Issue Number: OUD3C-share-技術-004-01

The University of Osaka
D3 Center

User's Guide

NEC Corporation

Third Government Office System Development Department

table of contents

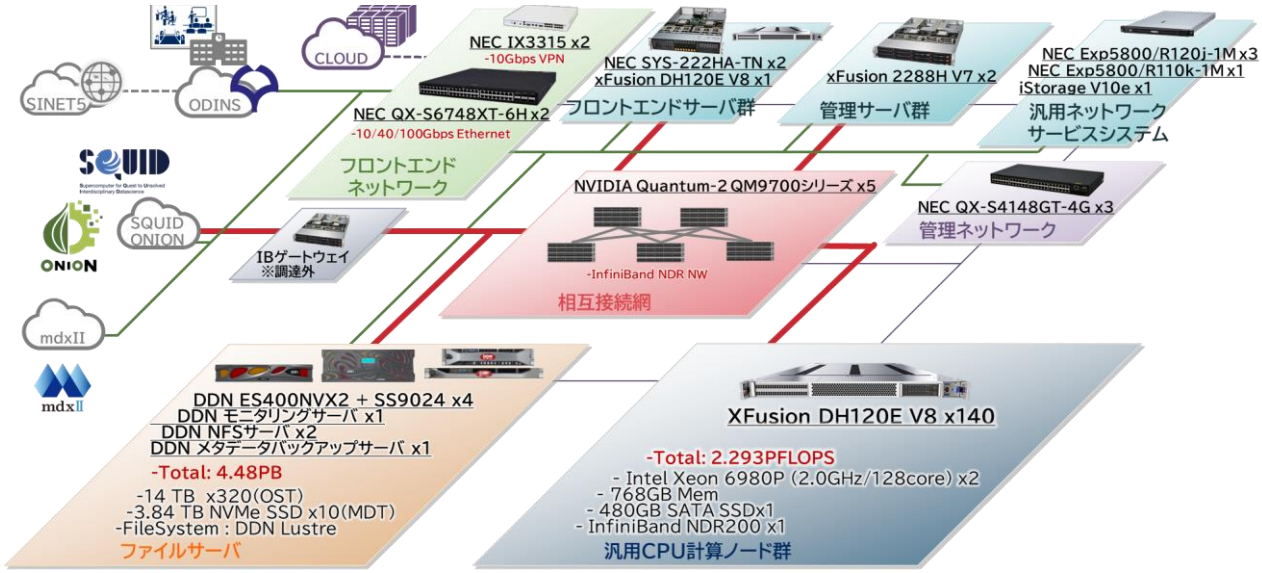
Revision history

Editions	date	Items	Overview	approval	inspection	creation
1.0	2025/08/29	All Items	Create a new one	NEC	NEC	NEC
2.0	2025/09/02	6.1 6.3.10	Fix how to run MotionCor3	Shiota	Shiota	Otsuka

1 System Overview

1.1 Overall System Diagram

OCTOPUS is a supercomputer system consisting of a general-purpose CPU computing node group and a file server, with a total theoretical computing performance of 16.591 PFLOPS.



Node Configuration	General Purpose CPU Compute Nodes 140 nodes (2.293 PFLOPS)	Computing Machine: Intel(R) Xeon(R) 6980P (2.0GHz/128core) x2 Memory Capacity: 768GB
Interconnect network	InfiniBand NDR200 (200 Gbps)	
File Server	DDN EXAScaler (HDD 4.48PB)	

1.2 Computing Environment

OCTOPUS provides a general-purpose CPU computing environment as a computing environment.

1.2.1 General Purpose CPU Computing Environment

The general-purpose CPU nodes in the general-purpose CPU computing environment consist of 140 units of the "DH120E V8".

DH120E V8

・ General Purpose CPU Compute Node

item		configuration
Total number of nodes		140 units
Server Configuration	processor	Intel(R) Xeon(R) 6980P (2.0GHz/128core) x2
	Memory Configuration	768 GB (32GB DDR5-6400 DIMM x24)
	hard disk	480GB SATA SSD x1
	interface	InfiniBand NDR200 x1, 1000BASE-T x2

・ Software Environment

item	configuration
OS	Rocky Linux 9.4 (64bit)
compiler	Intel oneAPI
MPI	Intel MPI

As a software environment, "Intel oneAPI", a C/C++/FORTRAN development environment that is highly optimized for Intel processors, is available. The "Intel MPI" shipped with the same environment is the standard MPI.

1.3 Storage area

The following storage environments are available for OCTOPUS:

Storage area	HDD Storage
feature	High-speed, high-capacity data storage area
capacity	home : 100GiB work: 5TiB+ additional purchase
File System	DDN ExaScaler (Lustre)
Effective Capacity	3.58PB
Disk Unit	14TB 7,200rpm ML-SAS-HDD
hardware	DDN ES400NVX2 + SS9024 x4

OCTOPUS offers a variety of data access methods for the purpose of a data-utilizing factory. The following are the references for each access method.

How to get there	protocol	use	References
High-speed parallel access	Lustre	Fast and parallel access in the system	3.1.1How to use the file system
CLI Access	SCP/SFTP	How to transfer data outside the system using the CLI	2.3 File Transfer
Browser access	HTTPS	How to transfer data outside the system using a web browser	7.1File transfer in a web browser
Inter-system access	NFS	How to exchange data with a peripheral system (SQUID)	7.2 File transfer from SQUID

1.4 Two types of front-ends

1.4.1 Features of each front-end

OCTOPUS offers two types of front-ends. The characteristics of each are as follows:

environment	HPC Front-end	Occupying Front-end
feature	Environment for HPC Application Development Visualization Environment Batch Job Entry Environment	Dedicated front-end environment for usage groups using virtual machines Batch Job Entry Environment
Number of nodes	2	(Increase or decrease depending on individual application)
application	Amazon DCV Server	
other	NVIDIA L4 x1	Separate application

1.4.2 HPC Front-end

The HPC front-end consists of two nodes of the "SYS-222HA-TN". It has an Intel(R) Xeon(R) 6980P CPU and can be used with a graphics accelerator for remote visualization processing, visualization applications, etc.

・ HPC Front-end

item		configuration
Total number of nodes		2 nodes
Server Configuration	processor	Intel(R) Xeon(R) 6980P (2.0GHz/128core) x2
	Memory Configuration	768 GB (32GB DDR5-6400 DIMM x24)
	hard disk	960GB SATA SSD x2
	interface	InfiniBand NDR200 x1, 10GBASE-T x2, 1000BASE-T x2, BMC x1
	GPU	NVIDIA L4 x1

・ Software Environment

item	configuration
OS	RHEL 9.4 (64bit)
Job Entry Environment	NEC NQSV
Remote Visualization Applications	Amazon DCV Server

As for the software environment, "NEC NQSV" is available as a job input environment. In addition, it is possible to remotely display the desktop screen on the HPC front-end and use the GPU on the HPC front-end to visualize and process it.

For information on how to use the job input environment, refer to "5 How to Run Programs" and "3.2 How to Use the Front-end for HPC" for how to use the remote visualization application.

1.4.3 Occupied Front-end

The occupied front-end consists of a single node. I have an Intel(R) Xeon(R) 6980P as a CPU. By applying for use separately, a front-end environment dedicated to the application group using a virtual machine is prepared.

・ Occupied Front-end (Physical)

item		configuration
Total number of nodes		1 node
Server Configuration	processor	Intel(R) Xeon(R) 6980P (2.0GHz/128core) x2
	Memory Configuration	1,152 GB (48GB DDR5-6400 DIMM x24)
	hard disk	960GB SATA SSD x2
	interface	InfiniBand NDR200 x1, 10GBASE-T x2, 1000BASE-T x2, BMC x1

・ Software Environment

item	configuration
OS	RHEL 9.4 (64bit)
Job Entry Environment	NEC NQSV

As for the software environment, "NEC NQSV" is available as a job input environment.
For information on how to use the job input environment, see "5 How to Run Programs".

1.5 System Usage Environment

1.5.1 How to log in

As part of the security enhancement, this system uses two-factor authentication when logging in with SSH. When logging in, you need to prepare the app required for two-factor authentication and pre-install it on your device. For specific login instructions, please refer to "Logging in to the 2 front-ends".

1.5.2 Program Development

This system allows you to use C/C++/FORTRAN languages, as well as compilers and libraries for various program development. You can also use a variety of pre-built applications.

The environment variable settings associated with the use of these applications are managed by the "Environment Module". By using the module command, you can uniformly set the environment variables required for the use of the application.

In addition, we have prepared a base environment that summarizes the environment variable settings necessary for basic use. By loading the base environment first, it is possible to easily prepare the minimum required usage environment.

The base environment prepared by this system is as follows.

classification	Base Environment Module Name	substance
Compiler + MPI Environment +	BaseCPU/2025	Recommended Environment for Program Development for General Purpose CPU Computing Environments

library	BaseGCC/2025	Program development environment when using GCC
Language environment + modules	BasePy/2025	Program Development Environment for the Python Language
	BaseR/2025	Program Development Environment for R Language
	BaseJDK/2025	Program Development Environment for JAVA Language
	BaseJulia/2025	Program Development Environment for Julia Language
Application Environment	BaseApp/2025	Base environment for ISV and OSS application users

For specific instructions on how to use the environment settings, please refer to "3.1.3 Environment Settings" and "4 Program Development".

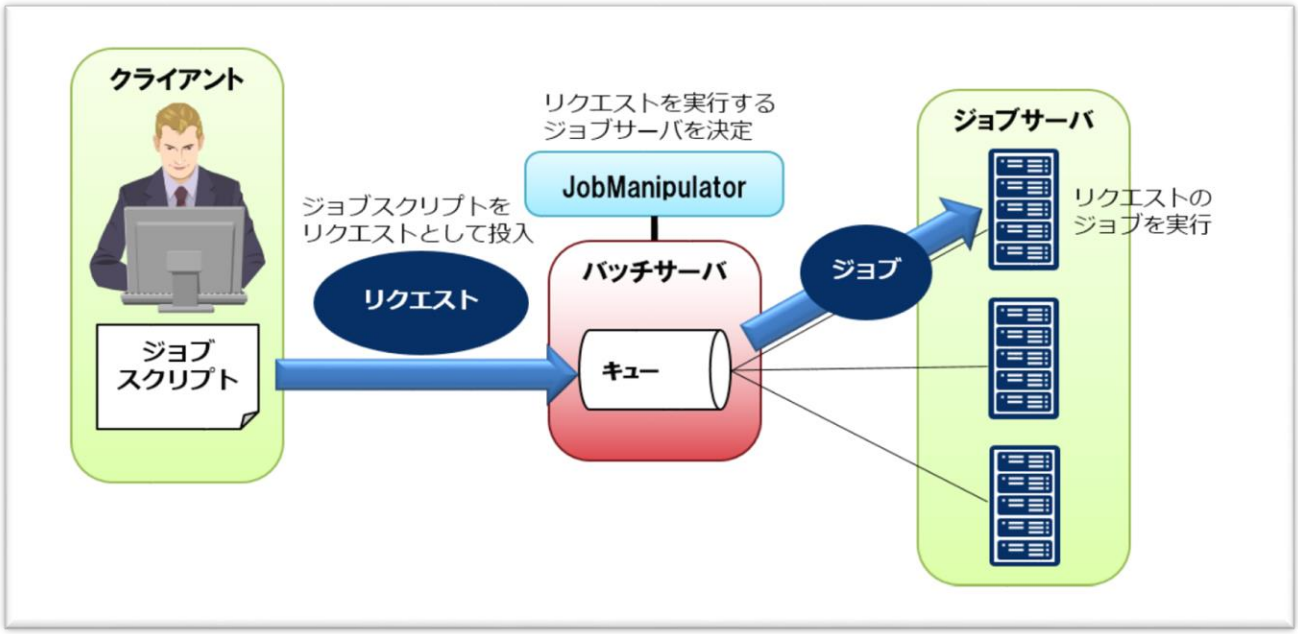
- Environment Modules official page
  http://modules.sourceforge.net/

1.5.3 Program execution

When executing a program, the computing environment is shared among many users. In order to prevent interference in program execution between users, the NEC NQSV job management system is equipped to control the order of execution and node control.

NEC NQSV enables batch and conversational use of computing resources. Request job execution (job input) from the front-end to the job management system. The job management system considers and judges the priority of other job requests, the amount of resources requested, and the usage status of OCTOPUS, and determines the execution order of the submitted job requests.

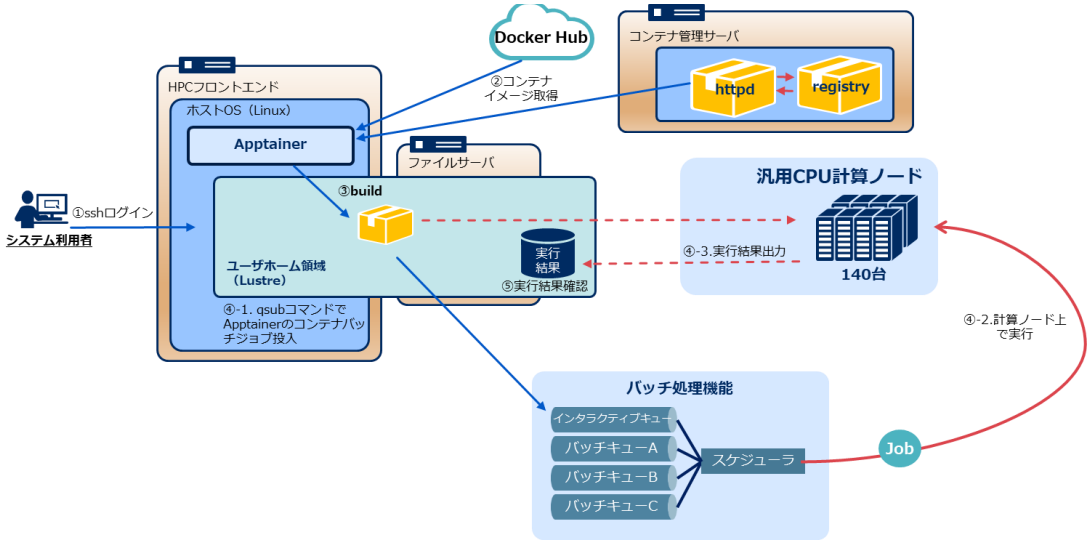
Since OCTOPUS is premised on sharing and use by many users, this batch processing method is adopted as the main program execution policy.



For specific instructions on how to use the job management system, please refer to "5 How to Run Programs".

1.5.4 Container Utilization

This systemallows the use of containers using A pptainer. It is possible to deploy container images published on the Internet or in the system to the home directory, customize them, and run them as jobs.



By using a suite of pre-configured software as containers, you can reduce the amount of time you need to prepare in advance, improve the repeatability of program execution, and use it without complex prior knowledge.

The specific use of containers is described in "4.4 How to use containers. For examples of running jobs using containers, see "How to Run Containers in 5.4".

Below is an overview of each step.

(1) Image acquisition

Apptainer containers typically use image files represented by the .sif extension. Image files can be downloaded from the Internet or published in the system, or files created by users can be transferred to the system. In this chapter, "4.3.1 Preparing the Container Image describes the following steps:

- Transfer from a local workstation to a system
- Retrieving an image from the OCTOPUS local registry
- Retrieving images from the Apptainer Library
- Get images from Docker Hub

Although containers published in each community do not necessarily work on this system, this book explains how to obtain images using various communities because it is useful for setting up the execution environment.

(2) Customize (build)

Customize the image file as needed to suit what you want to do. The process of reflecting the changes in the image file is called build. In this chapter, "4.4.2 Customizing and Building Container Images", we describe the following steps as customization methods that can be done within the system.

- How to customize via sandbox
- How to Describe Customizations in a Def File

The method of customization via the sandbox can be customized in the sense of a normal Unix shell, but users may be restricted because they do not have root privileges. If possible, it is more customizable to customize it on your own local workstation with root privileges and transfer the image file to the system.

(3) Job Request and Execution

Program execution in the system is based on execution by job request to the job management system. 「5.4 How to run a container" describes the procedure for executing a job using a container.

For more information on Apptainer, please refer to the man command or refer to the documentation available on the Internet.

- Apptainer Official Page
  https://apptainer.org/

- Apptainer 1.4 User Guide
  https://apptainer.org/docs/user/1.4/

2 Login to the front-end

2.1 How to log in with SSH

To access OCTOPUS, you need to access D3C-Sinet, a network of supercomputers connected to which the National Common Use Supercomputing System is connected. This system provides an HPC front-end and an occupied front-end to access OCTOPUS.

The connection information required for login and the hostname of the destination are as follows.

nape	server	Hostname
1	HPC Front-end	octopus.hpc.osaka-u.ac.jp
2	Occupied Front-end	voctopusXX.hpc.osaka-u.ac.jp

*If you want to use an occupied front-end, you need to apply separately. Once your request is approved, you'll be contacted to connect to.

For the above hostname, please access it using the following connection method:

How to connect	SSH
Authentication Methods	Two-step verification
OS Japanese Code	ja_JP. UTF-8

2.1.1 Installing a two-step verification app

In order to SSH log in to the front-end, you need to go through two-factor authentication using Google Authenticator, etc. Prepare the application required for two-factor authentication and install it on your device or smartphone.

Two-step verification applications that have been verified to connect are:

OS	application	remarks
Android	Google Authenticator	Google Play Store
iOS	Google Authenticator	Apple App Store
Windows	WinAuth	https://winauth.github.io/winauth/download.html
macOS	Step Two	Apple App Store

The following steps will use Google Authenticator for iOS as an example.

2.1.2 First Login

SSH access to the front-end using SSH commands or terminal software.

(1) First SSH Access

- **Login from UNIX OS (Linux, macOS)**

Use the ssh command.

An example of logging in to an HPC front-end (octopus.hpc.cmc.osaka-u.ac.jp) is as follows:

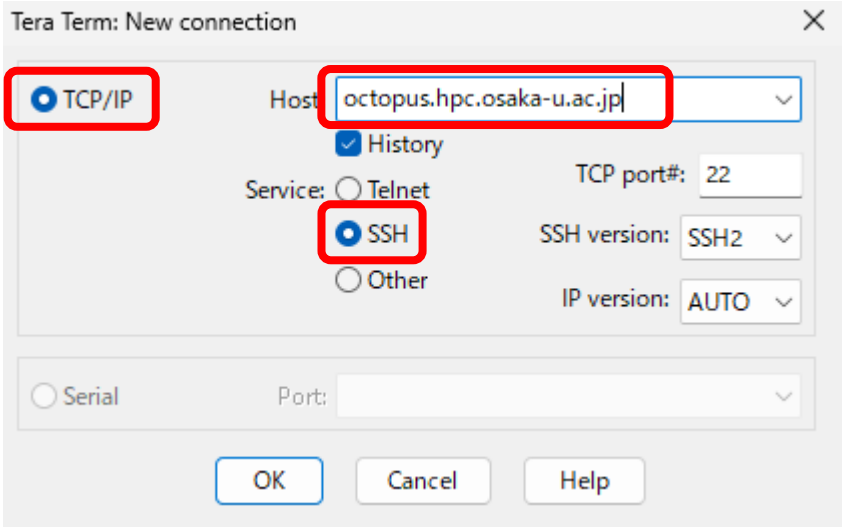
```
$ ssh (-l username) octopus.hpc.osaka-u.ac.jp
The authenticity of host 'octopus.hpc.osaka-u.ac.jp (133.1.66.X)' can't be established.
RSA key fingerprint is 32:fd:73:4e:7f:aa:5d:3c:2e:ab:37:83:d6:55:98:e2.
Are you sure you want to continue connecting (yes/no)? yes
                                    (Inquiries will only be made at the time of the first visit)
(Username)@octopus.hpc.osaka-u.ac.jp's password:
                                    (Enter the same password as the user management system.) )

* (-l username): Specified when the login username is different from the username on the local machine
```

- **Logging in from a Windows machine**

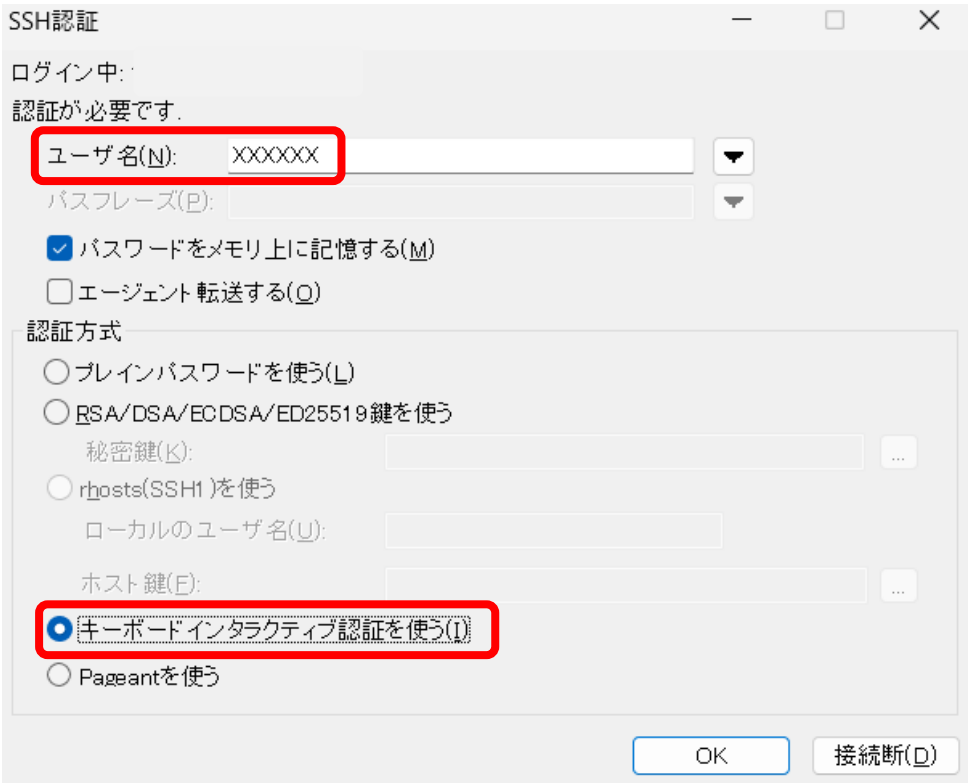
I will explain an example using the free software "Tera Term Pro".

Start "Tera Term Pro", and in the "Tera Term: New connection dialog",

- Select TCP/IP.
- Enter the host name (octopus.hpc.osaka-u.ac.jp in this case) in the Host field.
- Select SSH for the service.
- Click OK.



Enter your username on the SSH authentication screen and select "Use Keyboard Interactive Authentication" as the authentication method.



On the SSH authentication challenge screen, enter the same password as the user management system.

(2) Initial setup of two-step verification

When you log in for the first time, you will see the following in the terminal.

*If the terminal window is small, the QR code display will be broken. It is recommended to connect with the maximum window size.

```
Initiallize google-authenticator
Warning: pasting the following URL info your browser exposes the OTP secret to Google:
https://www.google.com/chart?chs=200*200&chld=M|0&cht=qr&otpauth://totp/user1@octop
us.hpc.osakau.ac.jp%3Fsecret%3DDXXXXXXXXCLI% 26issuer%3Doctopus.hpc.osaka-u.ac.jp
```

Your new secret key is: XXXXXXXXXXX

Enter code from app (-1 to skip): -1

Code confirmation skipped

Your emergency scratch codes are:

Use the QR code displayed on the screen or the secret key followed by "Your new secret key is:" to register for the 2-Step Verification application.

※ Depending on the window size of the terminal software, the QR code notation may be broken. Adjust the font size or use the URL and secret key provided.

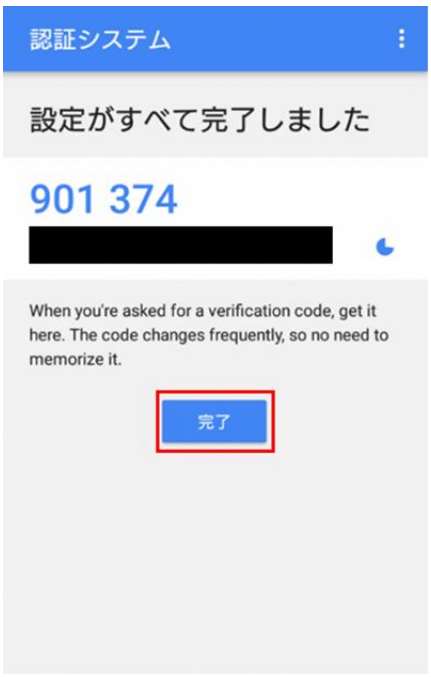
(3) Set up a two-step verification app

① Launch the Google Authenticator app and click the "Start" button.

② Select "Scan Barcode" or "Enter Provided Key" and enter the QR code or secret key displayed

in (2).

③ Once completed, the user is registered with Google Authenticator and issued a one-time password.



(4) End of first login

Go back to the terminal and type in for your question.

```
Your new secret key is: XXXXXXXXXXX
Enter code from app (-1 to skip): -1
Code confirmation skipped
Your emergency scratch codes are:
```

```
ok? Hit enter: (Enter)
```

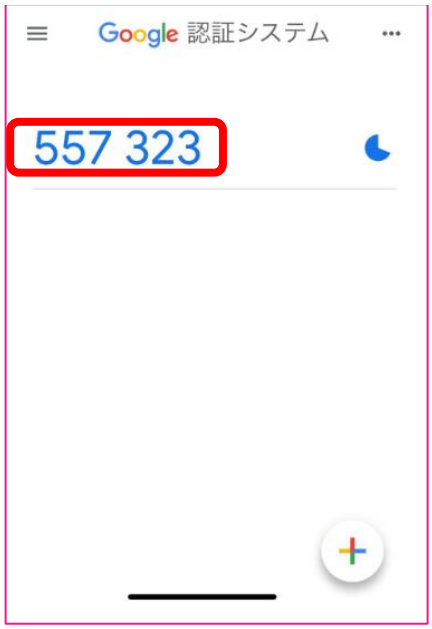
When prompted with **"Enter code from app (-1 to skip):"**, enter -1 to skip.
When "ok? Hit enter:" is displayed, press the Enter key. You will be logged out once.

*In "Your emergency scratch codes are:", you can issue only the number of codes specified in the system that can only be used once. However, OCTOPUS has set it to zero and is unusable from the perspective of security.

2.1.3 Second and subsequent logins

SSH login using the one-time password registered in the first login.

(1) Launch the 2-Step Verification app
Launch the two-step verification app to verify your one-time password.



*The one-time password is updated every 30 seconds. Since you will use the latest one-time password when logging in, keep the above screen displayed until you are logged in.

(2) SSH Access
SSH access is performed using terminal software or SSH commands. When logging in, use the one-time password obtained in (1).

- **Login from UNIX OS (Linux, macOS)**
  ssh command. An example of SSH login to a front-end (octopus.hpc.osaka-u.ac.jp) is as follows:

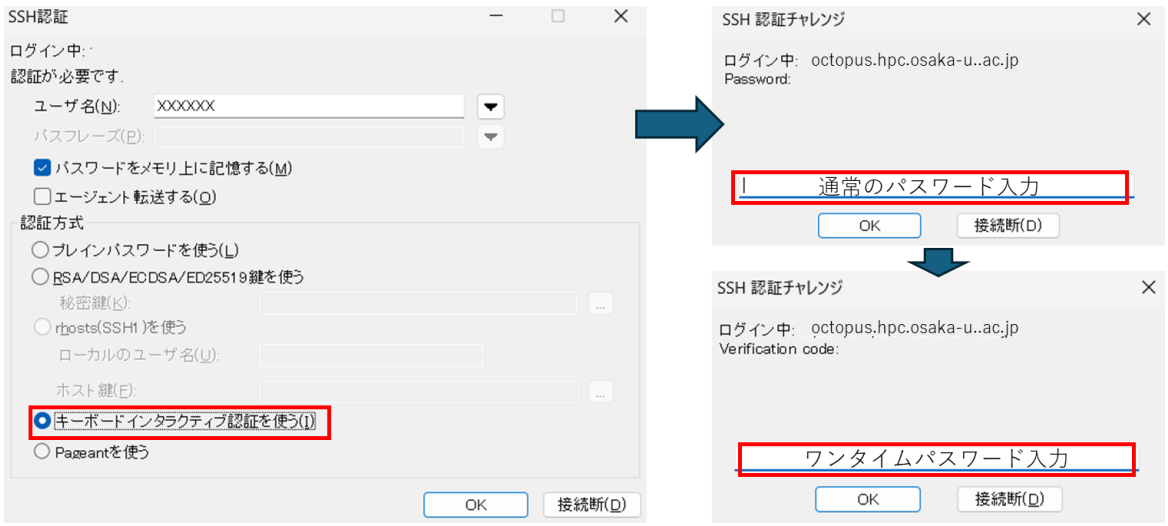
```
$ ssh (-l username) octopus.hpc.osaka-u.ac.jp
(Username)@octopus.hpc.osaka-u.ac.jp's password:
                    (Enter the same password as the user management system.) )
Verification code: *****
```

(Enter one-time password)

* (-l username): Specified when the login username is different from the username on the local machine

➢ **Login from Windows**

I will explain an example using the free software "Tera Term Pro". In the SSH authentication challenge part, enter your one-time password to log in.



2.1.4 How to recover if you can't authenticate 2-Step Verification

Administrator operation is required, so please contact your system administrator.

2.2 How to log in with OAuth

HPCI users can log in with OAuth authentication.

For information on how to log in, please refer to the user manual "HPCI Login Manual OAuth Compatible Version" published by the HPCI Operations Office.

➢ HPCI System Usage Information, Manuals, etc.
https://www.hpcioffice.jp/for_users/hpci_info_manuals
- HPCI Login Manual OAuth Compatible Version (HPCI-CA01-003-02 *As of August 2025)

The HPC front-end server (octopus.hpc.osaka-u.ac.jp) can log in with OAuth.

2.3 File Transfer

2.3.1 File Transfer on UNIX OS (Linux, macOS)

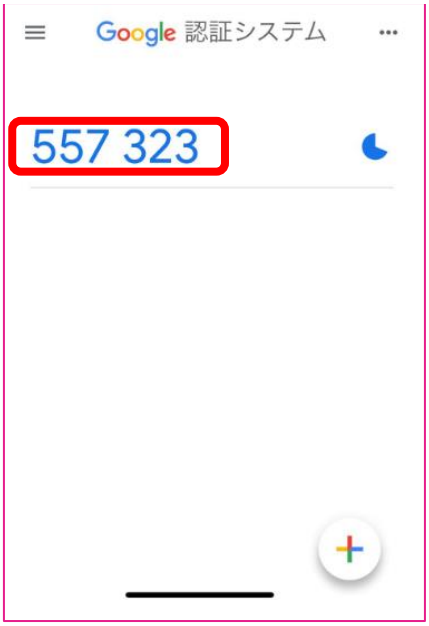
This section introduces an example of connecting to an FTP server from a UNIX-based OS (Linux, macOS) terminal such as a laboratory outside the D3 Center. It assumes that SFTP and SCP are installed

on the terminal side in advance. If you do not have SFTP or SCP installed, you can install it yourself or consult your terminal administrator.

*In order to transfer files, you need to set up two-factor authentication in advance. For details, please refer to the section "2.1.2 Initial Login".

(1) Launch the 2-Step Verification app

Launch the two-step verification app to verify your one-time password.



*The one-time password is updated every 30 seconds. Since you will use the latest one-time password when logging in, keep the above screen displayed until you are logged in.

➢ **Connection Instructions (for SFTP)**

Use the sftp command to connect to the FTP server. The following example is an example of connecting with the username a61234.

① Running SFTP Commands

Enter the following command:

```
$ sftp a61234@octopus.hpc.osaka-u.ac.jp
```


```
$ sftp a61234@octopus.hpc.osaka-u.ac.jp
```


② Password Entry

If this is your first time connecting to an FTP server, you will see a message like this, so enter yes.

```
The authenticity of host 'octopus.hpc.osaka-u.ac.jp' can't be established but
keys of different type are already known for this host.
RSA key fingerprint is 19:14:7f:28:54:39:16:9a:99:d0:db:93:d6:ff:f3:13.
Are you sure you want to continue connecting (yes/no)? yes
```

```
Warning: Permanently added 'XXXX.hpc.osaka-u.ac.jp,133.1.65.XX' (RSA)
to the list of known hosts.
```

You will be asked for a password, so you can enter the same password as the user management system to complete the connection to the FTP server.

```
a61234@octopus.hpc.osaka-u.ac.jp's password:    XXXXXXXX
a61234@octopus.hpc.osaka-u.ac.jp's password:    XXXXXXXX
(The password you entered will not be displayed)
```

③ One-time password entry

You will be asked for a Verification code, so enter the one-time password you got from the two-step verification app.

```
Verification code: ***** (enter one-time password)
```

④ FTP communication started

Just like regular FTP, file transfers are performed by get, put commands, etc.
The following example is an example of transferring the sample.f file to OCTOPUS' home.

```
sftp> put sample.f
Uploading sample.f to /octfs/home/a61234/sample.f
```

➢ **Connection Procedure (for SCP)**

If you use the scp command to transfer files, you must know the name of the file to be transferred and the directory name to which it is transferred.

① Executing SCP Commands

Enter the following command:

- When transferring the file sample.f from the current directory of the local terminal to the home directory of OCTOPUS.

```
$ scp sample.f a61234@octopus.hpc.osaka-u.ac.jp:
```

- When transferring the file sample.c in the abc directory under the OCTOPUS home directory to the current directory of the local terminal

```
% scp a61234@octopus.hpc.osaka-u.ac.jp:abc/sample.c sample2.c
```

※ SCP also allows you to transfer multiple files at once or recursively transfer files by directory. For more information, refer to the scp manual with the "man scp" command.

② Password Entry

When connecting to the SCP server for the first time, the following message will be

displayed. Enter yes to continue.

```
The authenticity of host 'octopus.hpc.osaka-u.ac.jp' can't be established but
keys of different type are already known for this host.
RSA key fingerprint is 19:14:7f:28:54:39:16:9a:99:d0:db:93:d6:ff:f3:13.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'XXXX.hpc.osaka-u.ac.jp,133.1.65.XX' (RSA)
to the list of known hosts.
```

You will be asked for a password, so enter the same password as the user management system.

```
a61234@octopus.hpc.osaka-u.ac.jp's password:     XXXXXXXX
(The password you entered will not be displayed)
```

③ One-time password entry

You will be asked for a Verification code, so enter the one-time password you got from the two-step verification app.

```
Verification code:   (Enter one-time password)
```

④ File Transfer

The file transfer will take place, and the transfer status, transfer file size, and transfer time will be displayed.

```
sample.f 100% |*****************************| 1326 00:01
```

2.3.2 File transfer in Windows

If you install file transfer software that supports SSH (Secure Shell) on your computer, you can connect to a file transfer server (hereinafter referred to as an FTP server). There are many applicable software, but here is an example of using the free software "WinSCP" to transfer files.

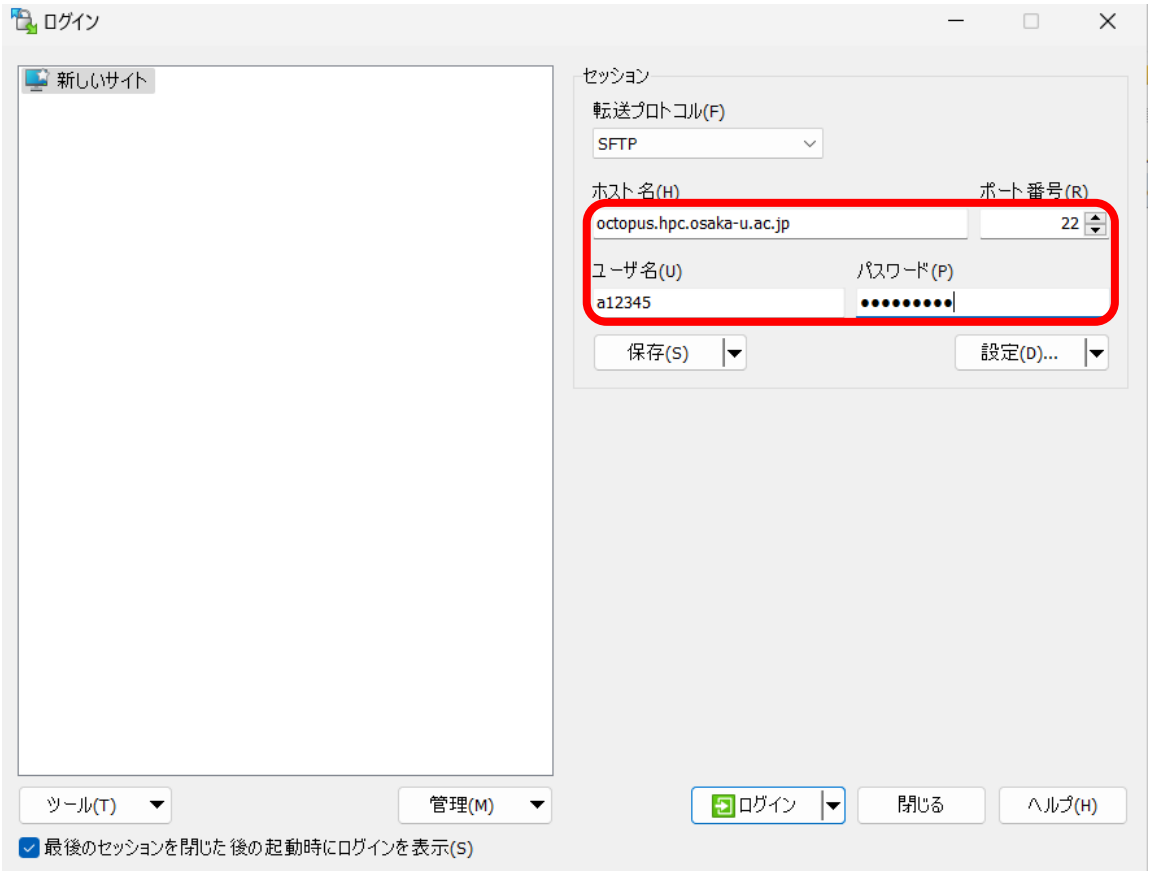
(1) Obtaining WinSCP

Download it from the official WinSCP website or other sources and follow the steps to install it.

(2) Launching WinSCP

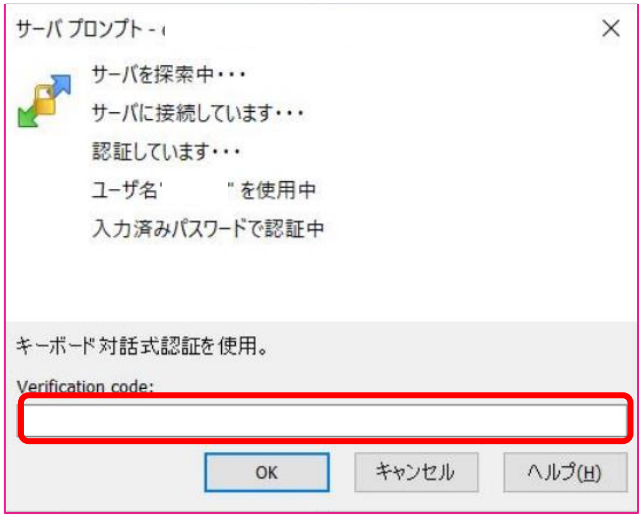
Start WinSCP and connect to the FTP server. After setting the following items, click the "Login" button.

```
Hostname: octopus.hpc.osaka-u.ac.jp
Port Number: 22
Username:    User ID
Password: User-managed password
```



(3) One-time password entry

You will be asked for a Verification code, so enter the one-time password you got from the two-step verification app.

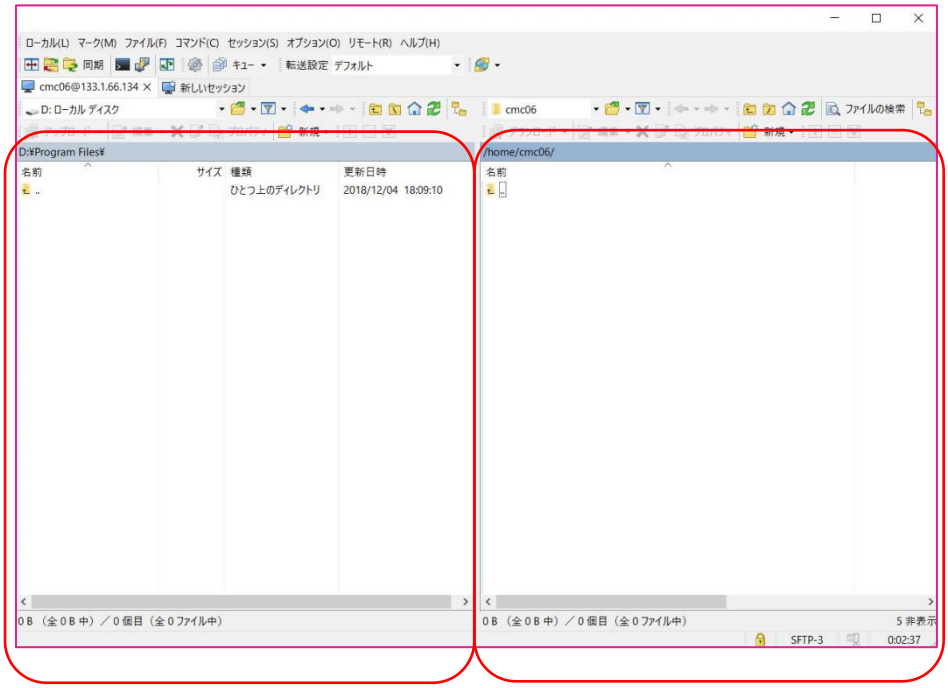


*When transferring text files such as program sources and NQSV job scripts, please set the transfer mode to "Text" in the "Settings" → "Preferences" items. After changing the settings, it is useful to "save" the session information.

(4) Start file transfer

After a successful login, the local (Windows) folder and the remote (OCTOPUS) folder will be

displayed on the left and right.



Local (Windows) side **Remote (OCTOPUS) side**

File transfer is possible by dragging and dropping the files to be transferred left and right.

3 Front-end environment

3.1 Common Matters

3.1.1 How to use the file system

From the front-end environment, the HDD area is directly available as a file system. The allocated disk space and its quota limits are as follows:

storage	File System	Domain Name	path	quota		remarks
				size	file	
HDD	EXAScaler (Lustre)	Home Area	/octfs/home/ (user number)	100GiB	---	Include browser access area
		Extended Areas	/octfs/work/ (Group Name) / (User Number)	5TiB (+)	Soft: 300 million Hard: 500 million	Additional purchases available

- **Home Area**

  This is the basic area given when registering as a user. The initial capacity is allocated 100 GiB.

  The file path (home) is as follows:

  ```
  /octfs/home/ (user number)
  ```

  For example: for user ID "user001" /octfs/home/user001

- **Extended Areas**

  This is an area where high-speed and high-capacity I/O is possible. The initial capacity is allocated 5 TiB. You can increase the capacity for each group by purchasing additional items.

  The file path (work) is as follows.

  ```
  /octfs/work/ (group name)/(user number)
  ```

  For example: user ID "user001" and group name "G012345" /octfs/work/G012345/user001

3.1.2 How to check your usage status

A usage_view is available as a command to check the usage status of the computer and file system from the front-end environment.

When you run usage_view command, you will see something like this:

```
$ usage_view
```

The items displayed above are as follows.

[Group summary]

- OCTOPUS points

Displays the points used by the group so far, the points claimed, the remaining points available, and the usage rate.

- HDD(GiB)

Displays the disk usage, total available disk space, available remaining disk space, and utilization of the extended space used by the group.
*The combined value of the disk space is displayed for the home area and the extended area.

[Detail]

- OCTOPUS points

For each user, the points used by the user so far, the points requested, the remaining points available, and the usage rate are displayed.

- Home(GiB)

For each user, the disk usage of the home space used by the user, the total available disk usage, and the remaining available disk space are displayed.

- HDD(GiB)

For each user, the disk usage of the extended space used by the user is displayed.
*The combined value of the disk space is displayed for the home area and the extended area.

[Node-hours]

- Usage

Displays the node time used by the group to date, by node group.

- Available

Displays the value of the available remaining OCTOPUS points converted to the available remaining node time. This is the node time when all remaining OCTOPUS points are used on that node.

- Available(Bonus)

Displays the value of the remaining OCTOPUS bonus points available converted to the remaining available node hours. This is the node time when all remaining OCTOPUS bonus points are used on that node.

3.1.3 Preferences

In this system, the environment variable settings of compilers, libraries, and applications are managed by "Environment modules". By using the module command, it is possible to uniformly set the environment variables required for the use of the application. Here are some of the main ways to use it:

command	explanation
module avail	List available development environments/apps
module list	List of Loaded Modules
module switch [file1] [file2]	Swap modules (file1 → file2)
module load [file]	Loading Modules
module unload [file]	Unloading the module
module purge	Unloading all loaded modules

module show [file]	View module details

This system provides a base environment that summarizes the environment variable settings required for basic use. By loading the base environment first, you can easily prepare the minimum required usage environment.

The base environment prepared by this system is as follows.

classification	Module Name	substance
Compiler+ MPI Environment + library	BaseCPU/2025	Recommended Environment for Program Development for General Purpose CPU Computing Environments
	BaseGCC/2025	Program development environment when using GCC
Language environment + modules	BasePy/2025	Program Development Environment for the Python Language
	BaseR/2025	Program Development Environment for R Language
	BaseJDK/2025	Program Development Environment for JAVA Language
	BaseJulia/2025	Program Development Environment for Julia Language
Application Environment	BaseApp/2025	Base environment for ISV and OSS application users

The following is an example of executing the module command.

① View the list of modules that can be loaded

```
$ module avail
------------------------------------- /system/apps/env/Base ------------------------------ ---------
BaseApp/2025   BaseCPU/2025   BaseExtra/2025   BaseGCC/2025   BaseJDK/2025   BaseJulia/2025   BasePy/2025
BaseR/2025
```

② Loading Modules

```
$ module load BaseCPU/2025
```

③ Check the Loaded Modules

```
$ module list
Currently Loaded Modulefiles:
 1) tbb/latest            3) umf/latest         5) mpi/latest        7) dal/latest                  9)
mkl/latest    11) debugger/latest  13) inteloneAPI/2025.2.0
 2) compiler-rt/latest    4) compiler/latest    6) advisor/latest    8) intel_ipp_intel64/latest   10)
vtune/latest  12) dpl/latest       14) BaseCPU/2025
```

④ View the list of additional modules that can be loaded

```
$ module avail
------------- /system/apps/rhel9/cpu/InteloneAPI/InteloneAPI2025.2/2025.2.0/modulefiles -------------
advisor/2025.2   compiler-intel-llvm/2025.2.0   compiler/2025.2.0   debugger/2025.2.0          dnnl/3.8.1
dpl/2022.9               intel_ippcp_intel64/2025.2   mkl/2025.2   tbb/2022.2   vtune/2025.4
```

```
advisor/latest   compiler-intel-llvm/latest     compiler/latest     debugger/latest           dnnl/latest
dpl/latest                intel_ippcp_intel64/latest  mkl/latest   tbb/latest  vtune/latest
ccl/2021.16.0    compiler-rt/2025.2.0           dal/2025.6          dev-utilities/2025.2.0   dpct/2025.2.0
intel_ipp_intel64/2022.2  ishmem/1.3.0                 mpi/2021.16  umf/0.11.0
ccl/latest       compiler-rt/latest              dal/latest          dev-utilities/latest     dpct/latest
intel_ipp_intel64/latest  ishmem/latest               mpi/latest   umf/latest

-------------- /system/apps/env/cpu/lib/inteloneAPI2025.2 --------------
hdf5/1.14.6    netcdf-c/4.9.3    netcdf-cxx/4.3.1    netcdf-fortran/4.6.2    pnetcdf-c/1.14.0    pnetcdf-
cxx/1.14.0  pnetcdf-fortran/1.14.0

-------------- /system/apps/env/cpu/Compiler --------------
inteloneAPI/2025.2.0

------------------ /system/apps/env/Base --------------
BaseApp/2025   BaseCPU/2025   BaseExtra/2025   BaseGCC/2025   BaseJDK/2025   BaseJulia/2025   BasePy/2025
BaseR/2025
```

(5) Loading additional modules

```
$ module load hdf5/1.14.6
```

For more information on module commands, please refer to the online documentation (man) or the documentation on the official website.

- Modules v5.3.0 : Docs >> module
  https://modules.readthedocs.io/en/v5.3.0/module.html

3.2 How to use HPC front-end

3.2.1 Preparation

This section describes how to prepare for the use of Amazon DCV (Desktop Cloud Visiualization) on the HPC front-end. Get the "Amazon DCV Client" from the following URL and install it on your PC. It is available for Windows, Linux, and macOS.

https://www.nice-dcv.com/latest.html

3.2.2 Creating a virtual session for an Amazon DCV server

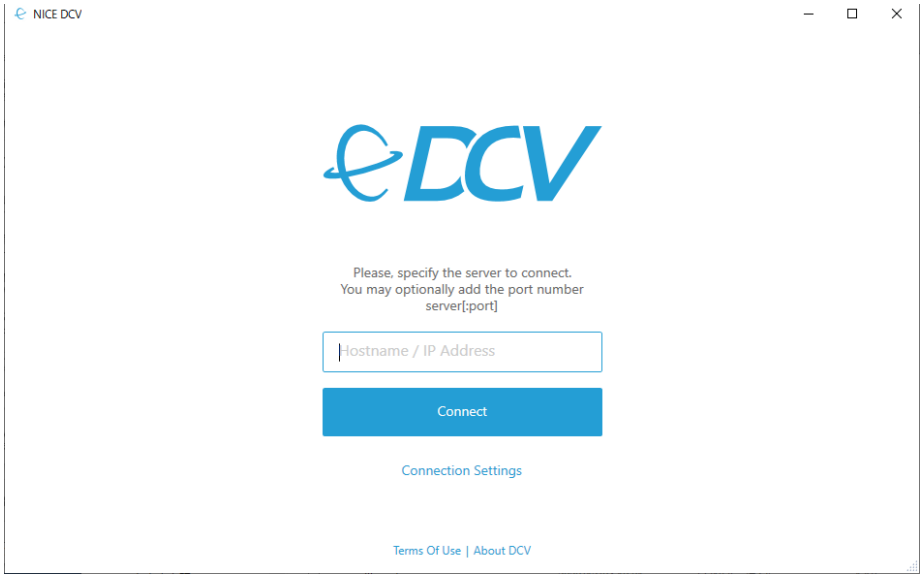
Log in to the HPC front-end and create a virtual session for the Amazon DCV server.

```
$ dcv create-session --type=virtual session name
(Create a virtual session)
$ dcv list-sessions
(Check the virtual session you created)
```

There are two HPC front-ends. You need to connect to the front-end server that created the virtual session. The server name and session name are "3.2.3. NICE DCV Client Launch", so please keep the information at hand.

3.2.3 Launching an Amazon DCV client

Launch DCVVIEWER. When you launch it, you will see a window like the one below.



「3.2.2. Connect to the server where you created the session in Create a Virtual Session on a NICE DCV Server. For example, if the server that created the virtual session was octopus1 and the session name was mysession, enter octopus1.hpc.osaka-u.ac.jp:5901/#mysession and click Connect.
When the connection to the Amazon DCV server on the HPC front-end is successful, the following screen is displayed. Enter your username, password, and click Login.

After successful authentication, you will see the HPC front-end desktop screen as shown below.

3.2.4 Deleting a virtual session on an Amazon DCV server

When you are no longer using Amazon DCV, log in to the HPC front-end and delete the virtual session on the Amazon DCV server.

```
$ dcv close-session session name
(Delete the virtual session)
```

You must connect to the server where you created the virtual session and delete it.

4 Program Development

4.1 Common Matters

In this system, from program creation to execution of creation modules, HPC front-end or occupied front-end is performed. The flow to the execution of the basic program is as follows. This section describes in detail the method of "(2) Generating (compiling) execution modules from source code".

① Create, edit, and modify source code
② Generating (compiling) execution modules from source code
③ Running the Execution Module

4.1.1 Using the Compiler MPI

Various programming languages are available for this system. The compiler, MPI, and accompanying library environment sets for using programming languages are grouped together as a base environment. This section provides an overview of the base environment for each programming language.

(1) C/C++ Language, FORTRAN Language

When developing programs in C/C++ and FORTRAN languages, compilers suitable for each computing environment of general-purpose CPUs can be used. It is also possible to create parallel execution modules with thread parallel and MPI parallel.

When using the module, follow "3.1.3. Preferences" and use "Environment Modules" to load the modules. Some modules have a recommended environment suitable for program development in each computing environment. The configuration of each recommended environment is as follows:

Computing Environment	Recommended Environment Module Names	compiler	MPI
General Purpose CPU Computing Environment	BaseCPU	Intel oneAPI	Intel MPI
(None)	BaseGCC	GNU Compiler	Open MPI

As an example of a specific procedure, if you use the recommended environment in a general-purpose CPU computing environment, load the module as follows.

```
$ module avail
------------------------------ /system/apps/env/Base ------------------------------
BaseApp/2025   BaseCPU/2025   BaseExtra/2025   BaseGCC/2025   BaseJDK/2025
BaseJulia/2025   BasePy/2025   BaseR/2025
```

```
$ module load BaseCPU/2025
```

As an example of a specific procedure, if you use the recommended environment in a general-purpose CPU computing environment, load the module as follows.

The recommended environment settings, each compilation command, and how to compile parallel programs are described in the following chapters.

- Compiler for general - purpose CPU computing environments.
  See 4.2 Compiling Programs for General Purpose CPU Computing Environments.
- GNU Compiler
  See 4.4 Compiling Programs for GPGPU Computing Environments.

(2) Other programming languages

A base environment is also available for programming languages other than C/C++ and FORTRAN. The available programming language environments are as follows:

Language environment	Module Name	version	explanation	remarks
Python3	BasePy	3.13.5	For Python Language	
R	BaseR	4.5.1	For R language	
JAVA	BaseJDK	21.0.8	For JAVA Languages	OpenJDK
Julia	BaseJulia	1.11.6	For Julia Languages	

Here's how to launch these program language environments.

A) Python

After logging in to the front-end, do the following:

```
$ module load BasePy
$ python3
```

B) R

After logging in to the front-end, do the following:

```
$ module load BaseR
$ R
```

C) JAVA

After logging in to the front-end, do the following:

```
$ module load BaseJDK
```

```
$ java --version
```

D) Julia

After logging in to the front-end, do the following:

```
$ module load BaseJulia
$ julia
```

Regarding the Python language, see "4.5 How to use Python" for more information.

4.1.2 Using the Compiler MPI

The libraries available in this system set environment variables using the module command.

In addition, modules in each library are versioned by <module name >/< version>. Libraries are included with each base environment. These libraries will be available after loading the base module.

The available libraries and language modules are as follows:

library Language Modules	Module Name			
	CPU	GCC	Py	R
Intel MKL (※)	○	-	-	-
GNU Scientific Library	-	○	-	-
HDF5	○	-	-	-
NetCDF	○	-	-	-
Parallel netcdf	○	-	-	-
Keras	-	-	○	-
PyTorch	-	-	○	-
TensorFlow	-	-	○	-
pbdR	-	-	-	○

* BLAS, LAPACK, and ScaLAPACK are included in Intel MKL.

For details on how to use the library, refer to "Using the Library" in "4.2 Compiling Programs for General Purpose CPU Computing Environments".

4.2 Compiling Programs for General-Purpose CPU Computing Environments

This section describes how to compile programs for a general-purpose CPU computing environment. The following instructions focus on compiling in the recommended environment, BaseCPU.

```
$ module load BaseCPU/2025
```

4.2.1 Serial execution

Compilation is done on the front-end.

The compilation commands for each programming language are as follows:

programming language	Intel oneAPI
FORTRAN	ifx
C	icx
C++	icpx

Below is an example compilation: The "-o" option allows you to specify the name of the executable file you want to create. If you don't specify it, the file name will be created with a.out.

First you need to load the environment variable settings module of the Intel compiler with the module command. When you load the BaseCPU module, it automatically loads the standard version of the Intel compiler.

```
$ module load BaseCPU/2025

$ ifx -o sample sample.f90
$ icx -o sample sample.c
$ icpx -o sample sample.cpp
```

4.2.2 Thread Parallel Execution

If you want to run in parallel with threads, specify the following options at compile time:

Parallel Models	Intel oneAPI
OpenMP	-qopenmp

The number of execution threads is defined at runtime by specifying the < number of execution threads> OMP_NUM_THREADS environment variables.

Below is an example compilation: First you need to load the environment variable settings module of the Intel compiler with the module command.

```
$ module load BaseCPU/2025

$ ifx -o sample_omp -qopenmp sample_omp.f90
```

4.2.3 MPI Parallel Execution

For MPI environments, Intel MPI is available.

The compile command is as follows:

programming language	For Intel MPI
FORTRAN	mpiifx
C	mpiicx
C++	mpiicpx

Below is an example compilation:

```
$ module load BaseCPU/2025

$ mpiifx -o sample_mpi sample_mpi.f90
$ mpiicx   -o sample_mpi sample_mpi.c
$ mpiicpx -o sample_mpi sample_mpi.cpp
```

4.2.4 Using the library

This section describes how to use the library in programs for general-purpose CPU computing environments.

(1) Intel Math Kernel Library(Intel MKL)

The Intel Math Kernel Library is a numerical algorithm library developed by Intel. A variety of libraries are available, including BLAS, LAPACK, ScaLAPACK, etc.

This system can be used in an environment with an Intel compiler that is loaded with a BaseCPU module. The following is an example of a link to a library in this system.

```
# Link MKL in Serial Execution
$ module load BaseCPU/2025
$ icx -qmkl=sequential sample.c                # dynamic link
$ icx -qmkl=sequential -static-intel sample.c  # static link

# Link MKL in thread parallel execution
$ module load BaseCPU/2025
$ icx -qmkl=parallel sample.c                  # dynamic link
$ icx -qmkl=parallel -static-intel sample.C    # Static Links

# Link MKL with MPI Parallel Execution
$ module load BaseCPU/2025
```

```
$ mpiicx -qmkl=cluster sample.c                  # dynamic link
$ mpiicx -qmkl=cluster -static-intel sample.c    # static link
```

For more information on how to use Intel MKL, please refer to the official online documentation.

- Intel Math Kernel Library - Documentation
  https://www.intel.com/content/www/us/en/developer/tools/oneapi/onemkl-documentation.html

(2) HDF5

HDF5 is a library for generating files that can be hierarchically structured and stored for large amounts of data. It is maintained for each computing environment. To take advantage of this library, give the option of -lhdf5 at compile time. Below is an example of a library link.

```
$ module load BaseCPU
$ module load hdf5
$ icx -o h5-sample h5-sample.c -lhdf5
```

(3) NetCDF

NetCDF is a library that allows you to use the widely used binary file format as a common data format for the purpose of storing scientific multi-dimensional data. This system is designed to be used in C, C++, and FORTRAN languages for each computing environment. To use the library, give an option such as -lnetcdf at compile time. Below is an example of a library link.

・C language

```
$ module load BaseCPU
$ module load netcdf-c
$ icx -o ncdf-sample ncdf-sample.c -lnetcdf
```

C++ language

```
$ module load BaseCPU
$ module load netcdf-cxx
$ icpx -o ncdf-sample ncdf-sample.cpp -lnetcdf_c++4 -lnetcdf
```

・Fortran language

```
$ module load BaseCPU
$ module load netcdf-fortran
$ ifx -o ncdf-sample ncdf-sample.f90 -lnetcdff -lnetcdf
```

(1) PnetCDF

PnetCDF is a library for simultaneous access of files in NetCDF format from parallel programs. This system is designed to be used in C, C++, and FORTRAN languages for each computing environment. To use the library, give an option such as -lpnetcdf at compile time. Below is an

example of a library link.

・C language

```
$ module load BaseCPU
$ module load pnetcdf-c
$ mpiicx -o pncdf-sample pncdf-sample.c -lpnetcdf
```

C++ language

```
$ module load BaseCPU
$ module load pnetcdf-cxx
$ mpiicpx -o pncdf-sample pncdf-sample.cpp -lpnetcdf
```

・Fortran language

```
$ module load BaseCPU
$ module load pnetcdf-fortran
$ mpiifx -o pncdf-sample pncdf-sample.f90 -lpnetcdf
```

4.3 Compiling with the GNU Compiler Collection

This system allows you to use the GNU Compiler Collection. Here, we will describe how to compile on BaseGCC, which is the base environment. When compiling the program, load the BaseGCC environment in advance.

```
$ module load BaseGCC/2025
```

4.3.1 Serial execution

Compilation is done on the front-end.

The compilation commands for each programming language are as follows:

programming language	GNU Compiler Collection
FORTRAN	gfortran
C	gcc
C++	g++

Below is an example compilation:

The "-o" option allows you to specify the name of the executable file you want to create. If you don't specify it, the file name will be created with a.out.

You must first load the GNU compiler's environment variable settings module with the module command. When you load the BaseGCC module, it automatically loads the standard version of the GNU compiler.

```
$ module load BaseGCC/2025

$ gfortran -o sample sample.f90
$ gcc -o sample sample.c
$ g++ -o sample sample.cpp
```

4.3.2 Thread Parallel Execution

If you want to run in parallel with threads, specify the following options at compile time:

Parallel Models	GNU Compiler Collection
OpenMP	-fopenmp

The number of execution threads is defined at runtime by specifying the < number of execution threads> OMP_NUM_THREADS environment variables.

Below is an example compilation: You must first load the GNU compiler's environment variable settings module with the module command.

```
$ module load BaseGCC/2025

$ gfortran -o sample_omp -fopenmp sample_omp.f90
```

4.4 How to use containers

With this system, it is possible to develop a program execution environment using containers using Apptainer (Singularity). This section describes how to prepare a container image and how to customize and build a container.
For an overview of how to use containers, refer to "1.5.4 Container Usage".

4.4.1 Preparing the container image

Container images can be obtained from local registries within OCTOPUS or from public registries on the Internet. You can also upload your own container images to OCTOPUS.

- **Transfer from a local workstation to a system**

  Container images prepared as image files can be transferred using scp commands, etc., just like regular files. For information on how to transfer files to this system, please refer to "2.3 How to Transfer Files".

- **Retrieving an image from the OCTOPUS local registry**

  Describes the steps to retrieve a container image that is published in OCTOPUS' local registry. For example, if you want to retrieve a container image called test registered in the local registry/master_image, run the following command:

```
$ apptainer build test.sif ¥
        oras://cntm:5000/master_image/test:1
```

The command format is as follows:

```
$ apptainer build <image filename> ¥
        oras://cntm:5000/< container image path >: <tagname>
```

After a successful retrieval of the container image, a container image (test.sif in the example above) is created in the current directory.

- **Get images from Docker Hub**

Walk through the steps to get a Docker container image from Docker Hub and save it as a container image for apptainer. For example, if you want to get a Rocky Linux container image from Docker Hub, run the following command:

```
$ apptainer build rocky9.sif docker://rockylinux:9
```

The command format is as follows:

```
$ apptainer build <image filename> docker://< container image path >:<tagname>
```

If the container image is successfully retrieved, a container image (rocky9.sif in the example above) is created in the current directory.

4.4.2 Customizing and building container images

Learn how to customize and build container images.

There are two ways to customize and build a container image within OCTOPUS: by customizing and building it via a sandbox, or by writing the customization in a def file.

- **How to customize via sandbox**

Learn how to customize and build via a sandbox.

(1) Creating a sandbox

First, create a sandbox from a container image. Navigate to the directory where you want to create a sandbox and run the command to create a sandbox. At this time, the group ownership and primary group of the directory you are moving must be the same, so the commands to be created are different in the home area and in the extended area. Also, if you add the -f (fakeroot) option, the user who created it may not be able to completely delete the sandbox. In this case, please follow the steps described in "(4) Delete a sandbox" to delete the sandbox.

As an example, if there is an image file called test.sif in the directory called mySandbox, here is an example of a command to create a sandbox called test.

・When creating a sandbox in the home area

```
$ cd ~/mySandbox
$ apptainer build -f --sandbox --fix-perms test test.sif
```

- When creating a sandbox in the expansion area

```
$ cd /octfs/work/<groupname>/<usernumber>/mySandbox
$ newgrp <group name>
$ apptainer build -f --sandbox --fix-perms test test.sif
```

The format of the apptainer command is as follows:

```
$ apptainer build -f --sandbox --fix-perms <sandbox name> < image filename>
```

(2) Launching a container

Next, launch the sandbox as a container.

```
$ apptainer run -f -w test
```

The command format is as follows:

```
$ apptainer run -f -w <sandbox name>
```

After successfully launching the container, you will see the Apptainer prompt shown below.

```
Apptainer>
```

From the above prompt, add packages using dnf, pip, etc. The commands you use to manipulate packages depend on the OS distribution you store in the container.
After you finish customizing the container image, use the following command to stop the container.

```
Apptainer> exit
```

(3) Image Filing

Finally, build the container image and generate the sif file.

```
$ apptainer build -f test.sif test
```

The command format is as follows:

```
$ apptainer build -f <sif filename> <sandbox name>
```

A successful build will create a sif file in the current directory.

(4) Deleting a sandbox

Normally, sandboxes can be deleted with the rm command.
As an example, if you want to delete a sandbox called test in a directory called mySandbox, run the following command:

```
$ cd ~/mySandbox
$ rm -fr test
```

The command format is as follows:

```
$ rm -fr <sandbox name>
```

However, depending on the container image you use, the user who created the sandbox may not be able to delete the sandbox completely. In this case, you can launch any container with privileges and delete the sandbox from within the container.

The command format is as follows:

```
$ apptainer exec -f --bind < path where the sandbox resides> < any container image path> ¥
 rm -r <sandbox name>
```

➢ **How to Describe Customizations in a Def File**

Def file to describe your customizations and learn how to customize them at build time.

(1) Creating a def file

First, create a def file that you will use for your build.

For example, if you want to customize a test registered in the local registry as a base container image, the def file will look like this:

```
1  Bootstrap: oras
2  From: cntm:5000/master_image/test:1
3
4  %files
5          ./test.conf /opt/test.conf
6          ./test_start.sh /opt/test_start.sh
7
8
9  %post
10         dnf install -y net-tools
11         chmod 755 /opt/test_start.sh
12
13 %runscript
           /opt/test_start.sh
```

- Overview of def files

(A) Bootstrap、From

Describe the type and location of the base image.

(B) %file

Describe the files you want to copy from the host OS to the container.

(C) %post

Include commands for customization.

(D) %runscript

Describe the actions to be automatically executed when the container starts.

*For more information on the def file, please refer to the Apptainer manual.
https://apptainer.org/docs/user/latest/definition_files.html

(2) Build an image

After the def file is created, build the container image and generate the sif file.

As an example, to build using a def file called test.def and create a test.sif, run the following command: First, create a def file that you will use for your build.

```
$ apptainer build -f test.sif test.def
```

The command format is as follows:

```
$ apptainer build -f <sif filename> <def filename>
```

A successful build will create a sif file in the current directory.

4.5 How to use Python

This system can use Python 3 as a Python language environment. When using the Python language, the base environment, BasePy, is loaded and used.

```
$ module load BasePy/2025
```

4.5.1 Interactive mode

For the Python language, you can use an interactive mode where you can enter the Python language on the CLI and interactively check the execution results. Examples of starting, running, and terminating interactive modes on this system are as follows.

```
# Loading Modules
$ module load BasePy

# Activating Interactive Mode
$ python3
Python 3.13.5 (main, Aug 13 2025, 18:27:42) [Clang 21.0.0git (icx 2025.2.0.20250605)] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>>

# Running the Python language
>>> print("hello python")          # Execution
```

```
hello python                    # execution result

# Exit Interaction Mode
>>> exit()
$
```

4.5.2 Program (script) execution

```
# Loading Modules
$ module load BasePy

# Program Execution
$ python3 sample.py
```

4.5.3 Adding Python Modules

In the Python language, a large number of Python modules are published in the PyPI (Python Package Index), and users can install additional ones in their own environments.

- PyPI
  https://pypi.org/

Additional module installations utilize the pip3 command. The main options for the pip3 command are:

command	explanation
pip3 list	Viewing the list of installed Python modules
pip3 install [pymod]	Installing Python Modules
pip3 show [pymod]	Python module details
pip3 uninstall [pymod]	Uninstalling Python Modules

In addition to the Python modules that are standard on this system, the following is an example of how to install additional modules by yourself. In the example, we have installed an additional dumper module. *The system will not install additional Python modules that have been developed as standard in this system, so please install the necessary modules by yourself.

```
# Loading Modules
$ module load BasePy

# Installing additional modules
$ pip3 install docker
```

4.5.4 Using the library

In the Python language, it is possible to use the TesorFlow, Keras, and Pytorch libraries. Each sample program is shown below

(1) TensorFlow(sample_tensorflow.py)

```
import tensorflow as tf
mnist = tf.keras.datasets.mnist

(x_train, y_train), (x_test, y_test) = mnist.load_data()
x_train, x_test = x_train / 255.0, x_test / 255.0
model = tf.keras.models.Sequential([
  tf.keras.layers.Flatten(input_shape=(28, 28)),
  tf.keras.layers.Dense(128, activation='relu'),
  tf.keras.layers.Dropout(0.2),
  tf.keras.layers.Dense(10)
])
predictions = model(x_train[:1]).numpy()
predictions

tf.nn.softmax(predictions).numpy()
loss_fn = tf.keras.losses.SparseCategoricalCrossentropy(from_logits=True)
loss_fn(y_train[:1], predictions).numpy()
# 2.835742

model.compile(optimizer='adam',
              loss=loss_fn,
              metrics=['accuracy'])
model.fit(x_train, y_train, epochs=5)
model.evaluate(x_test,  y_test, verbose=2)

probability_model = tf.keras.Sequential([
  model,
  tf.keras.layers.Softmax()
])
probability_model(x_test[:5])
```

(2) Keras(sample_keras.py)

```
import numpy as np
import tensorflow as tf
from tensorflow import keras
from tensorflow.keras import layers

# Model / data parameters
num_classes = 10
input_shape = (28, 28, 1)

# the data, split between train and test sets
(x_train, y_train), (x_test, y_test) = keras.datasets.mnist.load_data()

# Scale images to the [0, 1] range
x_train = x_train.astype("float32") / 255
x_test = x_test.astype("float32") / 255
# Make sure images have shape (28, 28, 1)
x_train = np.expand_dims(x_train, -1)
x_test = np.expand_dims(x_test, -1)
print("x_train shape:", x_train.shape)
print(x_train.shape[0], "train samples")
print(x_test.shape[0], "test samples")

# convert class vectors to binary class matrices
y_train = keras.utils.to_categorical(y_train, num_classes)
y_test = keras.utils.to_categorical(y_test, num_classes)

model = keras. Sequential(
    [
        keras. Input(shape=input_shape),
        layers. Conv2D(32, kernel_size=(3, 3), activation="relu"),
        layers. MaxPooling2D(pool_size=(2, 2)),
        layers. Conv2D(64, kernel_size=(3, 3), activation="relu"),
        layers. MaxPooling2D(pool_size=(2, 2)),
```

```
        layers. Flatten(),
        layers. Dropout(0.5),
        layers. Dense(num_classes, activation="softmax"),
    ]
)

model.summary()

batch_size = 128
epochs = 15

model.compile(loss="categorical_crossentropy",optimizer="adam",metrics=["accuracy"])
model.fit(x_train, y_train, batch_size=batch_size, epochs=epochs, validation_split=0.1)

score = model.evaluate(x_test, y_test, verbose=0)
print("Test loss:", score[0])
print("Test accuracy:", score[1])
```

(3) Pytorch(sample_pytorch.py)

```
import torch
import math

dtype = torch.float
device = torch.device("cpu")
# device = torch.device("cuda:0") # Uncomment this to run on GPU

# Create random input and output data
x = torch.linspace(-math.pi, math.pi, 2000, device=device, dtype=dtype)
y = torch.sin(x)

# Randomly initialize weights
a = torch.randn((), device=device, dtype=dtype)
b = torch.randn((), device=device, dtype=dtype)
c = torch.randn((), device=device, dtype=dtype)
```

```
d = torch.randn((), device=device, dtype=dtype)

learning_rate = 1e-6
for t in range(2000):
    # Forward pass: compute predicted y
    y_pred = a + b * x + c * x ** 2 + d * x ** 3

    # Compute and print loss
    loss = (y_pred - y).pow(2).sum().item()
    if t % 100 == 99:
        print(t, loss)

    # Backprop to compute gradients of a, b, c, d with respect to loss
    grad_y_pred = 2.0 * (y_pred - y)
    grad_a = grad_y_pred.sum()
    grad_b = (grad_y_pred * x).sum()
    grad_c = (grad_y_pred * x ** 2).sum()
    grad_d = (grad_y_pred * x ** 3).sum()

    # Update weights using gradient descent
    a -= learning_rate * grad_a
    b -= learning_rate * grad_b
    c -= learning_rate * grad_c
    d -= learning_rate * grad_d

print(f'Result: y = {a.item()} + {b.item()} x + {c.item()} x^2 + {d.item()} x^3')
```

Here's how to do it:

```
# Loading Modules
$ module load BasePy

# Running TensorFlow
$ python sample_tensorflow.py
```

```
# Keras Execution
$ python sample_keras.py

# Running Pytorch
$ python sample_pytorch.py
```

5 How to run a program

5.1 Common Matters

5.1.1 What is a job management system?

OCTOPUS enables batch and conversational use of computing resources using the NEC NQSV job management system. Request job execution (job input) from the front-end to the job management system. The job management system considers and determines the order in which the requested job requests are executed, taking into account the priority of other job requests, the amount of resources requested, and the usage status of OCTOPUS.
OCTOPUS adopts this batch processing method because it is assumed that it will be shared by a large number of users.

* NQSV "Requests" and "Jobs"

The following terms are used in the product manual as the concept of the NEC NQSV job management system.

- Request — The unit of batch usage requested by the user from the front-end.
- Jobs — Units of execution in a request that run on individual compute nodes

However, for the sake of understanding with other systems, the term "job" is used in this document to mean the unit of batch usage (equivalent to an NQSV request) requested by the front-end.

5.1.2 How to Submit Conversation Jobs

Conversational jobs are jobs that use compute nodes in a conversational (interactive) manner. If you want to use a conversational job, use the qlogin command.

A sample of the Enter Conversation Job command is provided below.

```
$ qlogin -q INT –group=G01234 [optional]
                ↑ Conversation queue name
↑ Group name to be billed
```

When you run the qlogin command, the request ID is numbered and displayed in the standard output as follows:

Request **1234.oct** submitted to queue: INT.
Waiting for **1234.oct** to start.

5.1.3 How to Load Batch Jobs

To submit a batch job from the front-end, perform the following steps:

① Creating a Job Script File
② Batch Job Requests to Job Management Systems

(1) Create a job script file

A job script file is a file that is required to describe a job request from a user. In this job script,

the user writes execution commands to perform calculations and describes the amount of resources required from the job management system as options (#PBS).

A sample job script (an example of serial execution in a general-purpose CPU computing environment) is listed below.

```
1   #!/bin/bash
2   #------- qsub option -----------
3   #PBS -q OCT
4   # ↑ Specifying the Queue Name to Submit a Batch Request to
5   #PBS --group=G01234
6   # ↑ Name of the group to be billed
7   #PBS -l elapstim_req=01:00:00
8   # ↑ Example of a job maximum execution time requirement value of 1 hour
9   #PBS -l cpunum_job=256
10  # ↑ Required number of CPU cores to be used
11  #PBS -m b
12  #↑ Send an email at the start of a batch request execution
13  #PBS -M user@xxx.jp
14  # ↑ Destination Address
15
16  #------- Program execution -----------
17  module load BaseCPU/2025
18  #↑ Load the base environment
19
20  cd $PBS_O_WORKDIR
21  # ↑ Go to the current directory at qsub runtime
22  ./a.out
23  # ↑ Executing the program
```

(2) Batch job request to the job management system

Use the command qsub provided by the scheduler as follows:

```
$ qsub [options] [job script filename]
```

When you run the qsub command, the request ID is numbered and displayed in the standard output as follows:

Request **1234.oct** submitted to queue: OCT.

The main options for the qsub command are:

option	function
-q [batch queue name]	Specify the queue to put the batch job in. (Required)
--group=[group name]	Run the job in the specified group. The budget of the specified group is consumed. (Required)
-l elapstim_req=[Elapsed Time Limit]	Specify the elapsed time limit value for the batch job.

	If not specified, the default values for each queue are applied. The format for specifying a time limit is as follows: hh:mm:ss hh time mm min ss seconds
-N [job name]	Specify a name for the job. If not specified, the batch script name becomes the job name.
-o [standard output filename]	Specifies the output file name for the standard output of the batch job. If not specified, it will be output to the directory when the job is submitted with the file name "job name.o request ID".
-e [standard error output file name]	Specifies the output file name for the standard error output for the batch job. If not specified, it will be printed as the file name "job name.e request id" in the directory when the request is submitted.
-j [o,e]	Merge the standard and standard error outputs of batch jobs. o: Outputs the merged result to the standard output. e: Prints the merged result to standard error output. (There must be a space between -j and o or e)
-M [email address]	Specify where you want to send the email. If you want to specify more than one, -M Email address 1,Email address 2 Please separate it with a ",".
-m [b,e,a]	Send an email about changes in the status of the batch job. b: Send an email when the job starts e: Send an email when the job ends A: Send an email when a job ends abnormally (There must be space between -m and b or e)

	You can specify more than one. e.g. email notifications at start and end -m be
-v environment variable	Specify the environment variables to use when running a batch job.
-r { y | n }	Specifies whether the batch job can be rerunn. Y: Rerun capable n: Liran not allowed
-b [Number of nodes]	Specify the number of nodes to run the job.
-T [MPI library name used]	Specification is required when performing MPI execution. If you are using an Intel compiler, Intel MPI is available. -T intmpi. If you want to use OpenMPI, specify -T openmpi.

5.1.4 About Job Management System Commands

You can check the status of the jobs you have entered with various commands in the job management system.

(1) Batch job confirmation

To check the status of a job that has been put in, use the qstat command.

- When displaying a list of jobs that users have entered

```
$ qstat
```

- When to view the list without omissions

```
$ qstat -l
```

The above is the lowercase letter L of the alphabet.
Width optimization is performed by specifying "--adjust-column" at the same time.

- When displaying the details of a specific job

```
$ qstat -f [request id]
```

Also, if you want to check the status of jobs submitted by the group to which the user belongs, use the qstatgroup command.

- When displaying a list of jobs submitted by a group to which the user belongs

```
$ qstatgroup
```

Use the sstat command to check the start time of the request you put in. If the execution start time is not determined, the time is not displayed.

```
$ sstat
```

During job execution, use the qcat command to view the contents of the standard output/standard error output. If you do not specify -e/-o, the job script is displayed.

```
$ qcat -e [request id] * For display of standard error output
$ qcat -o [request ID] *When displaying standard output
```

You can also combine the following options:

-f As the file content continues to grow, it outputs the added data.

-n Displays the specified number of rows. (If not specified, it is 10 lines.))

-b View from the beginning of the file. (If not specified, it will be displayed from the last line.))

When the job ends, it is no longer visible in the qstat command.

(2) Hold batch job

If you want to hold a job, use the qhold command. Holding will cause it to be out of the scheduling target and will not start execution.

```
$ qhold [request id]
```

(3) Unpending a Batch Job (Release)

Unhold (release) uses the qrls command. By releasing the hold, the job returns to its pre-hold state and is subject to scheduling again.

```
$ qrls [request id]
```

(4) Display of job information

If you want to view information about jobs that you have entered in the past, use the acstat command.

- When displaying information about jobs that the user has entered in the past (within the last 24 hours from the time the command is executed)

```
$ acstat
```

- When displaying information on jobs that the user has entered in the past (command execution year)

```
$ acstat -A
```

You can also use the acstatgroup command to view information about jobs that have been submitted in the past by the group to which the user belongs.

- When displaying information about jobs submitted in the past by the group to which the user belongs (within the last 24 hours from the time the command is executed)

```
$ acstatgroup
```

- When displaying information about jobs submitted in the past by the group to which the user belongs (command execution year)

```
$ acstatgroup -A
```

(5) Canceling a batch job

If you want to delete a request, use the qdel command.

```
$ qdel [request id]
```

If the batch job is in a running state (RUN), it sends a SIGTERM first, followed by a SIGKILL. You can specify the wait time before sending the SIGKILL by specifying the number of seconds of grace time in the -g option. If there is no designation, it is 5 seconds.

(6) Queue status check

If you want to view the state of the queue where the batch job is running, use the qstat command.

```
$ qstat -Q
```

5.2 Job Classes

This section describes the OCTOPUS job class. Each job class corresponds to a queue on the job management system, and users can use the computing environment by putting jobs into the queue.

The queue is divided into an input queue where users directly submit jobs, and an execution queue where users wait for the execution order.

utilization way	job class	Available Elapsed time	Available Maximum Number of Cores	Available memory	Number of concurrent nodes	remarks
all possession advantage use	OCT	120 hours	32,768core (256c/node)	95TiB (760GB/node)	128	Occupancy within the node
	OCT-H	120 hours	32,768core (256c/node)	95TiB (760GB/node)	128	Occupancy within the node ※1
	OCT-S	120 hours	128core	380GiB	1	※2
	DBG	10 points	512core (256c/node)	1,520GB (760GB/node)	2	for debugging
	INT	10 points	512core (256c/node)	1,520GB (760GB/node)	2	Interactive Use
	VA	120 hours	256core	760GB	1	pseudo-quantum annealing

Possession and Utilization	myOCT	unlimitedness	256core × Number of Nodes Occupied	760GB × Number of Nodes Occupied	Possession	Apply separately

*1 For those who want to spend more points and put in high-priority jobs to shorten the waiting time for execution.

*2 For those who want to allow sharing within nodes with other jobs to reduce point consumption.

5.3 How to use the general-purpose CPU computing environment

This section describes job scripts for general-purpose computing environments.

5.3.1 How to use serial execution

This is a sample job script that assumes program execution within a single node.

- queue : OCT
- group : G01234
- Elapsed time : 30 minutes (1 hour if not specified)

```
1   #!/bin/bash
2   #------- qsub option -----------
3   #PBS -q OCT
4   #PBS --group=G01234
5   #PBS -l elapstim_req=00:30:00
6
7   #------- Program execution -----------
8   module load BaseCPU/2025
9   module load xxx/xxx
10
11  cd $PBS_O_WORKDIR
12  ./a.out
```

- Line 9
  Describe what you had module load at the time of compilation.

This job script is only for executing the program, but it is also possible to specify additional options such as "send a notification to the email address when the job is finished" or "specify the name of the output result file".

5.3.2 How to use thread parallelization

This is a sample job script that assumes thread-parallel program execution within a single node.

- queue : OCT
- group : G01234

- Elapsed time : 30 minutes (1 hour if not specified)
- Total CPU Cores: 256 (128 cores× 2CPUs×1 node)

```
1   #!/bin/bash
2   #------- qsub option -----------
3   #PBS -q OCT
4   #PBS --group=G01234
5   #PBS -l elapstim_req=00:30:00
6   #PBS -v OMP_NUM_THREADS=256
7
8   #------- Program execution -----------
9
10  module load BaseCPU/2025
11  module load xxx/xxx
12
13  cd $PBS_O_WORKDIR
14  ./a.out
```

- Line 6
  Specify the number of parallel runs.

- Line 11
  Describe what you had module load at the time of compilation.

  Precautions when using
  Don't forget to specify "OMP_NUM_THREADS" in the execution script.
  If you do not specify "OMP_NUM_THREADS" or specify the wrong value, it may result in an unintended number of parallel executions. Please note.

5.3.3 How to use MPI

This is a sample job script that assumes 1,024 parallel execution within 4 nodes.

- queue : OCT
- group : G01234
- Elapsed time : 30 minutes (1 hour if not specified)
- Total CPU Cores: 1,024 (128 cores× 2CPUs× 4 nodes)
- Number of nodes: 4 (no need to specify when running within one node)
- MPI Library: Intel MPI

```
1   #!/bin/bash
2   #------- qsub option -----------
3   #PBS -q OCT
4   #PBS --group=G01234
5   #PBS -l elapstim_req=00:30:00
6   #PBS -b 4
7   #PBS -T intmpi
8
9   #------- Program execution -----------
```

```
10
11  module load BaseCPU/2025
12  module load xxx/xxx
13
14  cd $PBS_O_WORKDIR
15  mpirun ${NQSV_MPIOPTS} -np 1024 ./a.out
```

- Line 6
  Specify the number of nodes.

- Line 7
  When using MPI, specify -T intmpi .

- Line 12
  Describe what you had module load at the time of compilation.

- Line 15
  Specify ${NQSV_MPIOPTS} as an argument to mpirun. This designation passes the host on which the job will run to MPI.
  The environment variable PBS_O_WORKDIR is automatically assigned the path to the current directory where the qsub command was executed.

5.3.4 MPI+ Intra-node parallelism How to use

A sample job script that assumes program execution that generates 4 processes (1 process per node) and each process generates 256 threads.

- queue : OCT
- group : G01234
- Elapsed time : 30 minutes (1 hour if not specified)
- Total CPU Cores: 1,024 (128 cores× 2CPUs× 4 nodes)
- Number of nodes: 4 (no need to specify when running within one node)
- MPI Library: Intel MPI
- Number of threads: 256 (specified by environment variable OMP_NUM_THREADS)

```
1   #!/bin/bash
2   #------- qsub option -----------
3   #PBS -q OCT
4   #PBS --group=G01234
5   #PBS -l elapstim_req=00:30:00
6   #PBS -b 4
7   #PBS -T intmpi
8   #PBS -v OMP_NUM_THREADS=256
9
10  #------- Program execution -----------
11
12  module load BaseCPU/2025
```

```
13  module load xxx/xxx
14
15  cd $PBS_O_WORKDIR
16  mpirun ${NQSV_MPIOPTS} -np 4 ./a.out
```

- Line 7
  When using MPI, specify -T intmpi .

- Line 13
  Describe what you had module load at the time of compilation.

- Line 16
  Specify ${NQSV_MPIOPTS} as an argument to mpirun. This designation passes the host on which the job will run to MPI.
  The environment variable PBS_O_WORKDIR is automatically assigned the path to the current directory where the qsub command was executed.

5.3.5 Advanced usage

➢ **Manual Specification of the Number of Processes in a Node in Intel MPI**

In Intel MPI, it is possible to restrict processes in a node by specifying -ppn, -rr, and -perhost options (including I_MPI_PERHOST environment variables). However, if the -machinefile option is used together, the -machinefile option will take precedence.

The OCTOPUS job management system includes the -machinefile option among NQSV_MPIOPTS environment variables, so the -ppn, -rr, and -perhost options are disabled. If you want to limit the number of processes in a node, you can do so by specifying the number of cores required in the -l cpunum_job option. However, if you want to specify more details, such as using core pinning, etc., PBS_NODEFILE environment variables are used.

Below is a sample job script that utilizes PBS_NODEFILE environment variables to generate 128 processes (64 processes per node) on 2 nodes.

- queue : OCT
- group : G01234
- Elapsed time : 30 minutes (1 hour if not specified)
- Total CPU Cores : 128 (64 cores × 2 nodes)
- Number of nodes: 2 (no need to specify when running within one node)
- MPI Library: Intel MPI

```
1  #!/bin/bash
2  #------- qsub option -----------
3  #PBS -q OCT
4  #PBS --group=G01234
5  #PBS -l elapstim_req=00:30:00
6  #PBS -b 2
7  #PBS -T intmpi
```

```
8
9   #------- Program execution -----------
10
11  module load BaseCPU/2025
12  module load xxx/xxx
13
14  cd $PBS_O_WORKDIR
15  mpirun -hostfile ${PBS_NODEFILE} -np 128 -ppn 64 ./a.out
```

- Line 6
  Specify the number of nodes.

- Line 7
  When using MPI, specify -T intmpi .

- Line 12
  Describe what you had module load at the time of compilation.

- Line 15
  Specify -hostfile ${PBS_NODEFILE} as an argument to mpirun. Through this designation, MPI knows which host the job is running on.
  In addition, -ppn 64 can be specified to specify 64 processes per node.

➢ **Performance analysis with VTune profiler**

The VTune profiler allows you to perform performance analysis of user-developed serial and multi-threaded applications.

Below is a sample job script that uses the VTune Profiler to perform performance analysis.

- queue : OCT
- group : G01234
- Elapsed time : 30 minutes (1 hour if not specified)
- Total CPU Cores: 2 (2 cores× 1 node)
- MPI Library: Intel MPI

```
1   #!/bin/bash
2   #------- qsub option -----------
3   #PBS -q OCT
4   #PBS --group= G01234
5   #PBS -l elapstim_req=00:30:00
6   #PBS -T intmpi
7
8   #------- Program execution -----------
9   module load BaseCPU/2025
10  module load xxx/xxx
11
12  cd $PBS_O_WORKDIR
13
14   aps --result-dir=./result --collection-mode=all mpirun ${NQSV_MPIOPTS} -ppn 2 -np
```

```
    2 ./a.out
```

- Line 6
  When using MPI, specify -T intmpi .

- Line 9
  Describe what you had module load at the time of compilation.

- Line 14
  Continue with aps commands and options, and include the mpirun command. --result-dir specifies the destination path for the analysis results. Also, in --collection-mode, specify the list of data you want to collect, separated by commas, as shown below. In this example, we specify "All".
  HWC: Hardware Counter
  omp:OpemMPI Stats
  mpi:MPI Stats
  all: All

When you run a batch job, the report is printed in the current directory with the name of the aps_report_YYYYMMDD_HHMISS.html, and the performance analysis results are printed in the directory specified in --result-dir.

For information on how to use V Tune Profiler and how to view reports, please refer to the V Tune Profiler User Guide.

5.4 How to run a container

This section describes job scripts when using containers to run jobs. For an overview of how to use containers in this system, please refer to "1.5.4 Container Usage".

5.4.1 Overview of container execution

Learn about the minimum things you need to know to run containers. As a container for a standard environment, let's take the Rocky Linux　image file (rocky9.sif) as an example.

➢ **Execution Command**

Container execution is performed by specifying the exec subcommand. As an example, if you want to run the hostname command in the rocky9.sif container image, run the following command:

```
$ apptainer exec rocky9.sif hostname
```

The command format is as follows:

```
$ apptainer exec < the image filename > < run command in the container>
```

Note that the command you specify is **an execution command in the container.** If the command is pathless, the command explored with the PATH environment variable in the container is executed. Even if there is a path specified, the absolute path follows the file structure in the container.

➢ Environment Variables

Environment variables defined outside of the container are basically carried over to the container. However, environment variables that are explicitly defined on the container side, such as at build time, follow the container definition.

If you want to override environment variables defined in the container, you can pass them into the container either individually using the --env option or in bulk using the --env-file option.

- --env option to specify individually

```
$ apptainer exec --env MYVAR="My Value!" rocky9.sif myprog.exe
```

- --env-file option to specify in bulk

```
$ cat myenvfile
MYVAR="My Value!"
$ apptainer exec --env-file myenvfile rocky9.sif myprog.exe
```

For more information on environment variables, please refer to the official documentation below.
https://apptainer.org/docs/user/latest/environment_and_metadata.html

➢ Mounting the Host OS

If you want to read/write the host OS's file system from within the container, you can bind mount a specific directory on the host OS. Even without specifying options, **the following directories are mounted as standard and can be used in containers with the same path.**

Home directory: /octfs/home/ (user number)
Temporary Region :/tmp

For example, the command to run a program (a.out) placed in the home directory of the host OS from within the container is as follows:

```
$ apptainer exec rocky9.sif ./a.out
```

* In the above example, the current directory has been moved to home in the container.

To mount a specific directory from the host OS, use the --bind option. The format of the --bind option is as follows.

--bind < host OS path >: < path in container >: < mode>

The path and mode (ro/rw) in the container are optional. If omitted, the path in the container is mounted in read/write with the same path as the host OS.

For example, the command to run a program (a.out) placed in a directory on the extension area is as follows:

```
$ cd /octfs/work/ (group name)/ (user number)
$ apptainer exec --bind `pwd` rocky9.sif ./a.out
```

* In the above example, the environment variable PWD outside the container is carried over to the container, and the current directory in the container is the directory in the extension area.

For more information on mounting the host OS, please refer to the official documentation below.
https://apptainer.org/docs/user/latest/bind_paths_and_mounts.html

5.4.2 How to run in a general-purpose CPU computing environment

This is a sample job script that assumes the execution of a thread-parallel program within a single node in a general-purpose CPU computing environment.

- queue : OCT
- group : G01234
- Elapsed time : 30 minutes (1 hour if not specified)
- Total CPU Cores: 256 (128 cores × 2CPUs × 1 node)

```
1   #!/bin/bash
2   #------- qsub option -----------
3   #PBS -q OCT
4   #PBS --group=G01234
5   #PBS -l elapstim_req=00:30:00
6   #PBS -v OMP_NUM_THREADS=256
7
8   #------- Program execution -----------
9
10  cd $PBS_O_WORKDIR
11  apptainer exec --bind `pwd` image.sif ./a.out
12
```

- Line 6
  Specify the number of parallel runs.

- Line 11
  Specify the image file and run the apptainer exec command.

Precautions when using
Don't forget to specify "OMP_NUM_THREADS" in the execution script.
If you do not specify "OMP_NUM_THREADS" or specify the wrong value, it may result in an unintended number of parallel executions. Please note.

6 How to use the application

The system introduces the following applications.

6.1 Application List

- ISV Applications

ISV Applications	Available Hosts		
	octopusX	voctopusXX	General Purpose CPU
AVS/Express Developer	○	○	-
Gaussian	○	○	○
IDL	○	○	-

- OSS Applications

OSS Applications	Available Hosts		
	octopusX	voctopusXX	General Purpose CPU
ABINIT-MP	○	○	○
ADIOS	○	○	○
CTFFIND	○	○	○
FLASHcode	○	○	○
FreeFem++	○	○	○
GAMESS	○	○	○
GENESIS	○	○	○
GROMACS	○	○	○
Gnuplot	○	○	-
ImageMagick	○	○	-
LAMMPS	○	○	○
MotionCor3	○	-	-
NCView	○	○	○
OpenFOAM	○	○	○
ParaView	○	○	-
Quantum ESPRESSO	○	○	○
Relion	○	○	-
ResMAP	○	○	-
VisIt	○	○	○

6.2 How to use ISV applications

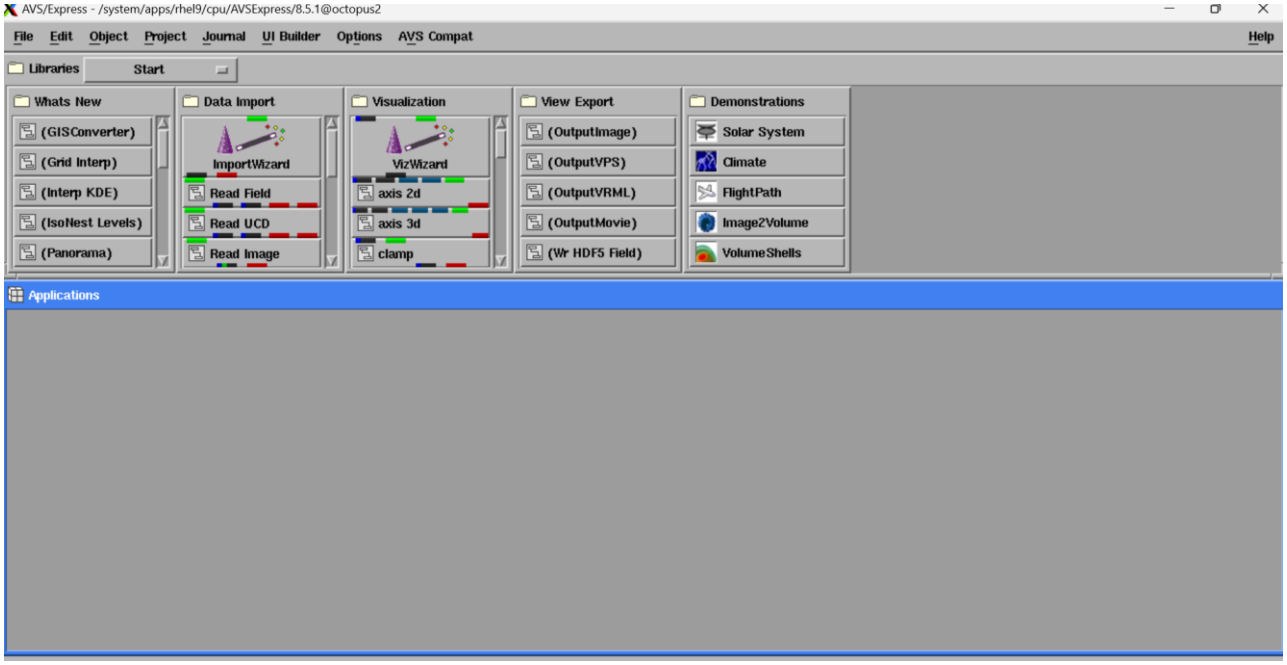
Learn how to use ISV applications available on OCTOPUS systems.

6.2.1 AVS/Express

AVS/Express, a visualization application, is available in version 8.5. The number of concurrent users of OCTOPUS is 5. Log in to the front-end using X terminal software, etc., and run the following command to start AVS/Express.

```
$ module load BaseApp
$ module load AVSExpress/8.5
$ express
```

The operation screen of "AVS/Express" is displayed as follows.



In addition, "AVS/Express" provides a license server on the SQUID system. If you want to install and use AVS/Express on your device, please specify the following license server and port number.

nape	License	Port Number
1	squidportal.hpc.cmc.osaka-u.ac.jp	XXXXX/tcp

The license is a floating license and the number of licenses is 5. If you are missing a license, you will see the following message in the terminal at startup:

```
Could not get license from server: license limit exceeded
```

6.2.2 Gaussian

Gaussian, a quantum chemistry calculation program, offers version g16. To use it, you must contact the Center to request registration in the Gaussian Application Group.

Run the program in a batch job. The following is an example of running a sample program:

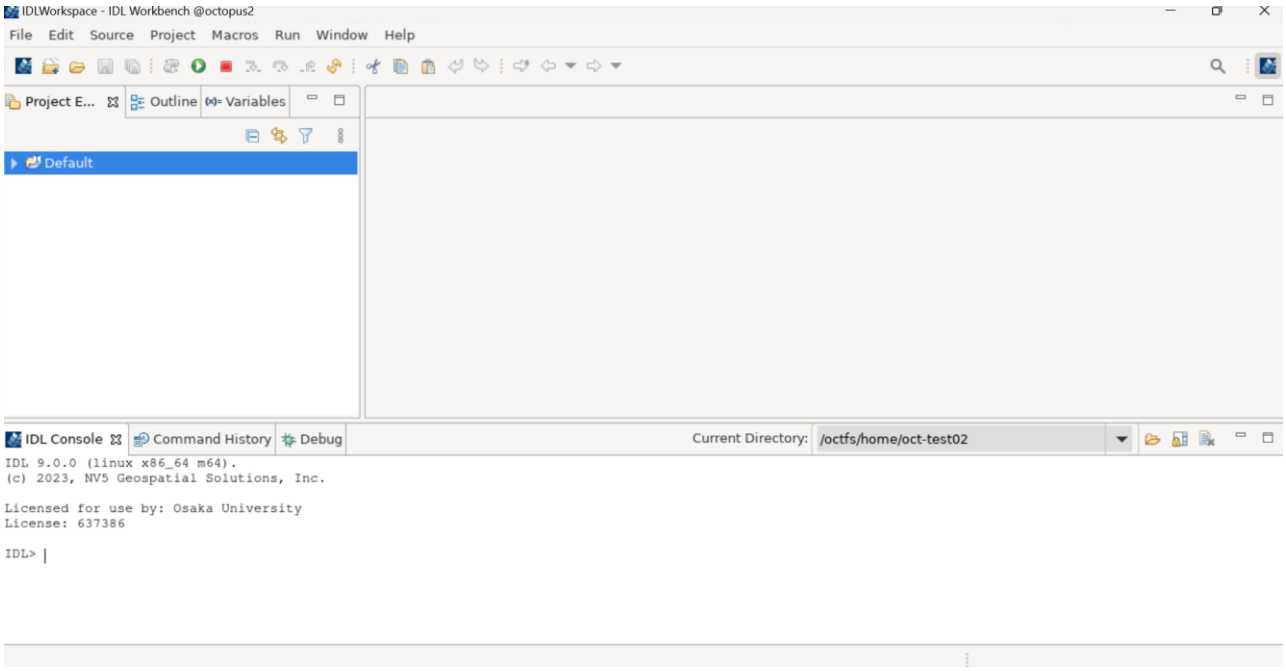
```
#!/bin/bash

#PBS -q OCT
#PBS --group=<group name>
#PBS -l elapstim_req=01:00:00

module load BaseApp
module load Gaussian

newgrp gaussian

cd $PBS_O_WORKDIR
g16 < Input File >& result
```

6.2.3 IDL

"IDL" offers version 9.0. The number of concurrent users will be 4. Log in to the front-end using X terminal software, etc., and run the following command to start "IDL".

```
$ module load BaseApp
$ module load IDL/9.0
$ idlde
```

The IDL operation screen is displayed as follows.



The license is a floating license and the number of licenses is 4. If you are missing a license, a dialog box appears with the following message:

```
Unable to license IDL.
```

6.3 How to use OSS applications

Learn how to utilize the main OSS applications available on OCTOPUS systems.

6.3.1 ABINIT-MP

ABINIT-MP is a software that enables fast fragment molecular orbital (FMO) calculations. By linking with the dedicated GUI BioStation Viewer, it is easy to create input data ~ analyze calculation results. Secondary perturbation calculations by four-body fragment deployment (FMO4) are also possible.

There are restrictions on the modules that can be used.

node classification	With F function		No F function	
	SMP Ready	Not SMP Supported	SMP Ready	Not SMP Supported
cpu	abinitmp_smp-fint	abinitmp-fint	abinitmp_smp	abinitmp

< CPU node>

```
#!/bin/bash
#PBS -q OCT-S
#PBS -l cpunum_job=20
#PBS --group=<group name>
#PBS -l elapstim_req=01:00:00
```

```
#PBS -b 1
#PBS -T intmpi

module load BaseCPU
module load BaseApp
module load abinit-mp/2.8

cd ${PBS_O_WORKDIR}
export MPI_PROC=20

#------- ABINIT-MP option ------- #
BINARY_NAME=abinitmp

#------- Input file ------- #
FILE_NAME=gly5-mpi
AJF_NAME=${FILE_NAME}.ajf

#------- Output file ------- #
NUM_CORE=_001n-020p-ver2rev8
AJF_NAME=${FILE_NAME}.ajf
OUT_NAME=${FILE_NAME}${NUM_CORE}.log

#------- Program execution ------- #
mpiexec -n ${MPI_PROC} ${BINARY_NAME} < ${AJF_NAME} > ${OUT_NAME}
```

6.3.2 ADIOS

After logging in to the front-end, do the following:

```
$ module load BaseApp
$ module load adios/1.13.1
```

Please refer to the manual below for instructions on how to use it.
https://users.nccs.gov/~pnorbert/ADIOS-UsersManual-1.13.1.pdf

6.3.3 CTFFIND

Run the program in a batch job. The following is an example of running a sample program:

```
#!/bin/bash
#PBS -q OCT
#PBS --group=<group name>
#PBS -l elapstim_req=01:00:00

module load BaseApp
module load CTFFIND/4.1.14

cd $PBS_O_WORKDIR
ctffind < ctffind_input.txt
```

6.3.4 FLASHcode

Run the program in a batch job. The following is an example of running a sample program:

```
#!/bin/bash
#PBS -q OCT-S
#PBS -l cpunum_job=4
#PBS --group=<group name>
#PBS -l elapstim_req=01:00:00
#PBS -b 1
#PBS -T intmpi

module load BaseApp
module load FLASHcode/4.8

cd $PBS_O_WORKDIR
cp -p /system/apps/rhel9/cpu/FLASHcode/InteloneAPI2025.2.0/4.8/examples/flash.par .

mpirun -np 4 flash4
```

6.3.5 FreeFem++

Run the program in a batch job. The following is an example of running a sample program:

```
#!/bin/bash
#PBS -q OCT
```

```
#PBS --group=<group name>
#PBS -l elapstim_req=24:00:00
#PBS -b 1
#PBS -T intmpi

module load BaseApp
module load FreeFemXX/4.15

cd ${PBS_O_WORKDIR}

cp -r /system/apps/rhel9/cpu/FreeFemXX/InteloneAPI2025.2.0/4.15/share/FreeFEM/4.15/examples .
cd examples
./go_CheckAllEdp.sh
```

6.3.6 GAMESS

Run the program in a batch job. The following is an example of running a sample program:

```
#!/bin/bash

#PBS -q OCT
#PBS --group=<group name>
#PBS -l elapstim_req=24:00:00
#PBS -b 1
#PBS -T intmpi

cd ${PBS_O_WORKDIR}

module load BaseApp
module load GAMESS/v2024.2.1

export SCR=${PBS_O_WORKDIR}
export USERSCR=${PBS_O_WORKDIR}
rungms exam01.inp 00 24
```

If you specify a scratch directory, define the following environment variables before the program runs:

```
$ export SCR=<scratch directory>
$ export USERSCR=<scratch directory>
```

6.3.7 Gnuplot

After logging in to the front-end, do the following:

```
$ gnuplot
```

6.3.8 ImageMagick

ImageMagick is a software for manipulating and displaying images.
Below is an example of how to change the extension of an image (JPG→PNG).

```
$ convert sample.jpg sample.png
```

6.3.9 LAMMPS

LAMMPS is an open-source molecular dynamics application.

The deployment package is as follows:

```
ASPHERE、ATC、AWPMD、BOCS、BODY、CG-DNA、CG-SPICA、CLASS2、COLLOID、
COLVARS、COMPRESS、CORESHELL、DIFFRACTION、DIPOLE、DPD-MESO、 DRUDE、
EFF、ELECTRODE、EXTRA-COMPUTE、EXTRA-DUMP、EXTRA-FIX、EXTRA-MOLECULE、
EXTRA-PAIR、FEP、GRANULAR、INTEL 、INTERLAYER、KSPACE、LATBOLTZ、MANIFOLD、
MANYBODY、MC、MDI、MGPT、MISC、MOFFF、MOLECULE、 NETCDF、OPENMP、OPT、
ORIENT、PERI、PHONON、POEMS、PYTHON、QEQ、QTB、REAXFF、REPLICA 、RIGID、
SHOCK、SMTBQ、SPH、SPIN、SRD 、TALLY、UEF、VORONOI
```

Run it in a batch job. An example job script is shown below.

```
#!/bin/bash
#PBS -q OCT
#PBS -l cpunum_job=256
#PBS --group=<group name>
#PBS -l elapstim_req=01:00:00
#PBS -b 1
#PBS -v OMP_NUM_THREADS=128

module load BaseApp
module load lammps/22Jul2025
cd $PBS_O_WORKDIR
mpirun ${NQSV_MPIOPTS} -np 2 lmp < ./in.lj
```

6.3.10 MotionCor3

Run it on the front-end.

```
module load BaseApp
module load MotionCor3/1.0.1


MotionCor3 -InMrc test_boxmovie.mrcs -OutMrc test_boxmovie_corrected.mrc
```

6.3.11 OpenFOAM

OpenFOAM is a fluid/continuum simulation platform.
Run it in a batch job. An example job script is shown below.

<Serial version>

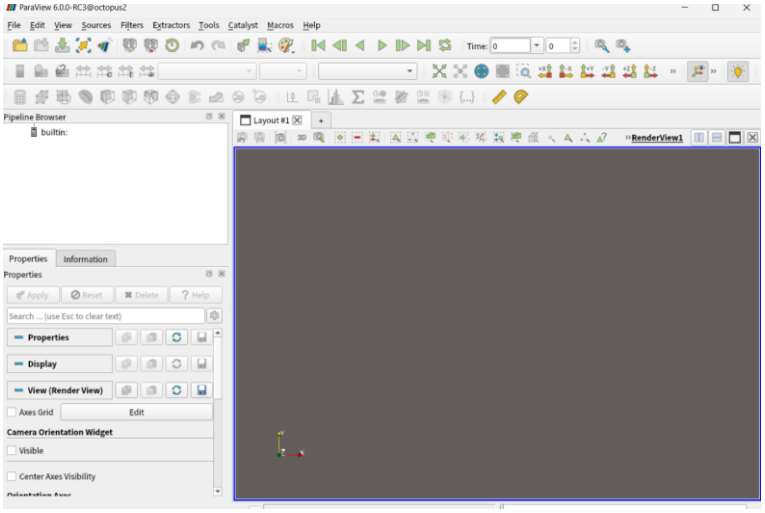
```
#!/bin/bash
#PBS -q OCT-S
#PBS -l cpunum_job=1
#PBS --group=<group name>
#PBS -l elapstim_req=01:00:00

module add BaseApp
module load OpenFOAM/v2412

blockMesh
icoFoam
```

6.3.12 ParaView

If you want to use "ParaView" on the front-end, use X terminal software, etc., log in to the front-end, and do the following.

```
$ module load BaseApp
$ module load ParaView/6.0.0
$ paraview
```

The ParaView operation screen is displayed as follows.



6.3.13 Relion

Relion is an open-source image processing software for electron microscopy. After logging in to the front-end, do the following:

```
$ module load BaseApp
$ module load relion/5.0.0
$ relion
```

The Relion operation screen is displayed as follows.

6.3.14 ResMAP

If you want to use "ResMAP" on the front-end, use X terminal software, etc., log in to the front-end, and perform the following.

```
$ module load BaseApp
$ module load ResMap/1.1.4
$ ResMap
```

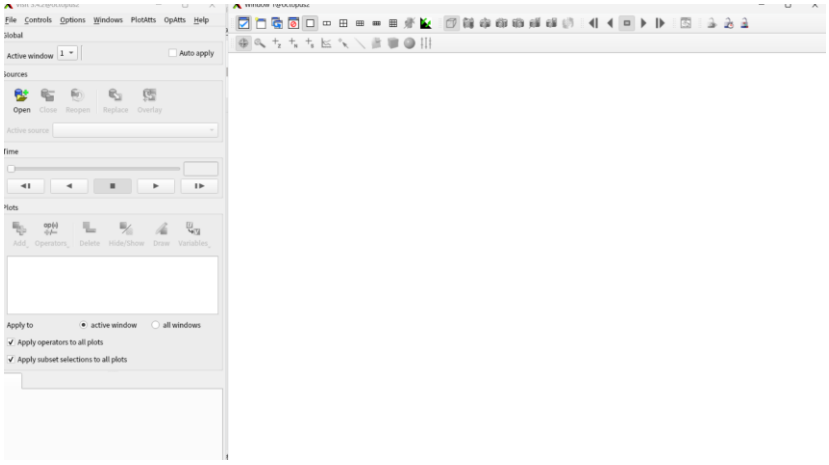
*Font-related errors and warnings are output at startup, but there is no problem with execution.
Please contact us if you have any problems with the implementation.

6.3.15 VisIt

VisIt is an open-source visualization software. After logging in to the front-end, do the following:

```
$ module load BaseApp
$ module load VisItv/3.4.2
$ visit
```

The VisIt operation screen is displayed as follows.



7 How to transfer files Advanced usage

7.1 File transfer in a web browser

As a way to transfer files to OCTOPUS in a web browser, we have prepared a Nextcloud environment. Files saved via Nextcloud are stored in the following paths in OCTOPUS and can be used from front-end and computing environments.

/octfs/home/(user number)/OnionWeb/

For example: for user ID "user001" /octfs/home/user001

7.1.1 login

Open your web browser and enter the following URL:

https://oct-dtg1.hpc.osaka-u.ac.jp

The login screen will be displayed, enter the same username and password as the user management system, and click "Login".

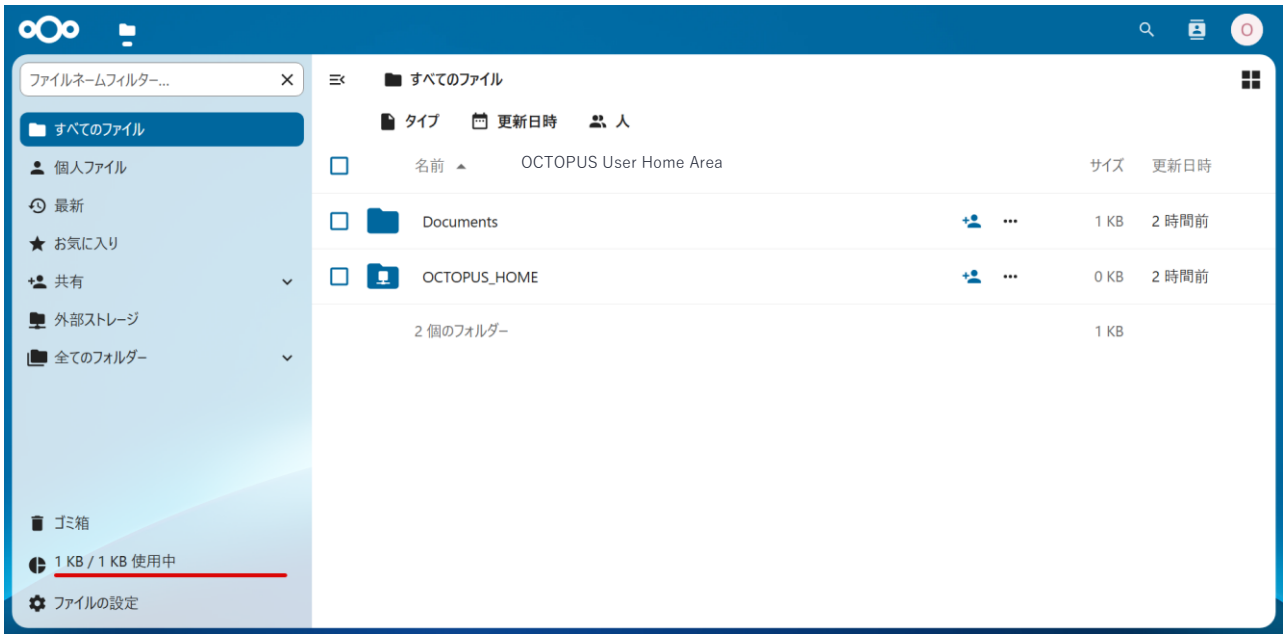
Set up two-factor authentication. Select "TOTP (Authenticator app)".

A QR code for Nectcloud will be displayed, so please read it with the two-factor authentication application installed on your device or smartphone.

Select "TOTP (Authenticator app)".

Enter the 6-digit verification code displayed in the two-factor authentication application.

When you log in, you will be taken to the following screen.



After logging in, your home area will be automatically mounted and displayed as a "OCTOPUS_HOME".

7.1.2 Basic usage

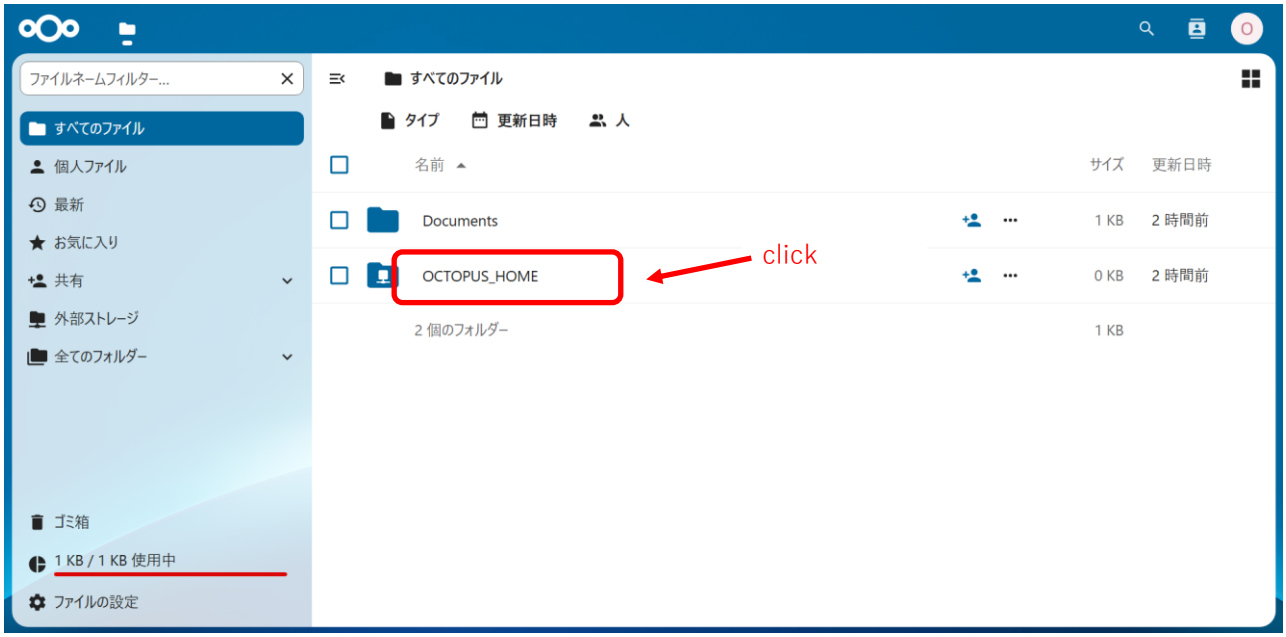
➢ **How to upload**

The file upload destination must be "OCTOPUS_HOME" or "7.1.4 This will be one of the storage added in "Add External Storage".

(1) Drag-and-drop upload

Open the location you want to upload and drag and drop the files you want to upload.
In the example below, we upload a file directly below "OCTOPUS_HOME".

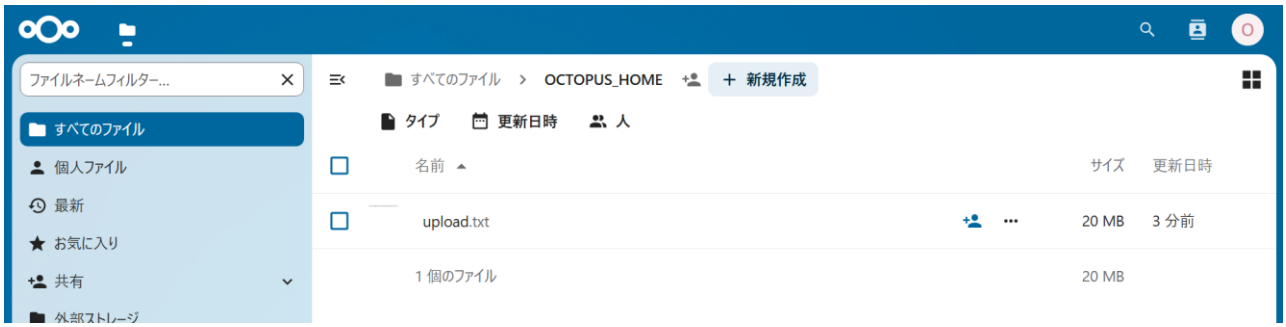
Click OCTOPUS_HOME.



Drag and drop the files you want to upload.

When the upload starts, you will see a progress bar like this, so wait a bit.

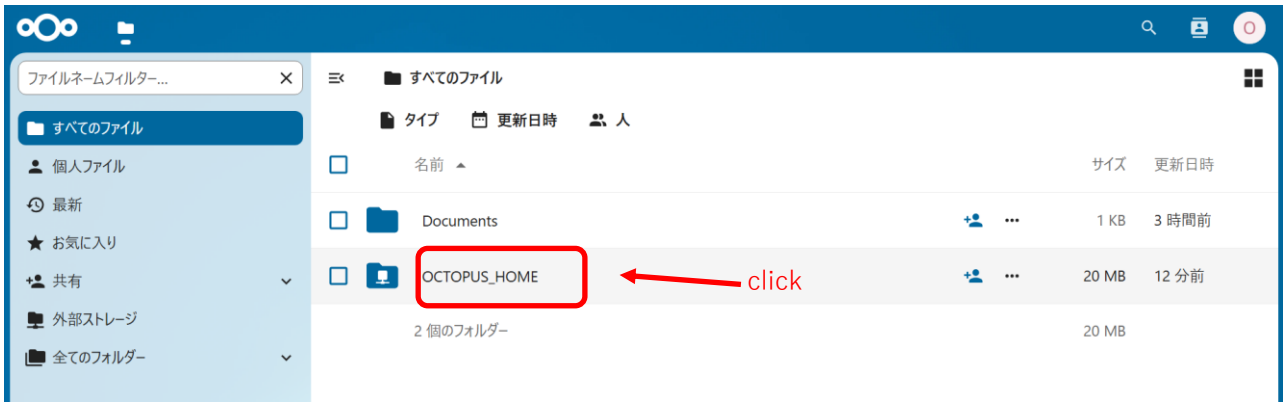
When you see the dragged and dropped file, the upload is complete.



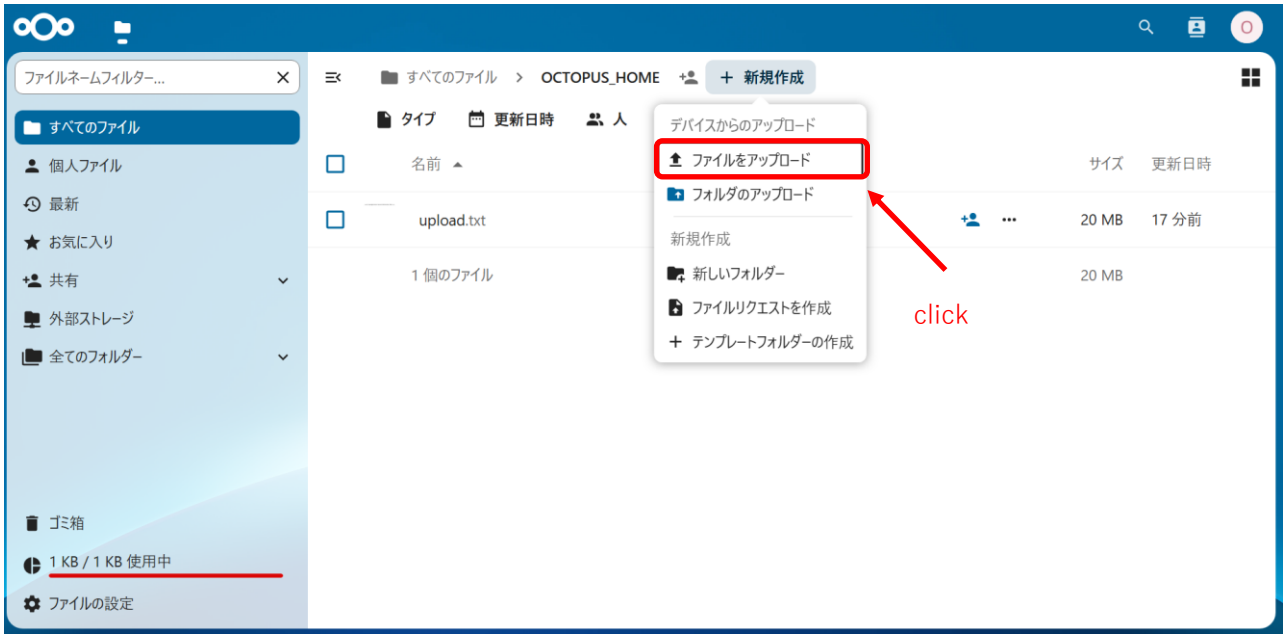
(2) Upload by reference

Open the location you want to upload and select the file you want to upload.
In the example below, we upload a file directly under the OCTOPUS User Home Area.

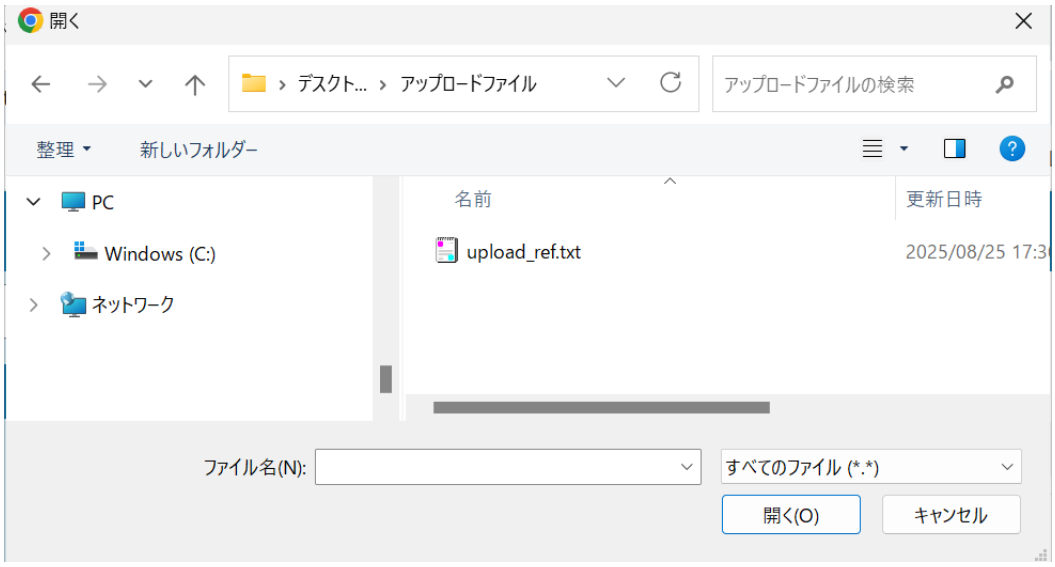
Click OCTOPUS User Home Area.



Click ⊕ and click Upload File that appears.



The file reference dialog appears, select the file you want to upload, and click Click Open



When the upload starts, you will see a progress bar like this, so wait a bit.

The upload is complete when the selected file is displayed.

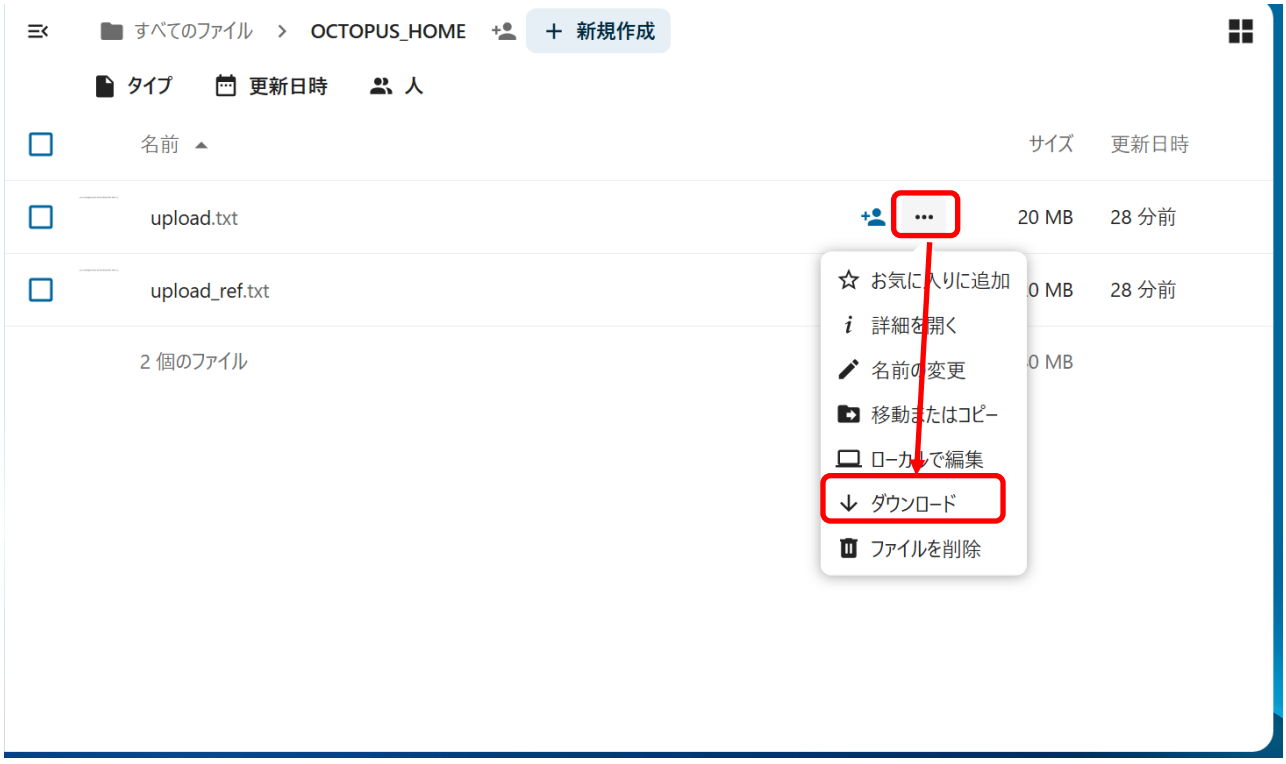
➢ How to download

(1) Single Download

If you want to download only one file, the steps are as follows.

In the example below, we will download the "test.txt" in the "OCTOPUS_HOME". Click "..." to the right of the file you want to download, and then click "Download" that appears to start the download.

Specifying the location of the download file depends on the settings of the web browser you are using.



(2) Batch download

You can also download multiple files at once.

If you download them together, the compressed file in zip format will be downloaded, so please unzip it separately.

In the example below, we download the "upload.txt" and "upload_ref.txt" together.

Tick the checkbox to the left of the file you want to download.

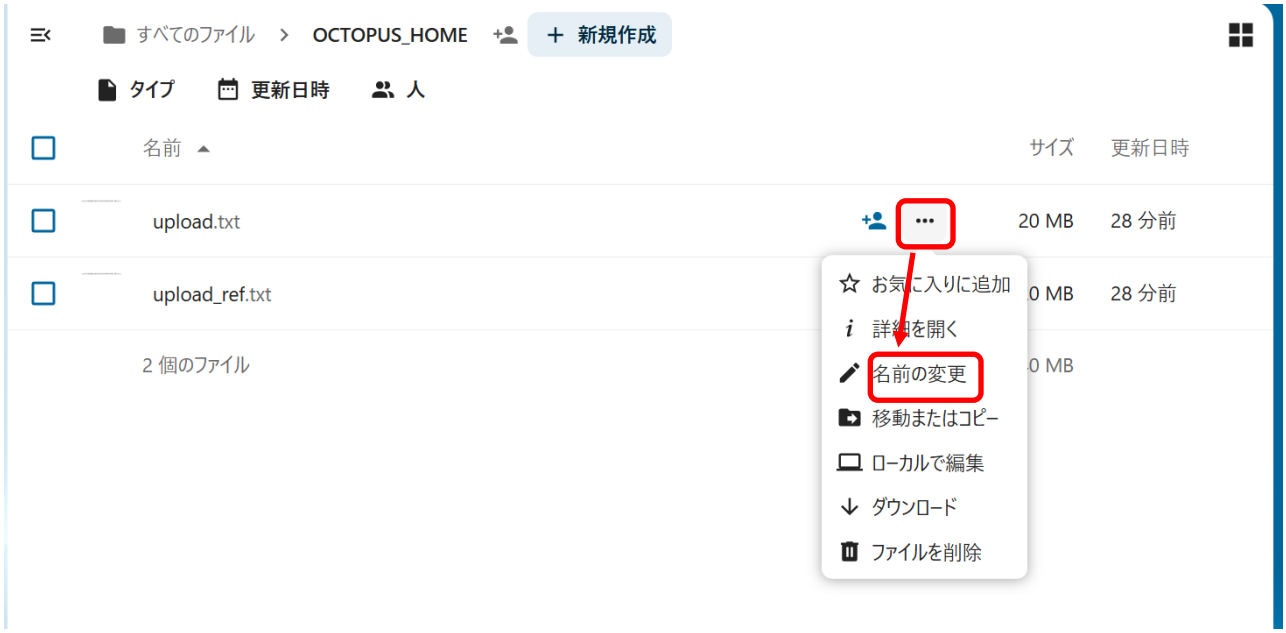
At the top of the file name list, click "... Click "Actions" and click "Download" that appears. Click on it.

Specifying the location of the download file depends on the settings of the web browser you are using.

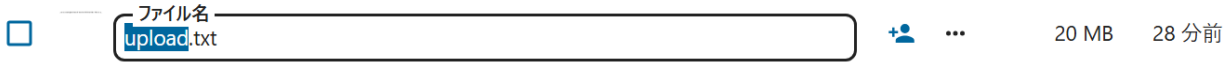
➢ **Folder Name and File Name Change**

To change the folder name and file name, follow these steps.
In the example below, we will rename "upload.txt".

Click ..., to the right of the folder or file you want to rename, and then click Rename that appears.

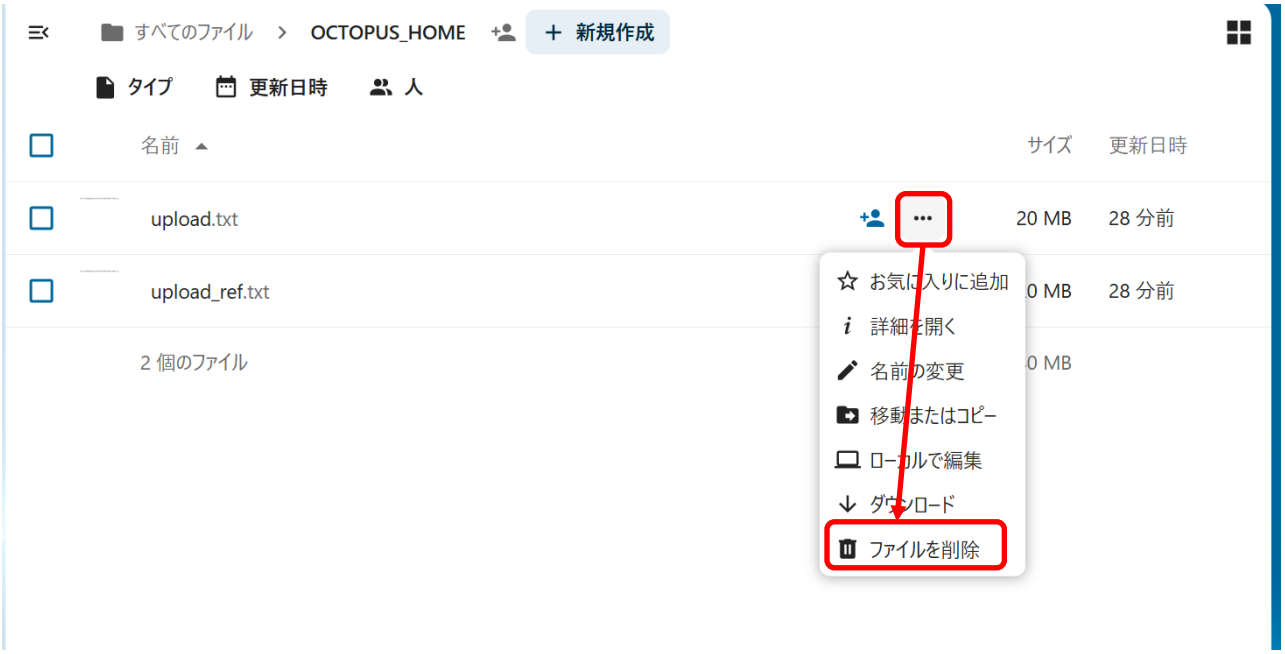


You will be able to edit the target file name, so change it to any name and enter the enter key.



➢ Deleting Files

The following steps are used to delete folder names and files. In the example below, we will remove "upload.txt". Click "..." on the right side of the folder or file you want to delete, and click "Delete File" that appears.



If the selected folder or file has disappeared from the file list screen, the deletion is complete.

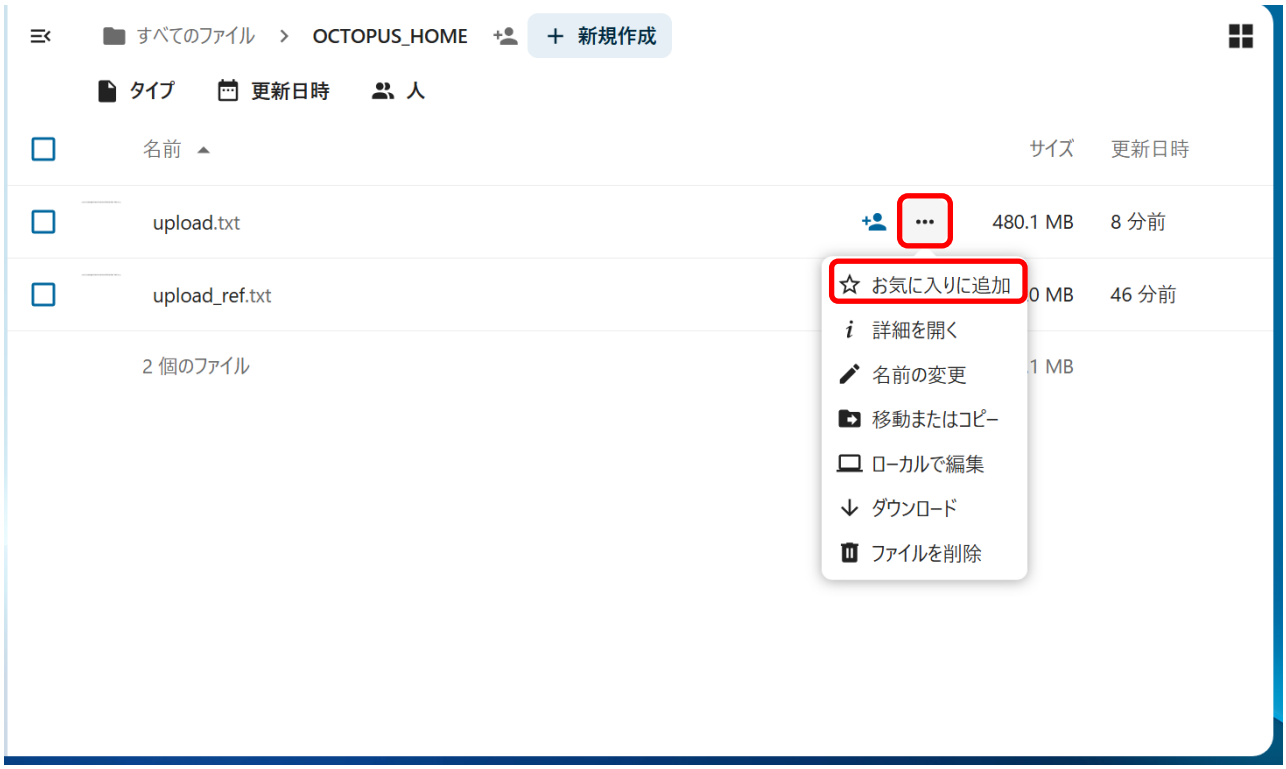
➢ **favorite**

You can quickly browse frequently used folders and files by adding them to your favorites.

(1) Add to favorite

To create a favorite folder or file, follow the following steps.
In the example below, we will add "upload.txt" to our favorites.

Click "..." to the right of the folder or file you want to add to favorites, and then click "Add to Favorites" that appears.



When you add it to your favorites, you will see a "★" mark in the upper right corner of the icon.

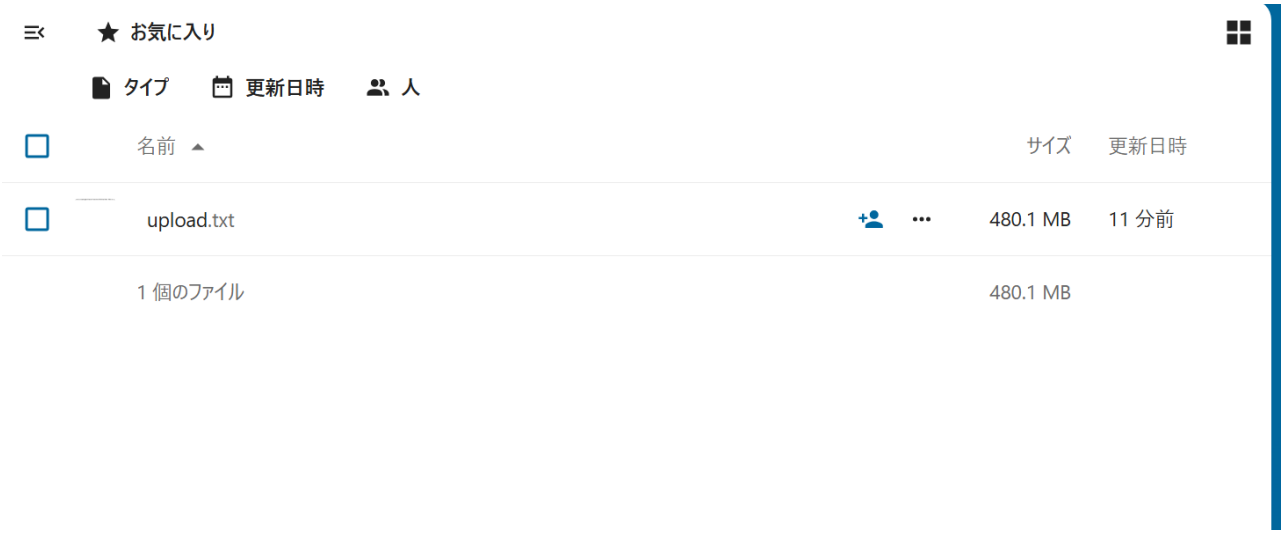
To browse the folders and files that are currently registered as favorites, click "Favorites" in the left menu, and then click the file you want to browse.

(2) Remove from favorites

To delete a folder file from favorites, follow these steps:
In the example below, we remove "upload.txt" from favorites.
Click "Favorites" in the left menu.

The file you want to remove from your favorites Click "..." on the right side and click "Remove from Favorites" that appears.

If the "★" mark is missing from the previous file, it is deleted from favorites.



7.1.3 Sharing Folders and Files

By specifying folders and files and publishing URLs for sharing, you can share folders and files with people who do not have accounts in a data-intensive environment.

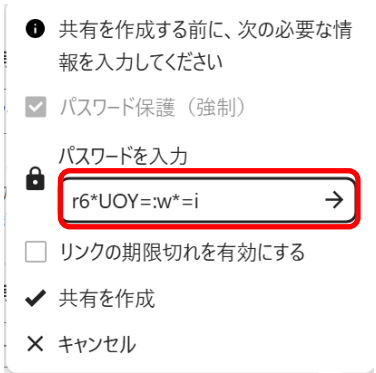
(1) Folder and file sharing settings (URL sharing)

To share folders and files, follow these steps:
The following example shares the "upload.txt".

Click the share mark to the right of the folder or file you want to share, then click <.

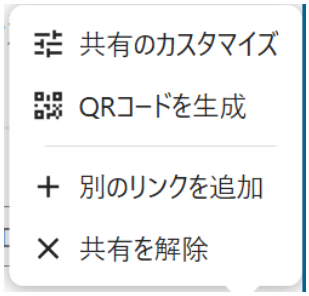
The following menu will appear on the right, click "+" to the right of "Share by URL".

The following screen will appear, so please refrain from using the password displayed in "Password Protection (Enforced)" or set a password of your choice. If you want to set an optional password, it must be at least 12 characters long. After that, click on "→" to the right of the password.



The icon on the right side of "Share by URL" will change to the following and click "...".

The following menu will appear, so you can change the settings as needed.
After making changes, click outside the menu.



Shared labels: labels managed by the sharing owner
Allow editing: This is not enabled due to feature limitations.
Hide downloads: Shared URLs hide files even when connected
It will not be enabled because of this.
Password protection (enforced): Cannot be removed. It can be changed arbitrarily.
After making changes, click on "→".
Set expiration date: You can set the expiration date of the shared URL.
Note to recipients: You can add a message in the top right corner of the share URL.

Click the clipboard icon to the right of "Share by URL" and copy the URL.

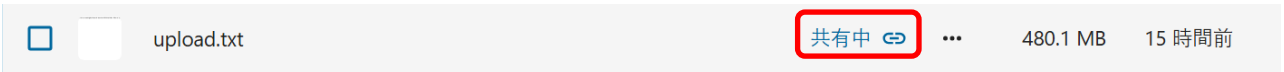
Send the password and copied URL to the person you want to share the folder or file with via email.

(2) Unsharing Folders and Files

To unshare a folder file, follow these steps:
In the example below, we unshare the "upload.txt".

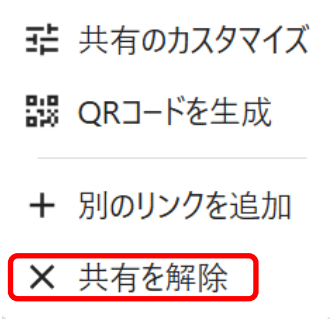
Click Share in the left menu.

Click Now Sharing to the right of the folder or file you want to unshare.



The following menu will appear on the right, click "..." to the right of "Share by URL".

The following menu appears, click Unshare.



If the icon on the right side of "Share by URL" changes to the following, the unsharing is complete.

7.1.4 Adding External Storage

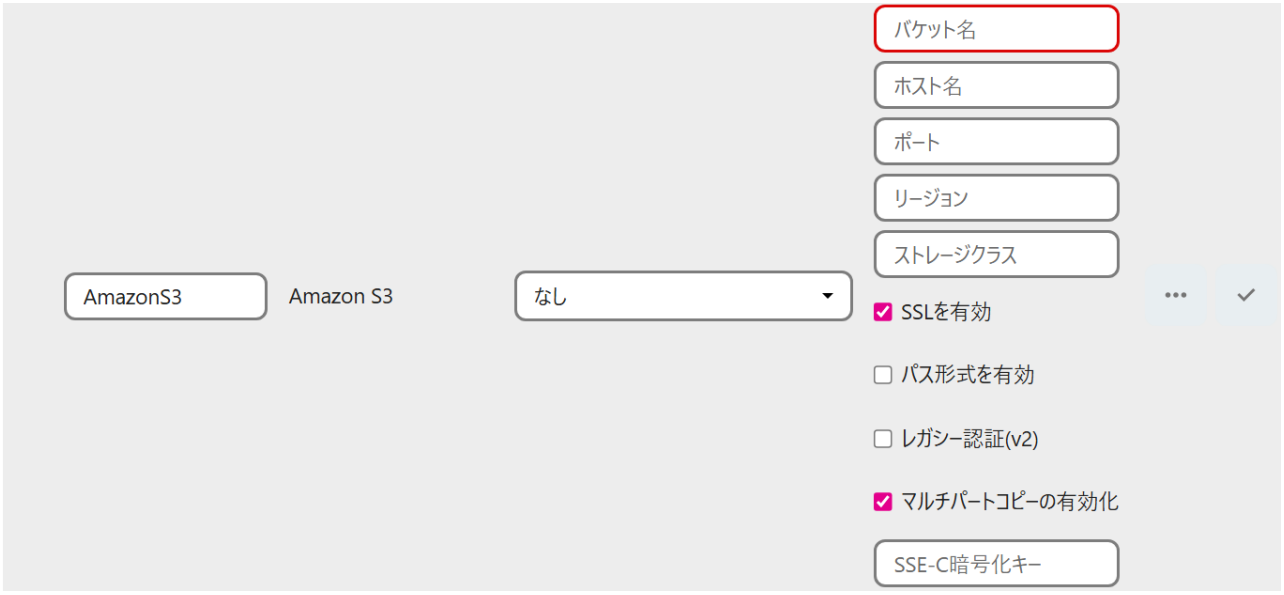
By adding external storage, you can use storage that supports the Amazon S3 API from the Nextcloud environment.

Click on the user icon in the top right corner of the screen and click on "Settings" from the menu that appears.

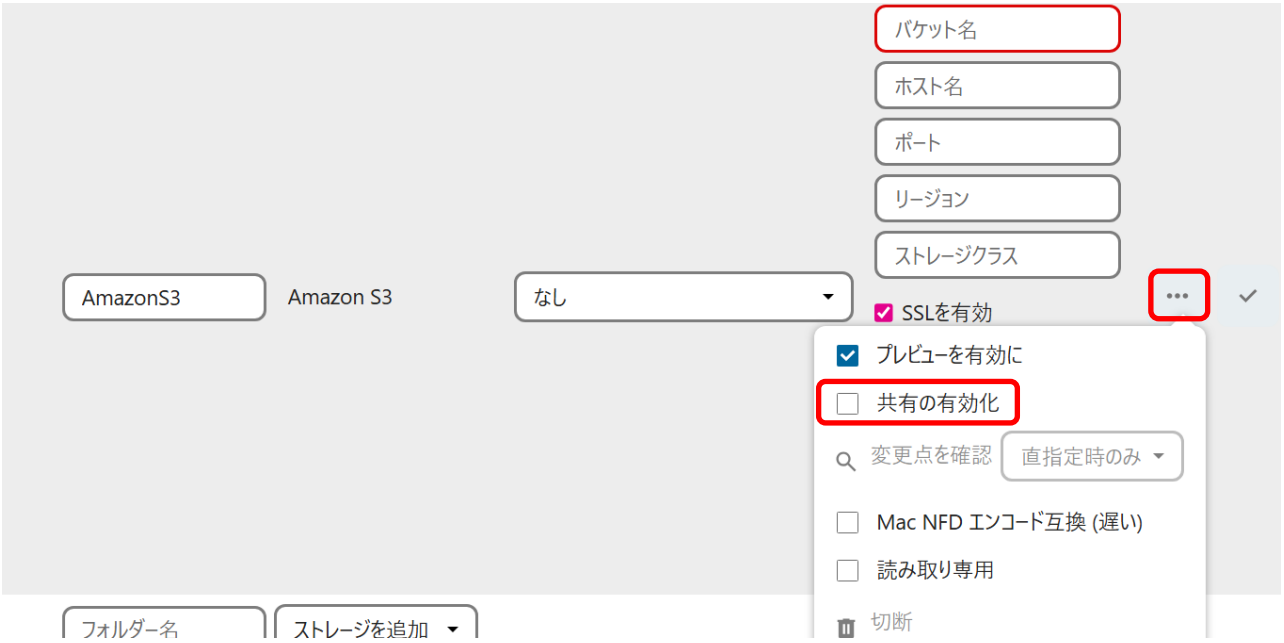
Click on "External Storage" in the left menu.

Under External Storage, enter a name of your choice in the Folder Name, and click Add Storage ・ Amazon S3.

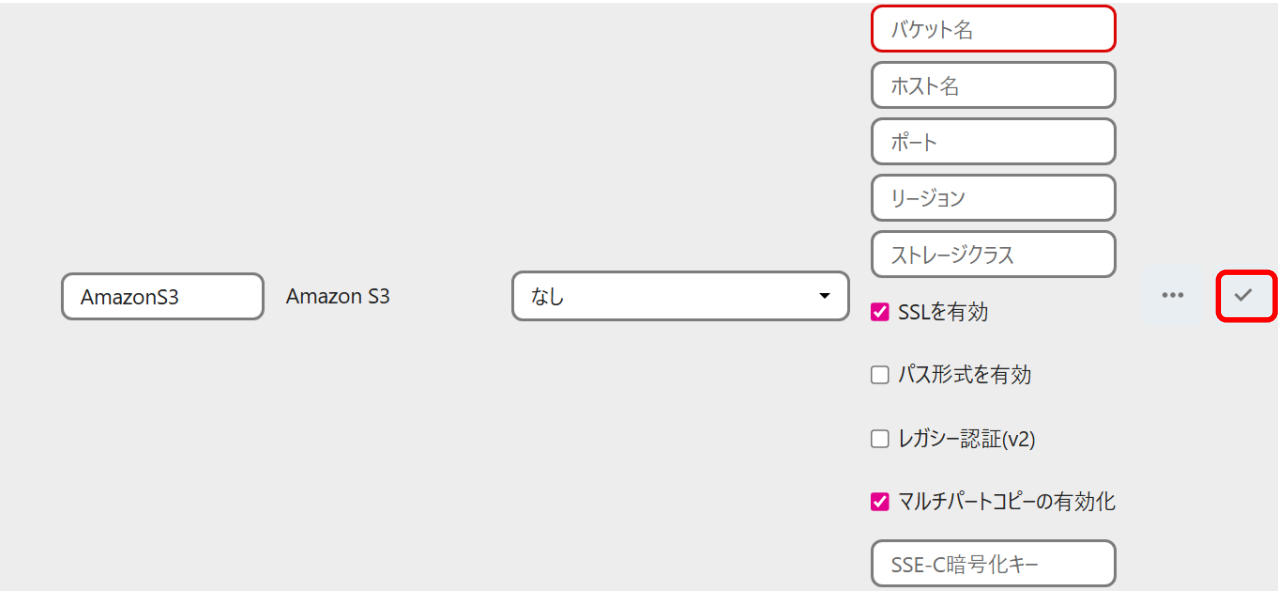
The following settings are displayed, and you need to enter the required information.



Click "..." on the right and check "Enable Sharing". (Not necessary if you don't want to share it)



Click "✓" on the far right after completing the settings.



7.1.5 How to use the app

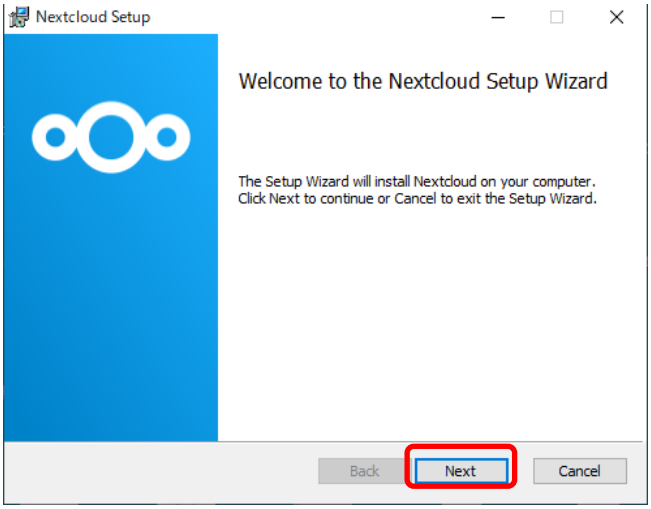
The Nextcloud environment is available with a desktop client app, which can also be used via the client app. The installation process of the client app involves a restart. When installing, save and run the file you are editing.

(1) install

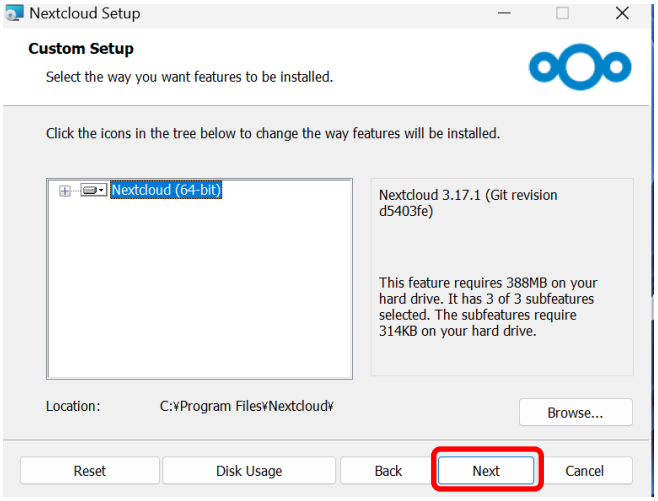
Download the installer from the following URL:

https://nextcloud.com/install/#install-clients

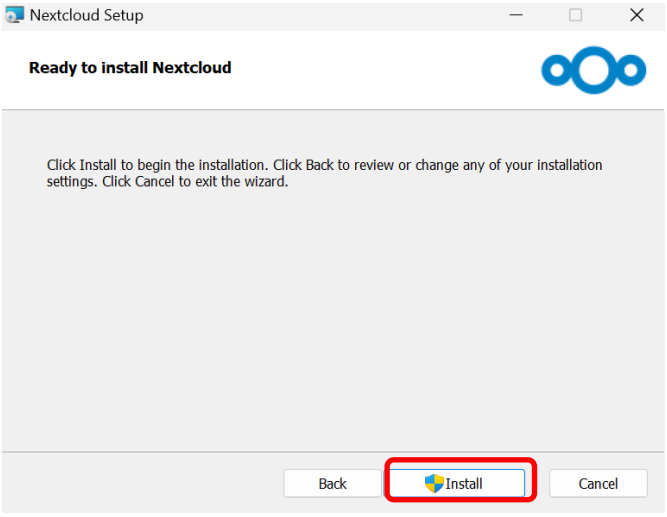
Run the downloaded installer.
The installation will start, click "Next".



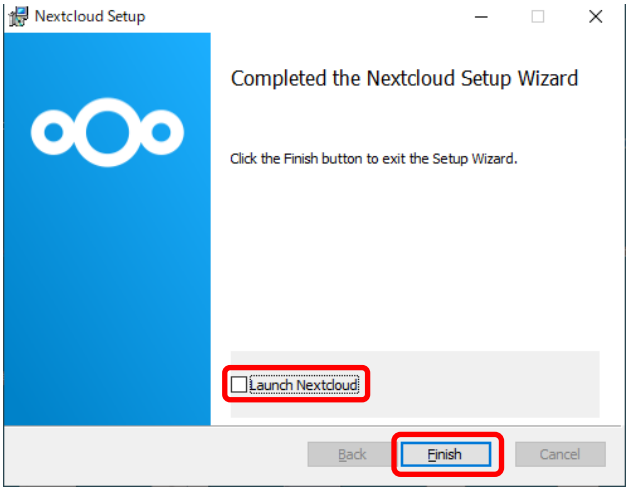
You don't need to change any settings, so click Next.



Click Install to perform the installation.



After the installation is complete, uncheck "Launch Nextcloud" and click "Finish".



(2) establishment

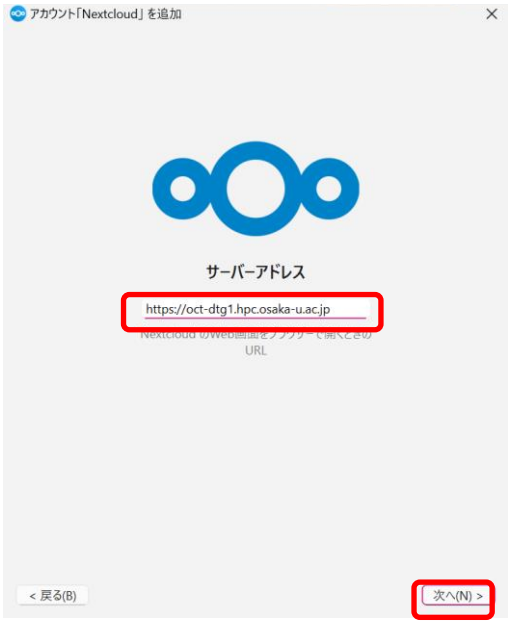
Nextckoud server to configure the connection.

After restarting your computer, Nextcloud will automatically start and the following image will

be displayed, click "Log in to your Nextcloud".

Enter the following URL in the Server Address and click Next.

https://oct-dtg1.hpc.osaka-u.ac.jp

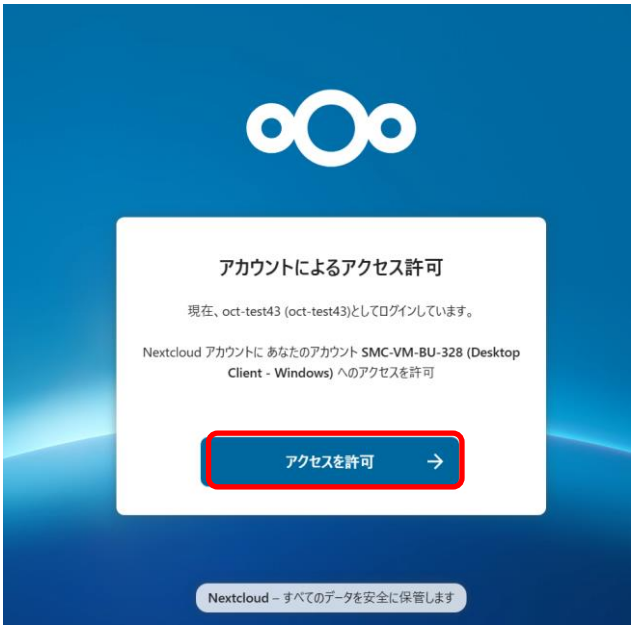


The screen below will automatically open the screen to allow authentication in a web browser.

The following screen will be displayed in your web browser, so click "Login".

The login screen below will appear, enter your username and password, and click "Login".

The following screen will appear, click "Allow Access".

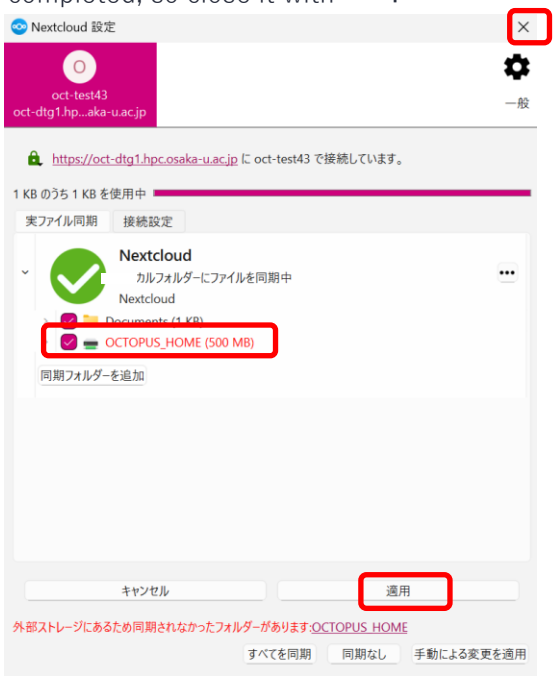


The following screen will appear, follow the on-screen instructions to close your web browser.

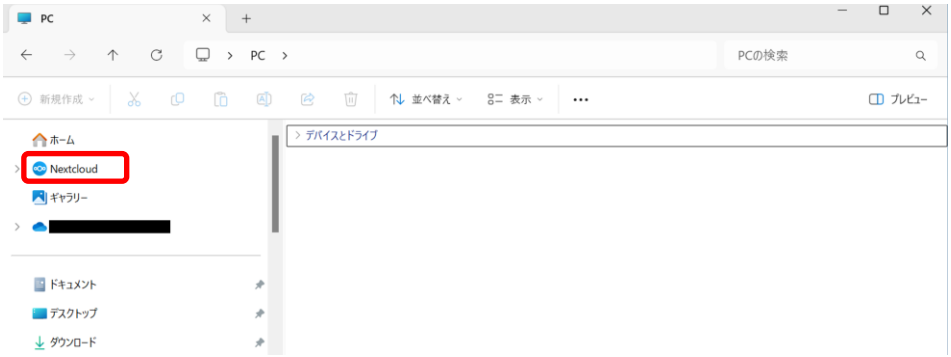
Return to the application and you will see the following screen, uncheck "Check folders larger than the specified capacity before syncing" and click "Connect..."

The Nextcloud settings screen will be displayed in the lower right corner of the screen, click on the username part, and then click "Settings" in the menu that appears.

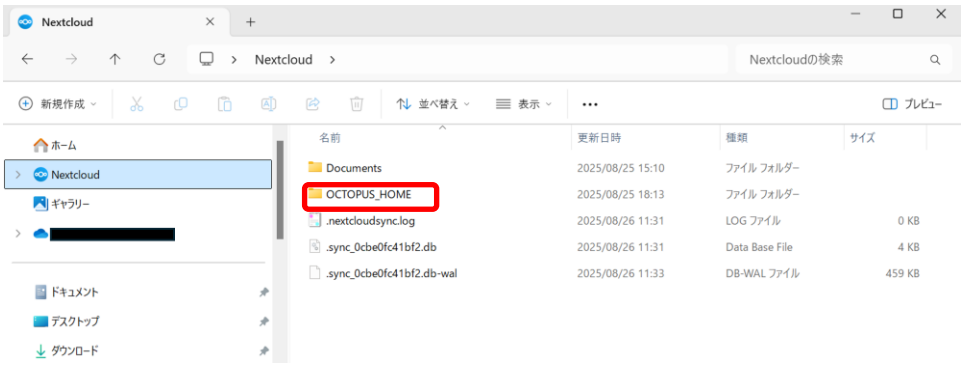
The following screen is displayed, check "OCTOPUS User Home Area" and click Apply. Wait for a while and the sync will be completed, so close it with "×".



When you open Explorer, click on "Nextcloud" on the left side.



「.」 Do not move or delete files with file names that begin with
Open the OCTOPUS User Home Area. You can read and write files.



7.2 File transfer from SQUID

In OCTOPUS and SQUID, the file system is a different system.
This section describes the procedure for transferring files between OCOTPUS ⇔ SQUID for those who also use SQUID.

This is how OCTOPUS and SQUID are using the same account to transfer files. The SQUID file system can be accessed from the OCTOPUS front-end server.

File System	File Path on OCTOPUS	File path when accessing from SQUID
OCTOPUS Home Area	/octfs/home	/octfs/home
Extended areas of OCTOPUS	/octfs/work	/octfs/work

File System	File path on SQUID	File path when accessing from OCTOPUS
SQUID's high-speed area	/sqfs/ssd	/sqfs/ssd
SQUID Home Area	/sqfs/home	/sqfs/home
SQUID Extension Areas	/sqfs/work	/sqfs/work

*You cannot access the SQUID file system from the OCTOPUS compute node, so please use it only for file transfer.

Here's how to transfer files: The user number is "A61234".

- OCTOPUS → SQUID (when working on the SQUID front-end) When transferring the file sample.c from the abc directory under the OCTOPUS home directory to the SQUID home directory

```
$ cp /octfs/home/a61234/abc/sample.c ~/
```

- SQUID → OCTOPUS (when working on the OCTOPUS front-end)
  If you want to transfer the file sample.c from the abc directory under the home directory of SQUID to the home directory of OCTOPUS

```
$ cp /sqfs/home/a61234/abc/sample.c ~/
```

8 Other Services

8.1 How to use the portal system for statistical information

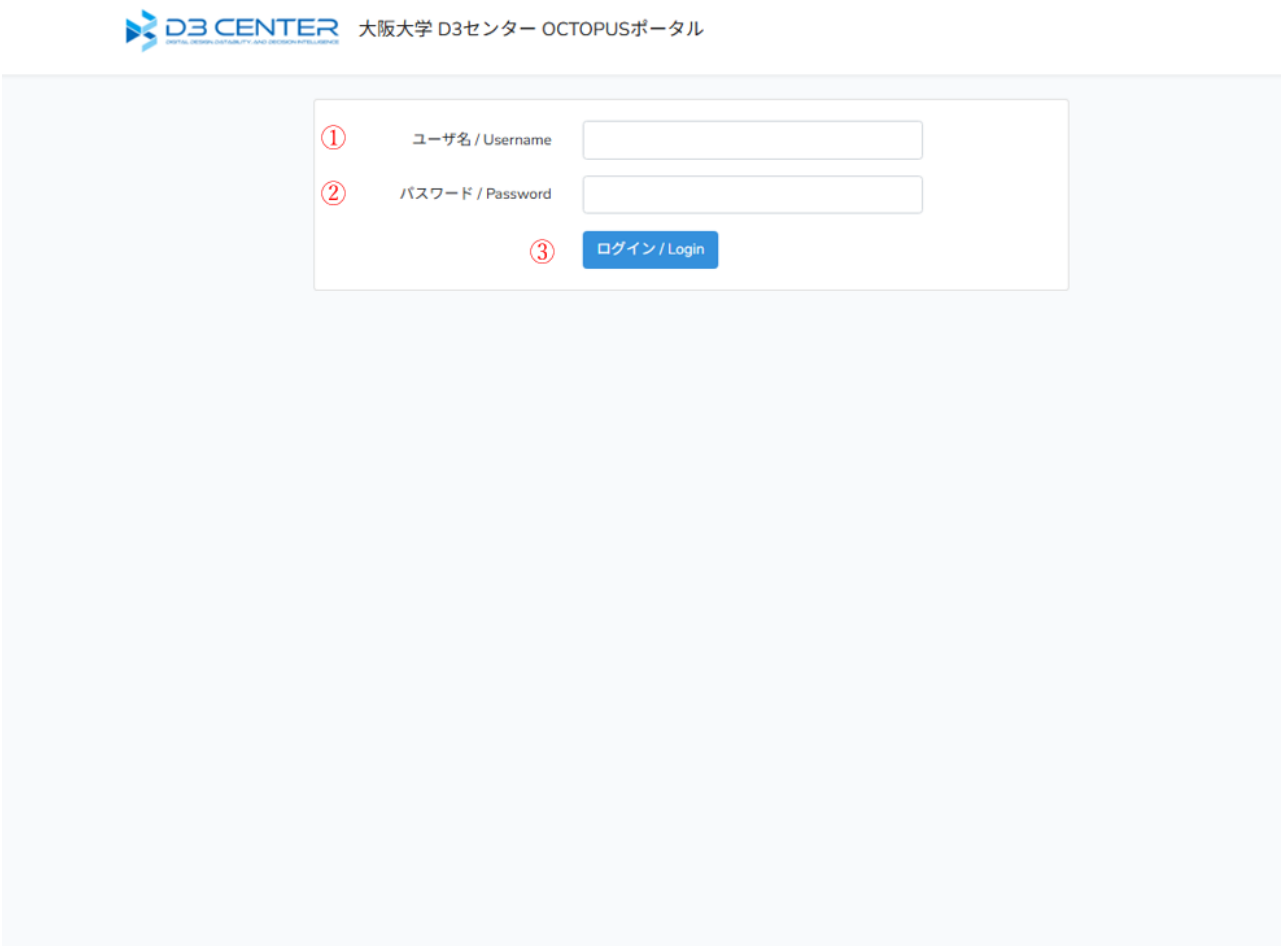
8.1.1 How to log in to the portal site

Logging in to the portal site also requires preparation for two-factor authentication in advance. 「2.1.1 Installing a two-step verification app」Please prepare a two-step verification app in advance.

※ The authentication code is separate for front-end login and portal login, so please set up two-factor authentication when you log in for the first time.

(1) Login screen

When you access https://octopusportal.hpc.osaka-u.ac.jp/portal/login, you will see the login screen of the portal system.

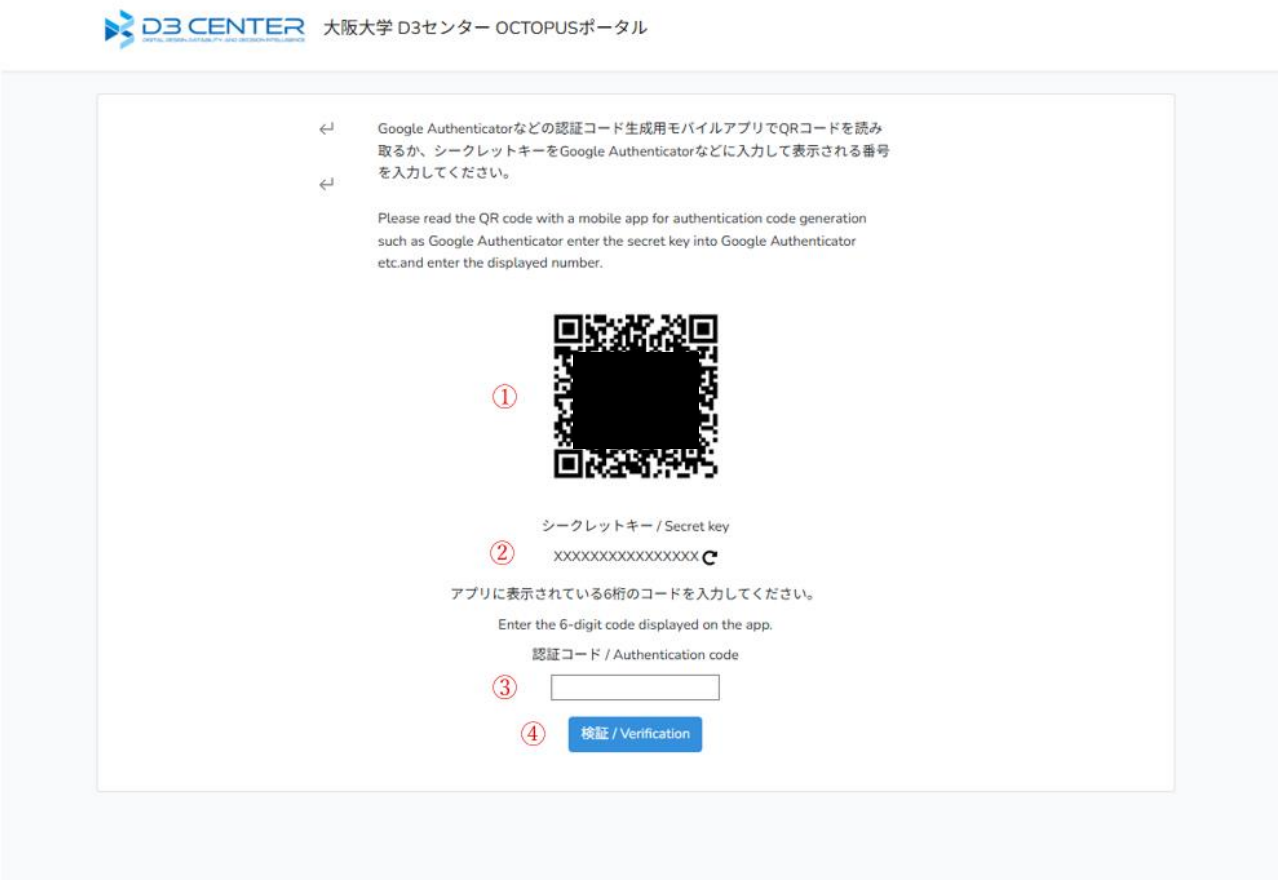


Login Instructions

① Please enter your user number in the "Username".
② Enter your login password in the "Password" field.
③ Click on the "Login" button.

(2) Two-step verification settings screen

If the logged-in user's two-factor authentication settings are not set, the login authentication will take you to the two-factor authentication settings screen.



◆ Two-Step Verification Setup Procedure

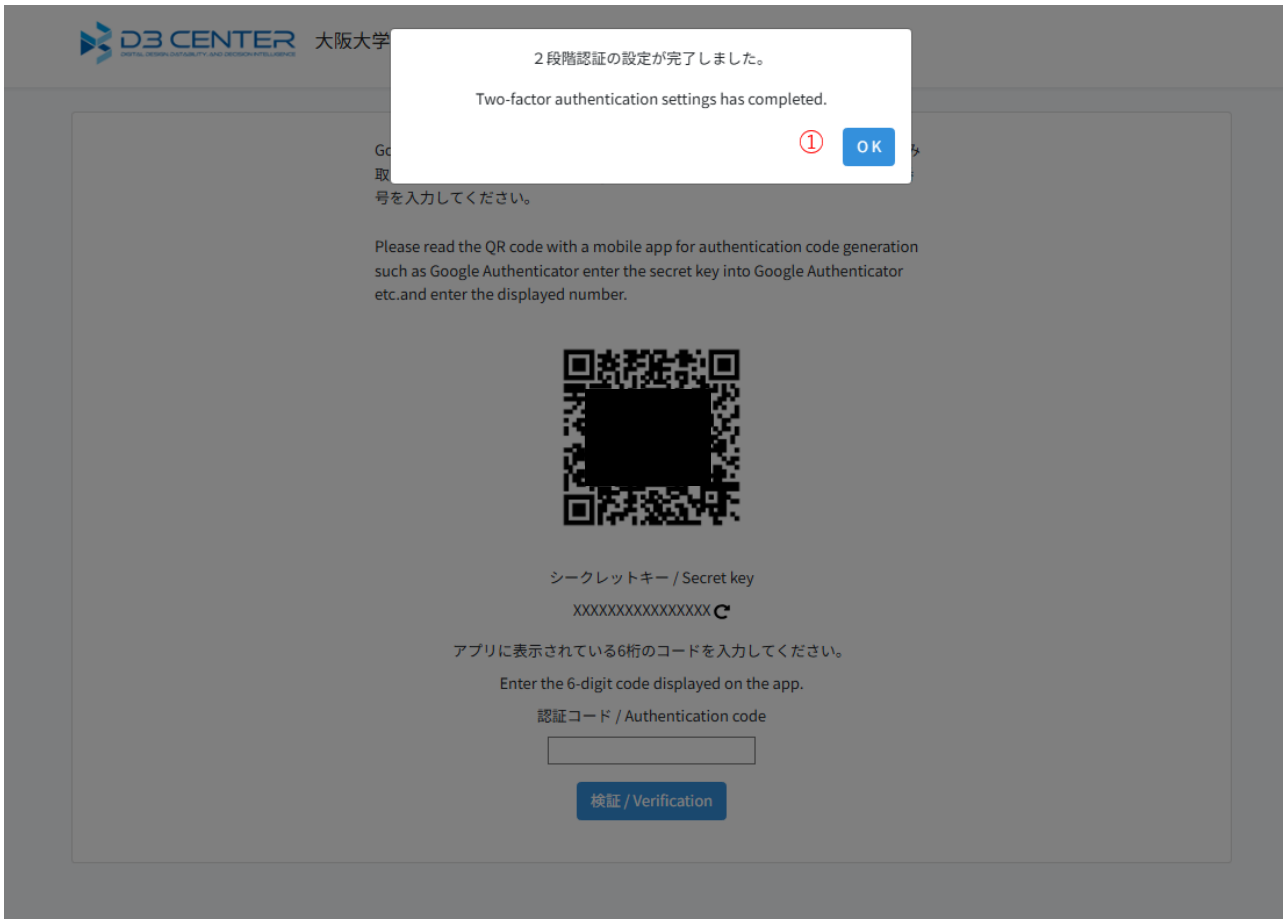
① Scan the QR code displayed in the two-step verification application.
② If you can't read the QR code, enter your secret key into the two-step verification application.
③ Enter the [Verification Code] displayed in the 2-Step Verification application into the "Authentication Code".
④ Click the "Verify" button.

◆ Completion dialog

Once the two-step verification setup is complete, a dialog will appear indicating that the process is complete.

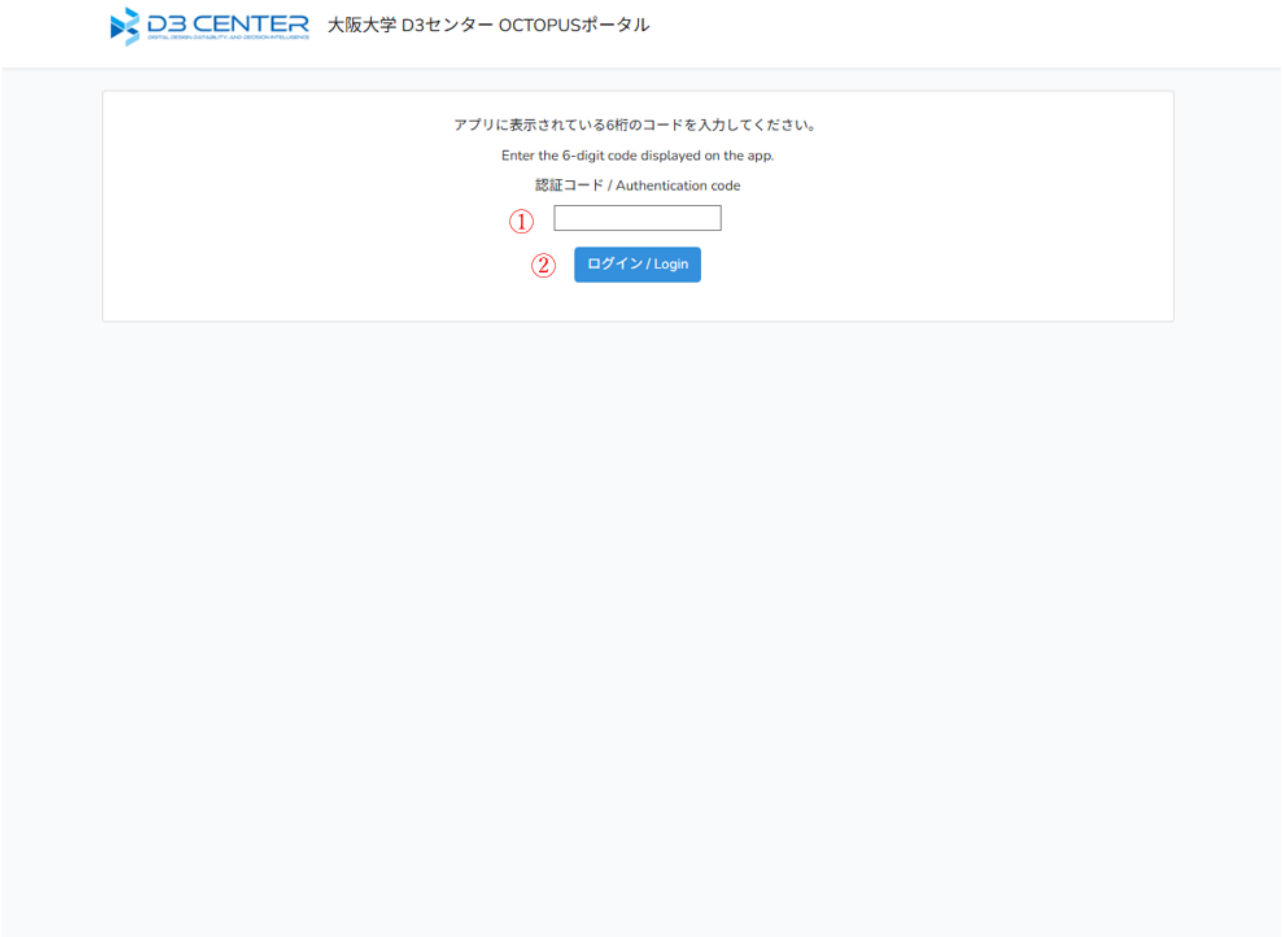
① OK button

Click the "OK" button to take you to the home screen.



(3) Two-step verification screen

If you have already set up two-step verification, you will be taken to the two-step verification screen after login authentication.



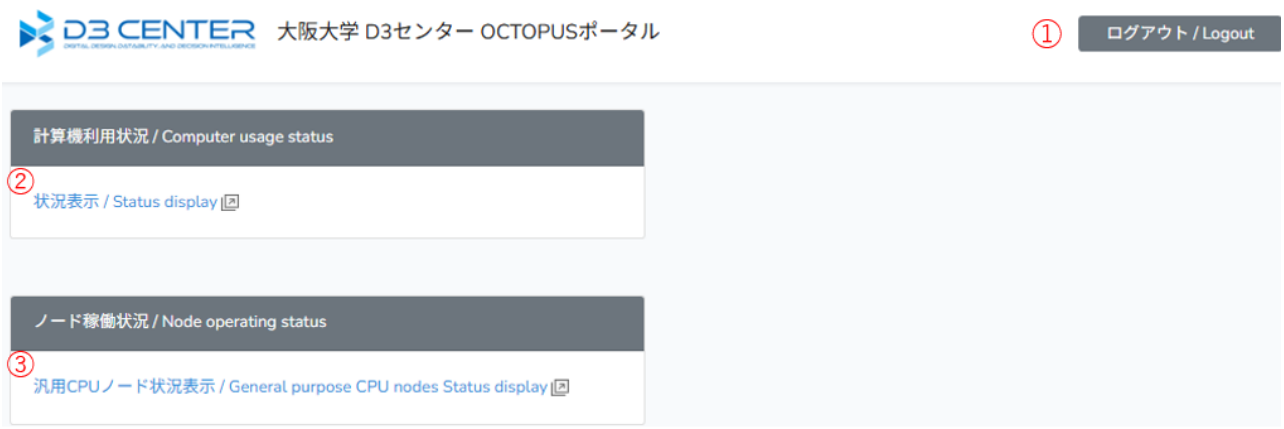
◆ Two-step verification procedure

① Enter the [Verification Code] displayed in the 2-Step Verification application into the "Verification Code".

② Click on the "Login" button.

(4) home screen

This is the home screen.



① Logout button

When you click the "Log Out" button, you will be redirected to the login screen after the logout process.

② Web display of calculator usage status

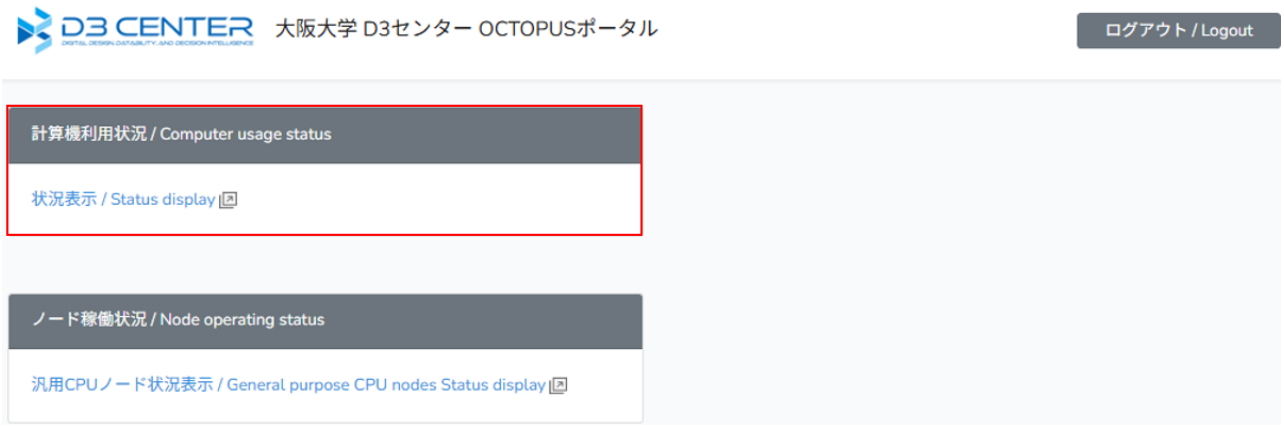
Select the "View Status" link to display the calculator usage status on the new tab screen.

③ Node Status Display

Select the General CPU Node Status Display link to display the node status on the new tab screen.

8.1.2 Web display of calculator usage status

On the home screen of the Portal System for Statistics, select the "Status Display" link of the Computer Usage Status to display the computer usage status on the new tab screen.



(1) Usage status search screen

Specify the year, period, group, and user for which you want to display the usage status.

◆ Search Procedure

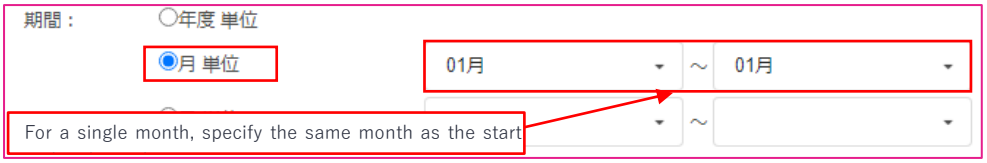
① Specify the year

Select the year for which you want to view usage. You can select from FY2025 ~ the current fiscal year.
However, you cannot select the year before the user registration date.
*If the registration date is 2026/4/1, you can choose from the 2026 academic year.
If the registration date is March 31, 2026, you can choose from the 2025 academic year.

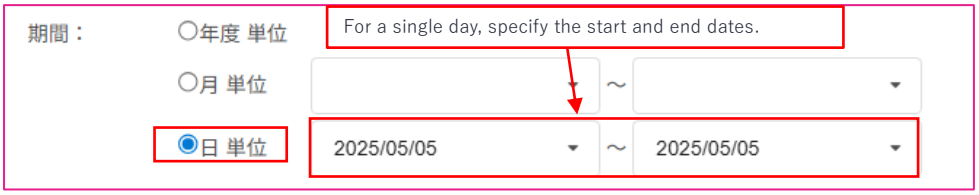
② Specify a time period

・If you want to display the total value of the annual unit, select "Year Unit".

・If you want to display it on a monthly basis, select "Monthly" and enter the start and end months.
Inputs made by only one of them will result in an error.



- If you want to display it on a daily basis, select "Daily" and enter the start and end dates.
Inputs made by only one of them will result in an error.

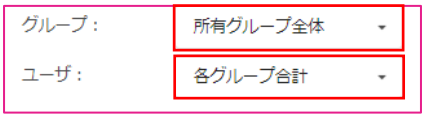


③ Designate groups and users

The data that can be selected varies depending on the logged-in user's permissions.

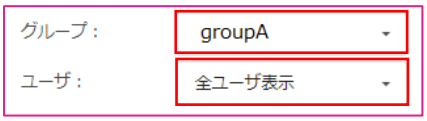
A) Application Representative

- If you want to display the total for each group for all groups for which you are the applicant representative



To the group, "Owning the entire group",
Assign Users Total for Each Group

- For a specific group for which you are the request representative, you want to view data for all members and the total group



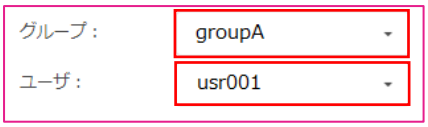
Specify any group and
Assign Users to "Show All Users"

- When to view data for specific groups and members

Specify any group and user

B) user

You can only view your data.



Pin your own group and user. You can't select it.

④ look up

Click the search button to display the search results screen.

*The screen will switch.

*If you want to change the search conditions, click the back button in the browser to redisplay the search screen.

(2) Search results screen

This is a screen for viewing information according to the specified conditions.

On this screen, you can click on the link of the information you want to view to display the search results.

◆ Procedure for viewing usage results

Click on the link for the information to view the details on a separate screen.

*By clicking on multiple links, you can display each information individually.

【Target Information】

A) Calculator Information

Node hours used, number of jobs put in

B) Information per occupancy/sharing

Usage node time (occupied, shared), number of input jobs (occupied, shared)

C) Queue-specific information

Node hours used, number of jobs put in

D) File System

Disk usage (Home, Extended Space)

(3) Usage Results Screen

Graphs and data are displayed based on the conditions specified on the previous screen.

*CSV output of display data is possible.

① Specify conditions for the fiscal year

The total value for the target year is displayed in graphs and data.

	2025
groupA	65.07
usr001	0.00
usr002	0.00
usr003	0.00
usr004	0.00
usr005	65.07
usr006	0.00

② Year-by-year CSV file image

Download the same content as the "Data Section" displayed on the screen in CSV format.

	A	B
1		2025
2	groupA	65.07
3	usr001	0
4	usr002	0
5	usr003	0
6	usr004	0
7	usr005	65.07
8	usr006	0

*Example of display in Excel

③ Specify conditions on a monthly and daily basis

Monthly (daily) trends are displayed in graphs and data.

	2024/04	2024/05	2024/06	2024/07	2024/08	2024/09	2024/10	2024/11	2024/12	2025/01	2025/02
usr001	0.00	0.00	0.00	8.00	8.00	8.00	8.00	8.00	8.00	21.00	22.00

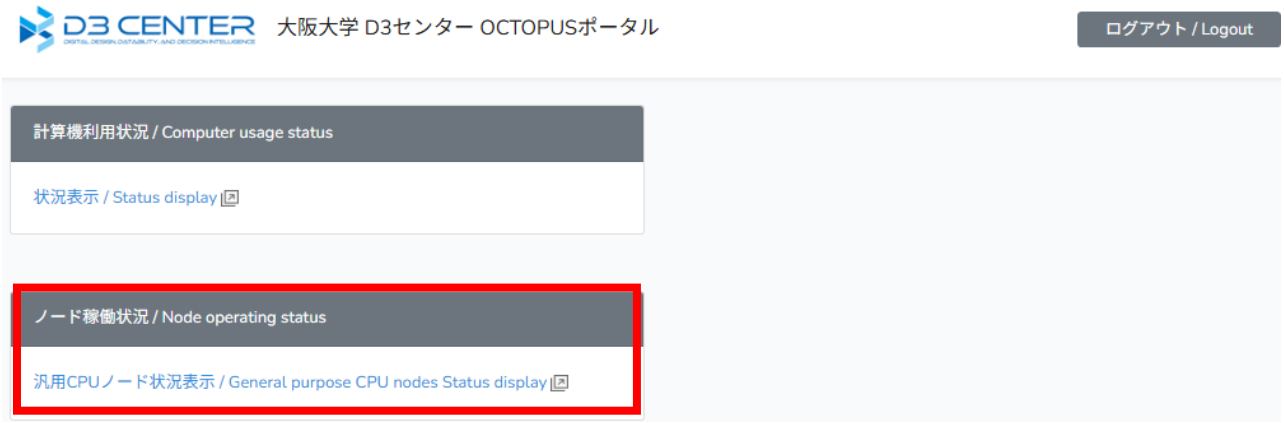
④ Monthly and daily CSV file images

Download the same content as the "Data Section" displayed on the screen in CSV format.

	A	B	C	D	E	F	G	H	I	J	K	L
1		2024/04	2024/05	2024/06	2024/07	2024/08	2024/09	2024/10	2024/11	2024/12	2025/01	2025/02
2	usr001	0	0	0	8	8	8	8	8	8	21	22

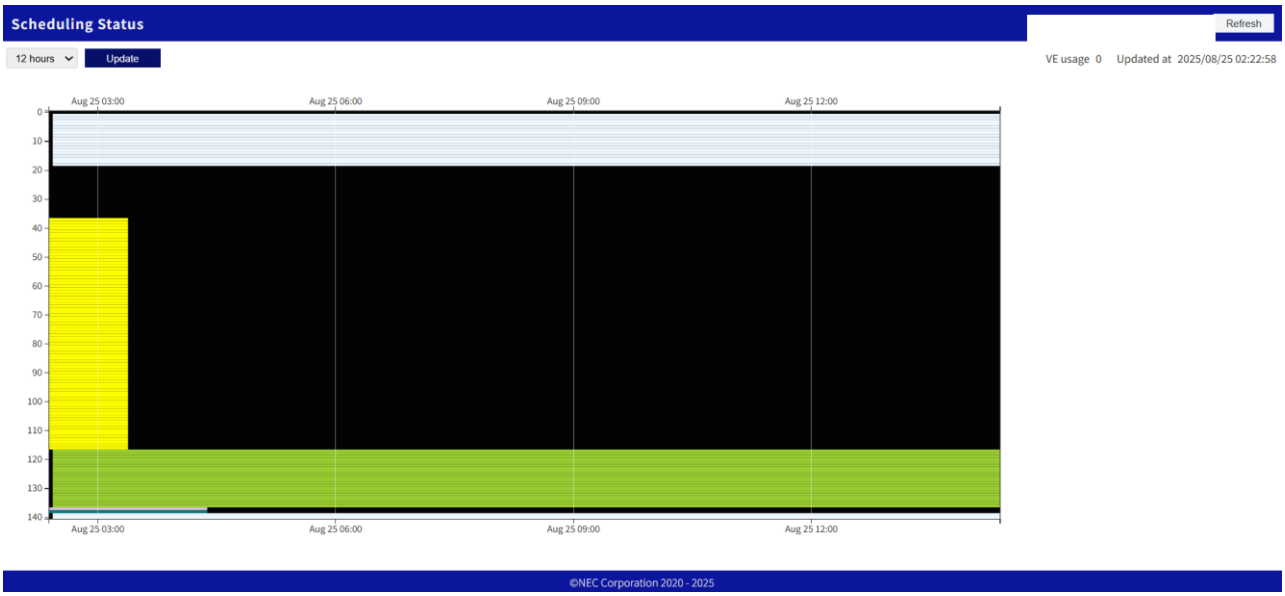
*Example of display in Excel

8.1.3 Node Operation Web View



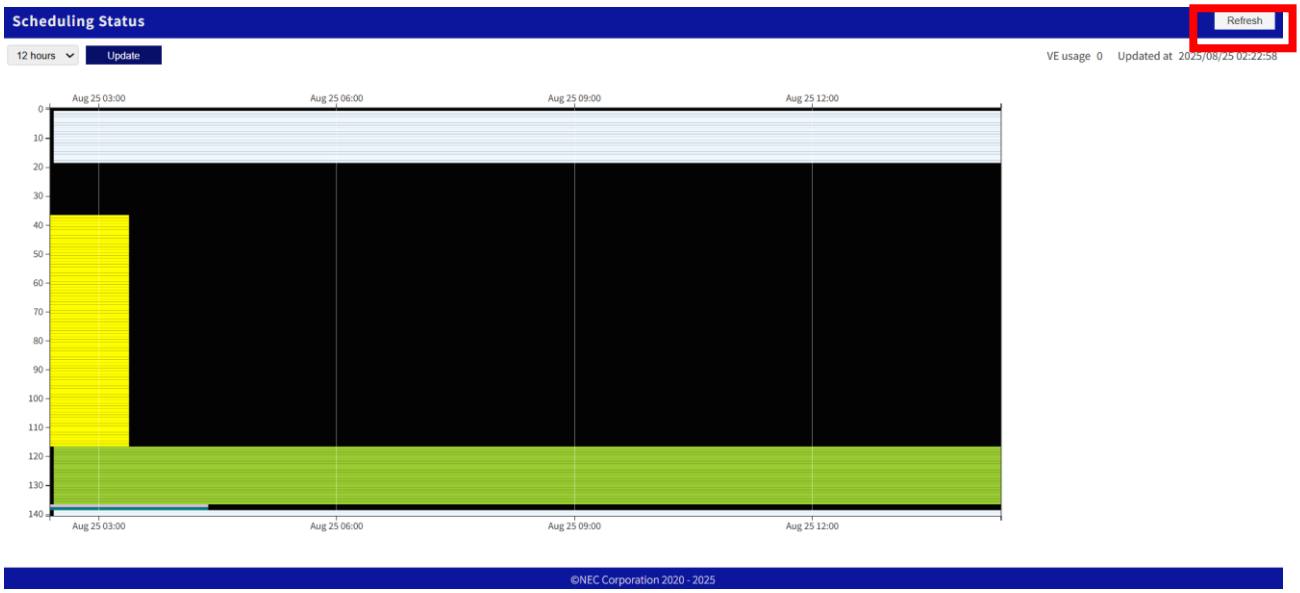
Select the link for General purpose CPU nodes status display.

After connecting to the General purpose CPU nodes status display, the following screen is displayed.
The vertical axis indicates the job server number, and the horizontal axis indicates the time.
The horizontal bars in the graph represent requests, and each request is assigned a color.



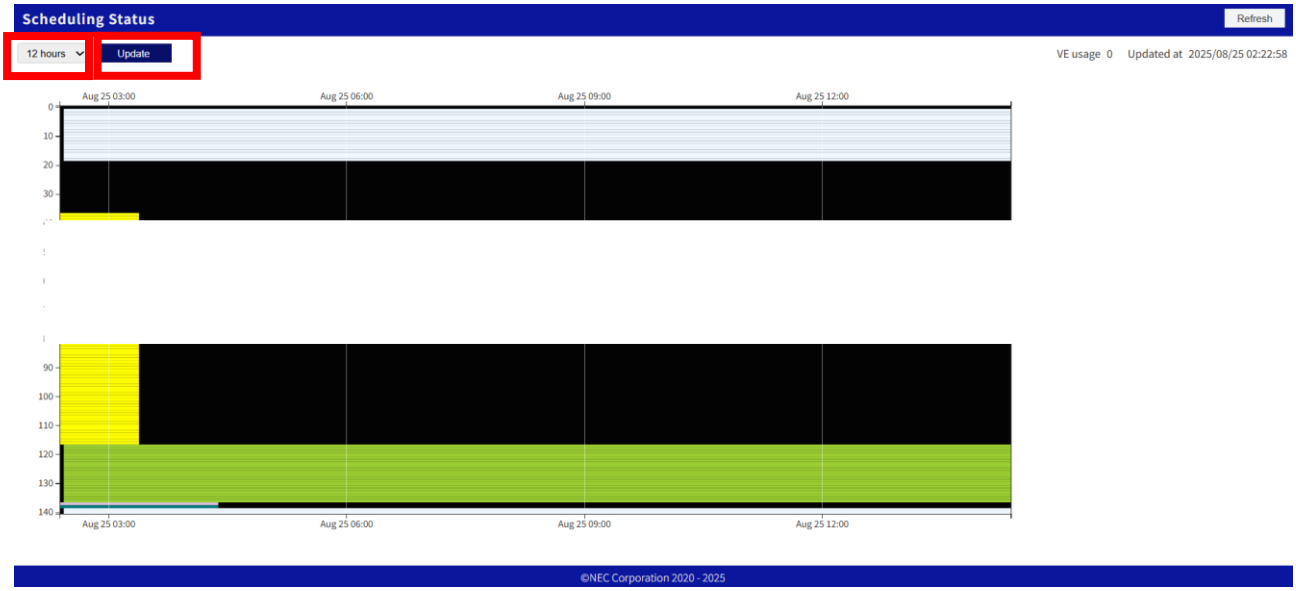
Hover over the horizontal bar in the graph to display the request ID.
When you press the horizontal bar, only the selected request is highlighted.

Press the "Refresh" button in the upper right corner of the screen to refresh the page with the latest node status.

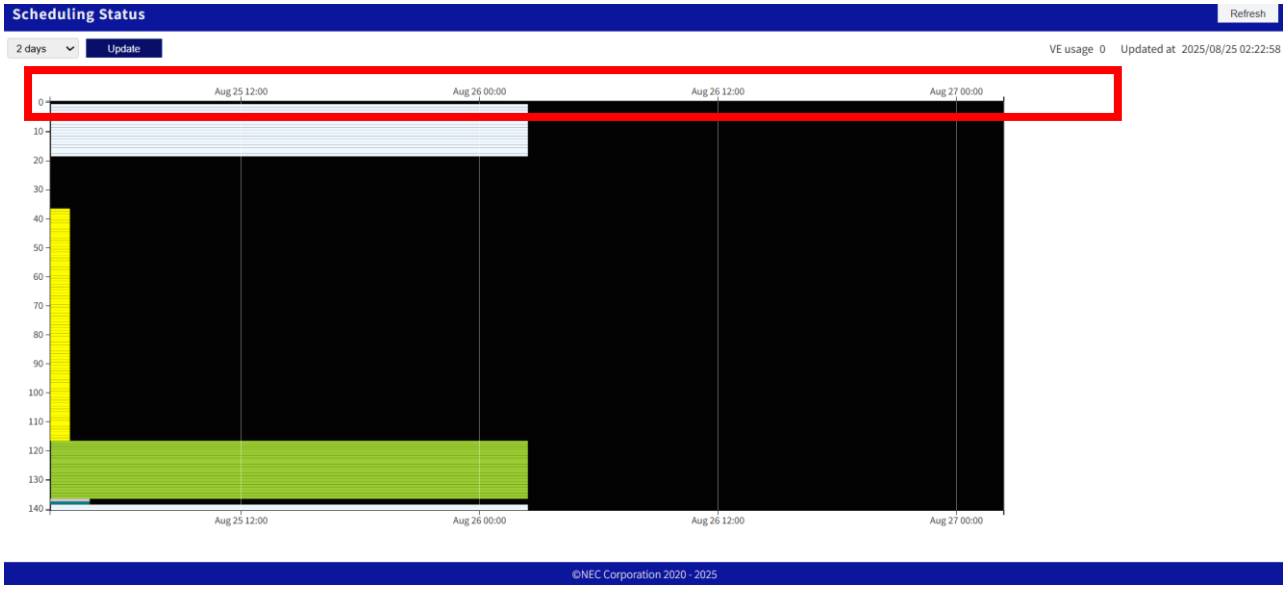


If you want to change the display period, select the number of days or hours you want to display from the drop-down in the top left of the screen.

After selecting any display period, press the "Update" button.



The display period changes.


Scheduling Status
Refresh
2 days
Update
VE usage 0
Updated at 2025/08/25 02:22:58
Aug 25 12:00
Aug 26 00:00
Aug 26 12:00
Aug 27 00:00
0
10
20
30
40
50
60
70
80
90
100
110
120
130
140
Aug 25 12:00
Aug 26 00:00
Aug 26 12:00
Aug 27 00:00
©NEC Corporation 2020 - 2025

more than