\Orchestrating a brighter world NEC

Management Number：OUCMC-super-技術-007-09

National University Corporation Osaka University

Cybermedia Center

User's manual

NEC

First Government & Public Solutions Division

Revision History

Ver	Date	Item	Overvie	Approval	Inspection	Create
1.0	2021/05/06	All	Create New	NEC	NEC	NEC
2.0	2021/05/26	1.2 1.4 1.5 3.1.3 3.3.4. 4.1.3. 5.2.2. 6.2.2. 6.2.4.	1.2 Add explanation of SW configuration 1.4 Add explanation of SW configuration 1.5 Add explanation of system usage environment 3.1.3 Add module command execution example 3.3.4. Add procedure for customizing jupyter 4.1.3. Fix how to start ddt and Parallel Debugger 5.2.2.Fix VEs upper bound of Vector job class 6.2.2. Add explanation of Gaussian 6.2.4. Add explanation of Gnuplot	NEC	NEC	NEC
3.0	2021/06/04	3.1.3 5.1.3 3.3.1 3.3.2 3.3.5 4.1.1 4.2.4 4.3.4 4.4.6	3.1.3 5.1.3 Delete description useing modules.sh 3.3.1 3.3.2 3.3.5 Modify the procedure in any directory. And add Stop Jupyer Container 4.1.1. Add other programming language environment 4.2.4 4.3.4 4.4.6 Add how to use the library in each environment	NEC	NEC	NEC

		4.5 4.7 6.2.5	4.5 Add How to User GNU Compiler Collection 4.7 Add How to Use Python 6.2.5 modify sample script to use GROMACS			
4.0	2021/06/29	5.1.3 5.1.4 6.1 6.2 6.3 6.4	5.1.3 (2) Add of batch job request to job management system 5.1.4 (5),(6) Add delete batch job, queue status check 6.1 Classify applications by type 6.2 Add ISV Applications 6.3 Add OSS Applications 6.4 National Project Applications	NEC	NEC	NEC
5.0	2021/8/16	1.5.4 4.4.6 4.6 4.7 4.7.4 5.6 6.3.3 6.3.5 6.3.10	1.5.4 Enhanced explanation about containers 4.4.6 Modify the procedure of HDF5 library for GPU node 4.6 Modify the flow of container usage. ex. image without using sandbox, image from NGC 4.7 Added how to use Python3 for GPU 4.7.4 Added library usage (TensorFlow, Keras, Pytorch) 5.6 Added how to run a container 6.3.3 Added how to use GAMESS 6.3.5 Added how to use GROMACS 6.3.10 Added how to use Quantum ESPRESSO	NEC	NEC	NEC
6.0	2021/9/38	5.4.3 5.4.4 6.3.10	5.4.3 5.4.4 6.3.10 Added venode option when executing mpirun of NEC MPI	NEC	NEC	NEC

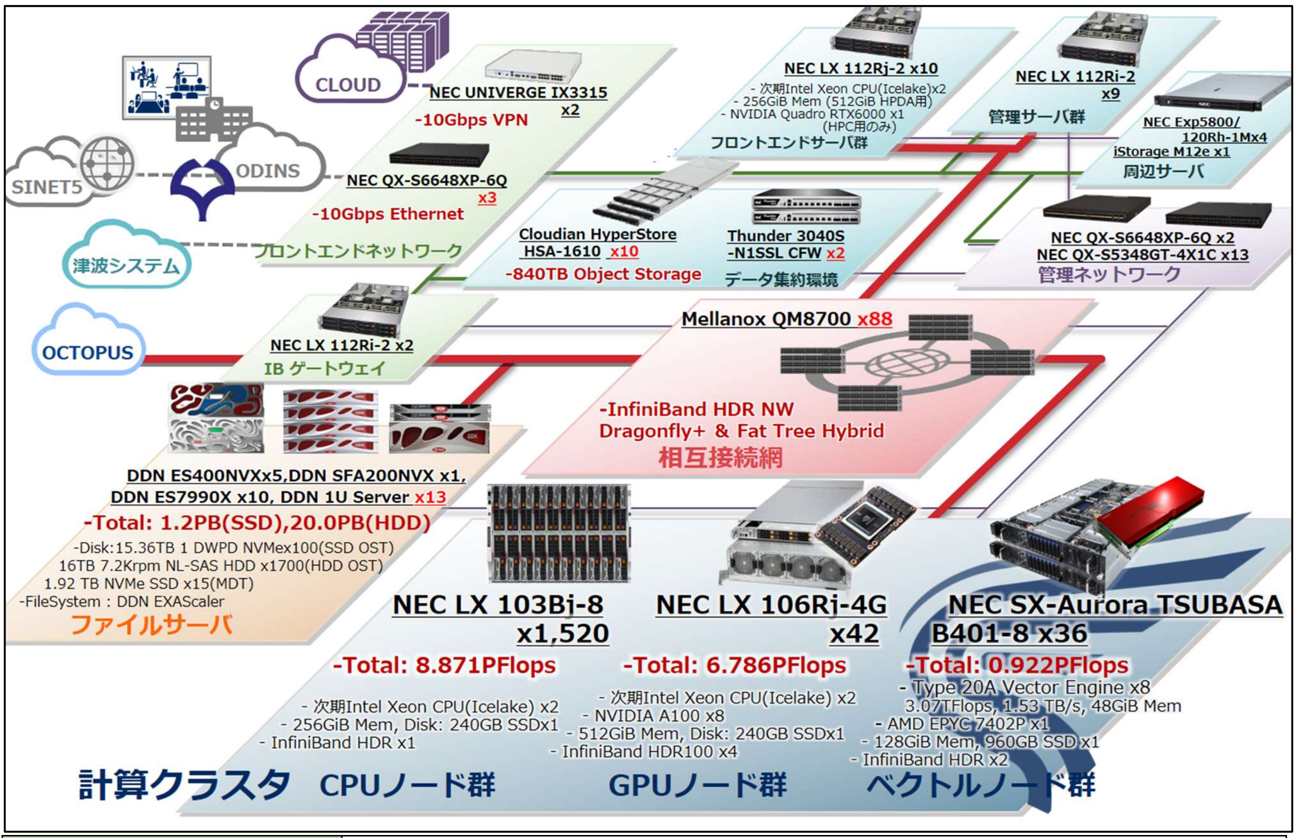
		5.6.1 6.1 6.3.11 7.2	5.6.1 Fixed a typo in the procedure to run a.out on the host 6.1 Added PHASE/0 to application list 6.3.11 Added PHASE/0 usage Revised due to review of bucket operation -Added s3dsbucket command -Added bucket creation / synchronization procedure			
7.0	2021/11/23	4.6.2	4.6.2 Revised create and delete sandbox procedure ・add (4) Delete sandbox	NEC	NEC	NEC
8.0	2021/12/16	1.1 1.3.1 1.3.2 7.4 8.1.1	ONION Expansion Project	NEC	NEC	NEC
9.0	2022/05/18	5 5.3.5 5.5.5 8.2.1 8.2.2 8.2.3	5 If the elapse time is not specified in the job script, set it to 1 hour 5.3.5 Added a manual specification method for the number of processes 5.5.5 Describes how to allocate CPU and GPU for each RANK 8.2.1 8.2.2 Replacement of home screen image 8.2.3 Added Node operatinon status Web	NEC	NEC	NEC

Summary

1. System Overview

1.1. Overall System View

"SQUID (**S**upercomputer for **Q**uest to **U**nsolved **I**nterdisciplinary **D**atascience)" is a supercomputer system that consists of general-purpose CPU nodes, Vector nodes, GPU nodes, and large-capacity File Servers, and has a total theoretical computing performance of 16.591 PFLOPS.


CLOUD
NEC UNIVERGE IX3315 x2
-10Gbps VPN
SINET5
ODINS
NEC QX-S6648XP-6Q x3
-10Gbps Ethernet
津波システム
フロントエンドネットワーク
NEC LX 112Rj-2 x10
- 次期Intel Xeon CPU(Icelake)x2
- 256GiB Mem (512GiB HPDA用)
- NVIDIA Quadro RTX6000 x1 (HPC用のみ)
フロントエンドサーバ群
NEC LX 112Ri-2 x9
管理サーバ群
NEC Exp5800/ 120Rh-1Mx4
iStorage M12e x1
周辺サーバ
Cloudian HyperStore HSA-1610 x10
-840TB Object Storage
Thunder 3040S -N1SSL CFW x2
データ集約環境
NEC QX-S6648XP-6Q x2
NEC QX-S5348GT-4X1C x13
管理ネットワーク
OCTOPUS
NEC LX 112Ri-2 x2
IB ゲートウェイ
Mellanox QM8700 x88
-InfiniBand HDR NW
Dragonfly+ & Fat Tree Hybrid
相互接続網
DDN ES400NVXx5,DDN SFA200NVX x1,
DDN ES7990X x10, DDN 1U Server x13
-Total: 1.2PB(SSD),20.0PB(HDD)
-Disk:15.36TB 1 DWPD NVMex100(SSD OST)
16TB 7.2Krpm NL-SAS HDD x1700(HDD OST)
1.92 TB NVMe SSD x15(MDT)
-FileSystem : DDN EXAScaler
ファイルサーバ
NEC LX 103Bj-8 x1,520
-Total: 8.871PFlops
- 次期Intel Xeon CPU(Icelake) x2
- 256GiB Mem, Disk: 240GB SSDx1
- InfiniBand HDR x1
NEC LX 106Rj-4G x42
-Total: 6.786PFlops
- 次期Intel Xeon CPU(Icelake) x2
- NVIDIA A100 x8
- 512GiB Mem, Disk: 240GB SSDx1
- InfiniBand HDR100 x4
NEC SX-Aurora TSUBASA B401-8 x36
-Total: 0.922PFlops
- Type 20A Vector Engine x8
3.07TFlops, 1.53 TB/s, 48GiB Mem
- AMD EPYC 7402P x1
- 128GiB Mem, 960GB SSD x1
- InfiniBand HDR x2
計算クラスタ
CPUノード群
GPUノード群
ベクトルノード群

Total computing performance	16.591 PFLOPS	
Node configuration	General-purpose CPU node cluster 1520 nodes（8.871 PFlops）	Intel® Xeon® Platinum 8368 (2.4GHz/38core) x2
	Vector node cluster 36 nodes（0.922PFlops）	Type20A Vector Engine (3.07 TFlops,1.53TB/s) x8 AMD EPYC™ 7402P (2.8GHz/24core) x1
	GPU node cluster 42 nodes（6.797PFlops）	Intel® Xeon® Platinum 8368 (2.4GHz/38core) x2 NVIDIA A100 GPU(9.7TFlops) x8
Interconnect network	InfiniBand HDR (200 Gbps)	

File server	DDN EXAScaler (SSD area 1.2 PB / HDD area 20.PB)

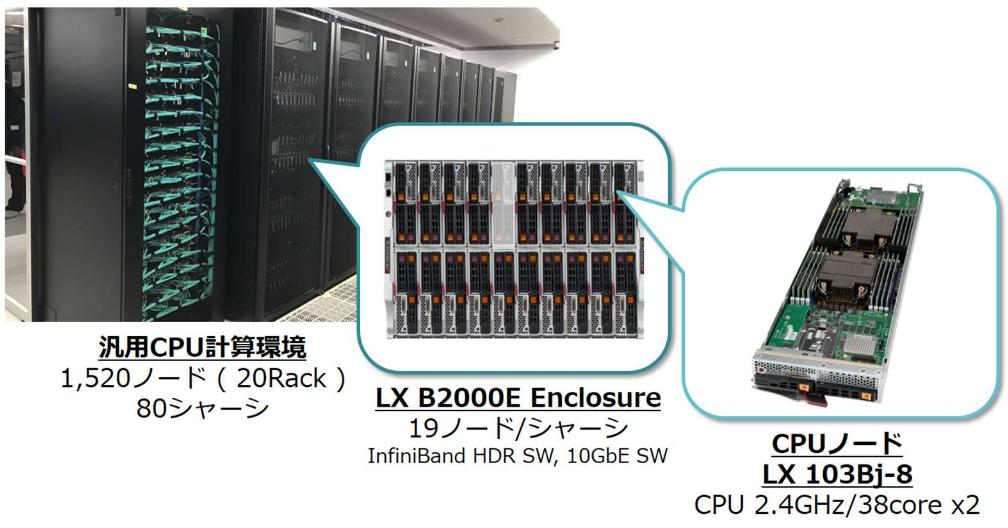
1.2. Thress types of Computiong Environments

SQUID has three types of computing environment, and the specification for each enviroment are as below.

Computational environment	General-purpose CPU environment	Vector computing environment	GPGPU computing environment
Use	General purpose computating For large-scale parallel execution	For high memory bandwidth / vector calculation by using vector engine	For high-speed calculation by GPGPU accelerator
Number of nodes	1,520 nodes	288 VE	42 nodes
Arithmetic	Intel® Xeon® Platinum 8368 (2.4GHz/28Core)x2	Type 20A Vector Engine (3.07 TFlops,1.53TB/s) x8	Intel® Xeon® Platinum 8368 (2.4GHz/28Core)x2
Accelerator	-	-	NVIDIA A100 x8
Memory capacity	256GiB	48GiB (HBM2)	512GiB
Interconnect network	InfiniBand HDR x1	InfiniBand HDR x2	InfiniBand HDR100 x4

1.2.1. General-purpose CPU environment

The general-purpose CPU node cluster consists of 1,520 nodes with blade system called "NEC LX 103Bj-8". The chassis on blade system has 19nodes, and 4 chassis are installed on the rack.



・Configuration for General-purpose CPU Node

Item		Configuration
Number of nodes		1,520 nodes
Node configuration	Processor	Intel® Xeon® Platinum 8368 (2.4 GHz/38core)　x2
	Memory configuration	256 GiB (16GiB DDR4-3200 RDIMM x16)
	hard disk	240GB SATA SSD x1
	interface	InfiniBand HDR x1、25/10GbE x2、BMCx1
Software environment	OS	CentOS 8.2 (64bit)
	compiler	Intel Compiler
	MPI	Intel MPI

・Software configuration

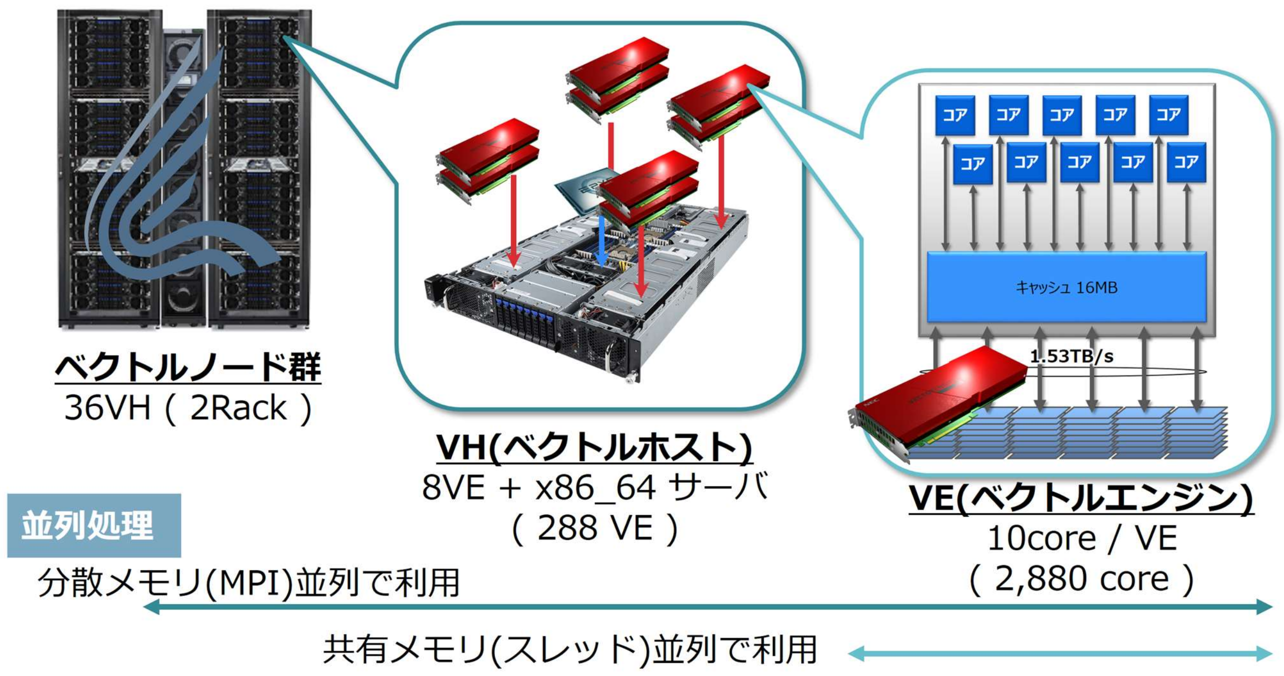
Item	Configuration
OS	CentOS 8.2 (64bit)
Compiler	Intel Parallel Studio
MPI	Intel MPI

As a software environment, Intel Parallel Studio, which is a highly optimized C / C ++ / FORTRAN development suite for Intel processors, can be used. The Intel MPI included in the Intel Parallel Studio is the standard MPI.

1.2.2. Vector computing environment

The vector node cluster consists of "NEC SX-Aurora TSUBASA B401-8" 288 VE (vector engine). "NEC SX-Aurora TSUBASA B401-8” has 8VEs on VH of x86_64 server, and Vector

Node Group consists of 36VH.



・ **Specifications for VE(Vector Engine)**

Item		Description
Total number of VE		288 VE (8VE per VH)
VE configuration	Model name	Type 20A
	Computational performance (double precision)	307 GFlops / 10core
	memory	48 GiB HBM2

・**Configurations for VH(Vector Host)**

Item		Configuration
Total number of VH		36 VH
VH configuration	Processor	AMD EPYC 7402P Processor(2.8GHz/24core) x1
	Memory configuration	128 GiB
	interface	InfiniBand HDR x2, 1000BASE-Tx1, BMCx1

・**Software configuration**

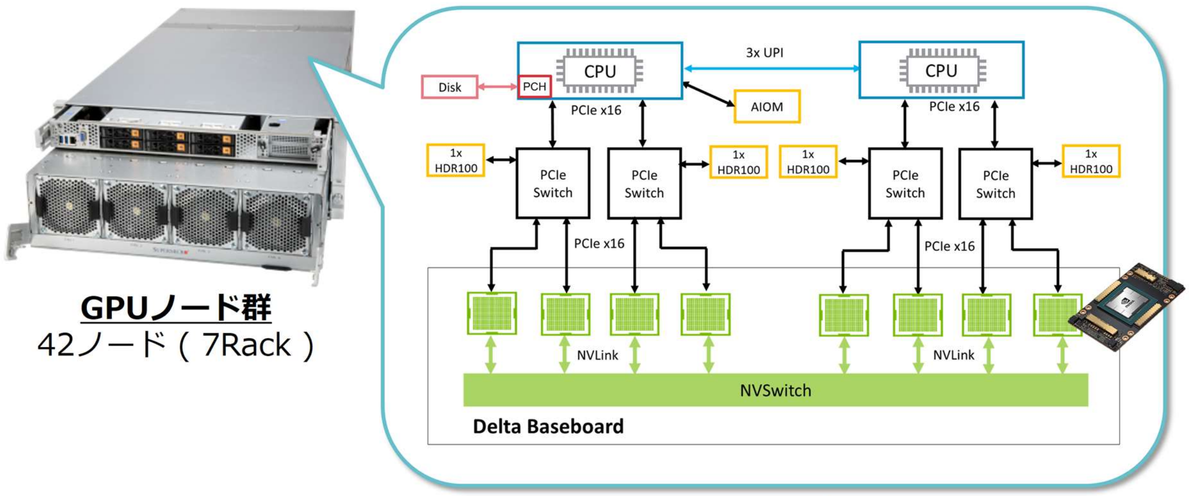
Item	Configuration
OS	CentOS 8.2 (64bit) + VEOS 2.7.6

Compiler	NEC SDK for Vector Engine
MPI	NEC MPI

As a software environment, NEC SDK for Vector Engine can be used. It is a development suite for C / C ++ / FORTRAN programs that can be executed on the Vector Engine. And NEC MPI is the standard MPI, which enables MPI communication directly from the program on the vector engine.

1.2.3. GPGPU computing environment

The GPU node cluster has 42 nodes of "NEC LX 106Rj-4G". It has two "Intel® Xeon® Platinum 8368" as the CPU and eight "NVIDIA A100" as the GPU on the 4U chassis. The eight GPUs are connected through NVSwitch on the Delta Baseboard.



GPUノード群
42ノード（7Rack）

GPUノード
CPU 2.4GHz/38core x2
GPU 9.7TFlops x8 + NVSwitch

・**Configuration for GPU node**

Item		Configuration
Number of nodes		42nodes
Server configuration	Processor	Intel® Xeon® Platinum 8368(2.4 GHz/38 core)　x2
	Memory configuration	512 GiB (32GiB DDR4-3200 ECC RDIMM x16)
	hard disk	240GB SATA SSD x1
	interface	InfiniBand HDR100 x4, 1000BASE-T x1, BMC x1
	GPGPU	NVIDIA A100 (SXM4) x8

・**Software configuration**

Item	Configuration
OS	CentOS 8.2 (64bit)
Compiler	NVIDIA HPC SDK
MPI	OpenMPI

As the software environment, NVIDIA HPC SDK can be used as a compiler of C / C ++ / FFORTRAN programs, and it includes CUDA , which is a program development tool for GPGPU. In addition, OpenMPI, which has a high affinity with the compiler, is used as the standard MPI.

1.3. Three types of Storage Areas

1.3.1. Features

Three types of storage are available on SQUID. The features of each storage are as below.

Storage area	SSD storage	HDD storage	Archive storage
Features	Depends on the All Flash environment, Ultra-high-speed IO area	High speed and large capacity, Data storage area	Flexible archiving area with replication and encryption capabilities
capacity	Additional purchase	home：10GiB work：5TiB + Additional purchase	Individual application
File system	DDN ExaScaler (Lustre)	DDN ExaScaler (Lustre)	Cloudian HyperStore
Total capacity	1.2 PB	20PB	1200TB(Bare metal)
Disk device	15.36TB NVMe SSD	16TB 7,200rpm NL-SAS	960GB SSD(Meta-data) 10TB SAS
hardware	DDN ES400NVX x5	DDN ES7990X x10	Cloudian HyperStore Appliance 1610 x10

＊ SSD and HDD storage can be used directly from the computiong environment.

1.3.2. Various Data Access Methods

SQUID provides various data access methods for the data utilization improvement. The reference destinations for each access method are shown below.

how to access	protocol	Use	Reference
High speed parallel access	Lustre	Fast and parallel access in the system	3.1.1 How to Use the File System
CLI access	SCP/SFTP	Data transfer method to the outside of the system by CLI	2.3 File Transfer
Browser access	HTTPS	How to transfer data to the outside of the system using a web browser	7.1 File Transfer with Web Browser
S3 access	S3	Data transfer method from external application by S3 API	7.2 File Transfer with S3 API
Inter-system access	NFS	How to exchange data with peripheral systems (OCTOPUS)	7.3 File Transfer from OCTOPUS
CIFS access	CIFS	Data linkage within ODINS	7.4 File Transfer with CIFS

1.4. Three types of Front-end nodes

1.4.1. Features

Three types of front-end nodes are available on SQUID. The features are as below.

environment	HPC front-end	HPDA front-end	Secure front-end
features	Environment for HPC application development. Visualization processing , batch job submission function	Notebook environment for HPDA. Batch job submission function	Dedicated front-end node with virtual machines. Batch job submission function
number of nodes	4 nodes	4 nodes	vriable by individual application
application	NICE DCV Server	Jupyter	
Remarks	NVIDIA Quadro RTX600 x1	512GiB memory (Twice as HPC)	Individual application

1.4.2. HPC Front-end

The HPC front-end consists of four "NEC LX 112R j-2" nodes. It has "Intel® Xeon®

Platinum" as a CPU, and can use graphic accelerators for remote visualization processing, visualization applications, etc.

・**HPC Frontend**

Item		Configuration
Number of nodes		4 nodes
Server configuration	Processor	Intel® Xeon® Platinum 8368(2.4 GHz/38 core)　x2
	Memory configuration	256 GiB (16GiB DDR4-2933 RDIMM x16)
	hard disk	960GB SSD x2
	interface	InfiniBand HDR x1、 10GBASE-T x2、1000BASE-T x1、IPMI2.0 x1
	GPU	NVIDIA Quadro RTX6000 x1

・**Software configuration**

Item	Configuration
OS	CentOS 8.2 (64bit)
Job Management	NEC NQSV
Remote visualization application	NICE DCV Server

As a software environment, the NEC NQSV can be used as the job management system. You can also use NICE DCV Server, which allows you to remotely display the desktop screen on the HPC frontend and perform visualization processing using the GPU on the server.

Refer to "5 Program Execution Method" for the job submission procdedure and "3.2 How to use the HPC Front-end" for how to use the remote visualization application.

1.4.1. HPDA Front-end

The HPDA front-end consists of four "NEC LX 112R j-2" nodes. It has two “Intel® Xeon® Platinum 8368” processors, and will become various data analysis environment by providing a user customizable NoteBook environment

・**HPDA Front-end**

Item		Configuration
Number of nodes		4 nodes
Server configuration	Processor	Intel® Xeon® Platinum 8368(2.4 GHz/38 core)　x2
	Memory configuration	512 GiB (16GiB DDR4-2933 RDIMM x16)
	hard disk	1TB SAS HDD x2
	interface	InfiniBand HDR x1、10GBASE-T x2、1000BASE-T x1、IPMI2.0 x1
Software environment	OS	CentOS 8.2 (64bit)
	NoteBook	Jupyter Notebook

・**Software configuration**

Item	Configuration
OS	CentOS 8.2 (64bit)
Job Management	NEC NQSV
Notebook	Jupyter Notebook

As for the software environment, the NEC NQSV environment can be used as the job managemnt system. You can also use Jupyter Notebook, which provides an interactive program development and execution environment on a web browser.

Refer to "5 Program Execution Method" for the job submission procdedure and "3.3 How to use the HPDA Front-end" for how to use the notebook.

1.5. 1.5. System usage environment

1.5.1. Login method

This system uses two-factor authentication when logging in to ssh as part of strengthening security. Before logging in, you need to prepare the app required for 2-factor authentication and pre-install it on your device.

For the detail of login procedure, refer to "2 Instructions for Front-end Login".

1.5.2. Program development

In this system, you can use compilers and libraries for developing various programs including C / C ++ / FORTRAN language. You can also use various pre-built applications.

Environment Modules manage the environment variable settings associated with the use of these software. By using the module command, it is possible to set the environment variables required for using the application in a unified manner.

This system provides a base environment that summarizes the environment variable settings required for each calculation environment. By loading the base environment first, you can easily arrange the environment variable settings.

The base environment prepared by this system is as follows.

Type	Module Name	Contents
Compiler + MPI + Library	BaseCPU/2021	Recommended environment for program development for general-purpose CPU nodes
	BaseVEC/2021	Recommended environment for program development for vector nodes
	BaseGPU/2021	Recommended environment for program development for GPU nodes
	BaseGCC/2021	Development environment when using GCC
Language environment + module	BasePy/2021	Program development environment for Python
	BasePyR/2021	Program development environment for R language
	BaseR/2021	Program development environment for JAVA
	BaseJulia/2021	Program development environment for Julia
Application	BaseApp/2021	Base environment for ISV and OSS application users

The recommended development environment of "1.2Thress types of Computiong Environments" is also maintained in the base environment. By using the recommended base environment, you can develop programs that are more suitable for compute node hardware.

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

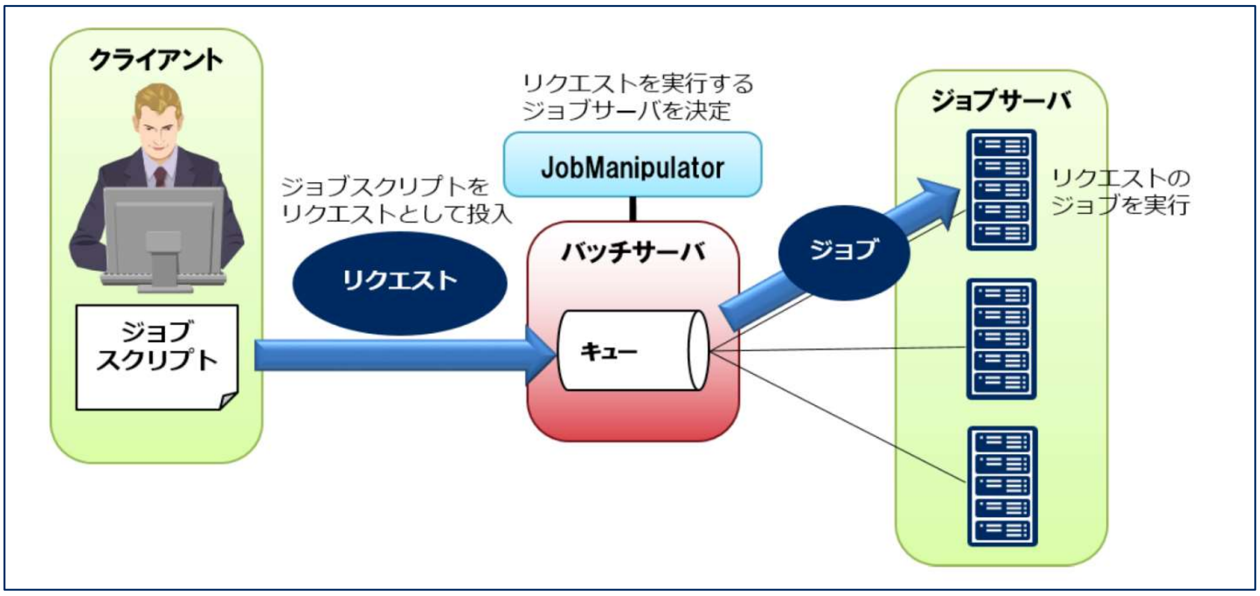
- Environment Modules officail
  http://modules.sourceforge.net/

1.5.3. Program execution

When executing the program, the calculation environment will be shared by many users. A job management system is provided to control the order of execution and node control so that program execution does not interfere with each other.

In this system, the job management system "NEC NQSV" enables batch use and conversational use of computational resources. Request job execution (submit a job) from the frontend to the job management system. For submitted job requests, the priority of other job requests, the amount of required resources, the usage status of SQUID, etc. are taken into consideration and judged by the job management system, and the execution order is determined.

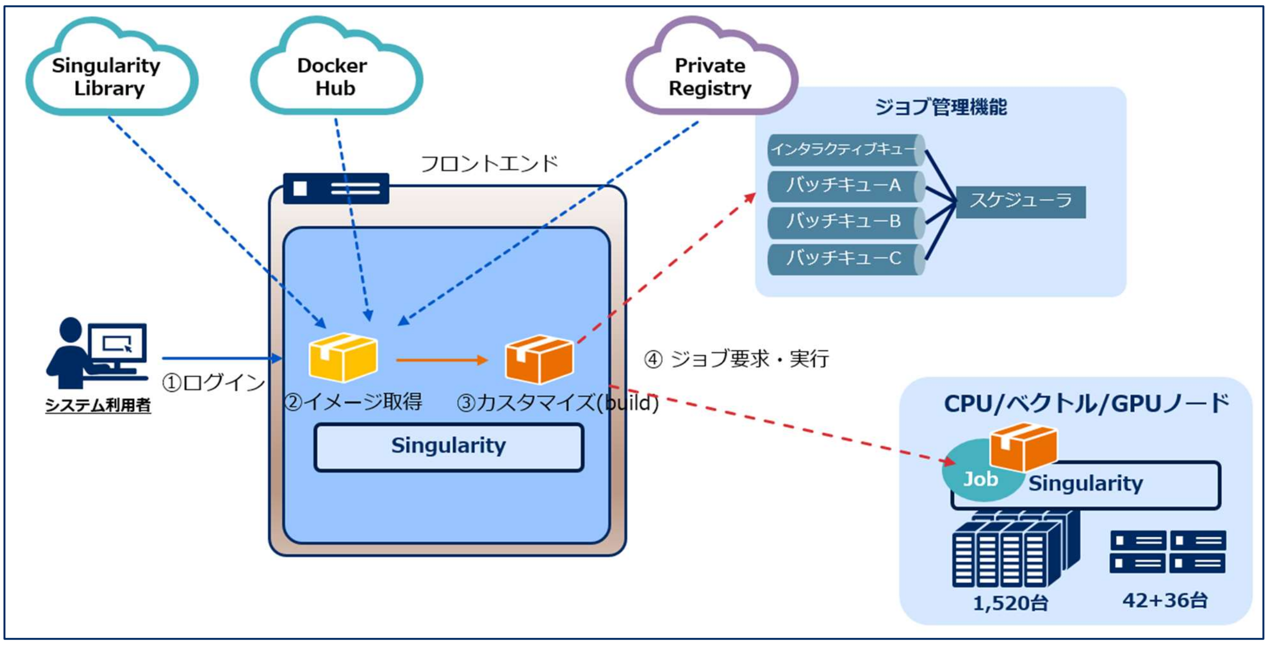
SQUID uses this batch processing method because it is assumed that it will be shared by a large number of users.



For details on how to use the job management system, refer to "5Program Execution Method".

1.5.4. Use container

In this system, it is possible to use containers using Singularity. You can expand the container image published on the Internet or in the system to the home directory, customize it, and execute it as a job.



By using the configured software set as a container, there are effects such as reducing the time and effort of preparation, improving the reproducibility of program execution, and being able to use it without complicated prior knowledge.

For details on how to use the container, refer to "4.6How to Use Containers". For an example of job execution using a container, also refer to "5.6Container".

The outline of each procedure is described below.

(1) Get container image

Singularity containers typically use image files with a .sif extension. Image files can be downloaded from the Internet or published on the system, or files created by yourself be transferred to the system. The chpter "4.6.1Preparation of container image" describes the following steps.

- Transfer from a local workstation into this system
- Get the image from the SQUID local registry
- Get an image from Docker Hub
- Appendix: Get image from NVIDIA GPU CLOUD (NGC)

The container published by each community does not always work on this system, but it is useful for preparing the execution environment, so the image acquisition method using various communities is explained in this document.

(2) Customize container (build)

Customize the image file as needed to suit what you want to do. The process of appling the changes in the image file is called build. In this chapter, "4.6.2Customize and Build Container Images" Describes the following steps as a customization method that can be performed in the system.

- How to customize via sandbox
- How to describe the customization contents in the def file

The method of customizing via sandbox can be customized like a normal UNIX shell, but there are cases where the user is restricted because you don't have root privileges. If possible, it is more unconstrained to customize on your own local workstation with root privileges and transfer the image files into the system.

(3) Job Request / Execution

Program execution in the system is basically executed by requesting a job to the job management system. "5.6 Container" describes the procedure for executing a job using a container.

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

- Singularity officail
  https://sylabs.io/singularity/

- Singularity 3.7 User Guide
  https://sylabs.io/guides/3.7/user-guide/

2. Instructions for Front-end Login

2.1. Login With SSH

In order to access SQUID, it is necessary to access the supercomputer network CMC-Scinet to which a large-scale computer system is connected. This system has an HPC front end and an HPDA front end for access to SQUID.

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

No	Server	Host Name
1	HPC frontend	squidhpc.hpc.cmc.osaka-u.ac.jp
2	HPDAfrontend	squidhpda.hpc.cmc.osaka-u.ac.jp

Access the above host name using the following connection method.

Connection method	SSH
Authentication method	2-factor Authentication
OS Japanese code	ja_JP.UTF-8

2.1.1. Installation of 2-factor Authentication Application

In order to SSH login to the front end severs, Google Authenticator's two-factor authentication would be needed. Download the authentification application and install it on your device (desktop PC or smart-phone, etc).

Availavle two-factor authentication applications

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 procedure uses Google Authenticator for iOS as an example.

2.1.2. First Login

Access with SSH to the front-end environment using SSH commands or terminal software.

(4) First SSH access

➢ **Login from UNIX-based OS (Linux, MacOS)**

Use the ssh command.

Example) Login to the HPC front-end environment (squidhpc.hpc.cmc.osaka-u.ac.jp).

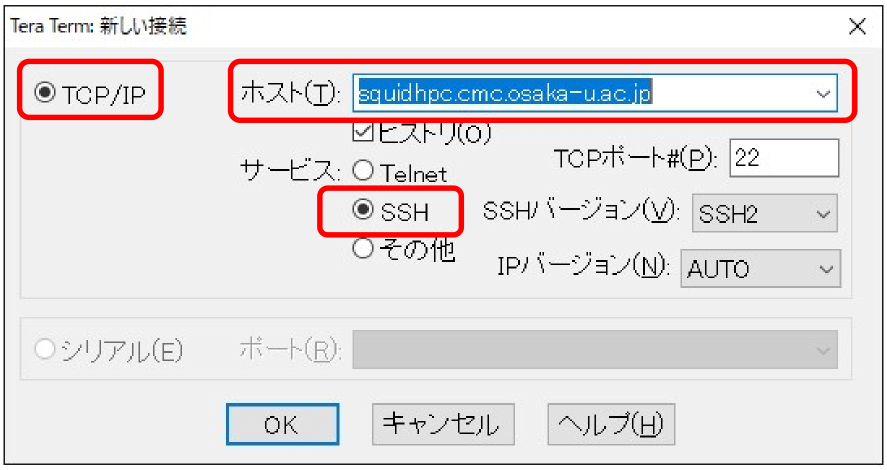
```
$ ssh (-l User Name) squidhpc.hpc.cmc.osaka-u.ac.jp
The authenticity of host ‘squidhpc.hpc.cmc.osaka-u.ac.jp (133.1.66.X)’ can’t be established.
RSA key fingerprint is 32:fd:73:4e:7f:aa:5d:3c:2e:ab:37:83:d6:55:98:e2.
Are you sure you want to continue connecting (yes/no)? yes
                                        (There is an inquiry only at the first access)
(User Name)@squidhpc.hpc.cmc.osaka-u.ac.jp's password: *****
                        (Enter the same password as the user management system.)

※(-l User Name) : Specify when the login user name is different from the user name of the local machine
```

➢ **Login from Microsoft® Windows®**

Access with SSH from a terminal software.

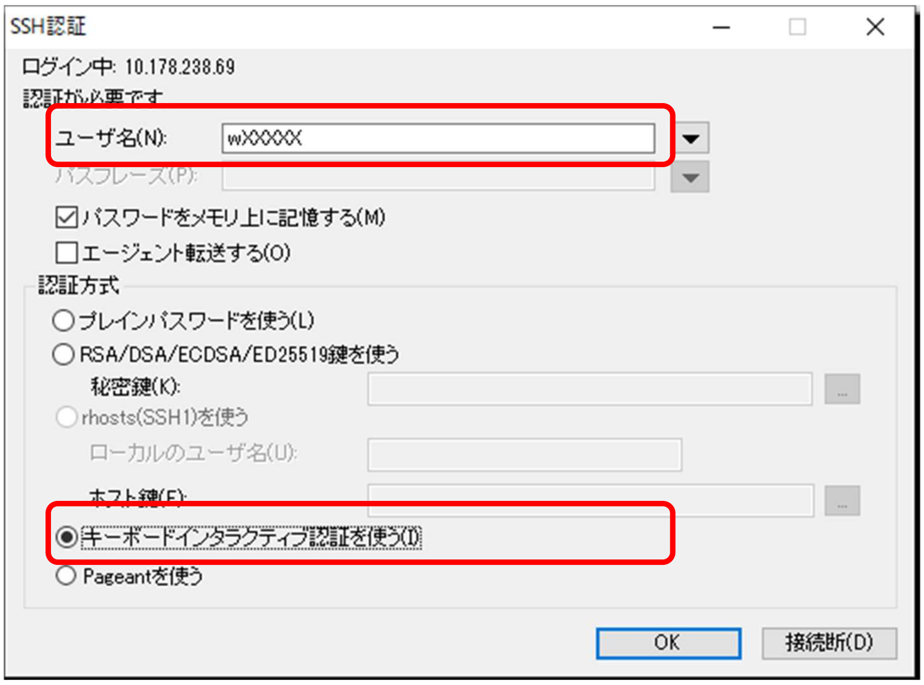
Example) Instructions for SSH access using “Tera Term Pro” free software.

- Start Tera Term Pro and open “Tera Term: New connection” dialog
- Select TCP / IP.
- Enter the host name (here, squidhpc.hpc.cmc.osaka-u.ac.jp) in the host field.
- Select SSH as the service.
- Click OK


Tera Term: 新しい接続

[✓ TCP/IP]　ホスト(T): squidhpc.cmc.osaka-u.ac.jp
☑ヒストリ(O)
サービス: ○Telnet　[✓ SSH]　○その他
TCPポート#(P): 22
SSHバージョン(V): SSH2
IPバージョン(N): AUTO
○シリアル(E)　ポート(R):

Enter the user name on the SSH authentication screen and select "キーボードインタラクティブ認証を使う" as the authentication method.


SSH認証

ログイン中: 10.178.238.69
認証が必要です
ユーザ名(N): wXXXXX
パスフレーズ(P):
☑パスワードをメモリ上に記憶する(M)
☐エージェント転送する(O)
認証方式
○プレインパスワードを使う(L)
○RSA/DSA/ECDSA/ED25519鍵を使う
秘密鍵(K):
○rhosts(SSH1)を使う
ローカルのユーザ名(U):
ホスト鍵(F):
[✓ キーボードインタラクティブ認証を使う(I)]
○Pageantを使う

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

(5) Initial setting of 2-factor authentication

When you log in for the first time, the following will be displayed on the terminal.

```
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@squidh
pc.hpc.cmc.osaka-
u.ac.jp%3Fsecret%3DDXXXXXXXXCLI%26issuer%3Dsquidhpc.hpc.cmc.osaka-u.ac.jp
```

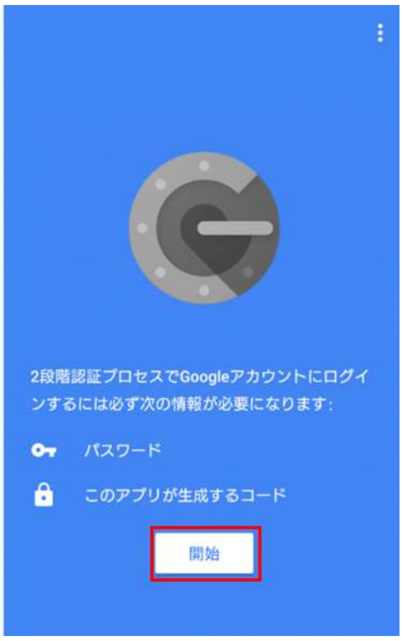
```
Your new secret key is: XXXXXXXXXXX
Enter code from app (-1 to skip): -1
Code confirmation skipped
Your emergency scratch codes are:
```

Use the QR code displayed on the screen or the secret key following "Your new secret key is:" to register with the two-step authentication application.

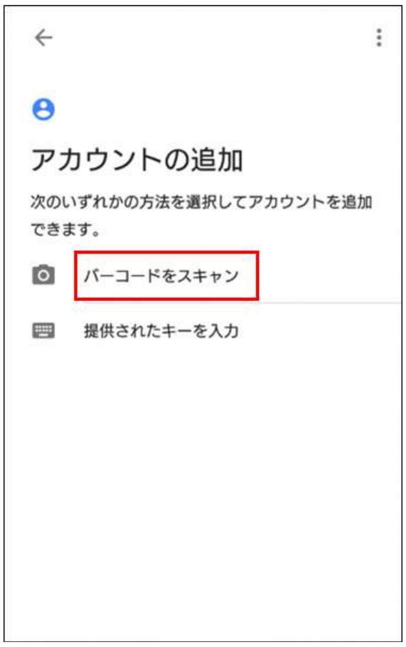
※ The QR code notation may be corrupted depending on the window size of the terminal software. Please adjust the font size or use the URL and secret key provided.

(6) 2-factor authentication app settings

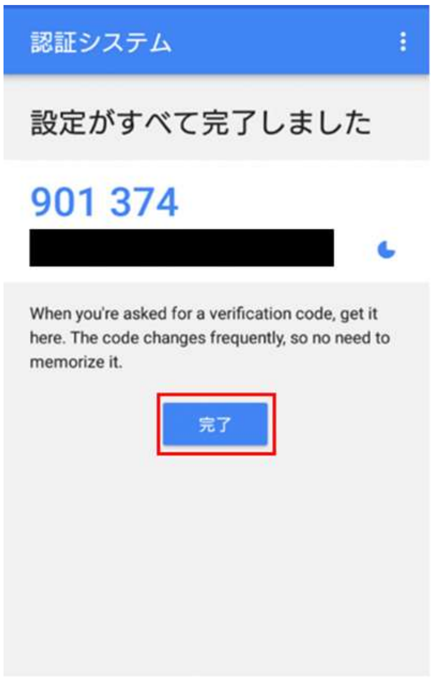
① Start the Google Authenticator app. and click the "Start" button.


2段階認証プロセスでGoogleアカウントにログインするには必ず次の情報が必要になります:
パスワード
このアプリが生成するコード
開始

② Select "Scan Barcode" or "Enter the provided key" and enter the QR code or secret key displayed in (2).



③ When you completed, the target user will be registered in the Google authentication system and a one-time password will be issued.



When you're asked for a verification code, get it here. The code changes frequently, so no need to memorize it.

(7) End of initial login

Go back to the terminal and fill in the questions.

```
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 キー)
```

Enter -1 for "Enter code from app (-1 to skip):" to skip.

"OK? Hit enter:" is displayed. Press Enter to log out once.

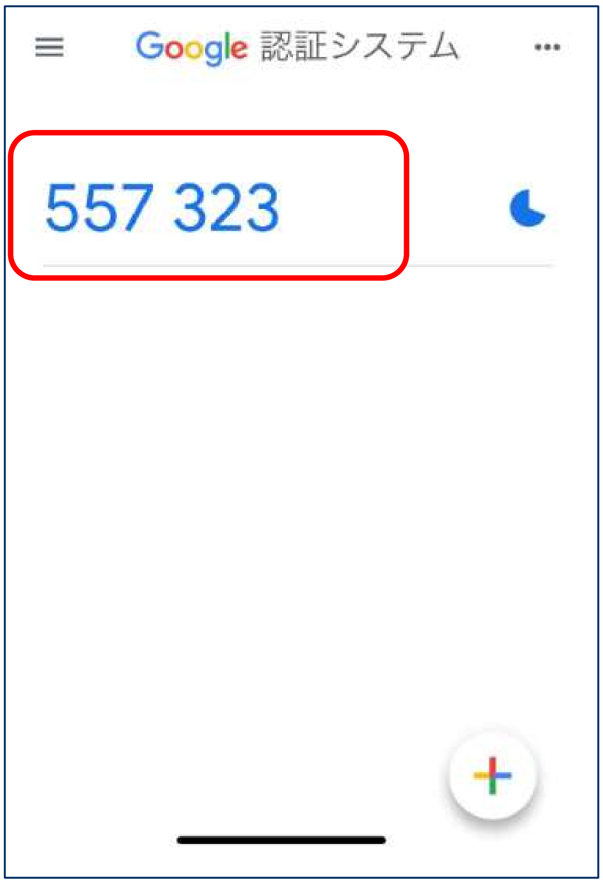
Note: In "Your emergency scratch codes are:", it is possible to issue a specified number of codes that can be used only once, but it is set to 0 for the security reason.

2.1.3. Second and subsequent Logins

SSH login is performed using the one-time password registered at the first login.

(1) Launch the 2-factor authentication app

Launch the two-step verification app and check your one-time password.



※ The one-time password is updated every 30 seconds. The latest one-time password will be used when logging in, so keep the above screen displayed until login is complete.

(2) SSH accesss

Use the terminal software or SSH command to perform SSH access. When logging in, use the one-time password obtained in (1).

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

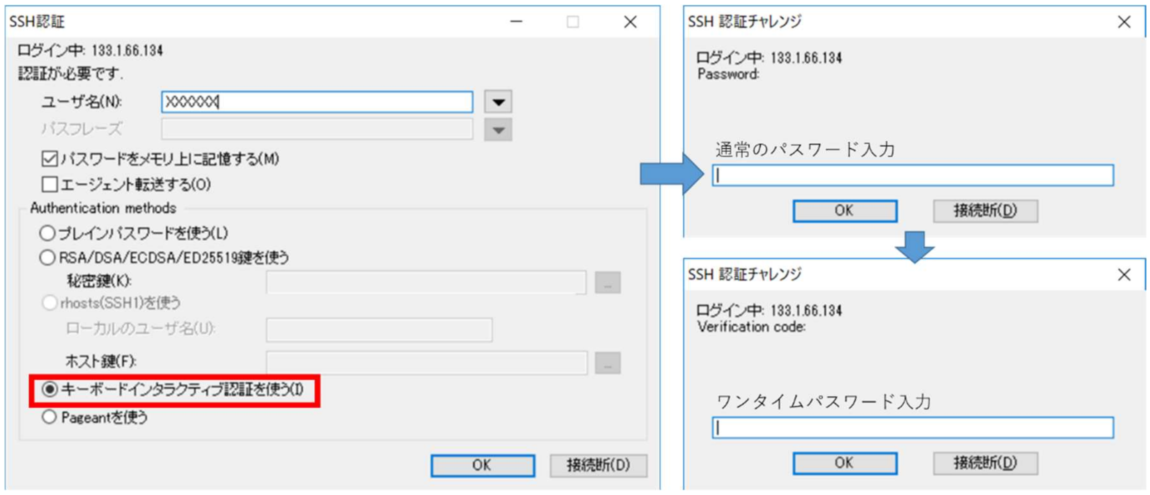
Use the ssh command. The following is an example of SSH login to the front-end environment (squidhpc.hpc.cmc.osaka-u.ac.jp).

```
$ ssh (-l User Name) squidhpc.hpc.cmc.osaka-u.ac.jp
(User Name)@squidhpc.hpc.cmc.osaka-u.ac.jp's password: *****
                          (Enter the same password as the user management system.)
Verification code: *****
                          (Enter one-time password)


※(-l User Name) : Specify when the login user name is different from the user name of
```

the local machine

➢ **Login from a Microsoft® Windows®**

An example using the free software Tera Term Pro is explained. In the SSH authentication challenge part, enter the one-time password to log in.


SSH認証

ログイン中: 133.1.66.134
認証が必要です.

ユーザ名(N): XXXXXX
パスフレーズ
☑ パスワードをメモリ上に記憶する(M)
☐ エージェント転送する(O)

Authentication methods
○ プレインパスワードを使う(L)
○ RSA/DSA/ECDSA/ED25519鍵を使う
秘密鍵(K):
○ rhosts(SSH1)を使う
ローカルのユーザ名(U):
ホスト鍵(F):
◉ キーボードインタラクティブ認証を使う(I)
○ Pageantを使う

SSH 認証チャレンジ

ログイン中: 133.1.66.134
Password:

通常のパスワード入力

SSH 認証チャレンジ

ログイン中: 133.1.66.134
Verification code:

ワンタイムパスワード入力

2.1.4. Recovery instruction if Two-factor Authentication became unavailable

Administrator operation is necessary, contact system administrator.

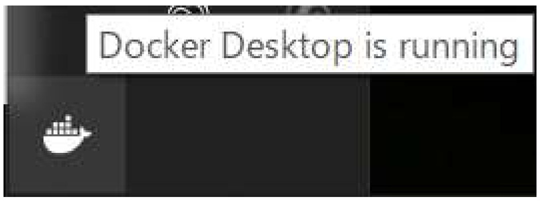
2.2. How to login with GSI-SSH

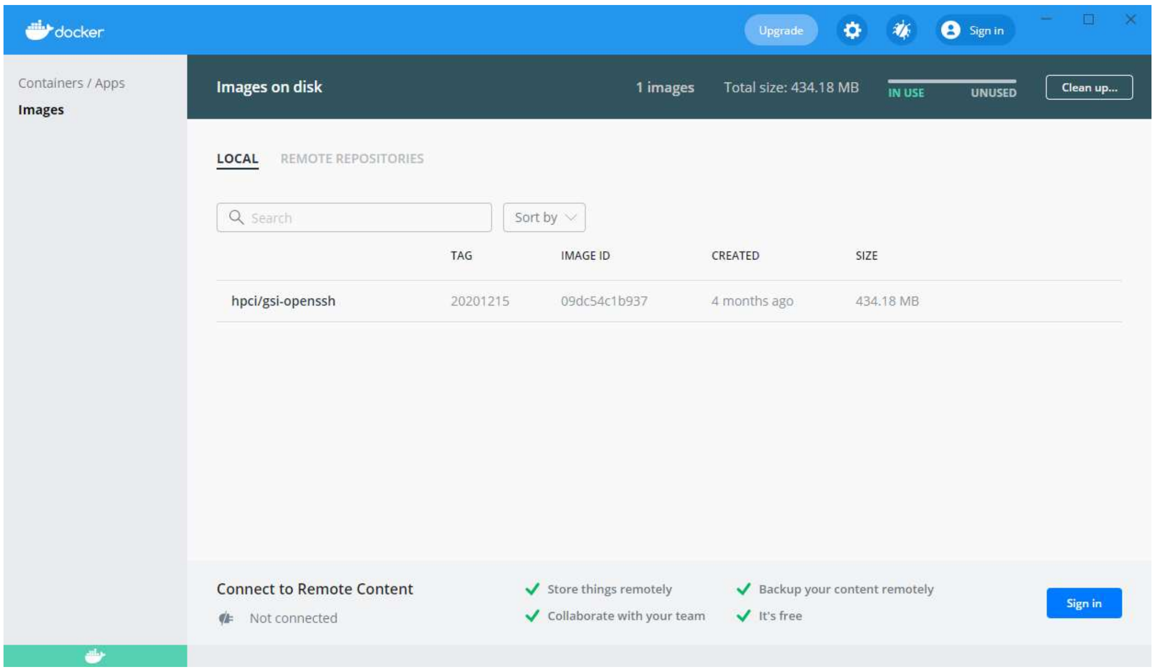
If HPCI user, user can login with GSI authentication. Here, we will introduce how to log in from Microsoft® Windows10®.

To login with GSI authentication, refer to "2.1. Digital Certificate Issuance Procedure" in the user manual "HPCI Login Manual (HPCI-CA01-001-21)" published by the HPCI Operations Secretariat in advance. In addition, please issue a digital certificate and a proxy certificate. Also, refer to "1.3.2. Docker Container image for GSI-OpenSSH" in the same manual, and install Docker Container image for GSI-OpenSSH.

(1) Start Docker Desktop

Make sure Docker Desktop for Windows® is running.





Images on disk   1 images   Total size: 434.18 MB   IN USE   UNUSED   Clean up...

LOCAL   REMOTE REPOSITORIES

	TAG	IMAGE ID	CREATED	SIZE
hpci/gsi-openssh	20201215	09dc54c1b937	4 months ago	434.18 MB

Connect to Remote Content
Not connected
✓ Store things remotely
✓ Collaborate with your team
✓ Backup your content remotely
✓ It's free
Sign in

(2) Launch Windows® PowerShell

Start Windows® PowerShell to run the docker command. Start Windows® PowerShell with general user privileges.

(3) Load Docker image (installation only)

Run the following command in Windows® PowerShell to load the Docker image.

```
PS C:¥Users¥username> docker load -i gsi-openssh-20201215.tar.bz2
c33ea345ab20: Loading layer [==================================================>]  242.3MB/242.3MB
039c616acc15: Loading layer [==================================================>]  31.23kB/31.23kB
7bf8d15f20d6: Loading layer [==================================================>]  4.096kB/4.096kB
dd7e690f7986: Loading layer [==================================================>]  737.3kB/737.3kB
Loaded image: hpci/gsi-openssh:20201215
PS C:¥Users¥username>
PS C:¥Users¥username> docker images hpci/gsi-openssh
REPOSITORY          TAG         IMAGE ID        CREATED         SIZE
hpci/gsi-openssh    20201215    09dc54c1b937    3 months ago    434MB
```

(4) Start Docker container

Execute the following command in Windows® PowerShell to start the loaded Docker container.

```
PS C:¥Users¥username> docker run -d --rm --name gsi-openssh -v
c:¥Users¥username¥Documents:/home/hpciuser/work hpci/gsi-openssh:20201215
61506efcd6a00f422aeacf04fb4819307f591b173b79b06fdaa5b64c1ac1b827
PS C:¥Users¥username>
```

(5) Start GSI-OpenSSH bash

Execute the following command to execute GSI-OpenSSH in / bin / bash.

```
PS C:¥Users¥username> docker exec -i -t gsi-openssh /bin/bash
[hpciuser@a4dc58b0dee1 ~]$
```

(6) Download proxy certificate

Execute the following command to download the proxy certificate.

```
[hpciuser@a4dc58b0dee1 ~]$ myproxy-logon -s portal.hpci.nii.ac.jp -l hpciXXXXXX
                                                                      Task ID
```

Execute the following command to check the obtained proxy certificate information.

```
[hpciuser@a4dc58b0dee1 ~]$ grid-proxy-info
subject   : /C=JP/O=NII/OU=HPCI/CN=username[hpciXXXXXX]/CN=XXX/CN=XXX/CN=XXX/CN=XXX
issuer    : /C=JP/O=NII/OU=HPCI/CN=username [hpciXXXXXX]/CN=XXX/CN=XXX/CN=XXX
identity : /C=JP/O=NII/OU=HPCI/CN=username [hpciXXXXXX]
type       : RFC 3820 compliant impersonation proxy
strength : 2048 bits
path       : /tmp/x509up_u2000
timeleft : 11:59:39
[hpciuser@a4dc58b0dee1 ~]$
```

(7) Login to login server

Specify the login server and login with GSI authentication.

・When logging in to the HPC login server

```
[hpciuser@a4dc58b0dee1 ~]$ gsissh -p 2222 squidhpc.hpc.cmc.osaka-u.ac.jp
Last login: Thu Dec 17 21:21:52 2020 from XX.XX.XX.XX
-bash-4.2$ pwd
/sqfs/home/username
```

・When logging in to the HPDA login server

```
[hpciuser@a4dc58b0dee1 ~]$ gsissh -p 2222 squidhpda.hpc.cmc.osaka-u.ac.jp
Last login: Thu Dec 17 21:21:52 2020 from XX.XX.XX.XX
-bash-4.2$ pwd
/sqfs/home/username
```

(8) Stop Docker container

Execute the following command to log out from the login server and bash.

```
-bash-4.2$ exit
[hpciuser@a4dc58b0dee1 ~]$
```

```
PS C:¥Users¥username>
```

Execute the following command to stop the Docker container.

```
PS C:¥Users¥username> docker stop gsi-openssh
gsi-openssh
PS C:¥Users¥username>
```

2.3. File Transfer

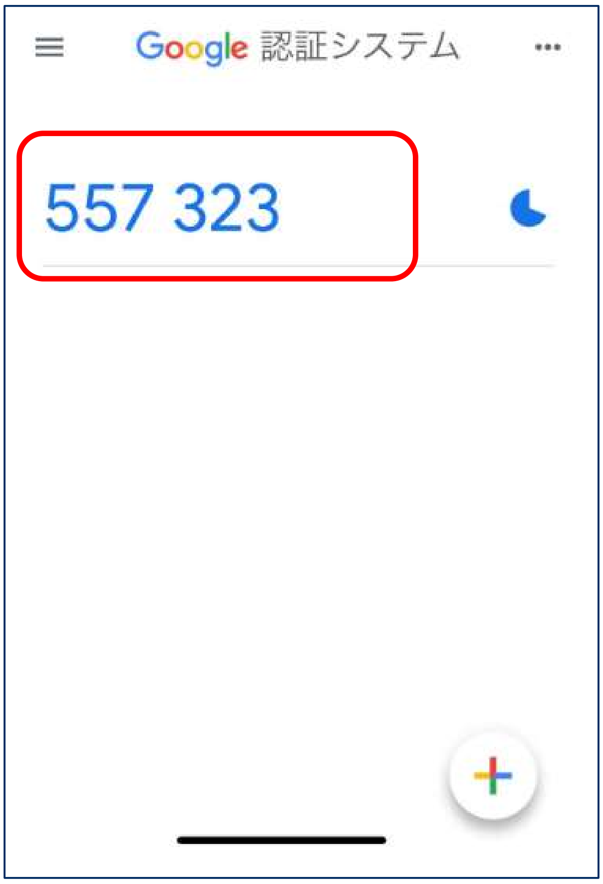
2.3.1. File Transfer On UNIX-based OS (Linux, MacOS)

Here, we will introduce an example of connecting to an FTP server from a UNIX-based OS (Linux, MacOSX) terminal such as a laboratory outside the Cybermedia Center. It is assumed that SFTP or SCP is installed on the terminal side in advance, so if SFTP or SCP is not installed, please install it yourself or consult with the terminal administrator.

※For file transfer, the initial setting of 2-factor authentication is required in advance. For details, refer to the section "2.1.2 First login".

(1) Launch the 2-factor authentication app

Launch the two-step verification app and check your one-time password.



※The one-time password is updated every 30 seconds. Keep displayed above screen so that you can use the latest on-time password.

➢ Connection procedure (for SFTP)

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

(2) SFTP command execution

Enter the following command

```
$ sftp a61234@squidhpc.hpc.cmc.osaka-u.ac.jp
$ sftp a61234@squidhpda.hpc.cmc.osaka-u.ac.jp
```

(3) Enter password

When connecting to the FTP server for the first time, the following message will be appeared, and enter yes.

```
The authenticity of host 'octopus.hpc.cmc.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.cmc.osaka-u.ac.jp,133.1.65.XX' (RSA) to the
 list of known hosts.
```

When you are asked the password, enter the password same as the user manangement system.

```
a61234@squidhpc.hpc.cmc.osaka-u.ac.jp's password:    XXXXXXXX
a61234@squidhpda.hpc.cmc.osaka-u.ac.jp's password:    XXXXXXXX
(The password you entered is not displayed)
```

(4) Enter one-time password

When you are asked the Verification code, enter the one-time password you obtained from the two-step verification app.

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

(5) Start FTP communication

File transfer is performed using the get and put commands in the same way as normal

FTP. The following example transfers the “sample.f “file to the home of a large computer system.

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

➢ **Connection Procedure (for SCP)**

When transferring files using the scp command, it is necessary to know the file name to be transferred and the directory name of the transfer destination in advance.

(1) Execution of SCP command

Enter the following command

・ **When transferring the file sample.f on the current directory of the local terminal to the home directory of SQUID**

```
$ scp sample.f a61234@squidhpc.hpc.cmc.osaka-u.ac.jp:
$ scp sample.f a61234@squidhpda.hpc.cmc.osaka-u.ac.jp:
```

・ **When transferring the file sample.c in the abc directory under the home directory of SQUID to the current directory of the local terminal**

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

※ scp can also transfer the multiple files at once, or recursively transfer files across directories. For more information, refer to the scp manual with the "man scp" command.

(2) Enter password

When connecting to the FTP server for the first time, the following message will be appeared, and enter yes.

```
The authenticity of host 'squidhpc.hpc.cmc.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.cmc.osaka-u.ac.jp,133.1.65.XX' (RSA) to
```

```
the list of known hosts.
```

When connecting to the FTP server for the first time, the following message will be appreared, and enter yes.

```
a61234@squidhpc.hpc.cmc.osaka-u.ac.jp's password:    XXXXXXXX
a61234@squidhpda.hpc.cmc.osaka-u.ac.jp's password:    XXXXXXXX
(The password you entered is not displayed)
```

(3) Enter a one-time password

When you are asked the verification code, enter the one-time password you obtained from the two-step verification app.

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

(4) File transfer

The file transfer will be started. The transfer status, transfer file size and transfer time will be displayed as below.

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

2.3.2. File Transfer on Windows®

If you install SSH (Secure Shell) compatible file transfer software on your computer, you can connect to a file transfer server (hereinafter referred to as FTP server). There are many applicable software, and this section shows the file transfer using free software WinSCP as a reference.

(1) Obtaining WinSCP

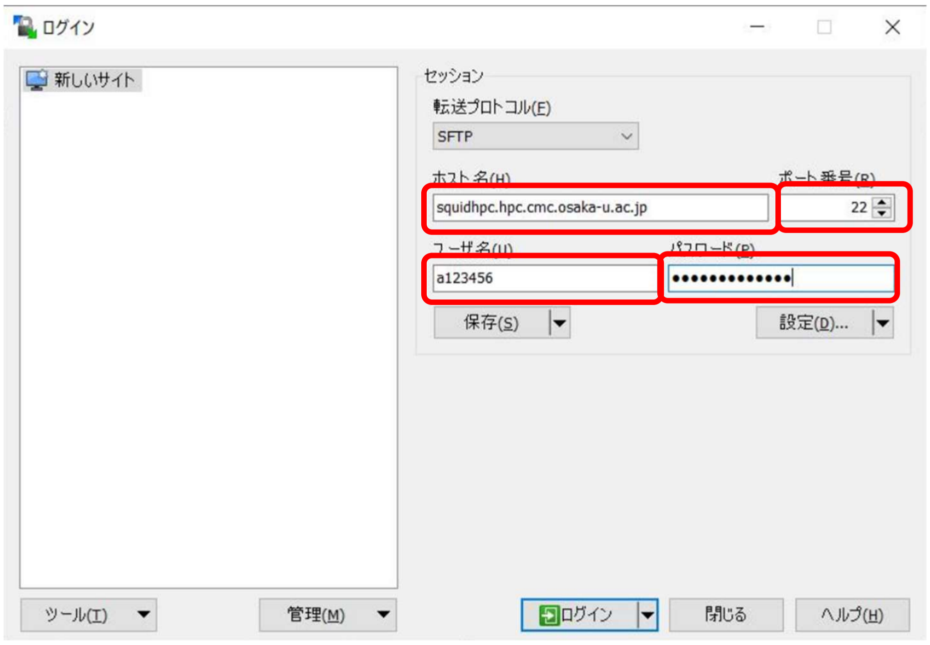
Download the WinSCP from the WinSCP official website and install it.

(2) Launch WinSCP

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

```
Port Name:   squidhpc.hpc.cmc.osaka-u.ac.jp or squidhpda.hpc.cmc.osaka-u.ac.jp
Prot Number: 22
```

User Name:　User number
Password: User management password


ログイン

セッション

転送プロトコル(F) SFTP

ホスト名(H) squidhpc.hpc.cmc.osaka-u.ac.jp

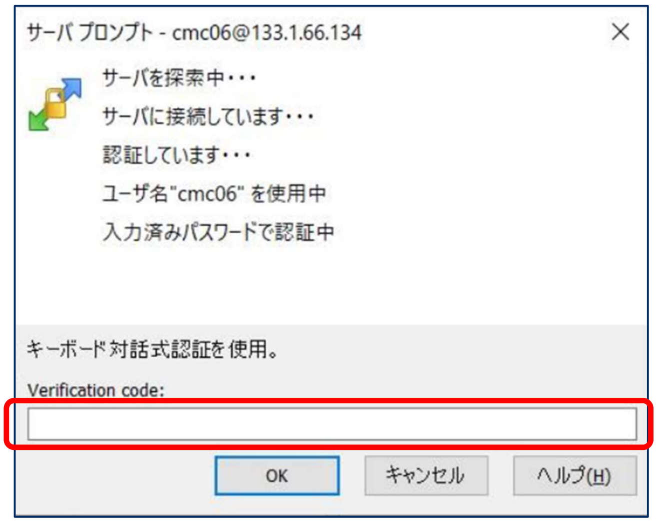
ポート番号(R) 22

ユーザ名(U) a123456

パスワード(P) •••••••••••••

(3) Enter a one-time password

When you are asked the verification code, enter the one-time password obtained from the two-step verification app.


サーバ プロンプト - cmc06@133.1.66.134

サーバを探索中・・・

サーバに接続しています・・・

認証しています・・・

ユーザ名"cmc06" を使用中

入力済みパスワードで認証中

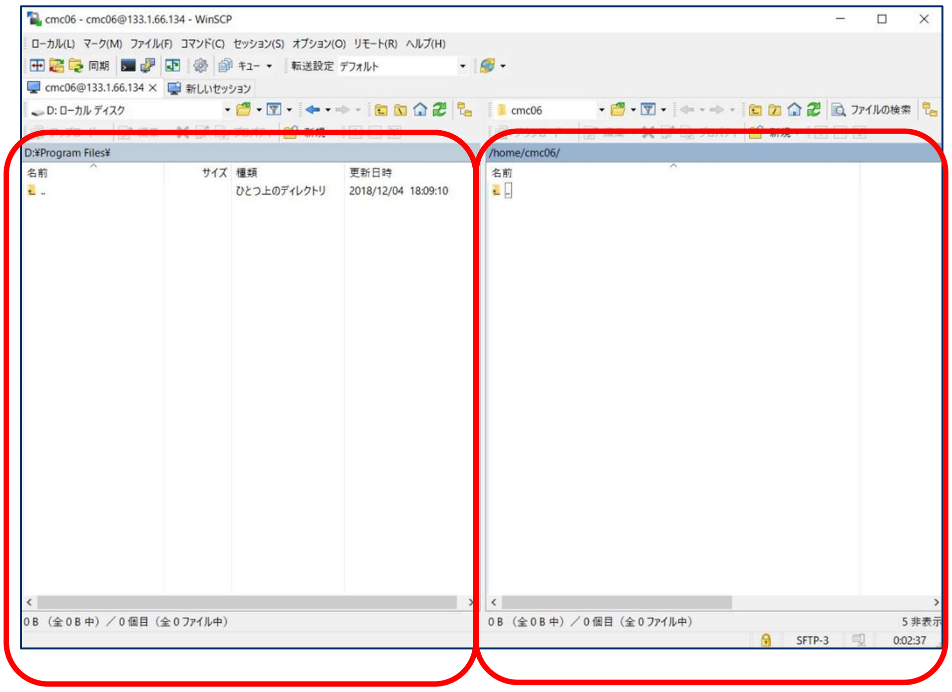
キーボード対話式認証を使用。

Verification code:

※When transferring a text file such as a program and NQS script, set the transfer mode to "テキスト" in the "設定" → "環境設定" item. After changing the settings, it is convenient to "保存" the session information.

(4) Start file transfer

When you log in successfully, the local (Windows®) folder and the remote (SQUID) folder will be displayed on the left and right respectively.



Local (Windows) side **Remote (SQUID) side**

You can transfer files by dragging and dropping the files to be transferred to the left or right.

3. Front-end Environment

3.1. Common Instructions

3.1.1. How to Use the File System

The SSD area and HDD area can be used directly from the front-end environment as a file system. The allocated disk space and its quota limit are as below.

Storage	File System	Area Name	Path	Quota		Remarks
				Size	File	
HDD	EXAScaler (Lustre)	Home area	/sqfs/home/(User Number)	10GiB	---	Includes browser access area
		Extended area	/sqfs/work/(Group Name)/(User Number)	5TiB (+)	---	Additional purchase possible
			/sqfs/s3/(UUID)			For S3 access
SSD	EXAScaler (Lustre)	High speed area	/sqfs/ssd/(Group Name)/(User Number)	0B (+)	---	Additional purchase possible

➢ **Home Area**

This is the minimum area given when registering as a user. 10GiB will be allocated as the initial capacity.

The file path (home) is as follows.

```
/sqfs/home/(User Number)
```

ex: User Number is "user001" /sqfs/home/user001

➢ **Extended Area**

This is an area where high-speed and large-capacity I / O is possible. It has the following features

- 5TiB is allocated as the initial capacity, and additional capacity can be purchased for each group by application.
- An area for accessing with S3 API from outside SQUID is prepared with another path.

The file path (work) is as follows.。

```
/sqfs/work/(Group Name)/(User Number)
```

ex: User Number is"user001"、Group Name is G012345" /sqfs/work/G012345/user001

The file path (s3) is as follows.

```
/sqfs/s3/(UUID)
```

※ The UUID is identification information assigned to each access key used for S3 API access, and can be confirmed when the access key is generated.

➢ **High Speed Area**

High Speed Area is SSD storage area that can access with futher fast I/O.

Key features are as below:

- Ultra-fast file system with SSD storage
- There is no initial capacity allocation, but additional capacity can be purchased for each group by application.

The file path (ssd) is as follows.

```
/sqfs/ssd/(Group Name)/(User Number)
```

ex: User Number is"user001"、Group Name is"G012345" /sqfs/ssd/G012345/user001

3.1.2. How to check usage status

The usage status on computer and file system are able to check from the front-end server. Usage status can be displayed using "usage_view command as below.

```
$ usage_view
---------------+ Group: G012345 +---------------

[Group summary]

          SQUID points     HDD(GiB)    SSD(GiB)
------- -------------- ---------- ----------
usage              0.0        87.9         0.0
limit          11000.0     46080.0     20480.0
remain         11000.0     45992.1     20480.0
rate(%)            0.0         0.2         0.0


[Detail]

          SQUID points              Home(GiB)    HDD(GiB)    SSD(GiB)
------- -------------- ---------------------- ---------- ----------
user001            0.0    0.0 /   10.0 /   10.0         0.0         0.0
user002            0.0    0.0 /   10.0 /   10.0         0.0         0.0
user003            0.0    0.0 /   10.0 /   10.0         0.0         0.0

[Node-hours]

                                 Usage      Available*
```

```
------------------- -------------- --------------
CPU           node          0.00        9090.90
GPU           node          0.00        8184.52
Vector        node          0.00        8461.53
Azure  CPU    node          0.00        6547.61
       GPU    node          0.00           0.00
OCI    CPU    node          0.00        6547.61
       GPU    node          0.00           0.00
Secure CPU    node          0.00        6666.66
       GPU    node          0.00        6666.66
       Vector node          0.00        6666.66

 * Available Node-hour is estimated based on remains of SQUID, Bonus point.
```

Displayed informations are as belwow:

[Group summary]

- SQUID points

  Shows a points that group are used until now, applicated point, available point, and the usage rate.
- Bonus points

  It shows the bonus points that the group has used until now, the bonus points that have been assigned, the remaining bonus points that can be used, and the usage rate. (Bonus points are points given when using the trial system, etc.)
- HDD(GiB)

  Shows the disk usage, total available disk space, remaining available disk space, and utilization of the extended space used by the group.

  ※For the disk capacity, the total value of the home area and the extended area is displayed.
- SSD(GiB)

  It shows the disk usage, total available disk space, available remaining disk space, and usage rate in the high-speed area used by the group.

[Detail]

- SQUID points

  Shows a points that each user are used until now, applicated point, available point, and the usage rate.
- Home(GiB)

  For each user, the disk usage of the home area used by the user / total usable disk usage / remaining usable disk capacity is displayed.
- HDD(GiB)

For each user, the disk usage of the extended area used by the user is displayed.
※For the disk capacity, the total value of the home area and the extended area is displayed.

- SSD(GiB)
  For each user, the disk usage of the high-speed area used by the user is displayed.

[Node-hours]

- Usage
  Displays the node time used by the group until now by node group.
- Usage(Bonus)
  Shows the node time that the group has used up to now with bonus points by node group.
- Available
  Displays the value of the remaining available SQUID points converted to the available remaining node time. Node time when all remaining SQUID points are used on that node
- Available(Bonus)
  Remaining Available SQUID Displays the value obtained by converting bonus points to the remaining available node time. This is the node time when all the remaining SQUID bonus points are used on that node.

3.1.3. Environment Settings

In this system, the environment variable settings of the compiler, library, and application environment are managed by Environment modules. By using the module command, it is possible to set the environment variables required for using the application in a unified manner. The main usage is shown below.

Command	Description
module avail	List of available development environments / apps
module list	List of loaded modules
module switch [file1] [file2]	Module replacement (file1 → file2)
module load [file]	Module loading
module unload [file]	Module unload

module purge	Unload all loaded modules
module show [file]	Detailed display of modules

This system provides a base environment that summarizes the environment variable settings required for each calculation environment. By loading the base environment first, you can easily arrange the environment variable settings.

The base environment prepared by this system is as follows.

Type	Module Name	Contents
Compiler + MPI + Library	BaseCPU/2021	Recommended environment for program development for general-purpose CPU nodes
	BaseVEC/2021	Recommended environment for program development for vector nodes
	BaseGPU/2021	Recommended environment for program development for GPU nodes
	BaseGCC/2021	Development environment when using GCC
Language environment + module	BasePy/2021	Program development environment for Python
	BasePyR/2021	Program development environment for R language
	BaseR/2021	Program development environment for JAVA
	BaseJulia/2021	Program development environment for Julia
Application	BaseApp/2021	Base environment for ISV and OSS application users

The following is example of module command executions.

① show available module list.

```
$ module avail
--------------------------- /system/apps/env/Base --------------------------------------------
BaseApp/2021(default)     BaseExtra/2021(default)     BaseGPU/2021(default)
BaseJulia/2021(default)    BaseR/2021(default)         BaseCPU/2021(default)
BaseGCC/2021(default)     BaseJDK/2021(default)       BasePy/2021(default)
BaseVEC/2021(default)
```

② load one of base environment modules

```
$ module load BaseCPU/2021
Loading BaseCPU/2021
  Loading requirement: intel/2020update4
```

③ check currently loaded mossules

```
$ module list
Currently Loaded Modulefiles:
 1) intel/2020update4    2) BaseCPU/2021(default)
```

④ show additional modue list

```
$ module avail
----------------------- /system/apps/env/cpu/Compiler ------------------------------------
intel/2020update4   inteloneAPI/2021.2

----------------------- /system/apps/env/cpu/mpi/intel2020u4 ----------------------------
intelmpi/2020update4

----------------------- /system/apps/env/cpu/lib/intel2020u4 ------------------------------
hdf5/1.10.5                netcdf-c/4.7.4              netcdf-cxx/4.3.1
netcdf-fortran/4.5.3       pnetcdf-c/1.12.2            pnetcdf-cxx/1.12.2
pnetcdf-fortran/1.12.2

------------------------ /system/apps/env/cpu/devtool -------------------------------------
ArmForge/21.0   XcalableMP/1.3.3

----------------------- /system/apps/env/Base ----------------------------------------------
BaseApp/2021(default)      BaseExtra/2021(default)    BaseGPU/2021(default)
BaseJulia/2021(default)     BaseR/2021(default)        BaseCPU/2021(default)
BaseGCC/2021(default)      BaseJDK/2021(default)      BasePy/2021(default)
BaseVEC/2021(default)
```

⑤ load additional module

```
$ module load hdf5/1.10.5
```

For details on the module command, refer to the online document (aka man) or the document on the official website.

- Modules v4.5.2 : Docs >> module
  https://modules.readthedocs.io/en/v4.5.2/module.html

3.2. How to use the HPC Front-end

3.2.1. Preparation for use

This section describes the preparation for using NICE DCV (Desktop Cloud Visiualization) on the HPC front end.

Obtain the NICE DCV client from the following URL and install it on your PC. Windows® OS, Linux OS, and Mac OS versions are available.

https://www.nice-dcv.com/2020-2.html

3.2.2. Creating a virtual session of the NICE DCV server

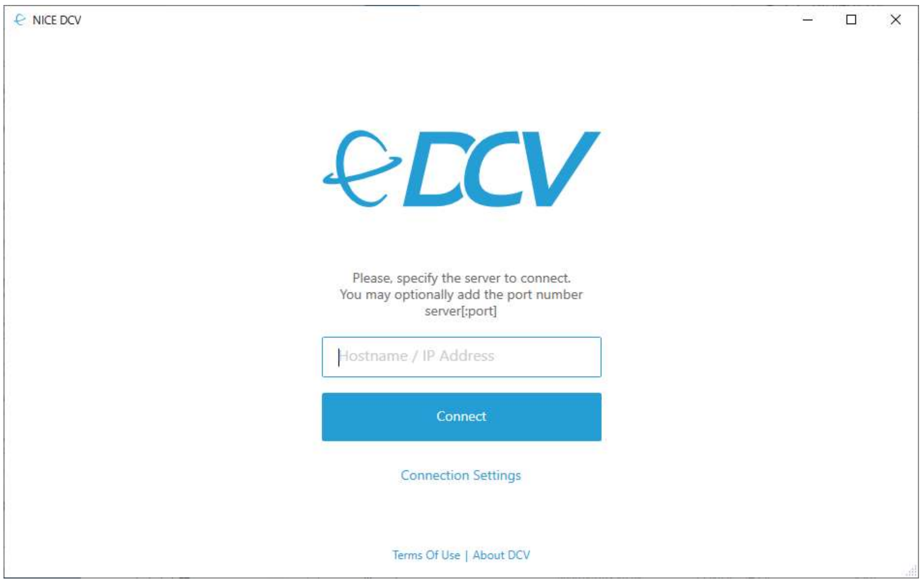
Login HPC front-end and create a virtual session of the NICE DCV Server.

```
$ dcv create-session --type=virtual session_name
(create virtual session)
$ dcv list-sessions
(check the created virtual session)
```

You need to connect to the server that created the virtual session in the 4 HPC front-end nodes. Please make a note of the node name and session name as they will be used in "3.2.3. Starting the NICE DCV Client".

3.2.3. Starting the NICE DCV Client

Start "dcvviewer". When started, the following window will be appered.

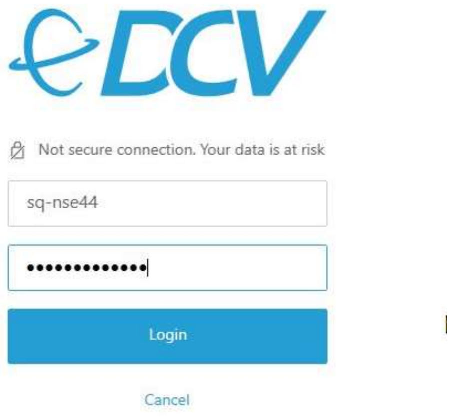


Please, specify the server to connect.
You may optionally add the port number
server[:port]

Connect to the server that created the instructions in " 3.2.2. Creating a virtual session of the NICE DCV server" .

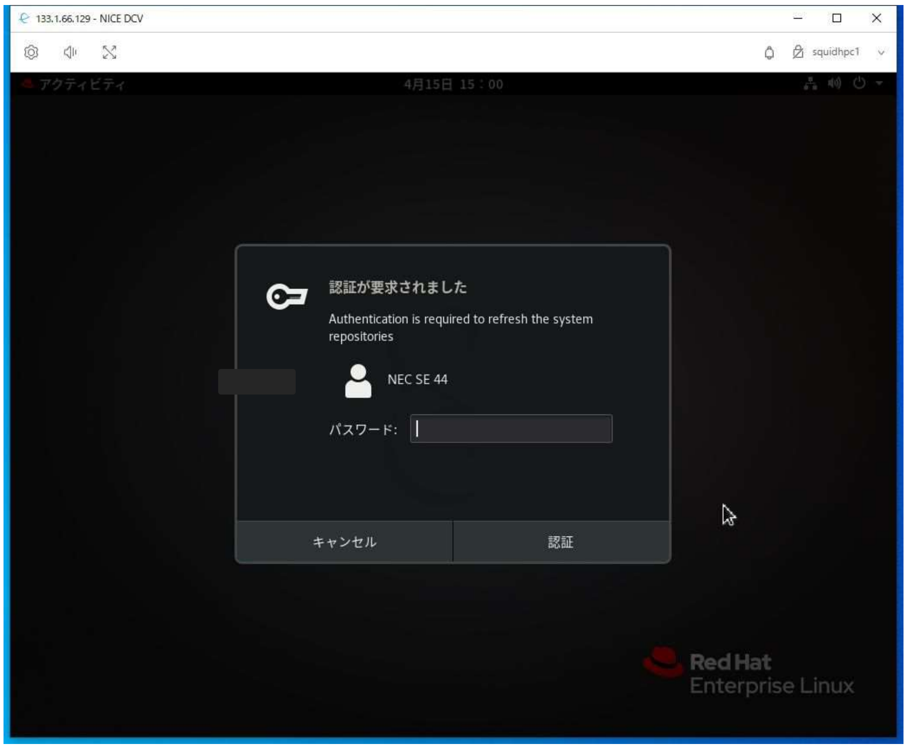
For example, in case the node name is "squidhpc1", and the virtual session name is "mysession", then input "squidhpc1.hpc.cmc.osaka-u.ac.jp:5901/#mysession" and click [Connect].

When the connection to the NICE DCV server on the HPC front end is successful, the following screen will be displayed. Enter your user name and password, and click [Login].

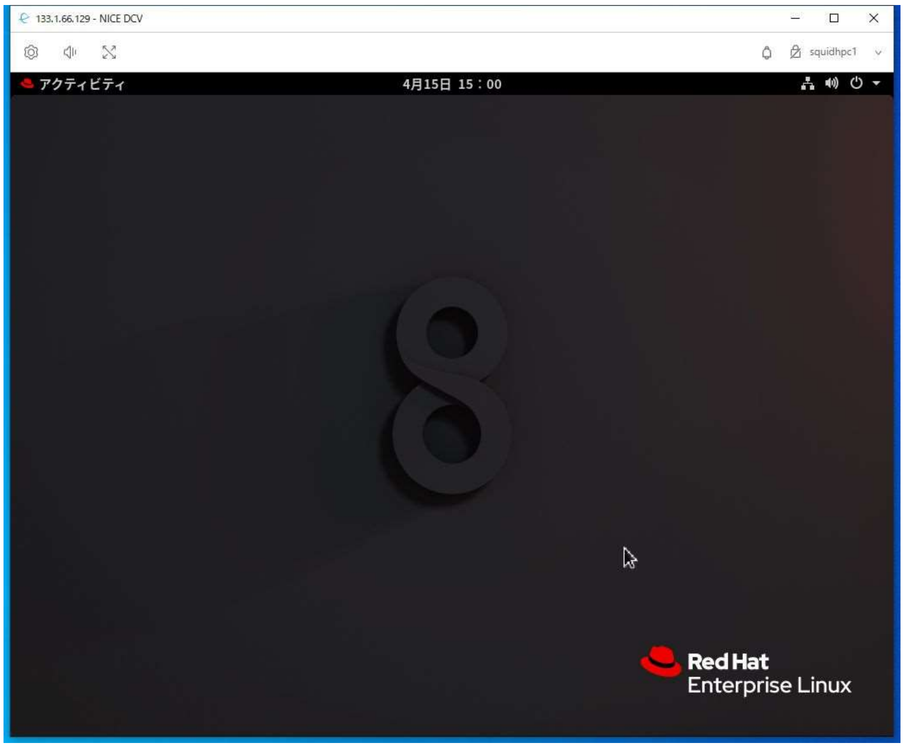


After login to NICE DCV successfully, the following HPC front-end desktop login screen will be displayed.

Enter your password and click Authenticate.



If the authentication is successful, the following HPC front-end desktop screen will be displayed.



3.3. How to use the HPDA Front-end

3.3.1. Preparation for Use

Describes how to prepare for using the jupyter container on the HPDA front end.

- Get jupyter container image

  Get the jupyter container image from the local registry.

  To get the jupyter container image, run the following command in the logged-in user's directory.

```
$ singularity pull jupyter.sif oras://cntm:5000/master_image/jupyter:1.0
```

  When the acquisition is successful, the image file jupyter.sif will be created.

  If the image file jupyter.sif is already exists, delete it in advance and then execute the above command.

  ※If a directory called jupyter exists, command execution will fail.

3.3.2. Start Jupyter Container

Describes how to start the jupyter container on the HPDA front-end.

- Executing a shell script for starting a jupyter container

In the same directory with jupyter.sif, execute the shell script for starting the jupyter container to start the jupyter container.

```
$ run_jupyter_container.sh
```

If the jupyter container is successfully started, the terminal will display the following and the Singularity prompt will be displayed.

```
$ run_jupyter_container.sh
jupyter notebook URL : https://squidhpda1.hpc.cmc.osaka-u.ac.jp:10146  ①
INFO:      Converting SIF file to temporary sandbox...
jupyter login token : 359b8b7886d9344fd2eccc15776c62f10de9a91d0b13f1a1  ②
Singularity>
```

① jupyter notebook URL

This is the URL for accessing the jypyter notebook with a web browser.

The port number is in the range of 10000 to 11000, and a free number is automatically assigned.

② jupyter login token

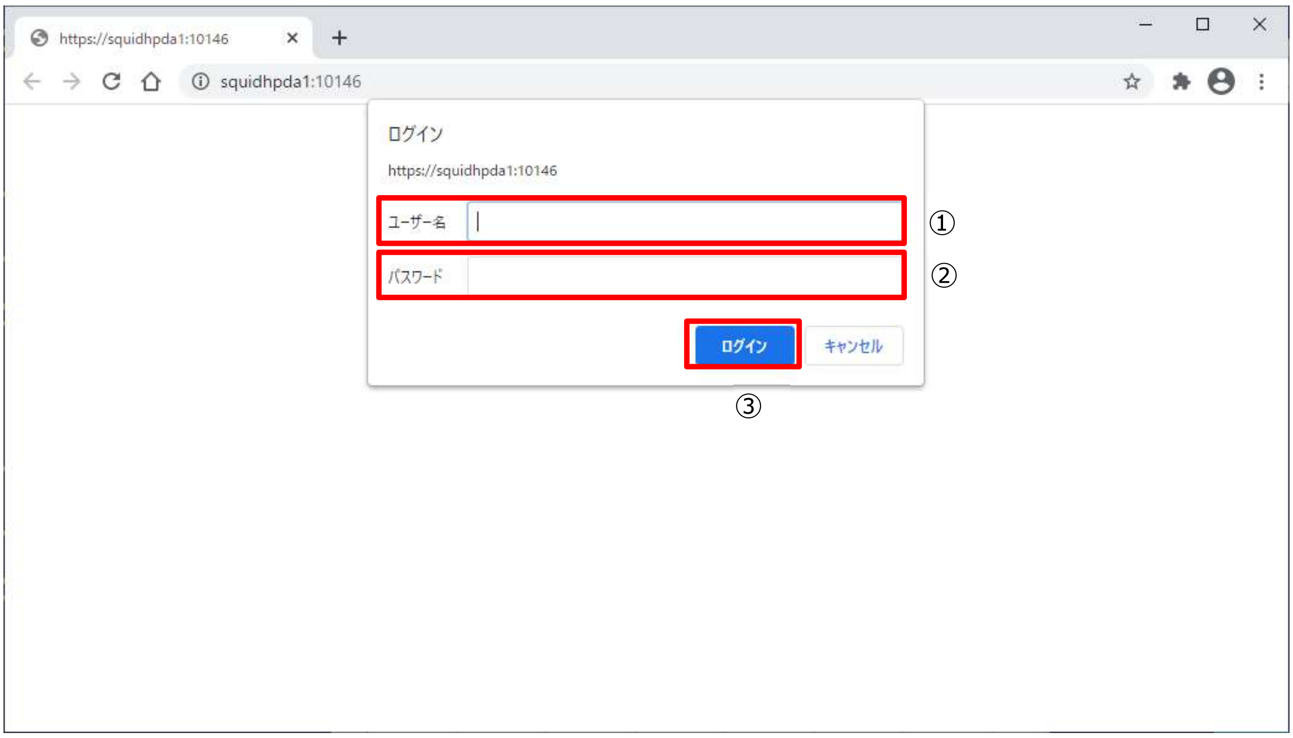
This token is entered on the jupyter authentication screen when accessed from a web browser.

3.3.3. Access to Jupyter Container

This section describes how to access the jupyter container from a web browser.

- User Name / password authentication

  Start a web browser and specify the jupyter notebook URL displayed in the terminal in "3.3.2 Starting the jupyter container".



① User Name

Enter the username you used to log in to the HPDA front end.

② Password

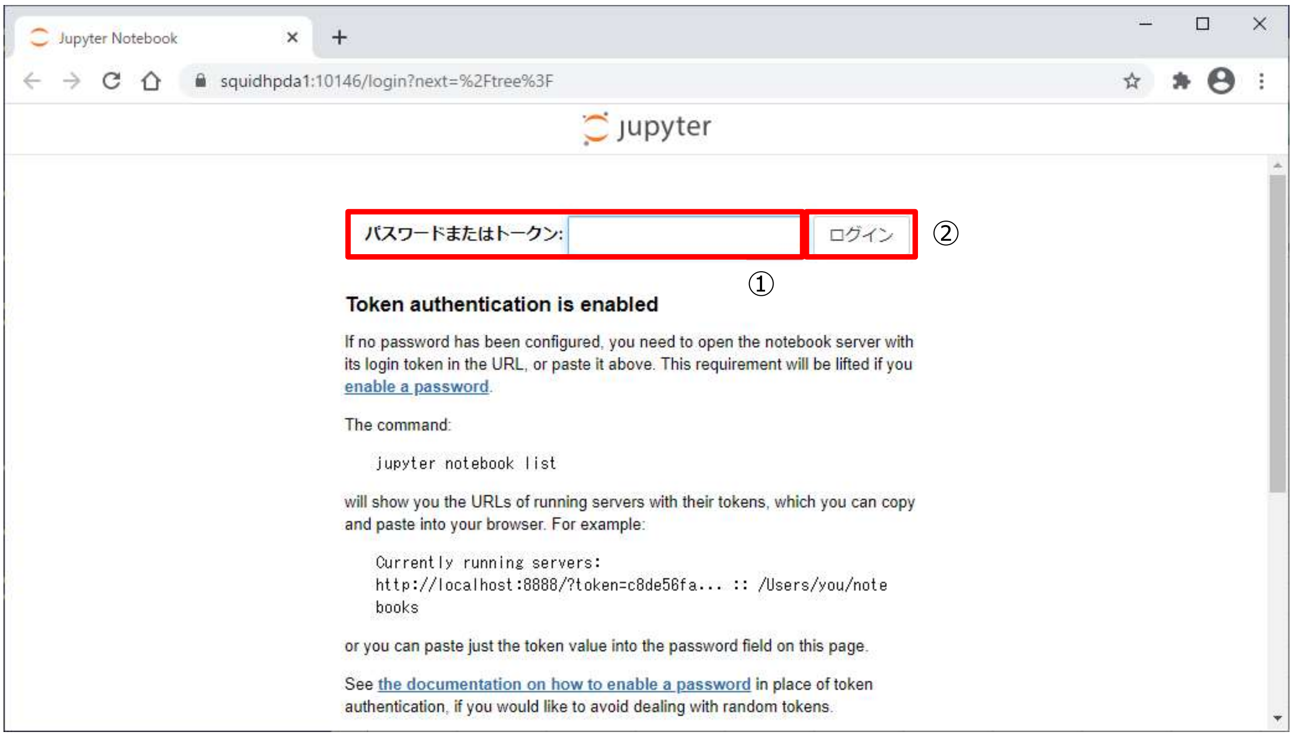
Enter the password you used to log in to the HPDA front end.

③ Login

Click the login button to authenticate with your username / password。

- Authentication by jupyter token

  If the user name / password authentication is successful, the jupyter authentication screen will be displayed.


Jupyter Notebook
squidhpda1:10146/login?next=%2Ftree%3F
jupyter
パスワードまたはトークン:
ログイン
①
②

Token authentication is enabled

If no password has been configured, you need to open the notebook server with its login token in the URL, or paste it above. This requirement will be lifted if you enable a password.

The command:

```
jupyter notebook list
```

will show you the URLs of running servers with their tokens, which you can copy and paste into your browser. For example:

```
Currently running servers:
http://localhost:8888/?token=c8de56fa... :: /Users/you/note
books
```

or you can paste just the token value into the password field on this page.

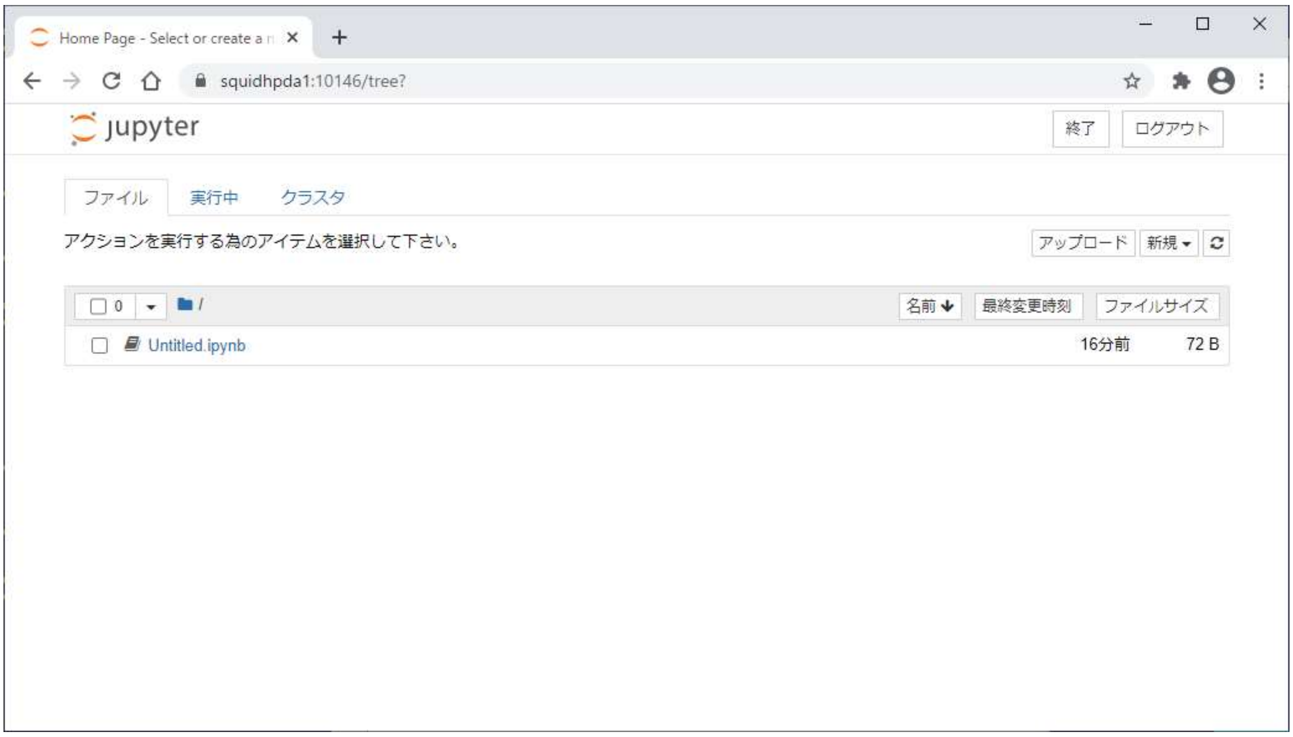
See the documentation on how to enable a password in place of token authentication, if you would like to avoid dealing with random tokens.

① Password or token

Enter the jupyter login token displayed in the terminal in "3.3.2 Starting the jupyter container".

② Login

Click the login button.



If the authentication with the jupyter token is successful, the initial screen of the jupyter notebook will be displayed.

3.3.4. Customize Jupyter Container

The jupyter container is provided without components such as libraries installed so that it can be customized according to the user's own taste. If necessary, please install additional components.

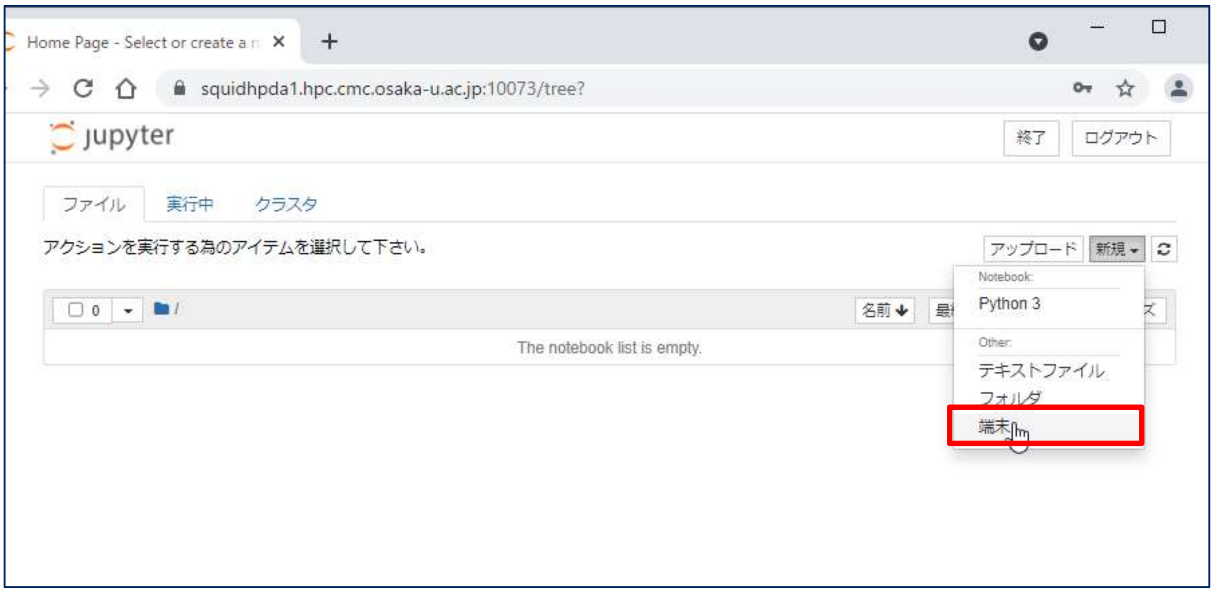
Describes how to install additional components in a jupyter container. Here, we will introduce how to install numpy and matplotlib as an example.

- install numpy

  Work from the state where the initial screen of jupyter Notebook is displayed.

  ① start terminal

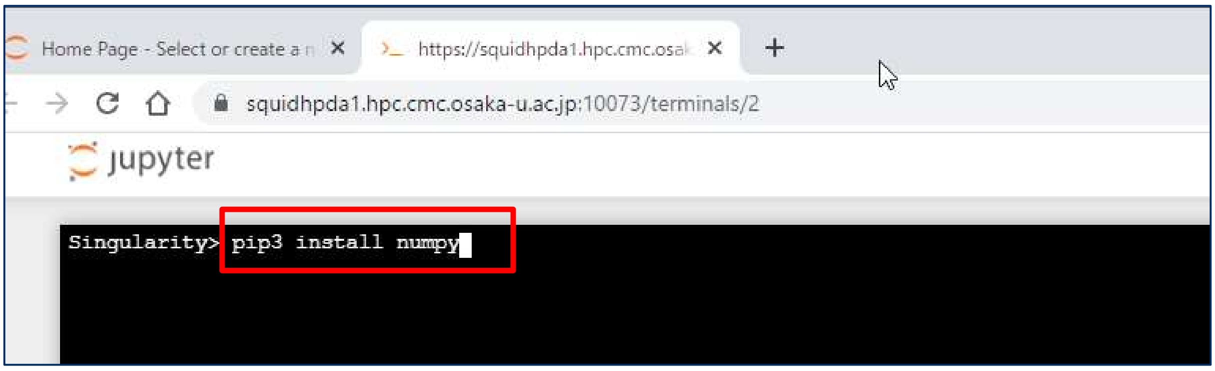
  Click the new button at the top right of the initial screen and select "端末".


Home Page - Select or create a n
squidhpda1.hpc.cmc.osaka-u.ac.jp:10073/tree?
jupyter
終了
ログアウト
ファイル
実行中
クラスタ
アクションを実行する為のアイテムを選択して下さい。
アップロード
新規
名前
The notebook list is empty.
Notebook:
Python 3
Other:
テキストファイル
フォルダ
端末

② execute pip3 command

Enter the following command in the terminal and press the Enter key.

command : pip3 install numpy


Home Page - Select or create a n
https://squidhpda1.hpc.cmc.osak
squidhpda1.hpc.cmc.osaka-u.ac.jp:10073/terminals/2
jupyter

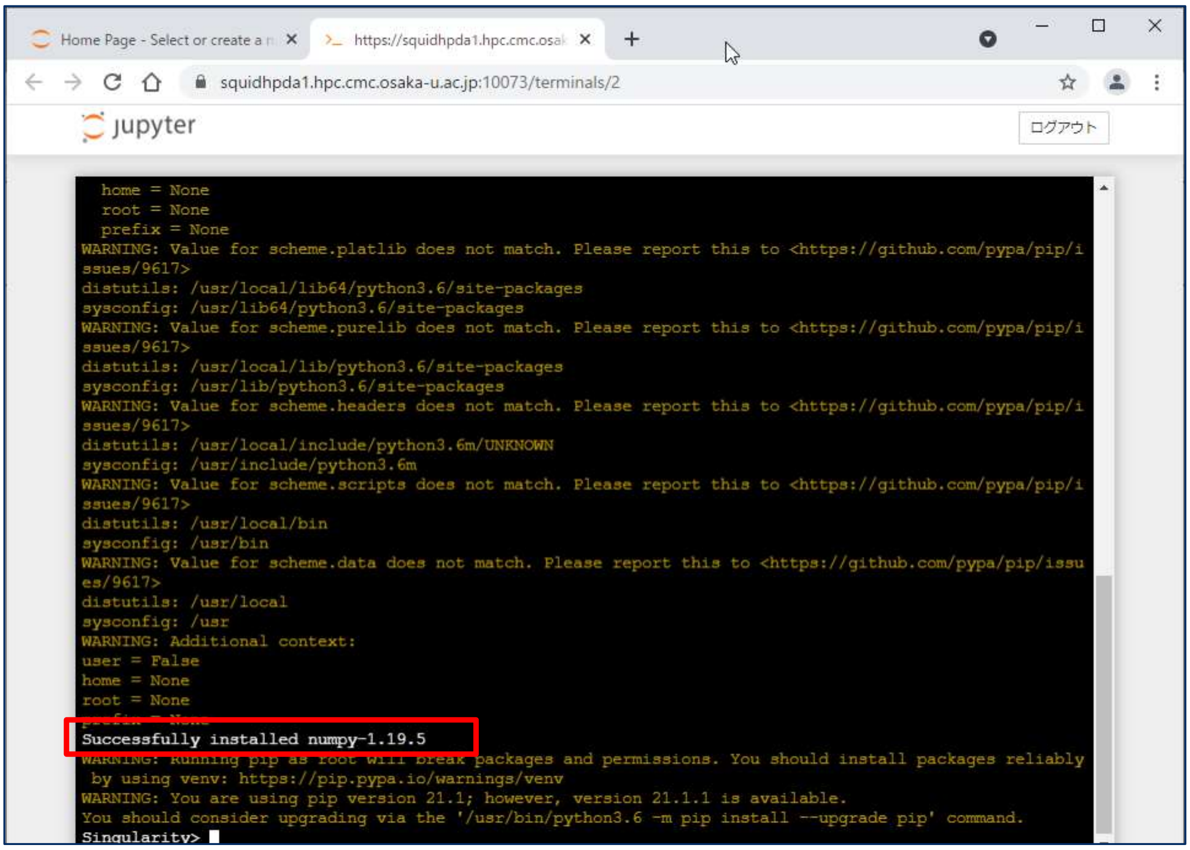
```
Singularity> pip3 install numpy
```

③ check installation resault

Confirm that the installation was successful with the following message.

message : Successfully installed numpy-x.x.x

※x.x.x is displayed numpy version which was installed.


Home Page - Select or create a n
https://squidhpda1.hpc.cmc.osak
squidhpda1.hpc.cmc.osaka-u.ac.jp:10073/terminals/2
jupyter
ログアウト

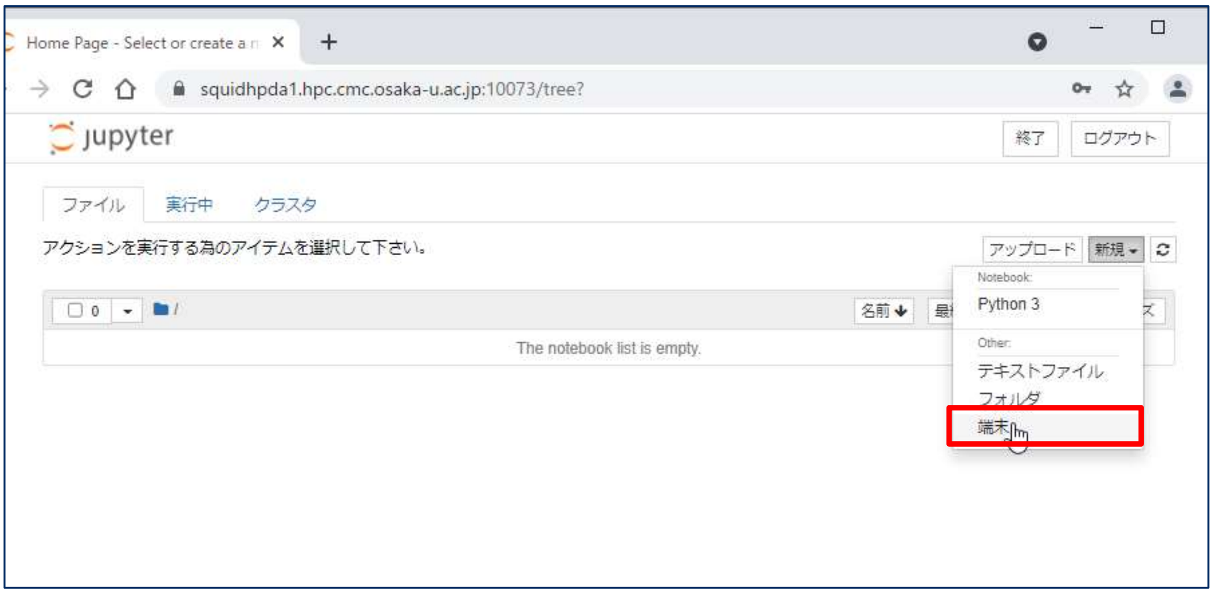
```
  home = None
  root = None
  prefix = None
WARNING: Value for scheme.platlib does not match. Please report this to <https://github.com/pypa/pip/i
ssues/9617>
distutils: /usr/local/lib64/python3.6/site-packages
sysconfig: /usr/lib64/python3.6/site-packages
WARNING: Value for scheme.purelib does not match. Please report this to <https://github.com/pypa/pip/i
ssues/9617>
distutils: /usr/local/lib/python3.6/site-packages
sysconfig: /usr/lib/python3.6/site-packages
WARNING: Value for scheme.headers does not match. Please report this to <https://github.com/pypa/pip/i
ssues/9617>
distutils: /usr/local/include/python3.6m/UNKNOWN
sysconfig: /usr/include/python3.6m
WARNING: Value for scheme.scripts does not match. Please report this to <https://github.com/pypa/pip/i
ssues/9617>
distutils: /usr/local/bin
sysconfig: /usr/bin
WARNING: Value for scheme.data does not match. Please report this to <https://github.com/pypa/pip/issu
es/9617>
distutils: /usr/local
sysconfig: /usr
WARNING: Additional context:
user = False
home = None
root = None
prefix = None
Successfully installed numpy-1.19.5
WARNING: Running pip as root will break packages and permissions. You should install packages reliably
 by using venv: https://pip.pypa.io/warnings/venv
WARNING: You are using pip version 21.1; however, version 21.1.1 is available.
You should consider upgrading via the '/usr/bin/python3.6 -m pip install --upgrade pip' command.
Singularity>
```

- install matplotlib

  Work from the state where the initial screen of jupyter Notebook is displayed.

  ① start terminal

  Click the new button at the top right of the initial screen and select "端末".



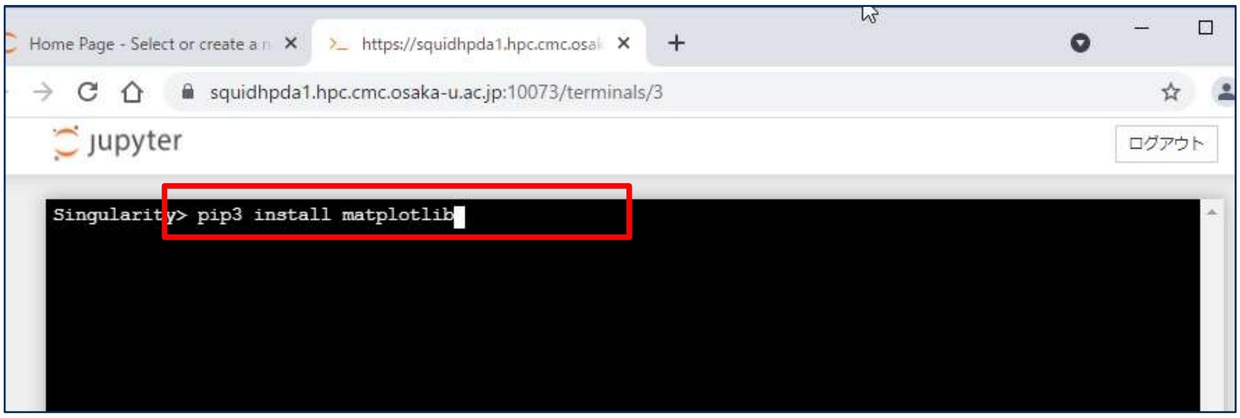
アクションを実行する為のアイテムを選択して下さい。

The notebook list is empty.

② execute pip3 command

Enter the following command in the terminal and press the Enter key.

command : pip3 install matplotlib


Home Page - Select or create a n
https://squidhpda1.hpc.cmc.osak
squidhpda1.hpc.cmc.osaka-u.ac.jp:10073/terminals/3
jupyter
ログアウト

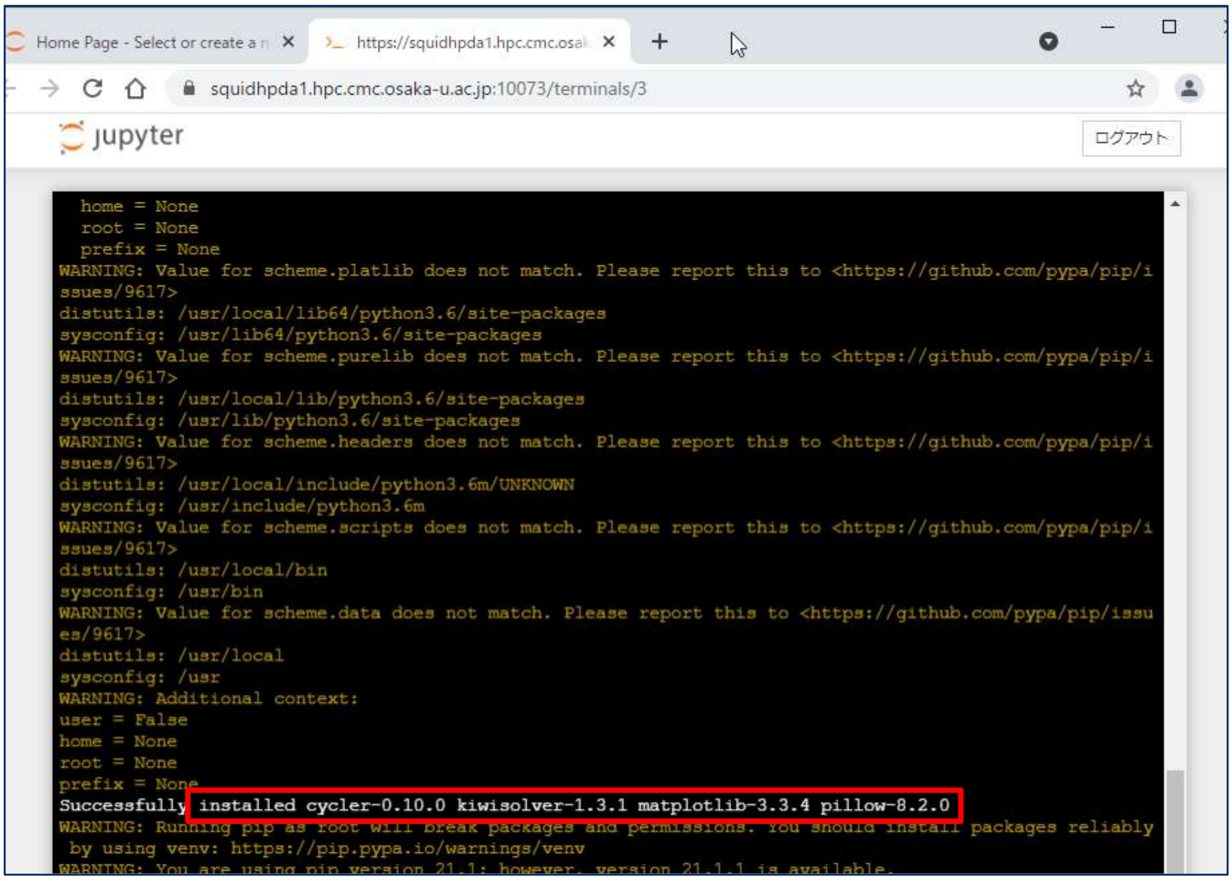
```
Singularity> pip3 install matplotlib
```

③ check installation resault

Confirm that the installation was successful with the following message.

message：Successfully installed cycler-x.x.x kiwisolver-x.x.x matplotlib-x.x.x pillow-x.x.x

※x.x.x is displayed matplotlib version which was installed and dependent packages.


Home Page - Select or create a n
https://squidhpda1.hpc.cmc.osak
squidhpda1.hpc.cmc.osaka-u.ac.jp:10073/terminals/3
jupyter
ログアウト

```
    home = None
    root = None
    prefix = None
WARNING: Value for scheme.platlib does not match. Please report this to <https://github.com/pypa/pip/i
ssues/9617>
distutils: /usr/local/lib64/python3.6/site-packages
sysconfig: /usr/lib64/python3.6/site-packages
WARNING: Value for scheme.purelib does not match. Please report this to <https://github.com/pypa/pip/i
ssues/9617>
distutils: /usr/local/lib/python3.6/site-packages
sysconfig: /usr/lib/python3.6/site-packages
WARNING: Value for scheme.headers does not match. Please report this to <https://github.com/pypa/pip/i
ssues/9617>
distutils: /usr/local/include/python3.6m/UNKNOWN
sysconfig: /usr/include/python3.6m
WARNING: Value for scheme.scripts does not match. Please report this to <https://github.com/pypa/pip/i
ssues/9617>
distutils: /usr/local/bin
sysconfig: /usr/bin
WARNING: Value for scheme.data does not match. Please report this to <https://github.com/pypa/pip/issu
es/9617>
distutils: /usr/local
sysconfig: /usr
WARNING: Additional context:
user = False
home = None
root = None
prefix = None
Successfully installed cycler-0.10.0 kiwisolver-1.3.1 matplotlib-3.3.4 pillow-8.2.0
WARNING: Running pip as root will break packages and permissions. You should install packages reliably
 by using venv: https://pip.pypa.io/warnings/venv
WARNING: You are using pip version 21.1; however, version 21.1.1 is available.
```

3.3.5. Stop Jupyter Container

Describes how to stop the jupyter container on the HPDA front end.

Changes in the running jupyter container will be automatically saved in jupyter.sif.

- Stop jupyter container

  In the directory which you start jupyter container as "3.3.2 Start Jupyter Container", execute the shell script for stopping the jupyter.

```
$ stop_jupyter_container.sh
```

If the jupyter container is successfully stopped, the following message will be displayed in the terminal.

```
$ stop_jupyter_container.sh
INFO:   Starting build...
INFO:   Creating SIF file...
INFO:   Build complete: ./jupyter_tmp.sif
INFO:   Stopping jupyter instance of /var/tmp/rootfs-940268085/root (PID=19101)
$
```

4. Program Development

4.1. Common Matters

In this system, the front end for HPC or the front end for HPDA is used to generate an execution module from program creation.

The flow up to the execution of the basic program is as follows, but this chapter describes the method of "② Generating (compiling) the execution module from the source code".

① Create / edit / modify source code

② Generation (compilation) of execution module from source code

③ Execution of execution module

4.1.1. Use of Compiler / MPI

Various programming languages are available in this system. When using programming languages, the set of environment variables for compiler, MPI and accompanying libraries is organized as a base environment. This section provides an overview of the base environment for each programming language.

(1)C/C++, FORTRAN language

For C/C++ language and FORTRAN language program development, this system is equipped with a compiler suitable for each computer of CPU, VectorEngine, and GPGPU. In addition, parallel execution by thread parallel and MPI parallel is also possible.

To use it, load the modules using Environment Modules according to "3.1.3. Environment Settings". Some modules have a configuration suitable for program development in each computing environment as a recommended environment. The configuration of each recommended environment is as follows.

computing environment	recommended module	compiler	MPI
General-purpose CPU environment	BaseCPU	Intel Parallell Studio	Intel MPI
Vector computing environment	BaseVEC	NEC SDK for VE	NEC MPI
GPGPU computing	BaseGPU	NVIDIA HPC SDK	Open MPI

environment		CUDA	
--	BaseGCC	GNU Compiler	Open MPI

※ You can use BaseGPU modules for General-purpose CPU environment, etc. But you can do more suitable program development for computing environemnt.

To apply the recommended environment for CPU nodes, load the module as follows.

```
$ module avail
-------------------------------- /system/apps/env/base ------------------------------
BaseCPU/2021(default)     BaseVEC/2021     BaseGPU/2021        BaseGCC/2021
BasePy/2021                  BaseR/2021     BaseApp/2021

$ module load BaseCPU/2021
```

Details on the recommended environment settings, compilation commands, and how to compile parallel programs are described in the following chapters.

- General-purpose CPU environment
  Refer to "4.2 compiling a program for General-purpose CPU environment ".
- Vector computing environment
  Refer to "4.3 compiling a program for Vector computing environment ".
- GPGPU computing environment
  Refer to "4.4 compiling a program for GPGPU computing environment ".
- GNU compiler
  Refer to "4.5 How to Use GNU Compiler Collection".

(2) Other programing languages

Base environments are also available for programming languages other than C / C ++ and FORTRAN. The available programming language environments are as follows.

computing environment	module name	version	description	remarks

Python	BasePy	3.6	Python language	Python2 also availabe
R	BaseR	4.0.3	R language	
JAVA	BaseJDK	11	JAVA language	OpenJDK
Julia	BaseJulia	1.6.1	Julia language	

To start these programming language environments, do the following:

A) Python

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

B) R

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

C) JAVA

```
$ module load BaseJDK
$ java --version
```

D) Julia

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

The details of the Python language are described below.

- Python lanuage

  Refer to "4.7 How to Use Python".

4.1.2. Use of The Library

For the libraries that can be used in this system, set environment variables with the module command.

Also, the modules of each library are versioned by <module name> / <version>.。

The available libraries and language modules are as follows.

library and module	module name					
	CPU	VEC	GPU	GCC	Py	R
Intel MKL (※)	○	-	-	-	-	-
NEC Numeric Library Collection (※)	-	○	-	-	-	-
GNU Scientific Library	-	-	-	○	-	-
HDF5	○	○	○	-	-	-
NetCDF	○	○	○	-	-	-
Parallel netcdf	○	○	○	-	-	-
Keras	-	-	-	-	○	-
PyTorch	-	-	-	-	○	-
TensorFlow	-	-	-	-	○	-
pbdR	-	-	-	-	-	○

※ BLAS, LAPACK and ScaLAPACK are included in the "NEC Numeric Library Collection" and "Intel MKL".

The specific contents of how to use the library are described in the section "How to Use Libraries" in the following section.

"4.2 compiling a program for General-purpose CPU environment"

"4.3 compiling a program for Vector computing environment"

"4.4 compiling a program for GPGPU computing environment"

4.1.3. How to use Development Support Tool

This system has the following development support tools. Use the development support tool by logging in to the interactive queue.

Development support tool	General-purpose CPU	Vector computing	GPGPU computing
Arm Forge DDT/MAP	○	-	○
NEC Parallel Debugger	-	○	-

- How to start Arm Forge DDT / MAP Arm Forge DDT/MAP の起動方法

  Arm Forge is a development assistance tools suite that includes debugging, profilers, optimization tools, and more for C / C ++ / FORTRAN applications on Linux.

The main components are Arm DDT, which is a debugger, and Arm MAP, which is a profiler for parallel applications.Follow the procedure below to start it.

```
$ module load BaseCPU/2021
$ module load ArmForge/21.0
$ ddt
```

- How to start NEC Parallel Debugger

  NEC Parallel Debugger is a debugging tool for Vector Engine, and provides a plug-in for debugging MPI applications for Eclipse PTP (Parallel Tools Platform). NEC Parallel Debugger is started by the following procedure.

(1) Start the X server (Xming, VcXsrv, etc.) in advance on the host terminal that logs in to the front-end environment.

(2) Login to the loan-end environment with X forwarding enabled.

(3) Start Eclipse PTP.

```
$ module load BaseVEC/2021
$ module load EclipsePTP/4.7.3a
$ eclipse
```

(4) The Eclipse Launcher window will open. Enter the Workspace directory (the directory that stores the make environment of the application to be debugged) and click "Launch".

For details on how to use it, refer to the following page of NEC Aurora Forum (https://www.hpc.nec).

Aurora Forum（https://www.hpc.nec)
LEARN MORE ＞ NEC SDK ＞ NEC Parallel Debugger User's Guide

4.2. compiling a program for General-purpose CPU environment

This section deescribes how to compile a program for general-purpose CPU nodes. Also, this section describes how to compile on Base CPU, which is the recommended environment for general-purpose CPU nodes. Please load the BaseCPU environment when compiling the program.

```
$ module load BaseCPU/2021
```

4.2.1. Serial Execution

Compile carries out on the front end server.

The compile commands for each program language are as follows.

Program Language	Intel Parallel Studio
FORTRAN	ifort
C	icc
C++	icpc

The following is a compilation example. With the "-o" option, you can specify the name of the executable file to be created. If not specified, the file name will be created as a.out.

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

```
$ module load BaseCPU/2021

$ ifort -o sample sample.f90
$ icc -o sample sample.c
$ icpc -o sample sample.cpp
```

4.2.2. Thread Parallel Execution

To execute in parallel threads, specify the following options at compile time.

Parallel model	Intel Parallel Studio
OpenMP	-qopenmp
Automatic parallel	-parallel

The number of execution threads is specified by defining the environment variable OMP_NUM_THREADS = <number of execution threads> at runtime.

The following is a compilation example. First you need to load the Intel compiler environment variable settings module with the module command

```
$ module load BaseCPU/2021

$ ifort -o sample_omp -qopenmp sample_omp.f90
$ ifort -o sample_smp -parallel sample_smp.f90
```

4.2.3. MPI Parallel Execution

Intel MPI is available for MPI environment

The compile command is as follows

programming language	For Intel MPI
FORTRAN	mpiifort
C	mpiicc
C++	mpiicpc

The following is a compilation example

```
$ module load BaseCPU/2021

$ mpiifort -o sample_mpi sample_mpi.f90
$ mpiicc   -o sample_mpi sample_mpi.c
$ mpiicoc -o sample_mpi sample_mpi.cpp
```

4.2.4. How to Use Libraries

Describes how to use the library in a program for general-purpose CPU computing environment.

(1) Intel MKL

Intel Math Kernel Library is a numerical calculation library developed by Intel. Various libraries including BLAS, LAPACK, ScaLAPACK, etc. can be used.

In this system, you can use MKL with Intel compiler environment loaded by the BaseCPU module. The following is a link example of the library in this system.

```
# Link MKL with serial execution
$ module load BaseCPU/2021
$ icc -mkl=sequential sample.c                       # dynamic link
$ icc -mkl=sequential -static-intel sample.c         # static link

# Link MKL with thread parallel execution
$ module load BaseCPU/2021
$ icc -mkl=parallel sample.c                         # dynamic link
$ icc -mkl=parallel -static-intel sample.o           # static link

# Link MKL with MPI parallel execution
$ module load BaseCPU/2021
$ mpiicc -mkl=cluster sample.c                       # dynamic link
$ mpiicc -mkl=cluster -static-intel sample.c         # static link
```

For details on how to use Intel MKL, refer to the official online documentation.

- Modul Intel Math Kernel Library – Documentation

  https://software.intel.com/content/www/us/en/develop/articles/intel-math-kernel-library-documentation.html

(2) HDF5

HDF5 is a library for generating files that can store a large amount of data in a hierarchical structure. It is prepared for each calculation environment. When linking the library, add the -lhdf5 option at compile time. The following is an example of a library link.

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

(3) NetCDF

NetCDF is a binary file format that is widely used as a common data format for the scientific multidimensional data. In this system it is prepared so that it can be used from C language / C ++ language / FORTRAN language for each calculation environment. When linking the library, add options such as -lnetcdf at compile time. The following is an example of a library link.

・**C language**

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

・C++ language

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

・FORTRAN language

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

(4) PnetCDF

PnetCDF is a library for simultaneous access to NetCDF format files from parallel programs. In this system, it is prepared so that it can be used from C language / C ++ language / FORTRAN language for each calculation environment. When linking the library, add options such as -lpnetcdf at compile time. The following is an example of a library link.

・C language

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

・C++language

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

・FORTRAN language

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

4.3. compiling a program for Vector computing environment

Describes how to compile a program for vector nodes. This section describes how to compile on BaseVEC, which is the recommended environment for vector nodes. Please load the BaseVEC environment when compiling the program.

```
$ module load BaseVEC/2021
```

4.3.1. Serial Execution

NEC compiler is used for serial execution for VE. The compile command is as follows. Run the compilation on the front end.

Programming Language	NEC SDK for VE
Fortran	nfort
C	ncc
C++	nc++

The main compile options are: For detailed option information and environment variable information, use the man command.

Optino	Function
-On (n=0~4)	Optimization options 0: Do not apply optimization, automatic vectorization, parallelization, inline expansion 1: Apply optimization / automatic vectorization without side effects 2: Apply optimization / automatic vectorization with side effects (default value) 3: Apply optimization / automatic vectorization with side effects and multiple loop optimization 4: Apply maximum optimization / automatic vectorization with side effects that deviate from the language specifications
-o	Specify the name of the executable file to be created
-mparallel	Enable automatic parallel

	The number of execution threads is specified in the environment variable OMP_NUM_THREADS at the time of execution.
-fopenmp	Enable OpenMP functionality The number of execution threads is specified in the environment variable OMP_NUM_THREADS at the time of execution.
-finline-functions	Perform automatic inline expansion
-p	Enable profiler Performance measurement result file gmon.out is output after execution Display the contents of gmon.out with the ngprof command
-proginf	Generate an executable file for the PROGINF function (default) Output PROGINF information when the environment variable VE_PROGINF = YES (or DETAIL) is set at run time
-ftrace	Generate an object file and an executable file for the FTRACE function
-traceback	When the environment variable VE_TRACEBACK is set at the time of execution, an object file and an executable file that output traceback information are generated.
-g	Generate minimal information such as line number information required for backtrace output
-report-all	Output diagnostic message list and edit list
-report-diagnostics	Output diagnostic message list
-report-format	Output edit list

The following is an example of compiling each program language using the NEC compiler. First load the BaseVEC module and then execute various compile commands.

```
$ module load BaseVEC/2021
$ nfort -o sample sample.f90
$ ncc -o sample sample.c
$ nc++ -o sample sample.cpp
```

4.3.2. Thread Parallel Execution

The NEC compiler can use two thread parallel executions, automatic parallel and OpenMP parallel.

For automatic parallel, specify "-mparallel" at compile time, and for OpenMP parallel, specify "-fopenmp" at compile time. The number of threads is specified in the environment variable OMP_NUM_THREADS at run time.。

ex1) 8-thread execution by automatic parallel

```
$ module load BaseVEC/2021
$ nfort -mparallel test.f              # Compile with automatic parallelism enabled
$ export OMP_NUM_THREADS=8             # Specify the number of execution threads (※)
$ ./a.out                              # Run executfile (※)
```

(※) is described in the job script and executed in batch format.

ex2) OpenMP 8 threads execution in parallel

```
$ module load BaseVEC/2021
$ nfort -fopenmp test.f                # Compile with automatic parallelism enabled
$ export OMP_NUM_THREADS=8             # Specify the number of execution threads (※)
$ ./a.out                              # Run executfile (※)
```

(※) is described in the job script and executed in batch format.

4.3.3. How to Compile When Performing MPI Parallel Execution

NEC MPI is used for MPI execution for VE. The compile command is as follows. Compile on the front end.

Check the compiler option information by specifying the "-help" option.

Programming Language	NEC MPI
Fortran	mpinfort
C	mpincc
C++	mpinc++

ex) 8 processes (4VE x 2 processes) executed by MPI parallel

```
$ module load BaseVEC/2021
$ mpinfort mpi_test.f                    # Compile
$ mpirun -np 8 -ve 0-3 ./a.out           # Specify execution by mpirun (※1)
$ mpirun -np 8 -venode -nn 4 ./a.out     # Specify execution by mpirun (※2)
$ mpirun -np 8 -venode -node 0-3 ./a.out  # Specify execution by mpirun (※3)
```

(※1)(※2) is described in the job script and executed in batch format.

4.3.4. How to Use Libraries

Describes how to use the library in a program for vector calculation environment.

(1) NEC Numeric Library Collection

NEC NLC is a numerical calculation library optimized for the SX-Aurora TSUBASA system in the vector calculation environment. It includes ASL and MathKeisan of SX-ACE system, and various libraries such as BLAS, LAPACK, ScaLAPACK, etc. can be used.

In this system, necnlc module (32bit integer version) and necnlc-mpi module (32bit integer + distributed memory parallel) are prepared in the environment of BaseVEC module. The following is an example of linking the library in this system.

・**Without distributed memory parallel**

```
# Statically link ASL native interface 32bit integer library
$ module load BaseVEC/2021
$ module load necnlc/2.3.0
$ nfort -c sample.f                               # compile
$ nfort -lasl_sequential -static sample.o         # link
```

・**With distributed memory parallel**

```
# Static linking of ScaLAPACK distributed memory parallel library
$ module load BaseVEC/2021
$ module load necnlc-mpi/2.3.0
$ mpinfort -c sample.f                                       # compile
$ mpinfort -lscalapack -llapack -lblas_sequential  ¥   # link
             -static sample.o
```

＊ It is also possible to make a dynamic link without adding the -static option.

For details on how to use NEC NLC, refer to the official online document.

- NEC Numeric Library Collection 2.3.0 User's Guide
  https://www.hpc.nec/documents/sdk/SDK_NLC/UsersGuide/main/en/index.html

(2) HDF5

HDF5 is a library for generating files that can store a large amount of data in a hierarchical structure. It is prepared for each calculation environment. When linking the library, add the -lhdf5 option at compile time. The following is an example of a library link.

```
$ module load BaseVEC
$ module load hdf5
$ ncc -o h5-sample h5-sample.c -lhdf5 -ldl
```

(3) NetCDF

NetCDF is a binary file format that is widely used as a common data format for the scientific multidimensional data. In this system it is prepared so that it can be used from C language / C ++ language / FORTRAN language for each calculation environment. When linking the library, add options such as -lnetcdf at compile time. The following is an example of a library link.

・C language

```
$ module load BaseVEC
$ module load netcdf-c
$ ncc -o ncdf-sample ncdf-sample.c -ldl -lnetcdf -lhdf5_hl -lhdf5 -lsz
```

・C++ language

```
$ module load BaseVEC
$ module load netcdf-cxx
$ nc++ -o ncdf-sample ncdf-sample.cpp ¥
          -lnetcdf_c++4 -lnetcdf -lhdf5 -lhdf5_hl -ldl -lsz
```

・FORTRAN language

```
$ module load BaseVEC
$ module load netcdf-fortran
$ nfort -o ncdf-sample ncdf-sample.f90 -lnetcdff -lnetcdf -lhdf5 -lhdf5_hl -ldl -lsz
```

(4) PnetCDF

PnetCDF is a library for simultaneous access to NetCDF format files from parallel programs. In this system, it is prepared so that it can be used from C language / C ++ language / FORTRAN language for each calculation environment. When linking the library, add options such as -lpnetcdf at compile time. The following is an example of a library link.

・C language

```
$ module load BaseVEC
$ module load pnetcdf-c
$ mpincc -o pncdf-sample pncdf-sample.c -lpnetcdf -lmpi
```

・C++ language

```
$ module load BaseVEC
$ module load pnetcdf-cxx
$ mpinc++ -o pncdf-sample pncdf-sample.cpp -lpnetcdf -lmpi -lmpi++
```

・FORTRAN language

```
$ module load BaseVEC
$ module load netcdf-fortran
$ mpinfort -o pncdf-sample pncdf-sample.f90 -lpnetcdf -lmpi
```

4.4. compiling a program for GPGPU computing environment

4.4.1. Serial Execution

Compile is done on the front end.

The compile commands for each program language are as follows.

Programming Language	NVidia HPC SDK
FORTRAN	nvfortran
C	nvc
C++	nvc++
CUDA	nvcc

The following is a compilation example. With the "-o" option, you can specify the name of the executable file to be created. If not specified, the file name will be created as a.out.

First you need to load the NVidia HPC SDK environment variable settings module with the module command. The standard version of the compiler is automatically loaded when you load the BaseGPU module.

```
$ module load BaseGPU/2021

$ nvfortran -o sample sample.f90
$ nvc -o sample sample.c
$ nvc++ -o sample sample.cpp
```

4.4.2. Thread Parallel Execution

To execute in parallel threads, specify the following options at compile time.

Parallel model	NVidia HPC SDK
OpenMP	-mp
Automatic parallel	-Mconcur

The number of execution threads is specified by defining the environment variable OMP_NUM_THREADS = <number of execution threads> at runtime.

The following is a compilation example. First you need to load the NVidia HPC SDK environment variable settings module with the module command.

```
$ module load BaseGPU/2021

$ nvfortran -o sample_omp -mp sample_omp.f90
$ nvfortran -o sample_smp -Mconcur sample_smp.f90
```

4.4.3. MPI Parallel Execution

OpenMPI is available for MPI environment.

The compile command is as follows.

programming language	For Intel MPI
FORTRAN	mpifort
C	mpicc
C++	mpic++

The following is a compilation example.

```
$ module load BaseGPU/2021

$ mpifort -o sample_mpi sample_mpi.f90
$ mpicc   -o sample_mpi sample_mpi.c
$ mpic++ -o sample_mpi sample_mpi.cpp
```

4.4.4. Compile For CUDA

Numerical calculations using GPGPU are possible with SQUID's GPU node group. This page describes the procedure for compiling your own program using CUDA with SQUID.

As a development environment for GPU, NVCC Compiler is available for NVidia HPC SDK. In addition, the Fortran compiler (nvfortran) supports CUDA Fortran. The command to compile a program using CUDA is as follows. For CUDA C, the program extension is ".cu" or ".CU", and for CUDA Fortran it is ".cuf" or ".CUF".

Programming Language	NVidia HPC SDK
CUDA Fortran	nvfortran
CUDA C	nvcc

The following is a compilation example.

- CUDA Fortran compiler

```
$ module load BaseGPU/2021
$ nvfortran -o sample_cuda sample_cuda.cuf
```

- CUDA C compiler

```
$ module load BaseGPU/2021
$ nvcc -o sample_cuda sample_cuda.cu
```

4.4.5. Options for OpenACC

Specify the following options when compiling a program using OpenACC

Programming Language	NVidia HPC SDK
OpenACC C	-acc

The following is a compilation example.

```
$ module load BaseGPU/2021

$ nvc -o sample_acc -acc sample_openacc.c
```

4.4.6. How to Use Libraries

Describes how to use the library in the program for GPGPU computing environment.

Regarding the use of the library, there are notes on the FORTRAN compiler (nvfortran) for GPGPU computing environment. The nvfortran does not support to change the search path for include files according to environment variables like CPATH. Therefore, the CPATH environment variable set by the module command must be passed with the -I option at compile time. The execution example is as follows.

```
$ nvfortran -o sample sample.f90   -I${CPATH}
```

(1) HDF5

HDF5 is a library for generating files that can store a large amount of data in a hierarchical structure. It is prepared for each calculation environment. When linking the library, add the -lhdf5 option at compile time. The following is an example of a library link.

```
$ module load BaseGPU
$ module load hdf5
$ nvc -o h5-sample h5-sample.c -lhdf5 -R${LD_RUN_PATH}
```

※ Since it conflicts with the OS standard hdf5 package, add the -R $ {LD_RUN_PATH} option and prioritize the corresponding package.

(2) NetCDF

NetCDF is a binary file format that is widely used as a common data format for the scientific multidimensional data. In this system it is prepared so that it can be used from C language / C ++ language / FORTRAN language for each calculation environment. When linking the library, add options such as -lnetcdf at compile time. The following is an example of a library link.

・C language

```
$ module load BaseGPU
$ module load netcdf-c
$ nvc -o ncdf-sample ncdf-sample.c -lnetcdf
```

・C++ language

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

・FORTRAN language

```
$ module load BaseGPU
$ module load netcdf-fortran
$ nvfortran -o ncdf-sample ncdf-sample.f90  -I${CPATH} -lnetcdff -lnetcdf
```

(3) PnetCDF

PnetCDF is a library for simultaneous access to NetCDF format files from parallel programs. In this system, it is prepared so that it can be used from C language / C ++ language / FORTRAN language for each calculation environment. When linking the library, add options such as -lpnetcdf at compile time. The following is an example of a library link.

・C language

```
$ module load BaseGPU
$ module load pnetcdf-c
$ mpicc -o pncdf-sample pncdf-sample.c -lpnetcdf
```

・**C++ language**

```
$ module load BaseGPU
$ module load pnetcdf-cxx
$ mpic++ -o pncdf-sample pncdf-sample.cpp -lpnetcdf
```

・**FORTRAN language**

```
$ module load BaseGPU
$ module load pnetcdf-fortran
$ mpifort -o pncdf-sample pncdf-sample.f90 -I${CPATH} -lpnetcdf
```

4.5. How to Use GNU Compiler Collection

You can use the GNU Compiler Collection. This section describes how to compile on BaseGCC base environment. Please load the BaseGCC environment in advance when compiling the program.

```
$ module load BaseGCC/2021
```

4.5.1. Serial Execuion

Compile is done on the front end.

The compile commands for each program language are as follows.

Programming Language	GNU Compiler Collection
FORTRAN	gfortran
C	gcc
C++	g++

The following is a compilation example. With the "-o" option, you can specify the name of the executable file to be created. If not specified, the file name will be created as a.out.

First you need to load the Intel compiler environment variable settings module with the module command. Loading the BaseGCC module automatically loads the standard version of the Intel compiler.

```
$ module load BaseGCC/2021

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

4.5.2. Thread Parallel Execution

To execute in parallel threads, specify the following options at compile time.

Parallel model	GNU Compiler Collection
OpenMP	-fopenmp

The number of execution threads is specified by defining the environment variable OMP_NUM_THREADS = <number of execution threads> at runtime.

The following is a compilation example. First you need to load the GNU compiler environment variable settings module with the module command

```
$ module load BaseGCC/2021


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

4.6. How to Use Containers

This section describes instructions getting various base container images registered in the local registry, customize the container, and build.

For an overview of how to use the container, see "1.5.4 Use container"

4.6.1. Preparation of container image

The container image can be obtained from the local registry in SQUID or the public registry on the Internet. You can also upload your own container image to SQUID.

➢ **Transfer from local workstation to system**

The container image prepared as an image file can be transferred with the scp command etc. in the same way as a normal file. For details on how to transfer files to this system, refer to "2.3　File Transfer".

➢ **Get container image from SQUID local registry**

This section describes the procedure for acquiring the container image published in SQUID's local registry. As an example, to get the container image called test registered in the local registry / master_image, execute the following command.

```
$ singularity build test.sif ¥
            oras://cntm:5000/master_image/test:1.0
```

The command format is as follows.

```
$ singularity build <image file name> ¥
            oras://cntm:5000/<image path>:<tag>
```

If the container image is successfully acquired, the container image (test.sif in the above example) will be created in the current directory.

➢ **Get container image from Singularity Library**

This section describes the procedure for acquiring the singularity container image from the singularity library.

As an example, to get the container image of centos from the singularity library, execute the following command.

```
$ singularity build centos.sif ¥
        library://emmeff/centos/centos:8
```

The command format is as follows.

```
$ singularity build <image file name> ¥
         library://<image path>:<tag>
```

If the container image is successfully acquired, the container image (centos.sif in the above example) will be created in the current directory.

- **Get container image from Docker Hub**

Here are the steps to get a Docker container image from Docker Hub and save it as a singularity container image. As an example, to get the container image of centos from Docker Hub, execute the following command.

```
$ singularity build centos.sif   ¥
        docker://docker.io/library/centos:latest
```

The command format is as follows.

```
$ singularity build <image file name> ¥
        docker://docker.io/<image path>:<tag>
```

If the container image is successfully acquired, the container image (centos.sif in the above example) will be created in the current directory.

- **Appendix : Get an image from NVIDIA GPU CLOUD (NGC)**

NVIDIA GPU CLOUD is a website provided by NVidia, and documents for using GPGPU and various utilities are published. The container image is also open to the public in the form of Docker container, and can be obtained from this system.

※To obtain it, you need to register as an NGC user yourself.

- NGC Catalog Containers
  https://ngc.nvidia.com/catalog/containers

In the following example, the procedure is explained on the assumption that user registration on NGC and issuance of API Key for docker container have been completed.

First, set the user name and API Key for NGC login in the environment variables. The user name is fixed to $ outhtoken, and the API Key is the Key generated in the NGC site.

```
$ export SINGULARITY_DOCKER_USERNAME='$oauthtoken'
$ export SINGULARITY_DOCKER_PASSWORD=<API Key>
```

Then get the container image. As an example, to get HPC-Benchmarks 21.4-hpl from NGC, execute the following command.

```
$ singularity build hpc-benchmarks:21.4-hpl.sif ¥
        docker://nvcr.io/nvidia/hpc-benchmarks:21.4-hpl
```

The command format is as follows.

```
$ singularity build <image file name> ¥
        docker://nvcr.io/nvidia/<image path>:<tag>
```

If the container image is successfully acquired, the container image (hpc-benchmarks: 21.4-hpl.sif in the above example) will be created in the current directory.

4.6.2. Customize and Build Container Images

Describes how to customize and build the container image.

There are two ways to customize and build a container image in SQUID: one is to customize and build via sandbox, and the other is to write the customization contents in the def file and build.

➢ How to customize via sandbox

Learn how to customize and build via sandbox.

(1) Create sandbox

First, create a sandbox from the container image. Go to the directory where you want to create the sandbox and run the command to create the sandbox. At this time, the group ownership of the directory to be moved and the primary group must be the same, so the command to be created differs between the home area and the extended area or high-speed area. With the -f (fakeroot) option, the created user may not be able to completely remove sandbox. In this case, delete the sandbox according to the procedure described in (4) Delete sandbox.

As an example, here is an example of command execution when creating a sandbox called test when there is an image file called test.sif in a directory called mySandbox.

・When creating a sandbox in the home area

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

・When creating a sandbox in the extended area / high-speed area

```
$ cd /sqfs/work/<group name>/<user name>/mySandbox
$ newgrp <group name>
$ singularity build -f --sandbox --fix-perms test test.sif
```

The format of the singularity command is as follows.

```
$ singularity build -f --sandbox --fix-perms <sandbox name> <image file name>
```

(2) Start container

Then start the sandbox as a container.

```
$ singularity run -f -w test
```

The format of the singularity command is as follows.

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

If the container starts successfully, you will be prompted for Singularity as shown below.

```
Singularity>
```

From the above prompt, add a package using dnf or pip. The commands used to operate the package depend on the OS distribution stored in the container.

After customizing the container image, stop the container with the following command.

```
Singularity> exit
```

(3) Build image

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

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

The format of the singularity command is as follows.

```
$ singularity build -f <sif file name> <sandbox name>
```

If the build is successful, a sif file will be created in the current directory.

(4) Delete sandbox

Normally, sandbox can be deleted with the rm command. For 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 名>
```

However, depending on the container image, the user who created the sandbox may not be able to completely delete the sandbox. In this case, it is possible to start any container with privilege and delete sandbox from the container.

For example, if you want to delete the sandbox called test in the directory called mySandbox using the container image of alpine, execute the following command.

```
$ singularity exec -f --bind ~/mySandbox library://alpine rm -r test
```

The command format is as follows.

```
$ singularity exec -f --bind <path to sandbox> <any container name> ¥
  rm -r <sandbox name>
```

➢ How to describe the customization contents in the def file

Describe the customization contents in the def file and explain how to customize at build time.

(1) Creating a def file

First, create a def file to use for the build.

As an example, if you want to customize the test registered in the local registry as the base container image, the def file will be as follows.

```
Bootstrap: oras
From: cntm:5000/master_image/test:1.0

%files
        ./test.conf /opt/test.conf
        ./test_start.sh /opt/test_start.sh

%post
        dnf install -y net-tools
        chmod 755 /opt/test_start.sh

%runscript
        /opt/test_start.sh
```

- Overview of def file

  (ア)　Bootstrap、From

  Describe the type and location of the base image

  (イ)　%file

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

  (ウ)　%post

  Describe the command for customization

  (エ)　%runscript

  Describes the process to be automatically executed when the container is started.

  ※*For details on the def file, refer to the singularity manual.
  https://sylabs.io/guides/3.7/user-guide/definition_files.html

(2) Buid image

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

As an example, to build using a def file called test.def and create test.sif, run the following command.

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

The format of the singularity command is as follows.

```
$ singularity build -f <sif file name> <def file name>
```

If the build is successful, a sif file will be created in the current directory.

4.7. How to Use Python

In this system, Python3 and Python2 is available as the Python language environment. When using the Python language, load BasePy base environment module.

```
$ module load BasePy/2021
```

The standard Python environment when BasePy is loaded is Python3. When using Python2, load additional modules of Python2.

```
$ module load BasePy/2021
$ module load python2/2.7
```

Also, when using GPU-compatible Python3, switch modules.

```
$ module load BasePy/2021
$ module --force switch python3/3.6 python3/3.6.GPU
$ module load BaseGPU/2021
$ module load cudnn/8.2.0.53
```

4.7.1. Use in interactive mode

The Python language can use the interactive mode to check the execution result interactively while inputting the python language on the CLI. The following is an example of starting, executing, and terminating the interactive mode on this system.

(1) Python3

```
# load modules
$ module load BasePy

# start intreractive mode
$ python3
[GCC Intel(R) C++ gcc 8.3.1 mode] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

```
# do Python language
>>> print("hellow python")          # Execution content
hello python                        # Execution result

# end intreractive mode
>>> exit()
$
```

(2) Python2

```
# load modules
$ module load BasePy
$ module load python2

# start intreractive mode
$ python2
Python 2.7.18 (default, Apr 19 2021, 13:14:04)
[GCC Intel(R) C++ gcc 8.3.1 mode] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>>

# do Python language
>>> print "hello python"            # Execution content
hello python                        # Execution result

# end intreractive mode
>>> exit()
$
```

4.7.2. Execute Program and Script

It is also possible to batch execute Python programs (scripts) created with a text editor, etc., just like a normal programming language. An example of the execution procedure of a Python program on this system is as follows.

(1) Python3

```
# load modules
$ module load BasePy
```

```
# execute program
$ python3   sample.py
```

(2) Python2

```
# load modules
$ module load BasePy
$ module load python2

# execute program
$ python2 sample.py
```

4.7.3. Add Python Module

As for the Python language, many Python modules are published on the PyPI (Python Package Index). When additional modules are needed, you can additionally install it to your home directory.

- PyPI
  https://pypi.org/

Use the pip command (pip3 / pip2) to install additional modules. The main options for the pip command are:

コマンド	説明
pipX list	display list of installed python modules
pipX install [pymod]	install python module
pipX show [pymod]	display details of specified python module
pipX uninstall [pymod]	uninstall python module

* The "pipX" part is "pip3" for python3 and "pip2" for python2.

In addition to the Python module that is maintained as standard in this system, the execution example when additionally installing by the user is as follows. (For example: installation of dumper module)

(1) Python3

```
# load module
$ module load BasePy
```

```
# install additional module
$ pip3 install dumper
```

(2) Python2

```
# load module
$ module load BasePy
$ module load python2

# install additional module
$ pip2 install dumper
```

4.7.4. Using python modules

In the Python language, you can use the TesorFlow, Keras, and Pytorch libraries.

Each sample program is as follows.

(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')
```

The execution method is as follows.

```
# load python environment
$ module load BasePy

# run TensorFlow
$ python sample_tensorflow.py

# run Keras
$ python sample_keras.py

# run Pytorch
```

```
$ python sample_pytorch.py
```

5. Program Execution Method

5.1. Common Items

5.1.1. Explanation of Job Management System

In SQUID, the job management system "NEC NQSV" enables batch use and conversational use of computational resources. Request job execution (submit a job) from the front-end node to the job management system. For the requested job request, the priority of other job requests, the amount of required resources, the usage status of the large-scale computer system, etc. are taken into consideration and judged by the job management system, and the execution order is determined. Since the large-scale computer system of this center is supposed to be shared by a large number of users, such a batch processing method is adopted.

※ NQSV "Request" and "Job"

The following terms are used as the concept of the job management system "NQSV".

・Request: A unit of batch usage requested by the user from the front end

・Job: A unit of execution in a request that runs on an individual compute node

In this document, the term "job" is used to mean the unit of batch usage (corresponding to NQSV request) requested from the front end, considering the ease of understanding including other systems.

5.1.2. Interactive Job Submission Method

Interactive job is a job to use compute nodes conversationally (interactively). If you want to use interactive job, use the qlogin command.

A sample conversation job submission command (example of executing a general-purpose CPU node) is described below.

```
$ qlogin -q INTC --group=G01234 [options]
              ↑interactive queue name
                       ↑group name
```

When you execute the qlogin command, the request ID is numbered and displayed on the stdout as shown below.

```
Request 1310.sqd submitted to queue: INTC.
Waiting for 1310.sqd to start.
```

5.1.3. Job Submission Method

To submit a job from the front end, follow the procedure below.

① Create a job script file

② Job request to the job management system

(1) Creating A Job Script File

A job script file is a file required to describe a job request from a user. The user writes an execution command to perform the calculation in this job script, and also describes the amount of resources required for the job management system as an option (#PBS).

A sample job script (example of serial execution on a general-purpose CPU node) is described below.

```
#!/bin/bash
#------- qsub option -----------
#PBS -q SQUID
# ↑ Specifying the queue name to submit a batch request
#PBS --group=G01234
# ↑ Group name to be charged
#PBS -l elapstim_req=01:00:00
# ↑ Maximum job execution time request value 1 hour example
#PBS -l cpunum_job=76
# ↑ Required value for the number of CPU cores to use
#PBS -m b
# ↑ Send an email at the start of batch request execution
#PBS -M user@hpc.cmc.osaka-u.ac.jp
# ↑ Destination address

#------- Program execution -----------
module load BaseCPU/2021
# ↑ Load the base environment

cd $PBS_O_WORKDIR
# ↑ Move to the current directory when qsub is executed
./a.out
# ↑ Program execution
```

(2) Job Request to The Job Management System

Job requests to the scheduler are made as follows using the command qsub provided by the scheduler.

```
$ qsub  [option]  [job script file name]
```

When you execute the qsub command, the request ID is numbered and displayed on the standard output as shown below.

Request **1182.sqd** submitted to queue: SQUID.

The main options of the qsub command are: There are options common to the system, options for vector nodes, options for CPU nodes, and options for GPU nodes. Please check the options according to the system to be used.

【common options】

Options	Description
-q [queue name]	Specify the queue to submit batch jobs. **(Required)**
--group=[group name]	Executes the job in the specified group. The point of the specified group is consumed. **(Required)**
-l elapstim_req=[elapsed time limit]	Specify the elapsed time limit for batch jobs. If not specified, the default value for each queue will be applied.Its format is **hh:mm:ss** **hh** hours **mm** minutes **ss** seconds
-N [job name]	Specify the name of the job. If not specified, the batch script name will be the job name.
-o [stdout file name]	Specify the output file name of the standard output of the batch job. If not specified, the file name of "[job name].o[request ID]" is output to the directory where the job is submitted.
-e [stderr file name]	Specify the output file name of the standard error of the batch job. If not specified, the file name of "[job name].e[request ID]" will be output to the directory where the request is submitted.
-j [o,e]	Merge the standard output and standard error output of a batch job. o: Outputs the merged result to standard output. e: Outputs the merged result to standard error. (A space is required between -j and o or e)
-M [mail address]	Specify the destination of the email. When specifying more than one -M email_address1,email_address2 Please separate with "," like.
-m [b,e,a]	We will send you an email about changes in the

	status of batch jobs. b: Send an email when the job starts e: Send an email when the job is finished a: Send an email when the job ends abnormally (A space is required between -m and b or e) You can specify more than one. Example) Email notification at start and end -M be
-v [environmet valiables]	Specifies the environment variables to use when running the batch job.
-r { y | n }	Specifies whether to rerun the batch job. Y: Rerun is possible N: No rerun

【options for CPU nodes】

Options	Description
-b [number of nodes]	Specifies the number of nodes to run the job.
-T [mpi library]	Must be specified when executing MPI. If you are using the Intel compiler, Intel MPI is available. Specify -T intmpi. If you are using the NVIDIA HPC SDK compiler, OpenMPI is available. Specify -T openmpi.

【options for GPU nodes】

Options	Description
-l gpunum_job=[GPUs per node]	Specify the number of GPUs to use per node.
--gpuum-lhost=[GPUs per node]	Same as "-l gpunum_job" above.
-T [mpi library]	Must be specified when executing MPI. If you are using the Intel compiler, Intel MPI is available. Specify -T intmpi. If you are using the NVIDIA HPC SDK compiler, OpenMPI is available. Specify -T openmpi.

【options for Vector nodes】

Options	Description
--venode=[VEs]	Specify the total number of VEs to use.
--venum-lhost=[VEs per logical host]	Specify the number of VEs that make up the logical host (roughly understood as [Number of VEs to reserve in 1VH]).
-T necmpi	Must be specified when executing MPI with SX-Aurora TSUBASA.

5.1.4. Job Management System Commands

The status of the submitted job can be checked with various commands of the job management system.

(1) Check Batch Job

Use the qstat command to check the status of submitted jobs.

- When displaying a list of jobs submitted by the user

```
$ qstat
```

- When displaying the list without omission

```
$ qstat   -l
```

The above are lowercase letters L.

Width optimization is performed by specifying "--adjust-column" at the same time.

- When displaying the details of a specific job

```
$ qstat   -f   [Request ID]
```

Use the sstat command to check the execution start time of the submitted request. If the execution start time has not been decided, the time will not be displayed.

```
$ sstat
```

Use the qcat command to display the contents of standard output / standard error output during job execution. If you do not specify -e / -o, the job script will be displayed.

```
$ qcat  -e  [Request ID]    ※When displaying standard error output
$ qcat  -o  [Request ID]    ※For standard output display
```

It is also possible to combine the following options.

-f	Outputs the added data as the contents of the file continue to grow.
-n	Displays the specified number of lines. (If not specified, it is for 10 lines.)
-b	Display from the beginning of the file. (If not specified, it will be displayed from the last line.)

When the job is finished, it disappears with the qstat command.

(2) Hold Batch Job (Hold)

To hold the submitted job, use the qhold command. By holding it, it will be excluded from the scheduling target and execution will not start.

```
$ qhold  [Request ID]
```

(3) Release Hold Of Batch Job (Release)

Use the qrls command to release the hold. By releasing the hold, the job returns to the state before the hold was released and is subject to scheduling again.

```
$ qrls  [Request ID]
```

(4) Display job information

Use the acstat command to display job information submitted by the user in the past.

- When displaying job information submitted by the user in the past (within the past 24 hours from the time the command was executed)

```
$ acstat
```

- When displaying information (command execution year) of jobs submitted by the user in the past

```
$ acstat -A
```

Also, use the acstatgroup command to display information on jobs submitted by the group to which the user belongs in the past.

- When displaying job information submitted in the past by the group to which the user belongs (within the past 24 hours from the time the command was executed)

```
$ acstatgroup
```

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

```
$ acstatgroup -A
```

(5) Delete Batch Job

Use the qdel command to delete the submitted request.

```
$ qdel [Request ID]
```

If the batch job is in the run state (RUN), SIGTERM is sent first, then SIGKILL. You can specify the wait time before sending a SIGKILL by specifying the number of seconds of grace time with the -g option. default is 5 seconds.

(6) Display Queue Status

Use the qstat command to display the status of the queue on which batch jobs run.

```
$ qstat -Q
```

5.2. Job Class

This section describes the SQUID job class. Each job class corresponds to a queue on the job management system, and the user can use the calculation environment by submitting a job to the queue.

The queue is divided into an input queue in which the user directly submits a job and an execution queue that waits for the execution order.

5.2.1. Job Class for General-Purpose CPU Envirnment

How to Use	Job class	Available elapsed time	Maximum number of cores	Available memory	Number of nodes	Remarks
Shared use	SQUID	24hours	38,912core (76c/node)	124TiB (248GB/node)	512	Occupied use inside the node
	SQUID-R	24hours	38,912core (76c/node)	124TiB (248GB/node)	512	＊1

	SQUID-H	24hours	38,912core (76c/node)	124TiB (248GB/node)	512	* 2
	SQUID-S	24hours	38core	124GiB	1	* 3
	DBG	10 minutes	152core (76c/node)	496GiB (248GB/node)	2	For debugging
	INTC	10 minutes	152core (76c/node)	496GiB (248GB/node)	2	Interactive use
Occupied use	mySQUID	Unlimited	76core × Number of occupied nodes	248GiB × Number of occupied nodes	Occupied number	

*1 For those who want to reduce the execution waiting time by allowing the use of routes with a narrow bandwidth of the interconnection network.

*2 For those who want to increase the point consumption, submit high priority jobs, and shorten the execution waiting time.

*3 For those who want to reduce point consumption by allowing sharing within the node with other jobs

5.2.2. Job Class For Vector Computing Environment

How to Use	Job class	Available elapsed time	Maximum number of cores	Available memory	Number of nodes	Remarks
Shared use	SQUID	24hours	2,560core (10c/VE)	12TiB (48GB/VE)	256	
	SQUID-H	24hours	2,560core (10c/VE)	12TiB (48GB/VE)	256	*1
	SQUID-S	24hours	40core	192GiB	4	*2
	DBG	10 minutes	20core (10c/VE)	96GiB (48GB/VE)	4	For debugging
	INTV	10 minutes	20core (10c/VE)	96GiB (48GB/VE)	4	Interactive use
Occupied use	mySQUID	Unlimited	10core × Occupied VE number	48GiB × Occupied VE number	Occupied number	

*1 For those who want to increase the point consumption, submit high priority jobs, and shorten the execution waiting time.

*2 For those who want to reduce point consumption by allowing sharing within the node with other jobs

5.2.3. Job Class for GPGPU Computing Environment

How to Use	Job class	Available elapsed time	Maximum number of cores	Available memory	Number of nodes	Remarks
Shared use	SQUID	24hours	2,432core (76c/node)	15.75TiB (504GB/node)	32	
	SQUID-H	24hours	2,432core (76c/node)	15.75TiB (504GB/node)	32	*1
	SQUID-S	24hours	38core	252GiB	1	*2
	DBG	10 minutes	152core (76c/node)	1,008GiB (504GB/VE)	2	For debugging
	INTG	10 minutes	152core (76c/node)	1,008GiB (504GB/VE)	2	Interactive use
Occupied use	mySQUID	Unlimited	76core × Number of occupied nodes	504GiB × Number of occupied nodes	Occupied number	

*1 For those who want to increase the point consumption, submit high priority jobs, and shorten the execution waiting time.

*2 For those who want to reduce point consumption by allowing sharing within the node with other jobs

5.3. General-Purpose CPU Enveronment

This section describes job scripts for general-purpose CPU environment.

5.3.1. How to Use Serial Execution

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

● queue	: SQUID
● group	: G01234
● elapse time	: 30 minuites（1 hour if not specified）

```
1   #!/bin/bash
2   #------- qsub option -----------
3   #PBS -q SQUID
4   #PBS --group=G01234
5   #PBS -l elapstim_req=00:30:00
6
7   #------- Program execution -----------
8   module load BaseCPU/2021
9   module load xxx/xxx
10
11  cd $PBS_O_WORKDIR
12  ./a.out
```

- line 9

  Describes what was module loaded at compile time

This job script only executes the program, but by specifying additional options, it is also possible to make a request such as "Send a notification to the email address at the end of job execution" and "Specify the name of the output result file".

5.3.2. How to Use Thread Parallel Execution

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

● queue	: SQUID
● group	: G01234
● elapse time	: 30 minuites（1 hour if not specified）
● Total number of cores	: 76（=38 core x 2 CPU x 1 node）

```
1   #!/bin/bash
2   #------- qsub option -----------
3   #PBS -q SQUID
```

```
4    #PBS --group=G01234
5    #PBS -l elapstim_req=00:30:00
6    #PBS -v OMP_NUM_THREADS=76
7
8    #------- Program execution -----------
9
10   module load BaseCPU/2021
11   module load xxx/xxx
12
13   cd $PBS_O_WORKDIR
14   ./a.out
```

- line 6

  Specify the number of parallel executions.

- line 11

  Describes what was module loaded at compile time.

Precautions when using

Be sure to specify "OMP_NUM_THREADS" in the execution script.

If "OMP_NUM_THREADS" is not specified, or if an incorrect value is specified, execution may occur with an unintended parallel number. Please be careful.

5.3.3. How to Use MPI Execution

This is a sample job script that assumes 304 parallel execution on 4 nodes.

● queue	: SQUID
● group	: G01234
● elapse time	: 30 minuites（1 hour if not specified)
● Total number of cores	: 304（=38 core ×2 CPU ×4 nodes ）
● number of nodes	: 4 (No need to specify when executing within one node)
● MPI library	: IntelMPI

```
#!/bin/bash
#------- qsub option -----------
#PBS -q SQUID
#PBS --group=G01234
#PBS -l elapstim_req=00:30:00
#PBS -b 4
#PBS -T intmpi

#------- Program execution -----------
```

```
10
11   module load BaseCPU/2021
12   module load xxx/xxx
13
14   cd $PBS_O_WORKDIR
15   mpirun ${NQSV_MPIOPTS} -np 304 ./a.out
```

- line 6

  Specify the number of nodes.

- line 7

  Specify -T intmpi when using MPI

- line 12

  Describes what was module loaded at compile time.

- line 15

  Specify $ {NQSV_MPIOPTS} as an argument of mpirun. Through this specification, the host on which the job will be executed is passed to MPI.

  The path of the current directory where the qsub command is executed is automatically assigned to the environment variable PBS_O_WORKDIR.

5.3.4. How to Use MPI + Thread Parallel Execution

This is a sample job script that assumes 4 processes (1 process per node) with 4 nodes and each process generates 76 threads.

● queue	: SQUID
● group	: G01234
● elapse time	: 30 minuites（1 hour if not specified)
● Total number of cores	: 304（=38 core ×2 CPU ×4 nodes ）
● number of nodes	: 4 (No need to specify when executing within one node)
● MPI library	: IntelMPI
● Number of threads	: 76（Specified by OMP_NUM_THREADS variable)

```
1    #!/bin/bash
2    #------- qsub option -----------
3    #PBS -q SQUID
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=76
9
10   #------- Program execution -----------
11
12   module load BaseCPU/2021
13   module load xxx/xxx
14
15   cd $PBS_O_WORKDIR
16   mpirun ${NQSV_MPIOPTS} -np 4 ./a.out
```

- line 7

  Specify -T intmpi when using MPI.

- line 13

  Describes what was module loaded at compile time.

- line 16

  Specify $ {NQSV_MPIOPTS} as an argument of mpirun. Through this specification, the host on which the job will be executed is passed to MPI.
  The path of the current directory where the qsub command is executed is automatically assigned to the environment variable PBS_O_WORKDIR.

5.3.5. Advanced usage

➢ **Manual specification of the number of processes in a node in Intel MPIs**

you can limit the processes within a node by specifying the -ppn, -rr, and -perhost options (including the I_MPI_PERHOST environment variable) in Intel MPI. However, when used with the -machinefile option, the -machinefile option takes precedence.

In SQUID's job management system, the -ppn, -rr, and -perhost options are invalid because the NQSV_MPIOPTS environment variable contains the -machinefile option. If you want to limit the number of processes in the node, you can specify the required number of cores in the -l cpunum_job option. However, if you want to make more detailed specifications such as using core pinning etc merge, use the PBS_NODEFILE environment variable.

The following is a sample job script that uses the PBS_NODEFILE environment variable to generate 128 processes (64 processes per node) on 2 nodes.

● queue	: SQUID
● group	: G01234
● elapse time	: 30 minuites（1 hour if not specified)
● Total number of cores	: 128（=64 process x 2 node)
● number of nodes	: 2（No need to specify when executing within one node)
● MPI library	: IntelMPI

```
1   #!/bin/bash
2   #------- qsub option -----------
3   #PBS -q SQUID
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/2021
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

  Specify -T intmpi when using MPI.

- line 12

  Describes what was module loaded at compile time.

- line 15

  Specify $ {NQSV_MPIOPTS} as an argument of mpirun. Through this specification, the host on which the job will be executed is passed to MPI.

  In addition, by specifying -ppn 64, it is possible to specify 64 processes per node.

5.4. Vector Computing Environment

This section describes the job script for vector nodes.

5.4.1. How to Use Serial Execution

This is a sample job script that assumes program execution within 1VE.

● queue	: SQUID-S
● group	: G01234
● elapse time	: 30 minuites（1 hour if not specified)
● Total number of VE	: 1

```
1   #!/bin/bash
2   #------- qsub option -----------
3   #PBS -q SQUID-S
4   #PBS --group=G01234
5   #PBS -l elapstim_req=00:30:00
6   #PBS --venode=1
7
8   #------- Program execution -----------
9   module load BaseVEC/2021
10  module load xxx/xxx
11
12  cd $PBS_O_WORKDIR
13  ./a.out
```

- line 6

  Specify the number of nodes.

- line 10

  Describes what was module loaded at compile time

5.4.2. How to Use Thread Parallel Execution

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

● queue	: SQUID-S
● group	: G01234
● elapse time	: 30 minuites（1 hour if not specified)
● Number of threads	: 10 (Specified by OMP_NUM_THREADS variable)

```
#!/bin/bash
#------- qsub option -----------
#PBS -q SQUID-S
#PBS --group=G01234
#PBS -l elapstim_req=00:30:00
#PBS -v OMP_NUM_THREADS=10
#PBS --venode=1

```

```
9    #------- Program execution -----------
10
11   module load BaseVEC/2021
12   module load xxx/xxx
13
14   cd $PBS_O_WORKDIR
     ./a.out
```

- line 6

  Specify the number of parallel executions.

- line 12

  Describes what was module loaded at compile time.

Precautions:

Be sure to specify "OMP_NUM_THREADS" in the execution script.

If "OMP_NUM_THREADS" is not specified, or if an incorrect value is specified, execution may occur with an unintended parallel number.

5.4.3. How to Use MPI Execution

This is a sample job script that assumes 400 parallel execution with 40VE.

● queue	: SQUID
● group	: G01234
● elapse time	: 30 minuites（1 hour if not specified）
● Total number of VE	: 40
● MPI library	: NEC MPI
● Number of parallels	: 400(40VE×10 core)

```
1    #!/bin/bash
2    #------- qsub option -----------
3    #PBS -q SQUID
4    #PBS --group=G01234
5    #PBS -l elapstim_req=00:30:00
6    #PBS --venode=40
7    #PBS -T necmpi
8
9    #------- Program execution -----------
10
11   module load BaseVEC/2021
12   module load xxx/xxx
13
14   cd $PBS_O_WORKDIR
```

```
15   mpirun -venode -np 400 ./a.out
```

- line 7

  -T necmpi must be specified when using MPI. Also, only necmpi can be used.

- line 12

  Describes what was module loaded at compile time.

- line15

  NQSV automatically assigns MPI processes to VH and VE.
  If the number of processes is not a multiple of 8 when using 9VE or more, the number of processes assigned to VE may be uneven. If you want to assign all MPI processes to VE, we recommend that you specify the -venode option.

5.4.4. How to Use MPI + Thread Parallel Execution

This is a sample job script that assumes 40 processes (1 process per VE) with 40VE and each process generates 8 threads.

● queue	: SQUID
● group	: G01234
● elapse time	: 30 minuites（1 hour if not specified）
● Total number of VE	: 40
● Number of VEs per logical host	: 8（Up to 8 per host）
● MPI library	: NEC MPI
● Number of threads	: 10 (Specified by OMP_NUM_THREADS variable)

```
#!/bin/bash
#------- qsub option -----------
#PBS -q SQUID
#PBS --group=G01234
#PBS -l elapstim_req=00:30:00
#PBS --venode=40
#PBS -T necmpi
#PBS -v OMP_NUM_THREADS=10

#------- Program execution -----------

module load BaseVEC/2021
module load xxx/xxx

```

```
15    cd $PBS_O_WORKDIR
16    mpirun -venode -np 40 ./a.out
```

- line 7

  T necmpi must be specified when using MPI. Also, only necmpi can be used.

- line 8

  Specify the number of threads. The number of processes per VE x the number of threads should not exceed 10, which is the number of cores.

- line 13

  Describes what was module loaded at compile time.

- line 16

  NQSV automatically assigns MPI processes to VH and VE.
  If the number of processes is not a multiple of 8 when using 9VE or more, the number of processes assigned to VE may be uneven. If you want to assign all MPI processes to VE, we recommend that you specify the -venode option.

5.5. GPGPU Computing Environment

This section describes the job script for GPU nodes.

5.5.1. How to Use Serial Execution

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

● queue	: SQUID
● group	: G01234
● elapse time	: 30 minuites（1 hour if not specified）
● Number of GPU	: 1

```
#!/bin/bash
#------- qsub option -----------
#PBS -q SQUID
#PBS --group=G01234
#PBS -l elapstim_req=00:30:00
#PBS -l gpunum_job=1

#------- Program execution -----------
module load BaseGPU/2021
```

```
10   module load xxx/xxx
11
12   cd $PBS_O_WORKDIR
13   ./a.out
```

- line 6

  Specifies to use 8 GPUs per node.

- line 10

  Describes what was module loaded at compile time.

5.5.2. How to Use Thread Parallelization

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

● queue	: SQUID
● group	: G01234
● elapse time	: 30 minuites （1 hour if not specified）
● Number of threads	: 76

```
1    #!/bin/bash
2    #------- qsub option -----------
3    #PBS -q SQUID
4    #PBS --group=G01234
5    #PBS -l elapstim_req=00:30:00
6    #PBS -l gpunum_job=8
7    #PBS -v OMP_NUM_THREADS=76
8
9    #------- Program execution -----------
10
11   module load BaseGPU/2021
12   module load xxx/xxx
13
14   cd $PBS_O_WORKDIR
     ./a.out
```

- line 7

  Specify the number of parallel executions.

- line 11

  Describes what was module loaded at compile time.

Precautions:

Be sure to specify "OMP_NUM_THREADS" in the execution script.

If "OMP_NUM_THREADS" is not specified, or if an incorrect value is specified, execution may occur with an unintended parallel number.

5.5.3. How to Use MPI Execution

This is a sample job script that assumes OpenMPI program execution that creates a total of 4 processes on 2 nodes.

● queue	: SQUID
● group	: G01234
● elapse time	: 30 minuites（1 hour if not specified)
● Number of nodes	: 2
● MPI library	: OenMPI

```
1   #!/bin/bash
2   #------- qsub option -----------
3   #PBS -q SQUID
4   #PBS --group=G01234
5   #PBS -l elapstim_req=00:30:00
6   #PBS -b 2
7   #PBS -l gpunum_job=8
8   #PBS -T openmpi
9   #PBS -v NQSV_MPI_MODULE=BaseGPU/2021
10
11  #------- Program execution -----------
12
13  module load BaseGPU/2021
14  module load xxx/xxx
15
16  mpirun ${NQSV_MPIOPTS} -np 4 -npernode 2  ${PBS_O_WORKDIR}/mpi_prog
```

- line 9

  Specify the module name required when starting OpenMPI. When specifying multiple modules, it is possible to specify multiple modules separated by a colon (:).

  ex.) specify BaseGPU/2021 and cuda/11.3 modules

  ```
  -v NQSV_MPI_MODULE=BaseGPU/2021:cuda/11.3
  ```

- line 13

  Specify the MPI environment settings with the module command. This is the setting on the master node. The settings for the slave node are made with the qsub option (lines 8 and 9 of the script).

- line 16

Specify $ {NQSV_MPIOPTS} as an argument of mpirun. Through this specification, the host on which the job will be executed is passed to MPI.

The path of the current directory where the qsub command was executed is automatically assigned to the environment variable PBS_O_WORKDIR.

5.5.4. How to Use MPI + Thread Parallel Execution

This is a job script sample that assumes the execution of an OpenMPI program that creates two processes on each node with two nodes and creates 12 threads in each process by OpenMP.

● queue	: SQUID
● group	: G01234
● elapse time	: 30 minuites（1 hour if not specified）
● Number of nodes	: 2
● MPI library	: OenMPI
● Number of threds	: 12（Specified by OMP_NUM_THREADS variable）

```
1   #!/bin/bash
2   #------- qsub option -----------
3   #PBS -q SQUID
4   #PBS --group=G01234
5   #PBS -l elapstim_req=00:30:00
6   #PBS -b 2
7   #PBS -l gpunum_job=8
8   #PBS -T openmpi
9   #PBS -v NQSV_MPI_MODULE=BaseGPU/2021
10  #PBS -v OMP_NUM_THREADS=12
11
12  #------- Program execution -----------
13
14  module load BaseGPU/2021
15  module load xxx/xxx
16
17  mpirun ${NQSV_MPIOPTS} -np 4 -npernode 2 --bind-to socket ¥
                                      --report-bindings  ${PBS_O_WORKDIR}/mpi_prog
```

- line 9

  Specify the module name required when starting OpenMPI. When specifying multiple modules, it is possible to specify multiple modules separated by a colon (:).

  ex.) specify BaseGPU/2021 and cuda/11.3 modules

```
-v NQSV_MPI_MODULE=BaseGPU/2021:cuda/11.3
```

- line 14
  Specify the MPI environment settings with the module command. This is the setting on the master node. The settings for the slave node are made with the qsub option (lines 8 and 9 of the script).

- line 17
  Specify $ {NQSV_MPIOPTS} as an argument of mpirun. Through this specification, the host on which the job will be executed is passed to the MPI.
  The path of the current directory where the qsub command was executed is automatically assigned to the environment variable PBS_O_WORKDIR.
  In OpenMPI, to allocate 1 process to 1 CPU and 12 threads to 12 cores, set "--bind-to socket" etc. as in the above sample. You can display the status of core allocation to a process by specifying "--report-bindings".

5.5.5. Advanced usage

➢ **Manual specification of GPU used by MPI process**

SQUID's job management system and OpenMPI do not control the GPU used by each MPI process. Therefore, depending on the program, the GPU used may collide between MPI processes (rank).

If you want to control the GPU used for each MPI process, you can handle it by preparing a wrapper script for executing the program and specifying the GPU number to be used for each MPI process (rank).

This is a job script sample that creates 8 processes on one node and assumes OpenMPI program execution. As a wrapper script, we will explain two types, one is to centrally distribute to the CPU (mpiwrap-seq.sh) and the other is to distribute to the CPU (mpiwrap-alt.sh).

● queue	: SQUID
● group	: G01234
● elapse time	: 30 minuites（1 hour if not specified）
● Number of nodes	: 1
● MPI library	: OenMPI

```
1    #!/bin/bash
2    #------- qsub option -----------
3    #PBS -q SQUID
4    #PBS --group=G01234
5    #PBS -l elapstim_req=00:30:00
6    #PBS -b 1
7    #PBS -l gpunum_job=8
8    #PBS -T openmpi
9    #PBS -v NQSV_MPI_MODULE=BaseGPU/2021
10
11   #------- Program execution -----------
12
13   module load BaseGPU/2021
14   WRAP=${PBS_O_WORKDIR}/mpiwrap-seq.sh
15
16   mpirun ${NQSV_MPIOPTS} -np 8 -npernode 8 ¥
17    --bind-to socket ${WRAP} ${PBS_O_WORKDIR}/mpi_prog
18
```

- line 14

  Specifies the path of the wrapper script to use for WRAP.

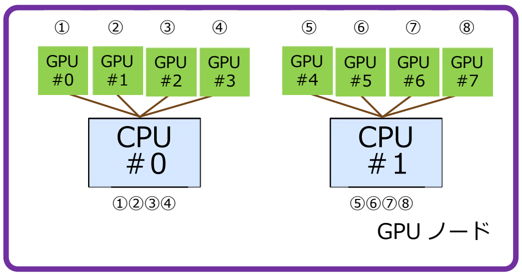
- line 17

  Specify $ {WRAP} as an argument of mpirun.

  You can enable the wrapper script.

※ Rapper script example to be centrally placed on the CPU.

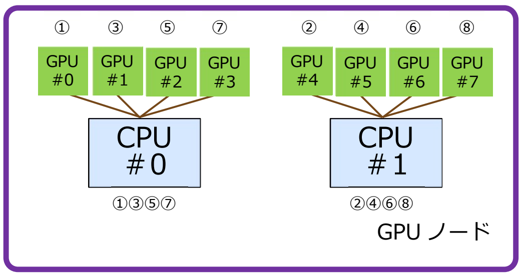
mpiwrap-seq.sh : CPU and GPU are assigned in the order shown in the figure, starting from rank 0.



```
#!/bin/bash

echo LANK=${OMPI_COMM_WORLD_LOCAL_RANK}

case ${OMPI_COMM_WORLD_LOCAL_RANK} in
    0 ) # CPU 0, GPU 0
        export CUDA_VISIBLE_DEVICES=0
        numactl -N 0 -l $@
        ;;
    1 ) # CPU 0, GPU 1
        export CUDA_VISIBLE_DEVICES=1
        numactl -N 0 -l $@
        ;;
    2 ) # CPU 0, GPU 2
        export CUDA_VISIBLE_DEVICES=2
        numactl -N 0 -l $@
        ;;
    3 ) # CPU 0, GPU 3
        export CUDA_VISIBLE_DEVICES=3
        numactl -N 0 -l $@
        ;;
    4 ) # CPU 1, GPU 4
        export CUDA_VISIBLE_DEVICES=4
        numactl -N 1 -l $@
        ;;
    5 ) # CPU 1, GPU 5
        export CUDA_VISIBLE_DEVICES=5
        numactl -N 1 -l $@
        ;;
    6 ) # CPU 1, GPU 6
        export CUDA_VISIBLE_DEVICES=6
        numactl -N 1 -l $@
        ;;
    7 ) # CPU 1, GPU 7
        export CUDA_VISIBLE_DEVICES=7
        numactl -N 1 -l $@
        ;;
esac
```

※ Rapper script example distributed to CPU

mpiwrap-alt.sh : CPU and GPU are assigned in the order shown in the figure, starting from rank 0.



```
#!/bin/bash

echo LANK=${OMPI_COMM_WORLD_LOCAL_RANK}

case ${OMPI_COMM_WORLD_LOCAL_RANK} in
    0 ) # CPU 0, GPU 0
        export CUDA_VISIBLE_DEVICES=0
        numactl -N 0 -l $@
        ;;
    1 ) # CPU 1, GPU 4
        export CUDA_VISIBLE_DEVICES=4
        numactl -N 1 -l $@
        ;;
    2 ) # CPU 0, GPU 1
        export CUDA_VISIBLE_DEVICES=1
        numactl -N 0 -l $@
        ;;
    3 ) # CPU 1, GPU 5
        export CUDA_VISIBLE_DEVICES=5
        numactl -N 1 -l $@
        ;;
    4 ) # CPU 0, GPU 2
        export CUDA_VISIBLE_DEVICES=2
        numactl -N 0 -l $@
        ;;
    5 ) # CPU 1, GPU 6
        export CUDA_VISIBLE_DEVICES=6
        numactl -N 1 -l $@
        ;;
    6 ) # CPU 0, GPU 3
        export CUDA_VISIBLE_DEVICES=3
        numactl -N 0 -l $@
        ;;
    7 ) # CPU 1, GPU 7
        export CUDA_VISIBLE_DEVICES=7
        numactl -N 1 -l $@
        ;;
esac
```

5.6. Container

This section describes the job script when executing a job using a container. For an overview of how to use the container in this system, refer to "1.5.4 Use container"

5.6.1. Overview of container execution

This section describes the minimum contents that should be understood when executing a container. The centos image file (centos.sif) is taken as an example of a container in a standard environment.

➢ Execution command

The container is executed by specifying the exec subcommand. As an execution example, when executing the hostname command in the centos.sif container image, execute the following command.

```
$ singularity exec centos.sif hostname
```

The command format is as follows.

```
$ singularity exec <image file name> <command in container>
```

Please note that the specified command is an execution command in the container. If the command does not specify a path, the command found in the PATH environment variable in the container will be executed. Even if the path is specified, the absolute path follows the file structure in the container.

➢ Environment variables

Environment variables defined outside the container are basically inherited inside the container as well. However, environment variables that are explicitly defined on the container side at the time of build etc. follow the definition on the container side.

When overwriting the environment variables defined on the container side, it is possible to pass them in the container by specifying them individually with the --env option or collectively with the --env-file option.

・single specification with --env option

```
$ singularity exec --env MYVAR="My Value!" centos.sif myprog.exe
```

・batch specification with --env-file option

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

For more information about environment variables, please refer to the official document below.

https://sylabs.io/guides/3.7/user-guide/environment_and_metadata.html

➢ Mount host OS

If you want to read / write the file system of the host OS from within the center, you can use it by bind-mounting a specific directory of the host. The following directories are mounted as standard and can be used in the container with the same path.

Home directory	: /sqfs/home/<user name>
Temporary area	: /tmp

As an example, the command to execute the program (a.out) placed in the home directory of the host OS from inside the container is as follows.

```
$ singularity exec centos.sif ./a.out
```

※ In the above example, the current directory has been moved to home inside the container.

If you want to mount a specific directory on the host OS, use the --bind option. The format of the --bind option is as follows.

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

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

As an example, the command to execute the program (a.out) placed in the directory on the extension area is as follows.

```
$ cd /sqfs/work/<group name>/<user name>
$ singularity exec --bind `pwd` centos.sif ./a.out
```

※ In the above example, the environment variable PWD outside the container is inherited inside the container, and the current directory inside the container is the directory on the extension area.

For details on mounting the host OS, refer to the official document below.

https://sylabs.io/guides/3.7/user-guide/bind_paths_and_mounts.html

5.6.2. How to run container on CPU nodes

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

● Queue	: SQUID
● Group	: G01234
● elapse time	: 30 minuites （1 hour if not specified）
● Total CPU cores	: 76 （=38core×2CPU×1node）

```
1   #!/bin/bash
2   #------- qsub option -----------
3   #PBS -q SQUID
4   #PBS --group=G01234
5   #PBS -l elapstim_req=00:30:00
6   #PBS -v OMP_NUM_THREADS=76
7
8   #------- Program execution -----------
9
10  cd $PBS_O_WORKDIR
11  singularity exec --bind `pwd` image.sif ./a.out
12
```

- line 6

  Specify the number of parallel executions.

- line 11

  Specify the image file and execute the singularity exec command.

Precautions when using

Be sure to specify "OMP_NUM_THREADS" in the execution script.

If "OMP_NUM_THREADS" is not specified, or if an incorrect value is specified, execution may occur with an unintended parallel number. Please be careful.

5.6.3. How to run container on GPU nodes

This is a sample job script that assumes thread parallel program execution within one node in a GPGPU calculation environment.

● Queue	: SQUID
● Group	: G01234

- elapse time : 30 minuites（1 hour if not specified）
- Number of threds : 76（Specified by OMP_NUM_THREADS variable)

```
1   #!/bin/bash
2   #------- qsub option -----------
3   #PBS -q SQUID
4   #PBS --group=G01234
5   #PBS -l elapstim_req=00:30:00
6   #PBS -l gpunum_job=8
7   #PBS -v OMP_NUM_THREADS=76
8
9   #------- Program execution -----------
10
11  cd $PBS_O_WORKDIR
12  singularity exec --nv --bind `pwd` image.sif ./a.out
13
```

- line 7

  Specify the number of parallel executions.

- line 12

  Specify the image file and execute the singularity exec command. Add the --nv option to make GPGPU available from inside the container.

Precautions when using

Be sure to specify "OMP_NUM_THREADS" in the execution script.

If "OMP_NUM_THREADS" is not specified, or if an incorrect value is specified, execution may occur with an unintended parallel number. Please be careful.

6. Applications

This system has the following applications

6.1. Application List

- ISV Applications

ISV application	available host				
	squidhpc	squidhpda	cpu	vec	gpu
AVS/Express Developer	○	○	-	-	-
Gaussian	○	○	○	-	-
IDL	○	○	-	-	-

- OSS Applications

OSS application	available host				
	squidhpc	squidhpda	cpu	vec	gpu
ADIOS	○	○	○	-	-
GAMESS	○	○	○	-	-
Gnuplot	○	○	-	-	-
GROMACS	○	○	○	-	○
ImageMagick	○	○	-	-	-
LAMMPS	○	○	○	-	-
NcView	○	○	○	-	-
Octave	○	○	○	-	-
OpenFORM	○	○	○	-	-
ParaView	○	○	-	-	-
Quantum ESPRESSO	-	-	-	○	-
PHASE/0	-	-	-	○	-
Relion	○	○	○	-	-
VisIt	○	○	○	-	-

- National Project Applications

National Project application	available host				
	squidhpc	squidhpda	cpu	vec	gpu

ABINIT-MP	○	○	○	○	-
HΦ	○	○	○	-	-
MODYLAS	○	○	○	-	-
NTChem	○	○	○	-	-
OpenMX	○	○	○	-	-
SALMON	○	○	○	-	-
SMASH	○	○	○	-	-

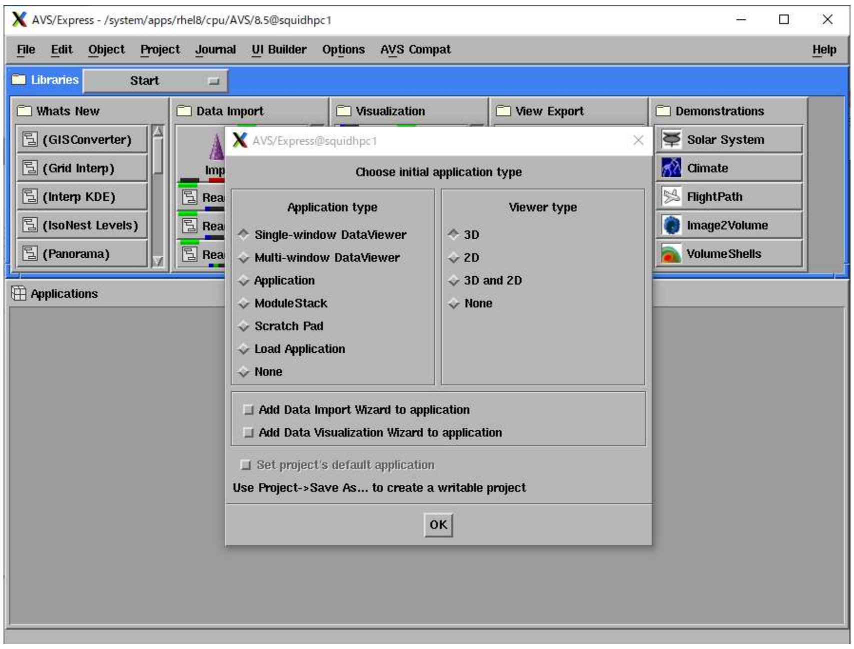
6.2. How to Use ISV Applications

6.2.1. AVS/Express

AVS / Express is available in version 8.5. The number of simultaneous users in SQUID is five. After logging in to the front end using X terminal software, execute the following command to start AVS / Express.

```
$ module load BaseApp/2021
$ module load AVSExpress/8.5
$ express
```

The AVS / Express operation screen is displayed as shown below.



In addition, AVS / Express provides a license server with SQUID. To install and use AVS / Express on your device, specify the following license server and port number.

No	License server	Port Number
1	squidportal.hpc.cmc.osaka-u.ac.jp	27935/tcp

The number of floating licenses is 5. If the license is insufficient, the following message will be displayed on the terminal at startup.

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

6.2.2. Gaussian

Gaussian, a quantum chemistry calculation program, is installed version g16. To use it, you need to contact the center to apply for registration with the Gaussian application group.

Execute the program as a batch job. The following is an example of executing the sample program.

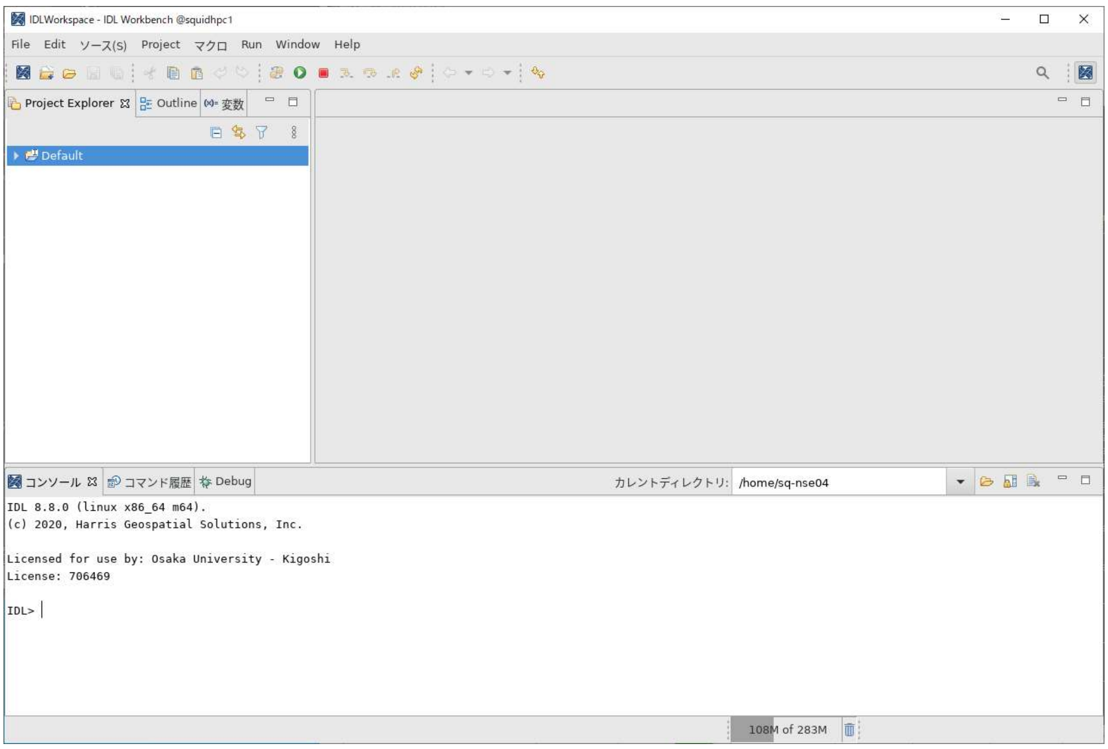
```
#!/bin/bash

#PBS -q SQUID
#PBS --group="group name"
#PBS -l elapstim_req=01:00:00

module load BaseApp
module load Gaussian

newgrp gaussain

cd $PBS_O_WORKDIR
g16root/g16/g16 < input >& result
```

6.2.3. IDL

IDL offers version 8.8. SQUID refers to the OCTOPUS license server, and the total number of simultaneous users is 4 including OCTOPUS. After logging in to the front end using X terminal software etc., execute the following command to start IDL.

```
$ module load BaseApp/2021
$ module load IDL/8.8
$ idlde
```

The IDL operation screen is displayed as shown below.


IDLWorkspace - IDL Workbench @squidhpc1
File Edit ソース(S) Project マクロ Run Window Help
Project Explorer
Outline
変数
Default
コンソール
コマンド履歴
Debug
カレントディレクトリ: /home/sq-nse04
108M of 283M

```
IDL 8.8.0 (linux x86_64 m64).
(c) 2020, Harris Geospatial Solutions, Inc.

Licensed for use by: Osaka University - Kigoshi
License: 706469

IDL>
```

The number of floating licenses is 4. If the license is insufficient, a dialog box with the following message will be displayed.

```
Unable to license IDL.
```

6.3. How to Use OSS Applications

6.3.1. ADIOS

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

```
$ module load BaseApp/2021
$ module load adios/1.13.1
```

※We are preparing how to use it. Please wait.

6.3.2. GAMESS

Execute the program as a batch job. The following is an example of executing the sample program.If nothing is specified, / sqfs / work / (group name) / (user number) / scr will be generated as the scratch directory.

```
#!/bin/bash
#PBS -q SQUID
#PBS -l cpunum_job=76
```

```
#PBS --group=<group name>
#PBS -l elapstim_req=01:00:00
#PBS -b 2
#PBS -T intmpi

module load BaseApp/2021
module load GAMESS/v2020.2

cd $PBS_O_WORKDIR
rungms exam01.inp 00 24
```

When specifying the scratch directory, define the following environment variables before executing the program.

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

6.3.3. Gnuplot

After logging in to the front end, do the following.

```
$ gnuplot
```

6.3.4. GROMACS

GROMACS has different module names and binary names depending on the difference between single precision / double precision, the difference between single version / MPI version, and the presence or absence of GPU support. The module name and binary name of each are as follows.

種別	cpu (non GPU accelerated)		gpu (GPU accelerated)	
	Module name	Binary name	Module name	Binary name
Single precision Single execution	gromacs/2021.2	gmx	gromacs/2021.2.GPU	gmx
Single precision MPI execution	gromacs/2021.2	gmx_mpi	gromacs/2021.2mpi.GPU	gmx_mpi
Double precision Single execution	gromacs/2021.2	gmx_d	–	–
Double precision MPI execution	gromacs/2021.2	gmx_mpi_d	–	–

Below are some execution examples of the sample program. Execute the program as a batch job.

<CPU node:Single precision / Single execution>

```
#!/bin/bash
#PBS -q SQUID
#PBS -l cpunum_job=76
#PBS --group=<grpop　name>
#PBS -l elapstim_req=01:00:00
#PBS -b 1
#PBS -j o

module load BaseApp/2021
module load gromacs/2021.2
export OMP_NUM_THREADS=64

cd $PBS_O_WORKDIR
gmx grompp -f MD1.mdp -c ./replace_po_mg.gro -p ./topol.top -o md1.tpr
gmx mdrun -deffnm md1 -ntomp 64
```

<CPU node: Double precision/MPI execution>

```
#!/bin/bash
#PBS -q SQUID
#PBS -l cpunum_job=76
#PBS --group=<group name>
#PBS -l elapstim_req=01:00:00
#PBS -b 2
#PBS -T intmpi
#PBS -v OMP_NUM_THREADS=1

module load BaseApp/2021
module load gromacs/2021.2

cd $PBS_O_WORKDIR
mpirun ${NQSV_MPIOPTS} -np 152 -ppn 76 gmx_mpi_d mdrun -ntomp 1 -s ch36.tpr
```

<GPU node: Single precision / MPI execution>

```
#!/bin/bash
#PBS -q SQUID
#PBS --group=<group name>
#PBS -l elapstim_req=01:00:00
#PBS -b 2
#PBS -T openmpi
#PBS -l gpunum_job=8
#PBS -v NQSV_MPI_MODULE=BaseGCC/2021:cuda/11.2
#PBS -v OMP_NUM_THREADS=6

module load BaseApp/2021
module load gromacs/2021.2mpi.GPU

cd $PBS_O_WORKDIR
export GMX_MPI=`which gmx_mpi`

mpirun ${NQSV_MPIOPTS} -np 2 -npernode 1 ${GMX_MPI} mdrun -s poly-ch2.tpr
```

6.3.5. ImageMagick

ImageMagick is software for manipulating and displaying images.

The following is an example of how to change the extension of an image (JPG → PNG).

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

6.3.6. LAMMPS

LAMMPS is an open source molecular dynamics application.The introduced packages are as follows.

The installation package is as follows.

```
ASPHERE,BODY,CLASS2,COLLOID,COMPRESS,CORESHELL,DIPOLE,GRANULAR,KSPACE,
MANYBODY,MC,MISC,MOLECULE,MPIIO,OPT,PERI,POEMS,PYTHON,QEQ,REPLICA,RIGID,
SHOCK,SNAP,SPIN,SRD,VORONOI,USER-ATC,USER-AWPMD,SER-BOCS,USER-CGDNA,
USER-CGSDK,USER-COLVARS,USER-DIFFRACTION,USER-DPD,USER-DRUDE,USER-EFF,
USER-FEP,USER-INTEL,USER-LB,USER-MANIFOLD,USER-MEAMC,USER-MESODPD,
USER-MGPT,USER-MISC,USER-MOFFF,USER-NETCDF,USER-OMP,USER-PHONON,USER-QTB,
USER-REAXC,USER-SMTBQ,USER-SPH,USER-TALLY,USER-UEF
```

Please execute it as a batch job. An example of a job script is as follows.

```
#!/bin/bash
#PBS -q SQUID
#PBS -l cpunum_job=76
#PBS --group="group name"
#PBS -l elapstim_req=01:00:00
#PBS -b 1
#PBS -v OMP_NUM_THREADS=38

module load BaseApp
module load lammps/29-Oct-20
cd $PBS_O_WORKDIR
mpirun ${NQSV_MPIOPTS} -np 2 lmp < ./in.lj
```

6.3.7. Octave

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

```
$ module load BaseApp
$ module load octave/5.2.0
$ octave
```

6.3.8. OpenFOAM

OpenFOAM is a fluid / continuum simulation platform.

Please execute it as a batch job. An example of a job script is as follows.

＜Sirial＞ ※For example, using the pitzDaily that comes with the software tutorial.

```
#!/bin/bash
#PBS -q SQUID
#PBS -l cpunum_job=76
#PBS --group="group name"
#PBS -l elapstim_req=01:00:00
#PBS -b 1

module load BaseApp
module load OpenFOAM/v2012
```

```

blockMesh
simpleFoam
```

<MPI>

```
#!/bin/bash
#PBS -q SQUID
#PBS -l cpunum_job=76
#PBS --group="group name"
#PBS -l elapstim_req=01:00:00
#PBS -b 2

module load BaseApp
module load OpenFOAM/v2012

cd $PBS_O_WORKDIR
blockMesh
cp 0.orig 0 -r
decomposePar
mpirun ${NQSV_MPIOPTS} -np 152 interFoam -parallel
```

6.3.9. Quantum ESPRESSO

Quantum ESPRESSO is an OSS for electronic structure calculation and material modeling based on first-principles calculation. As of August 2021, only the vector node version is available.

The only module provided in the vector node version is the plane wave basis SCF calculation package (PWscf). An example of a job script is as follows.

```
#!/bin/bash
#-------------- qsub option qsub option -------------------------
#PBS -q DBG
#PBS --group=sq-test
#PBS -l elapstim_req=00:10:00
#PBS --venode=2
#PBS -T necmpi
#-------------- Program execution Program execution ---------
```

```
module load BaseApp/2021
module load QuantumESPRESSO/6.4.1.VEC

export LM=`which pw.x`
cd $PBS_O_WORKDIR
mpirun -venode -np 16 -version /opt/nec/ve/bin/mpisep.sh ${LM} ¥
                        "-npool 2 -nband 1 -ntg 1 -ndiag 4 -input ausurf.in"
```

6.3.10. PHASE/0

PHASE/0 is an OSS for first-principles electronic state calculation based on density functional theory. As of August 2021, only the vector node version is available.

The module provided in the vector node version is only the first-principles electronic state calculation package (phase). An example of a job script is as follows.

```
#!/bin/sh

#-------------- qsub option qsub option ----------------------
#PBS -q DBG
#PBS --group=sq-test
#PBS -l elapstim_req=00:10:00
#PBS --venode=1
#PBS -T necmpi
#-------------- Program execution Program execution ---------

module load BaseApp/2021
module load PHASE0/2020.01

export NMPI_SEPSELECT="3"
export NMPI_PROGINF="DETAIL"
export VE_PROGINF="DETAIL"
export VE_ACC_IO="1"
export LM=`which phase`
cd $PBS_O_WORKDIR

echo "Start at `date`"
mpiexec –venode -np 8 ${LM}
echo "End at `date`"
```

6.3.11. Paraview

(1) Use on HPC Front-end

If you want to use ParaView with HPC front-end, please log in to NICE DCV in advance. For how to use NICE DCV, refer to "3.2. How to use HPC front-end".

After log in to NICE DCV, do the following:

```
$ module load BaseApp/2021
$ module load ParaView/5.8.1
$ paraview
```

(2) Use on HPDAFront-end

When using ParaView with the HPDA front-end, use X terminal software, etc., log in to the HPDA front-end, and then execute the following.

```
$ module load BaseApp/2021
$ module load ParaView/5.8.1
$ paraview
```

6.3.12. Relion

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

```
$ module load BaseApp/2021
$ module load relion/3.1.1
$ relion
```

6.3.13. Visit

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

```
$ module load BaseApp/2021
$ module load VisItv/3.1.3
$ visit
```

6.4. How to Use National Project Applications

6.4.1. ABINIT-MP

ABINIT-MP is an application software which can perform efficient quantum chemical calculations of large molecules such as proteins based on the fragment molecular orbital (FMO) method. It includes graphical user interface system (named BioStation Viewer) to assist both preparation of input-data and analysis of ourput-data. The second-order perturbation calculation under a four-body fragment expansion (FMO4-MP2) is available as well.

CPU node version and vector node version are available, but there are restrictions on the modules that can be used.

Node type	with F function		without F function	
	SMP	non-SMP	SMP	non-SMP
cpu	abinitmp_smp-fint	abinitmp-fint	abinitmp_smp	abinitmp
vec	-	-	-	abinitmp-vec

Execute the program as a batch job. The following is an example of executing the sample program.

<CPU node>

```
#!/bin/bash
#PBS -q SQUID
#PBS -l cpunum_job=76
#PBS --group="group name"
#PBS -l elapstim_req=01:00:00
#PBS -b 1
#PBS -T intmpi

#PBS -v NUM_PROCS=19
#PBS -v NUM_THREADS=4
```

```
#PBS -v OMP_NUM_THREADS=${NUM_THREADS}
#PBS -v OMP_STACKSIZE=5G

#PBS -v BINARY_NAME=abinitmp_smp
#PBS -v FILE_NAME=gly5
#PBS -v NUM_CORE=_16n-144p-4t-smp-2020.1-intel
#PBS -v OUT_NAME=${FILE_NAME}${NUM_CORE}
ulimit –s unlimited

module load BaseApp/2021
module load abinit-mp/1.22

cd ${PBS_O_WORKDIR}
mpirun ${NQSV_MPIOPTS} -np ${NUM_PROCS} ${BINARY_NAME} ¥
                                                < ${FILE_NAME}.ajf > ${OUT_NAME}
```

<VEC node>

```
#!/bin/bash
#PBS -q SQUID
#PBS --group="group name"
#PBS -T necmpi
#PBS -l elapstim_req=01:00:00
#PBS --venode=4
#PBS -v NUM_PROCS=40
#PBS -v FILE_NAME=gly5
#PBS -v AJF_NAME=${FILE_NAME}.ajf
#PBS -v VE_FORT5=${AJF_NAME}

module load BaseApp/2021
module load abinit-mp/1.22.VEC

cd ${PBS_O_WORKDIR}
BINARY_NAME=`which abinitmp-vec`

mpirun -np ${NUM_PROCS} ${BINARY_NAME}
```

6.4.2. HΦ

HΦ is an open source software package analyzing the quantum many-body lattice models based on the numerical exact diagonalization method. The software is capable of computing the ground state, low-excitation state, excitation spectra and thermodynamic quantities at the finite temperature of the various many-body quantum systems, such as multi-orbital Hubbard model,Heisenberg model, and Kondo lattice model.

Execute the program as a batch job. The following is an example of executing the sample program.

```
#!/bin/bash
#PBS -q SQUID
#PBS -l cpunum_job=76
#PBS --group="group name"
#PBS -l elapstim_req=01:00:00
#PBS -b 1
#PBS -v OMP_NUM_THREADS=8

module load BaseApp/2021
module load HPhi/3.4.0

cd $PBS_O_WORKDIR
mpirun ${NQSV_MPIOPTS} -np 4 HPhi -s stan.in
```

6.4.3. MODYLAS

MODYLAS is an Open Source Software (license permission required) package for general-purpose classical molecular dynamics. The software supports various methods required for the molecular dynamics computation in nano-molecules and biomolecules including the Fast Multiple Method (FMM) of treating long-range electrostatic interactions.

Execute the program as a batch job. The following is an example of executing the sample program.

```
#!/bin/bash
#PBS -q SQUID
#PBS -l cpunum_job=76
```

```
#PBS --group="group name"
#PBS -l elapstim_req=01:00:00
#PBS -b 8
#PBS -T intmpi
#PBS -v OMP_NUM_THREADS=24
#PBS -v PARALLEL=24

module load BaseApp/2021
module load MODYLAS/1.0.4

cd $PBS_O_WORKDIR
mpirun ${NQSV_MPIOPTS} -np 8 modylas pyp111
```

6.4.4. NTChem

NTChem is an application software package for the molecular electronic structure calculation based on the Gaussian-type basis functions. Various kinds of functionalities for quantum chemistry approaches are available to users. It is designed for high-performance quantum chemistry calculations on massively parallel computers.

Execute the program as a batch job. The following is an example of executing the sample program.

```
#!/bin/bash
#PBS -q SQUID
#PBS -l cpunum_job=76
#PBS –group="group name"
#PBS -l elapstim_req=01:00:00
#PBS -T intmpi
#PBS -b 1
#PBS -v OMP_NUM_THREADS=1

module load BaseApp/2021
module load NTChem/12.0

cd ${PBS_O_WORKDIR}
```

```
./CH3CHO.sh
```

<CH3CHO.sh>

```
#!/bin/bash

export mol=CH3CHO
export curdir=`pwd`
export wrkdir=$curdir/ntchem

mkdir -p $curdir/ntchem

export nprocs=16
export mpirun="mpirun ${NQSV_MPIOPTS} -np $nprocs "
export sub=_mpi

export ntchem=$curdir/ntchem.sh

rm -f $wrkdir/$mol.* $wrkdir/INPUT
cd $wrkdir
cp -pr $curdir/${mol}.inp $wrkdir/INPUT
mpirun basinp${sub}.exe > $curdir/${mol}.log 2>&1
dlfind.exe < $curdir/${mol}.inp > $curdir/${mol}.out 2>&1
#
mv Coords.xyz $curdir/${mol}.xyz
cd $curdir
```

<ntchem.sh>

```
$mpirun mdint1${sub}.exe >> $curdir/$mol.log 2>&1
$mpirun scf${sub}.exe >> $curdir/$mol.log 2>&1
$mpirun scfgrad${sub}.exe >> $curdir/$mol.log 2>&1
```

6.4.5. OpenMX

OpenMX is an Open Source Software package for the first-principles calculations. Using atom-localized basis functions and pseudopotential method, the software performs the electronic state computations for a wide range of physics systems such as crystals, interfaces, solutions.

Execute the program as a batch job. The following is an example of executing the sample program.

```
#!/bin/bash
#PBS -q SQUID
#PBS -l cpunum_job=76
#PBS --group="group name"
#PBS -l elapstim_req=01:00:00
#PBS -T intmpi
#PBS -b 2
#PBS -v OMP_NUM_THREADS=76

module load BaseApp/2021
module load OpenMX/3.9

cd $PBS_O_WORKDIR
mpirun ${NQSV_MPIOPTS} -np 2 openmx Methane.dat -nt 76
```

6.4.6. SALMON

SALMOM is an Open Source Software package based on first-principles calculations targeting researches related to interaction between light and materials. The software can be used for photo-excited electron dynamics and light propagation simulations based on time-dependent density functional theory using real-time and real-space grid method.

Execute the program as a batch job. The following is an example of executing the sample program.

```
#!/bin/bash
#PBS -q SQUID
#PBS -l cpunum_job=76
#PBS --group="group name"
#PBS -l elapstim_req=01:00:00
#PBS -b 1
#PBS -v OMP_NUM_THREADS=2
```

```
module load BaseApp/2021
module load salmon/2.0.1

cd $PBS_O_WORKDIR
mpirun ${NQSV_MPIOPTS} -np 4 salmon < C2H2_gs.inp
```

6.4.7. SMASH

SMASH is an Open Source Software package for massively parallel quantum chemical calculations. The software is capable of energy and geometry optimization calculations of the Hartree-Fock method, the DFT method and the MP2 method for nano-sized molecules without splitting or fragmenting.

Execute the program as a batch job. The following is an example of executing the sample program.

```
#!/bin/bash
#PBS -q SQUID
#PBS -l cpunum_job=76
#PBS --group="group name"
#PBS -l elapstim_req=01:00:00
#PBS -b 1
#PBS -v OMP_NUM_THREADS=38

module load BaseApp/2021
module load smash/2.2.0

cd $PBS_O_WORKDIR
mpirun ${NQSV_MPIOPTS} -np 2 smash < b3lyp-energy.inp
```

7. Advanced File Transfer Method

7.1. File Transfer with Web Browser

Nextcloud is prepared as a method of transferring files to SQUID with a web browser.

Files saved via Nextcloud are saved in the following path in SQUID and can be used from the front-end environment and calculation environment.

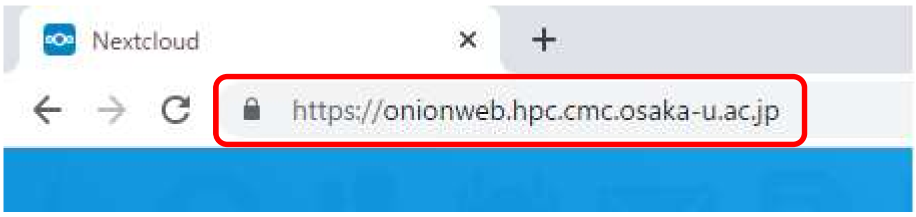
```
/sqfs/home/(User number)/OnionWeb/
```

ex : User number is "user001"　/sqfs/home/user001

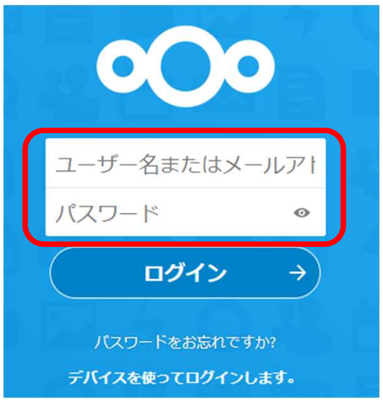
7.1.1. Login

Open a web browser and enter the following URL.

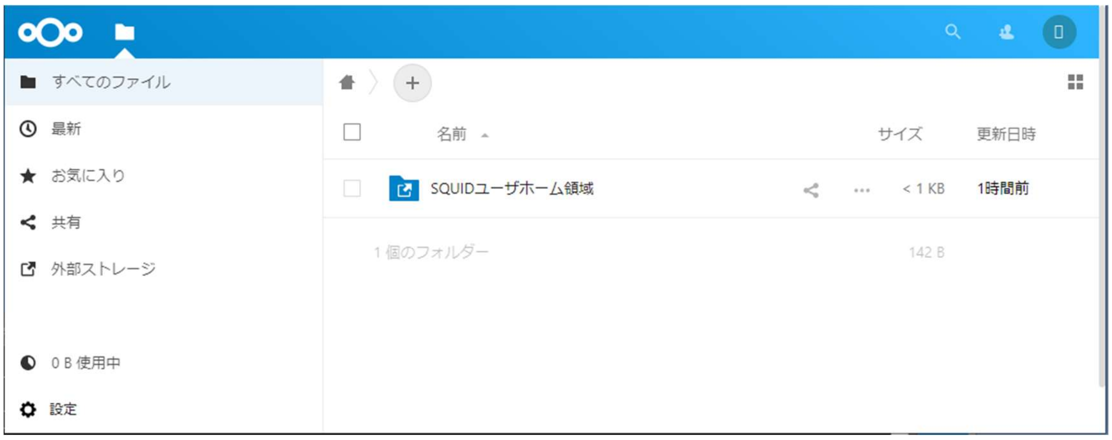
URL : https://onionweb.hpc.cmc.osaka-u.ac.jp



The following login screen will be displayed. Enter your user name and password and click "Login".


ユーザー名またはメールアド
パスワード
ログイン
パスワードをお忘れですか?
デバイスを使ってログインします。

After logging in, the following screen will be displayed.



After logging in, your home area will be automatically mounted and displayed as the "SQUID User Home Area".

7.1.2. Basic Usage

➢ **Upload Method**

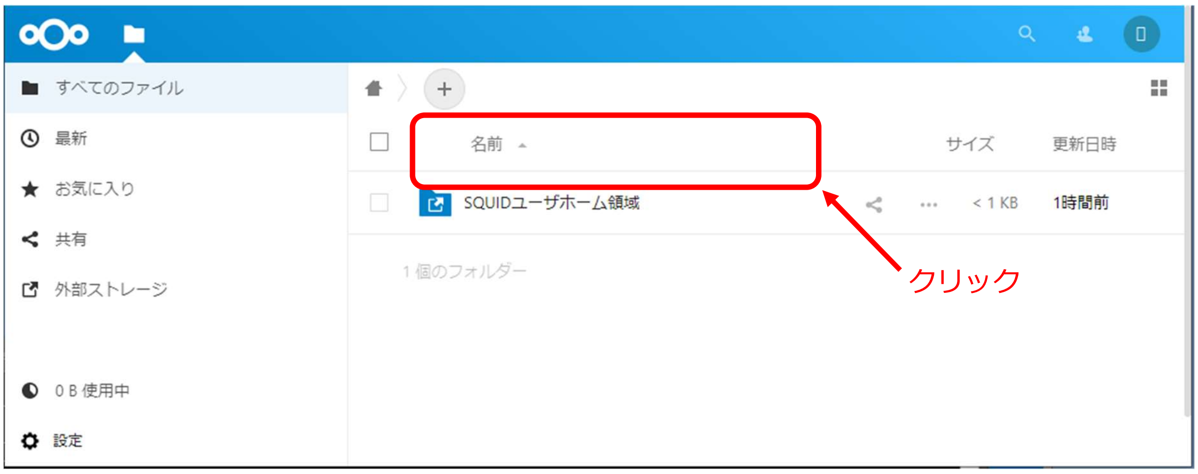
Files can only be uploaded into the added storage in "SQUID User Home Area" and "7.1.7 Adding External Storage" described later.

(1) Upload By Drag And Drop

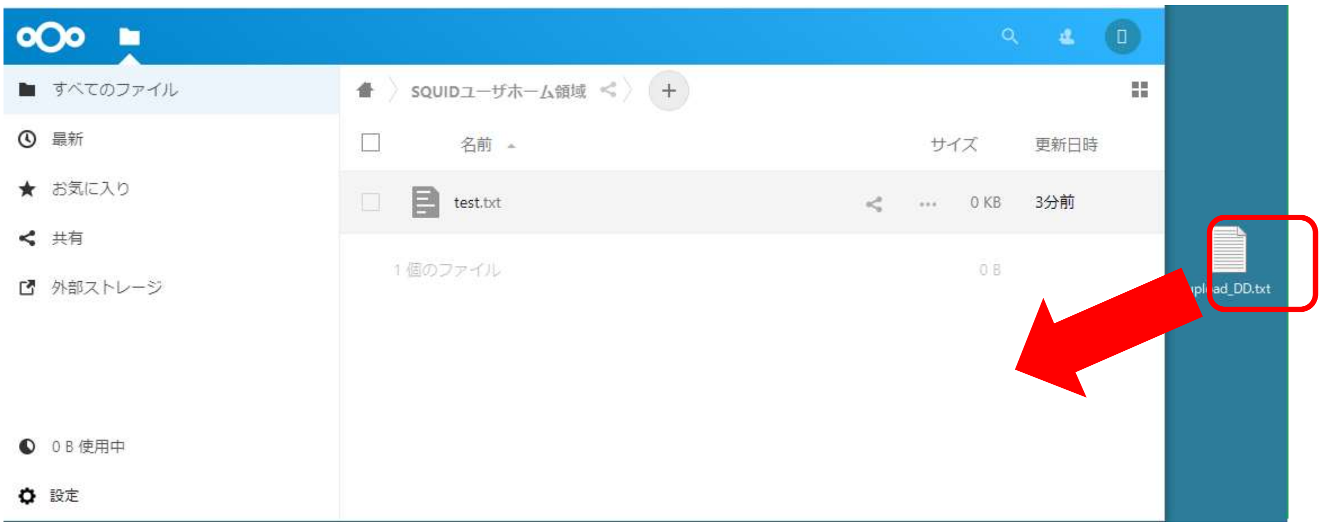
Open the location you want to upload and drag and drop the file you want to upload.

In the following example, the file will be uploaded directly under the "SQUID user home area".

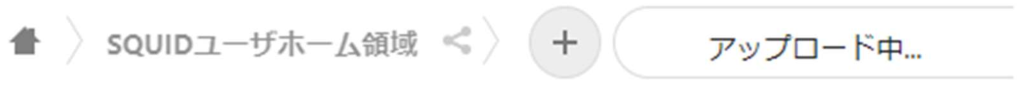
Click SQUID User Home Area.



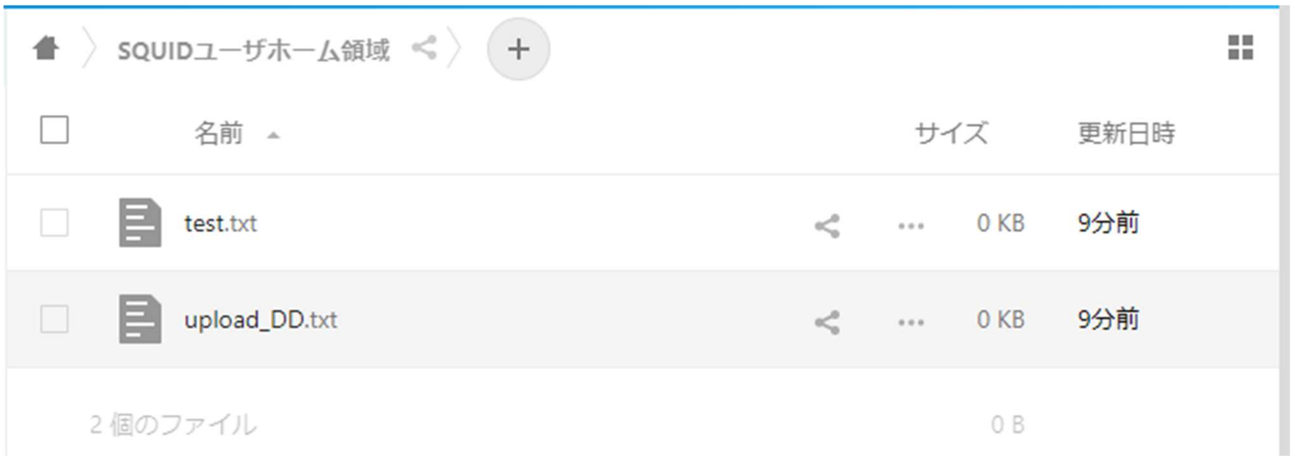
Drag and drop the file you want to upload.



When the upload starts, the following progress bar will be displayed, so wait for a while.



When the dragged and dropped file is displayed, the upload is complete.

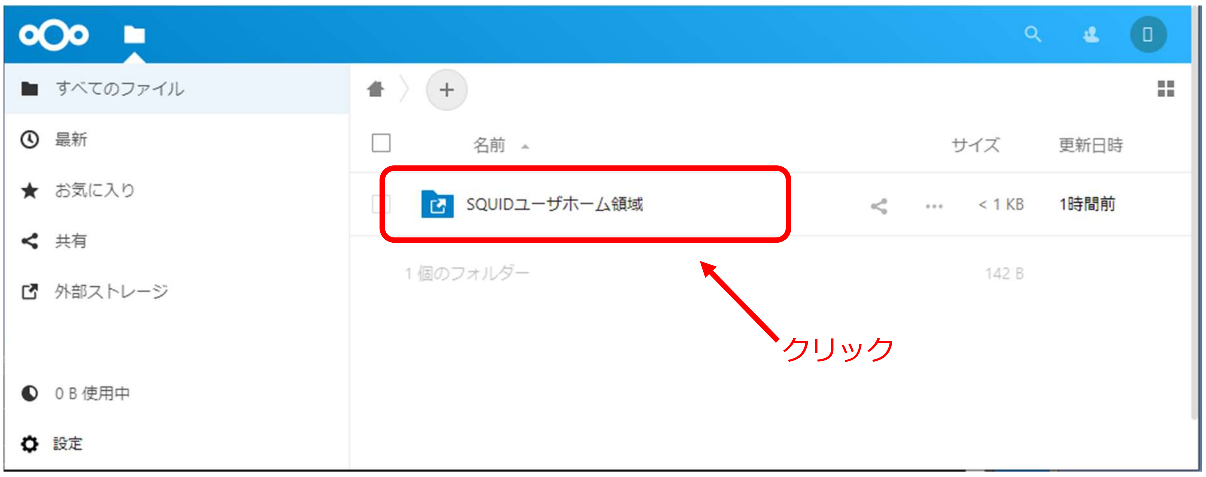


(2) Upload By Reference

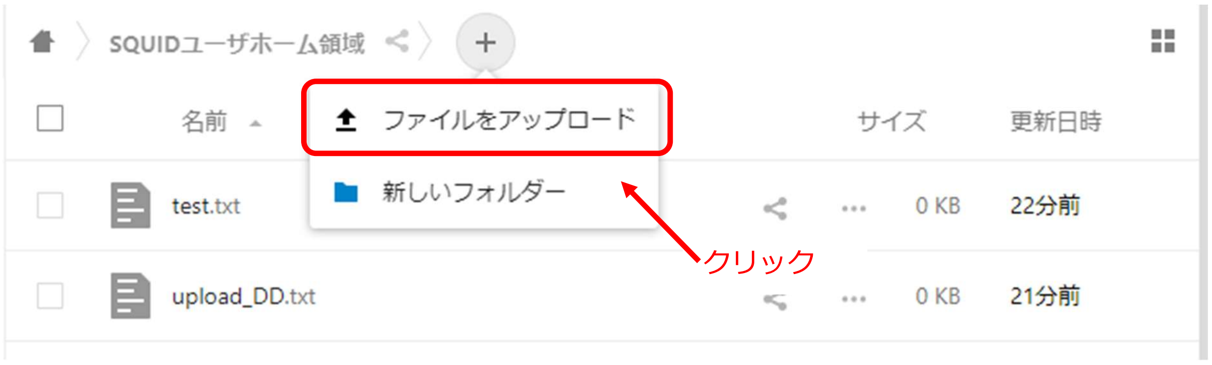
Open the location you want to upload and select the file you want to upload.

In the following example, the file will be uploaded directly under the "SQUID user home area".

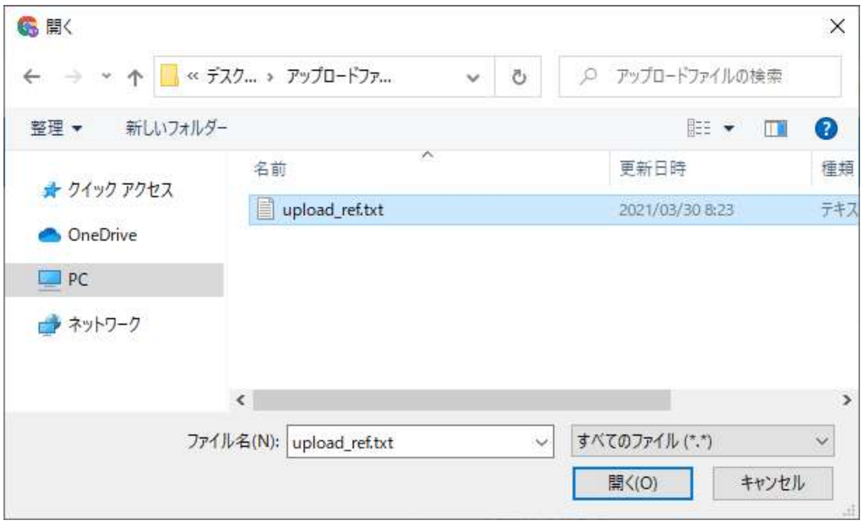
Click SQUID User Home Area.



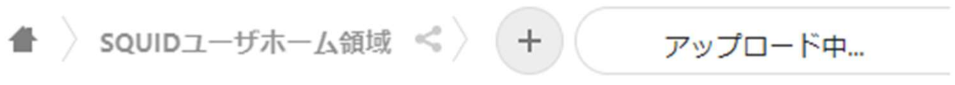
Click 「⊕」and then click "Upload File" that appears.



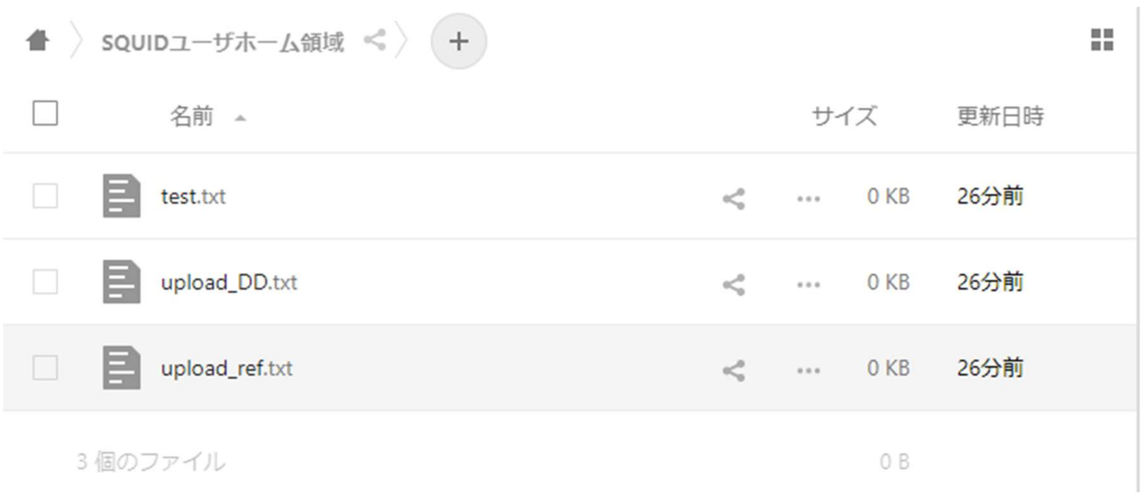
A file reference dialog will be displayed. Select the file you want to upload and select it. Click "Open"



When the upload starts, the following progress bar will be displayed, so wait for a while.



When the selected file is displayed, the upload is complete.


SQUIDユーザホーム領域

	名前	サイズ	更新日時
	test.txt	0 KB	26分前
	upload_DD.txt	0 KB	26分前
	upload_ref.txt	0 KB	26分前
	3 個のファイル	0 B	

➢ **How To Use Download**

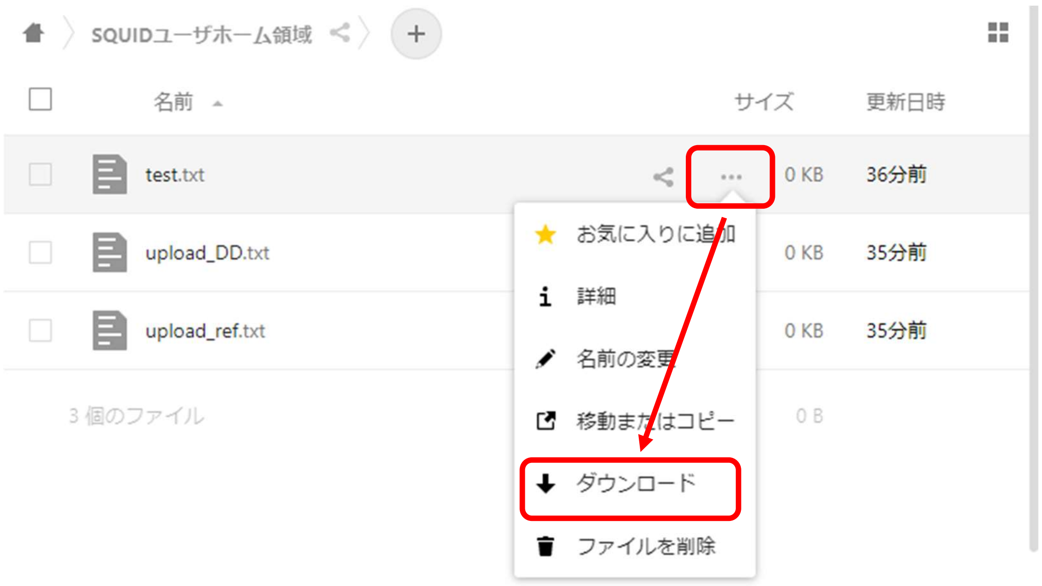
(1) Single Download

If you want to download only one file, follow the procedure below.

The example from the following image downloads "test.txt" in the "SQUID User Home Area".

Click "..." on the right side of the file you want to download, and "Download" is displayed. Click to start downloading.

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


SQUIDユーザホーム領域

	名前	サイズ	更新日時
	test.txt	0 KB	36分前
	upload_DD.txt	0 KB	35分前
	upload_ref.txt	0 KB	35分前
	3 個のファイル	0 B	

(2) Bulk Download

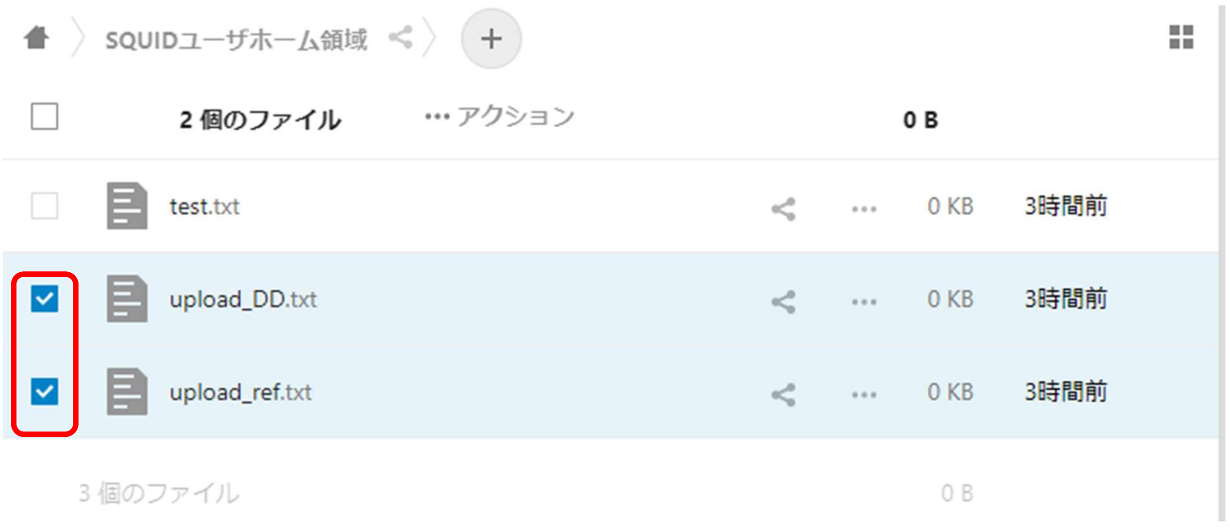
You can also download multiple files at once.

If you download them all at once, the files compressed in zip format will be downloaded.

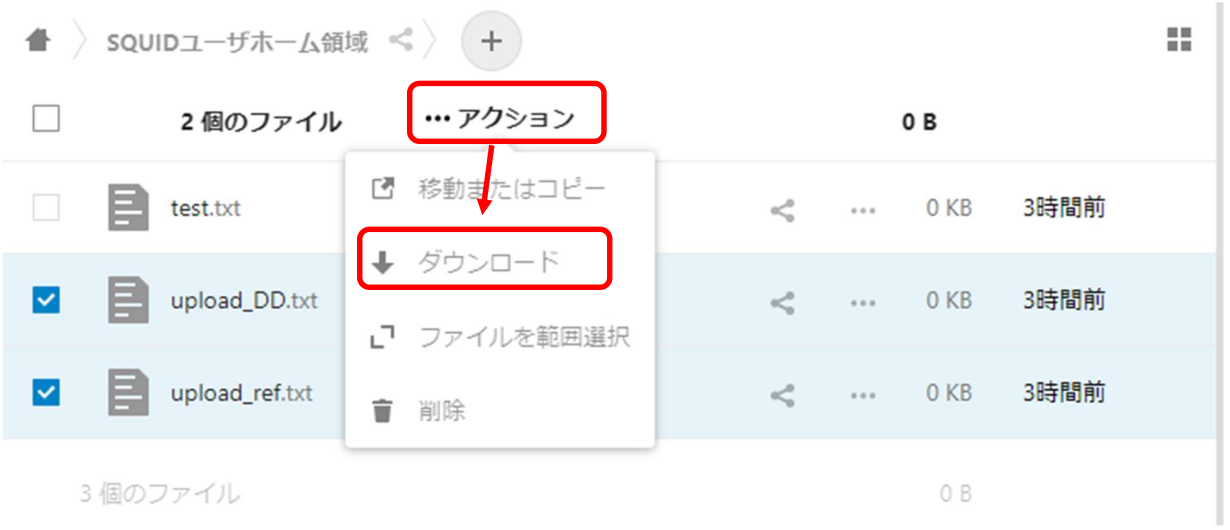
Please decompress it separately.

In the example from the following image, "upload_DD.txt" and "upload_ref.txt" are downloaded together.

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



Click "…アクション" at the top of the file name list, and then click "Download" that appears.



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

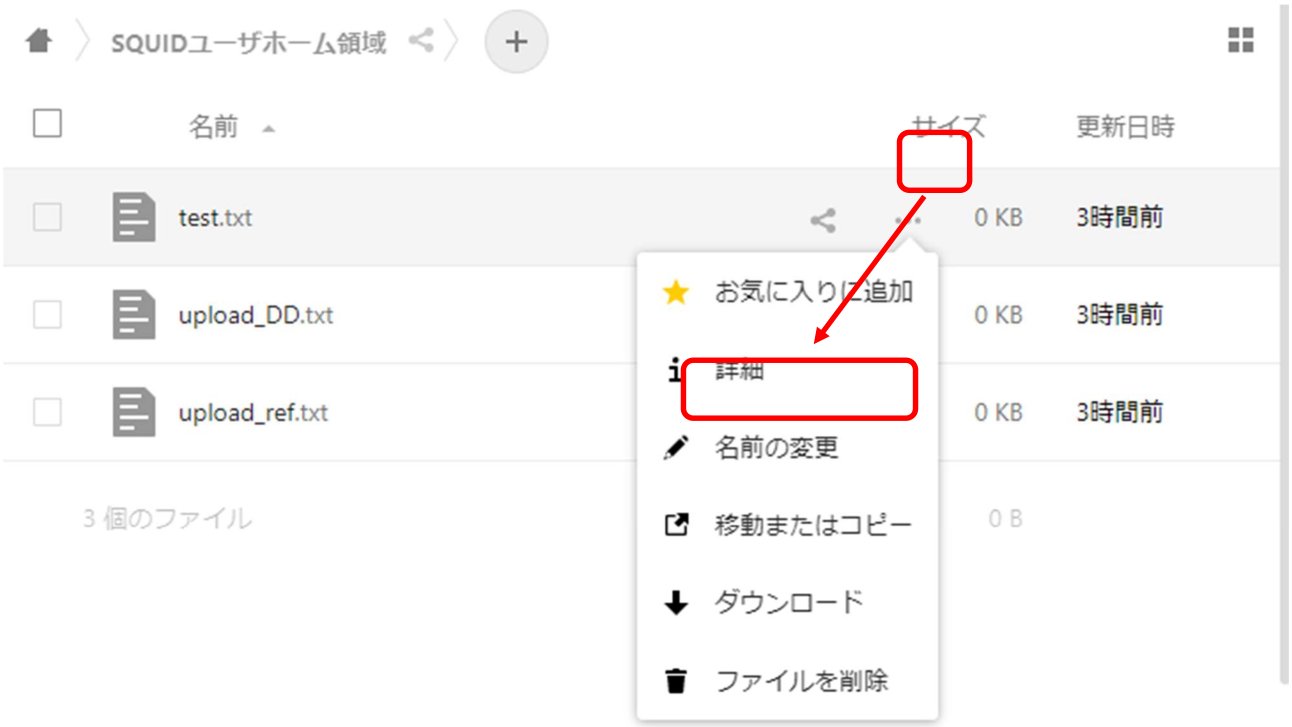
➢ **Change Folder Name / File Name**

Follow the procedure below to change the folder name and file name.

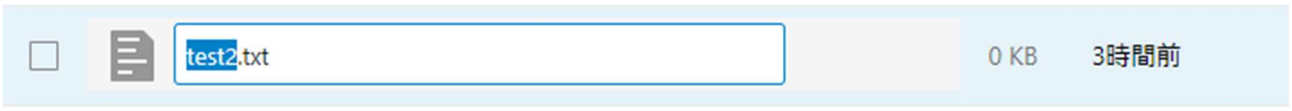
In the example from the following image, rename "test.txt" to "test2.txt".

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

Click "..." to the right of the folder / file you want to rename, then click Rename.



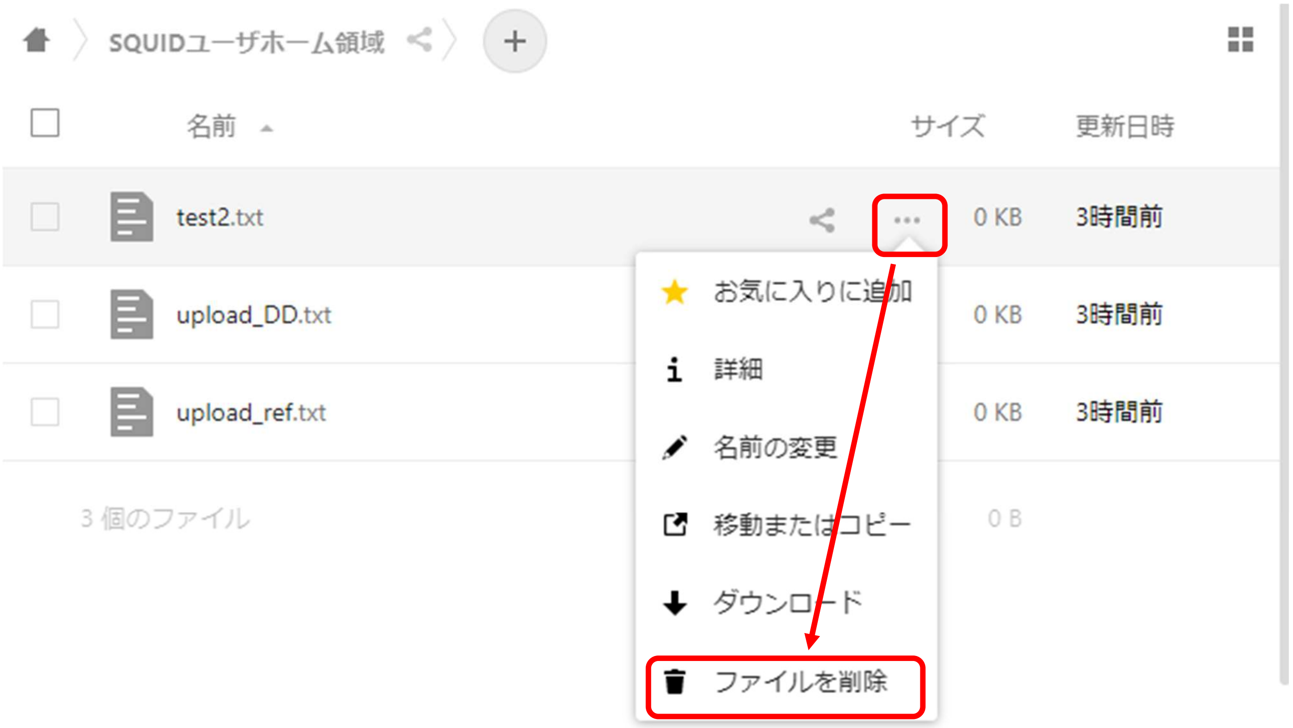
You will be able to edit the target file name, so change it to any name and enter the enter key.



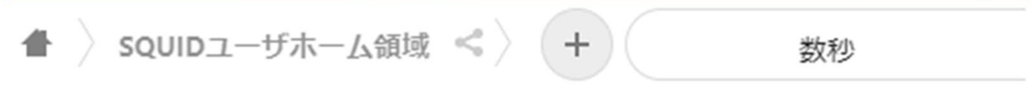
Specifying the save destination of the download file depends on the settings of the web browser you are using.

➢ **Delete File**

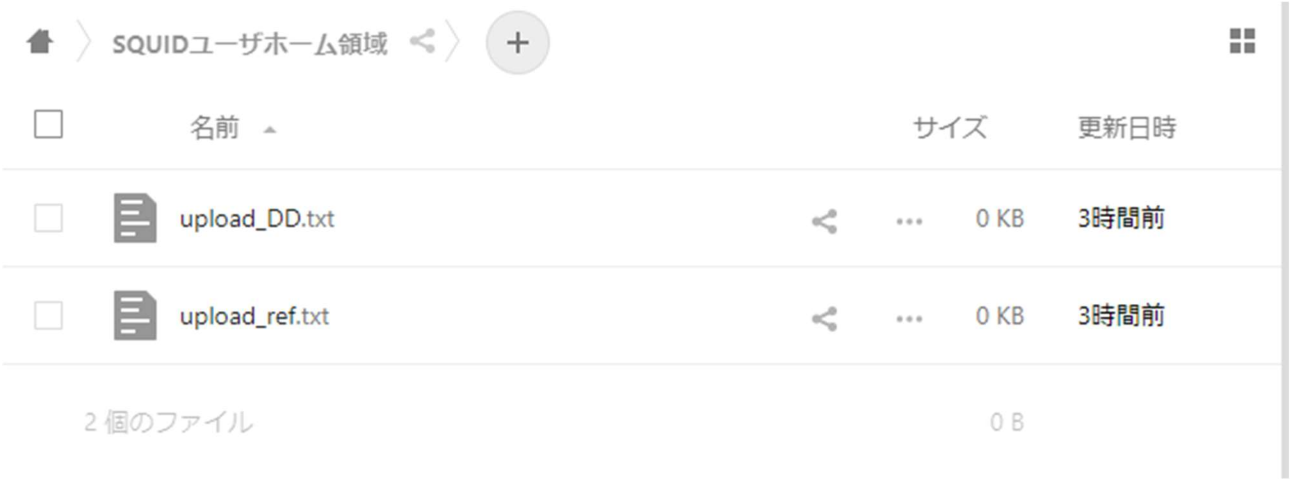
Follow the procedure below to delete folders and files. In the following example, "test2.txt" is deleted. Click "..." to the right of the folder / file you want to delete, and then click "Delete File" that appears.



When the deletion starts, the following progress bar will be displayed, so wait for a while.



ファイルの一覧画面から選択したフォルダ・ファイルが消えていれば削除完了となります。



➤ **Favorite**

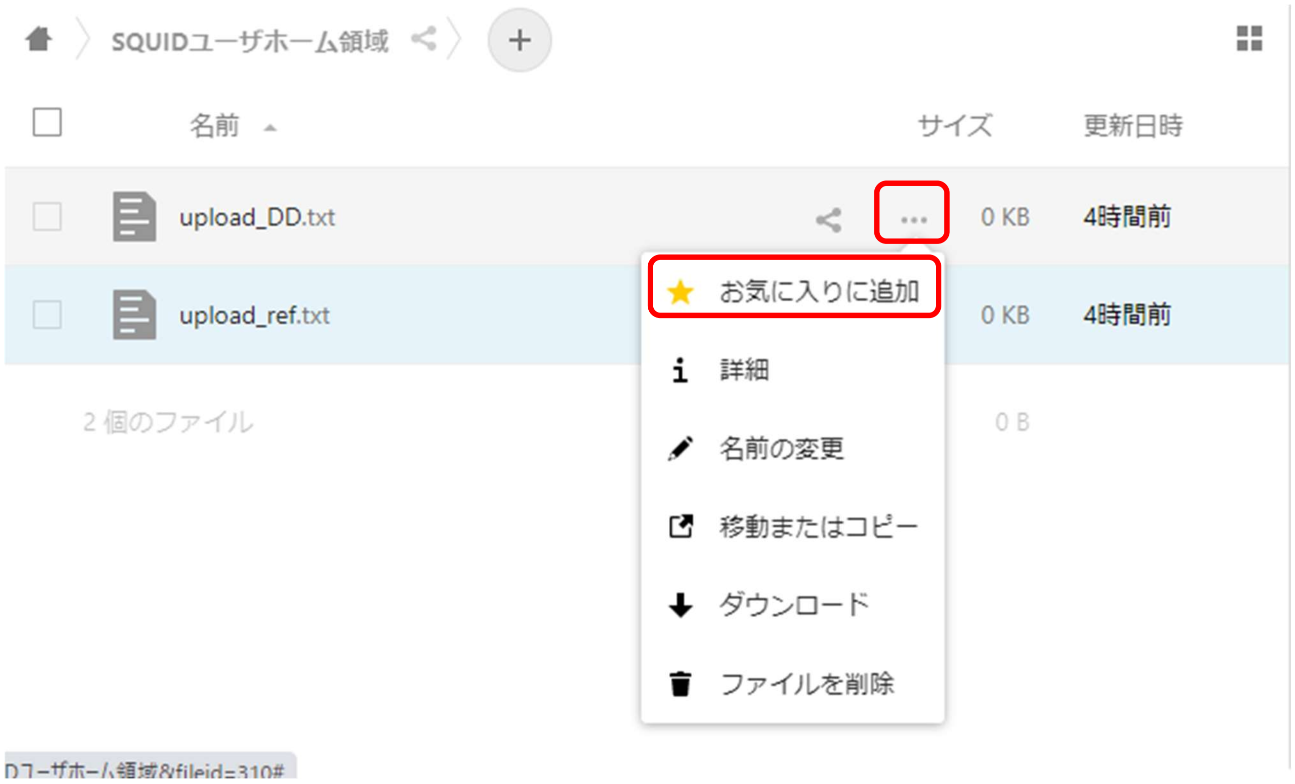
You can quickly refer to frequently used folders and files by registering them as favorites.

(1) Favorite Registration

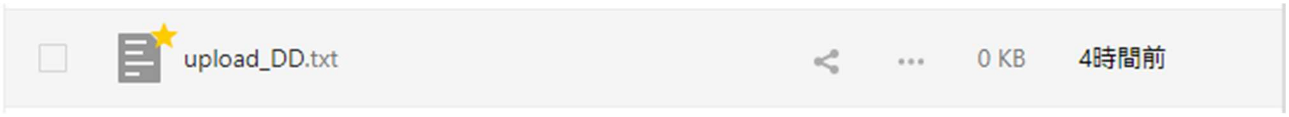
Follow the procedure below to register folders and files as favorites.

In the following example, "upload_DD.txt" is registered as a favorite.

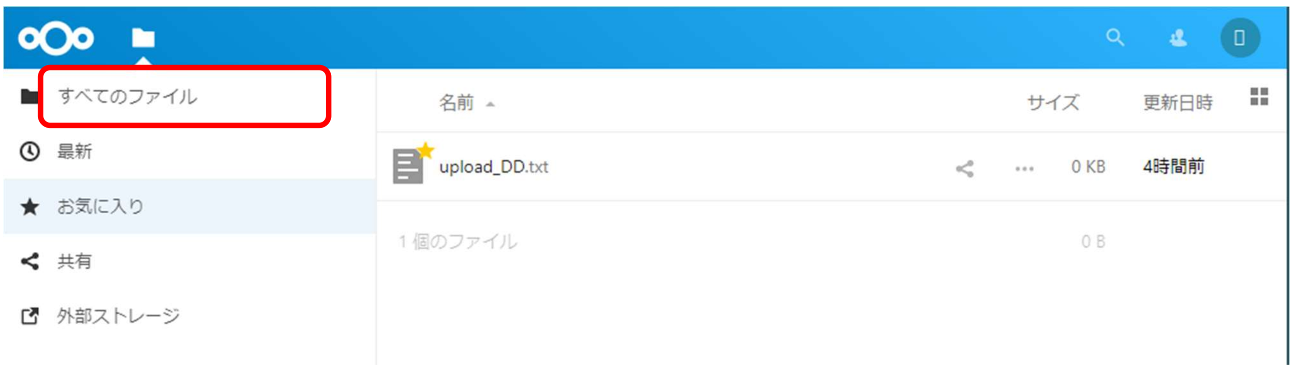
Click "..." on the right side of the folder / file you want to add to your favorites, and then click "お気に入りに追加" that appears.



When registered as a favorite, a "★" mark will be added to the upper right of the icon.



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

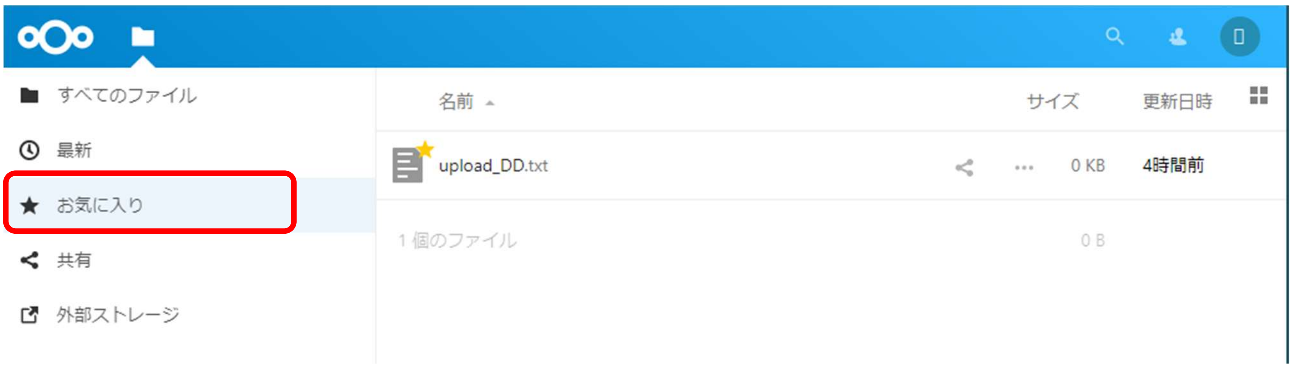


(2) Delete From Favorites

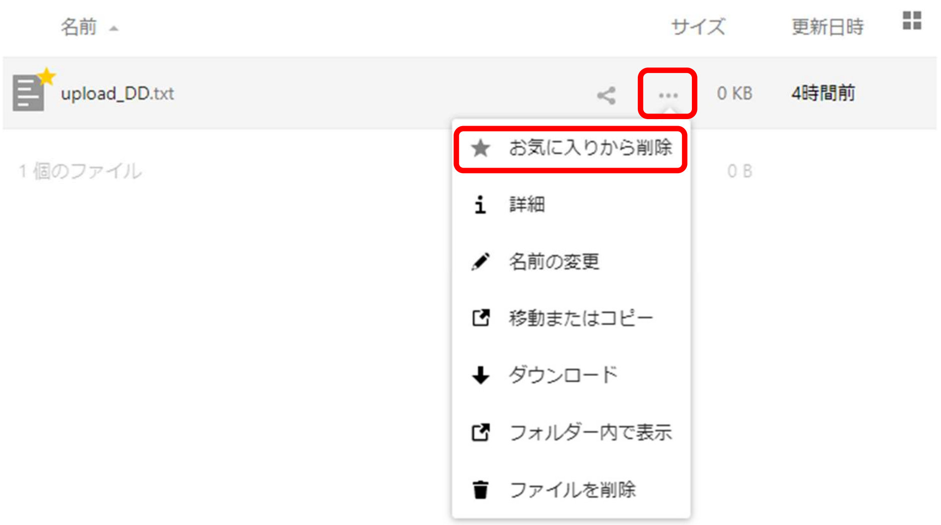
To delete a folder / file as a favorite, follow the procedure below.

The example from the following image removes "upload_DD.txt" from your favorites.

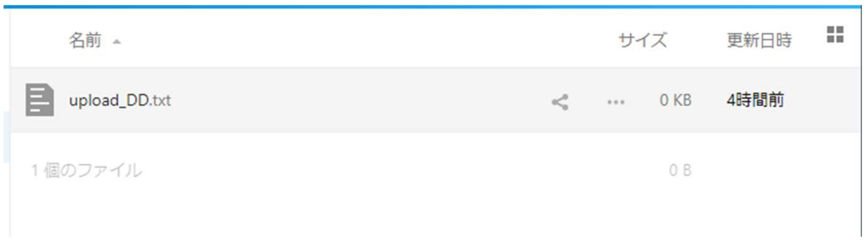
Click "Favorites" on the left menu.



Click "..." on the right side of the file you want to delete from Favorites, and then click "Delete from Favorites" that appears.



If the "★" mark disappears from the previous file, the deletion from favorites is complete.



7.1.3. Sharing Folders and Files

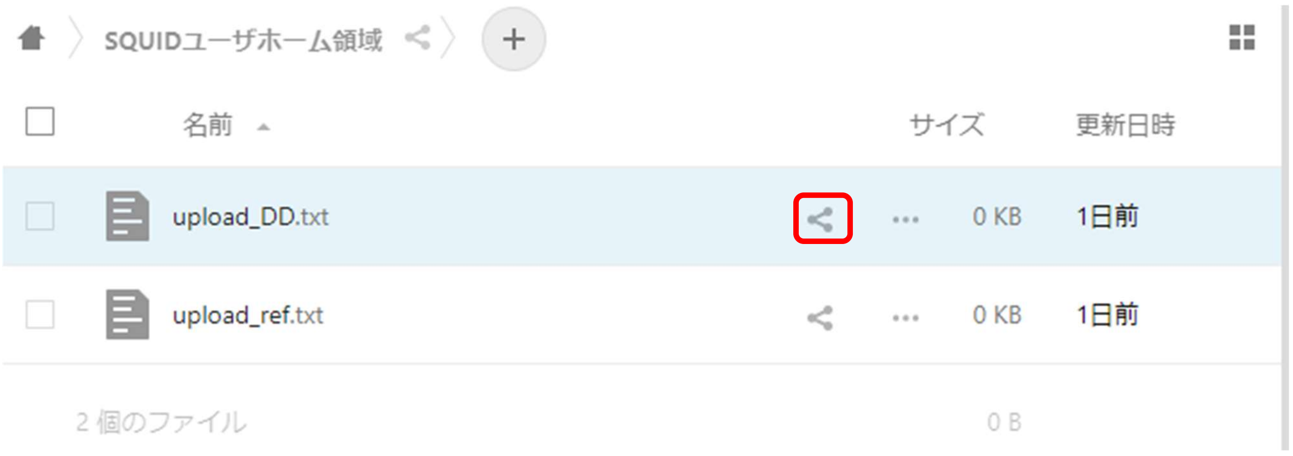
By specifying a folder or file and issuing a URL for sharing, you can share the folder / file with people who do not have an account in the data aggregation environment.

(1) Folder / File Sharing Settings (URL Sharing)

Follow the steps below to share folders and files.

The following example shares "upload_DD.txt".

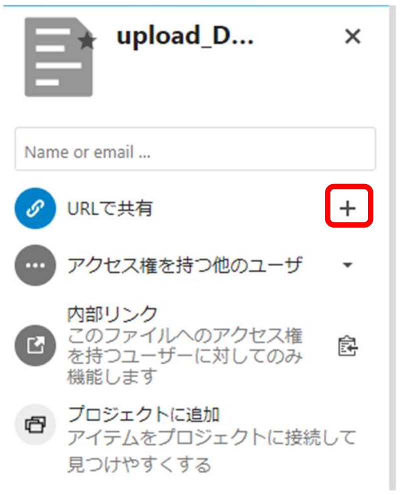
Click the sharing mark "<" on the right side of the folder / file you want to share.



SQUIDユーザホーム領域

	名前			サイズ	更新日時
	upload_DD.txt	<	...	0 KB	1日前
	upload_ref.txt	<	...	0 KB	1日前
	2 個のファイル			0 B	

The following menu will be displayed on the right side. Click "+" on the right side of " URL で共有".



upload_D...

Name or email ...

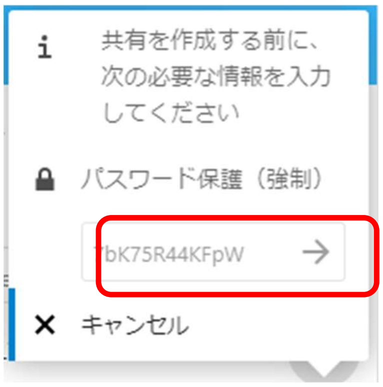
URLで共有 +

アクセス権を持つ他のユーザ

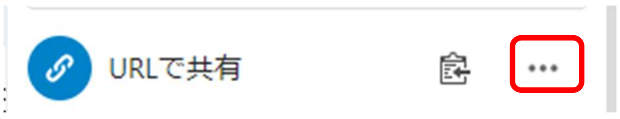
内部リンク
このファイルへのアクセス権を持つユーザーに対してのみ機能します

プロジェクトに追加
アイテムをプロジェクトに接続して見つけやすくする

The following screen will be displayed. Make a note of the password displayed in "パスワード保護（強制）", or set an arbitrary password. If you set an arbitrary password, it must be 12 characters or more. After that, click "→" on the right side of the password.

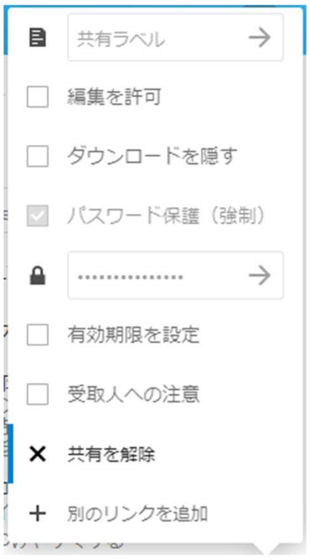


The icon on the right side of "Share by URL" will change as follows, so click "..."



The following menu will be displayed. Change the settings as necessary.

After making changes, click outside the menu.



共有ラベル：Labels for management by shared owners

編集を許可：It will not be enabled due to functional restrictions.

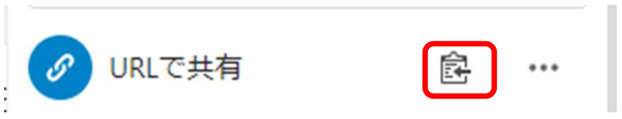
ダウンロードを隠す：Even if you connect to the shared URL, the file will be hidden and will not be enabled.

パスワード保護（強制）: It cannot be canceled. It can be changed arbitrarily. After making changes, click "→".

有効期限を設定：You can set the expiration date of the shared URL.

受取人への注意：You can add a message in the upper right corner of the shared URL.

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



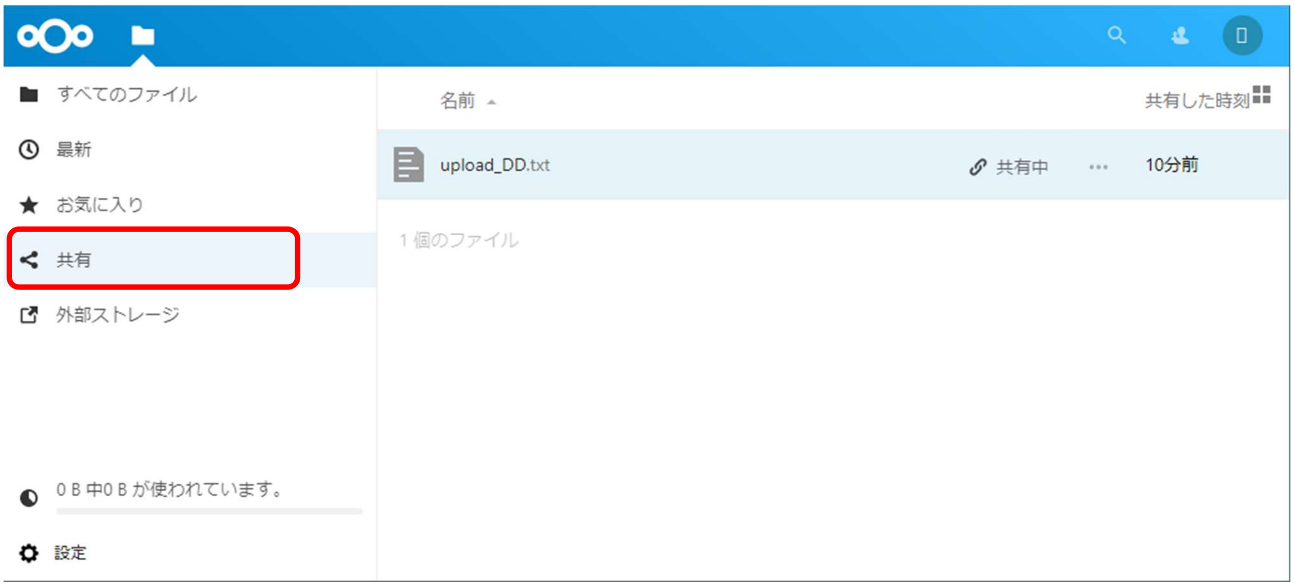
Send the password you wrote down and the copied URL to the person you want to share by e-mail.

(2) Canceling Folder / File Sharing

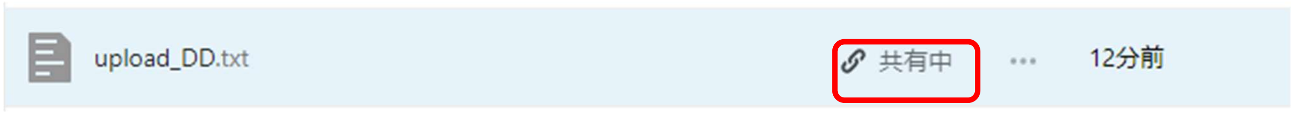
To unshare a folder / file, follow the procedure below.

In the following example, "upload_DD.txt" is unshared.

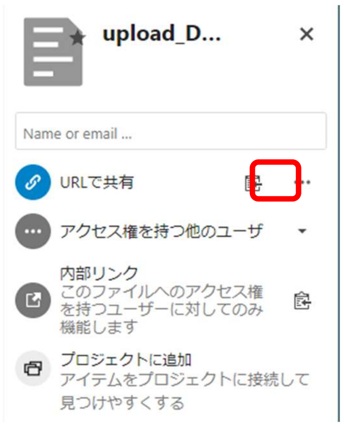
Click "Share" on the left menu.



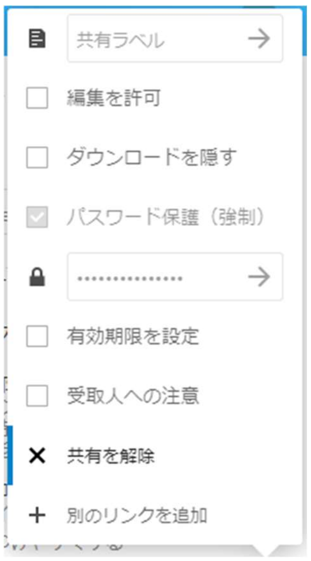
Click Sharing to the right of the folder / file you want to unshare.



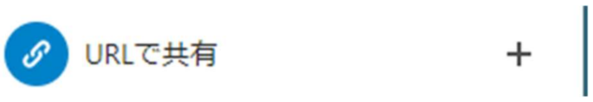
The following menu will be displayed on the right side, so click "..." on the right side of " URL で共有".


upload_D...
Name or email ...
URLで共有
アクセス権を持つ他のユーザ
内部リンク
このファイルへのアクセス権を持つユーザーに対してのみ機能します
プロジェクトに追加
アイテムをプロジェクトに接続して見つけやすくする

The following menu will be displayed. Click "共有を解除".



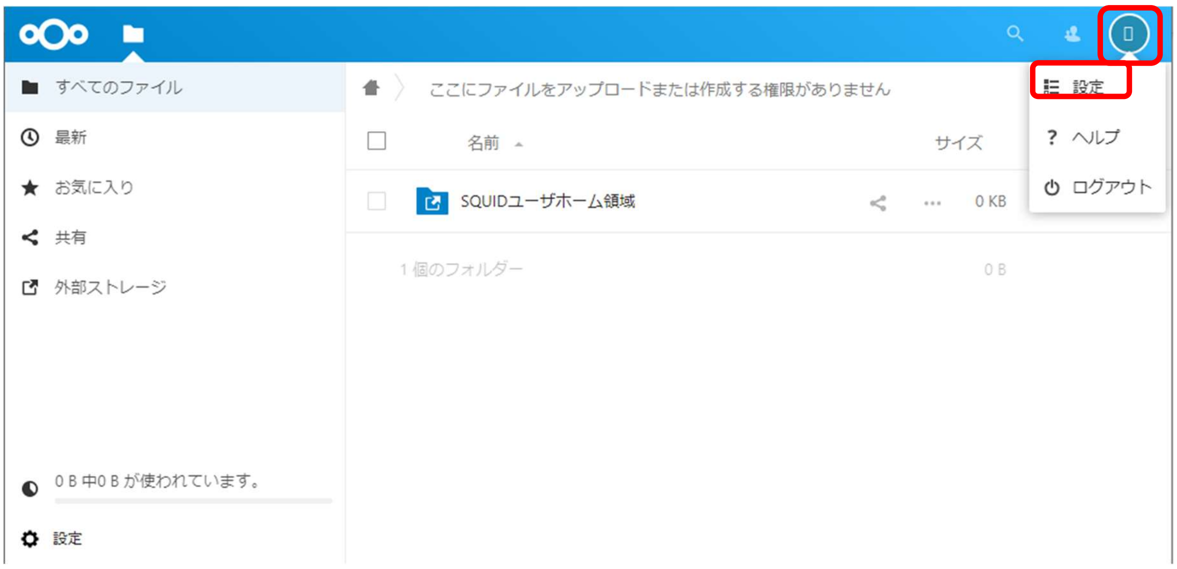
If the icon on the right side of " URL で共有"" changes as shown below, sharing cancellation is complete.



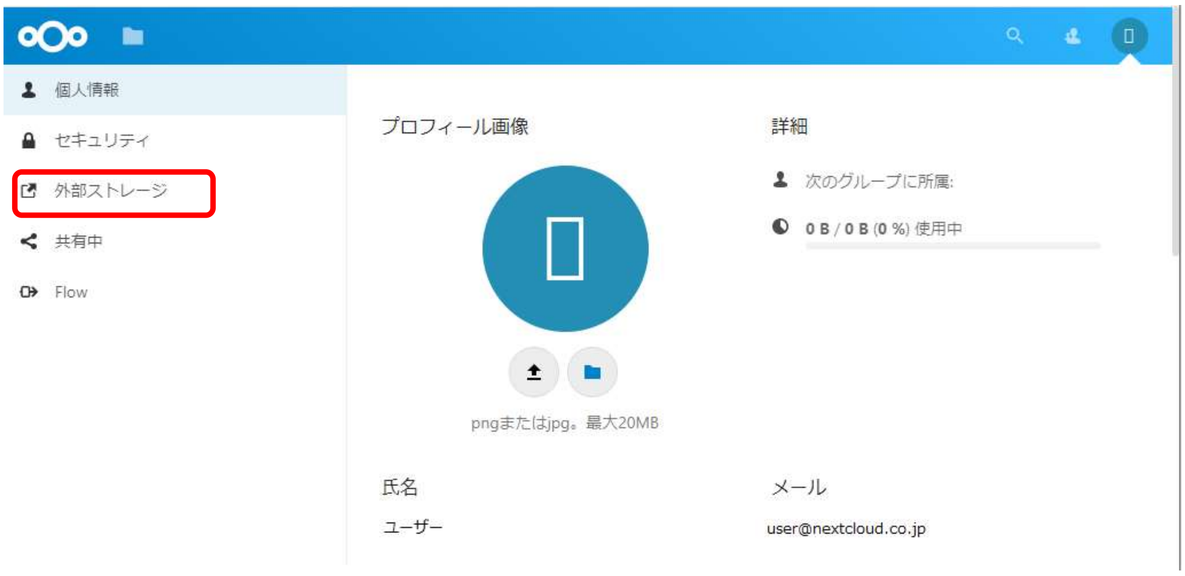
7.1.4. Addition Of External Storage

By adding external storage, it is possible to use storage compatible with Amazon S3 API from the Nextcloud environment. The archive storage described in "1.3.1. Features of each environment" can also be used from Nextcloud in this procedure.

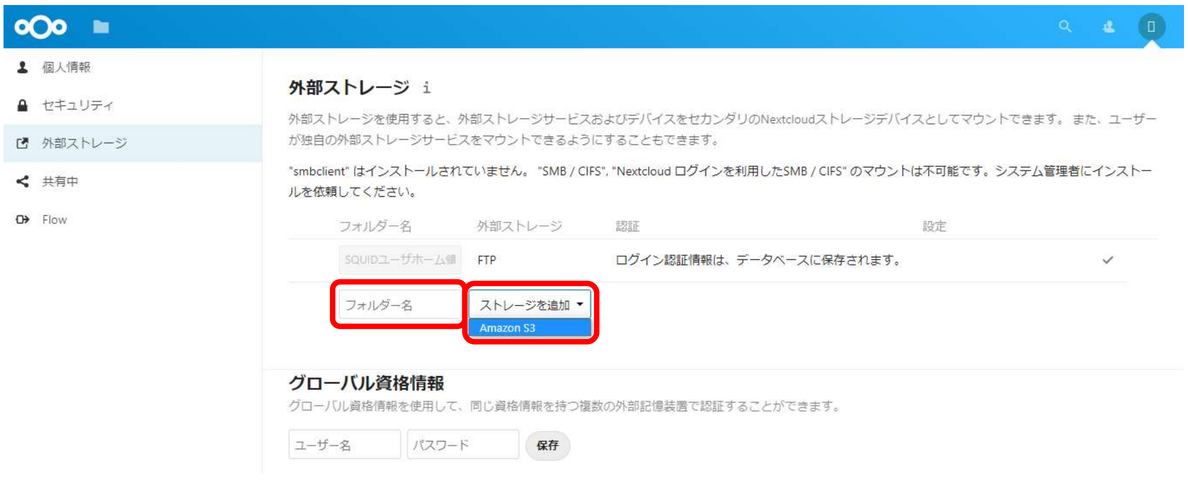
Click the user icon at the top right of the screen, and click "Settings" from the displayed menu.



Click "External Storage" on the left menu.



Enter an arbitrary name in "Folder Name" of "External Storage" and click "Add Storage" → "Amazon S3".



外部ストレージ i

外部ストレージを使用すると、外部ストレージサービスおよびデバイスをセカンダリのNextcloudストレージデバイスとしてマウントできます。 また、ユーザーが独自の外部ストレージサービスをマウントできるようにすることもできます。

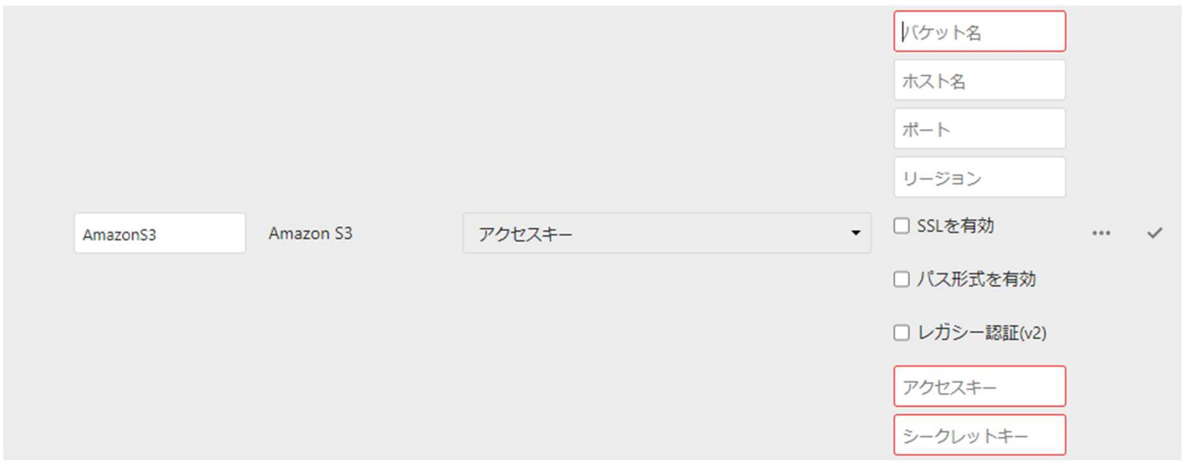
"smbclient" はインストールされていません。 "SMB / CIFS", "Nextcloud ログインを利用したSMB / CIFS" のマウントは不可能です。システム管理者にインストールを依頼してください。

フォルダー名	外部ストレージ	認証	設定
SQUIDユーザホーム領	FTP	ログイン認証情報は、データベースに保存されます。	✓
フォルダー名	ストレージを追加 ▾ / Amazon S3		

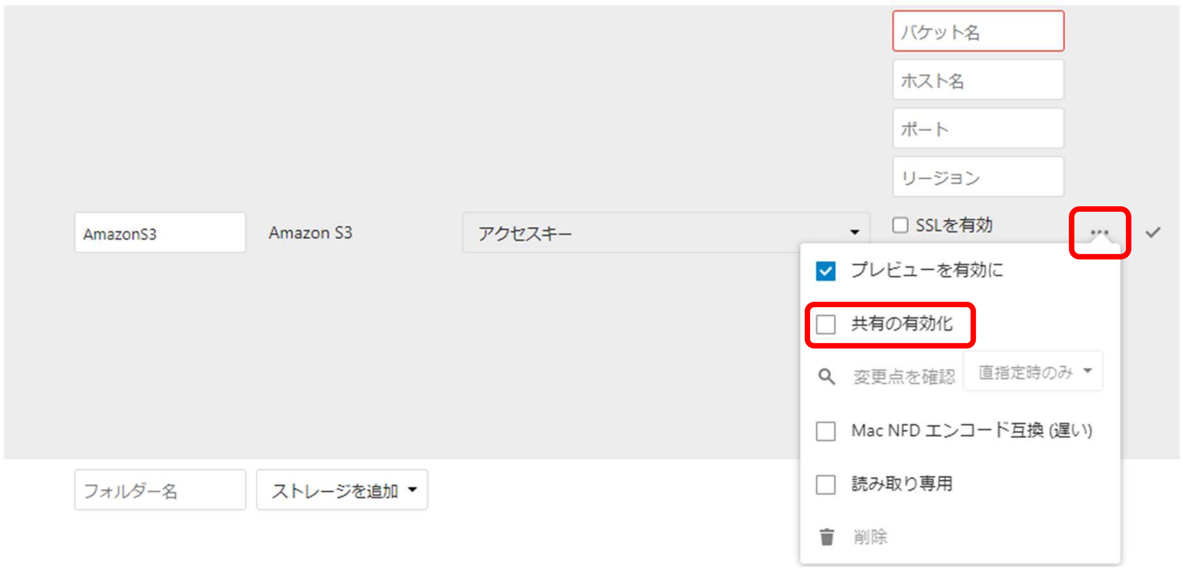
グローバル資格情報

グローバル資格情報を使用して、同じ資格情報を持つ複数の外部記憶装置で認証することができます。

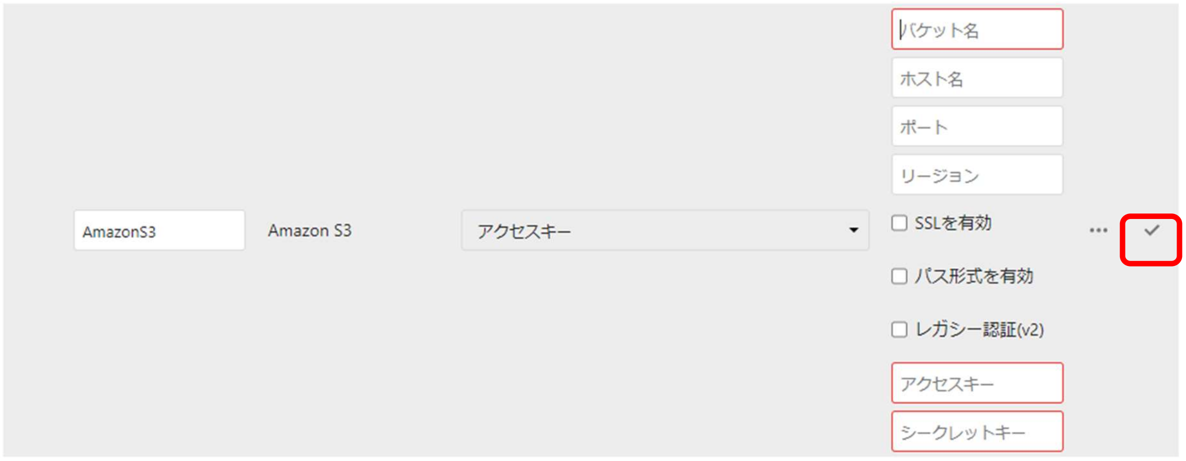
The following setting items will be displayed, so enter the required information.



Click "..." on the right side and check "共有の有効化". (Not required if not shared)



After completing the settings, click "✓" at the right end.



7.1.5. How To Use With The App

The Nextcloud environment has a desktop client app, which can also be used via the client app. The installation work of the client application involves a reboot. When installing, save the file you are editing, etc. and execute it.

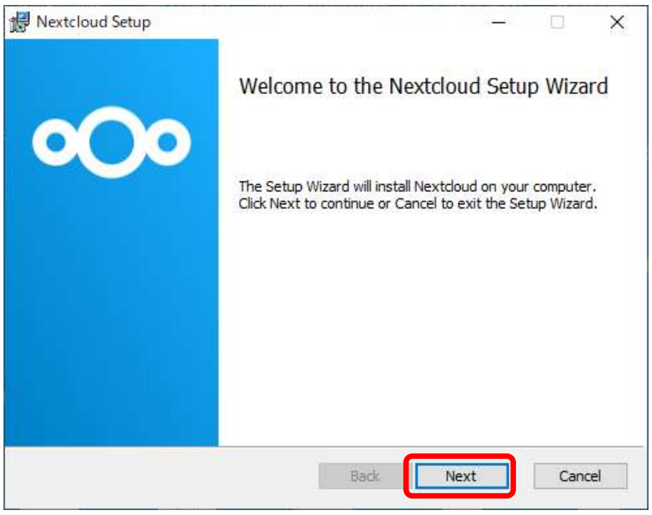
(1)Install

Download the installer from the following URL

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

Run the downloaded installer.

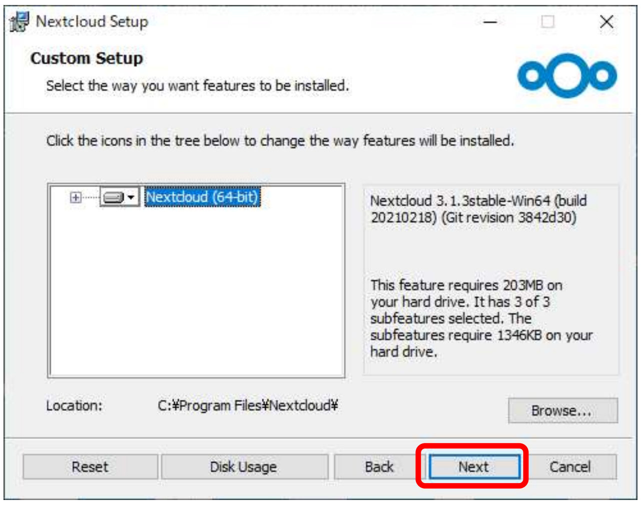
The installation will start, so click "Next".


Nextcloud Setup

Welcome to the Nextcloud Setup Wizard

The Setup Wizard will install Nextcloud on your computer.
Click Next to continue or Cancel to exit the Setup Wizard.

There is no need to change the settings, click "Next".


Nextcloud Setup

Custom Setup

Select the way you want features to be installed.

Click the icons in the tree below to change the way features will be installed.

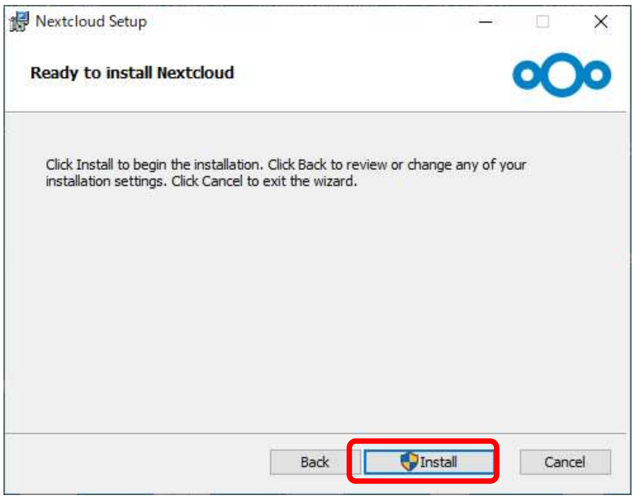
Nextcloud (64-bit)

Nextcloud 3.1.3stable-Win64 (build 20210218) (Git revision 3842d30)

This feature requires 203MB on your hard drive. It has 3 of 3 subfeatures selected. The subfeatures require 1346KB on your hard drive.

Location: C:¥Program Files¥Nextcloud¥

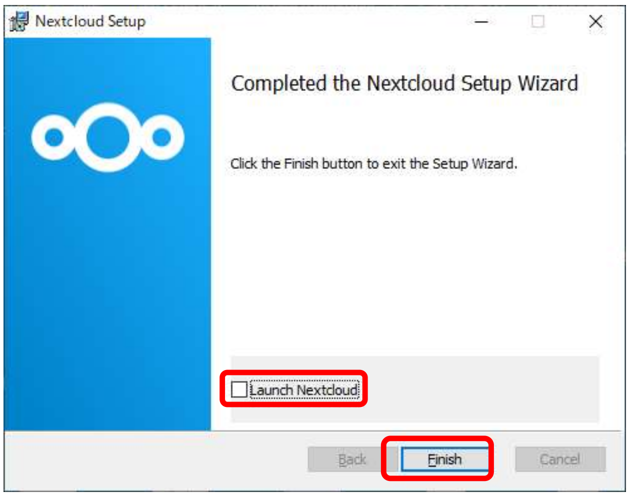
Click Install to perform the installation


Nextcloud Setup

Ready to install Nextcloud

Click Install to begin the installation. Click Back to review or change any of your installation settings. Click Cancel to exit the wizard.

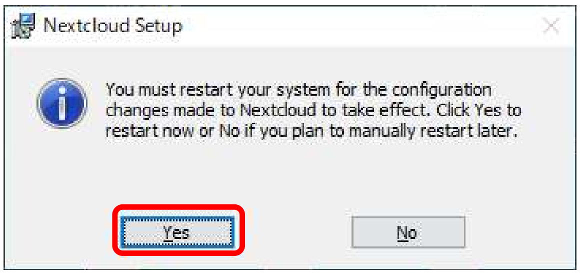
After the installation is complete, uncheck "Launch Nextcloud" and click "Finish".


Completed the Nextcloud Setup Wizard

Click the Finish button to exit the Setup Wizard.

A dialog prompting you to restart will be displayed. Click "Yes" to restart your computer.

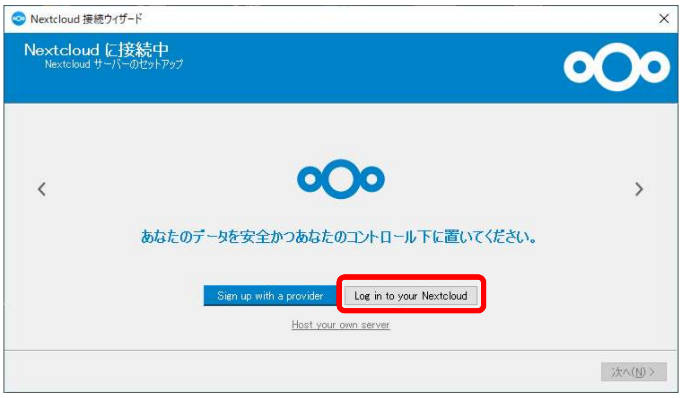
※ Please save the file you are editing in advance.


You must restart your system for the configuration changes made to Nextcloud to take effect. Click Yes to restart now or No if you plan to manually restart later.

(2)Settings

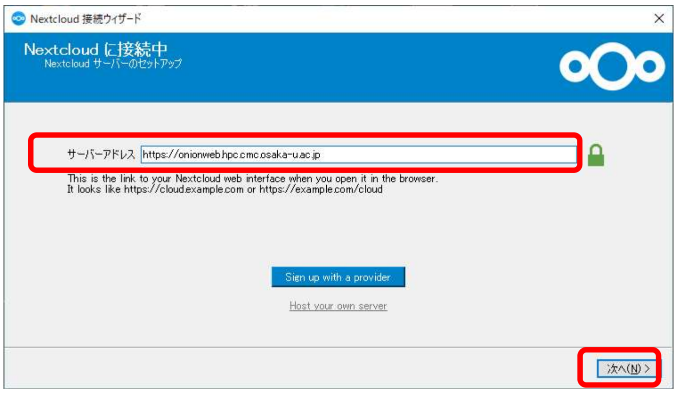
Make settings to connect to the Nextcloud server.

After restarting your computer, Nextcloud will start automatically and the following window will be appered. Click "Log in to your Nextcloud".


あなたのデータを安全かつあなたのコントロール下に置いてください。

Enter the following URL in Server Address and click 「次へ」.

URL : https://onionweb.hpc.cmc.osaka-u.ac.jp


Nextcloud 接続ウィザード

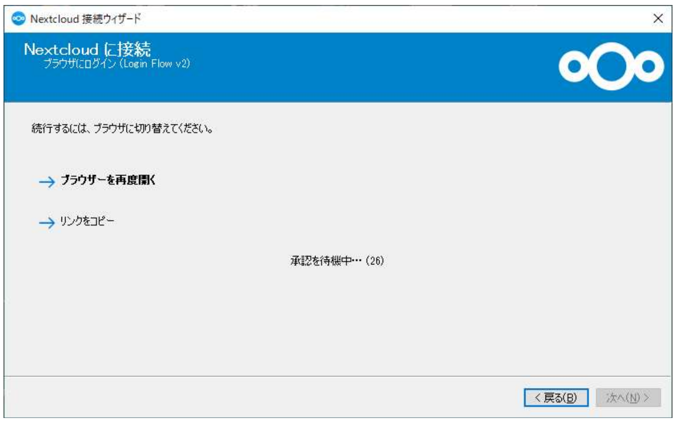
Nextcloud に接続中

Nextcloud サーバーのセットアップ

サーバーアドレス https://onionweb.hpc.cmc.osaka-u.ac.jp

This is the link to your Nextcloud web interface when you open it in the browser.
It looks like https://cloud.example.com or https://example.com/cloud

The following screen will be displayed, and the screen for authorizing authentication with a web browser will start automatically.


Nextcloud 接続ウィザード

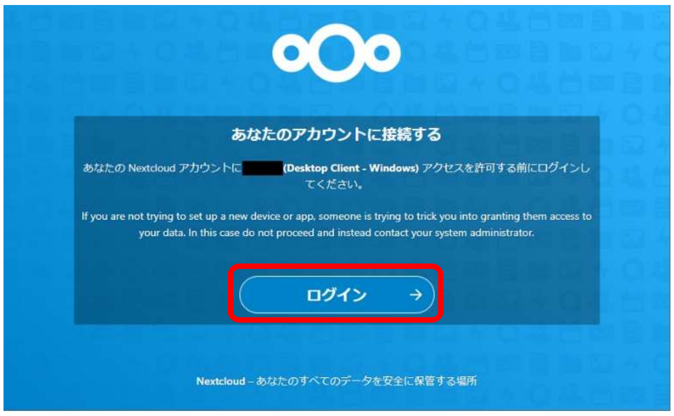
Nextcloud に接続

ブラウザにログイン (Login Flow v2)

続行するには、ブラウザに切り替えてください。

承認を待機中… (26)

The following screen will be displayed on your web browser. Click "Login".

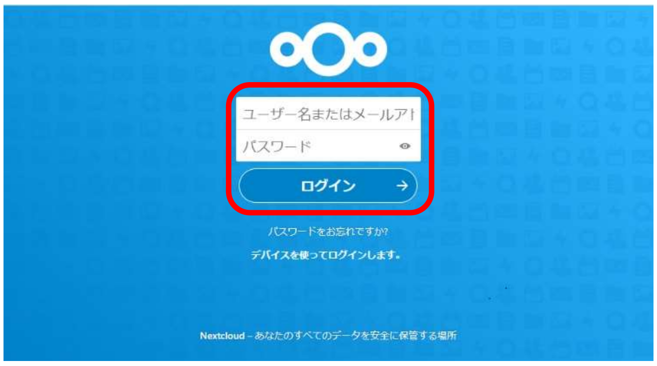

あなたのアカウントに接続する

あなたの Nextcloud アカウントに[redacted] **(Desktop Client - Windows)** アクセスを許可する前にログインしてください。

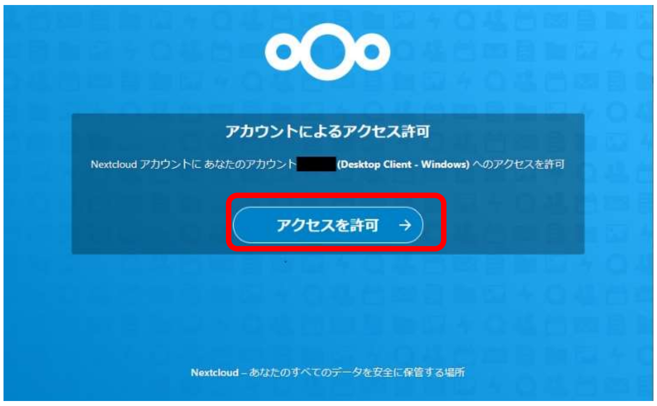
If you are not trying to set up a new device or app, someone is trying to trick you into granting them access to your data. In this case do not proceed and instead contact your system administrator.

Nextcloud – あなたのすべてのデータを安全に保管する場所

The following login screen will be displayed. Enter your user name and password and click "ログイン".

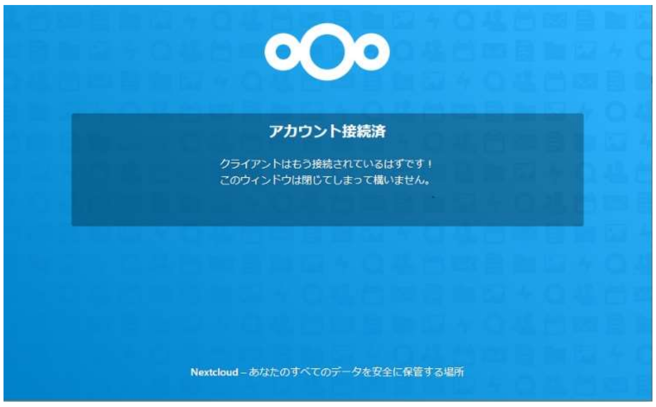

ユーザー名またはメールアド
パスワード
ログイン
パスワードをお忘れですか?
デバイスを使ってログインします。
Nextcloud – あなたのすべてのデータを安全に保管する場所

The following screen will be displayed. Click "アクセスを許可".


アカウントによるアクセス許可

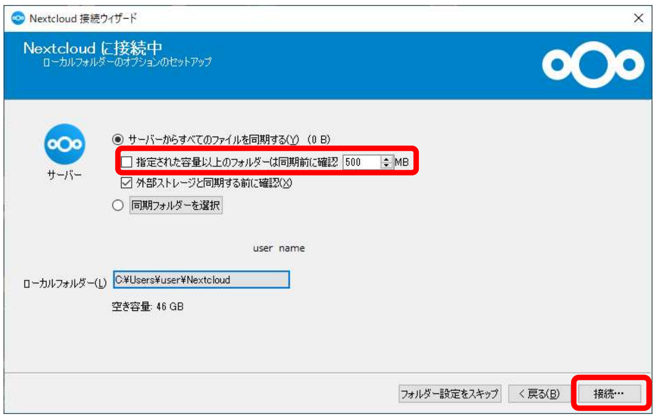
Nextcloud アカウントに あなたのアカウント [redacted] (Desktop Client - Windows) へのアクセスを許可

The following screen will be displayed. Follow the instructions on the screen to close the web browser.


アカウント接続済

クライアントはもう接続されているはずです！
このウィンドウは閉じてしまって構いません。

After returning to the application, the following screen will be displayed. Uncheck "指定された容量以上のフォルダーは同期前に確認" and click "接続 ...".


Nextcloud 接続ウィザード

Nextcloud に接続中

ローカルフォルダーのオプションのセットアップ

[✓ サーバーからすべてのファイルを同期する(Y) (0 B)]

☐ 指定された容量以上のフォルダーは同期前に確認 500 MB

☑ 外部ストレージと同期する前に確認(X)

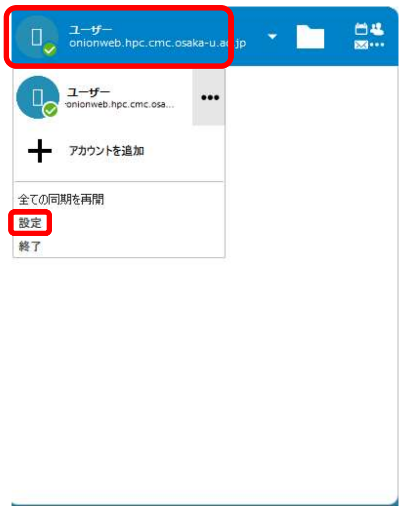
○ 同期フォルダーを選択

user name

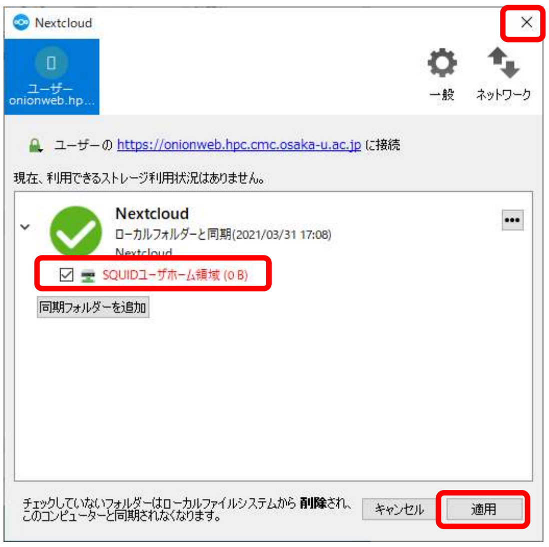
ローカルフォルダー(L) C:¥Users¥user¥Nextcloud

空き容量: 46 GB

The Nextcloud settings screen will be displayed at the bottom right of the screen. Click the user name, and then click "設定" in the displayed menu.



The following screen will be displayed. Check "SQUID User Home Area" and click "Apply". After waiting for a while, synchronization will be completed, so close it with "x".


Nextcloud

ユーザーの https://onionweb.hpc.cmc.osaka-u.ac.jp に接続

現在、利用できるストレージ利用状況はありません。

Nextcloud

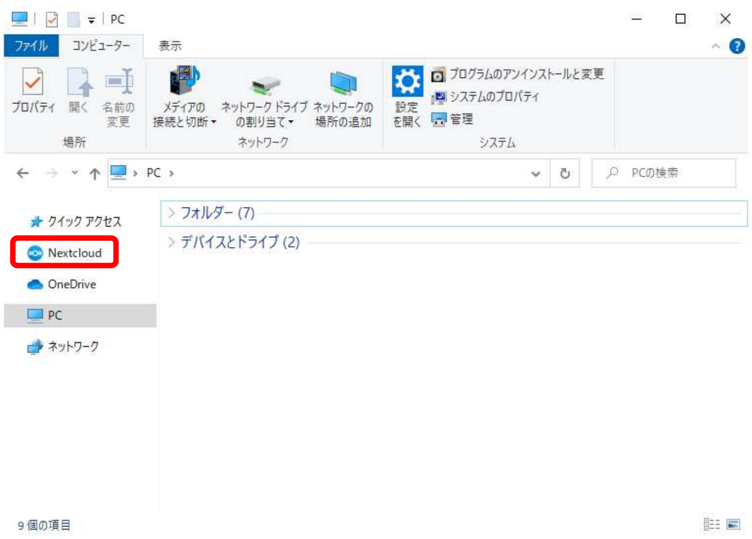
ローカルフォルダーと同期(2021/03/31 17:08)

Nextcloud

☑ SQUIDユーザホーム領域 (0 B)

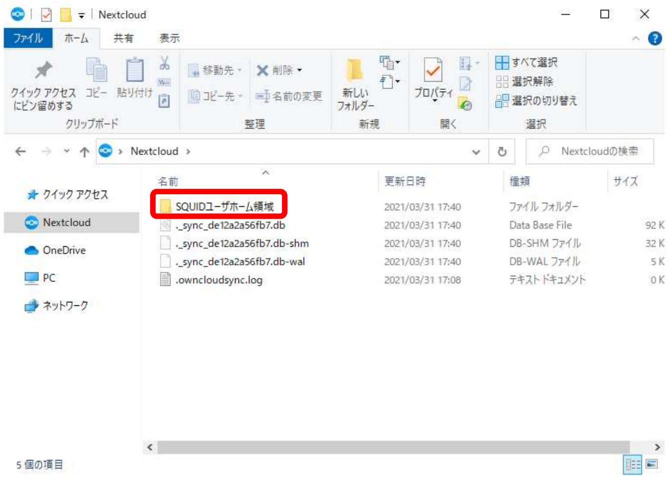
チェックしていないフォルダーはローカルファイルシステムから **削除**され、このコンピューターと同期されなくなります。

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



Do not move or delete files whose file names start with ".".

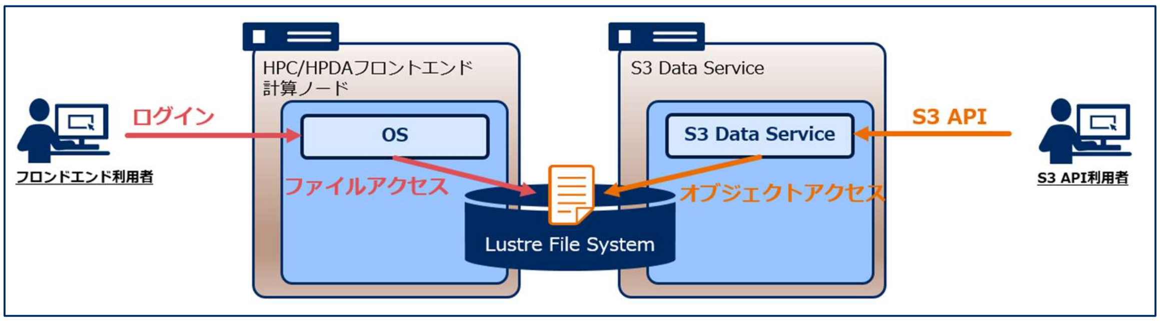
Open the "SQUID ユーザホーム領域". You can read and write files.



名前	更新日時	種類	サイズ
SQUIDユーザホーム領域	2021/03/31 17:40	ファイル フォルダー	
._sync_de12a2a56fb7.db	2021/03/31 17:40	Data Base File	92 K
._sync_de12a2a56fb7.db-shm	2021/03/31 17:40	DB-SHM ファイル	32 K
._sync_de12a2a56fb7.db-wal	2021/03/31 17:40	DB-WAL ファイル	5 K
.owncloudsync.log	2021/03/31 17:08	テキスト ドキュメント	0 K

7.2. File Transfer with S3 API

We have prepared an S3 Data Service environment as a method of transferring files to SQUID using the S3 API. S3 Data Service provides an environment for UNIX-like file access and object access from the S3 API for the same files on SQUID.



This makes it possible to call the S3 API from an application and perform data communication directly with SQUID. Objects (files) that communicate data are saved in the following paths in SQUID, and can be accessed directly from the front-end nodes and compute nodes.

```
/sqfs/s3/(UUID)/(Bucket name)/(Object name)
```

7.2.1. Generate Access Key

In order to use the S3 API, you need to generate an access key. Access keys are operated with the s3dskey command in the front-end environment.

To generate an access key, execute the following command.

```
$ s3dskey create --group=(group name)
```

--group : Specifies the group name to include in the extended space usage.

If the execution is successful, the following output will be returned and an access key will be generated.

```
+-----------+------------------------------------------+
| accesskey | AKIA7X1KXXXYYYZZZ000                     |
| enabled   | True                                     |
| fspaths   | None                                     |
| fsuid     | 60101:10                                 |
| secretkey | hagwyutos8TThj3SOv9ahPJMF8TTK2dYFcV9Qv7T |
| tag       | (Serial number): (User name): (Group name)|
| uuid      | ea60f00a60hufaweofapo12813nfawe9506e1216 |
+-----------+------------------------------------------+
```

accesskey	: Key to present when accessing S3 API
secretkey	: Password information associated with the access key
tag	: Tag information including the user name and group name that owns the access key
uuid	: Unique identifier for the key. Used when accessing as a file system

※ As of May 2021, the access key issuance limit is one per user and group.

7.2.2. Access Key Operation

The access key operation is a subcommand option of the s3dskey command, and the list of subcommands is as follows.

Command	Description
s3dskey list	List of created access keys
s3dskey create	Create a new access key
s3dskey disable	Disable access key

s3dskey enable	Enable access key
s3dskey reset	Secret key initialization

For details on the command, refer to the command help. To refer to the command help, execute the following command.

```
$ s3dskey --help
```

7.2.3. Create Bucket

In order to upload / download objects (files) through the S3 API, you need to create an object management unit called a bucket.

Use the s3dsbucket command to create a bucket.

```
$ s3dsbucket create --key=(access key) --bucket=(Bucket name)
```

--key	: Specify the pre-generated access key
--bucket	: Specify the bucket name to create

※ Specify the bucket name according to the naming convention below.

・ more than 3 characters and less than 255 characters

・ Only uppercase letters (A-Z), lowercase letters (a-z), numbers (0-9), periods (.), Hyphens (-), and underscores (_) can be specified.

・ The bucket name is unique and cannot be duplicated with a bucket name created by yourself or another user.

※ Bucket creation is also possible with the PUT BUCKET method of the S3 API. However, it is recommended to create it with the s3dsbucket command. Because the contents operated directly on the front-end cannot be synchronized with the S3 API.

7.2.4. Bucket Operation

Bucket operations are performed with subcommands of the s3dsbucket command. The list of subcommands is as follows.

コマンド	説明
s3dsbucket list	List of created buckets
s3dsbucket create	Create a new bucket
s3dsbucket sync	Synchronize files in the file system with objects in the S3 API

s3dsbucket delete	Delete bucket ＊ Possible only when there are no objects in the bucket

For details on the command, refer to the command help. To refer to the command help, execute the following command.

```
$ s3dsbucket --help
```

7.2.5. API Access

Object (file) operations are performed from various applications that support the S3 API. The following is an example of using s3curl, which is a command wrapper for curl, to use the S3 API. For information about s3curl, please refer to the following URL.

https://github.com/EMCECS/s3curl

(1) Definition Of Access Information

Create a file called .s3curl directly under your home directory.

```
 ~/.s3curl
```
```
  %awsSecretAccessKeys = (
            username => {
                 id => 'AKIA7X1KXXXYYYZZZ000',
                 key => 'hagwyutos8TThj3SOv9ahPJMF8TTK2dYFcV9Qv7T',
            },
);
push(@endpoints, 'squidgw.hpc.cmc.osaka-u.ac.jp')
```

username	: Access information alias. Any name can be specified
id	: access key
key	: Secret key

(2) Create a Object

Upload the object into the bucket created in "7.2.3 Create Bucket". Execute a command similar to the following to upload the object.

```
$ LANG=C
$ s3curl.pl --id username --put (Local file name) -- -s ¥
     -v https://squidgw.hpc.cmc.osaka-u.ac.jp/(Bucket name)/(Object name)
```

The uploaded object can be referenced on the SQUID front end with the following path.

```
/sqfs/s3/(UUID)/(Bucket name)/(Object name)
```

※ The UUID is an identifier associated with the access key displayed in "7.2.1Generate Access Key"

7.2.6. Synchronize Bucket

S3 Data Service allows UNIX-like file access and object access from the S3 API to the same object (file).Object meta-data (existence of files and directories) on the S3 API is managed separately from the file system, so if you operate the file directly from the front-end node etc., you need to synchronize the changes to the S3 Data Service.

※ To synchronize buckets, you need to create a bucket with the s3dsbucket command.

・ **How to sync from the command line**

Use the sync subcommand of the s3dsbucket command. Buckets can be synchronized by executing the following on the front end.

```
$ s3dsbucket sync --key=(access key) --bucket=(Bucket name)
```

--key	: Specify the pre-generated access key
--bucket	: Specify the bucket name to synchronize

・ **How to sync with S3 API**

An extension of the S3 Data Service, the PUT BucketSync method, is available. For s3curl, execute as follows.

```
$ LANG=C
$ s3curl.pl --id username --put=emptyPayload -- -s ¥
      -v https://squidgw.hpc.cmc.osaka-u.ac.jp/(Bucket name)?sync ¥
      "-H x-ddn-bucket-sync-ops:WRITE,UPDATE,DELETE"
```

For the endpoint name, specify "(Bucket URL) +"?Syn" " and optionally the extended headers (x-ddn-bucket-sync-ops) for the S3 Data Service. The argument can be specified by combining WRITE/UPDATE/DELETE. The contents of each are as follows.

- **WRITE**: Scan the file system and associate a new object with a file that is not associated with an object
- **UPDATE**: Scans the object in the bucket and updates the object's metadata if it has changed
- **DELETE**: Scan the objects in the bucket and delete the objects if the associated file is not found

If there is no extension header specified in synchronization from the command line or synchronization from S3 API, WRITE, UPDATE, DELETE will be specified.

7.3. File Transfer from OCTOPUS

The file system is different for SQUID and OCTOPUS.

This section describes the file transfer procedure between SQUID and OCTOPUS for those who also use OCTOPUS.

This is the file transfer method when using the same account with SQUID and OCTOPUS. The OCTOPUS file system can be accessed from SQUID's front-end server.

File system	File path on SQUID	File path when accessing from OCTOPUS
SQUID high speed area	/sqfs/ssd	/sqfs/ssd
SQUID home area	/sqfs/home	/sqfs/home
SQUID Extended area	/sqfs/work	/sqfs/work

File system	File path on OCTOPUS	File path when accessing from SQUID
OCTOPUS home area	/octfs/home	/octfs/home
OCTOPUS Extended area	/octfs/work	/octfs/work

* The OCTOPUS file system cannot be accessed from the SQUID calculation node, so use it only for file transfer.

The following is how to transfer files. The user number is "a61234"

- OCTOPUS → SQUID (when working on the SQUID front end)

  When transferring the file sample.c in 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)

  When transferring the file sample.c in the abc directory under the SQUID home directory to the OCTOPUS home directory

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

7.4. File Transfer with CIFS

Those who use a campus network of Osaka University(ODINS) can access the SQUID directories by using CIFS.

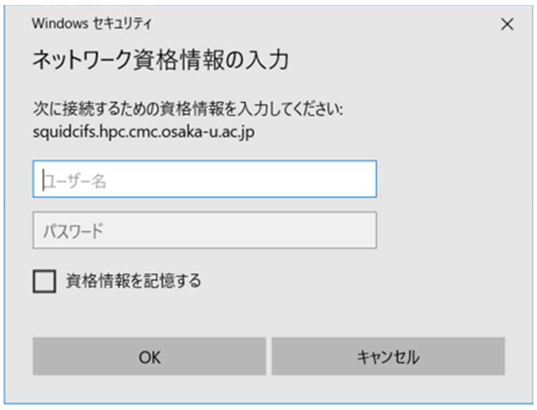
- /sqfs/ssd
- /sqfs/work
- /sqfs/home

You can access SQUID directories from your Windows terminal by using Shared directory path.

Shared directory path	SQUID directory
¥¥squidcifs.hpc.cmc.osaka-u.ac.jp¥ssd	/sqfs/ssd
¥¥squidcifs.hpc.cmc.osaka-u.ac.jp¥work	/sqfs/work
¥¥squidcifs.hpc.cmc.osaka-u.ac.jp¥home	/sqfs/home

For example, test user u6b398 (group G15284) access work directory.

Typing Shared directory path 「¥¥squidcifs.hpc.cmc.osaka-u.ac.jp¥work」 in Explorer, the authentication pop-up will be displayed.

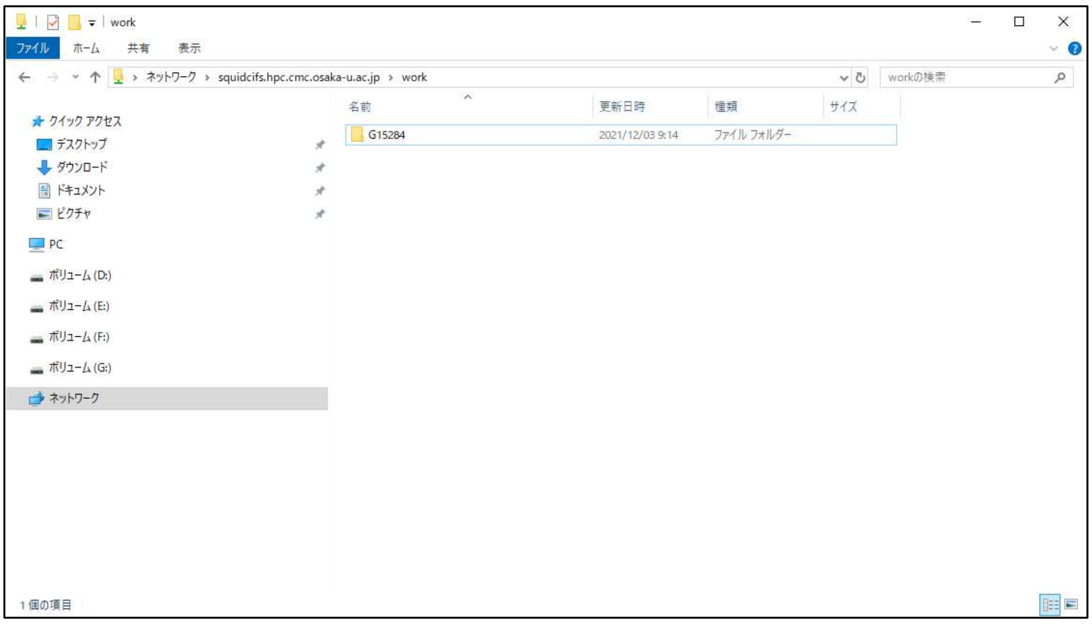

Windows セキュリティ

ネットワーク資格情報の入力

次に接続するための資格情報を入力してください:
squidcifs.hpc.cmc.osaka-u.ac.jp

If you type the user name and password that registered by applying for SQUID usage, you can access work directory.

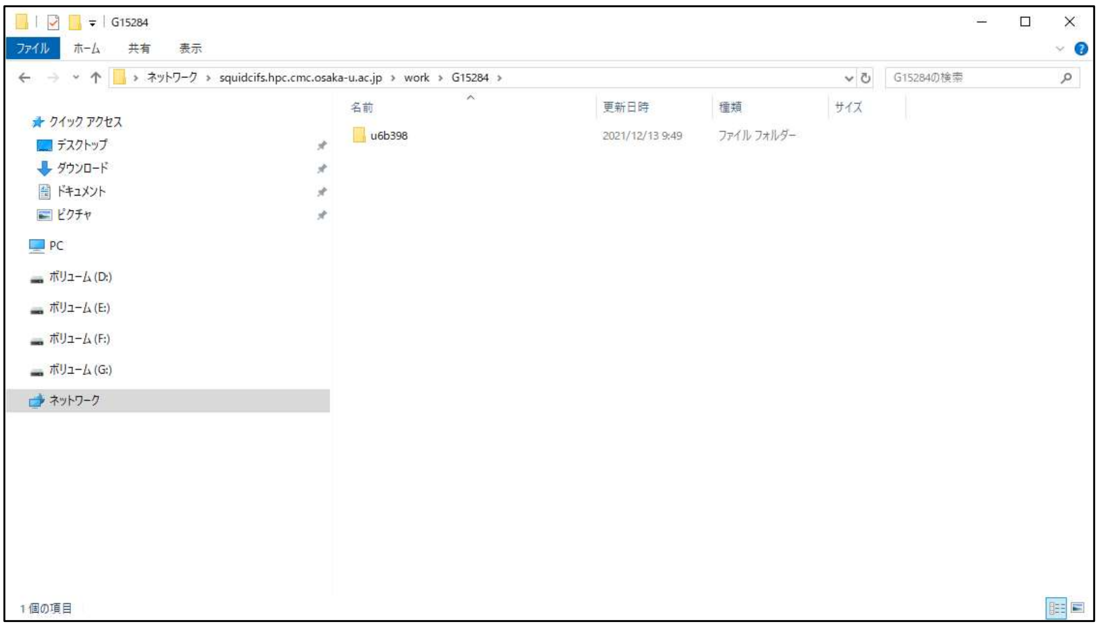
When you access ssd and work directory, first of all, you can see the group directory.



名前	更新日時	種類	サイズ
G15284	2021/12/03 9:14	ファイル フォルダー	

If you access group directory, you can see user’s directory.

Please use user’s directory to transfer your files.



名前	更新日時	種類	サイズ
u6b398	2021/12/13 9:49	ファイル フォルダー	

If you access home directory, firest of all, you can see user’s directory. (Not group directory)

8. Other Services

8.1. Archive Storage

You can use Cloudian's HyperStore, which is an object storage product, as archive storage for data aggregation.

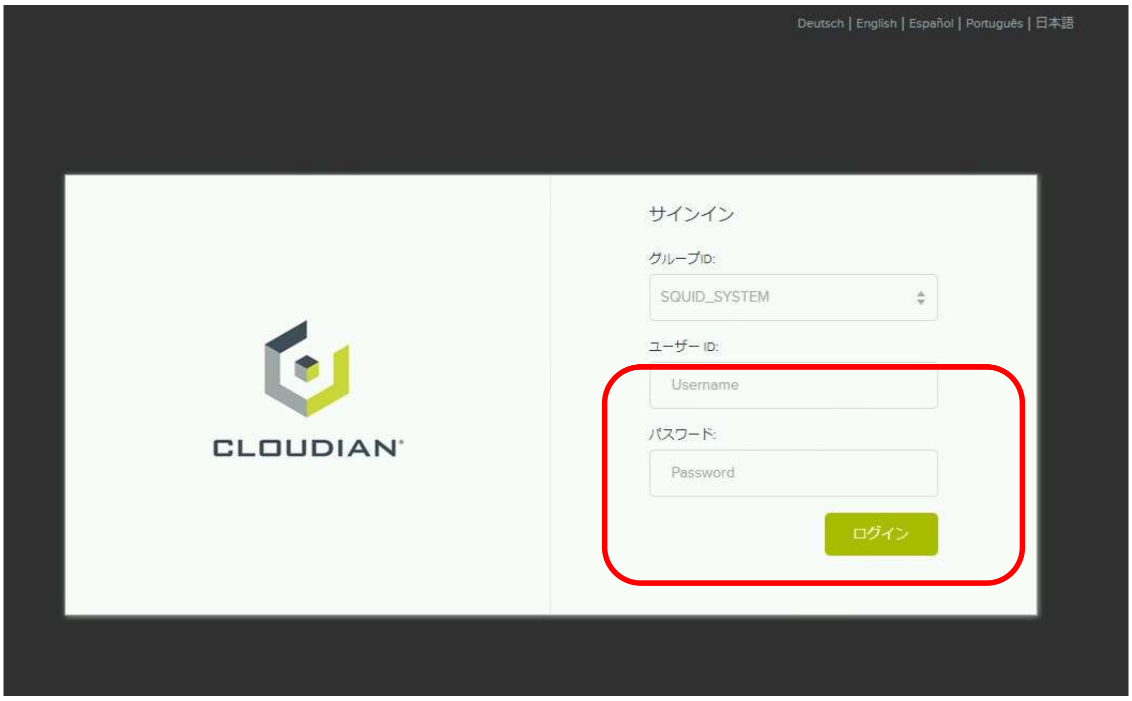
8.1.1. How to Use Cloudian

(1) Login

Open a web browser and enter the following URL.

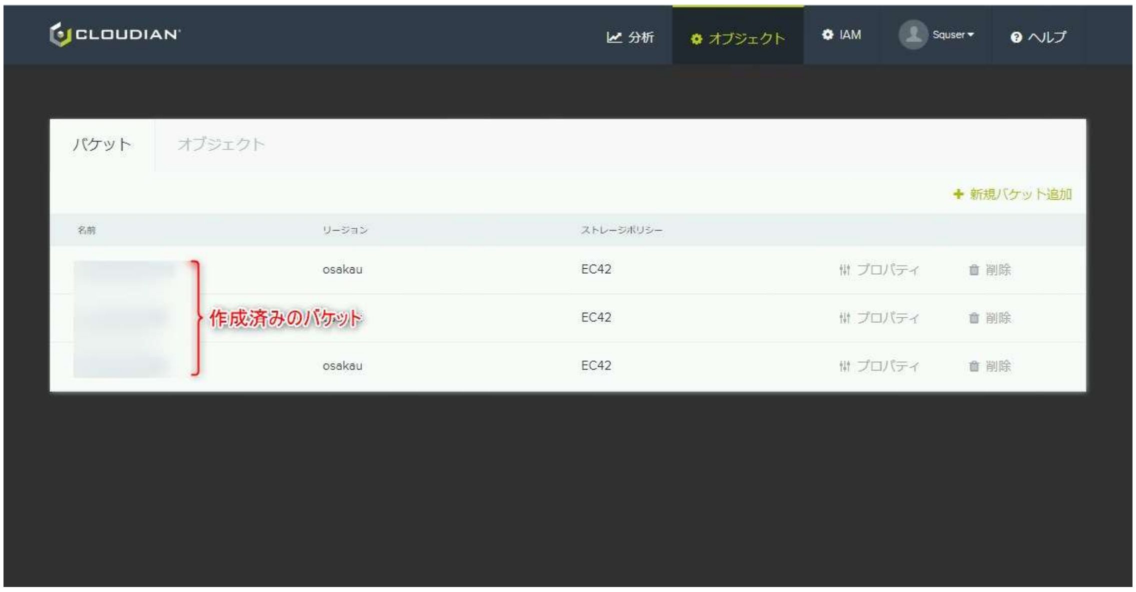
https://onionportal.hpc.cmc.osaka-u.ac.jp:8443/

The following login screen will be displayed. Enter your user name and password and click "Login".



(2) Screen After Login

After logging in, the following screen will be displayed.

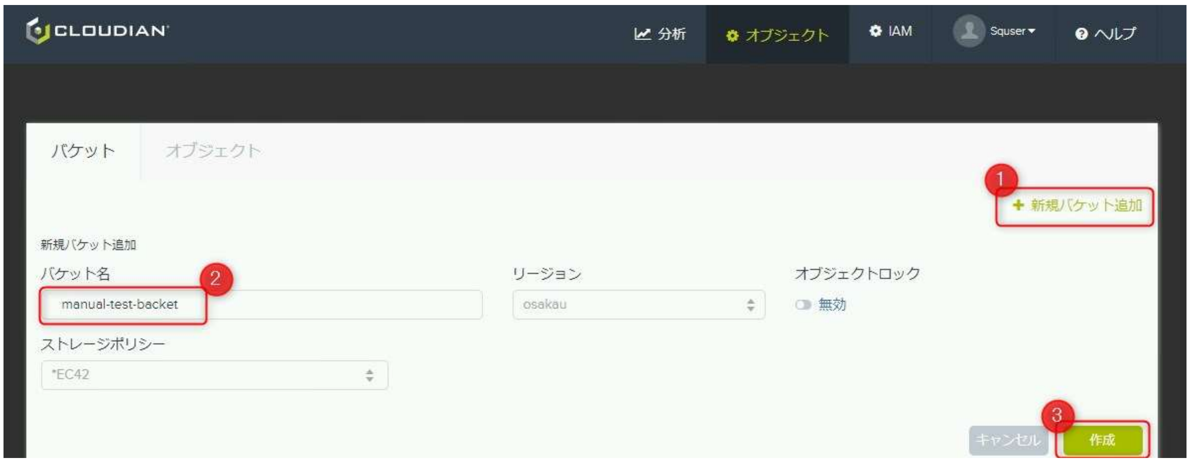


名前	リージョン	ストレージポリシー		
[redacted]	osakau	EC42	プロパティ	削除
[redacted]		EC42	プロパティ	削除
[redacted]	osakau	EC42	プロパティ	削除

After logging in, you will see the bucket you have created.

(3) Bucket Creation

From the login screen, click Add New Bucket (①) and enter an arbitrary bucket name. (②) After entering, click Create (③). Please note that entering an underscore for the bucket name will limit its functionality.



新規バケット追加

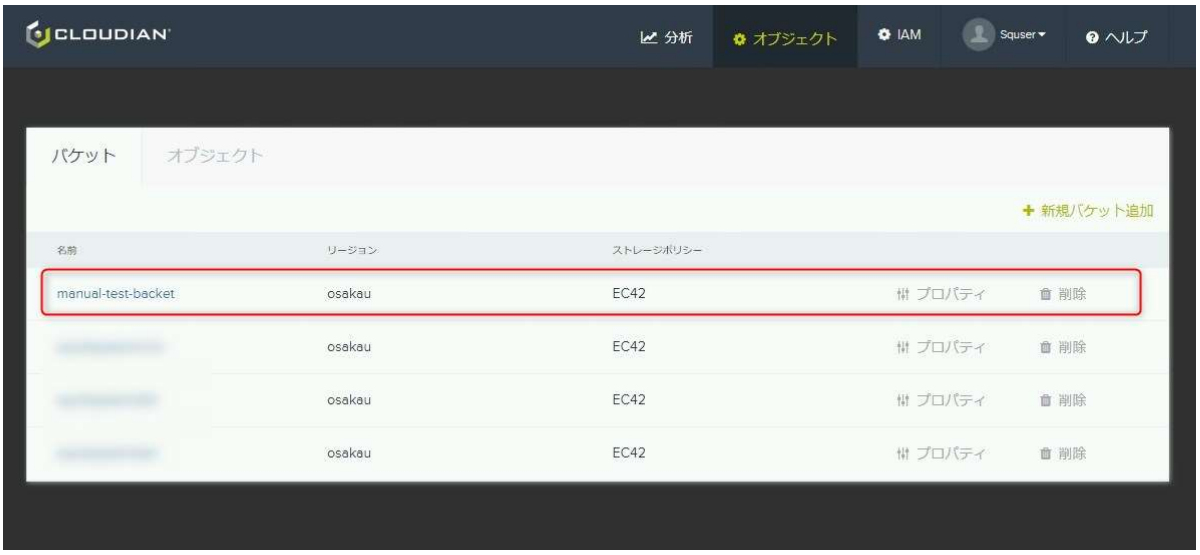
バケット名: manual-test-backet

リージョン: osakau

オブジェクトロック: 無効

ストレージポリシー: *EC42

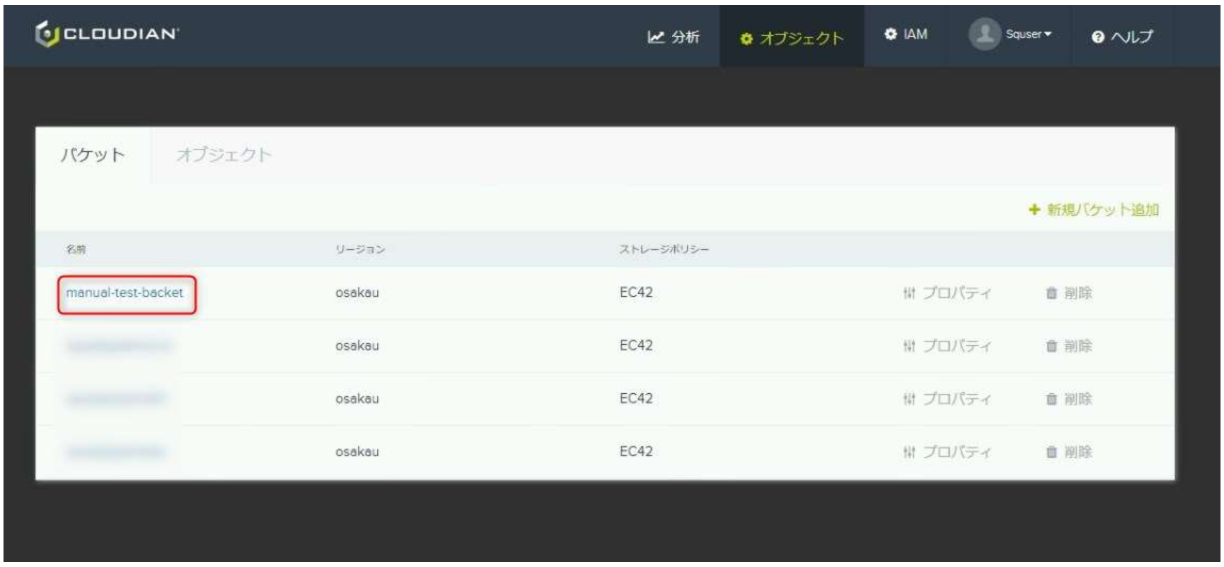
Created as follows.



名前	リージョン	ストレージポリシー		
manual-test-backet	osakau	EC42	プロパティ	削除
	osakau	EC42	プロパティ	削除
	osakau	EC42	プロパティ	削除
	osakau	EC42	プロパティ	削除

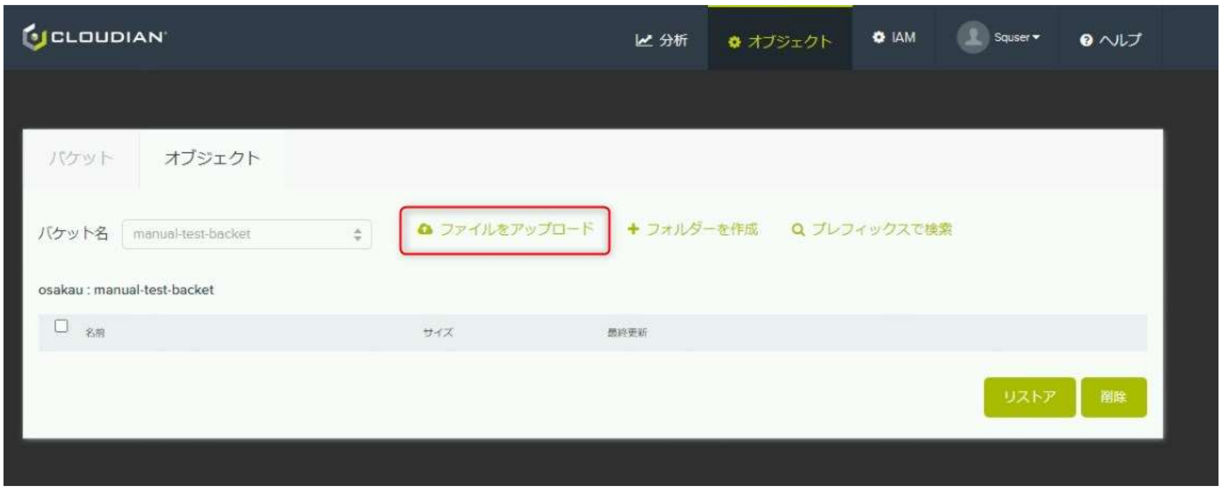
(4) File Upload

After logging in, click the created bucket to open it.



名前	リージョン	ストレージポリシー		
manual-test-backet	osakau	EC42	プロパティ	削除
	osakau	EC42	プロパティ	削除
	osakau	EC42	プロパティ	削除
	osakau	EC42	プロパティ	削除

Click Upload File.

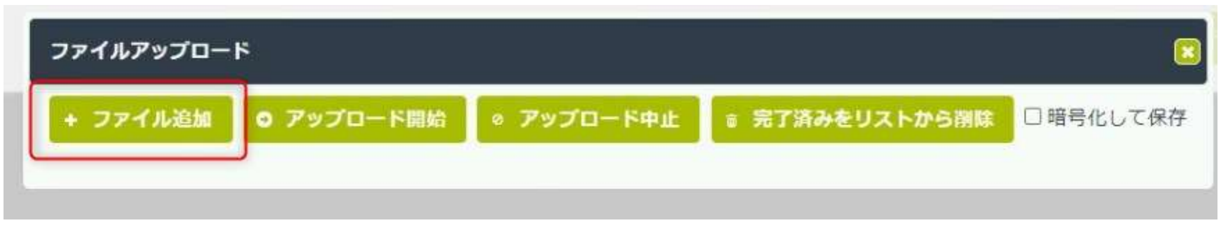


バケット名 manual-test-backet

osakau : manual-test-backet

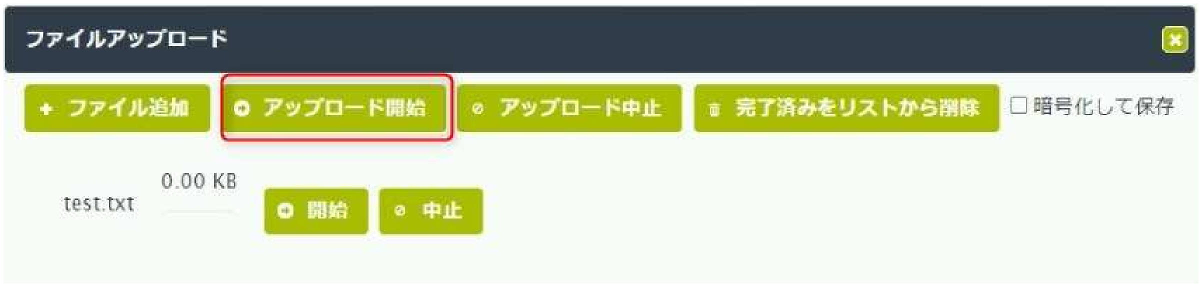
名前	サイズ	最終更新

Then click Add File.

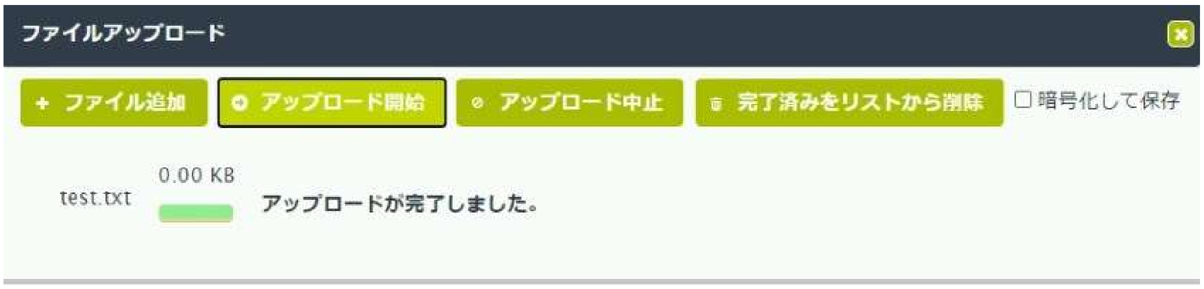


Select the file you want to upload. (Test.txt is selected in this document)

Click Start Upload.



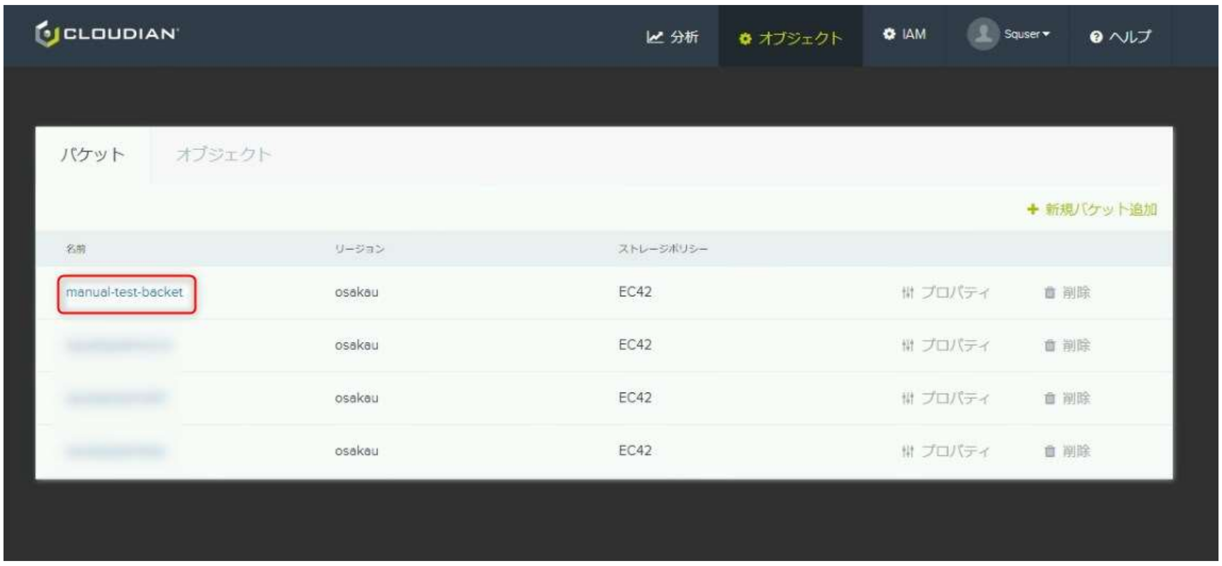
The upload will be completed as below



アップロードが完了しました。

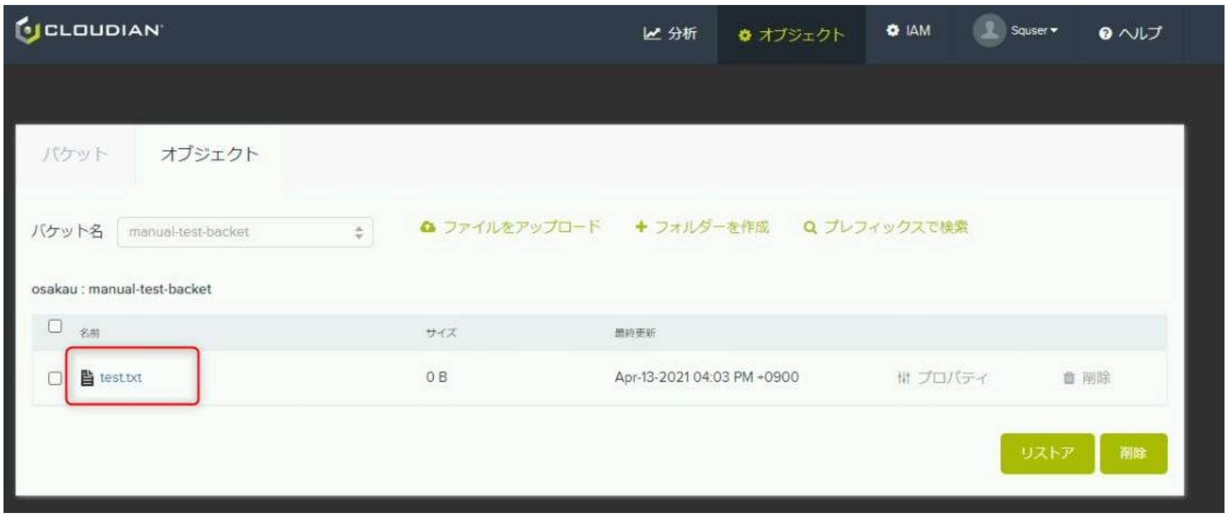
(5) File Download

After logging in, click the created bucket to open it.



名前	リージョン	ストレージポリシー		
manual-test-backet	osakau	EC42	プロパティ	削除
[redacted]	osakau	EC42	プロパティ	削除
[redacted]	osakau	EC42	プロパティ	削除
[redacted]	osakau	EC42	プロパティ	削除

Click the file name to download.



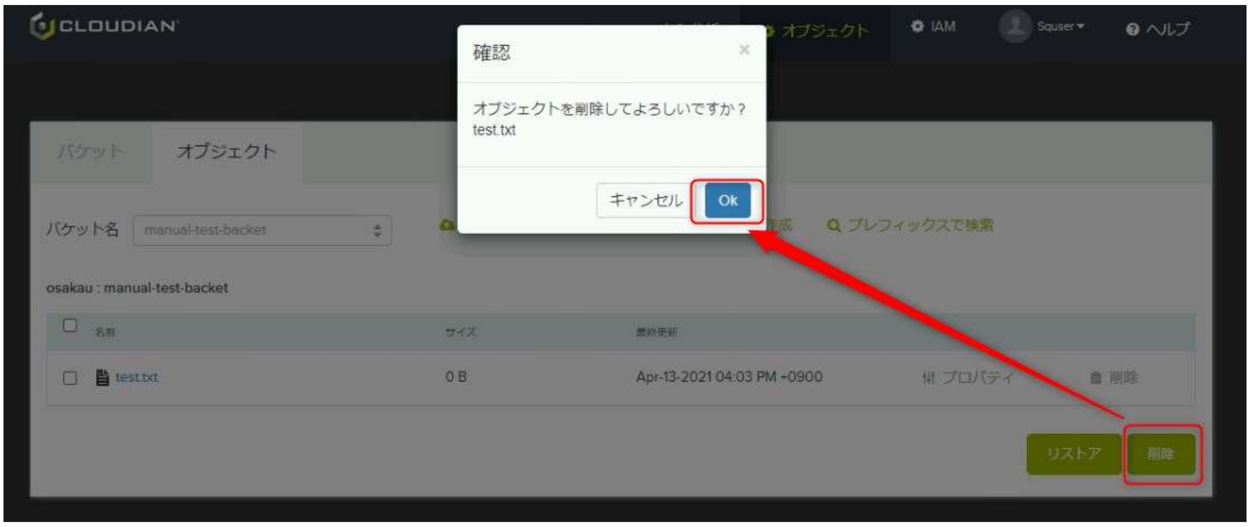
バケット名 manual-test-backet

osakau : manual-test-backet

名前	サイズ	最終更新		
test.txt	0 B	Apr-13-2021 04:03 PM +0900	プロパティ	削除

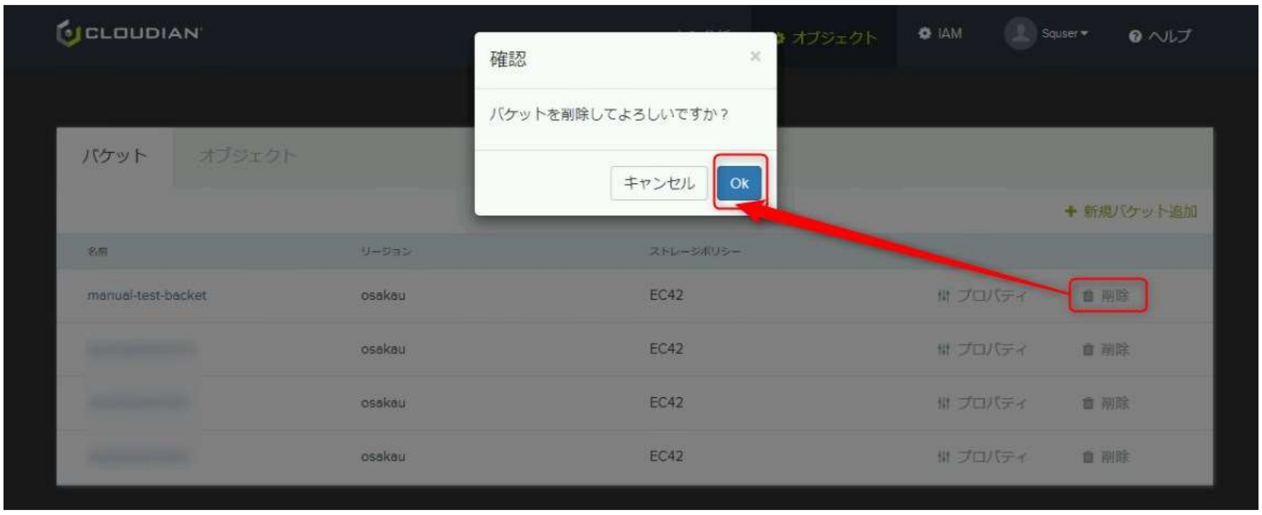
(6) Delete File

Select the file you want to delete, click Delete, confirm, and click OK.



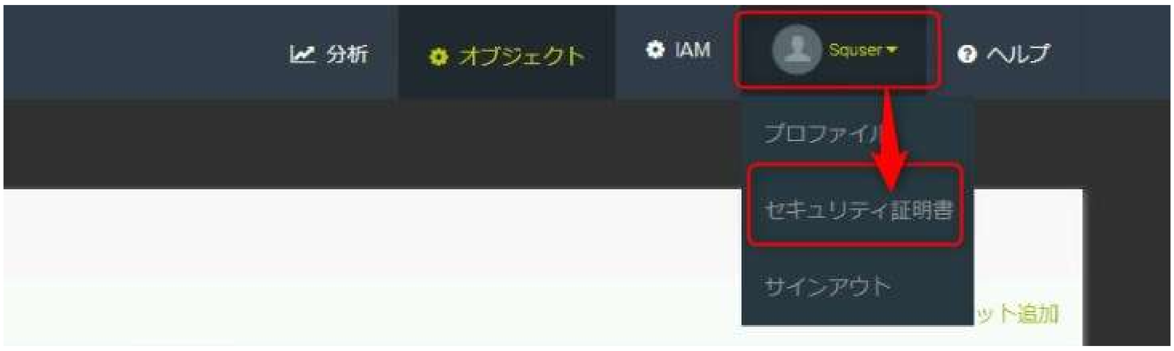
(7) Delete Bucket

Select the bucket you want to delete, click Delete, confirm and click OK.

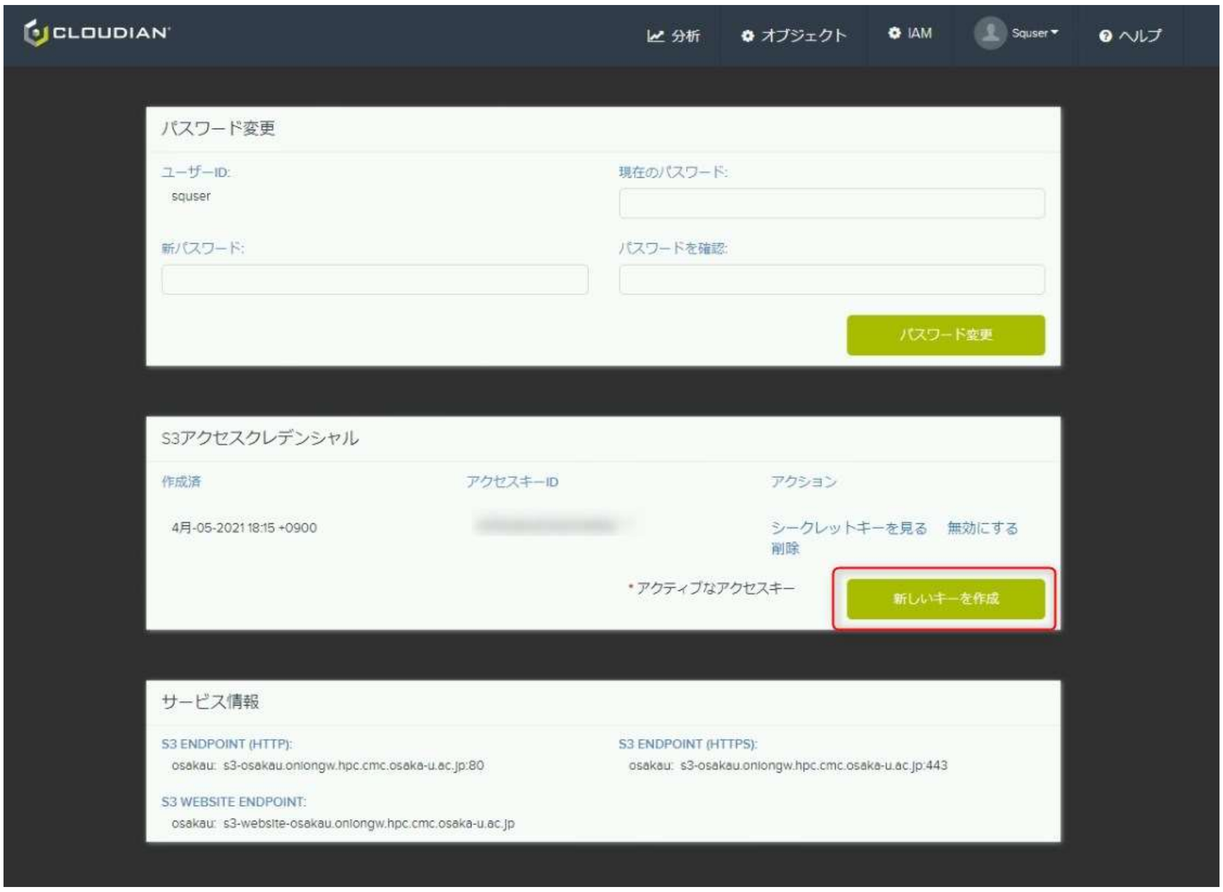


(8) Issuance Of Access Key And Secret Key

After logging in, click your username and then the security certificate.



Click Create New Key from S3 Access Credentials



パスワード変更

ユーザーID: squser

現在のパスワード:

新パスワード:

パスワードを確認:

S3アクセスクレデンシャル

作成済	アクセスキーID	アクション
4月-05-2021 18:15 +0900	[redacted]	シークレットキーを見る 無効にする 削除

• アクティブなアクセスキー

サービス情報

S3 ENDPOINT (HTTP):
osakau: s3-osakau.onlongw.hpc.cmc.osaka-u.ac.jp:80

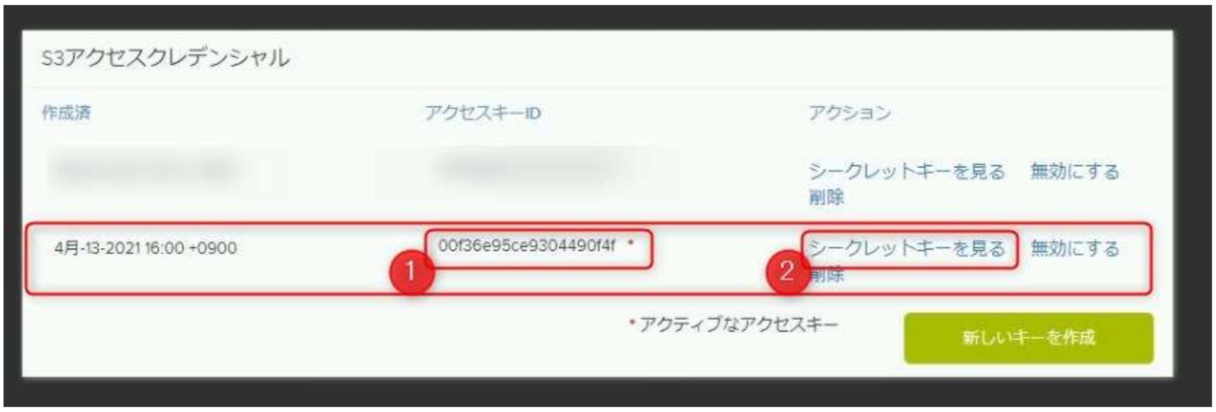
S3 ENDPOINT (HTTPS):
osakau: s3-osakau.onlongw.hpc.cmc.osaka-u.ac.jp:443

S3 WEBSITE ENDPOINT:
osakau: s3-website-osakau.onlongw.hpc.cmc.osaka-u.ac.jp

An access key (①) and a secret key will be issued.

The secret key is displayed by clicking on his secret key (②).

Keep the access key and secret key handy for future access.



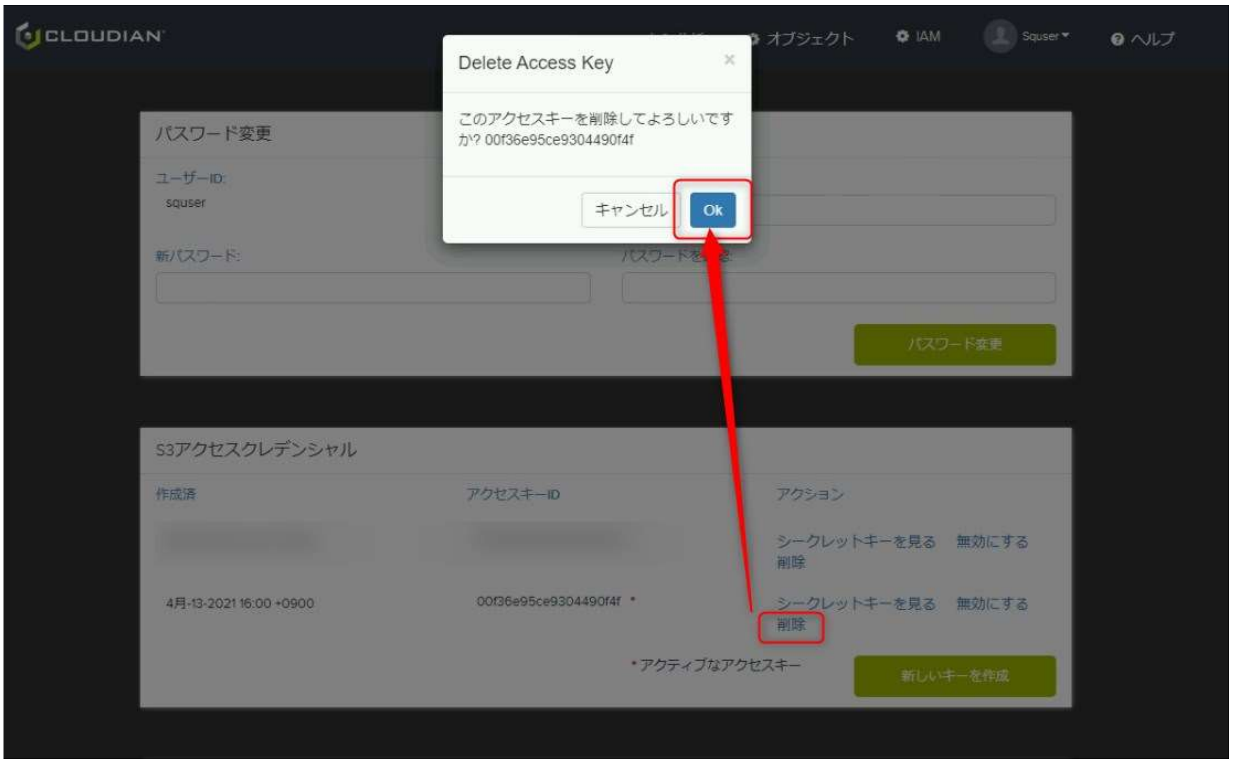
S3アクセスクレデンシャル

作成済	アクセスキーID	アクション
[redacted]	[redacted]	シークレットキーを見る 無効にする 削除
4月-13-2021 16:00 +0900	00f36e95ce9304490f4f *	シークレットキーを見る 無効にする 削除

• アクティブなアクセスキー

(9) Deletion Of Access Key And Secret Key

Click "削除 for the key you want to delete from the S3 access credentials, confirm it, and then click "OK".。


Delete Access Key

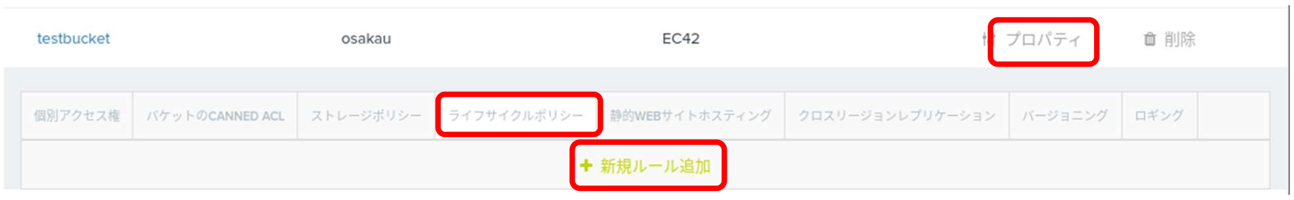
このアクセスキーを削除してよろしいですか? 00f36e95ce9304490f4f

(10) Auto-Tiering

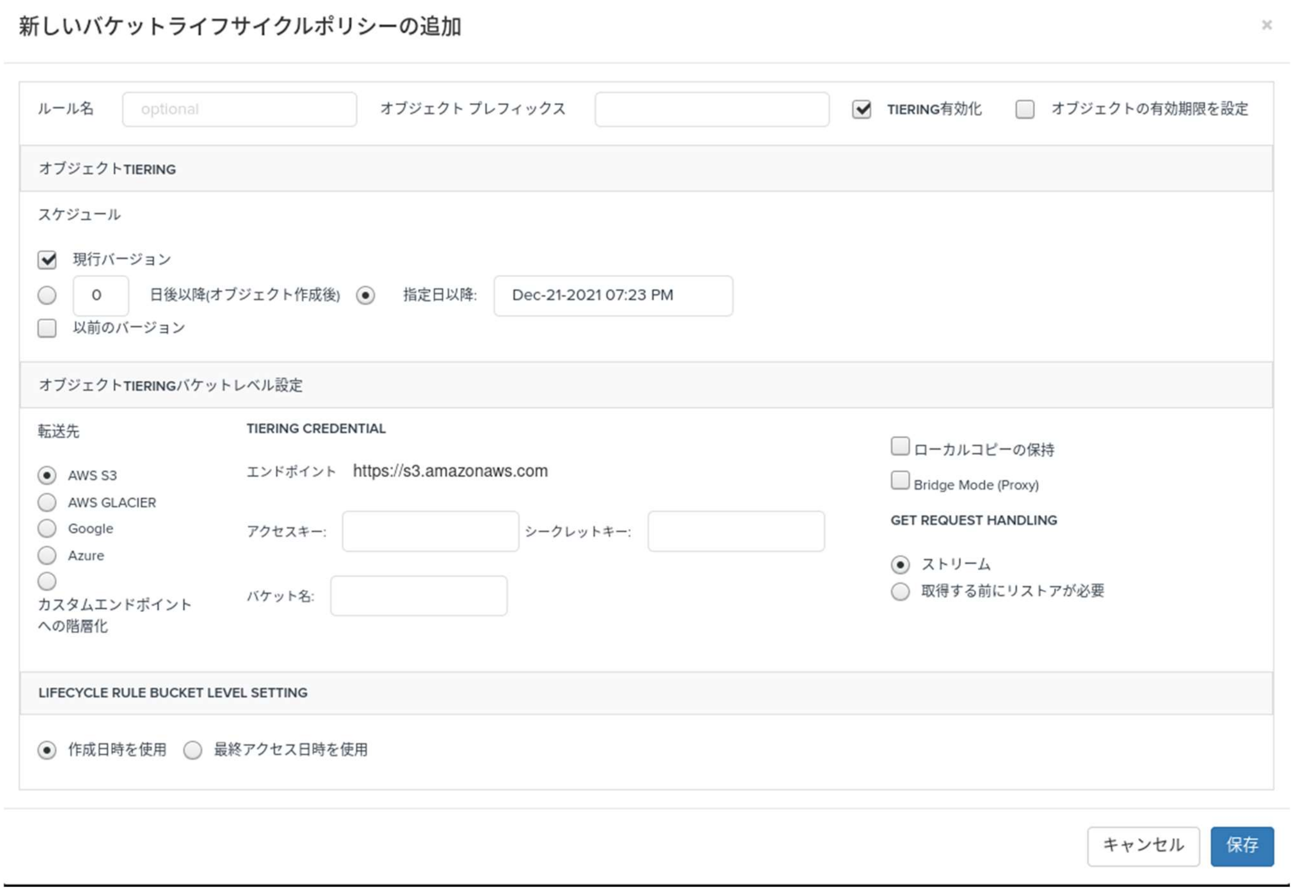
It is possible to automatically move objects that have passed the life cycle policy to other buckets (including buckets from vendors other than Cloudian), by using the Auto-Tiering function.

Log in from Cloudian web browser and click “property” (プロパティ) of the bucket which you want to set Auto-Tiering.

Click Property’s tab “Lifecycle Policy” (ライフサイクルポリシー) and click “Add New Rule” (新しいルール追加).



Click “Enable TIERING” (TIERING 有効化) on “the Add New Bucket Lifecycle Policy” screen and please set items. If there is a problem with the settings, an error message will be displayed, when you click the “Save” (保存) button.


新しいバケットライフサイクルポリシーの追加

ルール名	optional	オブジェクト プレフィックス	
✓ TIERING有効化	☐ オブジェクトの有効期限を設定		

オブジェクトTIERING

スケジュール

✓ 現行バージョン

○ 0 日後以降(オブジェクト作成後) ● 指定日以降: Dec-21-2021 07:23 PM

☐ 以前のバージョン

オブジェクトTIERINGバケットレベル設定

転送先: ● AWS S3 ○ AWS GLACIER ○ Google ○ Azure ○ カスタムエンドポイントへの階層化

TIERING CREDENTIAL

エンドポイント https://s3.amazonaws.com

アクセスキー:

シークレットキー:

バケット名:

☐ ローカルコピーの保持

☐ Bridge Mode (Proxy)

GET REQUEST HANDLING

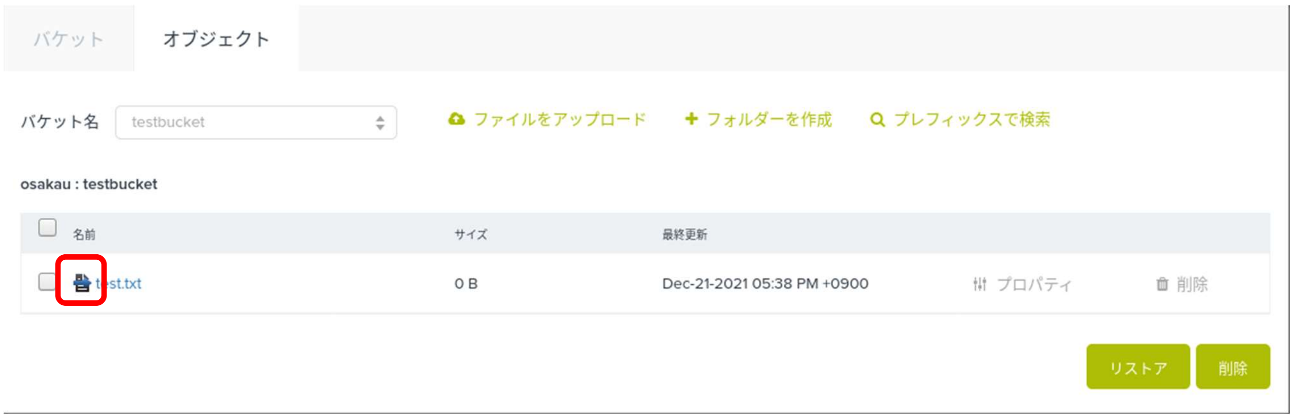
● ストリーム ○ 取得する前にリストアが必要

LIFECYCLE RULE BUCKET LEVEL SETTING

● 作成日時を使用 ○ 最終アクセス日時を使用

Auto-Tiering runs at 23:00 UTC (8:00 am Japan time) by default.

The object moved by Auto-Tiering changes file image.



バケット名 testbucket

osakau : testbucket

名前	サイズ	最終更新		
test.txt	0 B	Dec-21-2021 05:38 PM +0900	プロパティ	削除

8.2. Portal System for statistical information

The portal system for statistical information provides the following services.

・Computer usage status Web

・Node operation status Web

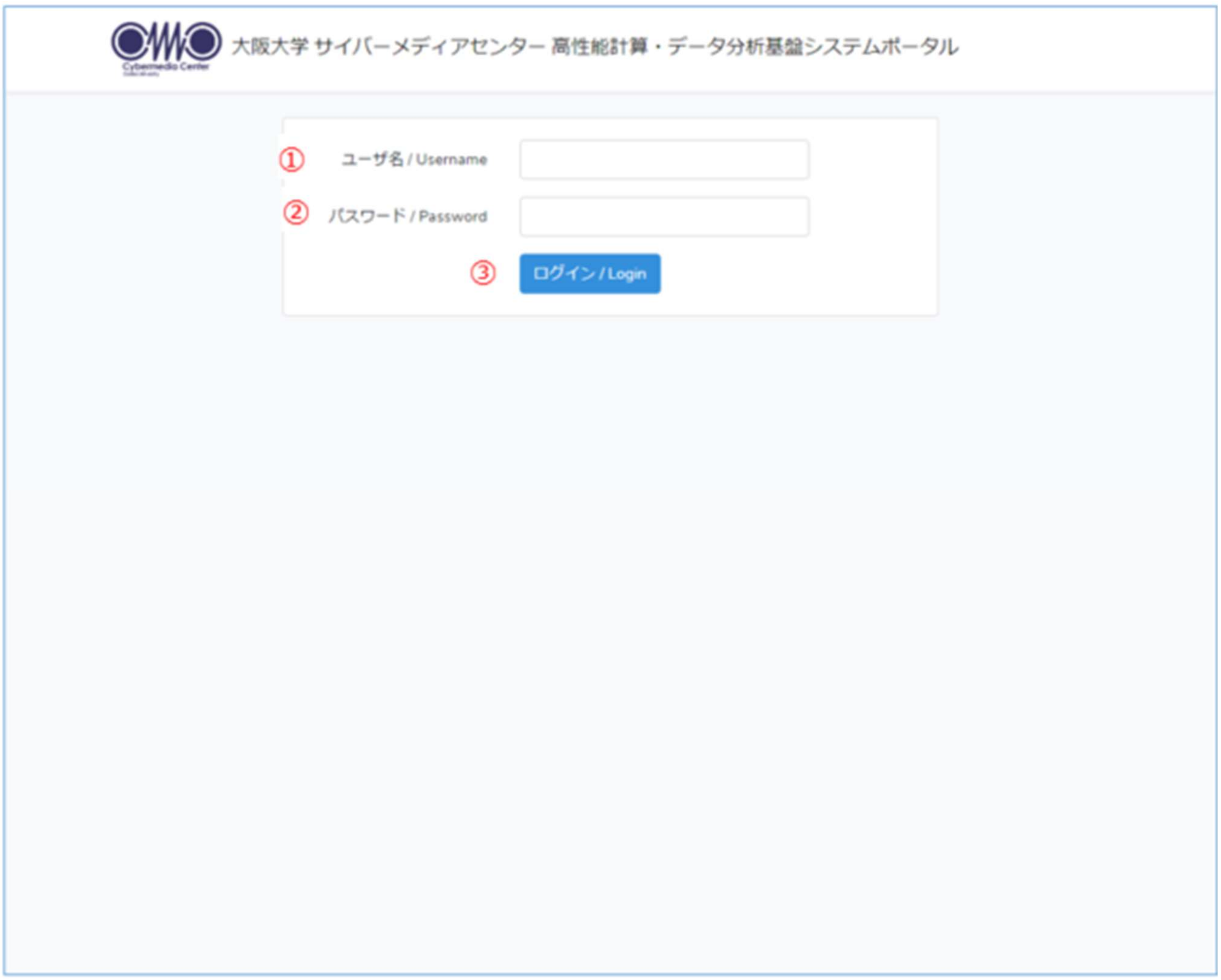
8.2.1. How to log in portal system

Login to the portal system also requires advance preparation for 2-factor authentication. Please prepare the 2-factor authentication application in advance by referring to "2.1.1 Installation the 2-factor authentication application".

- Authentication codes are independent for front-end login and portal system login, so please set 2-factor authentication when logging in for the first time.

(1) Login Page

When you access "https://squidportal.hpc.cmc.osaka-u.ac.jp/portal/login" , the portal system login page will be displayed.


大阪大学 サイバーメディアセンター 高性能計算・データ分析基盤システムポータル
①
ユーザ名 / Username
②
パスワード / Password
③
ログイン / Login

◆ Instractions

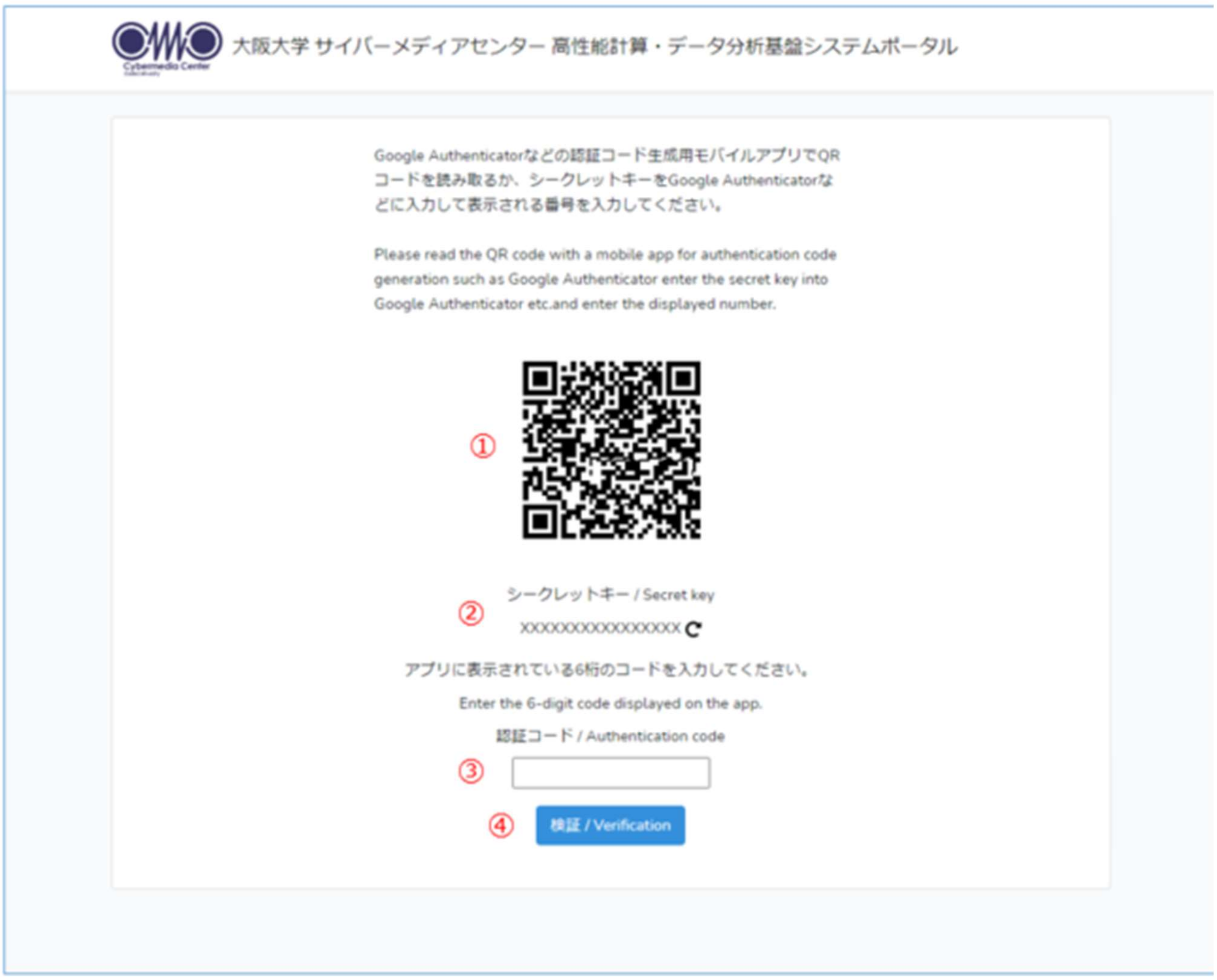
① input "user number" to [ユーザ名] form

② input "Login password" to [パスワード] form

③ Cick [ログイン]

(2) 2-factor authentication setting page

If the login user's 2-factor authentication setting has not been set, the screen will change to the 2-cactor authentication setting page after login authentication.


大阪大学 サイバーメディアセンター 高性能計算・データ分析基盤システムポータル

Google Authenticatorなどの認証コード生成用モバイルアプリでQRコードを読み取るか、シークレットキーをGoogle Authenticatorなどに入力して表示される番号を入力してください。

Please read the QR code with a mobile app for authentication code generation such as Google Authenticator enter the secret key into Google Authenticator etc.and enter the displayed number.

①

② シークレットキー / Secret key

XXXXXXXXXXXXXXXX

アプリに表示されている6桁のコードを入力してください。

Enter the 6-digit code displayed on the app.

認証コード / Authentication code

③

④ 検証 / Verification

◆ Instructions

① Read the QR code displayed on the 2-factor authentication application.

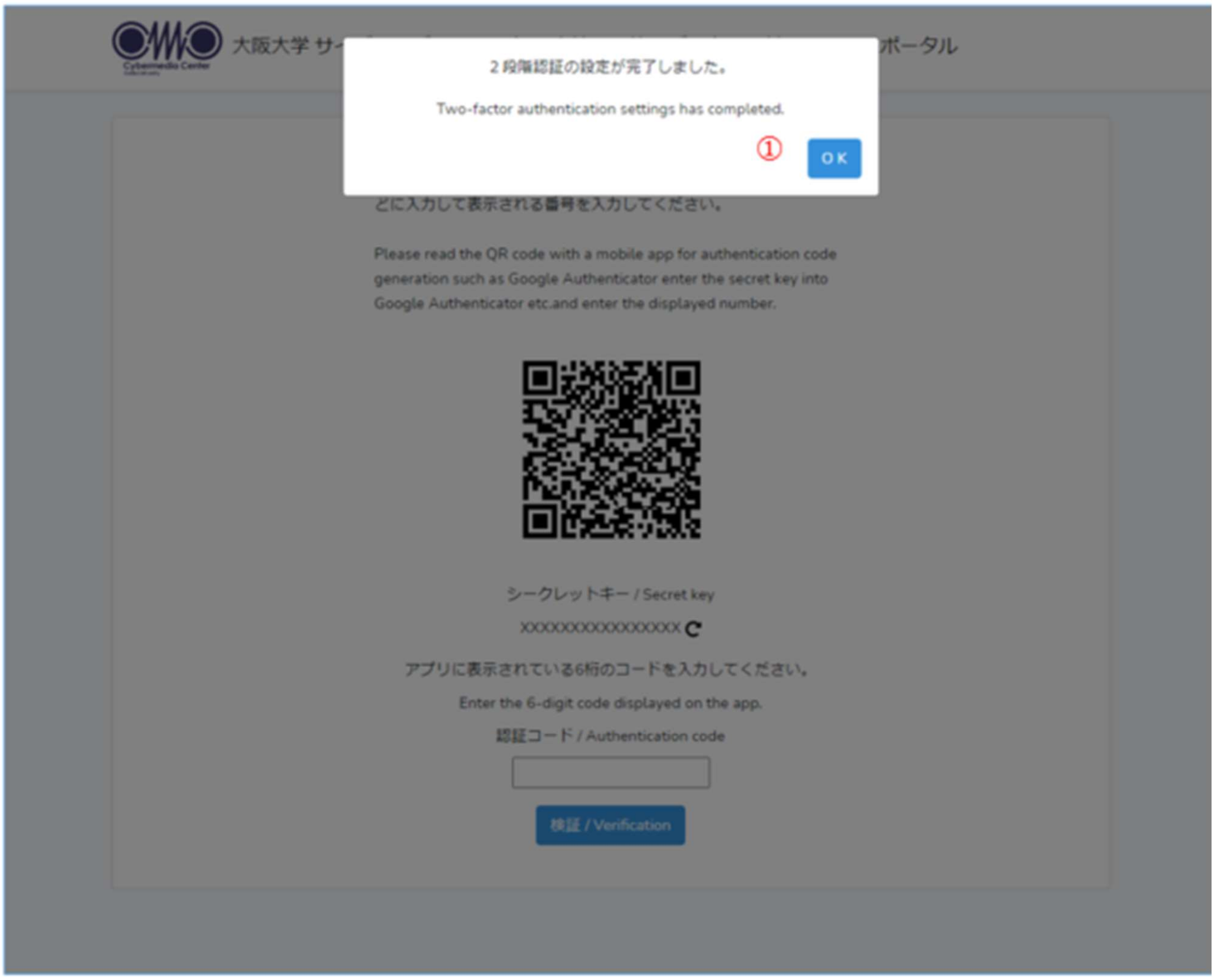
② If you cannot read the QR code, enter the secret key into the 2-factor authentication application.

③ Input the authentication code displayed on the 2-factor authentication application in the [認証コード] form

④ Click [Verification]. Performs authentication code verification processing and 2-factor authentication setting processing. When the process is complete, the completion dialog is displayed.

◆ Dialog

When the 2-factor authentication setting is completed, a dialog indicating the completion of processing is displayed.


大阪大学 サ ポータル

2 段階認証の設定が完了しました。

Two-factor authentication settings has completed.

①

どに入力して表示される番号を入力してください。

Please read the QR code with a mobile app for authentication code generation such as Google Authenticator enter the secret key into Google Authenticator etc.and enter the displayed number.

シークレットキー / Secret key

XXXXXXXXXXXXXXXX

アプリに表示されている6桁のコードを入力してください。

Enter the 6-digit code displayed on the app.

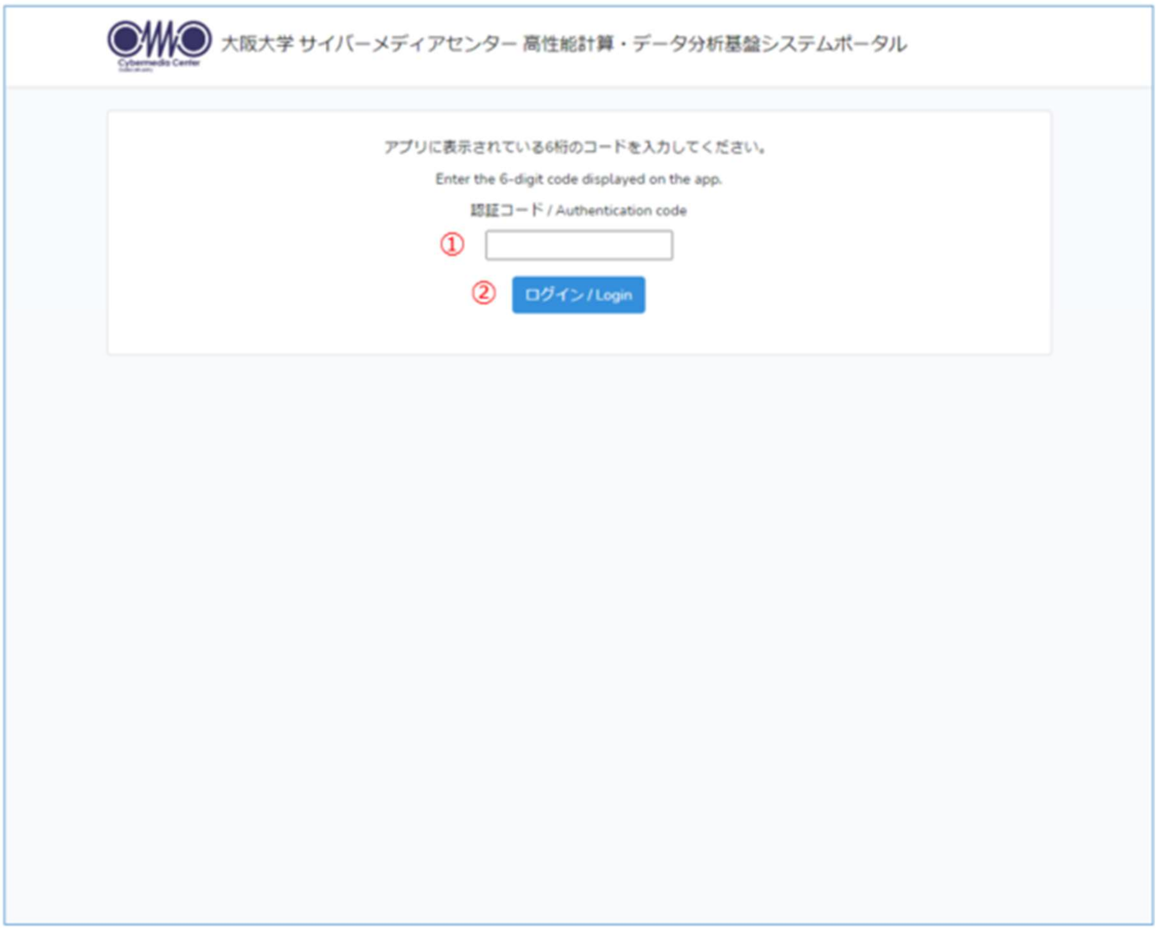
認証コード / Authentication code

① OK

Click [OK] to move portal home page.

(3) 2-factor authentication page

If the 2-factor authentication setting has already been set, the screen will change to the 2-step authentication screen after login authentication.



大阪大学 サイバーメディアセンター 高性能計算・データ分析基盤システムポータル

アプリに表示されている6桁のコードを入力してください。

Enter the 6-digit code displayed on the app.

認証コード / Authentication code

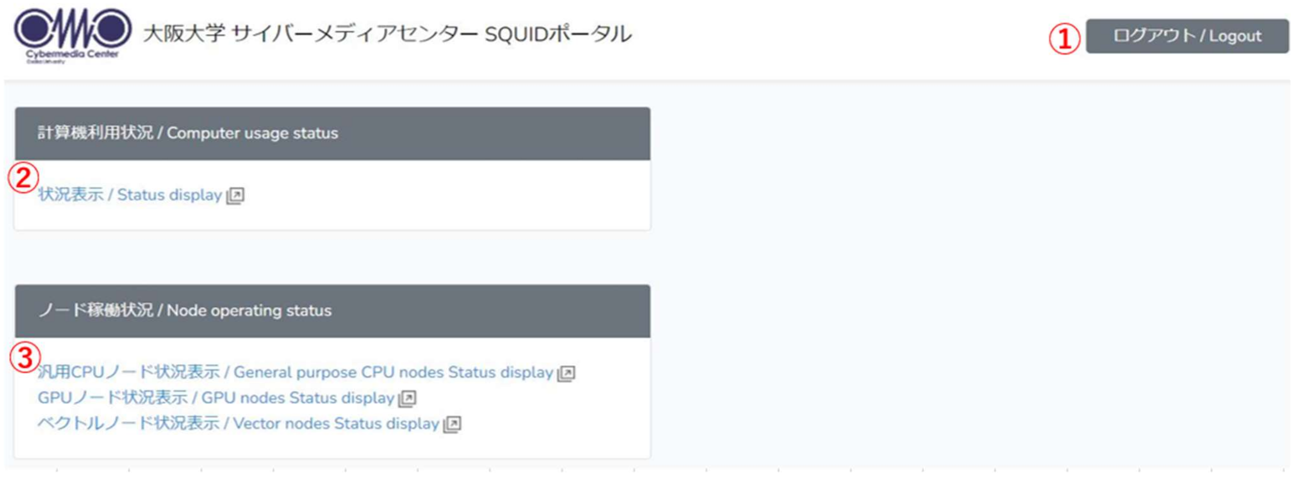
◆ Instructions

① Input the authentication code displayed on the 2-factor authentication application in the [認証コード] form

② Click [Login]

(4) Portal Home page

home page is following


大阪大学 サイバーメディアセンター SQUIDポータル

① ログアウト / Logout

計算機利用状況 / Computer usage status

② 状況表示 / Status display

ノード稼働状況 / Node operating status

③ 汎用CPUノード状況表示 / General purpose CPU nodes Status display
GPUノード状況表示 / GPU nodes Status display
ベクトルノード状況表示 / Vector nodes Status display

① Logout

Click the "Logout" button to log out and move to the login screen.

② Computer usage status

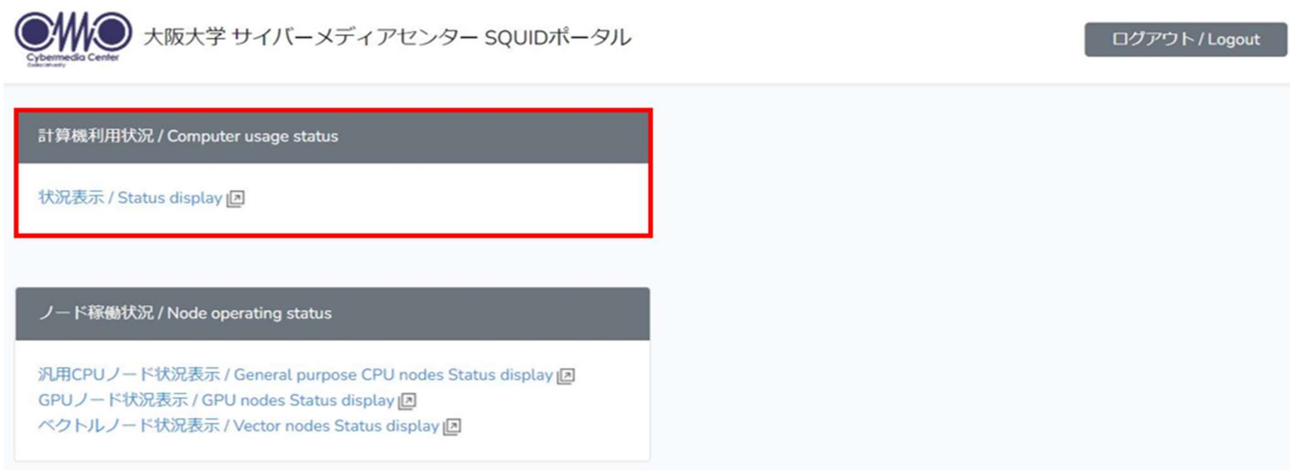
Select the "Status Display" link to display the computer usage status on a new tab screen.

③ Node operation status

Select the "Status Display" link to display the node operation status on a new tab screen.

8.2.2. Computer Usage Status Web

On the home page of the portal system, select the "Status display" link of "computer usage status" to display the computer usage status on a new tab screen.



大阪大学 サイバーメディアセンター SQUIDポータル

計算機利用状況 / Computer usage status

状況表示 / Status display

ノード稼働状況 / Node operating status

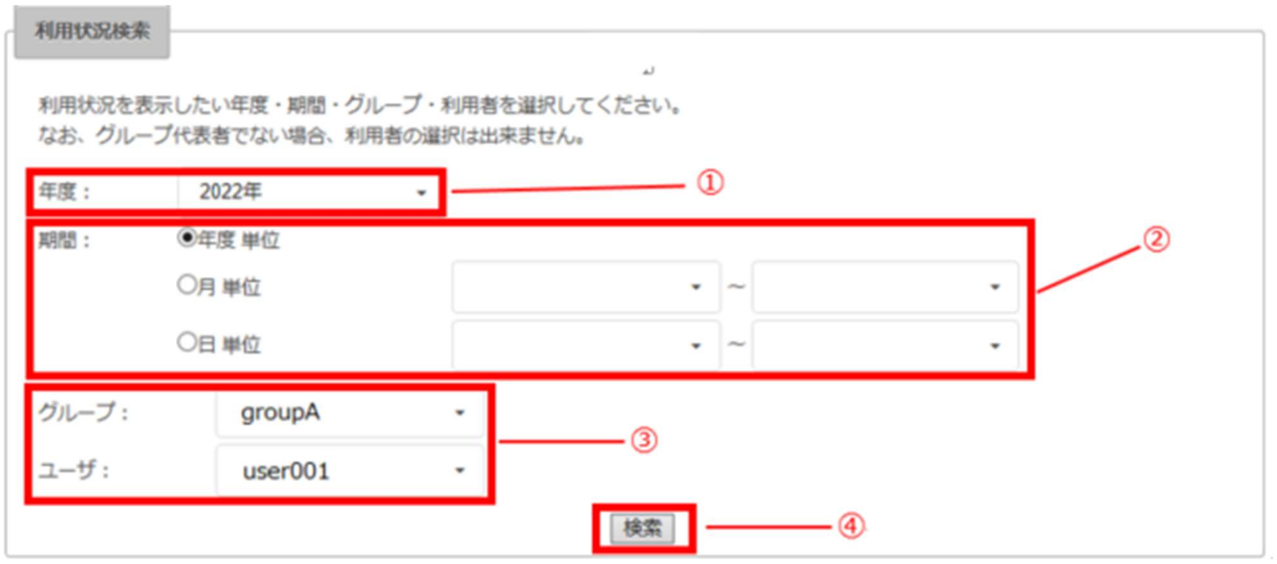
汎用CPUノード状況表示 / General purpose CPU nodes Status display

GPUノード状況表示 / GPU nodes Status display

ベクトルノード状況表示 / Vector nodes Status display

(1) Computer Usage Search Page

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



利用状況を表示したい年度・期間・グループ・利用者を選択してください。
なお、グループ代表者でない場合、利用者の選択は出来ません。

年度：	2022年
期間：	[✓ 年度 単位]
	月 単位　～
	日 単位　～
グループ：	groupA
ユーザ：	user001

◆ Instructions

① Specify the year

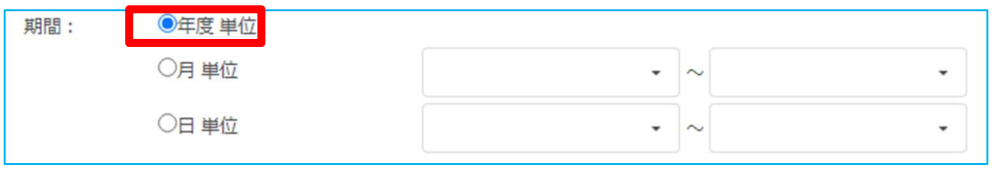
Select the year for which you want to view usage. You can select from 2021 to the current year.However, you cannot select the year before the user registration date.

*If the registration date is 2022/4/1, you can select from 2022.

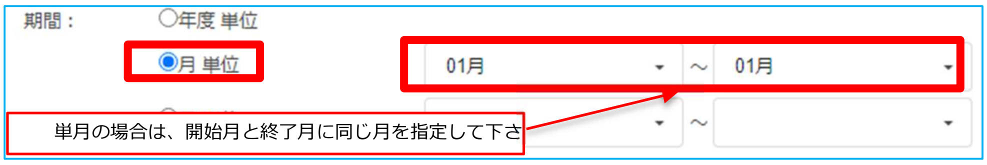
If the registration date is 2022/3/31, you can select from 2021.

② Specify the period

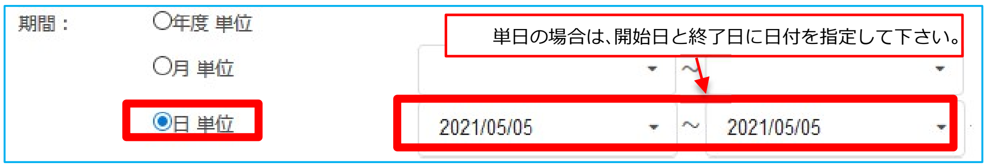
・Select "Yearly" to display the total value for each year.


期間：
[✓ 年度 単位]
○月 単位 ～
○日 単位 ～

・To display by month, select "Monthly" and enter the start and end months.
Input of only one of them will result in an error.


期間：
○年度 単位
[✓ 月 単位] 01月 ～ 01月
単月の場合は、開始月と終了月に同じ月を指定して下さ ～

・If you want to display by day, select "Daily" and enter the start date and end date.
Input of only one of them will result in an error.

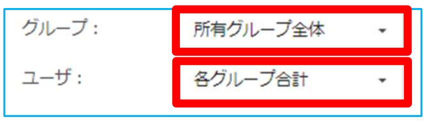

期間：
○年度 単位
○月 単位 ～
[✓ 日 単位] 2021/05/05 ～ 2021/05/05
単日の場合は、開始日と終了日に日付を指定して下さい。

③ Specify a group / user

The target data that can be selected differs depending on the operation authority of the logged-in user.

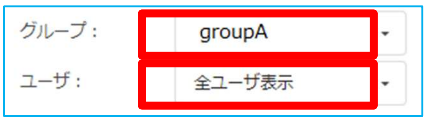
A) Application representative

- When displaying the total for each group for all groups for which you are the representative of the application


グループ： 所有グループ全体
ユーザ： 各グループ合計

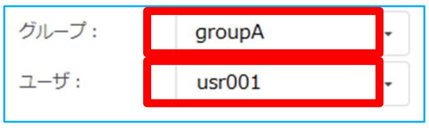
グループ ：[所有グループ全体]
ユーザ　 ：[各グループ合計]

- When displaying the data of all members and the total of the group for a specific group for which the applicant is the representative of the application.


グループ： groupA
ユーザ： 全ユーザ表示

グループ ：(specify group)
ユーザ　 ：[全ユーザ表示]

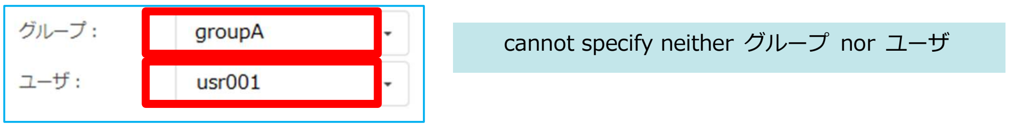
- When displaying the data of a specific member of a specific group


グループ： groupA
ユーザ： usr001

グループ ：(specify group)
ユーザ　 ：(specify user)

B)　Normal user

・　Users can only display their own data


グループ： groupA

ユーザ： usr001

cannot specify neither グループ nor ユーザ

④　Search

Click the search button to display the search result screen.

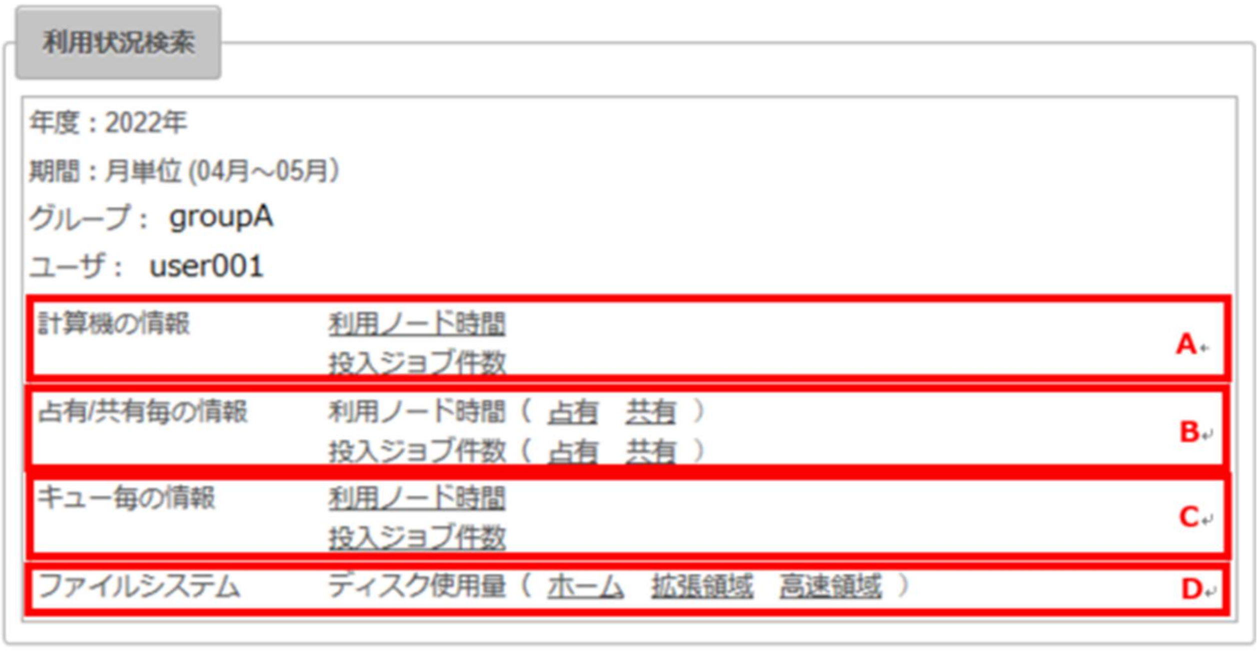
*The page will switch.

*To change the search conditions, click the browser's back button to redisplay the search screen.

(2) Search result page

This page is for viewing information according to the specified conditions.

Search results are displayed on this screen by clicking the link of the information to be viewed.


利用状況検索

年度：2022年

期間：月単位 (04月～05月)

グループ： groupA

ユーザ： user001

計算機の情報	利用ノード時間 投入ジョブ件数	A
占有/共有毎の情報	利用ノード時間（ 占有　共有 ） 投入ジョブ件数（ 占有　共有 ）	B
キュー毎の情報	利用ノード時間 投入ジョブ件数	C
ファイルシステム	ディスク使用量（ ホーム　拡張領域　高速領域 ）	D

◆　Instructions

Click the link of the target information to display the details in a separate screen.

* By clicking multiple links, each information can be displayed individually.

【Information】

A) Computer Information

Usage Node Hours , Number of Jobs

B) Information by occupancy / sharing

Usage Node Hours (occupancy / sharing), Number of Jobs (occupancy / sharing)

C) Information by Queue

Usage Node Hours , Number of Jobs

D) FileSystem

Storage Usage(home, extended, high-spped)

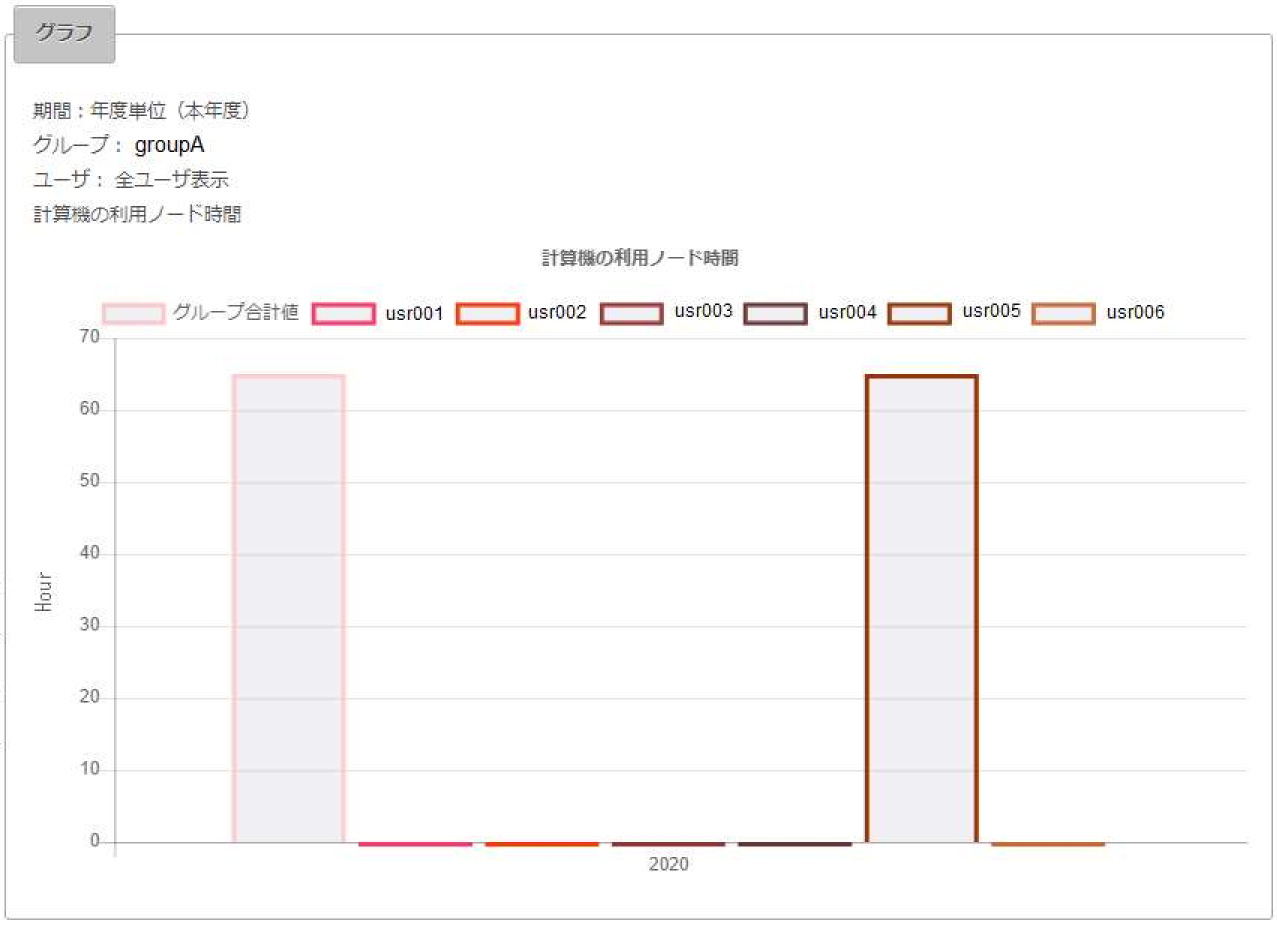
(3) Usage Result Page

Regarding the usage status, graphs and data will be displayed based on the conditions specified up to the previous screen.

* CSV output of display data is possible.

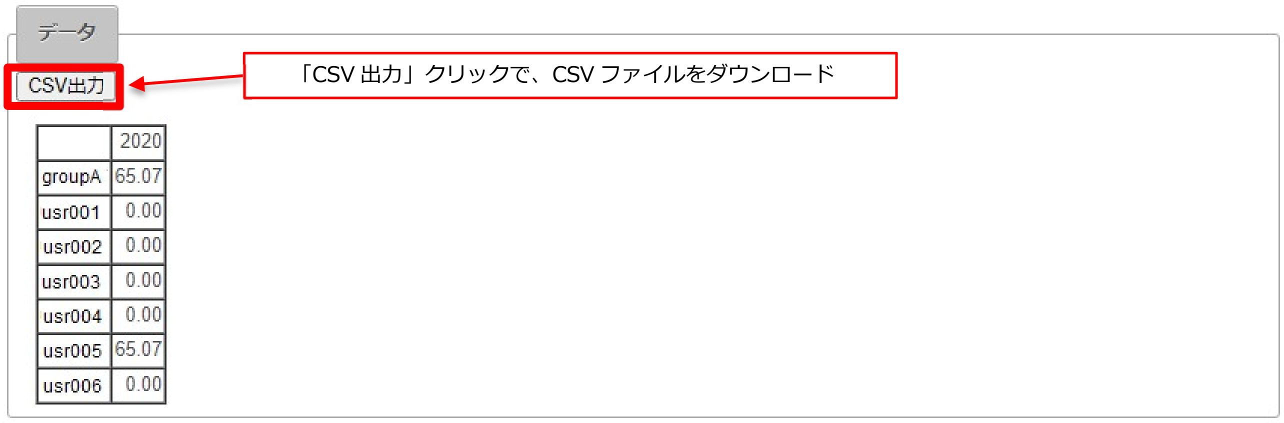
① Specifying conditions for each year

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



グラフ

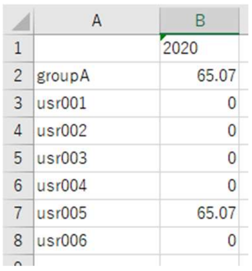
期間：年度単位（本年度）
グループ： groupA
ユーザ： 全ユーザ表示
計算機の利用ノード時間



	2020
groupA	65.07
usr001	0.00
usr002	0.00
usr003	0.00
usr004	0.00
usr005	65.07
usr006	0.00

② Annual CSV file example

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

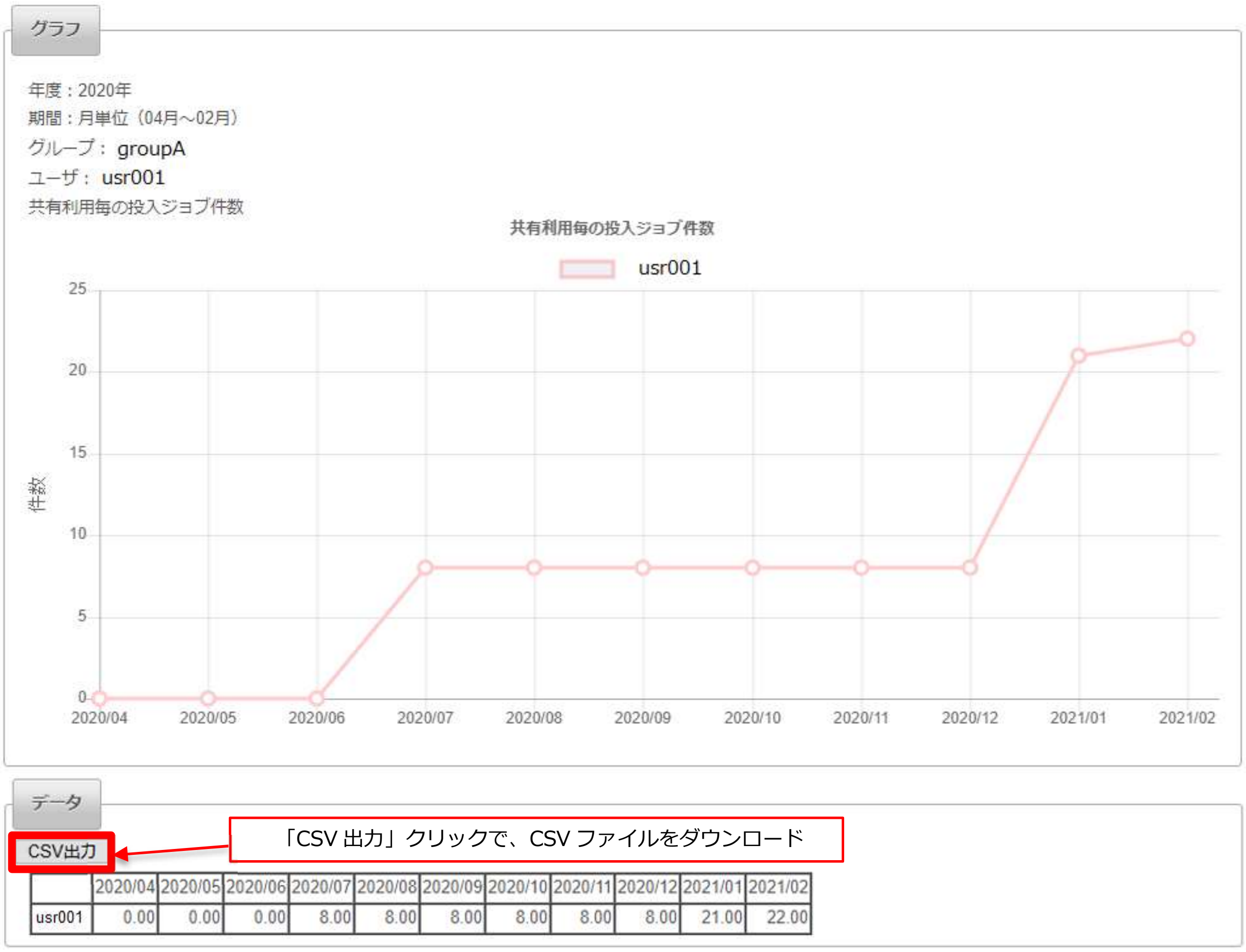


	A	B
1		2020
2	groupA	65.07
3	usr001	0
4	usr002	0
5	usr003	0
6	usr004	0
7	usr005	65.07
8	usr006	0

* This is a display example in Excel.

③ Monthly / daily condition specification

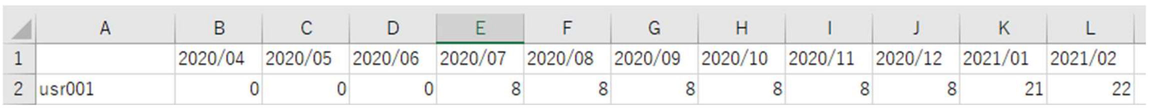
The transition for each month (day) is displayed as a graph and data.



	2020/04	2020/05	2020/06	2020/07	2020/08	2020/09	2020/10	2020/11	2020/12	2021/01	2021/02
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 / daily CSV file example

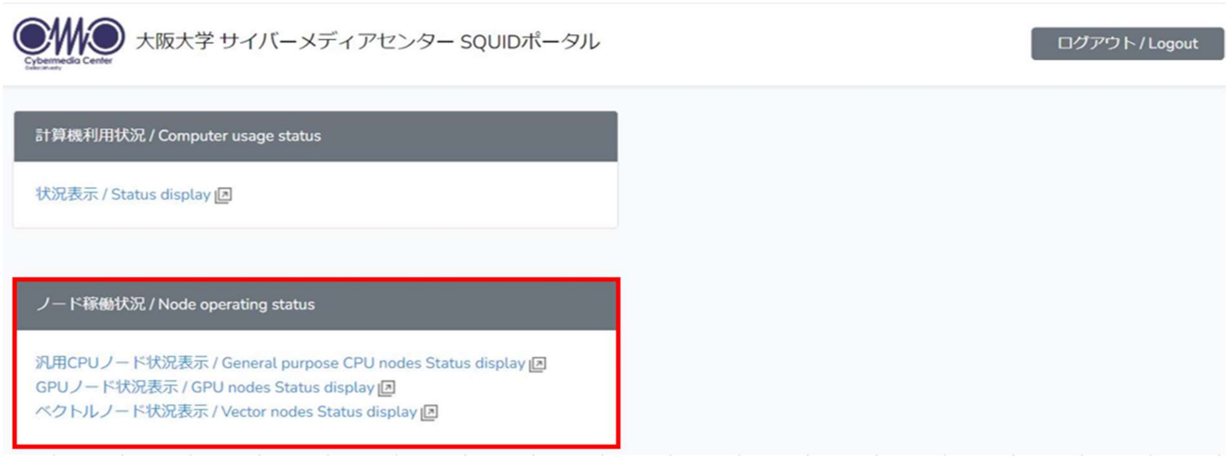
Download the same content as the "Data section" displayed on the screen in CSV format.



	A	B	C	D	E	F	G	H	I	J	K	L
1		2020/04	2020/05	2020/06	2020/07	2020/08	2020/09	2020/10	2020/11	2020/12	2021/01	2021/02
2	usr001	0	0	0	8	8	8	8	8	8	21	22

* This is a display example in Excel.

8.2.3. Node operation status Web


大阪大学 サイバーメディアセンター SQUIDポータル

ログアウト / Logout

計算機利用状況 / Computer usage status

状況表示 / Status display

ノード稼働状況 / Node operating status

汎用CPUノード状況表示 / General purpose CPU nodes Status display
GPUノード状況表示 / GPU nodes Status display
ベクトルノード状況表示 / Vector nodes Status display

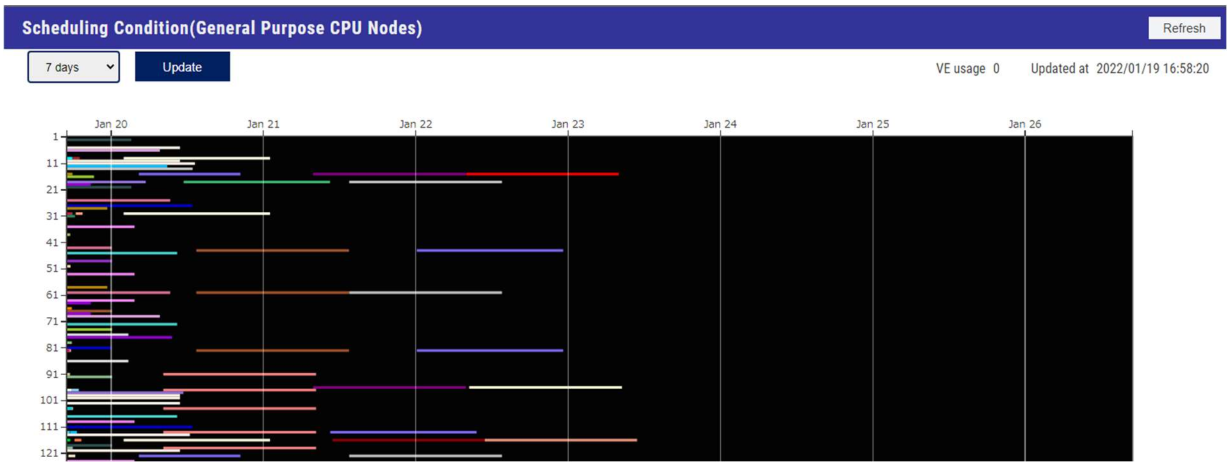
Select the link for the node you want to check.

The following describes a general-purpose CPU node as an example.

After connecting to ”General purpose CPU nodes Status display”, The following screen will be displayed.

The vertical axis shows the job server number, and the horizontal axis shows the time.

The horizontal bars in the graph represent requests, and a color is assigned to each request.


Scheduling Condition(General Purpose CPU Nodes)

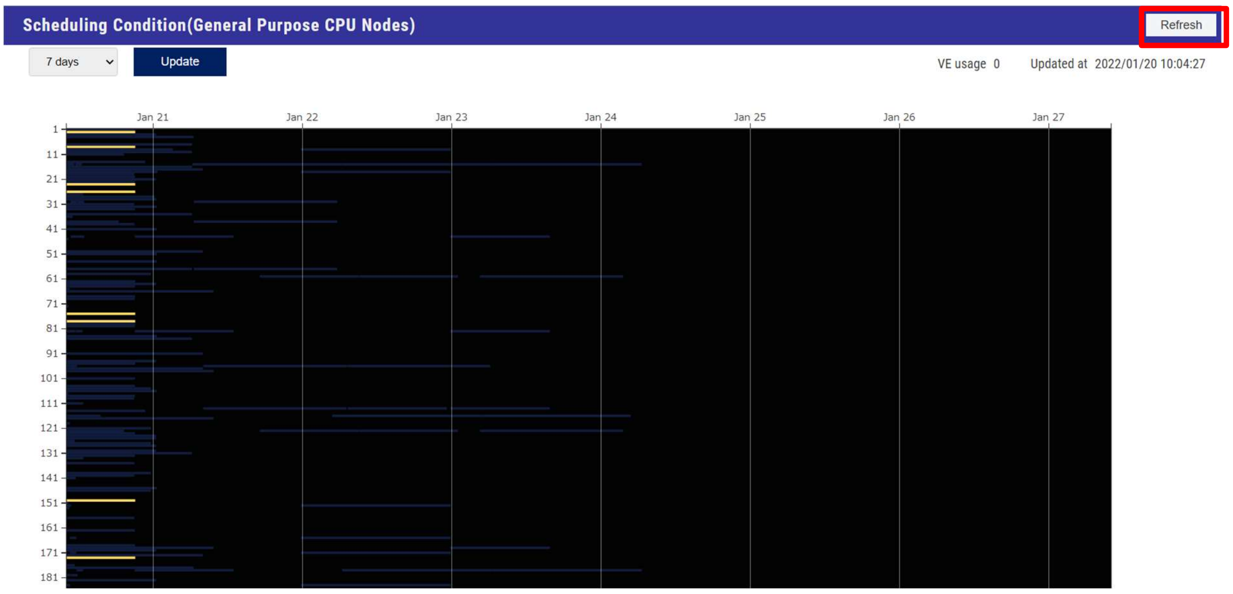
7 days

VE usage 0 Updated at 2022/01/19 16:58:20

Hover your cursor over the horizontal bar in the graph to see the request ID.

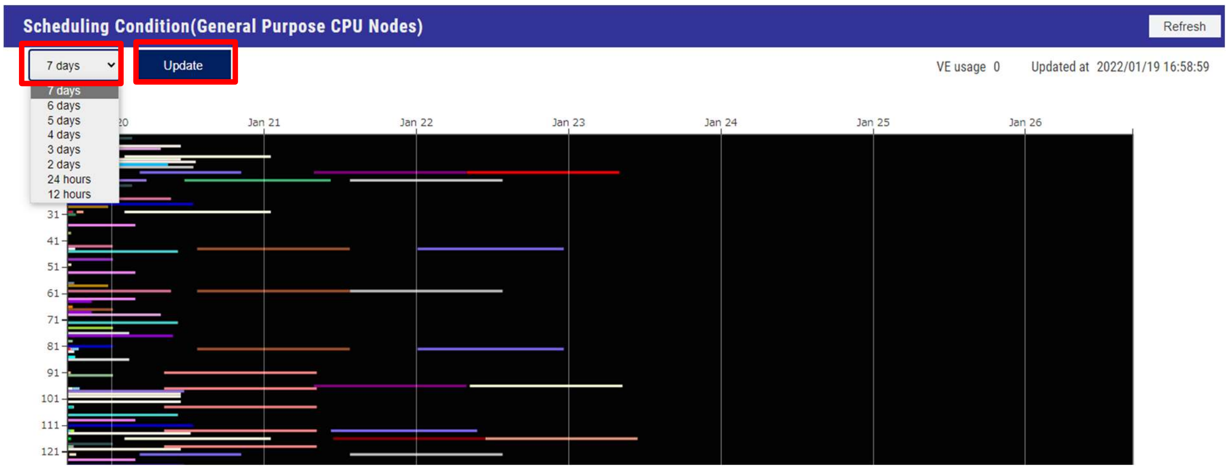
If you press the horizontal bar, only the selected request will be highlighted.

Click the “Refresh” button at the top right of the screen to refresh the page with the latest node status.

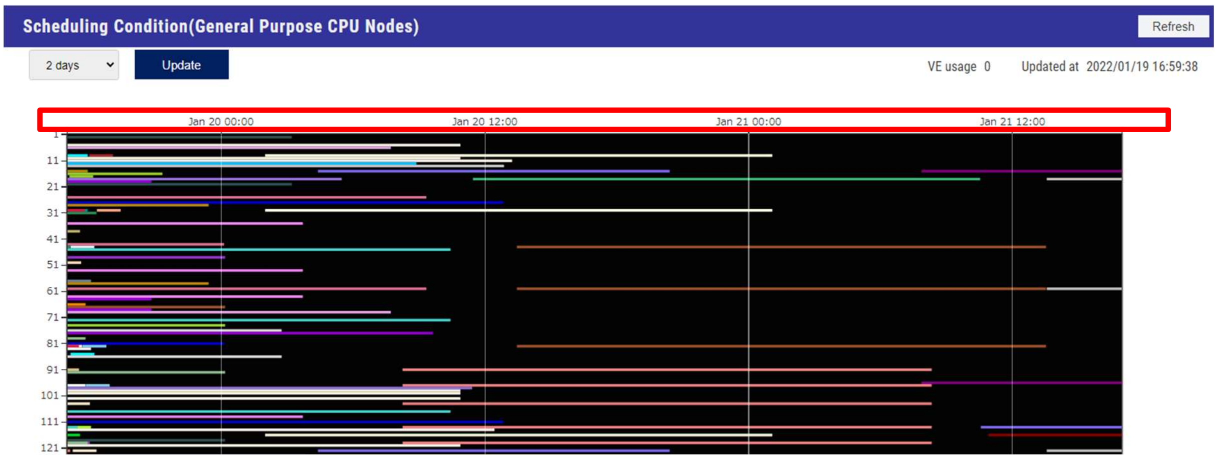

Scheduling Condition(General Purpose CPU Nodes)
Refresh
7 days
Update
VE usage 0
Updated at 2022/01/20 10:04:27

To change the display period, select the number of days or hours you want to display from the pull-down menu at the top left of the screen.

After selecting an arbitrary display period, click the "Update" button.



The display period will be changed.


Scheduling Condition(General Purpose CPU Nodes)
Refresh
2 days
Update
VE usage 0
Updated at 2022/01/19 16:59:38
Jan 20 00:00
Jan 20 12:00
Jan 21 00:00
Jan 21 12:00

fin

\Orchestrating a brighter world