



JSOL



"Private Fugaku" Installation Guide

April 1st, 2025
ver0.01

RIKEN Center for Computational Science/
JSOL Corporation / RIKEN SUURI Corporation

1. "Virtual Fugaku" Overview
 - (1) [What is "Virtual Fugaku"?](#)
 - (2) ["Satellite Fugaku" and "Private Fugaku"](#)
 - (3) [Reference Materials for "Virtual Fugaku"](#)
2. Building "Private Fugaku"
 - (1) "Private Fugaku" Configuration
 - [Example of Basic Configuration for "Private Fugaku"](#)
 - [Components of "Private Fugaku"](#)
 - [AWS Services Used in "Private Fugaku"](#)
 - (2) Cluster Environment Setup
 - [Network Configuration](#)
 - [Creating ParallelCluster Setup Instance](#)
 - [Setup of ParallelCluster](#)
 - [Creating External Storage](#)
 - [Setup of Cluster](#)
 - [Running Test Job](#)
 - [Transferring Data to the Shared Storage](#)
 - [Exporting to External Storage](#)
 - [Data Transfer from Local Environment to the Cluster](#)

INDEX

- (3) Installing "Virtual Fugaku"
 - [What is Singularity?](#)
 - [Installation of Singularity](#)
 - [Obtaining the "Virtual Fugaku" Environment](#)
 3. Running Applications
 - (1) Running applications installed on "Virtual Fugaku"
 - [Example Applications to Run](#)
 - [Execution of GENESIS](#)
 - [Execution of GROMACS](#)
 - [Execution of SCALE](#)
 4. Deleting "Private Fugaku"
 - (1) [Deleting the ParallelCluster Environment](#)
- Reference
- Reference 1:[Packages Installed in "Virtual Fugaku"](#)
- Reference 2:[List of Shared Storage in AWS ParallelCluster](#)

1. "Virtual Fugaku" Overview



(1) What is "Virtual Fugaku"?

Aim of "Virtual Fugaku"

- Widespread distribution of Software and Application outcomes of "Fugaku"
We aim to establish the de facto standard for supercomputer software stacks by packaging and globally distributing a highly optimized and fine-tuned software stack for the Fugaku supercomputer.
- Providing a Framework for Developing and Running Applications in a Standardized Environment
Traditionally, supercomputer users had to adapt to unique system configurations and familiarize themselves with new environments when migrating to a new system. Moreover, administrators had to rebuild software from scratch for each system update, resulting in significant time and resource costs. "Virtual Fugaku" addresses these challenges by offering a standardized environment, ensuring portability and optimizing software deployment.

Features of "Virtual Fugaku"

- Selecting software essential for cutting-edge research platforms.
- Since "Fugaku" is an actively operating flagship system, all software is continuously updated.
- The standard specification is application-centric (middleware is generally not included).
- The standard specification is defined based on general-purpose technologies such as Spack and container virtualization technologies.

(2) "Satellite Fugaku" and "Private Fugaku"

"Virtual Fugaku" has two environments: "Satellite Fugaku" and "Private Fugaku". This manual provides instructions for installing "Private Fugaku".

- "Satellite Fugaku"

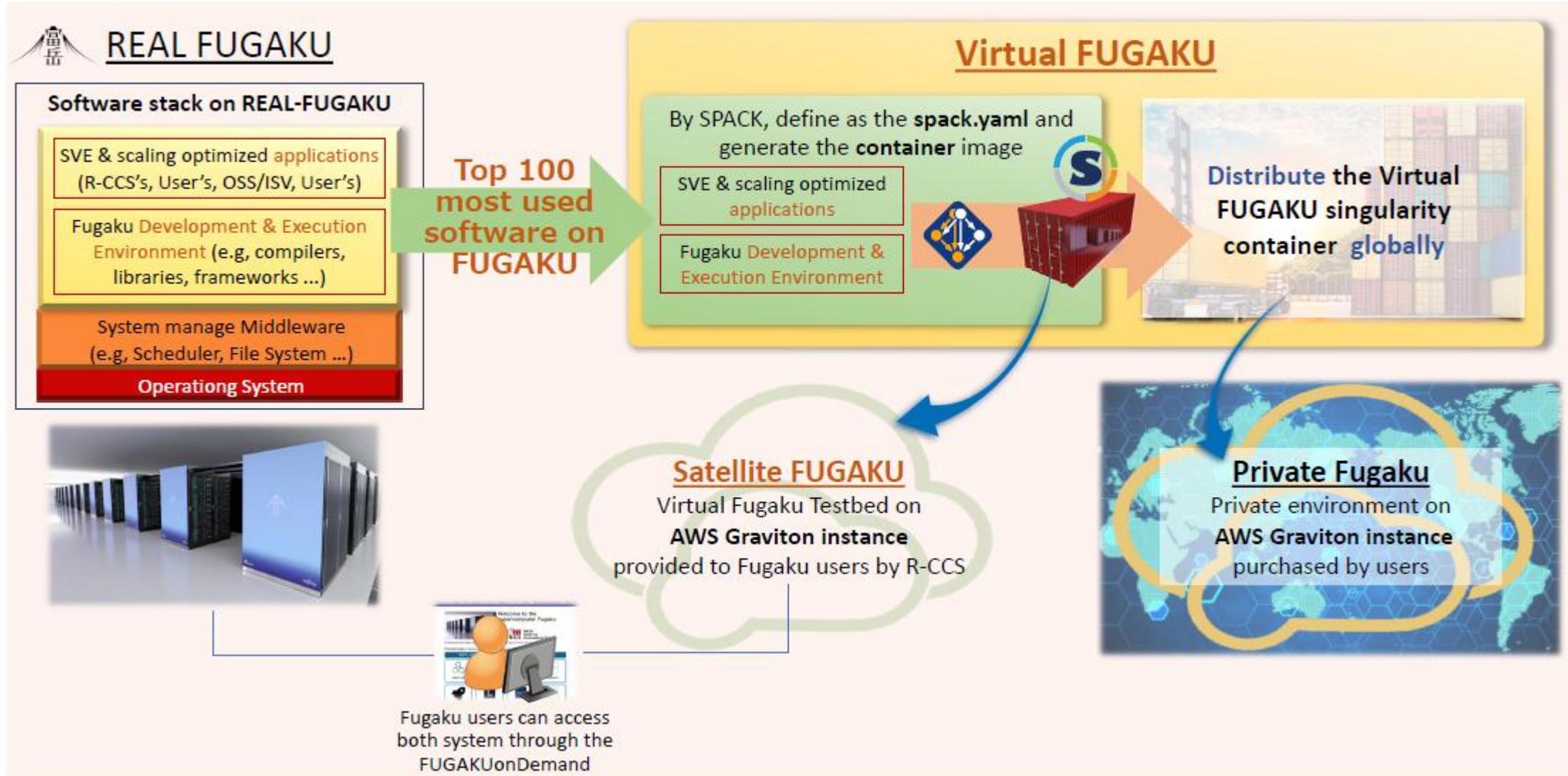
The system is a computing environment designed for testing and developing "Virtual Fugaku", an initiative proposed by R-CCS. It emulates the software stack of "Fugaku" on a commercial cloud and serves as a shared trial environment for Fugaku users".

- "Private Fugaku"

"Private Fugaku" is a self-hosted "Fugaku" environment where users deploy a pre-configured container image to their own AWS instance.

(2) "Satellite Fugaku" and "Private Fugaku"

Overview of "Satellite Fugaku" and "Private Fugaku"



(2) "Satellite Fugaku" and "Private Fugaku"

"Satellite Fugaku"

"Satellite Fugaku" is a computing system deployed on a commercial cloud that replicates the software stack of "Fugaku". It provides a testing and development environment for "Virtual Fugaku", an initiative proposed by R-CCS.

- Features
 - A restricted-access shared environment available only to "Fugaku" account holders.
 - Restricted to access from the "Fugaku" login node.
 - A small-scale environment intended for validation purposes.
- Use Cases
 - A testing environment for "Virtual Fugaku", available to "Fugaku" users.

Designed for

- "Fugaku" users interested in experimenting with "Virtual Fugaku".
- Software developers planning to deploy software on "Virtual Fugaku" in the future.
- By enabling "Fugaku" users to experiment with "Virtual Fugaku", the development team can evaluate and refine the software stack.

(2) "Satellite Fugaku" and "Private Fugaku"

"Private Fugaku"

"Private Fugaku" allows users to create a "Fugaku-like" environment by deploying "Virtual Fugaku" on their cloud instance or computing system. This enables "Fugaku" users to reproduce the "Fugaku" software environment in a private setting. Moreover, users can work within a highly secure and confidential environment with no obligation to disclose their results.

- Features

- No review process is required, and users can freely set up the environment on their system.
- There is no obligation to disclose results, enabling users to maintain a secure and confidential environment.

- Use Cases

- Secure and Confidential Simulation Environment

In cutting-edge research, simulations are conducted on "Fugaku", and during product development utilizing these results, simulations are conducted on "Private Fugaku" while maintaining confidentiality.

- Pre-Simulation Validation Environment for "Fugaku"

"Private Fugaku" serves as a small-scale computing environment for developing and validating simulation code and computational models. Once validated, they are expanded and executed on "Fugaku" for large-scale simulations.

(3) Reference Materials for "Virtual Fugaku"

Reference

For more detailed information about "Virtual Fugaku", please refer to the following RIKEN website and press release.

- "Virtual Fugaku" (R-CCS Official Website)
URL: <https://www.r-ccs.riken.jp/en/fugaku/virtual-fugaku/>
- A Major Step Towards Building
an Ecosystem Leading to the Next-Generation Supercomputing Infrastructure
URL: <https://www.r-ccs.riken.jp/en/outreach/topics/20240805-1/>

Contact



virtual-fugaku@ml.riken.jp

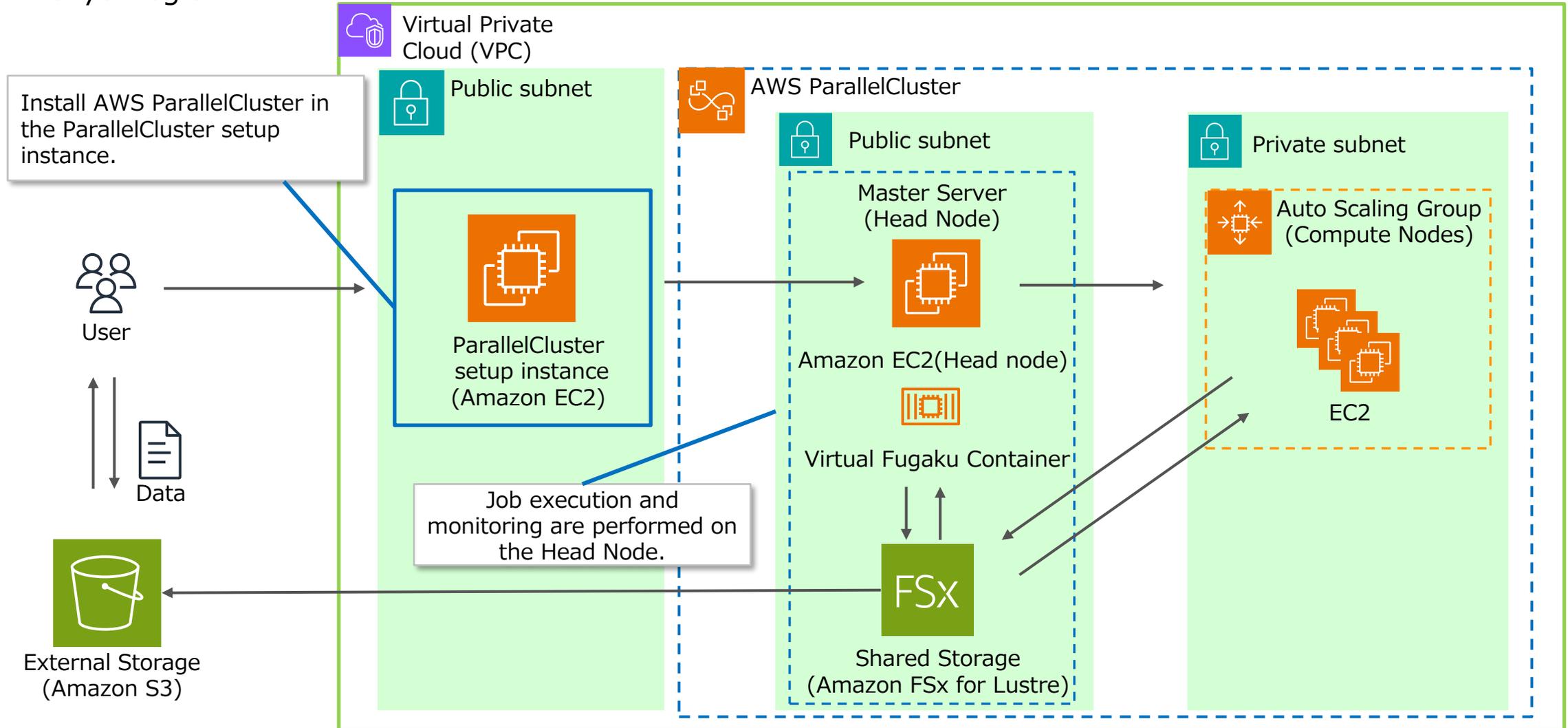
2. Building "Private Fugaku"

(1) "Private Fugaku" Configuration



Example of Basic Configuration for "Private Fugaku"

"Virtual Fugaku" has undergone testing and validation on AWS instances based on Graviton3/3E. This manual provides instructions for building "Private Fugaku" on the AWS cloud environment, using the following configuration as a reference. The environment can be deployed in any AWS region. For this manual, we use the Tokyo Region.



Components of "Private Fugaku"

"Private Fugaku" consists of the following components:

Head Node

The primary access point to compute nodes equipped with vast computational resources. It provides an environment where users can browse files, submit and monitor jobs, and compile code.

Compute Nodes

Nodes designed for long-running computations or tasks that require significant CPU and memory resources. Direct login to these nodes is not possible; users must execute applications and programs via job scripts submitted from the Head Node.

"Virtual Fugaku" Container

A Singularity container image that includes frequently used software from the "Fugaku" software stack. By deploying the "Virtual Fugaku" container image on a purchased cloud instance, users can create an environment equivalent to "Fugaku".

Shared Storage

A shared storage system for the Head Node and Compute Nodes, used to store input and output files.

External Storage

A storage system for long-term data retention. Since an AWS ParallelCluster environment is deleted along with all stored data upon removal, exporting files from the cluster's internal file system to external storage ensures data preservation even after the cluster is deleted.

AWS Services Used in "Private Fugaku"

The following AWS services are primarily used in "Private Fugaku":

Amazon VPC

An AWS service that allows users to create virtual networks. The cluster is built within a virtual network created with this service.

Amazon EC2

An AWS service that enables users to create virtual servers. AWS ParallelCluster is installed on a virtual server provisioned using this service for cluster setup.

AWS ParallelCluster

An open-source cluster management tool that simplifies deploying and managing high-performance computing (HPC) clusters. This tool is used to set up the cluster for "Private Fugaku".

AWS Services Used in "Private Fugaku"

The storage of "Private Fugaku" primarily uses the following AWS services:

AWS FSx for Lustre

This is a fully managed shared storage service provided by AWS. It is used as shared storage, allowing both the Head Node and Compute Nodes to store and delete data.

Amazon S3

This is an object storage service provided by AWS and is utilized as external storage. By exporting data stored in the cluster's shared storage to Amazon S3-based external storage, the data remains available even after the cluster is deleted.

AWS Services Used in "Private Fugaku"

For more details on each AWS service, please refer to the following URL.

- Amazon VPC
URL: https://aws.amazon.com/vpc/?nc1=h_ls
- Amazon EC2
URL: https://aws.amazon.com/ec2/?nc1=h_ls
- AWS ParallelCluster
URL: https://aws.amazon.com/hpc/parallelcluster/?nc1=h_ls
- AWS FSx for Lustre
URL: https://aws.amazon.com/fsx/lustre/?nc1=h_ls
- Amazon S3
URL: <https://aws.amazon.com/s3/>

2. Building "Private Fugaku" (2) Cluster Environment Setup



Network Configuration

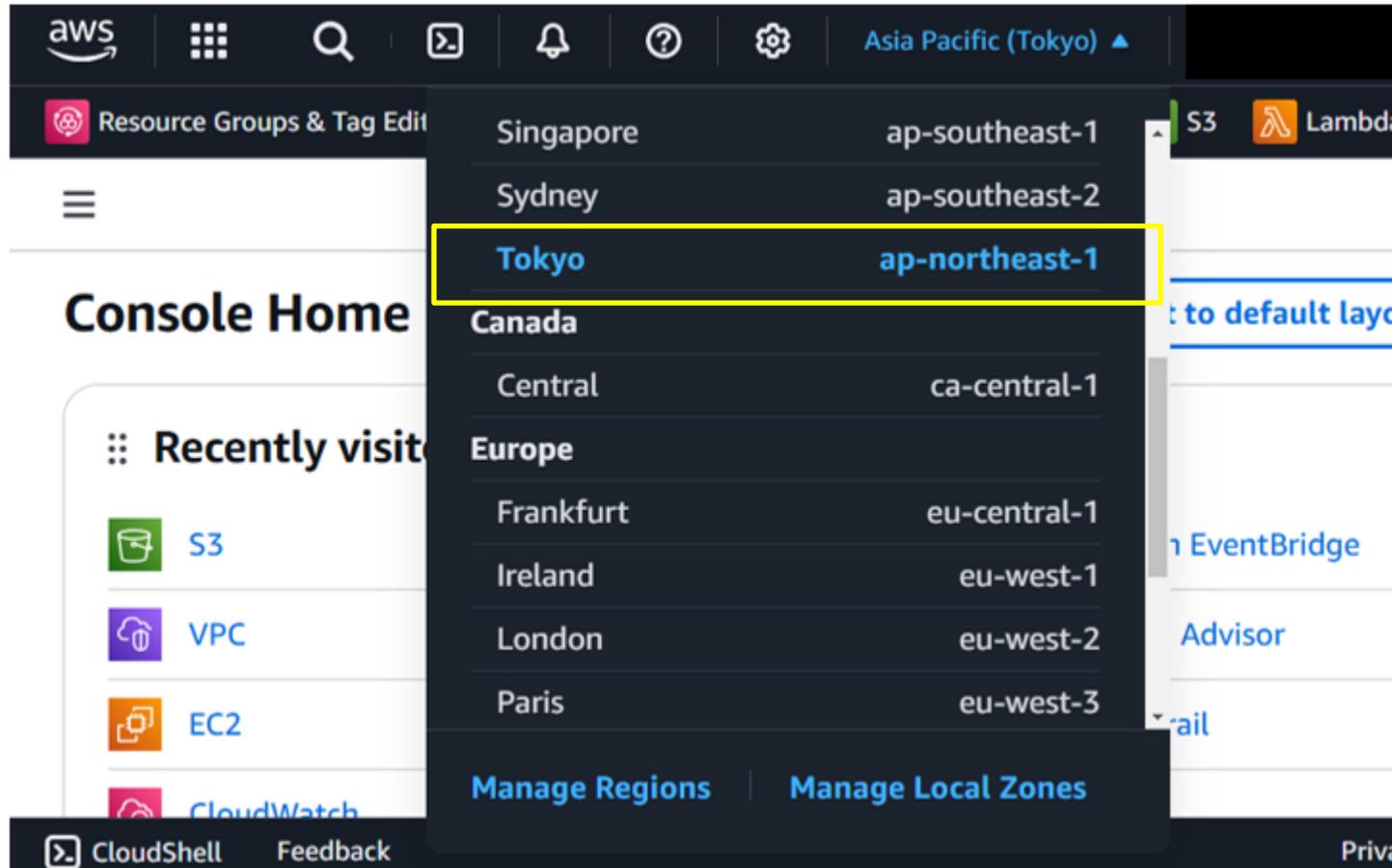
Before setting up a cluster with AWS ParallelCluster, network configuration must be completed. This section explains how to configure the network for cluster setup. The network environment is configured following the steps below, and each step is explained in detail on the following pages.

1. Choosing Region
2. Creating VPC
3. Enable DNS Hostnames
4. Creating Public Subnet
5. Creating Internet Gateway
6. Attaching the Internet Gateway to VPC
7. Creating Route Table
8. Associating Route Table
9. Adding Route Entry
10. Creating Key Pair
11. Creating Security Group

Network Configuration

1. Choosing Region

1.1 In the top-right corner of the AWS Management Console, open the dropdown menu and select "Tokyo (ap-northeast-1)".



Network Configuration

2. Creating VPC (1/3)

2.1 In the AWS Management Console, navigate to the VPC list page.

<https://ap-northeast-1.console.aws.amazon.com/vpcconsole/home?region=ap-northeast-1#vpcs>

2.2 Click "Create VPC" in the upper-right corner.

Your VPCs (5) [Info](#)

<input type="checkbox"/>	Name	VPC ID	State	Block Public...	IPv4 CIDR	IPv6 CIDR	DHCP option se
--------------------------	------	--------	-------	-----------------	-----------	-----------	----------------



Last updated
less than a minute ago

Actions

Create VPC

< 1 > | ⚙️

Network Configuration

2. Creating VPC (2/3)

2.3 Specify the parameters as follows.

Create VPC Info

A VPC is an isolated portion of the AWS Cloud populated by AWS objects, such as A

VPC settings

Resources to create Info
Create only the VPC resource or the VPC and other networking resources

VPC only VPC and more

Name tag - optional
Creates a tag with a key of 'Name' and a value that you specify.

pcluster-prod-vpc

IPv4 CIDR block Info

IPv4 CIDR manual input IPAM-allocated IPv4 CIDR block

IPv4 CIDR

10.0.0.0/16

CIDR block size must be between /16 and /28.

IPv6 CIDR block Info

No IPv6 CIDR block IPAM-allocated IPv6 CIDR block Amazon-provided IPv6 CIDR block IPv6 CIDR owned by me

Tenancy Info

Default

Select "VPC only".

Enter any name. In this example, use "pcluster-prod-vpc".

Set the value to "10.0.0.0/16".

Select "No IPv6 CIDR block".

Select "Default".

Network Configuration

2. Creating VPC (3/3)

2.4 Click "Create VPC".



2.5 The VPC creation is complete when the following screen appears.



Network Configuration

3. Enable DNS Hostnames (1/2)

3.1 In the AWS Management Console, navigate to the VPC list page.

<https://ap-northeast-1.console.aws.amazon.com/vpcconsole/home?region=ap-northeast-1#vpcs>

3.2 Select the VPC you created.

Your VPCs (1/5) [Info](#)

Search

Name	VPC ID	State	Block Public...	IPv4 CIDR	IPv6 CIDR
<input checked="" type="checkbox"/> pcluster-prod-vpc		Available	Off	10.0.0.0/16	-

Create VPC

3.3 Click "Actions" in the upper-right corner, then select "Edit VPC settings".

vpc- / pcluster-prod-vpc

Actions

- Create flow log
- Edit VPC settings

Details [Info](#)

Network Configuration

3. Enable DNS Hostnames (2/2)

3.4 Select "Enable DNS resolution" and "Enable DNS hostnames", then click "Save".

Edit VPC settings [Info](#)

VPC details

VPC ID



Name



DHCP settings

DHCP option set [Info](#)

dopt-

Check "Enable DNS resolution" and "Enable DNS hostnames".

DNS settings

Enable DNS resolution [Info](#)

Enable DNS hostnames [Info](#)

Network Address Usage metrics settings

Enable Network Address Usage metrics [Info](#)

Click "Save".

Cancel

Save

Network Configuration

4. Creating Public Subnet (1/3)

4.1 In the AWS Management Console, navigate to the subnet list page.

<https://ap-northeast-1.console.aws.amazon.com/vpcconsole/home?region=ap-northeast-1#subnets>

4.2 Click "Create Subnet".

Click "Create Subnet".



Network Configuration

4. Creating Public Subnet (2/3)

4.3 Set the parameters as follows.

Create subnet Info

VPC
VPC ID
Create subnets in this VPC.
 vpc-0935d040728d3cb41 (pcluster-prod-vpc)

Associated VPC CIDRs

IPv4 CIDRs
 10.0.0.0/16

Subnet settings
Specify the CIDR blocks and Availability Zone for the subnet.

Subnet 1 of 1

Subnet name
Create a tag with a key of 'Name' and a value that you specify.
 pcluster-prod-private-subnet-1a
The name can be up to 256 characters long.

Availability Zone Info
Choose the zone in which your subnet will reside, or let Amazon choose one for you.
 Asia Pacific (Tokyo) / ap-northeast-1a

IPv4 VPC CIDR block Info
Choose the VPC's IPv4 CIDR block for the subnet. The subnet's IPv4 CIDR must lie within this block.
 10.0.0.0/16

IPv4 subnet CIDR block
 10.0.11.0/24 256 IPs

Select "pcluster-prod-vpc", which was created earlier.

Enter any name. In this example, use "pcluster-prod-private-subnet-1a".

Set the value to "ap-northeast-1a".

Set the value to "10.0.0.0/16".

Set the value to "10.0.11.0/24".

Network Configuration

4. Creating Public Subnet (3/3)

4.4 Click "Create Subnet".



4.5 Subnet creation is complete when the following screen appears.



Network Configuration

5. Creating Internet Gateway (1/2)

5.1 In the AWS Management Console, navigate to the Internet Gateway list page.

<https://ap-northeast-1.console.aws.amazon.com/vpconsole/home?region=ap-northeast-1#igws>

5.2 Click "Create internet gateway".

The screenshot shows the AWS Management Console interface for Internet Gateways. At the top, there is a header with the text "Internet gateways (5) Info" and a refresh icon. Below this is a search bar with the placeholder text "Search". To the right of the search bar is an "Actions" dropdown menu, which is open, showing a "Create internet gateway" button. This button is highlighted with a blue box. A blue arrow points from the "Create internet gateway" button to a callout box that contains the text "Click 'Create internet gateway'." Below the search bar and actions menu is a table with the following columns: "Name", "Internet gateway ID", "State", and "VPC ID". The table is currently empty.

Network Configuration

5. Creating Internet Gateway (2/2)

5.3 Configure the parameters as follows, then click "Create internet gateway".

Create internet gateway [Info](#)

An internet gateway is a virtual router that connects a VPC to the internet. To create a new internet gateway, you must specify a name and a tag.

Enter any name. In this example, use "pcluster-prod-igw".

Internet gateway settings

Name tag

Creates a tag with a key of 'Name' and a value that you specify.

pcluster-prod-igw

Add new tag

You can add 47 more tags.

Click "Create internet gateway".

Cancel

Create internet gateway

5.4 The Internet Gateway is successfully created when the following message appears on the screen.

✓ The following internet gateway was created: igw-██████████ - pcluster-prod-igw. You can now attach to a VPC to enable the VPC to communicate with the internet.

Attach to a VPC



Network Configuration

6. Attaching the Internet Gateway to VPC (1/2)

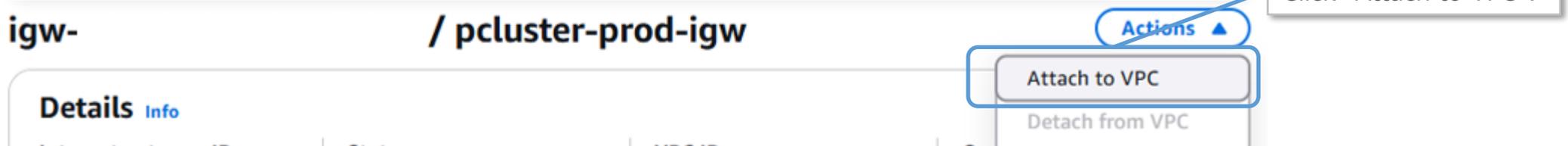
6.1 In the AWS Management Console, navigate to the Internet Gateway list page.

<https://ap-northeast-1.console.aws.amazon.com/vpconsole/home?region=ap-northeast-1#igws>

6.2 Select the Internet Gateway you created earlier.



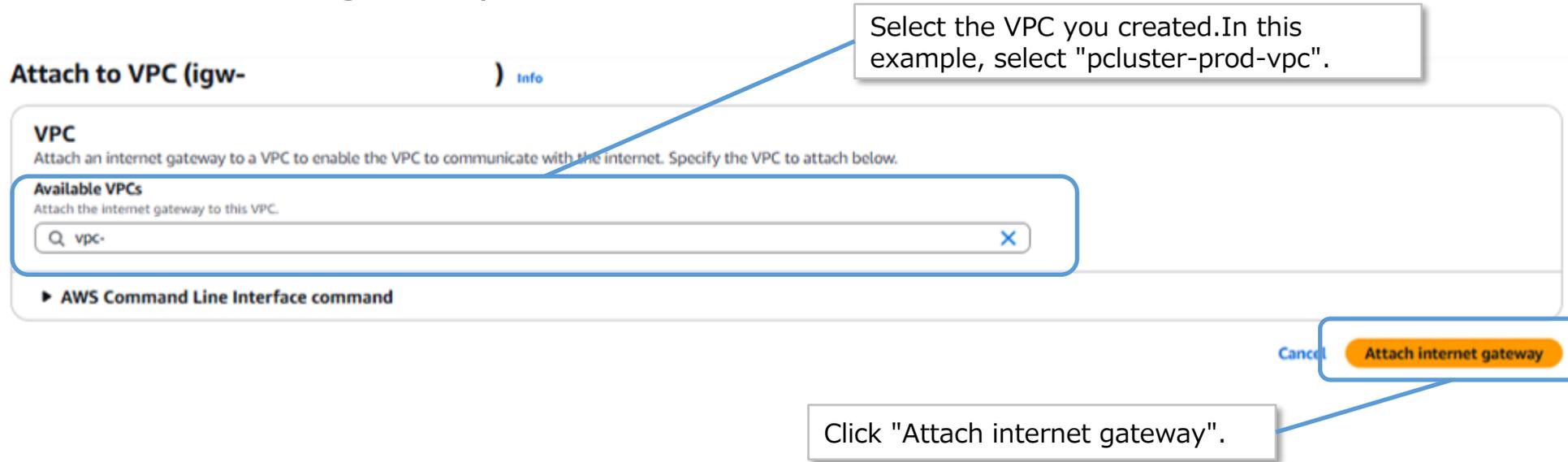
6.3 Click "Actions" in the upper-right corner, then select "Attach to VPC".



Network Configuration

6. Attaching the Internet Gateway to VPC (2/2)

6.4 From the "Available VPCs" dropdown menu, select the VPC you created, then click "Attach internet gateway".



Attach to VPC (igw-) Info

VPC
Attach an internet gateway to a VPC to enable the VPC to communicate with the internet. Specify the VPC to attach below.

Available VPCs
Attach the internet gateway to this VPC.

Q vpc- X

▶ AWS Command Line Interface command

Cancel Attach internet gateway

Select the VPC you created. In this example, select "pcluster-prod-vpc".

Click "Attach internet gateway".

6.5 The Internet Gateway is successfully attached to the VPC when the following message appears on the screen.



Network Configuration

7. Creating Route Table (1/3)

7.1 In the AWS Management Console, navigate to the Route Table list page.

<https://ap-northeast-1.console.aws.amazon.com/vpcconsole/home?region=ap-northeast-1#RouteTables>

7.2 Click "Create route table".



Click "Create route table".

Network Configuration

7. Creating Route Table (2/3)

7.3 Configure the parameters as follows, then click "Create route table".

Create route table info

A route table specifies how packets are forwarded between the subnets within your VPC, the internet, and your VPN connection.

Route table settings

Name - optional
Create a tag with a key of 'Name' and a value that you specify.

VPC
The VPC to use for this route table.

[Add new tag](#)
You can add 47 more tags.

[Cancel](#) [Create route table](#)

Enter any name. In this example, use "pcluster-prod-public-1a-rtb".

Select the VPC you created. In this example, select "pcluster-prod-vpc".

Click "Create route table".

Network Configuration

7. Creating Route Table (3/3)

7.4 The route table creation is complete when the following message appears on the screen.

A green success message banner with a white checkmark icon on the left and a white 'X' icon on the right. The text in the center reads: "Route table rtb-[redacted] | pcluster-prod-public-1a-rtb was created successfully."

Route table rtb-[redacted] | pcluster-prod-public-1a-rtb was created successfully. X

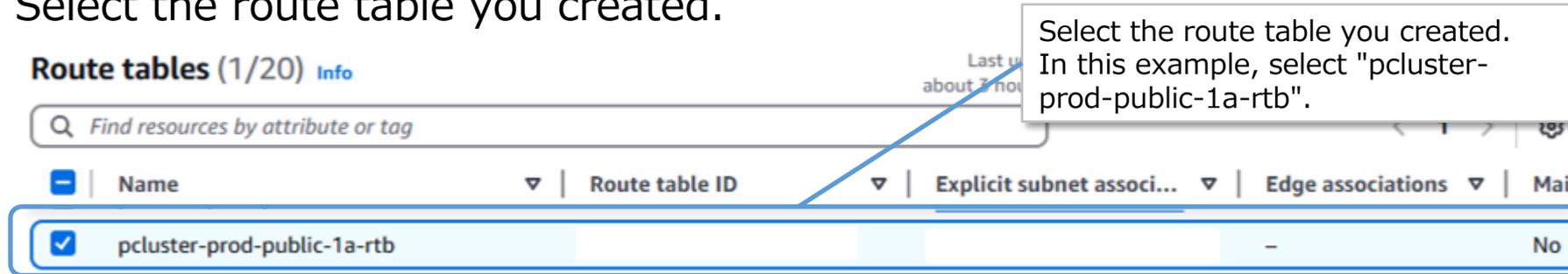
Network Configuration

8. Associating Route Table (1/2)

8.1 In the AWS Management Console, navigate to the Route Tables list page.

<https://ap-northeast-1.console.aws.amazon.com/vpccconsole/home?region=ap-northeast-1#RouteTables>

8.2 Select the route table you created.



Route tables (1/20) Info

Find resources by attribute or tag

Name	Route table ID	Explicit subnet associ...	Edge associations	Mail
<input checked="" type="checkbox"/> pcluster-prod-public-1a-rtb		-		No

8.3 Click "Actions" in the upper-right corner, then select "Edit subnet associations".



rtb- / pcluster-prod-public-1a-rtb

Actions

- Set main route table
- Edit subnet associations**
- Edit edge associations

Details Info

Route table ID	Main	Explicit subnet associations	E
----------------	------	------------------------------	---

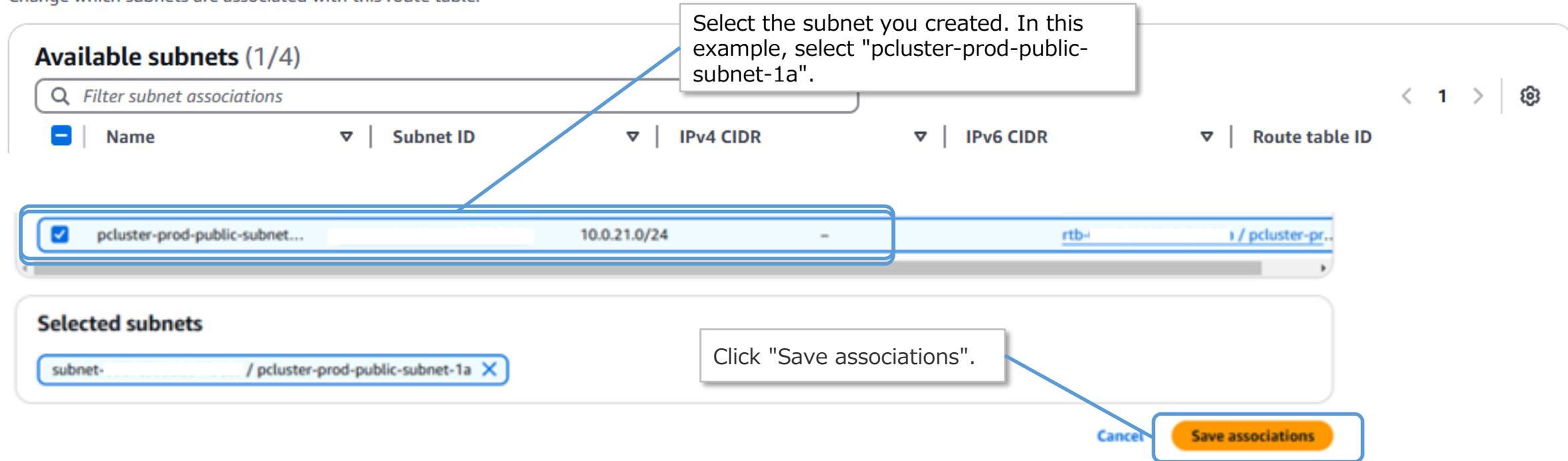
Network Configuration

8. Associating Route Table (2/2)

8.4 Select the subnet you created, then click "Save associations".

Edit subnet associations

Change which subnets are associated with this route table.



Select the subnet you created. In this example, select "pcluster-prod-public-subnet-1a".

Click "Save associations".

<input type="checkbox"/>	Name	Subnet ID	IPv4 CIDR	IPv6 CIDR	Route table ID
<input checked="" type="checkbox"/>	pcluster-prod-public-subnet...		10.0.21.0/24	-	rtb-... / pcluster-pr...

Selected subnets: subnet-... / pcluster-prod-public-subnet-1a

Buttons: Cancel, Save associations

8.5 The route table has been successfully associated when the following message appears on the screen.



✔ You have successfully updated subnet associations for rtb-... / pcluster-prod-public-1a-rtb.

Network Configuration

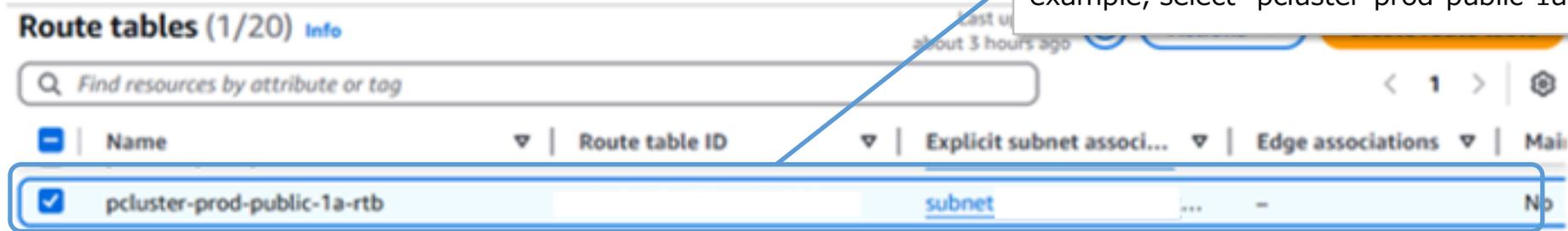
9. Adding Route Entry (1/3)

9.1 In the AWS Management Console, navigate to the Route Tables list page.

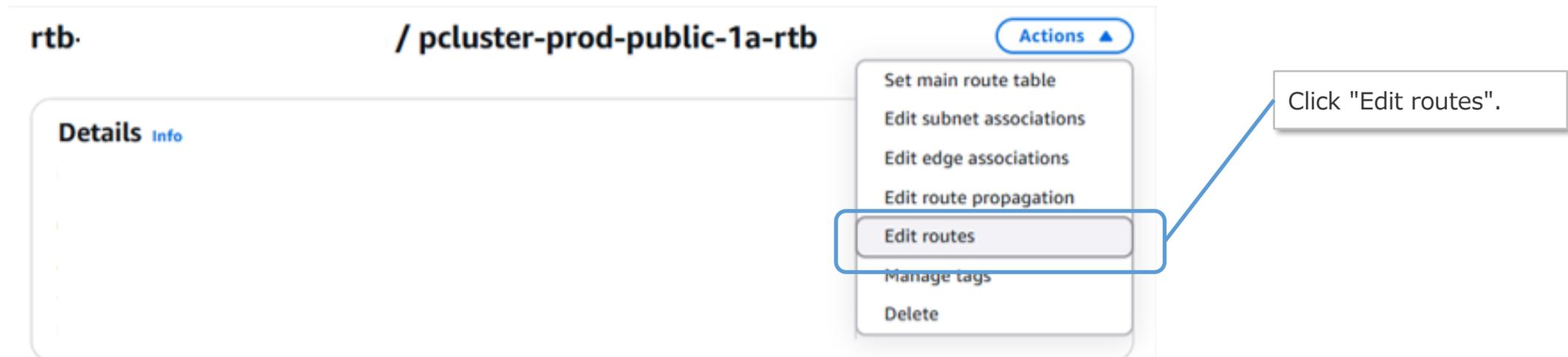
<https://ap-northeast-1.console.aws.amazon.com/vpccconsole/home?region=ap-northeast-1#RouteTables>

9.2 Select the route table you created.

Select the route table you created. In this example, select "pcluster-prod-public-1a-rtb".



9.3 In the upper-right corner, click "Actions", then select "Edit routes".

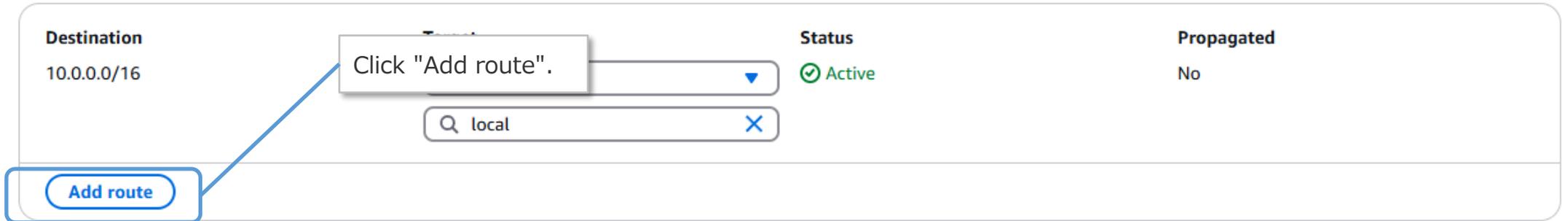


Network Configuration

9. Adding Route Entry (2/3)

9.4 Click "Add route".

Edit routes

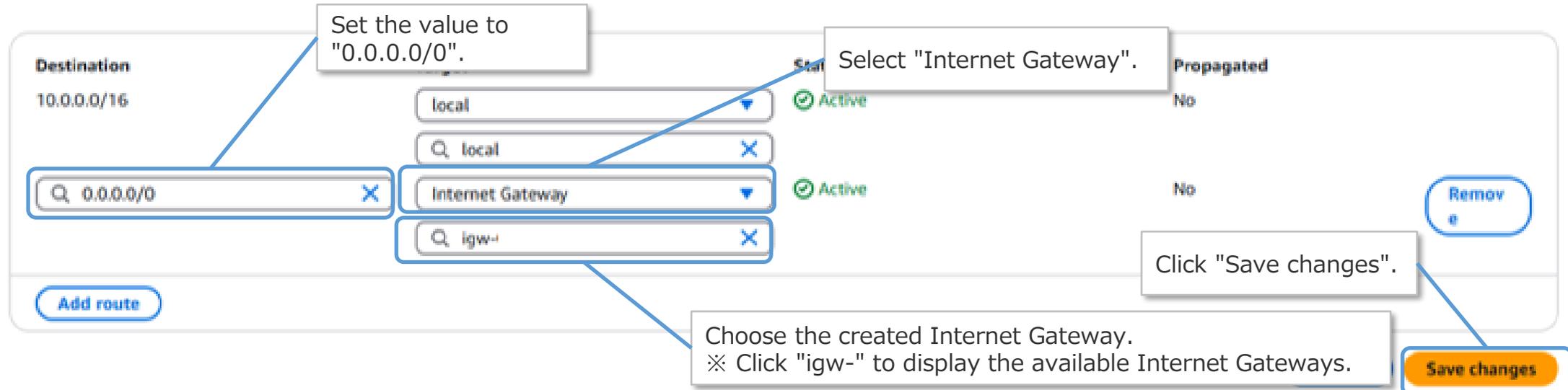


Destination	Status	Propagated
10.0.0.0/16	Active	No

Click "Add route".

Add route

9.5 "Configure the parameters as follows, then click "Save changes".



Destination	Status	Propagated
10.0.0.0/16	Active	No
0.0.0.0/0	Active	No

Set the value to "0.0.0.0/0".

Select "Internet Gateway".

Click "Save changes".

Choose the created Internet Gateway.
 ※ Click "igw-" to display the available Internet Gateways.

Add route

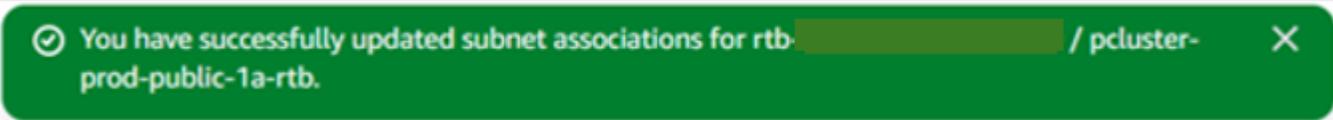
Remove

Save changes

Network Configuration

9. Adding Route Entry (3/3)

9.6 The route entry addition is complete when the following message appears on the screen.

A green notification bar with a white checkmark icon on the left and a white 'X' icon on the right. The text inside reads: "You have successfully updated subnet associations for rtb- [redacted] / pcluster-prod-public-1a-rtb."

✔ You have successfully updated subnet associations for rtb- [redacted] / pcluster-prod-public-1a-rtb. ✕

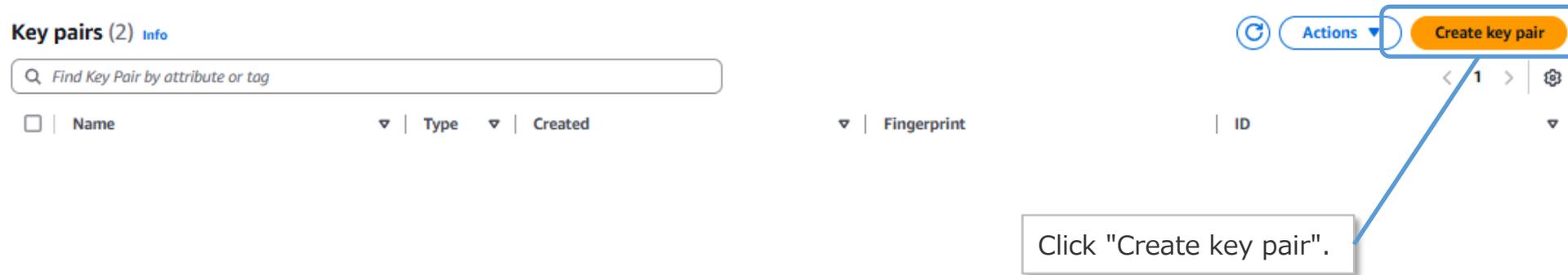
Network Configuration

10. Creating Key Pair (1/3)

10.1 In the AWS Management Console, navigate to the Key Pairs list page.

<https://ap-northeast-1.console.aws.amazon.com/ec2/home?region=ap-northeast-1#KeyPairs>

10.2 Click "Create key pair".



Network Configuration

10. Creating Key Pair (2/3)

10.3 Configure the parameters as follows.

Create key pair [Info](#)

Key pair
 A key pair, consisting of a private key and a public key, is a set of security credentials that you use to prove your identity when connecting to an instance.

Name

aws-pcluster-ed25519-20250122_tokyo

The name can include up to 255 ASCII characters. It can't include leading or trailing spaces.

Key pair type [Info](#)

RSA

ED25519

Private key file format

.pem
For use with OpenSSH

.ppk
For use with PuTTY

Enter any name. In this example, use "aws-pcluster-ed25519-yyyyymmdd_Tokyo"
 ※ Replace "yyyyymmdd" with the creation date.

Select "ED25519".

Select ".pem".

Network Configuration

10. Creating Key Pair (3/3)

10.4 Configure the parameters as follows, then click "Create key pair". After clicking "Create Key Pair", the generated private key will be downloaded automatically.



Click "Create key pair".

Cancel

Create key pair

10.5 The key pair has been successfully created when the following message appears on the screen.



✓ Successfully created key pair

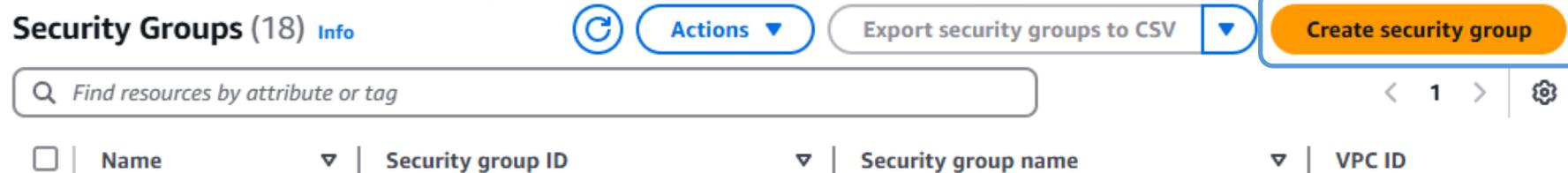
Network Configuration

11. Creating Security Group (1/4)

11.1 In the AWS Management Console, navigate to the Security Groups list page.

<https://ap-northeast-1.console.aws.amazon.com/ec2/home?region=ap-northeast-1#SecurityGroups>

11.2 Click "Create security group".



Security Groups (18) [Info](#)  **Actions**  Export security groups to CSV  **Create security group**

< 1 > 

<input type="checkbox"/>	Name	Security group ID	Security group name	VPC ID
--------------------------	------	-------------------	---------------------	--------

Click "Create security group".

Network Configuration

11. Creating Security Group (2/4)

11.3 Configure the parameters as follows.

Create security group Info

A security group acts as a virtual firewall for your instance to control inbound and outbound traffic. To create a new security group, click **Create Security Group**.

Basic details

Security group name Info

Name cannot be edited after creation.

Description Info

VPC Info

Inbound rules Info

Type <small>Info</small>	Protocol <small>Info</small>	Port range <small>Info</small>	Source <small>Info</small>	Description - optional <small>Info</small>	<small>Delete</small>
SSH	TCP	22	Anywhere-IPv4 0.0.0.0/0		<small>Delete</small>

Add rule

Enter any name. In this example, use "pcluster-prod-admin-sg".

Enter any description. In this example, use "For ParallelCluster Admin Server".

Select the VPC you created. In this example, select "pcluster-prod-vpc".

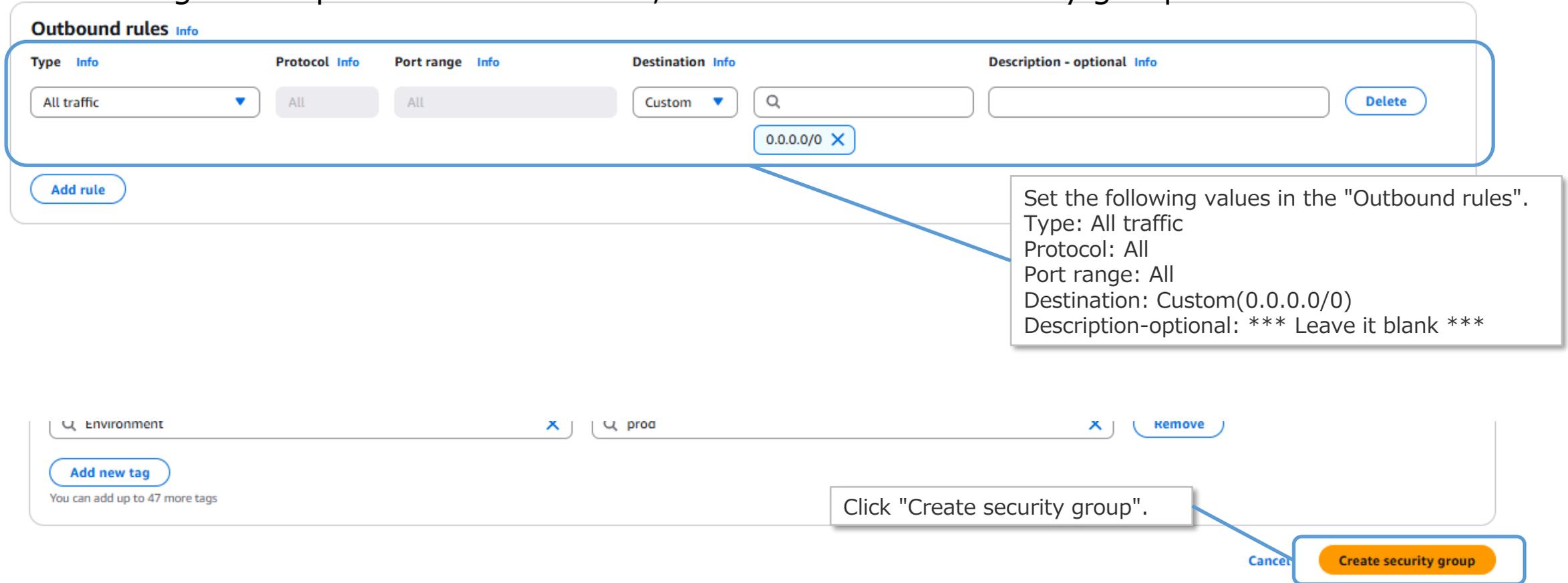
Custom
Anywhere-IPv4 ✓
Anywhere-IPv6
My IP
Anywh...

Set the following values in the "Inbound rules" section.
 Type: SSH
 Protocol: TCP
 Port range: 22
 Source: Anywhere-IPv4(0.0.0.0/0)
 Description-optional: *** Leave it blank ***

Network Configuration

11. Creating Security Group (3/4)

11.4 Configure the parameters as shown, then click "Create security group".



Outbound rules Info

Type <small>Info</small>	Protocol <small>Info</small>	Port range <small>Info</small>	Destination <small>Info</small>	Description - optional <small>Info</small>
All traffic	All	All	Custom 0.0.0.0/0	

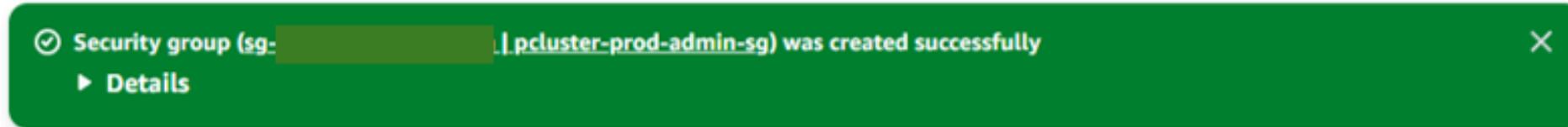
Set the following values in the "Outbound rules".
 Type: All traffic
 Protocol: All
 Port range: All
 Destination: Custom(0.0.0.0/0)
 Description-optional: *** Leave it blank ***

Click "Create security group".

Network Configuration

11. Creating Security Group (4/4)

11.5 The creation of the security group is complete when the following message appears on the screen.



Creating ParallelCluster Setup Instance

This section explains the procedure for creating a ParallelCluster setup instance. In this manual, this instance is used to set up a virtual environment, install AWS ParallelCluster, set up the cluster, and log into the cluster's Head Node. Follow the steps below to create the instance. Each step is explained in detail on the following pages.

1. Creating EC2 Instance
2. Allocating Elastic IP Address
3. Associating Elastic IP Address
4. Creating Access Key
5. Logging into the ParallelCluster Setup Instance

Creating ParallelCluster Setup Instance

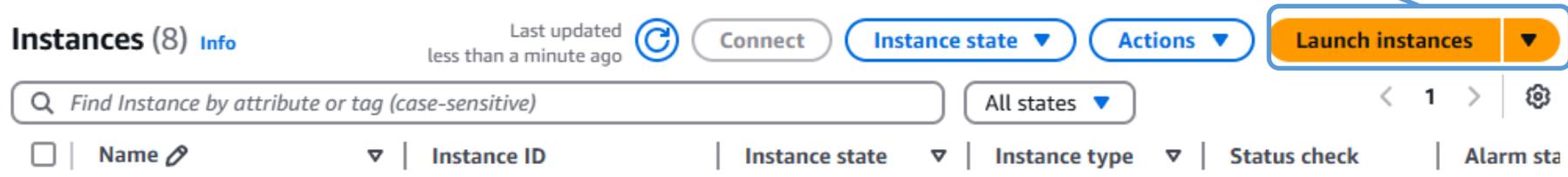
1. Creating EC2 Instance (1/8)

1.1 In the AWS Management Console, navigate to the Security Groups List page.

<https://ap-northeast-1.console.aws.amazon.com/ec2/home?region=ap-northeast-1#Instances>

1.2 Click "Launch instances" in the upper right corner.

Click "Launch instances".



The screenshot shows the AWS Management Console interface for the EC2 Instances page. At the top, it displays 'Instances (8)' with an 'Info' link. Below this, there are several action buttons: 'Connect', 'Instance state' (with a dropdown arrow), 'Actions' (with a dropdown arrow), and 'Launch instances' (highlighted in orange). A callout box with the text 'Click "Launch instances"' has a blue arrow pointing to the 'Launch instances' button. Below the buttons, there is a search bar with the placeholder text 'Find Instance by attribute or tag (case-sensitive)', a filter dropdown set to 'All states', and pagination controls showing '1' of 1 page. At the bottom, the start of a table is visible with columns for 'Name', 'Instance ID', 'Instance state', 'Instance type', 'Status check', and 'Alarm sta'.

Creating ParallelCluster Setup Instance

1. Creating EC2 Instance (2/8)

1.3 Configure the parameters as follows.

Launch an instance [Info](#)

Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Follow these simple steps below.

Enter a name of your choice. Here, "pcluster-prod-admin-1a-01" is used as an example.

Name and tags [Info](#)

Name

[Add additional tags](#)

Creating ParallelCluster Setup Instance

1. Creating EC2 Instance (3/8)

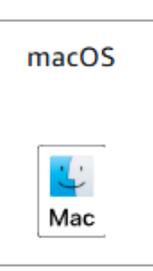
1.4 Configure the parameters as follows.

▼ **Application and OS Images (Amazon Machine Image)** [Info](#)

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below

🔍 Search our full catalog including 1000s of application and OS images

Recents | My AMIs | **Quick Start**

 <p>Amazon Linux</p> <p>aws</p>	 <p>macOS</p> <p>Mac</p>	 <p>Ubuntu</p> <p>ubuntu</p>	 <p>Windows</p> <p>Microsoft</p>	 <p>Red Hat</p> <p>Red Hat</p>	 <p>Browse more AMIs</p> <p>Including AMIs from AWS, Marketplace and the Community</p>
---	--	--	---	--	---

Click the "Amazon Linux" icon.

Creating ParallelCluster Setup Instance

1. Creating EC2 Instance (4/8)

1.5 Configure the parameters as follows.

Amazon Machine Image (AMI)

Amazon Linux 2023 AMI Free tier eligible
 ami-06c6f3fa7959e5fdd (64-bit (x86), uefi-preferred) / ami-0ffeb6c61663cf92e (64-bit (Arm), uefi)
 Virtualization: hvm ENA enabled: true Root device type: ebs

Description

Amazon Linux 2023 is a modern, general purpose Linux-based OS that comes with 5 years of long term support. It is optimized for AWS and designed to provide a secure, stable and high-performance execution environment to develop and run your cloud applications.

Amazon Linux 2023 AMI 2023.6.20250128.0 x86_64 HVM kernel-6.1

Architecture	Boot mode	AMI ID	Username
64-bit (x86)	uefi-preferred	ami-06c6f3fa7959e5fdd	ec2-user

Verified provider

Select "64-bit (x86)".

Creating ParallelCluster Setup Instance

1. Creating EC2 Instance (5/8)

1.6 Configure the parameters as follows.

▼ Instance type [Info](#) | [Get advice](#)

Instance type

t3.micro
Family: t3 2 vCPU 1 GiB Memory Current generation: true
On-Demand Windows **t3.micro** pricing: 0.0228 USD per Hour
On-Demand Linux base pricing: 0.0136 USD per Hour
On-Demand SUSE base pricing: 0.0136 USD per Hour
On-Demand RHEL base pricing: 0.0424 USD per Hour
On-Demand Ubuntu Pro base pricing: 0.0171 USD per Hour

Additional costs apply for AMIs with pre-installed software

Select "t3.micro".

All generations

[Compare instance types](#)

▼ Key pair (login) [Info](#)

You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.

Key pair name - required

aws-pcluster-ed25519-20250122_tokyo

Select the key pair you created earlier.

[Create new key pair](#)

Creating ParallelCluster Setup Instance

1. Creating EC2 Instance (6/8)

1.7 Configure the parameters as follows.

▼ **Network settings** [Info](#)

VPC - required | [Info](#)

vpc-10.0.0.0/16 | (pcluster-prod-vpc) ▼

Select the VPC you created. In this example, select "pcluster-prod-vpc".

Subnet | [Info](#)

subnet- | pcluster-prod-public-subnet-1a ▼

Select the subnet you created. In this example, select "cluster-prod-public-subnet-1a".

Auto-assign public IP | [Info](#)

Enable ▼

Select "Enable".

Firewall (security groups) | [Info](#)

A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to r

Create security group Select existing security group

Select "Select existing security group".

Common security groups | [Info](#)

Select security groups ▼

pcluster-prod-admin-sg sg- VPC: vpi ✕

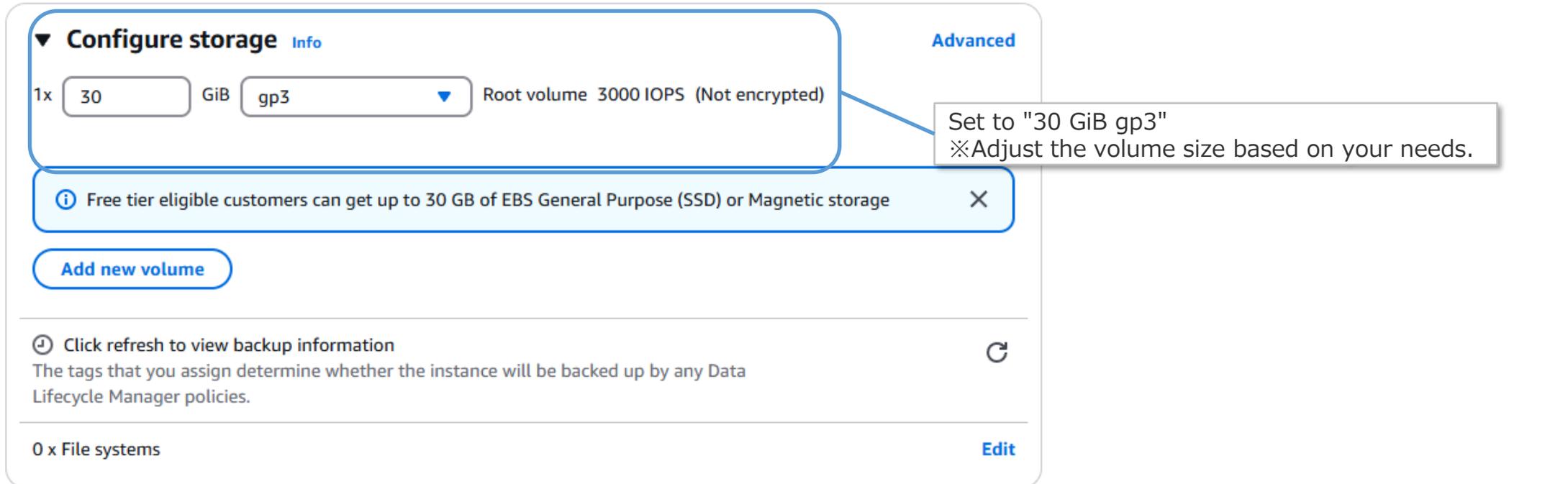
Select the security group you created. In this example, select "pcluster-prod-admin-sg".

Security groups that you add or remove here will be added to or removed from all your network interfaces.

Creating ParallelCluster Setup Instance

1. Creating EC2 Instance (7/8)

1.8 Configure the parameters as follows.



Configure storage [Info](#) Advanced

1x GiB Root volume 3000 IOPS (Not encrypted)

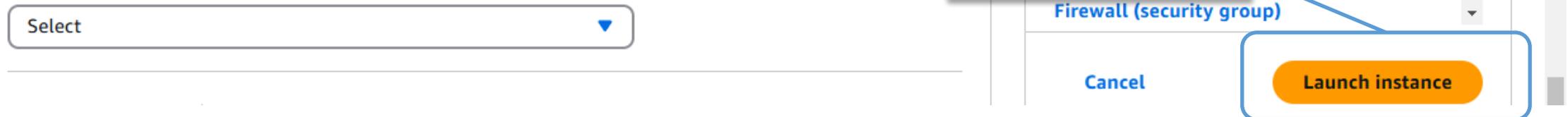
Free tier eligible customers can get up to 30 GB of EBS General Purpose (SSD) or Magnetic storage

[Add new volume](#)

[Click refresh to view backup information](#)
The tags that you assign determine whether the instance will be backed up by any Data Lifecycle Manager policies.

0 x File systems Edit

1.9 Click "Launch instance" on the right side.



Select

[Cancel](#) [Launch instance](#)

Creating ParallelCluster Setup Instance

1. Creating EC2 Instance (8/8)

1.10 Return to the EC2 Instances page and wait a few minutes. Then, verify that the EC2 instance being created has passed the '3/3 status checks.



The screenshot shows the AWS Management Console 'Instances' page. At the top, there are buttons for 'Connect', 'Instance state', 'Actions', and 'Launch instances'. Below these is a search bar and a filter dropdown set to 'All states'. The main table lists instances with columns for Name, Instance ID, Instance state, Instance type, Status check, and Actions. One instance, 'pcluster-prod-admin-1a-01', is highlighted in blue. Its status is 'Running' and its status check is '3/3 checks passed'. A blue box highlights the 'Status check' column for this instance, and a callout box points to it with the text: 'Verify that the "Status Check" displays '3/3 checks passed.'

Name	Instance ID	Instance state	Instanc...	Status check	AL
pcluster-prod-admin-1a-01		Running	t3.micro	3/3 checks passed	Vi

Verify that the "Status Check" displays '3/3 checks passed.

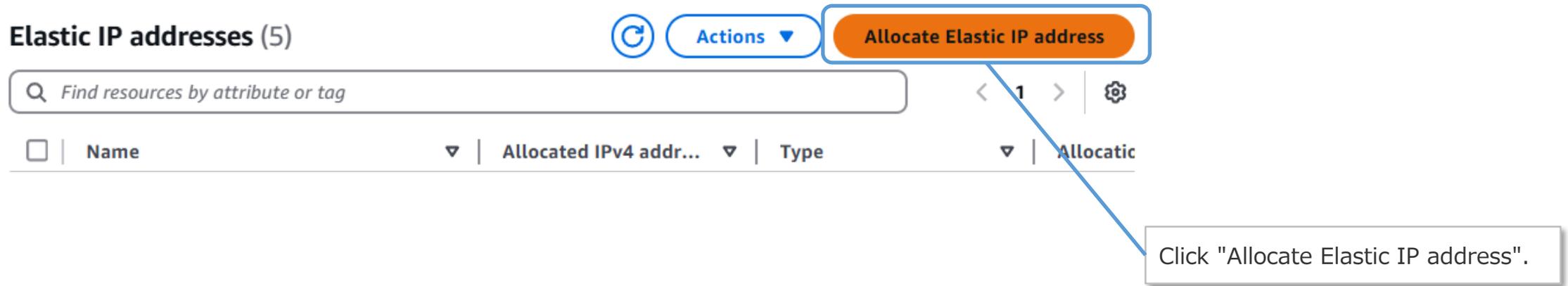
Creating ParallelCluster Setup Instance

2. Allocating Elastic IP Address (1/3)

2.1 To allocate an Elastic IP address, navigate to the Elastic IPs page.

<https://ap-northeast-1.console.aws.amazon.com/ec2/home?region=ap-northeast-1#Addresses>

2.2 Click "Allocate Elastic IP address" in the upper right corner.



The screenshot shows the AWS Elastic IP addresses console. At the top, there is a header "Elastic IP addresses (5)" with a refresh icon and an "Actions" dropdown menu. The "Allocate Elastic IP address" button is highlighted with a blue box. Below the header is a search bar with the placeholder text "Find resources by attribute or tag". To the right of the search bar are navigation controls: a left arrow, the number "1", a right arrow, and a settings gear icon. Below the search bar is a table header with columns: "Name", "Allocated IPv4 addr...", "Type", and "Allocatic". A blue arrow points from the "Allocate Elastic IP address" button to a callout box that says "Click 'Allocate Elastic IP address'".

Creating ParallelCluster Setup Instance

2. Allocating Elastic IP Address (2/3)

2.3 Configure the parameters as follows.

Allocate Elastic IP address [Info](#)

Elastic IP address settings [Info](#)

Public IPv4 address pool

- Amazon's pool of IPv4 addresses
- Public IPv4 address that you bring to your AWS account with BYOIP. (option disabled because no pools found) [Learn more](#)
- Customer-owned pool of IPv4 addresses created from your on-premises network for use with an Outpost. (option disabled because no customer owned pools found) [Learn more](#)
- Allocate using an IPv4 IPAM pool (option disabled because no public IPv4 IPAM pools with AWS service as EC2 were found)

Network border group [Info](#)

ap-northeast-1

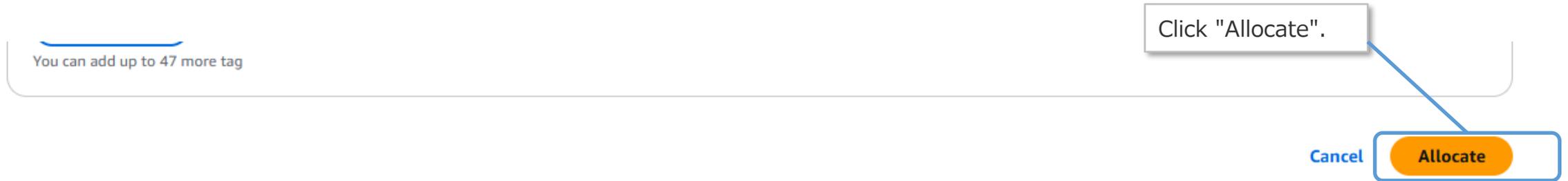
Select "Amazon's pool of IPv4 addresses".

Select "ap-northeast-1".

Creating ParallelCluster Setup Instance

2. Allocating Elastic IP Address (3/3)

2.4 Click "Allocate" in the lower right corner.



2.5 The Elastic IP address has been successfully allocated when the following message appears on the screen.



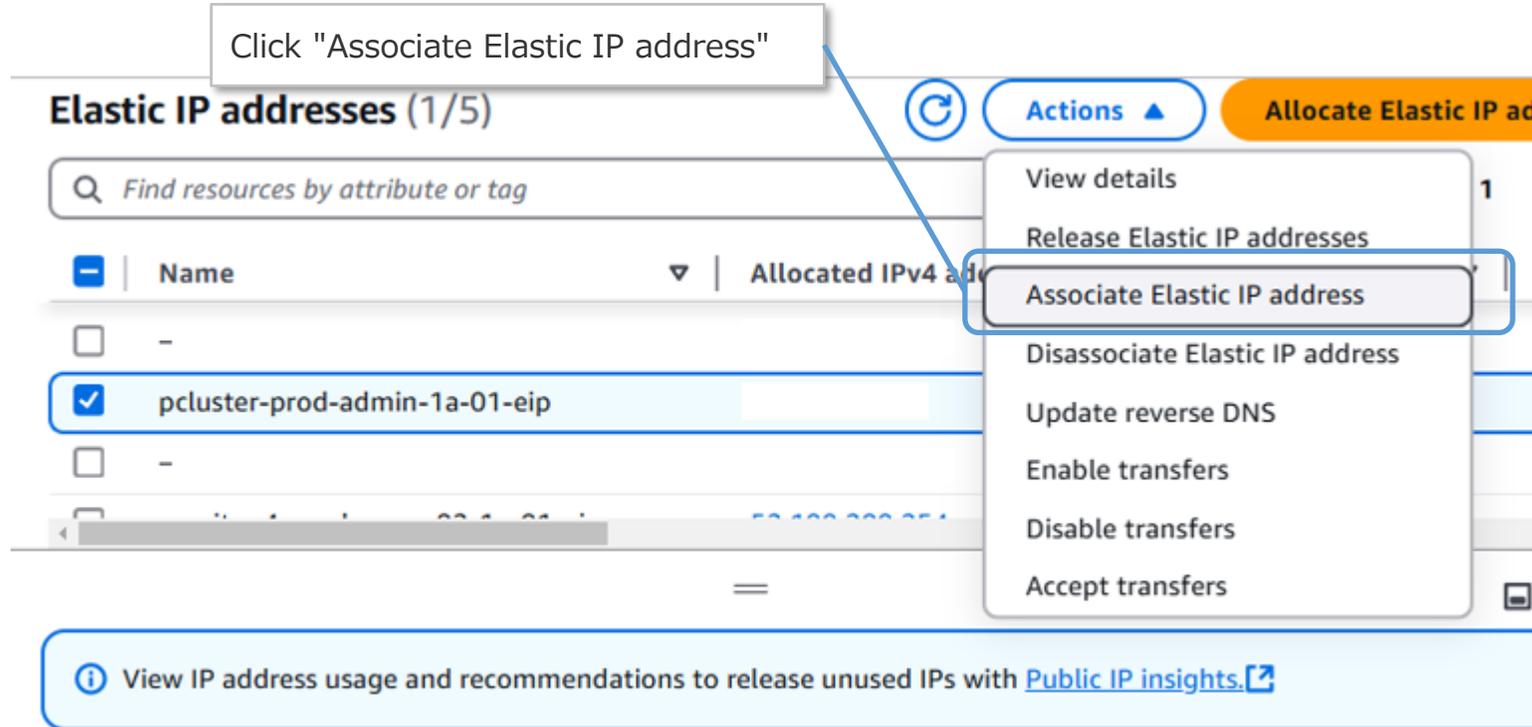
Creating ParallelCluster Setup Instance

3. Associating Elastic IP Address (1/2)

3.1 In the AWS Management Console, navigate to the EIP list page.

<https://ap-northeast-1.console.aws.amazon.com/ec2/home?region=ap-northeast-1#Addresses>

3.2 Select the EIP to assign to the EC2 instance, then click "Actions" in the top right corner and choose "Associate Elastic IP address".



Click "Associate Elastic IP address"

Elastic IP addresses (1/5)

Find resources by attribute or tag

Name	Allocated IPv4 address
-	
<input checked="" type="checkbox"/> pcluster-prod-admin-1a-01-eip	
-	

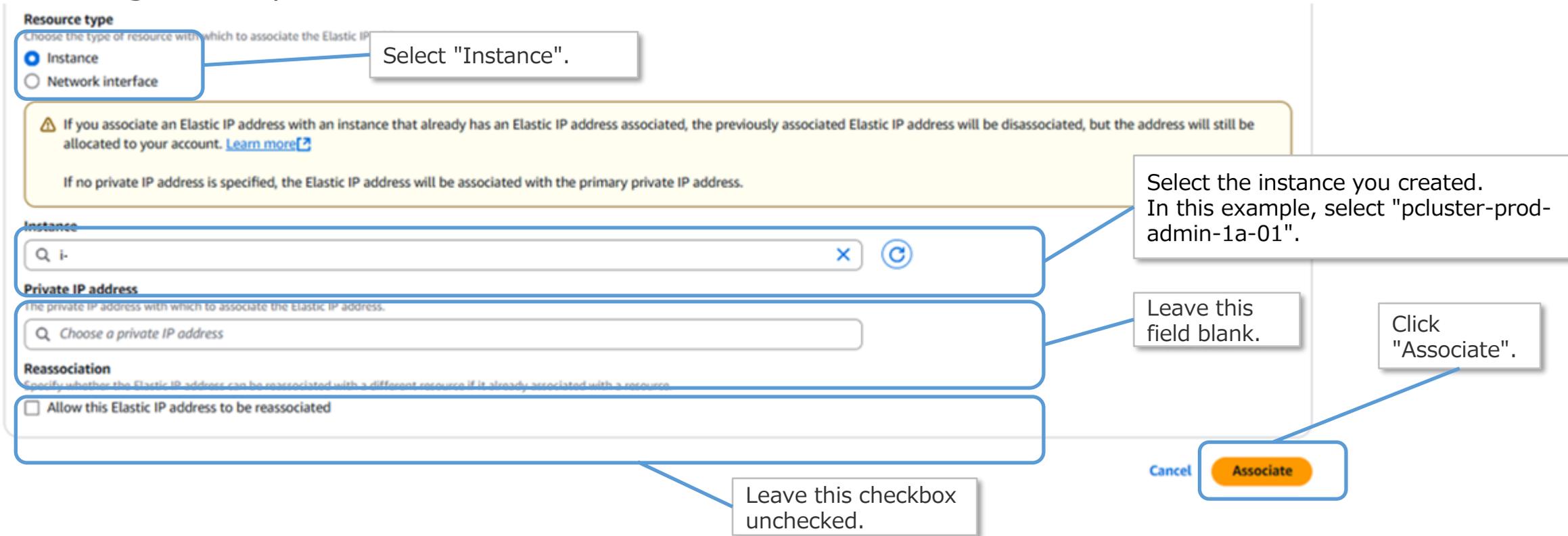
- View details
- Release Elastic IP addresses
- Associate Elastic IP address**
- Disassociate Elastic IP address
- Update reverse DNS
- Enable transfers
- Disable transfers
- Accept transfers

View IP address usage and recommendations to release unused IPs with [Public IP insights](#).

Creating ParallelCluster Setup Instance

3. Associating Elastic IP Address (2/2)

3.3 Configure the parameters as shown below, then click "Associate".



Resource type
Choose the type of resource with which to associate the Elastic IP address.

Instance Select "Instance".

Network interface

Instance
Select the instance you created. In this example, select "pcluster-prod-admin-1a-01".

Private IP address
The private IP address with which to associate the Elastic IP address.
Leave this field blank.

Reassociation
Specify whether the Elastic IP address can be reassociated with a different resource if it already associated with a resource.

Allow this Elastic IP address to be reassociated Leave this checkbox unchecked.

Cancel Associate Click "Associate".

3.4 The Elastic IP address is successfully associated when the following message appears on the screen.



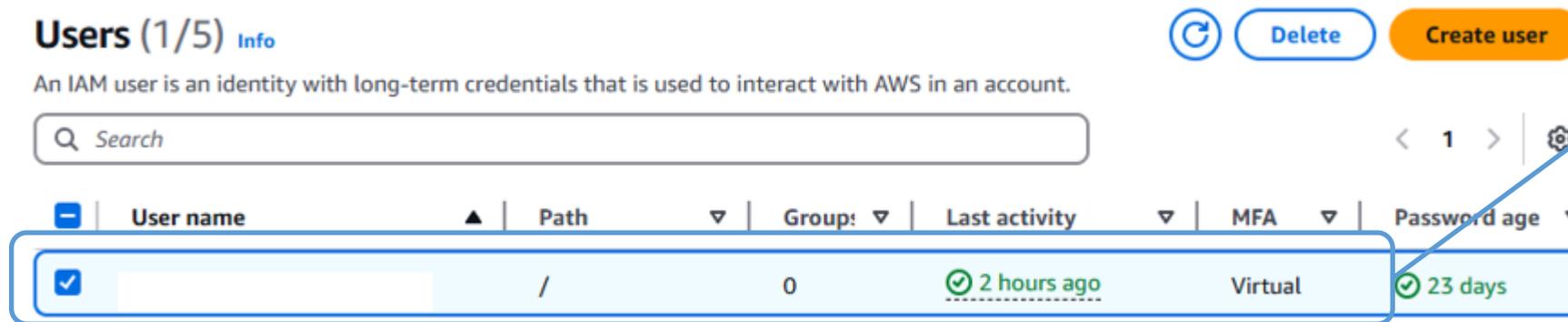
Creating ParallelCluster Setup Instance

4. Creating Access Key (1/4)

4.1 In the AWS Management Console, navigate to the IAM Users page.

<https://us-east-1.console.aws.amazon.com/iam/home?region=ap-northeast-1#/users>

4.2 Click the username of the "IAM user" you use to log in to the AWS Management Console.



Users (1/5) [Info](#)

An IAM user is an identity with long-term credentials that is used to interact with AWS in an account.

Search

Refresh Delete Create user

<input type="checkbox"/>	User name	Path	Group	Last activity	MFA	Password age
<input checked="" type="checkbox"/>	[Redacted]	/	0	2 hours ago	Virtual	23 days

Click the username of the "IAM user" you use to log in.

Creating ParallelCluster Setup Instance

4. Creating Access Key (2/4)

4.3 Click "Security credentials".



The screenshot shows the 'Security credentials' tab selected in the AWS IAM console. The 'Summary' section is empty. Below it, there are tabs for 'Permissions', 'Groups', 'Tags', 'Security credentials', and 'Last Accessed'. The 'Security credentials' tab is highlighted with a blue border. A callout box points to this tab with the text 'Click "Security credentials".'

4.4 Click "Create access key".



The screenshot shows the 'Access keys (1)' section in the AWS IAM console. It includes a description: 'Use access keys to send programmatic calls to AWS from the AWS CLI, AWS Tools for PowerShell, AWS SDKs, or direct AWS API calls. You can have a maximum of two access keys (active or inactive) at a time. [Learn more](#)'. A 'Create access key' button is highlighted with a blue border and a callout box pointing to it with the text 'Click "Create access key"'. Below the description, there is a table with columns for 'Description' and 'Status'. The 'Status' column shows a green checkmark and the word 'Active'. An 'Actions' dropdown menu is visible in the top right corner of the table area.

Creating ParallelCluster Setup Instance

4. Creating Access Key (3/4)

4.5 Select "Command Line Interface(CLI)".

Access key best practices & alternatives [Info](#)

Avoid using long-term credentials like access keys to improve your security. Consider the following use cases and alternatives.

Use case

- Command Line Interface (CLI)**
You plan to use this access key to enable the AWS CLI to access your AWS account.

Select "Command Line Interface(CLI)".

4.6 Check "I understand the above recommendations and want to proceed to create an access key", then click "Next".

Confirmation

- I understand the above recommendation and want to proceed to create an access key.

Check "I understand the above recommendations and want to proceed to create an access key".

Cancel

Next

Click "Next".

Creating ParallelCluster Setup Instance

4. Creating Access Key (4/4)

4.7 Click "Create access key".

Cancel

Previous

Create access key

Click "Create access key".

4.8 Take note of your "Access Key" and "Secret access key". Since the "Secret access key" is masked, click "Show" to reveal it, then record it securely.

Access key

If you lose or forget your secret access key, you cannot retrieve it. Instead, create a new access key and make the old key inactive.

Access key

Secret access key



Show

Take note of the "Access key" and "Secret access key".

4.9 Click "Done".

Access key best practices

- Never store your access key in plain text, in a code repository, or in code.
- Disable or delete access key when no longer needed.
- Enable least-privilege permissions.
- Rotate access keys regularly.

For more details about managing access keys, see the [best practices for managing AWS access keys](#).

Click "Done".

Download .csv file

Done

Setup of ParallelCluster

This section provides instructions for setting up AWS ParallelCluster. Follow the steps below to install and configure AWS ParallelCluster. All operations should be performed on the ParallelCluster setup instance. A detailed explanation of each step is provided on the following pages.

1. Installing AWS ParallelCluster
2. Cluster Setup

Setup of ParallelCluster

1. Installing AWS ParallelCluster (1/2)

1.1 Installing pip

Log in to the ParallelCluster setup instance and run the following command to install pip.

```
$ curl -O https://bootstrap.pypa.io/get-pip.py
$ python3 get-pip.py
```

Run the following command. If the version information appears, the installation was successful.

```
$ pip --version
-----
pip 24.3.1 from /home/ec2-user/apc-ve/lib/python3.9/site-packages/pip (python 3.9)
-----
```

1.2 Installing virtualenv

Run the following command to install virtualenv, a library for creating Python virtual environments.

```
$ python3 -m pip install --upgrade pip
$ python3 -m pip install --user --upgrade virtualenv
```

1.3 Creating a Virtual Environment

Run the following command to create a virtual environment

```
$ python3 -m virtualenv ~/apc-ve
```

1.4 Activating a Virtual Environment

Run the following command to activate the virtual environment.

```
$ source ~/apc-ve/bin/activate
```

Setup of ParallelCluster

1. Installing AWS ParallelCluster (2/2)

1.5 Installing ParallelCluster

Run the following command to install AWS ParallelCluster using pip.

```
(apc-ve) $ pip3 install aws-parallelcluster==3.10.1
```

Run the following command. If the version information appears, the installation was successful.

```
(apc-ve) $ pcluster version
```

```
-----  
{  
"version": "3.10.1"  
}  
-----
```

1.6 Installing Node.js

Run the following command to install Node.js on your system.

```
(apc-ve) $ curl -o- https://raw.githubusercontent.com/nvm-sh/nvm/v0.38.0/install.sh | bash  
(apc-ve) $ chmod ug+x ~/.nvm/nvm.sh  
(apc-ve) $ source ~/.nvm/nvm.sh  
(apc-ve) $ nvm install --lts
```

Run the following command. If the version information appears, the installation was successful.

```
(apc-ve) $ node --version
```

```
-----  
{  
v22.13.0  
}  
-----
```

Setup of ParallelCluster

2. Cluster Setup (1/4)

2.1 Configuring AWS Credentials

Run the following command and enter the required credentials.

```
$ aws configure
-----
AWS Access Key ID [None]: {Enter your Access Key}
AWS Secret Access Key [None]: {Enter your Secret Access Key}
Default region name [None]: ap-northeast-1
Default output format [None]: {Press Enter without providing any input.}
-----
```

※Enter the Access Key and Secret Access Key recorded in the "[Creating an Access Key](#)" section.

Setup of ParallelCluster

2. Cluster Setup (2/4)

2.2 Running the Configuration Creation Command

Run the following command.

```
(apc-ve)$ mkdir ~/pcluster-config  
(apc-ve)$ pcluster configure --config ~/pcluster-config/cluster-config-01.yaml
```

2.3 Entering Parameter Information

After running the configuration creation command, prompts will appear. Enter the required parameter values.

(1) Selecting Region: Select "ap-northeast-1".

```
AWS Region ID [ap-northeast-1]:ap-northeast-1
```

(2) Selecting Key Pair: Select the key pair created in "[Creating a Key Pair](#)" section.

```
EC2 Key Pair Name [aws-pcluster-ed25519-20250115_tokyo]:{Select the Key Pair for the Node}
```

(3) Selecting Scheduler: Select Slurm.

```
Allowed values for Scheduler:  
1. slurm  
2. awsbatch  
Scheduler [slurm]: slurm
```

Setup of ParallelCluster

2. Cluster Setup (3/4)

(4) Selecting OS: Select rhel8

```
Allowed values for Operating System:
1. alinux2
2. alinux2023
3. ubuntu2004
4. ubuntu2204
5. rhel8
6. rocky8
7. rhel9
8. rocky9 Operating System [alinux2]: rhel8
```

(5) Selecting the Head Node Instance Type:

Configure an instance based on Graviton3/3E. In this example, "m7g.medium" is selected.

```
Head node instance type [t2.micro]: m7g.medium Configure it based on your use case.
```

(6) Selecting the Queue Configuration:

Configure the queue based on your use case. (The following is an example command input.)

```
Number of queues [1]: 1 ... Setting the Number of Queues ※Set the number of queues. In Slurm,
compute nodes can be grouped into logical units called queues.
Name of queue 1 [queue1]: ... queue1 Set the queue name to "queue1".
[queue1]: queue1 Number of compute resources for queue1 [1]: 1 ... Set the number of compute resources allocated to "queue1".
Compute instance type for compute resource 1 in queue1 [t2.micro]: m7g.2xlarge ... Set the instance type for the compute nodes.
Maximum instance count [10]: 10 ... Set the maximum number of nodes available.
In this example, up to 10 nodes can be used for computation.
```

For the compute instance type, select an instance based on Graviton3/3E. In this example, "m7g.2xlarge" is selected. Estimate the required specifications for the Compute Instance Type based on your needs, and choose one from the following URL https://aws.amazon.com/ec2/instance-types/?nc1=h_ls

Setup of ParallelCluster

2. Cluster Setup (4/4)

(7) Automate VPC Creation: Select "y"

```
Automate VPC creation? (y/n) [n]: y
```

(8) Select Availability Zone: Select "ap-northeast-1a"

```
Allowed values for Availability Zone:
```

1. ap-northeast-1a
2. ap-northeast-1c
3. ap-northeast-1d Availability Zone [ap-northeast-1a]: ap-northeast-1a

(9) Select Network Configuration: Select "1"

```
Allowed values for Network Configuration:
```

1. Head node in a public subnet and compute fleet in a private subnet
2. Head node and compute fleet in the same public subnet

```
Network Configuration [Head node in a public subnet and compute fleet in a private subnet]: Head node in a public subnet and compute fleet in a private subnet
```

```
Network Configuration [Head node in a public subnet and compute fleet in a private subnet]: 1
```

After completing the input, ParallelCluster's VPC and network settings are automatically created. The process takes a few minutes, after which you can enter commands.

Creating External Storage

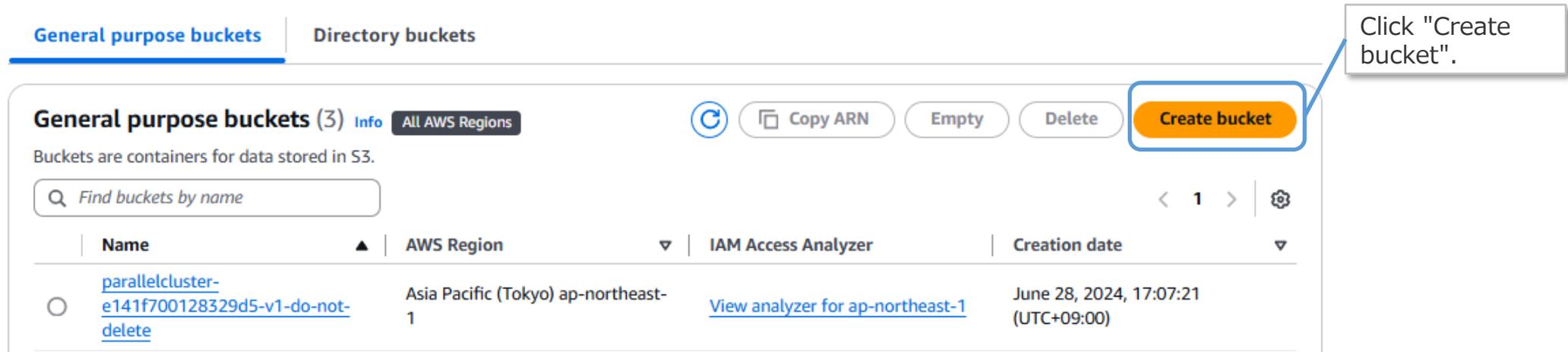
In the cluster environment set up according to this manual, an S3 bucket serves as long-term storage for computational results and other data. This section explains how to create an S3 bucket.

1. Creating S3 Bucket (1/4)

1.1 In the AWS Management Console, navigate to the S3 bucket list page.

<https://ap-northeast-1.console.aws.amazon.com/s3/buckets?region=ap-northeast-1&bucketType=general>

1.2 Click "Create bucket".



The screenshot shows the AWS Management Console interface for S3 buckets. The 'General purpose buckets' tab is selected. At the top right, there are several action buttons: 'Copy ARN', 'Empty', 'Delete', and 'Create bucket'. The 'Create bucket' button is highlighted with a blue box, and a callout bubble points to it with the text 'Click "Create bucket"'. Below the buttons, there is a search bar and a table of buckets. The table has columns for Name, AWS Region, IAM Access Analyzer, and Creation date. One bucket is listed with the name 'parallelcluster-e141f700128329d5-v1-do-not-delete', located in the 'Asia Pacific (Tokyo) ap-northeast-1' region, with a creation date of 'June 28, 2024, 17:07:21 (UTC+09:00)'. There is also a 'View analyzer for ap-northeast-1' link in the IAM Access Analyzer column.

Creating External Storage

1. Creating S3 Bucket (2/4)

1.3 Configure the parameters as follows.

Create bucket Info

Buckets are containers for data stored in S3.

General configuration

AWS Region
Asia Pacific (Tokyo) ap-northeast-1

Bucket type Info

General purpose
Recommended for most use cases and access patterns. General purpose buckets are the original S3 bucket type. They allow a mix of storage classes that redundantly store objects across multiple Availability Zones.

Directory
Recommended for low-latency use cases. These buckets use only the S3 Express One Zone storage class, which provides faster processing of data within a single Availability Zone.

Select "General purpose".

Bucket name Info

pcluster-prod-lustre-

Bucket name must be unique within the global namespace and follow the bucket naming rules. [See rules for bucket naming](#)

Enter any name. In this example, use "pcluster-prod-lustre-`{Account ID}`".

Copy settings from existing bucket - optional
Only the bucket settings in the following configuration are copied.

[Choose bucket](#)

Object Ownership Info

Control ownership of objects written to this bucket from other AWS accounts and the use of access control lists (ACLs). Object ownership determines who can specify access to objects.

ACLs disabled (recommended)
All objects in this bucket are owned by this account. Access to this bucket and its objects is specified using only policies.

ACLs enabled
Objects in this bucket can be owned by other AWS accounts. Access to this bucket and its objects can be specified using ACLs.

Select "ACLs disabled".

Object Ownership
Bucket owner enforced

Creating External Storage

1. Creating S3 Bucket (3/4)

1.4 Configure the parameters as follows.

Block Public Access settings for this bucket

Public access is granted to buckets and objects through access control lists (ACLs) and bucket policies. To ensure that public access to this bucket and its objects is blocked, turn on Block all public access for this bucket and its access points. AWS recommends that you turn on Block all public access to ensure that your applications will work correctly without public access to objects within, you can customize the individual settings below to suit your specific storage use cases. [Learn more](#)

Select the checkbox for "Block all public access".

Block all public access

Turning this setting on is the same as turning on all four settings below. Each of the following settings are independent of one another.

Block public access to buckets and objects granted through *new* access control lists (ACLs)

S3 will block public access permissions applied to newly added buckets or objects, and prevent the creation of new public access ACLs for existing buckets and objects. This setting doesn't change any existing permissions that allow public access to S3 resources using ACLs.

Block public access to buckets and objects granted through *any* access control lists (ACLs)

S3 will ignore all ACLs that grant public access to buckets and objects.

Block public access to buckets and objects granted through *new* public bucket or access point policies

S3 will block new bucket and access point policies that grant public access to buckets and objects. This setting doesn't change any existing policies that allow public access to S3 resources.

Block public and cross-account access to buckets and objects through *any* public bucket or access point policies

S3 will ignore public and cross-account access for buckets or access points with policies that grant public access to buckets and objects.

Bucket Versioning

Versioning is a means of keeping multiple variants of an object in the same bucket. You can use versioning to preserve, retrieve, and restore every version of every object stored in your Amazon S3 bucket. With versioning, you can easily recover from both unintended user actions and application failures. [Learn more](#)

Bucket Versioning

Disable

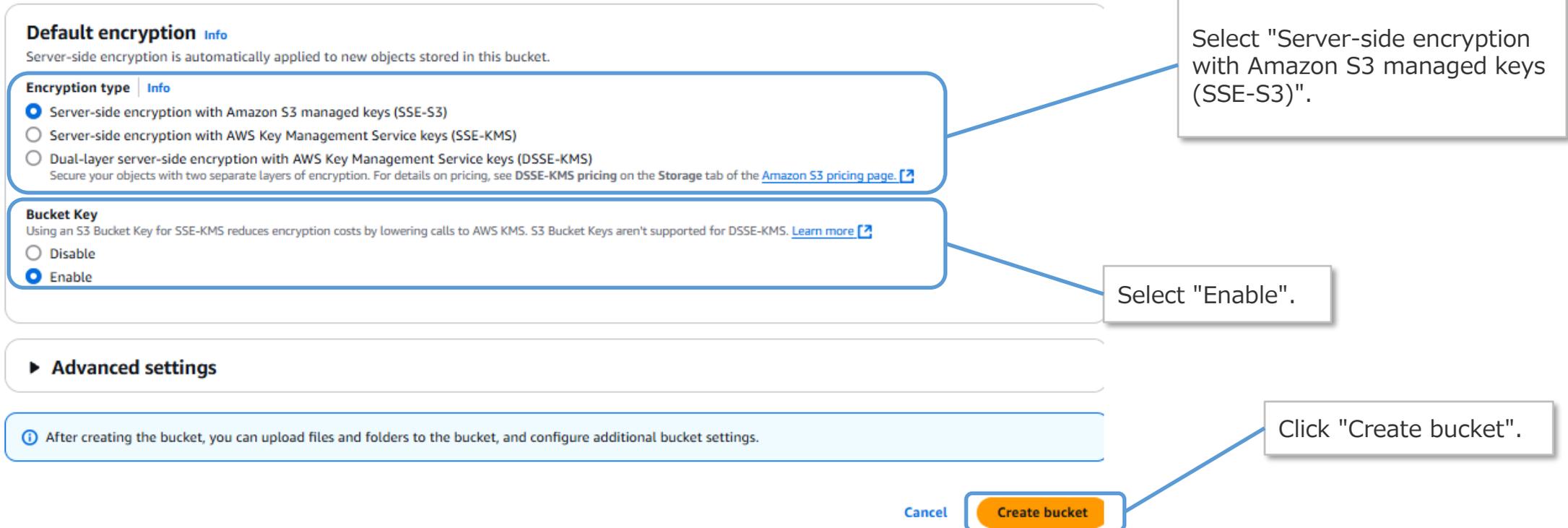
Enable

Configure this setting as needed. In this example, select the "Disable" option.

Creating External Storage

1. Creating S3 Bucket (4/4)

1.5 Configure the parameters as shown below, then click "Create bucket".



Default encryption Info
Server-side encryption is automatically applied to new objects stored in this bucket.

Encryption type Info

- Server-side encryption with Amazon S3 managed keys (SSE-S3)
- Server-side encryption with AWS Key Management Service keys (SSE-KMS)
- Dual-layer server-side encryption with AWS Key Management Service keys (DSSE-KMS)
Secure your objects with two separate layers of encryption. For details on pricing, see [DSSE-KMS pricing on the Storage tab of the Amazon S3 pricing page.](#)

Bucket Key
Using an S3 Bucket Key for SSE-KMS reduces encryption costs by lowering calls to AWS KMS. S3 Bucket Keys aren't supported for DSSE-KMS. [Learn more](#)

- Disable
- Enable

Advanced settings

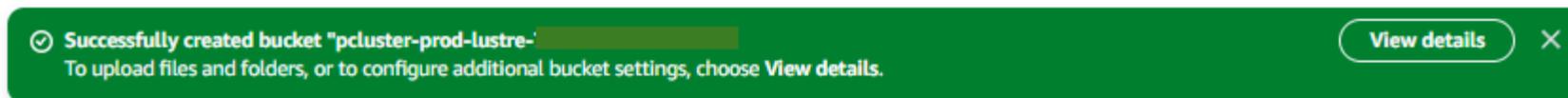
After creating the bucket, you can upload files and folders to the bucket, and configure additional bucket settings.

[Cancel](#) [Create bucket](#)

Callouts:

- Select "Server-side encryption with Amazon S3 managed keys (SSE-S3)".
- Select "Enable".
- Click "Create bucket".

1.6 The following screen confirms that the S3 bucket has been successfully created, marking the completion of the external storage setup.



Successfully created bucket "pcluster-prod-lustre-XXXXXXXXXX
To upload files and folders, or to configure additional bucket settings, choose [View details](#)

Setup of Cluster

This section explains the process of create a cluster. After setting up AWS ParallelCluster and configuring external storage, you can deploy the cluster. Follow the steps below to set up the cluster on the ParallelCluster setup instance. Each step of the cluster setup is explained in detail on the following pages.

1. Configuration of cluster-config-01.yaml
2. Creating Cluster
3. Logging in to the Head Node

Setup of Cluster

1. Configuration of cluster-config-01.yaml

1.1 Add the red-highlighted sections below to the cluster-config-01.yaml file, which is generated in the pcluster-config directory. This file contains the configuration settings specified in "ParallelCluster Setup".

<pre> Region: ap-northeast-1 Image: Os: rhel8 HeadNode: InstanceType: m7g.medium Networking: SubnetId: subnet-0aada1ac478ba9f42 LocalStorage: RootVolume: Size: 64 Encrypted: true VolumeType: gp2 DeleteOnTermination: true Ssh: KeyName: aws-pcluster-ed25519-20250 Scheduling: Scheduler: slurm SlurmQueues: - Name: queue1 ComputeResources: - Name: m7g2xlarge Instances: - InstanceType: m7g.2xlarge MinCount: 0 MaxCount: 10 Networking: SubnetIds: - subnet-0747376860b49e9b6 </pre>	<pre> SharedStorage: - MountDir: /lustre01 Name: lustre01 StorageType: FsxLustre FsxLustreSettings: ImportedFileChunkSize: 1024 DeploymentType: PERSISTENT_1 PerUnitStorageThroughput: 100 StorageCapacity: 1200 AutoImportPolicy: NEW_CHANGED_DELETED ExportPath: s3://pcluster-prod-lustre-{Account ID} ImportPath: s3://pcluster-prod-lustre-{Account ID} </pre>	<p>Configure the storage size as needed.</p> <p>Configure storage for the Head Node. Set the storage size to at least 64GB to accommodate downloading the "Virtual Fugaku" container.</p> <p>Configure Shared Storage</p> <p>Specify the name of the S3 bucket that you created. Example: pcluster-prod-lustre-{Account ID}</p> <p>Configuring shared storage in this file will automatically set it up and integrate it with the cluster.</p>
--	---	--

Setup of Cluster

2. Creating Cluster

2.1 Running the Cluster Creation Command

Run the following command to create a cluster. The creation process takes approximately 20 minutes, as FSx for Lustre and other additional components are included.

```
(apc-ve)$ pcluster create-cluster --cluster-name cluster01 --cluster-configuration ~/pcluster-config/cluster-config-01.yaml
```

2.2 Checking Cluster Status

Run the following command to check the cluster creation status. If "cloudFormationStackStatus" is "CREATE_IN_PROGRESS", the creation process is still ongoing. If it shows "CREATE_COMPLETE", the cluster creation is complete.

```
(apc-ve)$ pcluster describe-cluster --cluster-name cluster01 |grep "cloudFormationStackStatus"
```

Setup of Cluster

3. Logging in to the Head Node (1/2)

To log in to the Head Node, first access the ParallelCluster setup instance and then initiate the login process.

3.1 Activating the Virtual Environment

Log in to the "ParallelCluster Setup Instance", and execute the following command.

```
$ source ~/apc-ve/bin/activate
```

3.2 Saving the Private Key

Run the following command, paste the private key string of the key pair, create the private key file, and save it to the ParallelCluster setup instance.

```
(apc-ve)$ vi ~/.ssh/aws-pcluster-ed25519-20250115_tokyo.pem  
***** Paste the private key string of the key pair. *****
```

Setup of Cluster

3. Logging in to the Head Node (2/2)

3.3 Changing Private Key Permissions

Run the following command to change the private key permissions.

```
(apc-ve)$ chmod 600 ~/.ssh/aws-pcluster-ed25519-20250115_tokyo.pem
```

3.4 Connecting to the Head Node via SSH

Run the following command to establish an SSH connection to the Head Node.

```
(apc-ve)$ pcluster ssh --cluster-name cluster01 -i ~/.ssh/aws-pcluster-ed25519-20250115_tokyo.pem
```

If the following message appears, the connection to the Head Node was successful, and the cluster has been successfully set up.

```
Register this system with Red Hat Insights: insights-client --register  
Create an account or view all your systems at https://red.ht/insights-dashboard  
Last login: Tue Feb 18 04:56:09 2025 from xx.xx.xxx.xx
```

Running Test Job

In a cluster environment, including supercomputers, programs are executed as jobs. In the cluster set up following this manual, the job management system "Slurm" is used to submit and execute jobs. To submit a job using Slurm, use the following sbatch command.

```
$ sbatch [file name]
```

The following pages explain how to run a simple test job.

Running Test Job

1. Running Job (1/2)

1.1 Creating a Test Job Script

On the Head Node, run the following command to create a job script named sleep.sh with Vim. Then, add the following lines to the script.

```
$ vim sleep.sh
=====
#!/bin/bash
echo "sleeping..."
sleep 60
echo "done!"
=====
```

1.2 Running sbatch command

Run the following sbatch command to submit the job. Shortly after, the job will run on a compute node.

Note: Keep in mind that the compute node is automatically created when the job is submitted and deleted upon job completion.

```
$ sbatch sleep.sh
-----
Submitted batch job 1
-----
```

Running Test Job

1. Running Job (2/2)

1.3 Checking Job Status

To check the status of a job, run the "squeue" command.

```
$ squeue
-----
JOBID PARTITION NAME USER ST TIME NODES NODELIST(REASON)
1 queue1 sleep.sh ec2-user CF 0:08 1 queue1-dy-m7gmedium-12
-----
```

Note: For information on job state codes, refer to the following URL.

<https://slurm.schedmd.com/squeue.html#lbAG>

1.4 Computation Completion

After the computation completes, a file named slurm-xx.out (where xx is a number) will be generated. If the following output appears in the file, the computation has completed successfully.

```
$ cat slurm-1.out
-----
sleeping...
done!
-----
```

Transferring Data to the Shared Storage

This section explains how to transfer data to shared storage. The following directory on the Head Node is mounted with the Lustre file system (FSx for Lustre) and serves as shared storage. Data stored in the Lustre file system can be accessed or deleted from both the Head Node and compute nodes.

Shared Storage
(FSx for Lustre)

/lustre01

1. Data Transfer Procedure for Lustre Shared Storage

- Log into the Head Node and run the following command to copy a file from the Head Node to the Lustre file system. (The following command copies a file called "test.txt" to the Lustre file system.)

```
$ cp test.txt /lustre01/
```

- Log in to the Head Node and run the following command to copy data from the Lustre file system to the Head Node. (In this example, the command copies a file named test.txt from the Lustre file system.)

```
$ cp /lustre01/test.txt .
```

- You can view the data in the Lustre file system using the following command.

```
$ ls -l /lustre01/
-----
-rw-rw-r-- 1 ec2-user ec2-user 5 Jan 17 23:23 test.txt
-----
```

Exporting to External Storage

This section explains how to transfer data to external storage. When a cluster environment is deleted, any data stored in its shared storage is also erased. To prevent data loss, storing data in external storage is essential. The following pages guide you through the process of transferring data to external storage using an S3 bucket.

1. Exporting Data from Shared Storage to External Storage
2. Checking and Downloading Files in S3 Bucket

Exporting to External Storage

1. Exporting Data from Shared Storage to External Storage

Run the following command to export data from shared storage to external storage.

- Run the following command to copy data from the Lustre file system to an S3 bucket.

```
$ nohup find /lustre01/ -type f -print0 | xargs -0 -n 1 sudo lfs hsm_archive
```

- Run the following command to copy a specific file from the Lustre file system to an S3 bucket.(The command below copies test.txt to the S3 bucket.)

```
$ sudo lfs hsm_archive /lustre01/test.txt
```

Exporting to External Storage

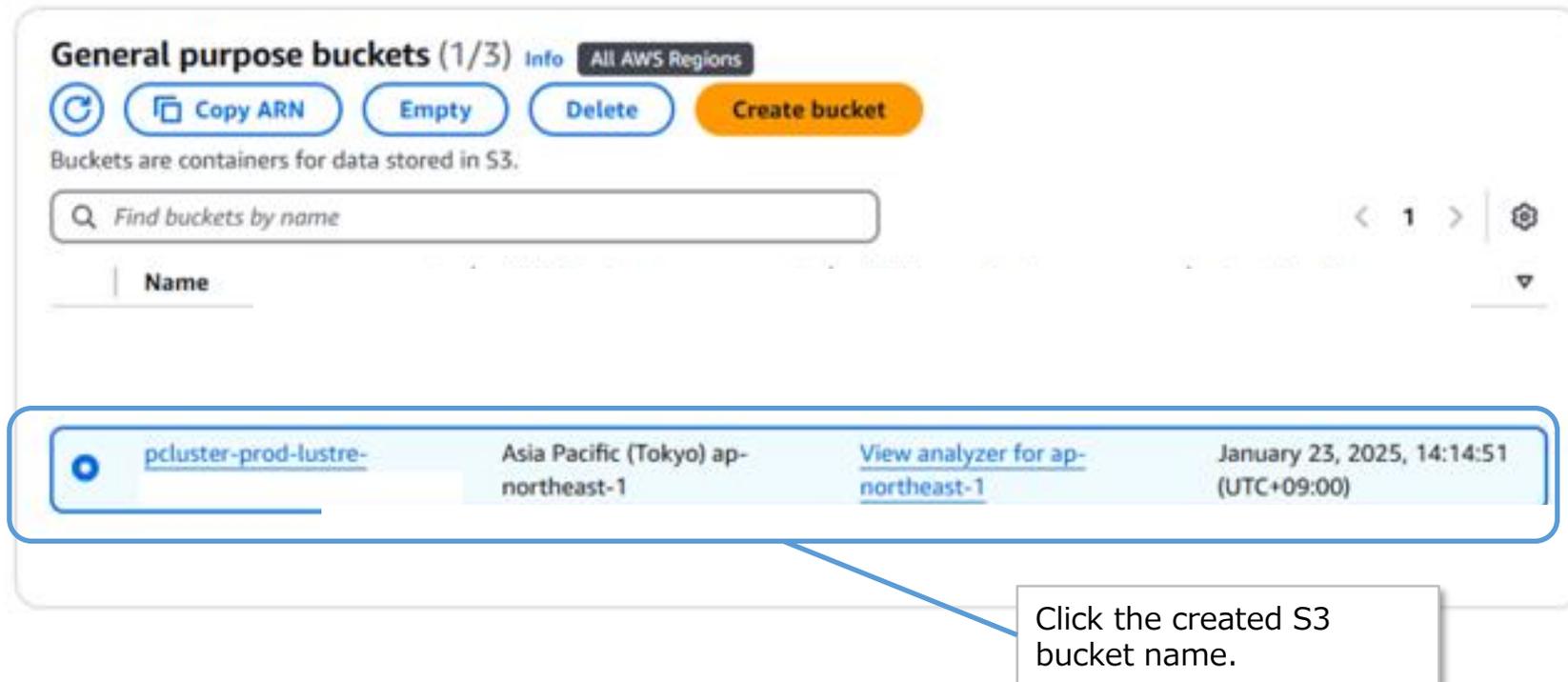
2. Checking and Downloading Files in S3 Bucket (1/3)

Follow the steps below to view and download files from an S3 bucket.

2.1 In the AWS Management Console, navigate to the S3 bucket list page.

<https://ap-northeast-1.console.aws.amazon.com/s3/buckets?region=ap-northeast-1&bucketType=general>

2.2 Click the S3 bucket name (pcluster-prod-lustre-xxxxxxx) that was created on the "[Creating External Storage](#)" section.



Exporting to External Storage

2. Checking and Downloading Files in S3 Bucket (2/3)

2.3 Confirm that the data from the Lustre file system is stored in the S3 bucket.

Objects (14)

Objects are the fundamental entities stored in Amazon S3. You can use [Amazon S3 inventory](#) to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. [Learn more](#)

Find objects by prefix

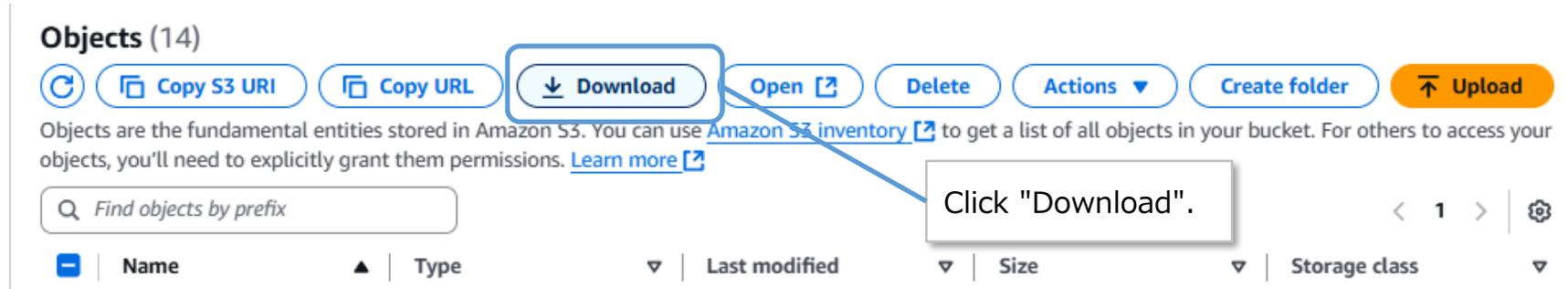
Name	Type	Last modified	Size	Storage class
<input type="checkbox"/> slurm-18.out	out	January 29, 2025, 18:48:14 (UTC+09:00)	3.9 KB	Standard

The data transferred from the Lustre file system appears.

Exporting to External Storage

2. Checking and Downloading Files in S3 Bucket (3/3)

2.4 If you need to save data to your local machine, select the target object and click "Download" to download it to your local machine.
(In this example, test.txt is selected in this case.)



Data Transfer from Local Environment to the Cluster

Transferring Data from the Local Environment to the Cluster

This section outlines the procedure for transferring data from a local environment to the cluster. When running computations on the cluster, it is necessary to upload the required data to the Head Node. The following pages guide you through the process of transferring data to Head Node.

- 1 . Uploading Files from the Local Environment to Shared Storage
- 2 . Retrieving Data from Shared Storage

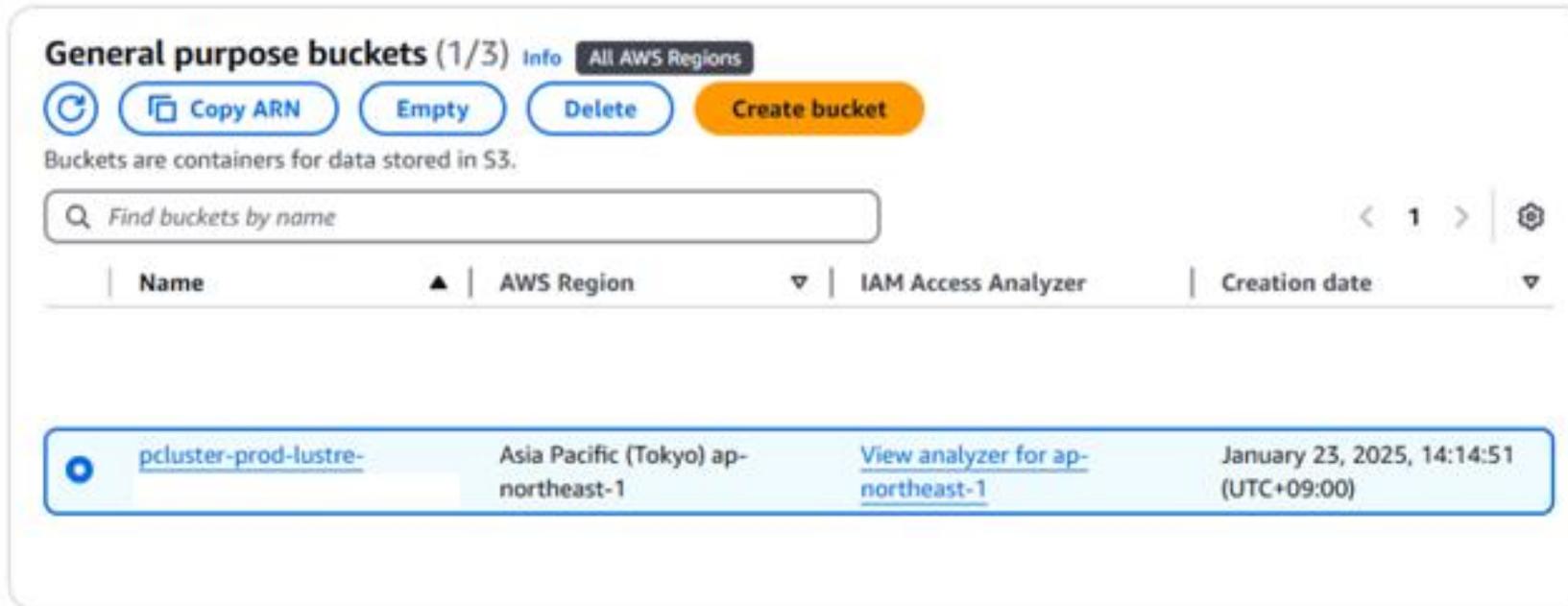
Data Transfer from Local Environment to the Cluster

1. Uploading Files from the Local Environment to Shared Storage(1/3)

1.1 In the AWS Management Console, navigate to the S3 bucket list page.

<https://ap-northeast-1.console.aws.amazon.com/s3/buckets?region=ap-northeast-1&bucketType=general>

1.2 Click the S3 bucket name that was created on the "[Creating External Storage](#)" section.



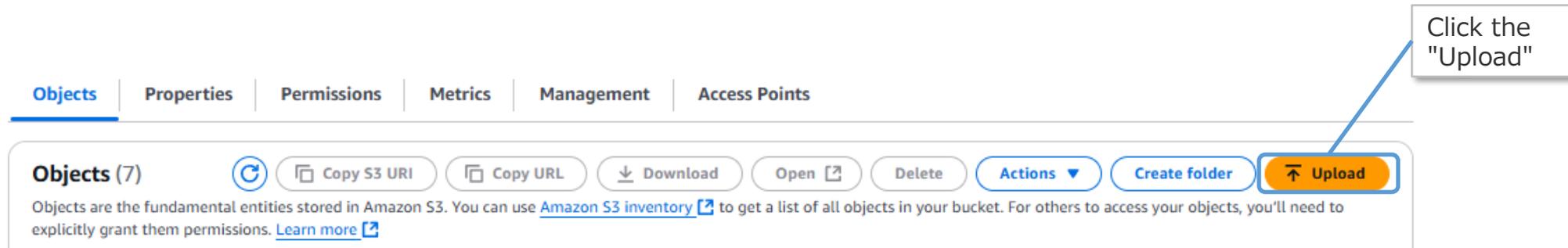
The screenshot shows the AWS Management Console interface for S3 buckets. At the top, it displays "General purpose buckets (1/3)" with an "Info" link and a "All AWS Regions" dropdown. Below this are several action buttons: a refresh icon, "Copy ARN", "Empty", "Delete", and a prominent orange "Create bucket" button. A descriptive text states "Buckets are containers for data stored in S3." Below this is a search bar labeled "Find buckets by name". The main area contains a table with columns for "Name", "AWS Region", "IAM Access Analyzer", and "Creation date". One bucket is listed and selected with a blue circle: "pcluster-prod-lustre-" in the "Name" column, "Asia Pacific (Tokyo) ap-northeast-1" in the "AWS Region" column, a link "View analyzer for ap-northeast-1" in the "IAM Access Analyzer" column, and "January 23, 2025, 14:14:51 (UTC+09:00)" in the "Creation date" column.

Name	AWS Region	IAM Access Analyzer	Creation date
pcluster-prod-lustre-	Asia Pacific (Tokyo) ap-northeast-1	View analyzer for ap-northeast-1	January 23, 2025, 14:14:51 (UTC+09:00)

Data Transfer from Local Environment to the Cluster

1. Uploading Files from the Local Environment to Shared Storage(2/3)

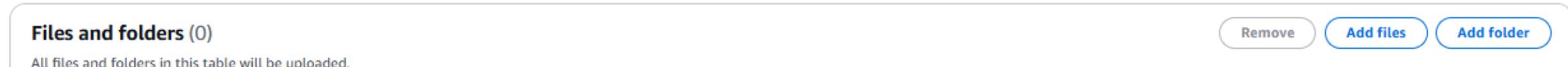
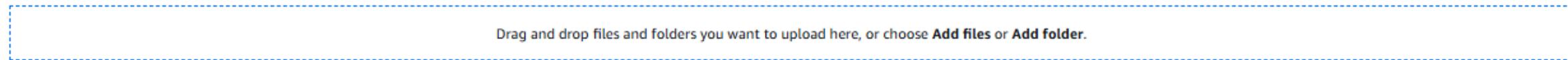
1.3 Click the "Upload".



1.4 Drop the files you wish to upload into the window.

Upload [Info](#)

Add the files and folders you want to upload to S3. To upload a file larger than 160GB, use the AWS CLI, AWS SDKs or Amazon S3 REST API. [Learn more](#)

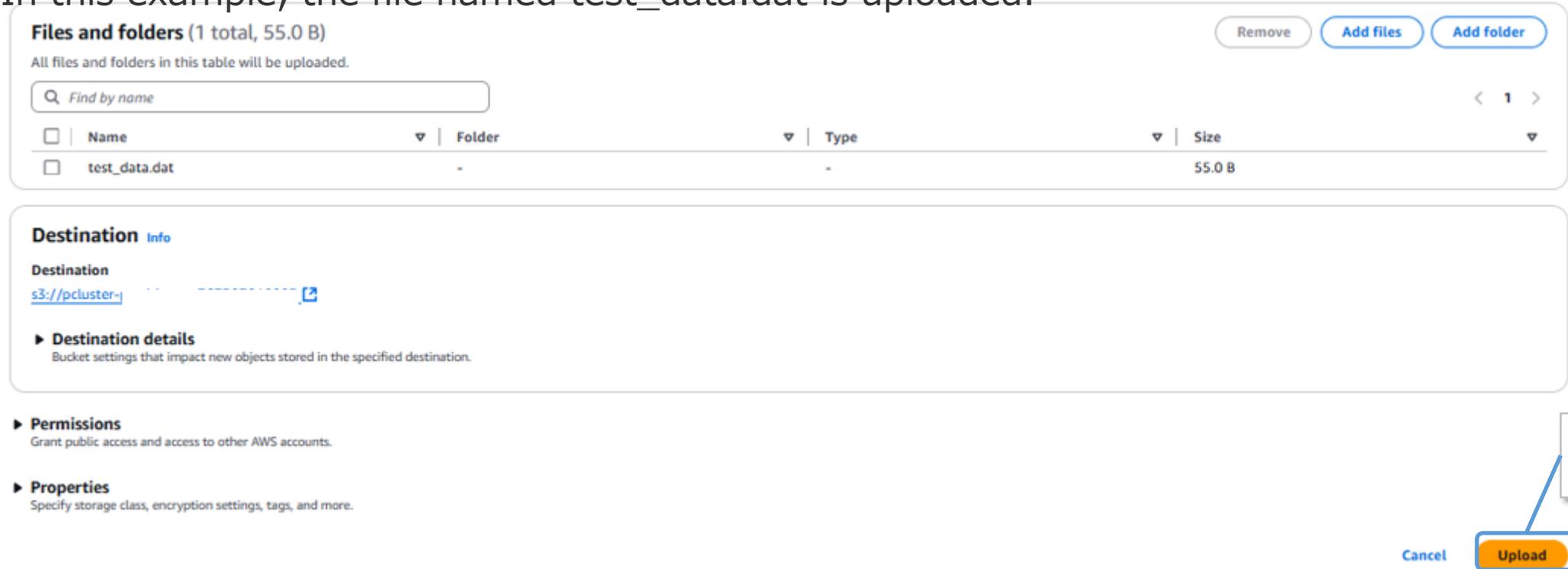


Data Transfer from Local Environment to the Cluster

2. Uploading Files from the Local Environment to Shared Storage(3/3)

2.1 Click the "Upload".

In this example, the file named test_data.dat is uploaded.



Files and folders (1 total, 55.0 B) Remove Add files Add folder

All files and folders in this table will be uploaded.

Find by name

<input type="checkbox"/>	Name	Folder	Type	Size
<input type="checkbox"/>	test_data.dat	-	-	55.0 B

Destination [Info](#)

Destination
s3://pcluster-

Destination details
Bucket settings that impact new objects stored in the specified destination.

Permissions
Grant public access and access to other AWS accounts.

Properties
Specify storage class, encryption settings, tags, and more.

Cancel Upload

Click the "Upload".

The following screen confirms that the upload to the S3 bucket is complete.



Upload succeeded
For more information, see the [Files and folders](#) table.

Data Transfer from Local Environment to the Cluster

2. Retrieving Data from Shared Storage

2.2 Log in to the Head Node and run the following command to verify the data in the Lustre file system.

```
$ ls /lustre01/  
-----  
slurm-1.out  test_data.dat  
-----
```

2.3 Execute the following command to transfer data from the Lustre file system to the Head Node. Here, test_data.dat is used as an example.

```
$ cp /lustre01/test_data.dat .
```

2. Building "Private Fugaku"

(3) Installing "Virtual Fugaku"



What is Singularity?

About Singularity

Singularity is a container platform primarily used in HPC. With Singularity, of the system configuration required to execute an application, the parts other than the kernel can be independently configured and maintained, and can be integrated with the application, so if the environment has the same architecture, the entire execution environment can be carried. Also, if you need to update the runtime libraries frequently, or if you need to make full use of various applications, you do not have to think about individually adjusting or switching each requirement. Furthermore, it is easy to save the old execution environment, which makes it easier to revalidate the data.

Comparison with other implementations

There are four distinctive features of Singularity compared to other container implementations:

- Since the image is created in a single file, managing container images is intuitive and easy.
- Since the image can be used as a single file without extraction, it reduces I/O to the shared file system during application execution.
- The ID and privileges of the started user are inherited inside the container. As a results, files, Hardware, and processes can be accessed as the same user.
- Secure operation is possible because no root authority or daemon is involved when creating an image or executing a container.

Installation of Singularity

This section explains the installation process for Singularity. The installation consists of the following steps. Each step is explained in detail on the following pages.

1. Installing dependencies
2. Installing Go
3. Compilation of Singularity
4. Configuring the Singularity Command Path

Installation of Singularity

1. Installing dependencies

Singularity is installed on the Head Node, so log in to the Head Node to install it.

1.1 Run the following command to install the dependencies.

```
# Install basic tools for compiling
sudo dnf groupinstall -y 'Development Tools'
# Install RPM packages for dependencies
sudo dnf install -y ¥
autoconf ¥
automake ¥
crun ¥
cryptsetup ¥
fuse ¥
fuse3 ¥
fuse3-devel ¥
git ¥
glib2-devel ¥
libseccomp-devel ¥
libtool ¥
squashfs-tools ¥
wget ¥
zlib-devel
```

Installation of Singularity

2. Installing Go

Go must be installed to install Singularity.

2.1 Run the following command to install Go.

```
$ sudo yum install -y golang
```

2.2 Go is installed successfully if you run the following command and see the version information.

```
$ go version  
=====  
go version go1.22.9 (Red Hat 1.22.9-1.module+el8.10.0+22500+aee717ef) linux/arm64  
=====
```

Installation of Singularity

3. Compilation of Singularity

3.1 Run the following command to download the Singularity source code. Make sure to use version v4.1.0.

```
$ export VERSION=4.1.0 && # adjust this as necessary ¥  
  wget https://github.com/sylabs/singularity/releases/download/v${VERSION}/singularity-ce-${VERSION}.tar.gz && ¥  
  tar -xzf singularity-ce-${VERSION}.tar.gz && ¥  
  cd singularity-ce-${VERSION}
```

3.2 Run the following command to compile Singularity.

Singularity needs to be accessible and executable from the compute nodes, so it must be installed in a shared location accessible to the compute nodes. Therefore, specify the shared location `"/opt/parallelcluster/shared"` using the `--prefix` option.

```
$ ./mconfig --prefix=/opt/parallelcluster/shared && ¥  
  make -C ./builddir && ¥  
  sudo make -C ./builddir install
```

Installation of Singularity

4. Configuring the Singularity Command Path

4.1 Run the following command to open and edit the .bashrc file.

```
$ vim ~/.bashrc
```

4.2 Add the following to the ~/.bashrc file, then save the file.

```
export PATH=/opt/parallelcluster/shared/bin:$PATH
```

4.3 Run the following command to apply the updated PATH environment variable.

```
$ source ~/.bashrc
```

4.4 Run the following command. If the version "4.1.0" appears, Singularity has been installed successfully.

```
$ singularity version
=====
4.1.0
=====
```

Obtaining the "Virtual Fugaku" Environment

To deploy a cluster environment similar to "Fugaku", download the Singularity container image containing the software stack used on "Fugaku". This section explains how to download the "Virtual Fugaku" container image.

1. Downloading the "Virtual Fugaku" Container Image File

1.1 Log in to the Head Node and run the following command from your home directory to download the Singularity container image from Sylabs Cloud.

```
$ singularity pull library://riken-rccs/virtual-fugaku/vf-ver1.1
```

After the download completes, "Virtual Fugaku" is now installed.

* The above is an example for version 1.1. Please specify the latest version in practice. You can check the latest version on [the official website of "Virtual Fugaku."](https://www.riken-rccs.riken.jp/en/fugaku/virtual-fugaku/) (<https://www.riken-rccs.riken.jp/en/fugaku/virtual-fugaku/>)

3. Running Applications

(1) Running applications installed on "Virtual Fugaku"



Example Applications to Run

This manual provides examples of running the following three applications that are installed on "Virtual Fugaku".

- GENESIS

It is highly parallel molecular dynamics simulation software. Using a specialized computational algorithm, it improves parallel computing efficiency and enables fast simulations of systems with 100 million atoms, simulating a cellular environment.

URL: https://www.r-ccs.riken.jp/software_center/software/genesis/overview/

- GROMACS

It is an open-source classical molecular dynamics software application, primarily used for biomolecular system simulations. It is known for its ability to execute high-speed parallel computations.

URL: <https://www.gromacs.org/>

- SCALE

It is a foundational library for next-generation meteorology, developed for broad usage across systems ranging from supercomputers to general-purpose computers.

URL: <https://scale.riken.jp/>

Execution of GENESIS

This section explains the process of running GENESIS, including downloading the required files, creating job scripts, executing them, and verifying the results.

1. Execution Process (1/3)

All subsequent operations should be performed on the Head Node.

1.1 Downloading Required Files

Run the following command in any directory.

```
$ wget https://www.r-ccs.riken.jp/labs/cbrt/wp-content/uploads/2020/12/benchmark_mkl_ver4_nocrowding.tar.gz  
$ tar -xzvf benchmark_mkl_ver4_nocrowding.tar.gz
```

1.2 Navigate to the "benchmark_mkl_ver4_nocrowding" directory.

```
$ cd benchmark_mkl_ver4_nocrowding
```

Execution of GENESIS

1. Execution Process (2/3)

1.3 Creating a Job Script

Create the job script "genesis.sh" in the "benchmark_mkl_ver4_nocrowding" directory as shown below.

•genesis.sh

```
#!/bin/bash
#SBATCH -p queue1
#SBATCH --ntasks=8
#SBATCH --nodes=2
#SBATCH --ntasks-per-node=4
#SBATCH --cpus-per-task=2
#SBATCH -J test_genesis

set -ex

SIFFILE=~/.vf-ver1.1_latest.sif

export SINGULARITYENV_LD_LIBRARY_PATH=/usr/lib64:/opt/amazon/efa/lib64
export SINGULARITY_BIND=${PWD},/opt/amazon,/usr/lib64/libefa.so.1,/usr/lib64/libibverbs.so.1
export OMP_NUM_THREADS=${SLURM_CPUS_PER_TASK}

cd npt/genesis1.6_2.5fs/jac_amber
mpirun --use-hwthread-cpus -n ${SLURM_NTASKS} singularity -v run ${SIFFILE} spdyn p${SLURM_NTASKS}.inp
```

1.4 Running the Job

Run the following command to submit the job.

```
$ sbatch genesis.sh
```

Execution of GENESIS

1. Execution Process (3/3)

1.5 Verifying the Results

After the job is completed, a file named slurm-xx.out will be generated in the directory where the job was submitted (xx represents the job ID). Check the file contents. If the following output appears, it means the application is working correctly.

```
Output_Time> Averaged timer profile (Min, Max)
total time      =      56.798
  setup         =       1.057
  dynamics      =      55.741
  energy        =      42.362
  integrator    =       6.591
  pairlist      =       4.504 (      4.376,      4.620)
=====以下省略=====
```

Execution of GROMACS

This section explains the process of running GROMACS, including downloading the required files, creating job scripts, executing them, and verifying the results.

1. Execution Process (1/3)

All subsequent operations should be performed on the Head Node.

1.1 Downloading Required Files

Run the following command in any directory.

```
$ wget https://ftp.gromacs.org/pub/benchmarks/ADH_bench_systems.tar.gz  
$ tar -xzvf ADH_bench_systems.tar.gz
```

1.2 Navigate to the "adh_cubic" directory inside the "ADH" directory.

```
$ cd ADH/adh_cubic
```

Execution of GROMACS

1. Execution Process (2/3)

1.3 Creating a Job Script

Create the job script "gromacs.sh" in the "adh_cubic" directory as shown below.

- gromacs.sh

```
#!/bin/bash
#SBATCH -p queue1
#SBATCH --ntasks=8
#SBATCH --cpus-per-task=2
#SBATCH --nodes=4#SBATCH --ntasks-per-node=2
#SBATCH -J test_gromacs
SIFFILE=~/.vf-ver1.1_latest.sif
export SINGULARITYENV_LD_LIBRARY_PATH=/usr/lib64:/opt/amazon/efa/lib64
export SINGULARITY_BIND=/opt/amazon,/usr/lib64/libefa.so.1,/usr/lib64/libibverbs.so.1
mpiexec --use-hwthread-cpus -n 1 singularity run ${SIFFILE} gmx_mpi grompp -f pme_verlet.mdp -c conf.gro -p topol.top -o ions.tpr
mpiexec --use-hwthread-cpus -n ${SLURM_NTASKS} singularity run ${SIFFILE} gmx_mpi mdrun -ntomp ${SLURM_CPUS_PER_TASK} -s ions.tpr
```

1.4 Running the Job

Run the following command to submit the job.

```
$ sbatch gromacs.sh
```

Execution of GROMACS

1. Execution Process (3/3)

1.5 Verifying the Results

After the job is completed, a file named slurm-xx.out will be generated in the directory where the job was submitted (xx represents the job ID). Check the file contents. If the following output appears, it means the application is working correctly.

```
Dynamic load balancing report:
DLB was turned on during the run due to measured imbalance.
Average load imbalance: 1.0%.
The balanceable part of the MD step is 68%, load imbalance is computed from this.
Part of the total run time spent waiting due to load imbalance: 0.7%.
Steps where the load balancing was limited by -rdd, -rcon and/or -dds: X 0 %
      Core t (s)   Wall t (s)      (%)
Time:      2017.288   126.081   1600.0      (ns/day)   (hour/ns)
Performance:    13.707     1.751
GROMACS reminds you: "Even if you are on the right track, you will get run over if you just sit there." (Will Rogers)
```

Execution of SCALE

This section explains the process of running SCALE, including downloading the required files, creating job scripts, executing them, and verifying the results. Additionally, it also explains how to visualize the computed results using the visualization tool GrADS.

1. Execution Process (1/4)

All subsequent operations should be performed on the Head Node.

1.1 Downloading Required Files

Run the following command in any directory.

```
$ wget https://scale.riken.jp/archives/scale-5.4.5.tar.gz  
$ tar -xzvf scale-5.4.5.tar.gz
```

1.2 Navigate to the "scale-5.4.5" directory.

```
$ cd scale-5.4.5
```

Execution of SCALE

1. Execution Process (2/4)

1.3 Creating a Job Script

Create the job script "scale.sh" in the "scale-5.4.5" directory as shown below.

·scale.sh

```
#!/bin/bash
#SBATCH -p queue1
#SBATCH --ntasks=2
#SBATCH --cpus-per-task=8
#SBATCH --nodes=2
#SBATCH --ntasks-per-node=1
#SBATCH -J test_scale
SIFFILE=~/.vf-ver1.1_latest.sif
export SINGULARITYENV_LD_LIBRARY_PATH=/usr/lib64:/opt/amazon/efa/lib64
export SINGULARITY_BIND=${PWD},/opt/amazon,/usr/lib64/libefa.so.1,/usr/lib64/libibverbs.so.1
cp -pr ../scale-5.4.5/scale-rm/test/tutorial/ideal/* .
#Preprocessing
cp sample/init_R20kmDX500m.conf ./init_R20kmDX500m.conf
mpexec --use-hwthread-cpus -n ${SLURM_NTASKS} singularity run ${SIFFILE} scale-rm_init init_R20kmDX500m.conf
#Run simulation
cp sample/run_R20kmDX500m.conf ./run_R20kmDX500m.conf
mpexec --use-hwthread-cpus -n ${SLURM_NTASKS} singularity run ${SIFFILE} scale-rm run_R20kmDX500m.conf
#Post processing
cp sample/net2g_R20kmDX500m.conf ./net2g_R20kmDX500m.conf
mpexec --use-hwthread-cpus -n ${SLURM_NTASKS} singularity run ${SIFFILE} net2g net2g_R20kmDX500m.conf
```

Execution of SCALE

1. Execution Process (3/4)

1.4 Running the Job

Run the following command to submit the job.

```
$ sbatch scale.sh
```

1.5 Result Output

After the job is completed, a file named slurm-xx.out and the following result files will be generated in the directory where the job was submitted (where xx represents the job ID). Download these result files to your local machine.

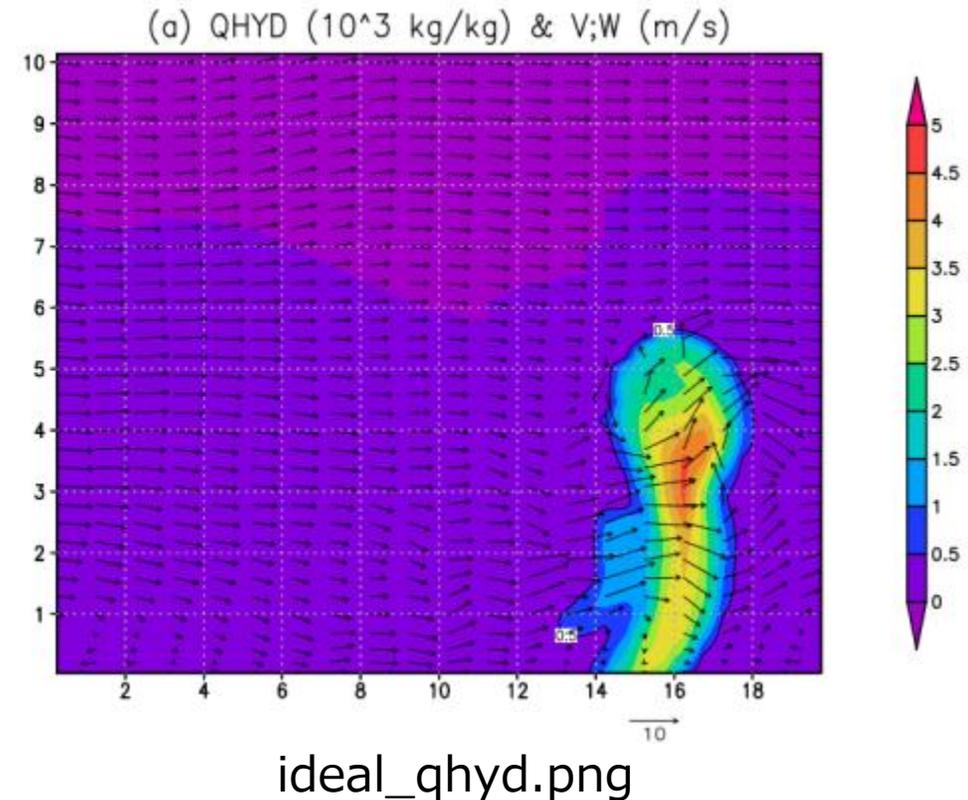
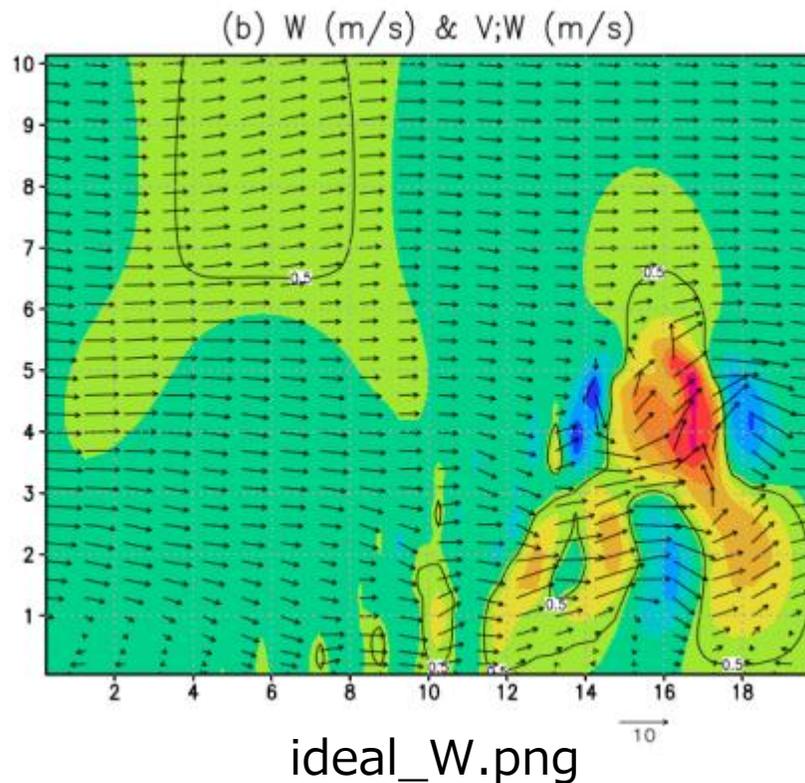
- QHYD_d01z-3d.ctf
- QHYD_d01z-3d.grd
- V_d01z-3d.ctf
- V_d01z-3d.grd
- W_d01z-3d.ctf
- W_d01z-3d.grd

Execution of SCALE

1. Execution Process (4/4)

1.6 Verifying the Results

To visualize the downloaded files on your local machine, use GrADS. If the following two visualization output files are generated after running GrADS, this confirms that the application is functioning correctly. The next page provides the procedure for visualizing data with GrADS.



Execution of SCALE

2. Visualizing SCALE Output with GrADS (1/2)

This section explains how to use GrADS to visualize SCALE output files in a local machine. This visualization process is based on operations in an Ubuntu 22.04.4 local machine.

2.1 Installing GrADS

To install GrADS, run the following command in your local machine.

```
$ sudo apt-get upgrade  
$ sudo apt-install grads
```

2.2 Obtaining the GrADS Visualization Script

The "checkfig_ideal.gs" script file, used for visualizing computation results, is located in "scale-5.4.5/scale-rm/test/tutorial/ideal". This directory was downloaded when running SCALE. Copy this file along with the computation result files to your local machine.

Execution of SCALE

2. Visualizing SCALE Results with GrADS (2/2)

2.3 Executing the Visualization

Navigate to the directory containing the computation result files and the "checkfig_ideal.gs" script. Then, run the following command to launch GrADS and generate the visualization output files.

```
$ grads -blc checkfig_ideal.gs
```

2.4 After running the command, the following visualization output files will be generated in this directory.

- ideal_qhyd.png
- ideal_W.png

4. Deleting "Private Fugaku"

(1) Deleting the Cluster Environment



(1) Deleting the Cluster Environment

This section explains how to delete the ParallelCluster environment. If the cluster environment is no longer needed, follow the steps outlined below to remove it. For a step-by-step guide, refer to the following pages.

1. Deleting Cluster
2. Deleting VPC for Cluster
3. Deleting ParallelCluster Setup Instance
4. Releasing Static IP Address

(1) Deleting the Cluster Environment

To delete the cluster environment, log into the head node via SSH and proceed with the necessary steps.

1. Deleting Cluster

1.1 To proceed with the deletion of the cluster, first activate the Python virtual environment using the following command.

```
$ source ~/apc-ve/bin/activate
```

1.2 To delete the cluster, execute the following command. The deletion process typically takes around 10 to 20 minutes.

```
(apc-ve)$ pcluster delete-cluster --region ap-northeast-1 --cluster-name cluster01
```

1.3 To check the cluster status, run the following command. If the "cloudFormationStackStatus" is "DELETE_IN_PROGRESS", the cluster deletion is in progress. When the deletion process completes, the command will return no output.

```
(apc-ve)$ pcluster describe-cluster --cluster-name cluster01 |grep "cloudFormationStackStatus"
```

(1) Deleting the Cluster Environment

2. Deleting VPC for Cluster (1/3)

2.1 To proceed with VPC deletion, first verify the stack status by running the following command.

```
(apc-ve)$ aws --region ap-northeast-1 cloudformation list-stacks --stack-status-filter "CREATE_COMPLETE" --query  
"StackSummaries[].StackName" |grep -e "parallelclusternetworking-"  
-----  
"parallelclusternetworking-pubpriv-xxxxxxxxxxxxxx",  
-----
```

"parallelclusternetworking-pubpriv-xxxxxxxxxxxxxx" is the CloudFormation stack identifier used for VPC creation. Ensure you retain this identifier for future reference.

2.2 Deleting the Stack

To delete the stack, run the following command. Substitute "parallelclusternetworking-pubpriv-xxxxxxxxxxxxxx" with the stack name you recorded earlier.

```
(apc-ve)$ aws --region ap-northeast-1 cloudformation delete-stack --stack-name parallelclusternetworking-pubpriv-  
xxxxxxxxxxxxxx
```

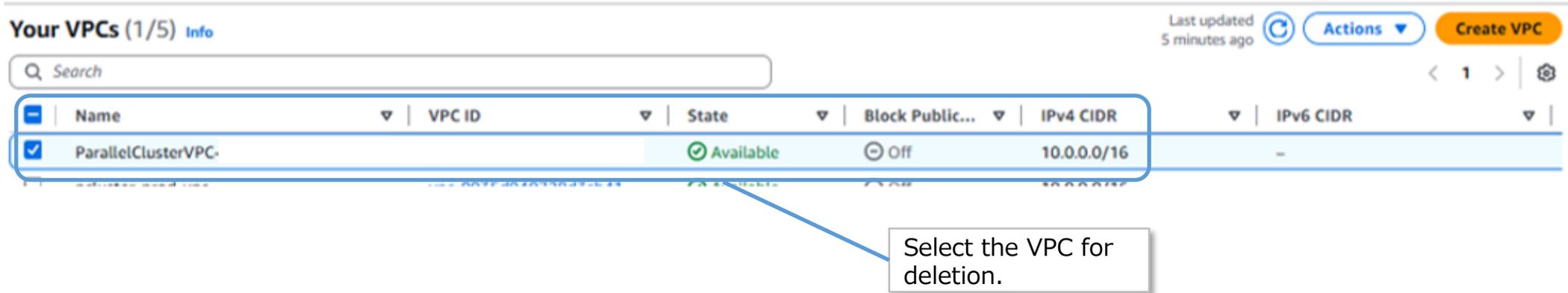
(1) Deleting the Cluster Environment

2. Deleting VPC for Cluster (2/3)

2.3 In the AWS Management Console, navigate to the VPC list page.

<https://ap-northeast-1.console.aws.amazon.com/vpconsole/home?region=ap-northeast-1#vpcs>

Locate and select the VPC formatted as "ParallelClusterVPC-xxxxxxxxxxxxxx", where "xxxxxxxxxxxxxx" represents a date-based identifier.



The screenshot shows the AWS Management Console interface for VPCs. The page title is "Your VPCs (1/5) Info". There is a search bar and a "Create VPC" button. A table lists VPCs with the following columns: Name, VPC ID, State, Block Public..., IPv4 CIDR, and IPv6 CIDR. The first row is selected, and a callout box points to it with the text "Select the VPC for deletion."

Name	VPC ID	State	Block Public...	IPv4 CIDR	IPv6 CIDR
ParallelClusterVPC-		Available	Off	10.0.0.0/16	-

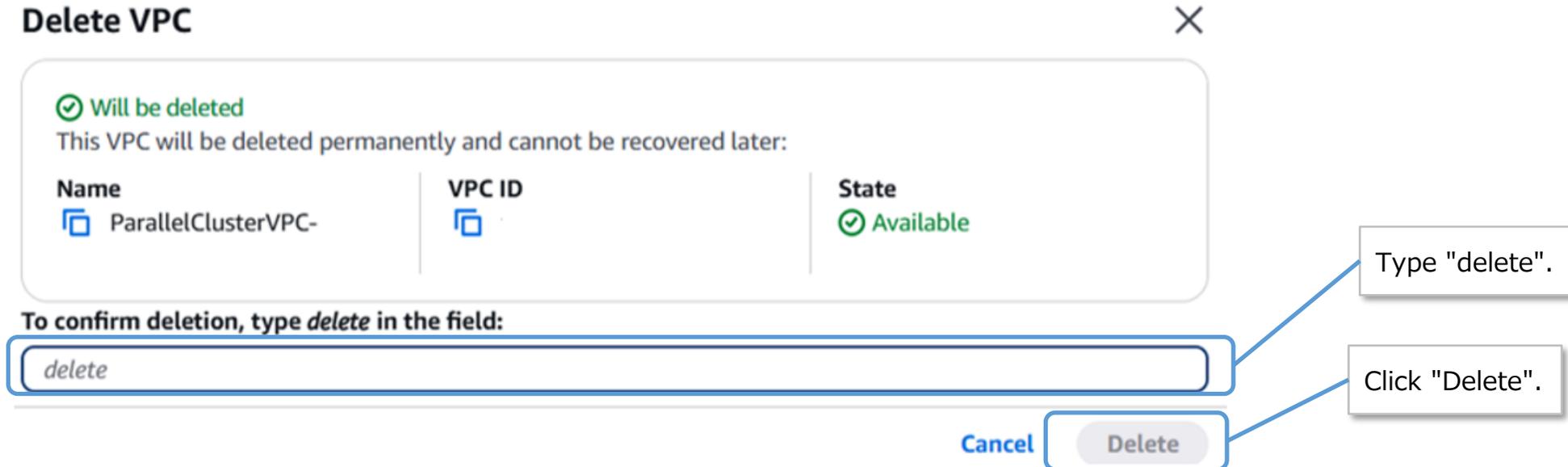
(1) Deleting the Cluster Environment

2. Deleting VPC for Cluster (3/3)

2.4 Click "Actions" from the top-right menu, and choose "Delete VPC".



2.5 Type "delete" in the confirmation field and click "Delete" to delete the VPC.



(1) Deleting the Cluster Environment

3. Deleting ParallelCluster Setup Instance (1/2)

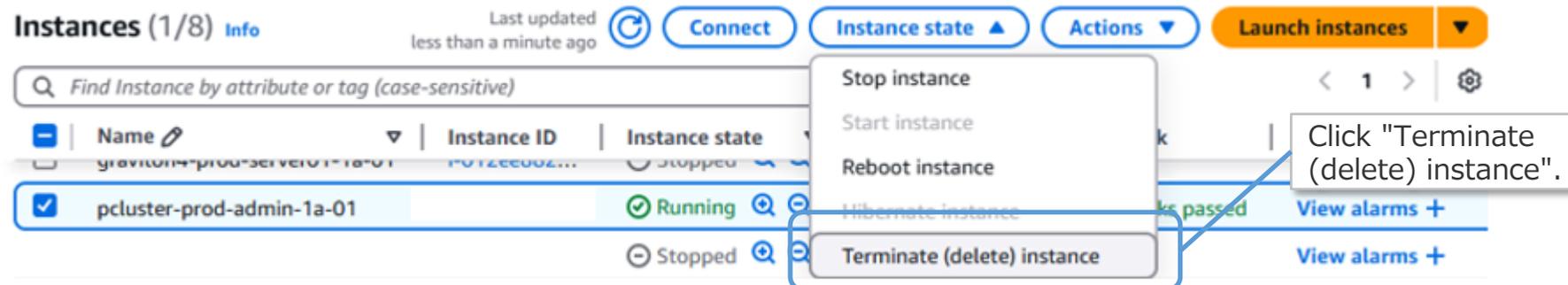
3.1 In the AWS Management Console, navigate to the EC2 Instances page.

<https://ap-northeast-1.console.aws.amazon.com/ec2/home?region=ap-northeast-1#Instances>

3.2 Select "pcluster-prod-admin-1a-01".



3.3 Click "Instance State" in the top-right corner, then select "Terminate (delete) instance".



(1) Deleting the Cluster Environment

3. Deleting ParallelCluster Setup Instance (2/2)

3.4 Click "Terminate (Delete)".

Terminate (delete) instance? ✕

⚠ On an EBS-backed instance, the default action is for the root EBS volume to be deleted when the instance is terminated. Storage on any local drives will be lost.

Are you sure you want to terminate these instances?

Instance ID	Termination protection
	✔ Disabled

Clean up associated resources

Associated resources may incur costs after these instances are terminated.

▶ **Release attached Elastic IPs**

To confirm that you want to delete the instances, choose the terminate button below. Instances with termination protection enabled will not be terminated. Terminating the instance cannot be undone.

Cancel
Terminate (delete)

Click "Terminate (delete)".

(1) Deleting the Cluster Environment

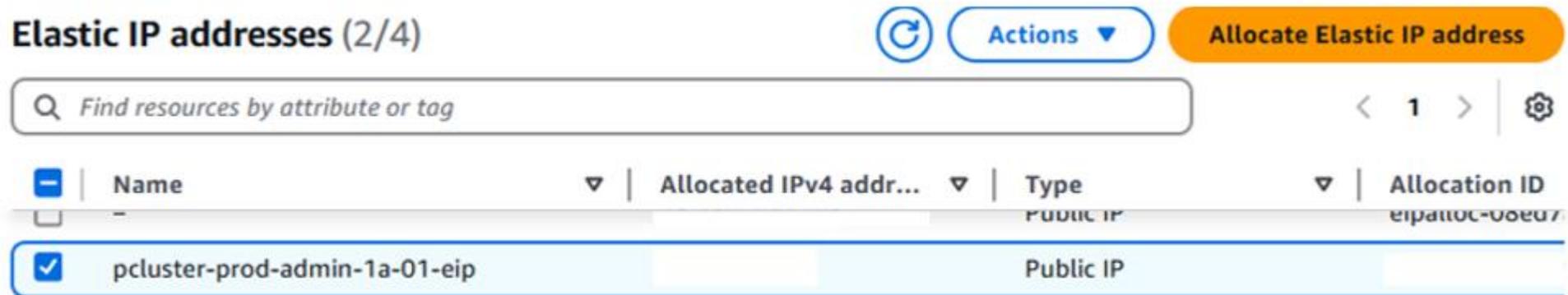
4. Releasing Static IP Address (1/3)

After the deletion of the ParallelCluster setup instance, release the static IP address assigned to the instance to free up resources.

4.1 In the AWS Management Console, navigate to the Elastic IPs page.

<https://ap-northeast-1.console.aws.amazon.com/ec2/home?region=ap-northeast-1#Addresses>

4.2 Select the Elastic IP address to be released.



Elastic IP addresses (2/4)

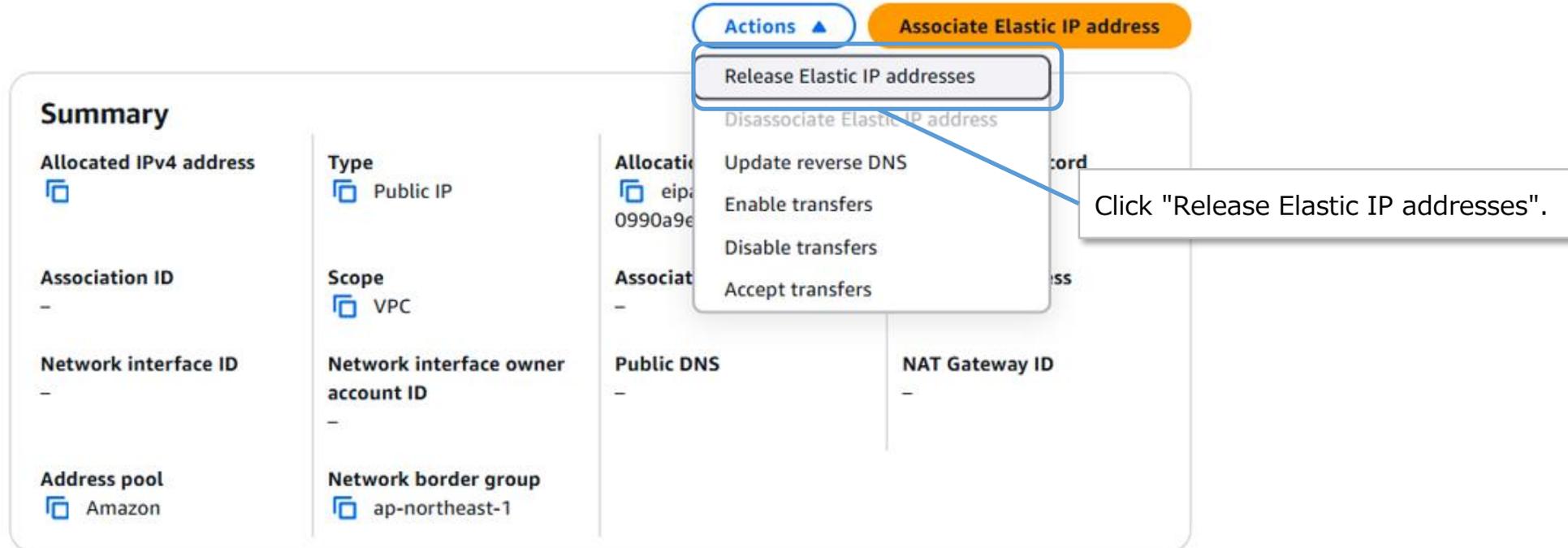
Find resources by attribute or tag

Name	Allocated IPv4 address	Type	Allocation ID
-		Public IP	eipalloc-00e07...
<input checked="" type="checkbox"/> pcluster-prod-admin-1a-01-eip		Public IP	

(1) Deleting the Cluster Environment

4. Releasing Static IP Address (2/3)

4.3 Click the "Actions" in the top-right corner, and choose "Release Elastic IP addresses".



The screenshot shows the AWS console interface for an Elastic IP address. The 'Actions' menu is open, and the 'Release Elastic IP addresses' option is highlighted. A callout box points to this option with the text 'Click "Release Elastic IP addresses".' The console displays the following details:

Summary	
Allocated IPv4 address	Type
Association ID	Public IP
Network interface ID	Scope
Address pool	VPC
	Network interface owner account ID
	Network border group
	ap-northeast-1

Additional fields visible in the console include: Allocation ID (eip-0990a9e...), Association ID, Public DNS, and NAT Gateway ID.

(1) Deleting the Cluster Environment

4. Releasing Static IP Address (3/3)

4.4 Click "Release".

Release Elastic IP addresses ✕

Will be released

If you release the following Elastic IP addresses, they will no longer be allocated to your account and you can no longer associate them with your resources.

Name	IPv4 address	Allocation ID
pcluster-prod-admin...		eipalloc-0990a9e0c9c5c5da7

Cancel
Release

Click "Release".

4.5 After the Elastic IP address is released, the system will display the following message.

Elastic IP addresses released.
 Elastic IP addresses

✕

Reference



Packages Installed in "Virtual Fugaku"

List of Packages Installed in "Virtual Fugaku"

The "Virtual Fugaku" container includes the following applications, which are frequently used on the "Fugaku" supercomputer (as of February 19, 2025). The list is continuously updated, so please check the following URL for the most up-to-date list of applications.

URL: <https://www.r-ccs.riken.jp/en/fugaku/virtual-fugaku/>

- GENESIS 2.1.3 (genesis)
- Gnuplot 6.0.0 (gnuplot)
- GROMACS 2024.2 (gromacs)
- GNU Scientific Library (GSL) 2.7.1 (gsl)
- Julia 1.10.2 (julia)
- LAMMPS 20230802.3 (lammmps)
- Metis 5.1.0 (metis)
- Open Babel 3.1.1 (openbabel)
- OpenFoam 2312 (openfoam)
- Paraview 5.12.1 (paraview)
- Parmetis 4.0.3 (parmetis)
- PETSC 3.21.2 (petsc)
- Atomic Simulation Environment 3.21.1 (py-ase)
- Matplotlib (py-matplotlib)
- MPI for Python (py-mpi4py)
- NumPy (py-numpy)
- pandas (py-pandas)
- scikit-learn (py-scikit-learn)
- SciPy (py-scipy)
- TOML (py-toml)
- Quantum Espresso 7.3.1 (quantum-espresso)
- SCALE 5.4.4 (scale)
- tmux 3.4 (tmux)

List of Shared Storage in AWS ParallelCluster

Shared Storage in the ParallelCluster Environment

The directories on the Head Node listed in the following table are mounted on the Compute Nodes via NFS. To share files necessary for job execution, store them in `"/home"` or `"/opt/parallelcluster/shared"` on the Head Node.

Shared Storage(EBS)
<code>/home</code>
<code>/opt/parallelcluster/shared</code>
<code>/opt/slurm (No write permissions)</code>