

ORU Data Factory

User Guide

ORU Data Factory is managed by the AIQU scheduler, which can be accessed through the Horizon client. This guide is intended for researchers employed at Örebro University who would like to use ORU Data Factory through Horizon and AIQU. External partners will be able to use ORU Data Factory later. This guide will be updated when the system is ready for external users. For official AIQU documentation, please visit: docs.aiqu.ai

Note that ORU Data Factory is for running GPU-intensive workloads. If you only need CPU power, you need to look for another alternative.

1. Accessing Data Factory

ORU Data Factory can be accessed by authorized researchers both on campus (i.e., inside Örebro University's network) and off campus. To access and use ORU Data Factory a user must first request access by submitting a case description.

1.1. Case Description

ORU users can request time limited access to ORU Data Factory and user accounts in AIQU by submitting a case description form [[link to the web-form](#)]. Users must also download and install Microsoft Authenticator on their mobile phone, which will be used (together with ORU username and password) to login to AIQU.

1.2. Access

Users (once they have been granted an account in the AIQU) can access ORU Data Factory through AIQU using the Horizon web client (<https://horizon.oru.se>). The Horizon client authenticates a user through username (ORU email), password (ORU account password), and the Microsoft Authenticator app. On successful login, users will be able to see links to AIQU and FileZilla client (see Figure 1).

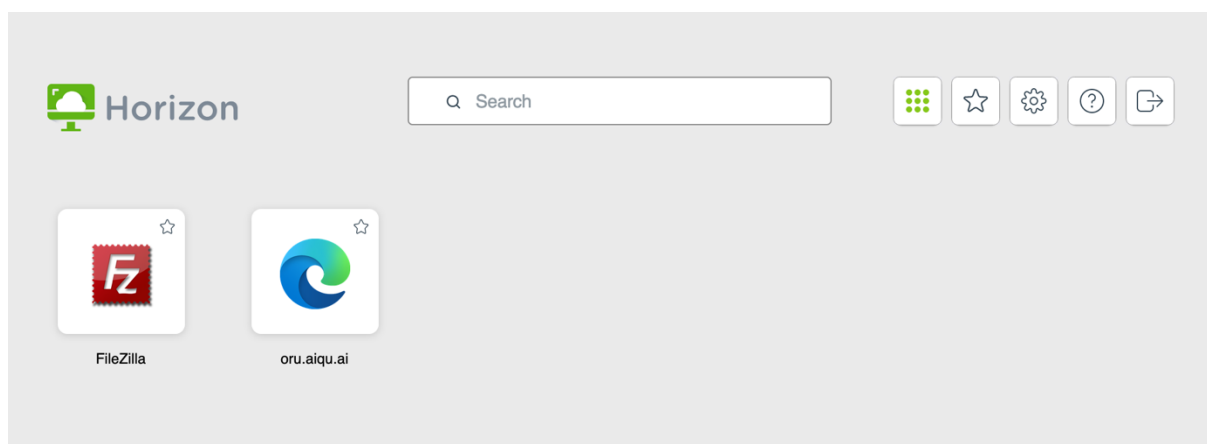


Figure 1. Horizon client

Here, in the Horizon client (Figure 1), users can click on oru.aiqu.ai to initiate an AIQU session, which can be used to add jobs to the job queue.

An alternative to the web client is to download and install the Horizon desktop client from VMWare:

https://customerconnect.vmware.com/en/downloads/info/slug/desktop_end_user_computing/vmware_horizon_clients/horizon_8?cd=1&hl=en&ct=clnk&gl=us

2. AIQU

Once the AIQU session is initiated, a user will be asked to login (see Figure 2). Users can use their ORU email and password to login to AIQU.

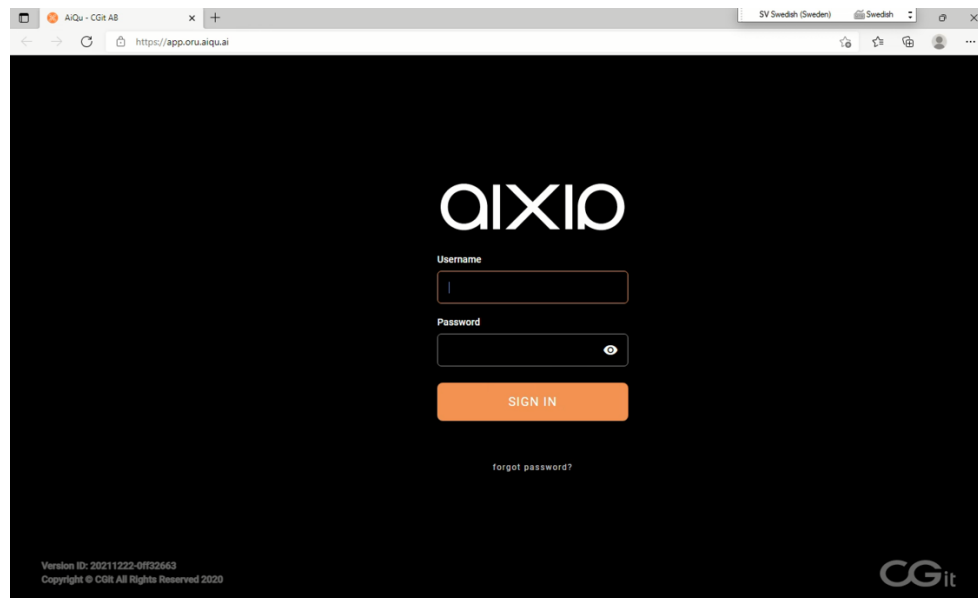


Figure 2. AIQU Login Screen

On Successful login, a user will see a screen somewhat like Figure 3.

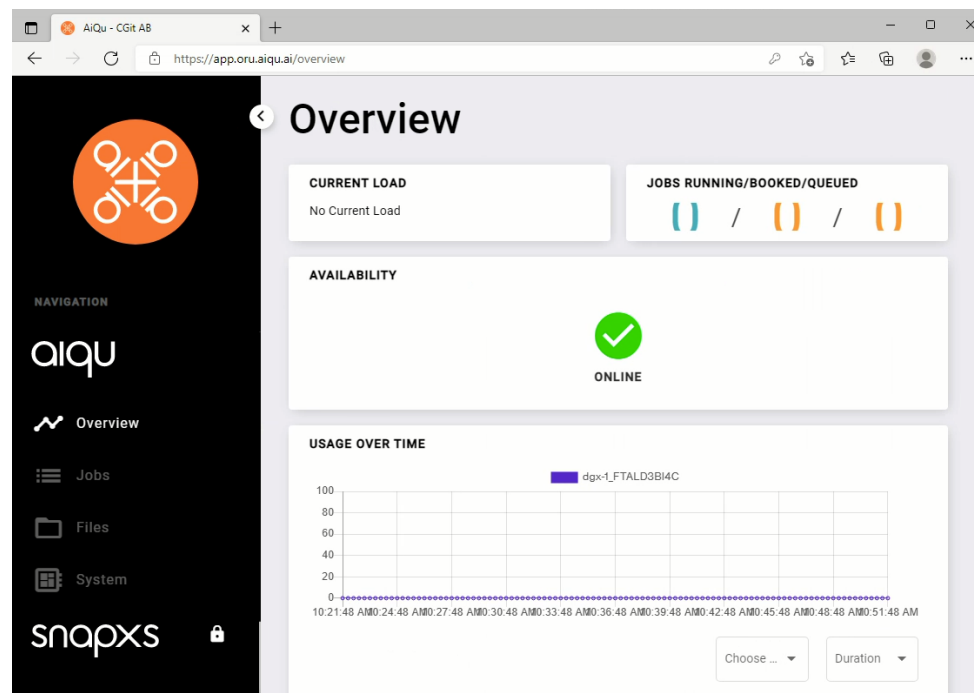


Figure 3. AIQU Main Screen

Users can schedule a job in the Data Factory through the **Jobs** link in AIQU (RHS Figure 3). Similarly, data can be transferred to/from ORU Data Factory using the **Files** link. These are further described in the subsections to follow.

2.1. Data Transfer

Due to legal requirements original data files **MUST** not be transferred to ORU Data Factory. Instead, a copy of the original data can be used in ORU Data Factory subject to the condition that the original data is stored somewhere else (i.e., not in the Data Factory).

Data can be transferred in and out of ORU Data Factory in two ways: a) Through AIQU, and b) through FileZilla.

Note! Since AIQU runs in a virtual Horizon environment, data transfer through AIQU can in some cases be problematic. For example, drag-and-drop may not work, and browsing files may not be able to access your local disc. In such cases, please try data transfer using FileZilla instead.

Data in AIQU is stored in storage buckets. There are two types of storage buckets, 1) User Storage Bucket, and 2) Project Storage Bucket. User storage buckets are personal and can only be accessed by the one owner. Project storage buckets are shared between many users. Currently working with project storage buckets is not smooth, since it is not possible to upload data to project storage buckets via FileZilla. In a future version of AIQU, it will be possible to transfer data from user storage buckets to project storage buckets. Until then it is recommended to use User storage buckets as far as possible. If you really need to use project storage buckets already now, one (not so smooth) way of getting your local data there is to first upload your data to a user storage bucket using FileZilla, then download each file to the Horizon file system in the AIQU file view, and then upload the data to a project storage bucket from the Horizon file system in the AIQU file view. If your original data is in ORU Research Drive, or any other file system available in the Horizon environment, none of this will be an issue, since you then can transfer your data to ORU Data Factory using the upload functionality in AIQU.

a) Data Transfer Using AIQU

Users can transfer files to the Data Factory using the **Files** link in AIQU. Figure 5 presents the screen for the Files menu. Here users can create folder and sub-folders as well as drag and drop files from their computers (these will be copied to the respective folder in the Data Factory).

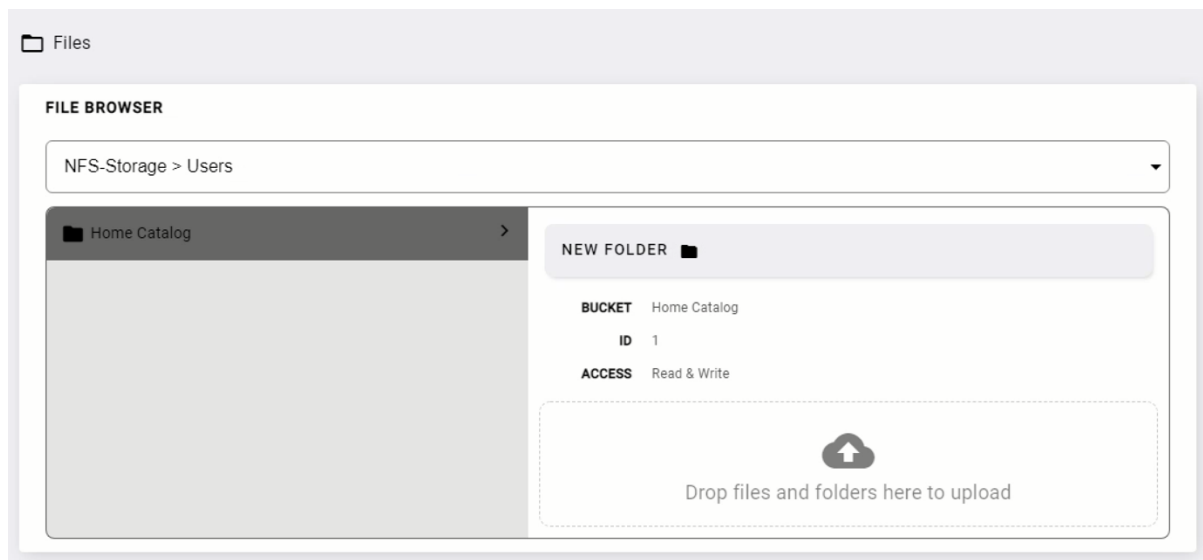


Figure 5. Files Menu AIQU

b) Data Transfer Using FileZilla

Another way to transfer data to/from DGX is through the FileZilla client. Users can invoke FileZilla from the Horizon (see, Figure 1) and by providing the following details:

- Host: 10.1.130.6
- Username: ORU username
- Password: ORU password
- Port: 22

It is possible to use FileZilla client from other systems. In this case, the host address should be: **10.1.115.65**. This option is currently only available on ORU-networks or via VPN.

2.2. Running Jobs

Jobs in AIQU are always run in a docker container. You can use public docker images from either Docker Hub (<https://hub.docker.com/>), Nvidia NGC (<https://catalog.ngc.nvidia.com/>), or ORU's registry (<https://docker-registry.oru.se>).

2.2.1 Running a script

To add a job to the queue in AIQU is quite straightforward. Label it with a proper name, select how many GPUs it can use, specify for how long the job should run (can be extended later), what docker image you want to use, which port(s) should be exposed by the running container, and optionally a command to run when the job starts (for example for starting a script, or executing an application). You can also mount a storage bucket to the job to be able to access data and link the job to a project.

Example of adding a job to the queue. This job will use one of Nvidia's public docker images with Tensorflow 2 and Python 3. It will use 4 GPUs and run for 240 minutes. The command "python train.py data/image" means a python script will be started and work on data from a storage bucket mounted on the container's path "/data". Port 8008 is exposed by the container so that we can view training progress on TensorBoard.

2.2.2 Running custom docker containers – use case Dockerhub

If you want to run custom docker containers, you need to push your docker image to Dockerhub and make it publicly available. Note that it is currently not possible to access private images from AIQU, therefore this is not an option for images that you need to keep secret. Pushing to Dockerhub requires an account. You can register for free at <https://hub.docker.com>.

Example of terminal commands for pushing an image to Dockerhub:

```
$ docker login
$ docker tag [TEST-NAME]:latest [DOCKERHUB-USER-ID]/[TEST-NAME]:latest
$ docker push [DOCKERHUB-USER-ID]/[TEST-NAME]:latest
$ docker logout
```

Now you can start your container as a job in AIQU by specifying its name in the Image-input field.

2.2.3. Interactive job – use case Jupyter Notebook

To run an interactive job in AIQU, such as Jupyter Notebook, one must use the right Docker image that includes the necessary script/program to be executed as an interactive job (in this case Jupyter notebook). You can add your interactive job as a normal job in AIQU WITH the right port number specified (I.e., the port that will be used by your container for the interactive process, which is jupyter in this example), see Figure 6.

Add job to queue

Job Label: Test

Number Of GPUs: 1 GPU

Time: 15

Image: nvidia/tensorflow:21.09-tf2-py3

Ports: 8888

Figure 6. Port for the interactive process

Once the job has started, you can connect to the Terminal as shown in Figure 7 by clicking on window icon under Actions or clicking the three dots.

Jobs

Queue & Job history

Queue & Job history	ID	Job Label	Image	Status	Time	Usage	Actions
<input type="checkbox"/>	113	Test	nvidia/tensorflow:21.09-tf2-py3	Running	0 H / 9 MIN	0%	

Figure 7. Connecting to Terminal

In the Terminal, you can run the interactive job command. In our example, it is the command to run Jupyter Notebook, see Figure 8.

```

root@7747e774d6c1:/workspace# jupyter notebook
[I 10:00:10.623 NotebookApp] Writing notebook server cookie secret to /root/.local/share/jupyter/runtime/notebook_cookie_secret
[I 10:00:11.317 NotebookApp] jupyter_tensorboard extension loaded.
[I 10:00:11.611 NotebookApp] JupyterLab extension loaded from /usr/local/lib/python3.8/dist-packages/jupyterlab
[I 10:00:11.612 NotebookApp] JupyterLab application directory is /usr/local/share/jupyter/lab
[I 10:00:11.615 NotebookApp] [JupyterText Server Extension] NotebookApp.contents_manager_class is (a subclass of) jupyter_text.TextFileContentsManager already - OK
[I 10:00:11.616 NotebookApp] Serving notebooks from local directory: /workspace
[I 10:00:11.616 NotebookApp] Jupyter Notebook 6.4.3 is running at:
[I 10:00:11.616 NotebookApp] http://hostname:8888/?token=9074b24ab6ed74806044d0bc01cf78b494091495e9ebbd0c
[I 10:00:11.616 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).

To access the notebook, open this file in a browser:
file:///root/.local/share/jupyter/runtime/nbsrvr-651-open.html
Or copy and paste this URL:
http://hostname:8888/?token=9074b24ab6ed74806044d0bc01cf78b494091495e9ebbd0c

```

Figure 8. Terminal command

Once the interactive process starts running in the terminal, we can go back to our job and click on it. Here, in the Actions, we choose “Open Public or Private” to open the port publicly or privately on the internet (Figure 9).

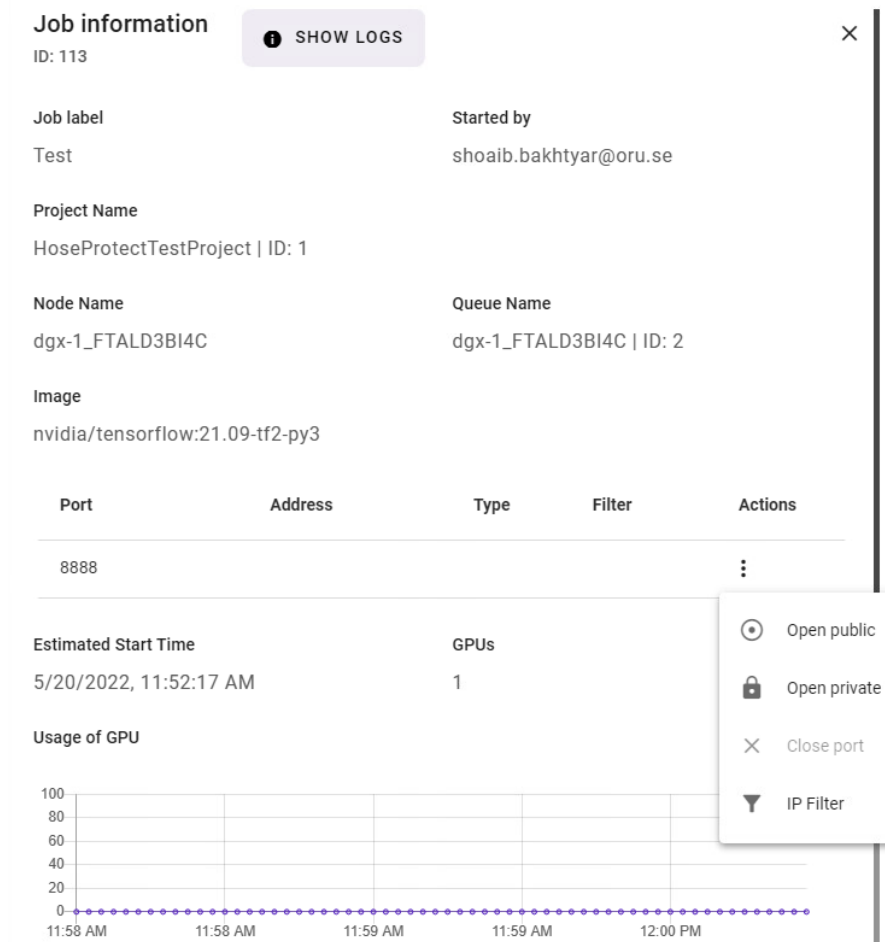


Figure 9. Ports for the interactive job

An address will be generated (Figure 10) that you can click on to access and run Jupyter Notebook in the browser. Note that you may have to copy and paste the generated token for the notebook from the Terminal for Jupyter Notebook to run properly.

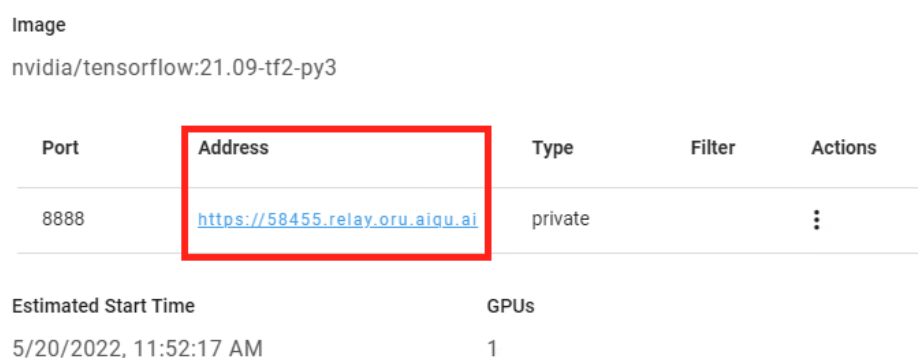


Figure 10. Address for accessing the interactive job (Jupyter)

2.3. Tokens

Running jobs and using storage in ORU Data Factory has a virtual cost of tokens in AIQU. All users and/or projects are given tokens when they are granted access to the service. In case you run out of tokens before your work is done, you need to contact your administrator so that your number of tokens can be increased.

2.4. Job Completion

Data (including both input and output data and any models) should be moved out of ORU Data Factory as soon as the job is completed. Any leftover data will be deleted within 10 days starting from the job completion date.