be given. When page size parameter is empty or <20, 20 results are returned by default. When page size parameter >100, 100 results are returned by default.

Status Codes:

- 200 OK --

Response JSON Object:

- **count** (integer) -- (required)
- **next** (string) --
- **previous** (string) --
- **results[].discovery** (string) --
- **results[].discovery_options** (object) --
- **results[].enabled** (boolean) -- Is the workflow available for use?
- **results[].external_only** (boolean) -- Signifies that the workflow is only allowed to operate on external targets
- **results[].fields** (object) --
- **results[].filter_rules** (object) --
- **results[].icon_classes** (object) --
- **results[].id** (integer) -- (read only)
- **results[].label** (string) -- Friendly name of the workflow (required)
- **results[].name** (string) -- Name of Workflow (required)
- **results[].preferred_queue** (string) -- Preferred queue for this workflow

- **results[].schedule_only** (boolean) -- Workflow only callable inside of a schedule
- **results[].visible** (boolean) -- Is the workflow visible on the UI?

POST /workflows/

API endpoint for viewing workflows.

Request JSON Object:

- **discovery** (string) --
- **discovery_options** (object) --
- **enabled** (boolean) -- Is the workflow available for use?
- **external_only** (boolean) -- Signifies that the workflow is only allowed to operate on external targets
- **fields** (object) --
- **filter_rules** (object) --
- **icon_classes** (object) --
- **id** (integer) -- (read only)
- **label** (string) -- Friendly name of the workflow (required)
- **name** (string) -- Name of Workflow (required)
- **preferred_queue** (string) -- Preferred queue for this workflow
- **schedule_only** (boolean) -- Workflow only callable inside of a schedule
- **visible** (boolean) -- Is the workflow visible on the UI?

Status Codes:

- [201 Created](#) --

Response JSON Object:

- **discovery** (string) --
- **discovery_options** (object) --
- **enabled** (boolean) -- Is the workflow available for use?
- **external_only** (boolean) -- Signifies that the workflow is only allowed to operate on external targets
- **fields** (object) --
- **filter_rules** (object) --
- **icon_classes** (object) --
- **id** (integer) -- (read only)
- **label** (string) -- Friendly name of the workflow (required)
- **name** (string) -- Name of Workflow (required)
- **preferred_queue** (string) -- Preferred queue for this workflow
- **schedule_only** (boolean) -- Workflow only callable inside of a schedule
- **visible** (boolean) -- Is the workflow visible on the UI?

GET /workflows/{id}/

API endpoint for viewing workflows.

Parameters:

- **id** (string) --

Status Codes:

- [200 OK](#) --

Response JSON Object:

- **discovery** (string) --
- **discovery_options** (object) --

- **enabled** (boolean) -- Is the workflow available for use?
- **external_only** (boolean) -- Signifies that the workflow is only allowed to operate on external targets
- **fields** (object) --
- **filter_rules** (object) --
- **icon_classes** (object) --
- **id** (integer) -- (read only)
- **label** (string) -- Friendly name of the workflow (required)
- **name** (string) -- Name of Workflow (required)
- **preferred_queue** (string) -- Preferred queue for this workflow
- **schedule_only** (boolean) -- Workflow only callable inside of a schedule
- **visible** (boolean) -- Is the workflow visible on the UI?

PATCH /workflows/{id}/

API endpoint for viewing workflows.

Parameters:

- **id** (string) --

Request JSON Object:

- **discovery** (string) --
- **discovery_options** (object) --
- **enabled** (boolean) -- Is the workflow available for use?
- **fields** (object) --
- **filter_rules** (object) --
- **icon_classes** (object) --
- **id** (integer) -- (read only)
- **label** (string) -- Friendly name of the workflow (required)
- **name** (string) -- Name of Workflow (required)
- **preferred_queue** (string) -- Preferred queue for this workflow
- **schedule_only** (boolean) -- Workflow only callable inside of a schedule
- **visible** (boolean) -- Is the workflow visible on the UI?

Status Codes:

- **200 OK** --

Response JSON Object:

- **discovery** (string) --
- **discovery_options** (object) --
- **enabled** (boolean) -- Is the workflow available for use?
- **fields** (object) --
- **filter_rules** (object) --
- **icon_classes** (object) --
- **id** (integer) -- (read only)
- **label** (string) -- Friendly name of the workflow (required)
- **name** (string) -- Name of Workflow (required)
- **preferred_queue** (string) -- Preferred queue for this workflow
- **schedule_only** (boolean) -- Workflow only callable inside of a schedule
- **visible** (boolean) -- Is the workflow visible on the UI?

DELETE /workflows/{id}/

API endpoint for viewing workflows.

Parameters:

- **id** (string) --

Status Codes:

- 204 No Content --

Tools

The following sections document add-on tools for Ngenea Hub

ngclient

NGCLIENT

SYNOPSIS

ngclient authenticate (-u USERNAME [-p PASSWORD] | -T TOKEN) [-k NAME]

ngclient migrate path... [-s site] [-r] [-p] [options...]

ngclient recall path... [-s site] [-r] [options...]

ngclient send path... [-s source] -t target [options...]

ngclient workflows COMMAND

ngclient features COMMAND

DESCRIPTION

ngclient is a CLI wrapper for the default Ngenea Hub workflows - migrate, recall, and send. It also provides a mechanism for generating authentication tokens.

ngclient settings can be read from a config file, rather than being passed on the command line. See **ngenea-client.conf(5)** for more information on the configuration format. CLI flags take precedence over config file settings.

The authenticate command can be used to generate a client key from a username or access token. The generated client key will be printed to stdout. That client key can then be used with the workflow sub-commands, either via the --client-key flag, or by saving it to the **ngenea-client.conf(5)** configuration file.

The workflows command group contains sub-commands for interacting with workflows, such as listing workflows and importing new one. See **ngclient-workflows(1)** for more details.

The features command group contains sub-commands for listing, enabling, or disabling feature flags. See **ngclient-features(1)** for more details.

OPTION SUMMARY

path...	One paths to call the workflow on
-T, --access-token TOKEN	Access token to authenticate with
-u, --username USERNAME	Username to authenticate with
-p, --password PASSWORD	Password for the authentication
username	
-k, --key-name NAME	Unique name for the client key
--base-url	Base URL of the {{ brand_name }}
API	
-c, --config CONFIG	Alternative configuration file
path	
--client-key KEY	Client API key to authenticate
with	
-D, --no-verify	Disable TLS verification
-s, --site SITE	Site to perform the workflow on
-t, --target TARGET	Site to send files to
-q, --queue QUEUE	Queue to perform the workflow on
-Q, --target-queue TARGETQUEUE	Queue to send files to
-d, --no-wait	Exit after job is submitted, don't
wait for it to complete	
--timeout SECONDS	wait for completion timeout. If
not set, wait indefinitely	
-r, --recursive	Perform task recursively.
-p, --premigrate	Premigrate files from site
-H, --hydrate	Hydrate files on the send target
site	
-h, --help	Print help message and exit

OPTIONS

- **-T, --access-token**

An access token to generate a client key with.

- **-u, --username**

Username to generate a client key with.

- **-p, --password**

Password to use in combination with --username

If --username is specified and --password isn't, you will be prompted to enter a password interactively. This may be preferable so that the password doesn't appear in shell history.

- **-k, --key-name**

A unique name to assign when generating a client key.

This will be displayed in the Ngenea Hub UI

If not specified, a random uuid will be generated for the key name.

- **--base-url**

Base URL of the Ngenea Hub API, which operations will be performed against.

This can be used to perform Ngenea Hub operations on a remote server.

If not specified, the default is `http://localhost:8000/api`

- **-c, --config**

The path to an alternative configuration file.

If not specified, the default configuration paths will be used. The default paths are in the user's HOME directory `$HOME/.config/ngenea/ngenea-client.conf`, and the global configuration at `/etc/ngenea/ngenea-client.conf`

See **ngenea-client.conf(5)** for more information on the configuration format.

Command line options take precedence over any corresponding config file settings.

- **-D, --no-verify**

Disables TLS Verification

By default, **ngclient** will verify TLS certificate at the remote end.

With this flag, requests made by **ngclient** will accept any TLS certificate presented by the server, and will ignore hostname mismatches and/or expired certificates.

- **--client-key**

Ngenea Hub authentication client key.

This can be generated via the Ngenea Hub REST API, or using **ngclient(1)** `authenticate`

- **-s, --site**

Site to use for workflows.

For migrate and recall, this is the site where the workflows execute. For send, this is the source site, from which files are sent.

Site does not have to match the node where **ngclient** is being called. This can be used to migrate/recall/send files from a remote site.

Note - shell-globbing will be evaluated on the local node. For a remote site, the files that the glob would match may differ.

- **-t, --target**

Target site for the send workflow

- **-q, --queue**

Queue to use for workflows. By default, source queue is set as default

For migrate and recall, this is the queue where the workflows execute. For send, this is the source queue, from which files are sent.

- **-Q, --target-queue**

Target queue for the send workflow

If the target queue is not provided, it will use the source queue. If the source queue is also not provided, the queue will be set to default.

- **-d, --no-wait**

Don't wait for workflows to complete.

By default, **ngclient** will wait for the workflow to complete, subject to --timeout.

With this flag, **ngclient** will exit immediately. The workflow will continue to execute independently. In that case, the workflow can be monitored in the Ngenea Hub UI

- **--timeout**

How long to wait for workflows to complete, in seconds.

If not specified, **ngclient** will wait indefinitely.

If the workflow doesn't complete within the timeout, the client will exit with an error. The workflow itself may continue to execute.

- **-r, --recursive**

Migrate or recall files and directories recursively.

- **-p, --premigrate**

Premigrate files

Premigrated files are migrated, but the data is kept resident.

- **-H, --hydrate**

Controls whether the send workflow 'hydrates' files on the target.

If false, files are only reverse stubbed on the target.

- **-h, --help**

Prints the help message.

EXAMPLES

GENERATE A CLIENT KEY

```
$ ngclient authenticate --username pixadmin -k ngclient-pixadmin
pixadmin's password:
jDBh2cRk6.LswQfylT2BtGiqTYUWhMBliipJmQNgr
```

NOTE: The `authenticate` command doesn't read config values from the config file, so the `--no-verify` and `--base-url` arguments should be passed as cmd arguments if default values aren't required

RECALL A FILE

```
ngclient recall /mmfs1/data/hello.txt -s site1 --client-key
jDBh2cRk6.LswQfylT2BtGiqTYUWhMBliipJmQNgr
```

For brevity, the site and client-key can be saved to the config file

PREMIGRATE A DIRECTORY RECURSIVELY

Assuming the site and client key has been saved to the config file

```
ngclient migrate /mmfs1/data/sample_data/cats -p -r
```

SEND A FILE TO A REMOTE SITE

```
ngclient send /mmfs1/data/hello.txt -s site1 -t site2 --hydrate
```

AVAILABILITY

Distributed as part of the `ngenea-hub-client` rpm, or the `ngclient` wheel (Python) for non-Red Hat based systems.

The `ngclient` wheel can be installed and run on any operating system.

Note - `transparent_recall(1)` is packaged along with `ngclient`, but `transparent_recall` will only work on Unix-based operating systems.

SEE ALSO

`ngenea-client.conf(5)`, `ngclient-workflows(1)`, `ngclient-features(1)`,
`transparent_recall(1)`, `ngmigrate(1)`, `ngrecall(1)`

LICENSE

2021 ArcaPix Limited

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

ngclient-workflows

NGCLIENT-WORKFLOWS

SYNOPSIS

- ngclient workflows** list [workflow-id] [options...]
- ngclient workflows** import workflow-file [options...]
- ngclient workflows** update workflow-id workflow-file [options...]
- ngclient workflows** delete workflow-id [options...]

DESCRIPTION

The list command is used to list one or more existing workflows from Ngenea Hub. By default, workflows are output in json format, one per line. The --yaml flag can be used to output as yaml.

The import command can be used to import a new, custom workflow from a json or yaml formatted file.

Currently it is not possible to invoke these custom workflows from **ngclient**, once created. They can be invoked from the Ngenea Hub UI or via the REST API.

The update command can be used to update an existing workflow from a json or yaml formatted file. The file can contain only the fields you want to change to perform a partial update, or a whole workflow definition for a full replacement.

NOTE - it's not possible to make partial changes to the fields or filter_rules blocks. They can only be replaced as a whole.

The delete command can be used to delete an existing workflow, by id.

Base URL and API key settings can be read from a config file, rather than being passed on the command line. See **ngenea-client.conf(5)** for more information on the configuration format. CLI flags take precedence over config file settings.

Interacting with workflows requires Ngenea Hub authentication. The **ngclient(1)** authenticate command can be used to generate a client key from a username or access token.

OPTION SUMMARY

workflow-id	Unique workflow identifier
workflow-file	File containing a custom workflow

definition

<code>--yaml</code>	List workflows in yaml format
<code>--base-url</code>	Base URL of the {{ brand_name }} API
<code>-c, --config CONFIG</code> <code>--client-key KEY</code>	Alternative configuration file path Client API key to authenticate with
<code>-h, --help</code>	Print help message and exit

OPTIONS

- **workflow-file**

Path to a json or yaml formatted file, containing a workflow definition.

If '-' is used, the workflow definition will be read from stdin.

The workflow format is described in the main documentation, section '4.4. Custom Workflows'

- **--yaml**

List workflows in yaml format.

By default, workflows are output in json format, one per line (jsonl).

The --yaml flag will output the workflows in structured yaml format. If multiple workflows are being listed, each one will be separated by a blank line.

- **--base-url**

Base URL of the Ngenea Hub API, which operations will be performed against.

This can be used to perform Ngenea Hub operations on a remote server.

If not specified, the default is `http://localhost:8000/api`

- **-c, --config**

The path to an alternative configuration file.

If not specified, the default configuration paths will be used. The default paths are in the user's HOME directory `$HOME/.config/ngenea/ngenea-client.conf`, and the global configuration at `/etc/ngenea/ngenea-client.conf`

See **ngenea-client.conf(5)** for more information on the configuration format.

Command line options take precedence over any corresponding config file settings.

- **--client-key**

Ngenea Hub authentication client key.

This can be generated via the Ngenea Hub REST API, or using **ngclient(1)** authenticate

- **-h, --help**

Prints the help message.

EXAMPLES

GENERATE A CLIENT KEY

```
$ ngclient authenticate --username pixadmin -k ngclient-pixadmin
pixadmin's password:

jDBh2cRk6.LswQfylT2BtGigtYUWhMB1iipJmQNgr
```

The following examples assume that the client key has been saved in the default config file.

GET AN EXISTING WORKFLOW

```
$ ngclient workflows list 1
{"id": 1, "name": "migrate", "label": "Migrate", "icon_classes":
["fa fa-cloud fa-stack-2x text-success", "fa fa-angle-up fa-stack-2x
text-light"], "discovery": "recursive", "enabled": true, "visible":
true, "fields": [], "filter_rules": [{"type": "all", "state": "all",
"action": [{"name": "dynamo.tasks.migrate"}], "description":
"Migrates a file off from a given path"]}]}
```

LIST ALL EXISTING WORKFLOWS IN YAML FORMAT

```
$ ngclient workflows list --yaml
id: 1
name: migrate
label: Migrate
discovery: recursive
...
enabled: true
visible: true

id: 2
name: premigrate
label: Premigrate
discovery: recursive
...
enabled: true
visible: true

...
```

(the above example output has been truncated)

IMPORT A CUSTOM WORKFLOW

Using the following workflow definition in json format

```
$ cat overwrite_workflow.json
{"name": "recall_overwrite", "label": "Overwrite On Recall",
"icon_classes": ["fa fa-cloud fa-stack-2x text-primary", "fa fa-
caret-down fa-stack-2x text-light"], "filter_rules": [{"type":
"all", "state": "all", "action": [{"name":
"dynamo.tasks.reverse_stub", "site": "*destinationsite",
"overwrite": true}]}], "fields": [{"name": "destinationsite",
"type": "enum[site]", "label": "Destination Site", "value": "site"}]}
```

Import the workflow as follows

```
ngclient workflows import overwrite_workflow.json
```

RENAME A WORKFLOW

With the change in yaml format, using '-' to read from stdin

```
echo "name: overwrite_on_recall" | ngclient workflows update 6 -
```

DELETE A WORKFLOW

```
ngclient workflows delete 6
```

AVAILABILITY

Distributed as part of the ngenea-hub-client rpm, or the ngclient wheel (Python) for non-Red Hat based systems.

The ngclient wheel can be installed and run on any operating system.

SEE ALSO

ngclient(1), ngenea-client.conf(5)

LICENSE

2021 ArcaPix Limited

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

ngclient-features

NGCLIENT-FEATURES

SYNOPSIS

ngclient features list [options...]

ngclient features enable name [options...]

ngclient features disable name [options...]

DESCRIPTION

The list command is used to list available feature flags for Ngenea Hub.

The enable and disable commands can be used to enable a named feature in Ngenea Hub.

Base URL and API key settings can be read from a config file, rather than being passed on the command line. See **ngenea-client.conf(5)** for more information on the configuration format. CLI flags take precedence over config file settings.

Interacting with features requires Ngenea Hub authentication. The **ngclient(1)** authenticate command can be used to generate a client key from a username or access token.

OPTION SUMMARY

name	Name of the feature to enable or disable
--json	List features in json format
--base-url	Base URL of the {{ brand_name }} API
-c, --config CONFIG	Alternative configuration file path
--client-key KEY	Client API key to authenticate with
-h, --help	Print help message and exit

OPTIONS

- **--json**

List features in json format.

By default, the list command will report features in a table-based format. The --json flag will report features in json format instead, one per line.
- **--base-url**

Base URL of the Ngenea Hub API, which operations will be performed against.

This can be used to perform Ngenea Hub operations on a remote server.

If not specified, the default is http://localhost:8000/api
- **-c, --config**

The path to an alternative configuration file.

If not specified, the default configuration paths will be used. The default paths are in the user's HOME directory \$HOME/.config/ngenea/ngenea-client.conf, and the global configuration at /etc/ngenea/ngenea-client.conf

See **ngenea-client.conf(5)** for more information on the configuration format.

Command line options take precedence over any corresponding config file settings.

- **--client-key**

Ngenea Hub authentication client key.

This can be generated via the Ngenea Hub REST API, or using **ngclient(1)** authenticate

- **-h, --help**

Prints the help message.

EXAMPLES

The following examples assume that the client key has been saved in the default config file.

LIST AVAILABLE FEATURES

```
$ ngclient features list
[X] searchui           Enable search features in the UI
[ ] bandwidth_controls Enable bandwidth controls in the UI
[ ] rbac               Enable role-based access controls
```

(The above are just examples and may not reflect actual feature flags)

ENABLE A FEATURE

```
ngclient features enable rbac
```

DISABLE A FEATURE

```
ngclient features disable searchui
```

AVAILABILITY

Distributed as part of the ngenea-hub-client rpm, or the ngclient wheel (Python) for non-Red Hat based systems.

The ngclient wheel can be installed and run on any operating system.

SEE ALSO

ngclient(1), ngenea-client.conf(5)

LICENSE

2021 ArcaPix Limited

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR

CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

ngclient-settings

NGCLIENT-SETTINGS

SYNOPSIS

ngclient settings refresh [options...]

DESCRIPTION

The settings refresh command is used to trigger a refresh of site settings in Ngenea Hub.

This means loading the latest settings from PixStor on one or all sites into Ngenea Hub. This may be useful if a setting has been manually changed on a site, or if the settings displayed in Ngenea Hub are otherwise wrong or out-of-date.

Base URL and API key settings can be read from a config file, rather than being passed on the command line. See **ngenea-client.conf(5)** for more information on the configuration format. CLI flags take precedence over config file settings.

Interacting with features requires Ngenea Hub authentication. The **ngclient(1)** authenticate command can be used to generate a client key from a username or access token.

OPTION SUMMARY

--site settings from	Name of a specific site to refresh
--base-url	Base URL of the {{ brand_name }} API
-c, --config CONFIG	Alternative configuration file path
--client-key KEY	Client API key to authenticate with
-h, --help	Print help message and exit

OPTIONS

- **--site**

Name of a specific site to refresh settings on.

If this flag isn't specified, settings will be refreshed for all sites registered in Ngenea Hub

- **--base-url**

Base URL of the Ngenea Hub API, which operations will be performed against.

This can be used to perform Ngenea Hub operations on a remote server.

If not specified, the default is `http://localhost:8000/api`

- **-c, --config**

The path to an alternative configuration file.

If not specified, the default configuration paths will be used. The default paths are in the user's HOME directory `$HOME/.config/ngenea/ngenea-client.conf`, and the global configuration at `/etc/ngenea/ngenea-client.conf`

See **ngenea-client.conf(5)** for more information on the configuration format.

Command line options take precedence over any corresponding config file settings.

- **--client-key**

Ngenea Hub authentication client key.

This can be generated via the Ngenea Hub REST API, or using **ngclient(1)** authenticate

- **-h, --help**

Prints the help message.

EXAMPLES

The following examples assume that the client key has been saved in the default config file.

REFRESH SETTINGS ON ONE SITE

```
$ ngclient settings refresh --site london
```

REFRESH SETTINGS ON ALL SITES

```
ngclient settings refresh
```

AVAILABILITY

Distributed as part of the `ngenea-hub-client` rpm, or the `ngclient` wheel (Python) for non-Red Hat based systems.

The `ngclient` wheel can be installed and run on any operating system.

SEE ALSO

`ngclient(1)`, `ngenea-client.conf(5)`

LICENSE

2023 ArcaPix Limited

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

ngclient-sites

NGCLIENT-SITES

SYNOPSIS

ngclient sites refresh [options...]

DESCRIPTION

The sites refresh command is used to trigger a refresh of site-specific metadata in Ngenea Hub.

Site metadata includes things like storage pools, filesystems, analytics, etc. This metadata is typically refreshed periodically, but if the data currently returned from Ngenea Hub seems wrong or out of date, this command can be used to force a refresh.

Base URL and API key settings can be read from a config file, rather than being passed on the command line. See **ngenea-client.conf(5)** for more information on the configuration format. CLI flags take precedence over config file settings.

Interacting with features requires Ngenea Hub authentication. The **ngclient(1)** authenticate command can be used to generate a client key from a username or access token.

OPTION SUMMARY

--site	Name of a specific site to refresh
settings from	
--base-url	Base URL of the {{ brand_name }} API
-c, --config CONFIG	Alternative configuration file path
--client-key KEY	Client API key to authenticate with
-h, --help	Print help message and exit

OPTIONS

- **--site**

Name of a specific site to refresh metadata on.

If this flag isn't specified, metadata will be refreshed for all sites registered in Ngenea Hub

- **--base-url**

Base URL of the Ngenea Hub API, which operations will be performed against.

This can be used to perform Ngenea Hub operations on a remote server.

If not specified, the default is `http://localhost:8000/api`

- **-c, --config**

The path to an alternative configuration file.

If not specified, the default configuration paths will be used. The default paths are in the user's HOME directory `$HOME/.config/ngenea/ngenea-client.conf`, and the global configuration at `/etc/ngenea/ngenea-client.conf`

See **ngenea-client.conf(5)** for more information on the configuration format.

Command line options take precedence over any corresponding config file settings.

- **--client-key**

Ngenea Hub authentication client key.

This can be generated via the Ngenea Hub REST API, or using **ngclient(1)** authenticate

- **-h, --help**

Prints the help message.

EXAMPLES

The following examples assume that the client key has been saved in the default config file.

REFRESH METADATA ON ONE SITE

```
$ ngclient sites refresh --site london
```

REFRESH METADATA ON ALL SITES

```
ngclient sites refresh
```

AVAILABILITY

Distributed as part of the `ngenea-hub-client` rpm, or the `ngclient` wheel (Python) for non-Red Hat based systems.

The `ngclient` wheel can be installed and run on any operating system.

SEE ALSO

`ngclient(1)`, `ngenea-client.conf(5)`

LICENSE

2023 ArcaPix Limited

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

ngenea-client.conf

NGENEACLIENTCONF

SYNOPSIS

The Ngenea Hub client configuration files is used to configure **ngclient(1)** and **transparent_recall(1)**

The default config file locations are in the user's HOME directory \$HOME/.config/ngenea/ngenea-client.conf, with a global config at /etc/ngenea/ngenea-client.conf.

If both configuration files exist, the user config will take precedence, with the global config used for any values not specified in the user config.

For example

```
# global config
[settings]
base_url = http://10.172.0.23:8000
site = default

# user config
[settings]
site = mysite
```

would result in base_url = http://10.172.0.23:8000/api, since it's not specified in the user config, and site = mysite since the value from the user config takes precedence.

NOTE - unless explicitly specified with the --config flag, both ngclient(1) and transparent_recall(1) will use this same default config files.

If a config file is explicitly specified with --config, the default configs will not be considered at all.

Command line options take precedence over any corresponding config file settings.

FILE FORMAT

ngenea-client.conf(5) uses an ini-style format.

It is made up of key = value lines under the [settings] section header.

```
[settings]
client_key = mykey
```

Boolean type values can be either true, false, yes, or no (case-insensitive)

Additional sections, or unrecognised keys are ignored.

PARAMETERS

- **base_url**

Base URL of the Ngenea Hub API, which operations will be performed against.

This can be used to perform Ngenea Hub operations on a remote server.

If not specified, the default is `http://localhost:8000/api`

- **client_key**

Ngenea Hub authentication client key.

This can be generated via the Ngenea Hub REST API, or using **ngclient(1)** authenticate

- **site**

The default site to use for workflows.

For migrate and recall, this is the site where the workflows execute. For send, this is the source site, from which files are sent.

- **queue**

The default queue to use for workflows. By default, source queue is set as default

For migrate and recall, this is the queue where the workflows execute. For send, this is the source queue, from which files are sent.

- **wait**

Whether to wait for workflows to complete.

If true (default), tools will wait for the workflow to complete, subject to timeout.

If false, tools will exit immediately. The workflow will continue to execute independently. In that case, the workflow can be monitored in the Ngenea Hub UI

- **timeout**

How long to wait for workflows to complete, in seconds.

If not set, tools will wait indefinitely.

If the workflow doesn't complete within the timeout, the client will exit with an error. The workflow itself may continue to execute.

- **hydrate**

For **ngclient(1)** send, controls whether sent files are hydrated on the target.

If false, files are only reverse stubbed on the target.

- **api_secure_verify**

Whether to enable/disable TLS Verification when communicating with Ngenea Hub REST API.

This is particularly useful when Ngenea Hub REST API is behind a self-signed certificate.

If false, TLS verification will be disabled.

If not specified, the default is true.

EXAMPLE

```
[settings]
base_url = http://mypixserver:8000
client_key = ...
site = mysite
queue = standard
wait = true
timeout = 180
hydrate = true
api_secure_verify = true
```

SEE ALSO

ngclient(1), transparent_recall(1)

LICENSE

2021 ArcaPix Limited

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

transparent_recall

TRANSPARENT_RECALL

SYNOPSIS

transparent_recall file [--config CONF]

DESCRIPTION

transparent_recall is a tool for recalling individual files via Ngenea Hub

Performing recalls via Ngenea Hub allows for monitoring progress via the Ngenea Hub UI. Individual recall tasks performed on demand, but for reporting are grouped together into one job per hour.

transparent_recall can be called directly to recall files, but typically would be installed as a filesystem policy rule. See **TRANSPARENT RECALL POLICY** for more info.

OPTION SUMMARY

file	One or more directory to export events from
--config CONF	Alternative configuration file location
-h, --help	Print help message and exit

OPTIONS

- **--config**

The path to an alternative configuration file.

If not specified, the default configuration paths will be used. The default paths are in the user's HOME directory \$HOME/.config/ngenea/ngenea-client.conf, and the global configuration at /etc/ngenea/ngenea-client.conf

See the **CONFIGURATION** section below and ngenea-client.conf(5) for more information.

Command line options take precedence over any corresponding config file settings.

- **--queue**

Queue to perform the transparent recall on.

- **-h, --help**

Prints the help message.

CONFIGURATION

transparent_recall requires authentication to be able to perform recalls via Ngenea Hub. To authenticate, a valid client_key must be placed in the configuration file.

A client key can be generated via the Ngenea Hub REST API, or using the **ngclient(1)** authenticate command.

Minimally, the configuration must include this client_key, as well as the site where recalls are performed.

```
[settings]
client_key = ...
site = thissite
```

The site must match the the node where the recall was triggered.

The queue can be specified in the configuration; otherwise, it will be set to the default option, default.

transparent_recall will respect any **ngclient(1)** recall configuration options, except for recursive. This includes timeout; by default it will wait indefinitely for the recall to complete.

See **ngenea-client.conf(5)** for more information on the configuration format and additional options.

TRANSPARENT RECALL POLICY

Transparent recall, by definition, is intended to be triggered automatically when an offline file is opened for reading or writing.

To enable transparent recall functionality using **transparent_recall**, the following rules can be added to the filesystem placement policy.

```
/* BEGIN: NGENEA TRANSPARENT RECALL POLICY */

define(xattr_stbsz,[INTEGER(XATTR('dmapi.APXstbsz'))])

RULE FileOpen EVENT 'OPEN'
    ACTION(SetDataEvent(0, OP_READ, CASE WHEN xattr_stbsz IS NULL
THEN 0 ELSE xattr_stbsz END) AND
        SetDataEvent(1, OP_WRITE+OP_TRUNC, 0))
    WHERE LENGTH(XATTR('dmapi.APXguid'))>0 AND
CountSubstr(MISC_ATTRIBUTES,'V')>0

RULE FileOpen_else EVENT 'OPEN' DIRECTORIES_PLUS

RULE FileData EVENT 'DATA'
    ACTION(system_lw('/usr/bin/python3 /usr/bin/transparent_recall '
|| getDetail('path_name') )=0)
    WHERE LENGTH(XATTR('dmapi.APXguid'))>0 AND
CountSubstr(MISC_ATTRIBUTES,'V')>0

RULE FileData_else EVENT 'DATA' DIRECTORIES_PLUS

/* END: NGENEA TRANSPARENT RECALL POLICY */
```

If transparent recall (EVENT) rules are already installed for ngenea (native), these rules should replace those equivalent rules.

Don't replace any rules besides the ngenea EVENT rules, e.g. don't replace any SET POOL rules.

See **mmchpolicy(1)** for how to change the filesystem placement policy

Adjusting the transparent recall queue

If an alternative queue has been established, for example `transparent-recall` it can be used for all transparent recall operations by using this policy alternative:

```
ACTION(system_lw('/usr/bin/python3 /usr/bin/transparent_recall --
queue transparent-recall ' || getDetail('path_name') )=0)
```

This substitutes the main `ACTION` to allow for all transparent recall operations to go through the custom queue exclusively. This means transparent recalls will not be effected by any other operations happening on the main

For more details on how to create and enable these custom queues, refer to the configuration section of the [ngeneahub](#) documentation.

WARNING

Due to limitations of GPFS, the above policy will not work on any paths which contain whitespace. Transparent recall will simply error.

An alternative policy substitutes the `system_lw` line for

```
ACTION(system('/usr/bin/python3 /usr/bin/transparent_recall ''' ||
getDetail('path_name') || ''')=0)
```

This will handle paths containing whitespace, but not those containing single quotes. Again, transparent recall will error.

There is currently no approach which works in all cases. You must choose the version which is most compatible with the sorts of paths you have on your system.

You can also provide an alternative queue for this revised policy similar to the above:

```
ACTION(system('/usr/bin/python3 /usr/bin/transparent_recall --queue
transparent-recall ''' || getDetail('path_name') || ''')=0)
```

TROUBLESHOOTING

When attempting to read an offline file, if the read process reports "Operation not permitted", and it's not due to permissions, the most likely cause is that the recall failed.

Logs for the **transparent_recall** command invocation can be found at `/var/adm/ras/mmfs.log.latest`

Logs for the transparent recall job can be viewed via Ngenea Hub.

WARNING - if reading a file triggers a recall, the read request will block until recall exits; it can't be interrupted (Ctrl+C) or killed (kill -9). If the recall job is 'stuck' and

no timeout is set, the only way to make the read process exit is to kill the recall job via Ngenea Hub.

AVAILABILITY

Distributed as part of the `ngenea-hub-client` rpm, or the `ngclient` wheel (Python) for non-Red Hat based systems.

Note - **transparent_recall** makes use of `flock(2)`, so can only be used on Unix-base operating systems.

SEE ALSO

`ngclient(1)`, `ngenea-client.conf(5)`, `ngrecall(1)`, `mmchpolicy(1)`

LICENSE

2021 ArcaPix Limited

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Backup

Backing Up HUB Static Configurations

Warning: The backup process only includes static configurations and selected database tables. Job and task-related data are **not included** in the backup.

Ngenea Hub provides a set of commands to efficiently **backup** and **restore** its critical static configurations and selected database tables, ensuring quick recovery and minimal downtime during system upgrades or failures.

These commands allow administrators to safeguard important configuration data, such as **schedules**, **users**, **workflows**, **external targets**, and **spaces**, while excluding transient job and task data, which helps streamline the process and reduce the overall backup size.

```
ngeneahubctl backup --help
```

```
Usage: ngeneahubctl backup [OPTIONS] COMMAND [ARGS]...
```

```
Backup and restore ngeneahub static configurations
```

```
Options:
```

```
--help Show this message and exit.
```

```
Commands:
```

```
export   Export ngeneahub static configurations to a file
restore  Restore ngeneahub static configurations from a file
```

Backup

The backup process captures key Ngenea Hub configurations and selected database tables, providing a snapshot of the system's critical static configurations.

To perform a backup Ngenea Hub's database container **ngeneahub_db_1** must be running or Ngenea Hub fully operational.

To start the Ngenea Hub database container separately run;

```
docker start ngeneahub_db_1
```

Command:

```
ngeneahubctl backup export --help

Usage: ngeneahubctl backup export [OPTIONS]

Export ngeneahub static configurations to a file

Options:
  -o, --output DIRECTORY  Location to store generated archive.
                          Default
                          '$(PWD)'
  --help                  Show this message and exit.
```

This command performs the following:

- **Database dump:** Exports selected tables related to schedules, users, workflows, external targets, global configurations, spaces etc into a SQL file.
- **Configuration backup:** Copies static configuration files from key directories:
 - /etc/ngenea/postgres (PostgreSQL settings)
 - /etc/ngenea/redis (Redis configurations)
 - /etc/ngenea/rabbitmq (RabbitMQ configurations)
 - /etc/ngenea/ngenea-client.conf (Hub client configuration)
 - /etc/sysconfig/ngeneahub (Hub configuration)
- **Archive creation:** Packs the SQL dump and configuration files into a compressed archive (ngeneahub_backup_<timestamp>.tar.gz) which will be written to the current working directory or a directory as specified by -o, --output command line option.

Restore

Note: Depending on the scenario a wipe maybe required before restoring a backup.

The restore process reinstates the backed-up database tables and configuration files, returning the Hub to its operational state following a failure or upgrade.

To perform a restoration Ngenea Hub should be fully operational.

Note: After performing a restoration all services needs to be restarted

Command:

```
ngeneahubctl backup restore <backup-file>
```

```
ngeneahubctl backup restore --help
```

```
Usage: ngeneahubctl backup restore [OPTIONS] PATH
```

```
Restore ngeneahub static configurations from a file
```

```
Options:
```

```
--help Show this message and exit.
```

This command performs the following:

- **Unpack the backup archive:** Extracts the contents of the backup file.
- **Restore configurations:** Copies the saved configuration files back into their respective directories, re-establishing the static system setup.
- **Truncate and reset tables:** Clears the selected database tables, resetting them to ensure no conflicts with the incoming data.
- **Import the SQL dump:** Inserts the backed-up data into the database, restoring schedules, users, workflows, targets, and spaces.

Ngenea Hub Changelog

Release notes - Ngenea Hub - 2.5.4: 2025-02-28

=====

Improvement

HUB2-2547 - Allow users to specify preferred queue for a workflow via API using "preferred_queue"

Bug

HUB2-2520 - Automated queue cleanup task no longer throws an exception on startup

HUB2-2541 - Space creation location picker now returns correct folders when not browsed prior in the file browser

Release notes - Ngenea Hub - 2.5.3: 2025-02-19

=====

Bug

HUB2-2611 - Ensured shares defined before version 2.5.3 could be removed from the salt pillar
HUB2-2545 - Renaming an NFS share no longer creates a new match-all share
HUB2-2588 - Creating a Samba share no longer results in endless looping jobs
HUB2-2600 - Existing NFS clients are no longer updated when new NFS client is added
HUB2-2601 - Editing an existing NFS share no longer has the potential to create new match-all shares

Release notes - Ngenea Hub - 2.5.2: 2025-02-10
=====

Improvement

HUB2-2531 - Hub login support for LDAPs and STARTTLS

Bug

HUB2-2545 - Renaming NFS shares no longer creates a new wildcard share

Release notes - Ngenea Hub - 2.5.1-1: 2025-01-13
=====

Bug

HUB2-2455 - Fixed tasks unexpectedly exiting while attempting to log job statistics

Release notes - Ngenea Hub - 2.5.0-2: 2024-12-13
=====

Updated Ngenea HSM dependency to 1.30.1-1

Release notes - Ngenea Hub - 2.5.0: 2024-11-13
=====

Feature

HUB2-1965 - Support for setting a Spaces permission mode
HUB2-2198 - ngeneahubctl now supports database export
HUB2-2199 - ngeneahubctl now supports database import
HUB2-2280 - Provide DownloadOnly workflow in file browser

Improvement

HUB2-1333 - Started tasks no stuck after a sudden worker exit

HUB2-1519 - Improve scanning performance for policies against single spaces
HUB2-1643 - ngclient now accepts relative paths
HUB2-2037 - Ngenea targets can be unlinked from all sites
HUB2-2068 - Added additional safety guards to ngeneahubctl wipe
HUB2-2099 - The Site management page now displays when sites are configured on their respective nodes
HUB2-2119 - ngclient now identifies authentication issues
HUB2-2240 - Stopping/restarting a worker causes tasks to get killed and lost
HUB2-2286 - Support ConfigFile, LocalFileRegex, LocalSymlinkTargets, RemoteLocationXAttrRegex, StorageKey in ngenea target advanced keys
HUB2-2294 - NFS share now display with multiple non-CIDR clients in the UI
HUB2-2300 - Ensure the email entry is omitted on the users table if not set
HUB2-2301 - Star character is no longer displayed when Search query is submitted
HUB2-2305 - Ensure Kerberos connections are bound as authenticated user
HUB2-2306 - The "Minimize" tooltip for left bar is no longer incorrectly sized
HUB2-2324 - Workflows that select a secondary site now default to the "default" queue if not provided
HUB2-2333 - Google cloud storage targets can now use CredentialsFile as the only credentials provider
HUB2-2335 - When rabbitmq is not the target broker the container no longer spawns
HUB2-2353 - Either credentials_json or credentials_files can be used when defining a target with Google Cloud Storage

Bug

HUB2-1288 - Fix page scrollbar sizing in Job list page
HUB2-1424 - Ensure Hub cannot overwrite existing users and groups from pixdjango
HUB2-1460 - Provide values for usage and capacity in Policy UI when no quota is in place on a space
HUB2-1584 - Resolve issue with per-process lock acquisition in transparent recall
HUB2-1806 - Ensure migrations to ngenea are limited to one source pool
HUB2-1937 - The automated job to expire old jobs has been optimised
HUB2-1955 - Resolve 500 error when deleting a schedules workflow
HUB2-1983 - NFS shares are now correctly configured with multiple clients
HUB2-1984 - Resolve traceback observed: AttributeError: 'QuerySet' object has no attribute 'objects' when using check_sync_state
HUB2-1986 - Ensured every client for the same NFS share have the same fsid
HUB2-1999 - Resolve GCS bucket creation failure due to mangled json auth
HUB2-2002 - Recursive navigate now reports failed paths
HUB2-2084 - Remove redis connection log error under normal operation

HUB2-2174 - Job descriptions are now case sensitive
HUB2-2216 - Non-existing ngenea target now correctly raises a 404
HUB2-2220 - Ensure Space mountpoint is not presented with double forward slash in the space wizard
HUB2-2224 - reverse_stub task no longer lists "--overwrite-local".
HUB2-2277 - Thread counts can now be explicitly for each default queue function
HUB2-2279 - Browsing external targets now correctly parse errors from listing cloud files
HUB2-2283 - Fix inability to change Space card colour
HUB2-2284 - Remote folder not matched of top level folder selection no longer causes import from site jobs to fail
HUB2-2307 - Shares can now be browsed when Samba allowed_hosts is null
HUB2-2317 - Target name too long error not correctly reported to user in UI
HUB2-2318 - Targets now correctly supports StorageKey numerical strings
HUB2-2322 - The 'invalid character' error message when browsing a space no longer appears when there are no files to browse

Release notes - Ngenea Hub - 2.4.4: 2025-02-07

=====

Feature

HUB2-2479 - Improved snapdiff based workflows performance providing snapdiff includes and excludes on the worker

Release notes - Ngenea Hub - 2.4.3: 2024-10-16

=====

Improvement

HUB2-2245 - Snapdiff based workflow moves now support same named queues

Release notes - Ngenea Hub - 2.4.2: 2024-10-14

=====

Improvement

HUB2-2214 - Transparent recall now supports alternative queues

Bug

HUB2-2170 - Resolves LDAP plaintext authentication failure
HUB2-2193 - GPFS Policies scans now run across multiple nodes
HUB2-2209 - Scheduled jobs without a second site now correctly run
HUB2-2225 - All ngenea target keywords are now supported

Release notes - Ngenea Hub - 2.4.1: 2024-10-07

=====

Bug

HUB2-2209 - Workflows without a destination site now correct run from a schedule

Release notes - Ngenea Hub - 2.4.0: 2024-10-04

=====

Feature

Multiple Job Queues

RabbitMQ as an optional broker

Improvement

HUB2-908 - Limit ngenea target creation to specific sites via API

HUB2-1544 - Move ngenea target settings to dedicated / external_targets endpoint

HUB2-1640 - Attempt to premigrate files from snapshot even if live copy doesn't exist

HUB2-1776 - Summary page update for Ngenea bucket configuration

HUB2-1898 - Improve performance of Job no-op filter

HUB2-1970 - Make the job link middle- & right- clickable on "job is created" popup

HUB2-1989 - API-Keys no longer require an email for creation

HUB2-2005 - Unify order of statuses in Job specific page

HUB2-2117 - Make node heartbeat check interval configurable

Bug

HUB2-599 - Workflow schedules will now wait until space creation completes on all sites to start

HUB2-1382 - The search_keyword argument correctly filters jobs via ID

HUB2-1641 - Snapdiff tasks no longer skip files not on the live filesystem

HUB2-1799 - Search filter chips no longer disappear after a search completes

HUB2-1802 - Policy related jobs are no longer always stuck in pending

HUB2-1973 - Disable rpm build-id debug files

HUB2-1994 - Login page now has correct page title

Documentation

HUB2-1820 - Update existing Plugin docs for new DAG capabilities

HUB2-1866 - Existing worker task options have been updated

Release notes - Ngenea Hub - 2.3.2: 2024-08-23

=====

Bug

HUB2-1893: Reverted; due to causing Bidirectional sync to fail intermittently

Release notes - Ngenea Hub - 2.3.1: 2024-08-09

=====

Improvement

HUB2-1685: Added documented new engine functionality to the worker
HUB2-1016: Double clicking thumbnails within search now opens a full screen preview
HUB2-1457: Fileset snapshots retention period language is improved for international users
HUB2-1895: Snapdiff workflows now support root filesets

Bug

HUB2-1672: Fix engine deadlock issue when running a non null discovery job
HUB2-1704: Shares deleted on the UI are now always deleted on the target system
HUB2-1707: Snapshot schedule update task is submitted when editing shares
HUB2-1762: Settings jobs can now be cancelled
HUB2-1859: Resolve rare race condition when storing task results
HUB2-1865: Ganesha exports now stored with numerical value for allowing persistence on shares reload
HUB2-1893: Snapdiff rotate task now correctly runs after processing task is revoked
HUB2-1916: Resolved flickering on second page of tasks in a job
HUB2-1944: When creating a space, the scheduled job now waits for the space to be completed before starting
HUB2-1957: Resolved race condition in site settings task
HUB2-1985: Deleting a client from an NFS export now removes the share from the site
HUB2-1798: UI automatic file browser refresh no longer display some restricted folders

Release notes - Ngenea Hub - 2.3.0: 2024-07-05

=====

Feature

Job Reporting improvements
LDAP and Kerberos integration
Space deletion via the UI
Dynamic job batching
Database size reduction with tunable settings

Improvement

HUB2-1190: Default locking method options are now applied to all default workflows
HUB2-1804: Custom plugin docs have been updated to support new engine functionality
HUB2-1891: Job details page now renders regardless of statistics being loaded
HUB2-1892: API_TIMEOUT and GATEWAY_TIMEOUT now default 10 minutes
HUB2-1356: Various database size reduction optimisations
HUB2-1644: Recursive jobs now use less calls to mmlsfileset
HUB2-1644: Recursive jobs now no longer navigate into excluded directories
HUB2-1691: Removed search is halted message when interrupting a search
HUB2-1700: SMB shares now supports definitions in the hosts to allow/deny
HUB2-1719: Ngenea worker analytics requests can now have their timeouts configured
HUB2-1281: Error and failed jobs can now be re-submitted
HUB2-1610: Job descriptions have been added to all non-settings jobs
HUB2-1621: Users can now to filter paths across all pages when searching for a related job
HUB2-1462: Installed internal SSL certificates can now be displayed on the command line
HUB2-1613: Plugins and internal tasks now use jobid consistently
HUB2-1664: Transparent recall is now using the new task engine

Bug

HUB2-1895: Snapdiff workflows now support root fileset snapshots
HUB2-1414: Circular symbolic links no longer cause analytics errors
HUB2-1491: Shares now correctly cannot contain white spaces
HUB2-1683: Users and groups are no longer allowed to contain white spaces
HUB2-1693: Alerts are now shown immediately
HUB2-1821: Site options are now pre-populated in bidirectional sync setup
HUB2-1839: Analytics caches are correctly removed on an analytics error
HUB2-1861: Downloading a file list now provides the correct output
HUB2-1550: Fixed the runtime for paused jobs
HUB2-1614: Default spaces are created correctly for filesystems not mounted at the standard default
HUB2-1694: Cancelling a snapdiff with specific timings no longer causes the job to remain in an unknown state
HUB2-1701: Permissions are now correctly applied to the alerts on the the site page
HUB2-1706: Storage metric graphs now correctly display on lower resolutions
HUB2-1737: Site labels now default to their names if not set
HUB2-1464: Custom snapdiff jobs now correctly track progress correctly in job list page
HUB2-810: Space creation no longer fails due to unrelated salt state errors

HUB2-847: Site network settings now allow CIDR values as 0-32 , with 0 included
HUB2-1055: Only one job is now created when spaces are deleted
HUB2-1187: Un-necessary SSL warnings have been removed in worker logs
HUB2-1299: Ensure text in UI tooltips no longer go off the screen
HUB2-1396: Site colour picker shows multiple shades of a colour selected
HUB2-1466: None snapdiff job runtimes are now consistent between summary and detail views
HUB2-1478: Job ids can now be longer than 22 characters
HUB2-1480: Client certificate expiry is now set correctly internally
HUB2-1484: Clicking "x" for site filter chip now behaves consistently
HUB2-1492: Long space names now display correctly in the "location" section of space creation as well as summary screen
HUB2-1493: UI now correctly states the correct maximum name size for spaces
HUB2-1535: Job progress bar in the job listing page now correct reflect progress
HUB2-1540: Job progress bar in the job listing page now shows progress without manual refresh
HUB2-1551: Task details modal tooltip no longer blinks
HUB2-1566: Job "id" field is now correctly set for settings tasks
HUB2-1574: Ordering of jobs in the job listing page when sorting by state is now consistent
HUB2-1587: Internal task non_discovery_rule_filter now shows date started
HUB2-1589: ngclient authenticate now correctly uses the api_secure_verify from config file
HUB2-1603: Schedules for recreated filesets now correctly run
HUB2-1608: The rightbar no longer flickers in the job details page when scrolled down
HUB2-1609: Pending item stats have been removed snapdiff jobs as they do not use them
HUB2-1620: Non UTC datetime in custom DAG tasks now correctly refresh
HUB2-1628: Ensured that on custom plugin install that the internal packages do not change version
HUB2-1629: Populate settings job no longer get stuck in the "NEW" state with an "Unknown" owner
HUB2-1631: Delete space task now runs even if create space hasn't completed yet
HUB2-1635: Transparent recall no longer fails if existing transparent recall job does not exist
HUB2-1666: The filter bar now always displays correctly in the file browser
HUB2-1669: The current view is now retaining on jobs detail page refresh

Release notes - Ngenea Hub - 2.2.5: 2024-06-21

=====

Improvement

HUB2-1852 - Ensured that storage analytics can have its request

timeout configured

Fix

HUB2-1852 - Pass API_TIMEOUT from the hub sysconfig file to the API container

Release notes - Ngenea Hub - 2.2.4: 2024-05-31

=====

Improvement

HUB2-1644 - Recursive navigate tasks no longer call mmlsfileset more than once

Bug

HUB2-1680 - Task failures no longer stop downstream tasks to stay in PENDING

Release notes - Ngenea Hub - 2.2.3: 2024-05-21

=====

Improvement

HUB2-1625 - The API wide timeout can now be configured

Bug

HUB2-1626 - The job details UI now waits for job results regardless of request duration

HUB2-1667 - The job list UI now waits for results regardless of request duration

HUB2-1682 - Job related web socket connections now waits for connection regardless of duration

HUB2-1619 - Jobs with the state "Failing" no longer disappear from jobs list page

Release notes - Ngenea Hub - 2.2.2: 2024-05-06

=====

Bug

HUB2-1630 - Fixed issue with bidirectional sync re-submission

Release notes - Ngenea Hub - 2.2.1: 2024-04-29

=====

Bug

HUB2-1630 - Transparent recall functionality now correctly submits with new engine

Release notes - Ngenea Hub - 2.2.0: 2024-03-28

=====

Feature

UI Colours
Search Tags
Custom Attributes for SMB
New DAG Job Engine

Improvement

HUB2-291 - Better wrap long space names
HUB2-365 - Different colors for task table chips instead of "gray" ones
HUB2-368 - Fix scrolling arrows showing up by default on rightbar
HUB2-652 - Utility for translating schedule object to user
HUB2-770 - Automatically name policy condition group titles
HUB2-863 - Task stats should apply the filter on job details page
HUB2-895 - Provide API support for editing search tags
HUB2-990 - Provide the ability to filter site alerts by severity
HUB2-1032 - Search: Support extensions starting with dot for core.extension filter
HUB2-1035 - Enforce unique space mount point (API)
HUB2-1072 - Ensure group titles wrap better
HUB2-1084 - Swap files above tasks in job details
HUB2-1089 - Provide the ability to set the number of executed policy threads in the UI
HUB2-1103 - Add buttons for pausing and resuming a job
HUB2-1104 - API response for job Owner is different
HUB2-1105 - Adjust Job details UI labels to new DAG statistics
HUB2-1107 - Adjust websocket responses to conform with new serializers
HUB2-1112 - Ensure NAS components are not visible if all sites in AD mode
HUB2-1162 - Site filtering for policies
HUB2-1213 - Ensure apbackup chip colours are consistent and correct site colour when input is changed
HUB2-1218 - Make top bar green
HUB2-1219 - Implement yellow progress bar across wizards
HUB2-1220 - Provide the ability to set the colour of the Site Chip in the Create Site wizard
HUB2-1222 - Updated colourful left bar menu buttons
HUB2-1223 - Updated login page presentation
HUB2-1224 - Cyclic colours for space cards
HUB2-1225 - Provide the ability to set the colour of the Space Card in the Space Wizard
HUB2-1226 - General CSS changes for button/checkbox/slider etc.

elements

- HUB2-1227 - Add SMB Custom Options Panel to Create Space Wizard
- HUB2-1228 - Add SMB Customs Options Panel to Update Space Modal
- HUB2-1230 - Modify multiselect dropdown component for search of a tag
- HUB2-1237 - Remove feature flags for stable features
- HUB2-1256 - Provide API support for Site chip colours
- HUB2-1259 - Increase intervals for site refreshes
- HUB2-1261 - Provide alphabetical ordering for managed keys
- HUB2-1263 - Add tag selector filter to Search filters (in main search filter list)
- HUB2-1264 - Add tag editor to search side panel for asset
- HUB2-1282 - Grant metadata update permission to worker service user
- HUB2-1291 - Switch site chips to double colour chips
- HUB2-1294 - Don't display muted alerts in the GUI
- HUB2-1310 - Update Hub favicon
- HUB2-1311 - Add danger notification
- HUB2-1312 - Make rightbar stay at the same place on scroll for global search page
- HUB2-1316 - Hide resubmit button for settings tasks
- HUB2-1328 - Better render "Date started" for cancelled tasks
- HUB2-1334 - Ensure backup UI components are only displayed if the site has nodes configured with the apbackup role
- HUB2-1346 - Implement database expiry page in global settings page
- HUB2-1358 - Make sticked sidebar on default for all elements
- HUB2-1376 - Re-expose grafana dashboard
- HUB2-1377 - Provide role lookup for apbackup feature
- HUB2-1378 - Add filter by job creator inside jobs
- HUB2-1389 - RPM of public facing html docs
- HUB2-1399 - Change text inside of tags multi-select
- HUB2-1401 - Remove Default Location toggle from space create wizard
- HUB2-1402 - Where a site only has one pool, auto-select it in the space creation drop-down
- HUB2-1411 - Document swagger URL
- HUB2-1417 - Add a base class to all non system worker tasks that check if it can run
- HUB2-1418 - Adjust the cancel process to use redis and change the cancelling states
- HUB2-1436 - Add the enable_plugin setting to the plugin docs
- HUB2-1438 - Type filter updates on Jobs page for DAG jobs
- HUB2-1442 - Job details show job status as Pausing when the job is in state Pausing
- HUB2-1446 - Remove settings task types from workflow filter list in the UI
- HUB2-1447 - Support for filtering on CANCELLING and PAUSING states in the UI
- HUB2-1450 - Document the /hubmetrics endpoint
- HUB2-1469 - Task stats should include PENDING task counts
- HUB2-1496 - Provide clarification that File Stats processes more than Files
- HUB2-1498 - Rename the "In-Progress" file stats label to "Pending"
- HUB2-1530 - Rename Job Type to Workflow in job creator

Bug

HUB2-395 - Prevent setting space mount point to existing path
HUB2-473 - Deleting any of the site config fields and submitting does not delete the fields
HUB2-581 - Resolve an issue whereby a resubmitted job displays Job Creator unknown
HUB2-703 - Client key name max length error
HUB2-807 - Invalid date message on filebrowser table
HUB2-812 - Resolve an issue whereby full screen for task logs does not display in full screen
HUB2-892 - Resolve an issue whereby a space cannot be deleted using the python shell without deleting snapshot_schedule first
HUB2-928 - Fix timeout message in search page
HUB2-946 - Small Browser Window Stylesheet Issue
HUB2-997 - Task type filter options are not loaded for some of the system jobs
HUB2-1006 - Ensure that site creation summaries are accurate
HUB2-1061 - Provide job not found page
HUB2-1079 - Resolve an issue whereby full screen video preview does not fullscreen
HUB2-1083 - Resolve an issue whereby joined sites have no job site labels
HUB2-1086 - Resolve an issue whereby key names can be obscured
HUB2-1136 - Very long file name takes all the width in file browser table
HUB2-1144 - Resolve an issue whereby a invalid nginx config can cause frontend service interruption
HUB2-1146 - Error handling error response from search
HUB2-1184 - Cursor jumps to end of line as you type in the space name textbox
HUB2-1185 - Job gets stuck in processing if permissions on ngmigrate do not allow execution
HUB2-1189 - Remove unneeded warnings in the worker logs
HUB2-1232 - Cannot add API key via UI if profile fields aren't set
HUB2-1242 - Resolve an issue whereby all NAS interfaces cannot be removed
HUB2-1245 - Resolve an issue whereby the Username is truncated in the top right
HUB2-1260 - Resolve an issue whereby the Sites page shifts off screen when the left bar is expanded
HUB2-1262 - Resolve an issue whereby flickering is observed when loading pages
HUB2-1279 - 500 Server error when trying to verify an auth token
HUB2-1284 - Resolve an issue whereby SAMBA configuration entries are duplicated on update or deletion
HUB2-1285 - Unable to save NAS settings
HUB2-1289 - Fix duplicate job detection preventing Samba keys from being updated more than once
HUB2-1290 - Resolve an issue whereby a basic auth login prompt is presented
HUB2-1293 - Pre-set public url is not populated on create site wizard
HUB2-1301 - Setting a schedule to 1 hour causes it to run every minute
HUB2-1307 - Unable to create NAS users after deleting NAS user
HUB2-1314 - Create and update Space wizard error if no ngenea targets

have been defined

- HUB2-1319 - Fix validation email for create user wizard & update user modal
- HUB2-1322 - Migration policy with three pools skips the second pool
- HUB2-1323 - Cannot create a policy with a schedule
- HUB2-1324 - Site settings - existing extra domain can not be deleted & validation needed for extra domain without name
- HUB2-1326 - public_url = None crashes UI for unconfigured site
- HUB2-1339 - Fix "get() returned more than one Server" issue
- HUB2-1359 - Fix height of the multi-select dropdowns to have valid minimum height
- HUB2-1360 - Selecting a tag for an item, assign it for all items with exact path, which is not unique
- HUB2-1362 - Job status appears as Ongoing for job in state UNKNOWN_FAILURE
- HUB2-1366 - Resolve an issue whereby ngenea target secrets are visible in log files
- HUB2-1367 - Spaces can suddenly become unlinked where multiple "sites" are defined for a given PixStor cluster
- HUB2-1379 - Filter input inside jobs is not working
- HUB2-1380 - transparent_recall does not honour api_secure_verify
- HUB2-1384 - Cannot change snapshot schedule on a space
- HUB2-1394 - SettingsTask deduplication logic prevents legitimate jobs from running
- HUB2-1400 - Redis will consume all available ram on a system over time
- HUB2-1403 - Fix settings refresh
- HUB2-1404 - Timestamp rules for policy weighting are incorrect
- HUB2-1412 - Space settings loads a blank screen if site label is not set
- HUB2-1413 - Job submitter does not show anything if site label is missing
- HUB2-1415 - Make sure the UI does not break when workflows are visible=False
- HUB2-1416 - Link to space path lost on login
- HUB2-1420 - Job submitter does not allow running workflow
- HUB2-1423 - DAGs submit custom tasks to the wrong queue
- HUB2-1425 - Tasks filter not applying to live updates
- HUB2-1426 - Unable to disable site apbackup
- HUB2-1434 - Tasks go into state "STARTED" even though they are still "PENDING"
- HUB2-1441 - Ngenea worker service cannot start up when no filesystem exists
- HUB2-1448 - Possible to cancel PAUSING jobs via the jobs overview
- HUB2-1455 - Job details rightbar shows ongoing when the job is in cancelling state
- HUB2-1467 - Job stats are not updated correctly when a DAG job is cancelled
- HUB2-1477 - Docker images manifest file is missing from rpms
- HUB2-1494 - UI is redirected to /undefined
- HUB2-1495 - Only in progress File stats are updating
- HUB2-1497 - Clicking the files stats number retrieves no content into the pop up JSON modal
- HUB2-1499 - job task list IDs are not being sorted correctly

HUB2-1500 - API request should call refresh token once expired, rather than logging the user out
HUB2-1509 - Long job descriptions break the UI on job table
HUB2-1517 - Cancelling a recursive job can cause tasks to be stuck in PAUSED
HUB2-1518 - Cancelling a job while "recursive_navigate" task is in started, causes objects to be stuck "In-progress"
HUB2-1521 - Error in return status of ngmigrate task
HUB2-1549 - Completed time not set for cancelled job with failed tasks
HUB2-1559 - UI does not allow having empty emails on user update modal

Release notes - Ngenea Hub - 2.1.1: 2024-02-08

=====

Bug

HUB2-1367 - Spaces can become unlinked where multiple "sites" are defined on the same PixStor cluster

Release notes - Ngenea Hub - 2.1.0: 2023-12-22

=====

Feature

Ability to configure Ngenea Backup
Ability to manage NAS users and groups
Ability to view disk metrics
New storage info page
New Alerts UI

Release notes - Ngenea Hub - 2.0.2: 2023-11-17

=====

Feature

New Search page, integrating PixStor Search into Hub
New Policies page, to configure and run data migration policies from the Hub UI

HUB2-27 - Prevent access to UI components according to user permissions
HUB2-660 - Display all GPFS filesystems in Site Dashboard
HUB2-691 - Ability to give space shares custom names
HUB2-692 - Ability to create multiple shares per space at different paths
HUB2-754 - API to fetch settings tasks list for a job
HUB2-791 - Global view of Jobs in the UI
HUB2-792 - Support for submitting custom workflows in the UI
HUB2-823 - Support for setting analytics and metrics URL from UI

HUB2-828 - Ability to specify target path for Space creation
HUB2-831 - Filesystems API endpoint
HUB2-1003 - System folders are hidden in the UI
HUB2-1014 - Ability to bookmark and link locations in the file browser
HUB2-1043 - Ability to view job file lists in job details page rightbar
HUB2-1071 - Added DNS search domains to Site settings
HUB2-1075 - Support for editing job Schedules in the UI

Improvement

HUB2-370 - Improved progress bars proportion in job details page
HUB2-527 - Sensitive settings such as ngenea secret keys are hidden from the UI and API
HUB2-785 - Include share path in file list for share jobs
HUB2-795 - Jobs page view persists when refreshing the page
HUB2-825 - All members of the group can be viewed via the "view all members" button
HUB2-833 - Improved UI for settings Job details
HUB2-834 - Adds user-friendly job descriptions
HUB2-835 - Snapshot time can only be set when the frequency is greater than 1 day
HUB2-861 - Ability to search for a job by ID on jobs page
HUB2-905 - Updated storage target type naming in ngenea wizard
HUB2-915 - "view task" modal is now full width in the jobs page
HUB2-927 - Users list is sorted alphabetically in the groups tab
HUB2-951 - Server address is now optional when creating S3 ngenea targets
HUB2-1025 - Tooltip for long values in the details sidebar
HUB2-1030 - Validation notice for Samba shares in the Space wizard
HUB2-1070 - Don't allow setting server names longer than NETBIOS allowed limit
HUB2-1124 - Updated labels and icons for default workflows

Bug

HUB2-595 - Create space parent directory if it doesn't already exist
HUB2-783 - Set owner for automated settings tasks to 'system'
HUB2-796 - Exclude certain interface types from site NAS settings
HUB2-805 - Filters not working on Users or Groups pages
HUB2-816 - Workflow file list selected whole space instead of individual files
HUB2-832 - Validation for space placement pools
HUB2-854 - Folder metadata not loading in the file browser
HUB2-856 - Fix settings page link for when a custom base URL is configured
HUB2-931 - None values being written to samba shares
HUB2-932 - Multiple delete jobs being triggered for the same share
HUB2-933 - Don't force changing site shortcode on create site wizard
HUB2-935 - Broken spaces page when only one site is configured
HUB2-949 - Go back button for "advanced configuration" breaks UI on ngenea target wizard summary
HUB2-953 - Add api_secure_verify support to ngclient

HUB2-1028 - Site settings weren't updated in the UI after applying a change

Release notes - Ngenea Hub - 2.0.1: 2023-09-19
=====

Bug

HUB2-931: Don't write None values to samba shares

Documentation

HUB2-673: Updated user docs for Ngenea Hub 2.0

Release notes - Ngenea Hub - 2.0.0: 2023-08-11
=====

Feature

Initial Release:

- Hub 2.0 UI
- Provide the ability to view pixstor sites and storage
- Provide the ability to configure pixstor sites from the Hub UI
- Provide the ability to create Spaces
- Provide the ability to view files globally across all Space associated sites
- Provide the ability to perform data workflows
- Provide the ability to monitor jobs and tasks
- Provide the ability to manage Hub users and groups
- Support for Active Directory/LDAP logon
- Worker support of Hub 2.0 functions
 - Analytics data
 - Get and Set pixstor settings for a site
 - Autojoin to Hub

License

Ngenea Hub is licensed under the ArcaPix EULA: <https://www.arcapix.com/licenses/EULA.txt>

ArcaPix EULA
January 2021
<http://www.arcapix.com/licenses/EULA.txt>
Copyright 2021 ArcaPix Limited

This end user licence agreement (EULA) is a legal agreement between you (the entity or individual who is using the software) and us (as applicable either Pixit Media Limited, company number 07298805 or

ArcaStream Ltd, company number 08346283) in respect of (as applicable):

- * the computer software described in the order form, proof of entitlement (POE), invoice or other document linking to this EULA (in each case as issued by or agreed in writing with us), or which otherwise incorporates or is governed by this EULA and the data supplied with that software (collectively, Software);

- * the application programming interface described in the order form, proof of entitlement (POE), invoice or other document linking to this EULA (in each case as issued by or agreed in writing with us), or which otherwise incorporates or is governed by this EULA and the data supplied with that application programming interface, in each case whether provided on a standalone basis or alongside the Software (collectively, API) and

- * printed materials and electronic documents associated with that Software or API (Documents).

The Software, API and Documents are collectively the "Work" and such term includes each of the Software, API and Documents as applicable.

We license use of the Work to you on the basis of this EULA. We do not sell the Work to you. We remain the owners of the Work at all times. By using the Work, or otherwise by clicking "accept" or otherwise indicating acceptance of this EULA, you confirm you accept the terms of this EULA. If you do not accept this EULA you may not use the Work.

Any services we provide, including but not limited to maintenance, support, development and hosting, will be under separate terms of business, but any software we provide to you incidentally in the course of those services, will also be governed by this EULA unless expressly stated that other licence terms will apply.

You should print a copy of this EULA for future reference.

1 Grant and scope of licence

1.1 In consideration of payment by you of the agreed licence fee and you agreeing to abide by the terms of this EULA, we grant to you a non-exclusive, non-transferable licence to use the Work on the terms of this EULA for the duration of your subscription. Your subscription will only be valid during the period for which you have a valid POE from us to use the Software and when your subscription expires this EULA will automatically terminate without the need for notice. Any termination of this EULA will also terminate your subscription and your POE will be invalidated.

1.2 You may download, install and use the Software for your own internal business purposes only on the systems, either physical or virtual detailed in the accompanying POE as identified in your purchase order (if applicable) or otherwise approved by us or our

authorised representatives.

1.3 You may not use the Work for the purposes of making its functionality available to third parties as a service, whether directly or indirectly, without our express written agreement.

1.4 You may access and use the API solely for the purposes of

1.4.1 internally developing applications which communicate and interoperate with software or systems detailed in (and for the purposes detailed in) the accompanying POE as identified in your purchase order (if applicable) or otherwise approved by us or our authorised representatives; and

1.4.2 making calls to the systems or software permitted under clause 1.4.1, subject to any limits detailed in the accompanying POE as identified in your purchase order (if applicable) or otherwise agreed with us or our authorised representatives.

1.5 Your display and use information received through the API or data derived from that information is in each case subject to any limits detailed in the accompanying POE as identified in your purchase order (if applicable) or otherwise agreed with us or our authorised representatives).

1.6 You may use any Documents in support of the use permitted under condition 1.2 and make copies of the Documents as are reasonably necessary for their lawful use.

1.7 This EULA does not grant you permission to use the trade names, trademarks, service marks, or product names of us or our contributors or licensors, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of any "licence" files.

2 Restrictions

2.1 Except as expressly set out in this EULA or as permitted by any local law, you undertake:

2.1.1 not to copy the Work except where such copying is incidental to normal use of the Software, or where it is strictly necessary for the purpose of back-up or operational security;

2.1.2 not to rent, lease, sub-license, loan, translate, merge, adapt, vary or modify the Work;

2.1.3 not to make alterations to, or modifications of, the whole or any part of the Work, nor permit the Work or any part of them to be combined with, or become incorporated in, any other programs or other documentation as applicable, other than as expressly permitted in writing by us;

2.1.4 not to disassemble, decompile, reverse-engineer or create

derivative works based on the whole or any part of the Software or API (except as expressly permitted by us in writing or clearly provided for within the functionality of the Software or any accompanying API we provide) nor attempt to do any such thing except to the extent that (by virtue of section 296A of the Copyright, Designs and Patents Act 1988) such actions cannot be prohibited because they are essential for the purpose of achieving inter-operability of the Software or API with another software program, and provided that the information obtained by you during such activities:

2.1.4.1 is used only for the purpose of achieving inter-operability of the Software or API with another software program; and

2.1.4.2 is not unnecessarily disclosed or communicated without our prior written consent to any third party; and

2.1.4.3 is not used to create any software which is substantially similar to the Software or API;

2.1.5 to keep all copies of the Work secure and to maintain accurate and up-to-date records of the number and locations of all copies of the Work;

2.1.6 to supervise and control use of the Work and ensure that the Work are only used by your employees (or such other individuals or entities as you may be expressly permitted in writing by us to allow to access or use the Work) in accordance with the terms of this EULA;

2.1.7 to include our copyright notice on and any "licence" text files in all entire and partial copies you make of the Work on any medium, however you may not use any component parts of the Work outside of or separately from the Work;

2.1.8 not to provide or otherwise make available the Work in whole or in part (including but not limited to program listings, object and source program listings, object code and source code), in any form to any person other than your employees without prior written consent from us; and

2.1.9 to comply with all applicable technology control or export laws and regulations.

2.2 Without prejudice to the restrictions in this EULA on copying, modifying or creating derivative works from the Work, where you (or someone on your behalf) creates (solely or in conjunction with others, and whether in object or source code form) any software or other work which is based on or derived from the Work (Derivative Work) in breach of this EULA or otherwise, then in consideration of the sum of 1 GBP (receipt and sufficiency of which you acknowledge), you hereby:

2.2.1 assign to us (by way of present assignment of future rights) all intellectual property rights in such Derivative Work and waive (and shall procure a waiver of) all moral rights arising under the

Copyright, Designs and Patents Act 1988 in relation to the Derivative Work and, so far as is legally possible, any broadly equivalent rights that may exist in any territory of the world; and

2.2.2 In the event that any rights in such Derivative Work are not assigned to us pursuant to clause 2.2.1, you hereby grant to us an exclusive, royalty-free, worldwide, transferrable, irrevocable, perpetual licence (together with the right to grant sub-licences) to use in any manner as we determine, any such Derivative Work.

2.3 For the avoidance of doubt, for the purposes of clause 2.2, Derivative Work shall not include works which merely link or bind by name an existing third party application to the interfaces of the Software or API but does include works which are created to integrate with, or to be processed using, the interface of the Software or any API which we provide.

2.4 You agree not to (by your act or omission) do, or permit to be done, any act that will or may weaken, damage or be detrimental to the Work or any of our intellectual property rights or our or any of our contributors or licensors' rights in such, or seek to register any rights in the Work or any part of it or seek to commence litigation against any third party in respect of any intellectual property infringement in relation to the Work or any part of it.

3 Intellectual property rights

3.1 You acknowledge that all intellectual property rights in the Work anywhere in the world belong to us or our licensors or contributors, that rights in the Work are licensed (not sold) to you, and that you have no rights in, or to, the Work other than the right to use them in accordance with the terms of this EULA.

3.2 You acknowledge (unless explicitly agreed in writing by us) that you have no right to have access to the Software in source code form.

4 Liability

4.1 You acknowledge that the Work has not been developed to meet your individual requirements, including any particular cybersecurity requirements you might be subject to under law or otherwise, and that it is therefore your responsibility to ensure that the facilities and functions of the Software and API as described in the Documents meet your requirements.

4.2 We only supply the Work for internal use by your business, and you agree not to use the Work for any other purposes unless expressly permitted in writing by us.

4.3 We shall not in any circumstances whatever be liable to you, whether in contract, tort (including negligence), breach of statutory duty, or otherwise, arising under or in connection with this EULA for:

4.3.1 loss of profits, sales, business, or revenue;

4.3.2 business interruption;

4.3.3 loss of anticipated savings;

4.3.4 loss or corruption of data or information or any loss arising from misconfiguration or incorrect implementation or use of any API;

4.3.5 loss of business opportunity, goodwill or reputation;

where any of the losses set out in condition 4.3.1 to condition 4.3.5 are direct or indirect; or

4.3.6 any special, indirect or consequential loss, damage, charges or expenses.

4.4 Other than the losses set out in condition 4.3 (for which we are not liable), our maximum aggregate liability under or in connection with this EULA whether in contract, tort (including negligence) or otherwise, shall in all circumstances not exceed a sum equal to the Licence Fee paid in the 12 months prior to the event first giving rise to any liability. This maximum cap does not apply to condition 4.5.

4.5 Nothing in this EULA shall limit or exclude our liability for:

4.5.1 death or personal injury resulting from our negligence;

4.5.2 fraud or fraudulent misrepresentation;

4.5.3 any other liability that cannot be excluded or limited by English law.

4.6 Save as required by applicable law or agreed to in writing, we provide the Work on an "AS IS" basis, without conditions, warranties, representations or other terms of any kind, either express or implied (and any such implied conditions, warranties, representations or other terms, whether implied by statute, common law or otherwise, are excluded to the fullest extent permitted by law), including, without limitation, any conditions, warranties, representations or other terms relating to title, non-infringement, merchantability, or fitness for a particular purpose. You are solely responsible for determining the appropriateness of using the Work and for any configuration or interface necessary for you to effectively use the Work and assume any risks associated with your exercise of permissions under this EULA.

4.7 Without prejudice to clause 4.6, where the API interacts with any software or system which is not provided by us, we are not responsible and shall have no liability in any way for such software or system.

5 Termination

5.1 We may terminate this EULA immediately by written notice to you if you commit a breach of this EULA which you fail to remedy (if remediable) within 14 days after the service of written notice requiring you to do so. Without prejudice to our rights under this clause 5.1, your rights under this EULA will terminate automatically without the need for notice if you commit a material breach of any of the terms of this EULA.

5.2 On termination for any reason:

5.2.1 all rights granted to you under this EULA shall cease;

5.2.2 you must immediately cease all activities authorised by this EULA; and

5.2.3 you must immediately and permanently delete or remove the Work from all computer equipment in your possession, and immediately destroy or return to us (at our option) all copies of the Work then in your possession, custody or control and, in the case of destruction, certify to us that you have done so.

6 Communications between us

6.1 We may update the terms of this EULA at any time on notice to you in accordance with this condition 6. Your continued use of the Work following the deemed receipt and service of the notice under condition 6.3 shall constitute your acceptance to the terms of this EULA, as varied. If you do not wish to accept the terms of the EULA (as varied) you must immediately stop using and accessing the Work on the deemed receipt and service of the notice.

6.2 If we have to contact you, we will do so by email or by pre-paid post to the address you provided in accordance with your order for or registration of the Work.

6.3 Note that any notice:

6.3.1 given by us to you will be deemed received and properly served 24 hours after it is first posted on our website, 24 hours after an email is sent, or three days after the date of posting of any letter; and

6.3.2 given by you to us will be deemed received and properly served 24 hours after an email is sent, or three days after the date of posting of any letter.

6.4 In proving the service of any notice, it will be sufficient to prove, in the case of posting on our website, that the website was generally accessible to the public for a period of 24 hours after the first posting of the notice; in the case of a letter, that such letter was properly addressed, stamped and placed in the post to the address of the recipient given for these purposes; and, in the case of an email, that such email was sent to the email address of the

recipient given for these purposes.

7 Events outside our control

7.1 We will not be liable or responsible for any failure to perform, or delay in performance of, any of our obligations under this EULA that is caused by an Event Outside Our Control. An Event Outside Our Control is defined below in condition 7.2.

7.2 An Event Outside Our Control means any act or event beyond our reasonable control, including without limitation failure of public or private telecommunications networks.

7.3 If an Event Outside Our Control takes place that affects the performance of our obligations under this EULA:

7.3.1 our obligations under this EULA will be suspended and the time for performance of our obligations will be extended for the duration of the Event Outside Our Control; and

7.3.2 we will use our reasonable endeavours to find a solution by which our obligations under this EULA may be performed despite the Event Outside Our Control.

8 Third Party Software

8.1 Any part or component of the Software which has been contributed or created by any third party (including any open-source software) and which is not owned by us (Third Party Software) shall be deemed to be incorporated within the Software for the purposes of this EULA (except where expressly provided to the contrary) and use of the Third Party Software shall be subject to (and you shall comply with) such additional terms as relate to such Third Party Software from time to time (Third Party Additional Terms), and such Third Party Additional terms shall take precedence over this EULA in relation to such Third Party Software. You shall indemnify and hold us harmless against any loss or damage which we may suffer or incur as a result of your breach of any Third Party Additional Terms howsoever arising, and we may treat your breach of any Third Party Additional Terms as a material breach of this EULA.

8.2 For the avoidance of doubt, the performance of, and any issues caused by or arising from, any Third Party Software shall be considered an Event Outside Our Control and (without prejudice to the provisions of this EULA in relation to warranties regarding the Software generally) all Third Party Software is provided on an "AS IS" basis and without conditions, warranties, representations or other terms of any kind, either express or implied (and any such implied conditions, warranties, representations or other terms, whether implied by statute, common law or otherwise, are excluded to the fullest extent permitted by law), including, without limitation, any conditions, warranties, representations or other terms relating to title, non-infringement, merchantability, or fitness for a particular purpose.

9 Other important terms

9.1 We may transfer our rights and obligations under this EULA to another organisation, but this will not affect your rights or our obligations under this EULA.

9.2 You may only transfer your rights or your obligations under this EULA to another person if we agree in writing.

9.3 This EULA and any document expressly referred to in it constitutes the entire agreement between us and supersedes and extinguishes all previous agreements, promises, assurances, warranties, representations and understandings between us, whether written or oral, relating to its subject matter. You agree that you shall have no remedies in respect of any statement, representation, assurance or warranty (whether made innocently or negligently) that is not set out in this EULA or any document expressly referred to in it. You agree that you shall have no claim for innocent or negligent misrepresentation or negligent misstatement based on any statement in this EULA or any document expressly referred to in it.

9.4 If we fail to insist that you perform any of your obligations under this EULA, or if we do not enforce our rights against you, or if we delay in doing so, that will not mean that we have waived our rights against you and will not mean that you do not have to comply with those obligations. If we do waive a default by you, we will only do so in writing signed by us, and that will not mean that we will automatically waive any later default by you.

9.5 Each of the conditions of this EULA operates separately. If any court or competent authority decides that any of them are unlawful or unenforceable, the remaining conditions will remain in full force and effect.

9.6 This EULA, its subject matter and its formation (and any non-contractual disputes or claims) are governed by English law. We both irrevocably agree to the exclusive jurisdiction of the courts of England and Wales.

Contact

Ngenea Hub is provided and supported by:

Pixit Media: <https://pixitmedia.com/contact-us/>

Kalray: <https://www.kalrayinc.com/contact-us/>