"Private Fugaku" Installation Guide

April 1st, 2025
ver0.01

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

INDEX

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"

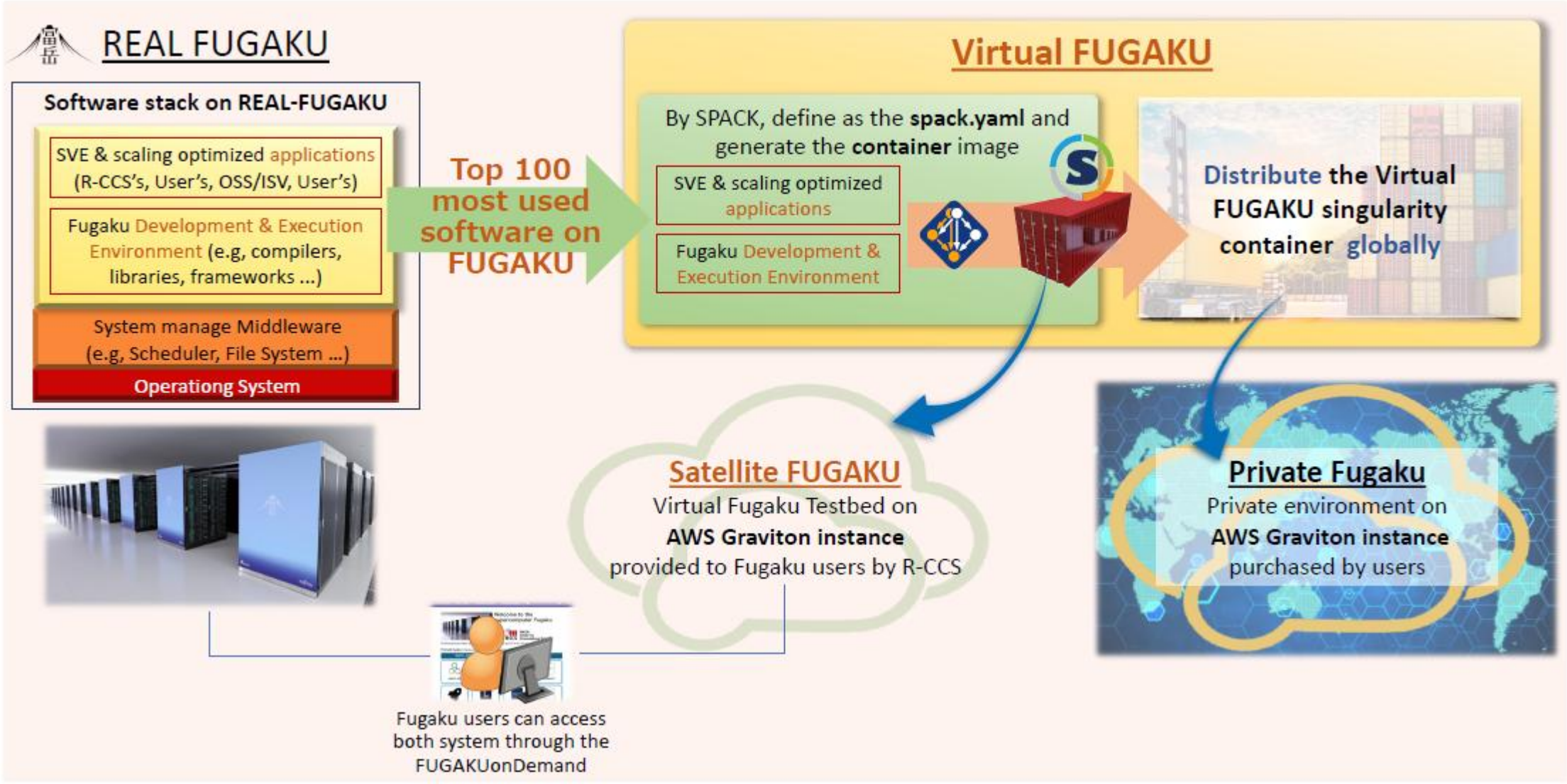
  The system is a computing environment designed for testing and developing "Virtual Fugaku", an initiative proposed by R-CCS. It emulates the software stack of "Fugaku" on a commercial cloud and serves as a shared trial environment for Fugaku users".

- "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/

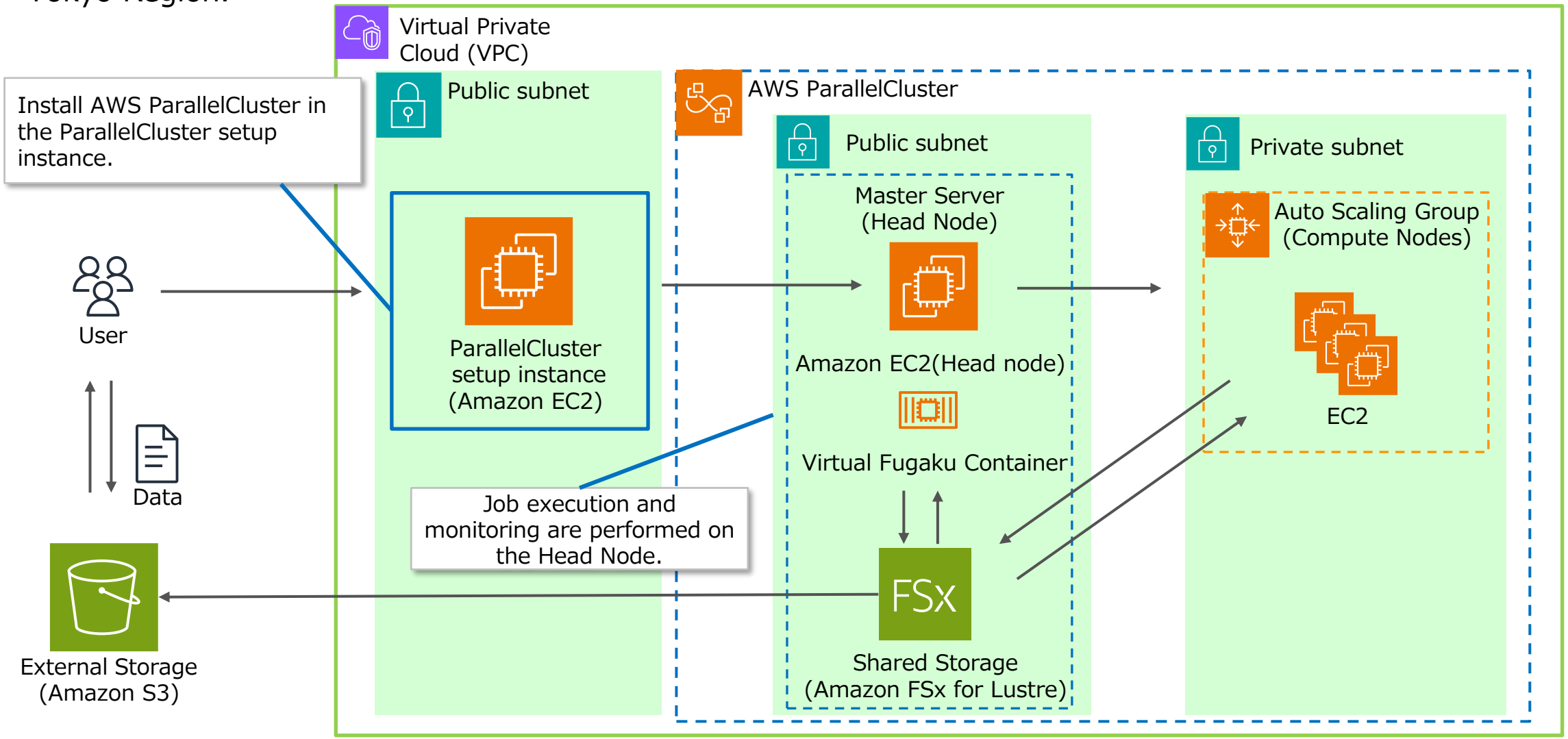
Contact

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

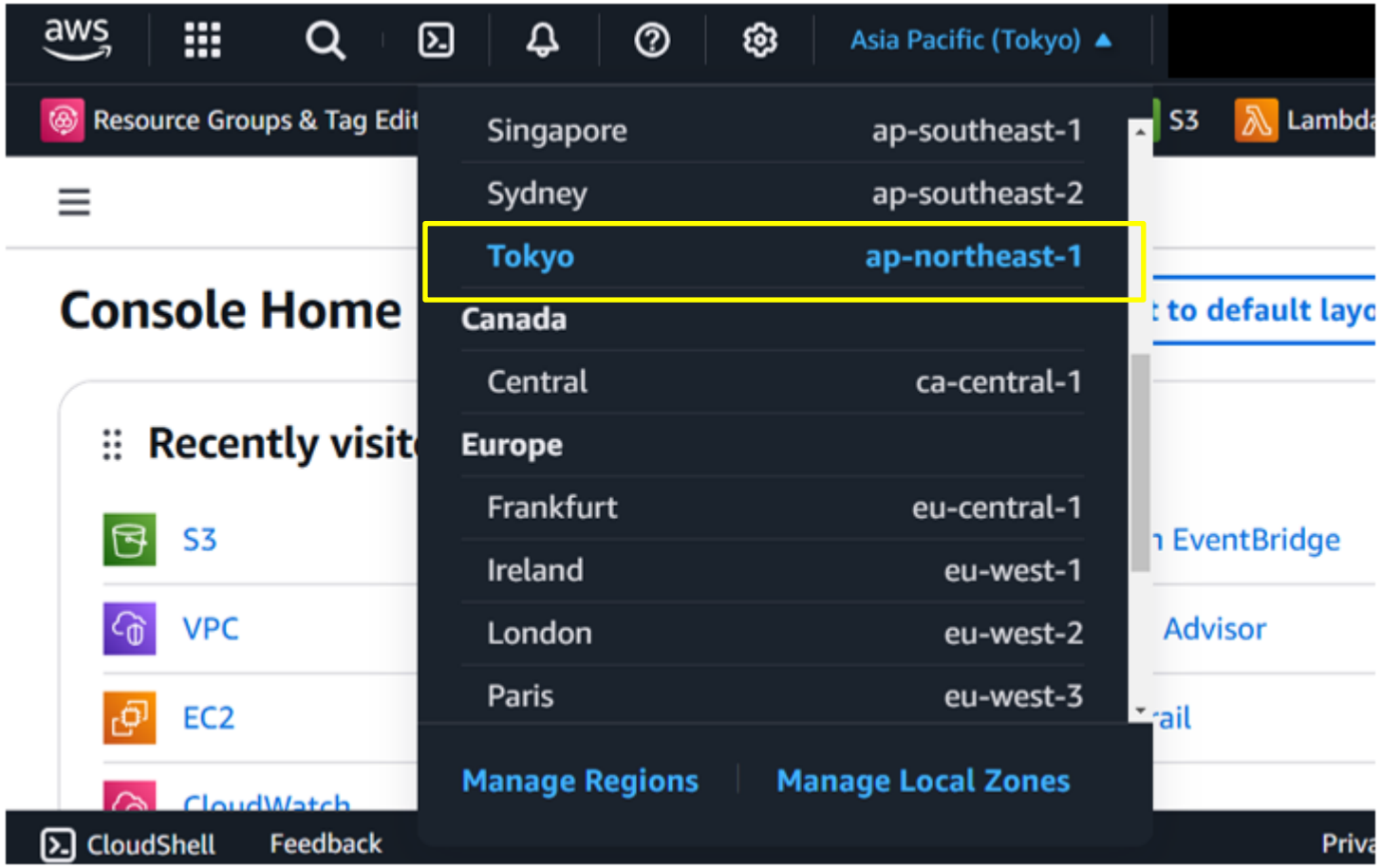
Before setting up a cluster with AWS ParallelCluster, network configuration must be completed. This section explains how to configure the network for cluster setup. The network environment is configured following the steps below, and each step is explained in detail on the following pages.

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)".



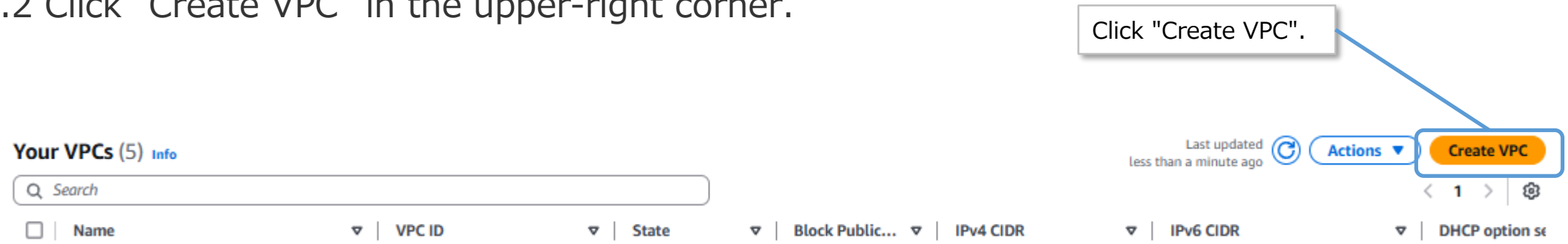
Network Configuration

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

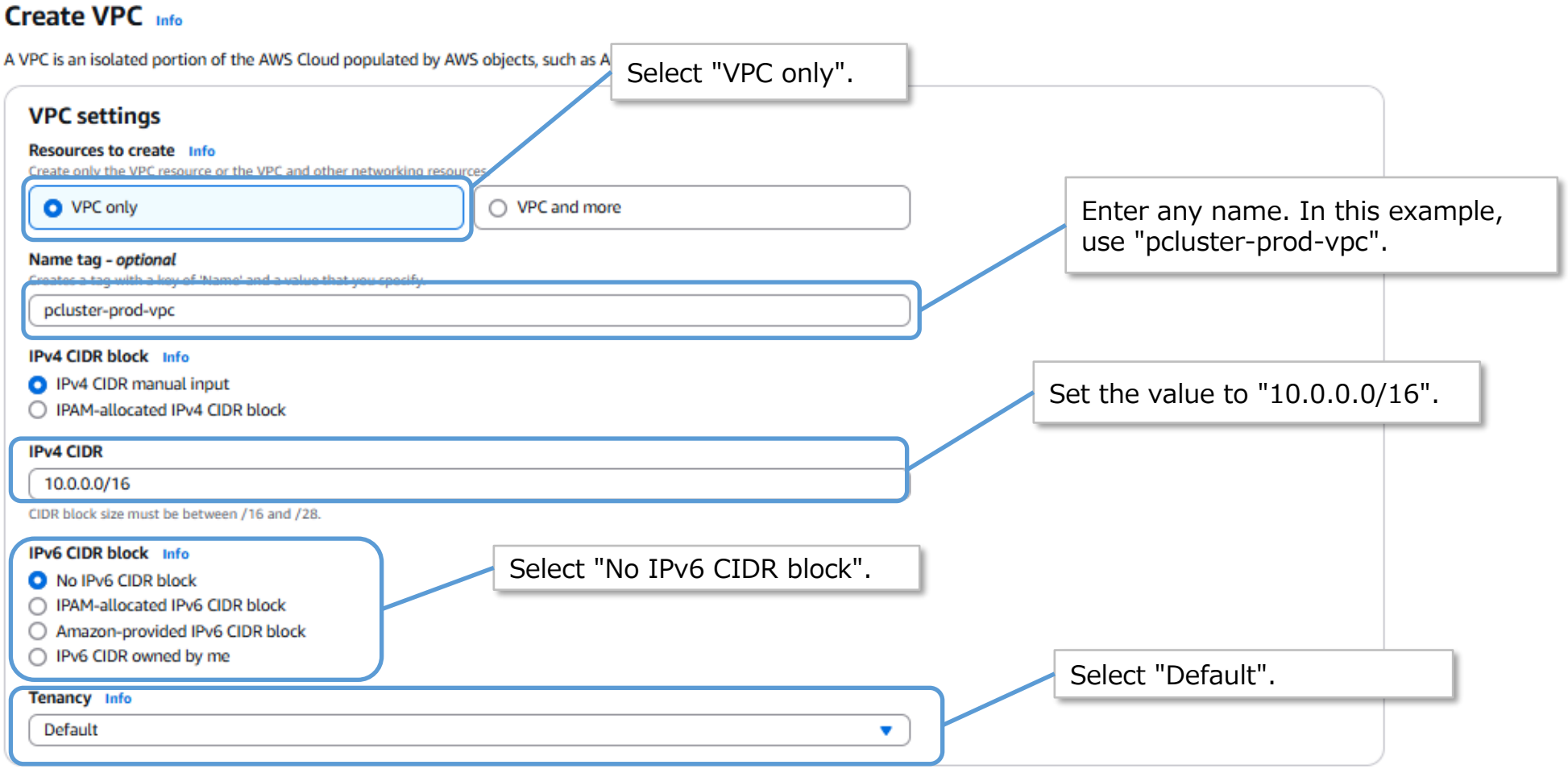
2.2 Click "Create VPC" in the upper-right corner.



Network Configuration

2. Creating VPC (2/3)

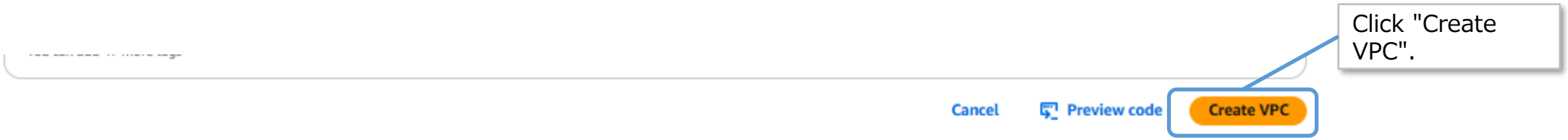
2.3 Specify the parameters as follows.



Network Configuration

2. Creating VPC (3/3)

2.4 Click "Create VPC".



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

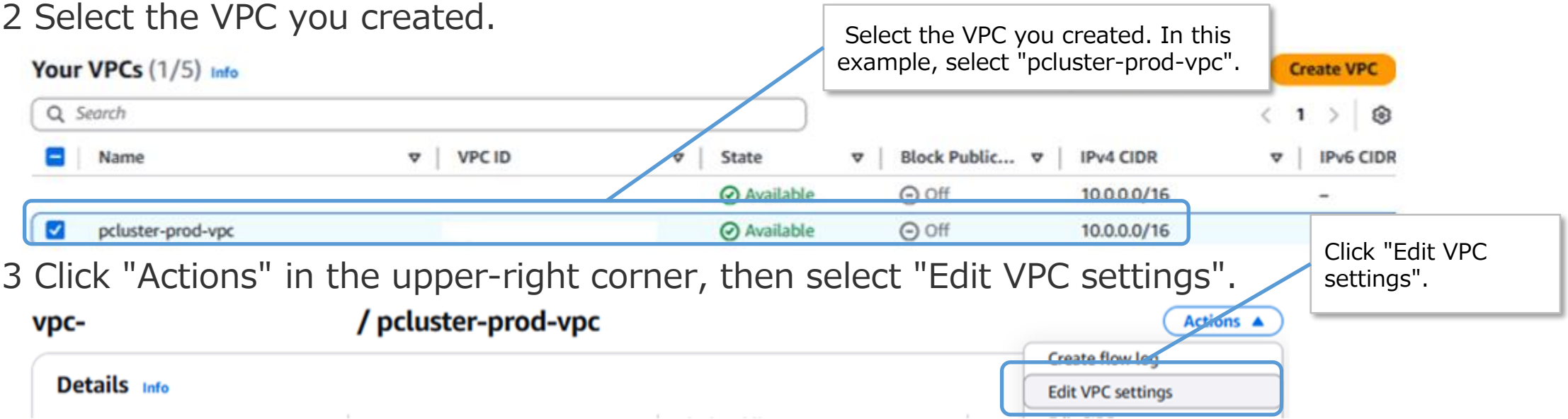
Network Configuration

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.



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

Network Configuration

3. Enable DNS Hostnames (2/2)

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

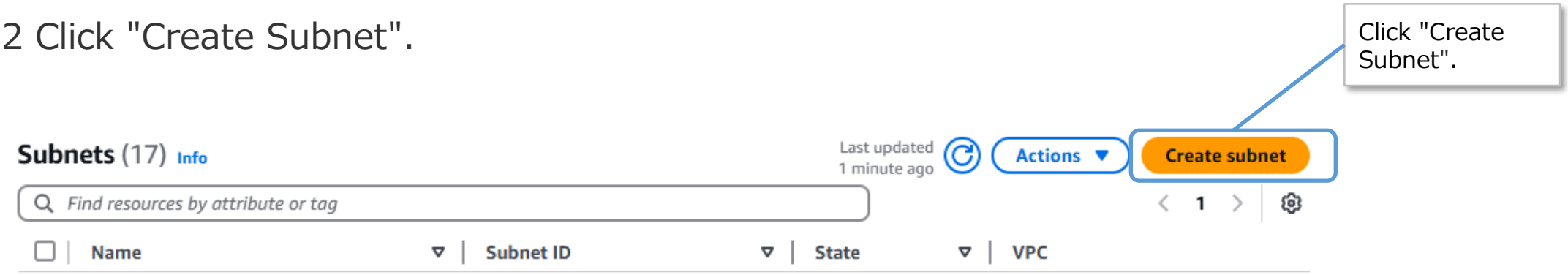
Network Configuration

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

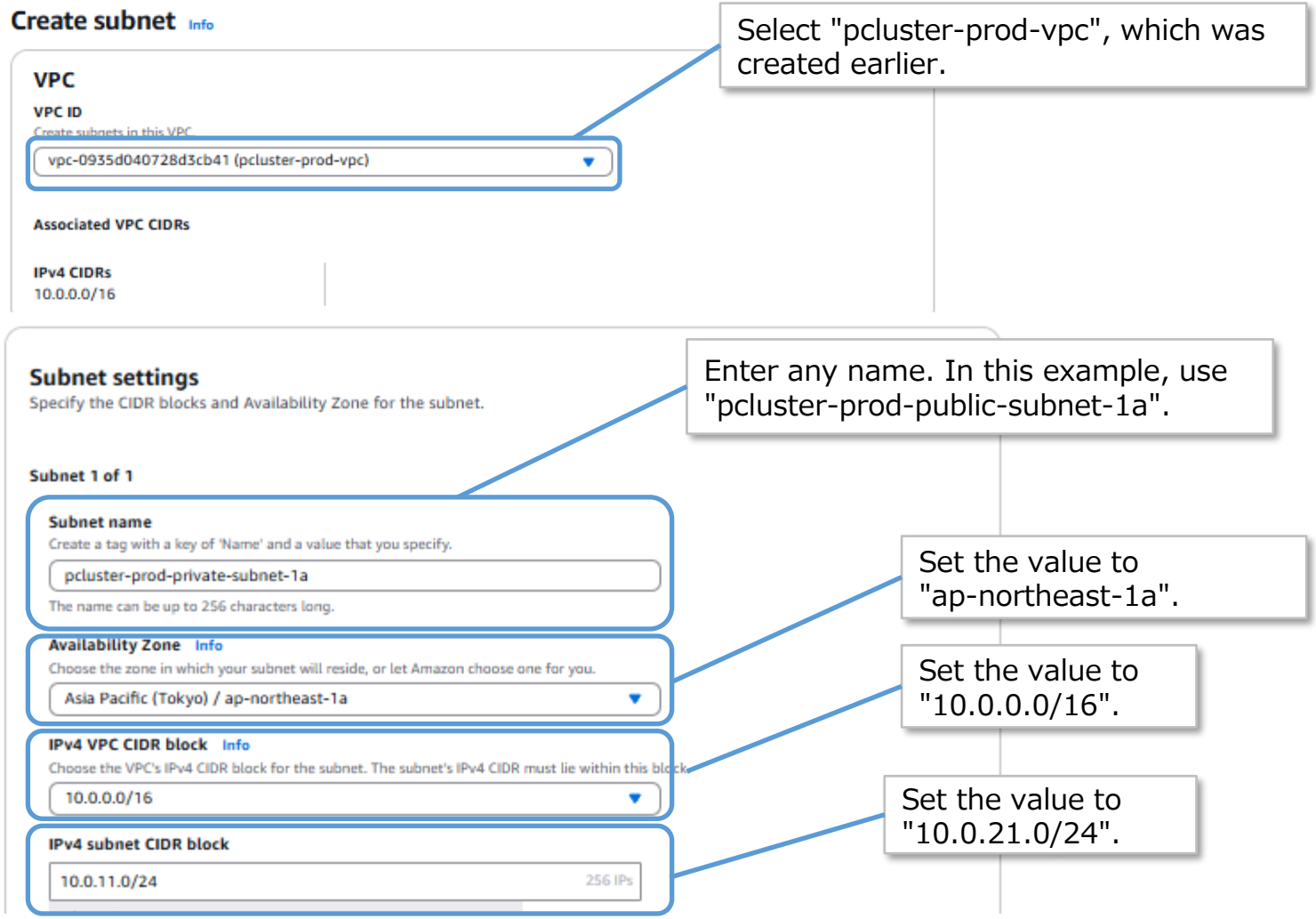
4.2 Click "Create Subnet".



Network Configuration

4. Creating Public Subnet (2/3)

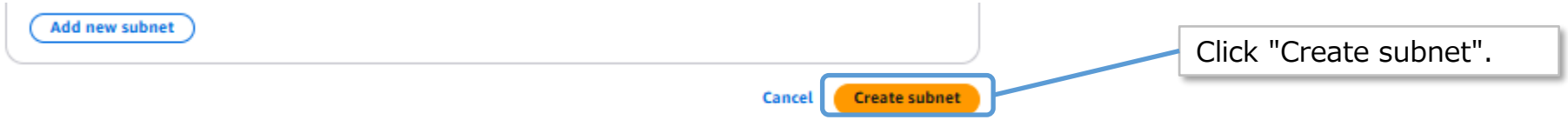
4.3 Set the parameters as follows.



Network Configuration

4. Creating Public Subnet (3/3)

4.4 Click "Create Subnet".



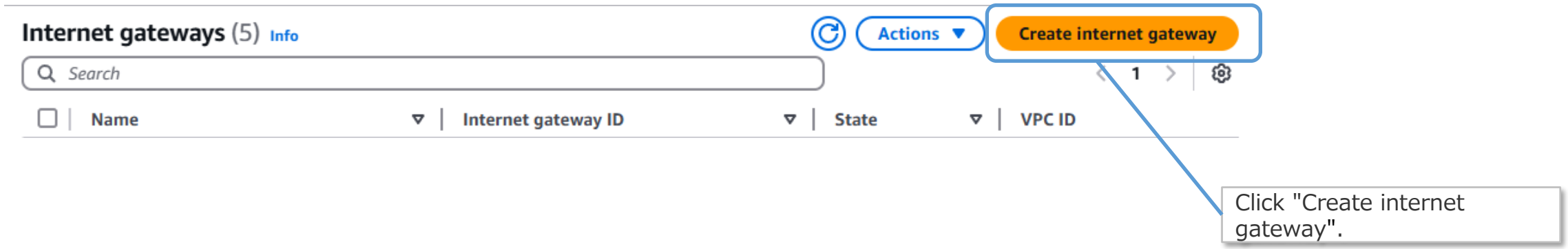
4.5 Subnet creation is complete when the following screen appears.

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/vpcconsole/home?region=ap-northeast-1#igws

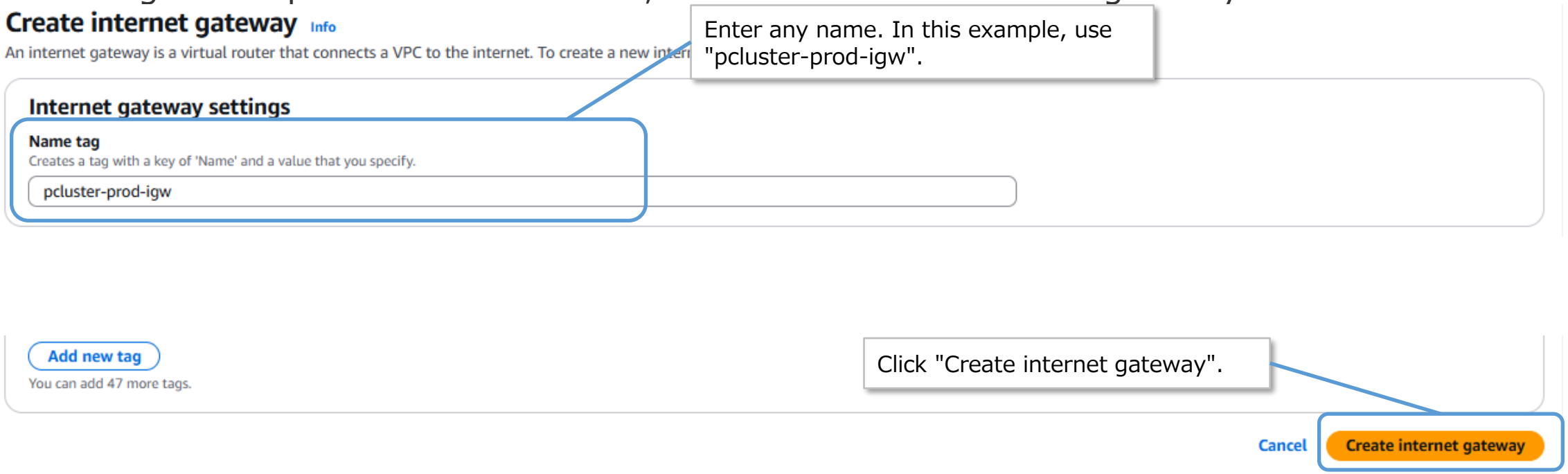
5.2 Click "Create internet gateway".



Network Configuration

5. Creating Internet Gateway (2/2)

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



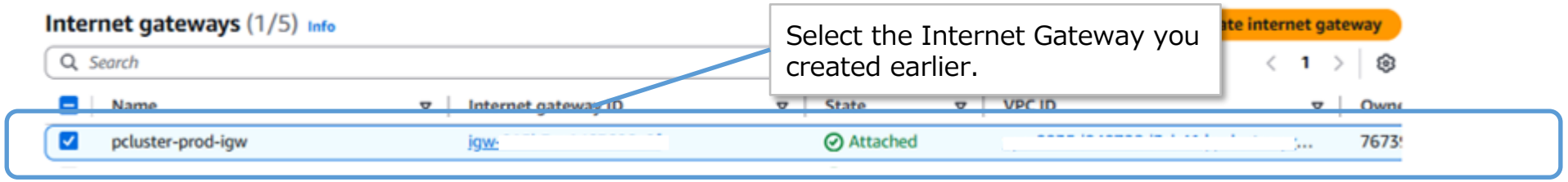
5.4 The Internet Gateway is successfully created when the following message appears on the screen.

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/vpcconsole/home?region=ap-northeast-1#igws

6.2 Select the Internet Gateway you created earlier.

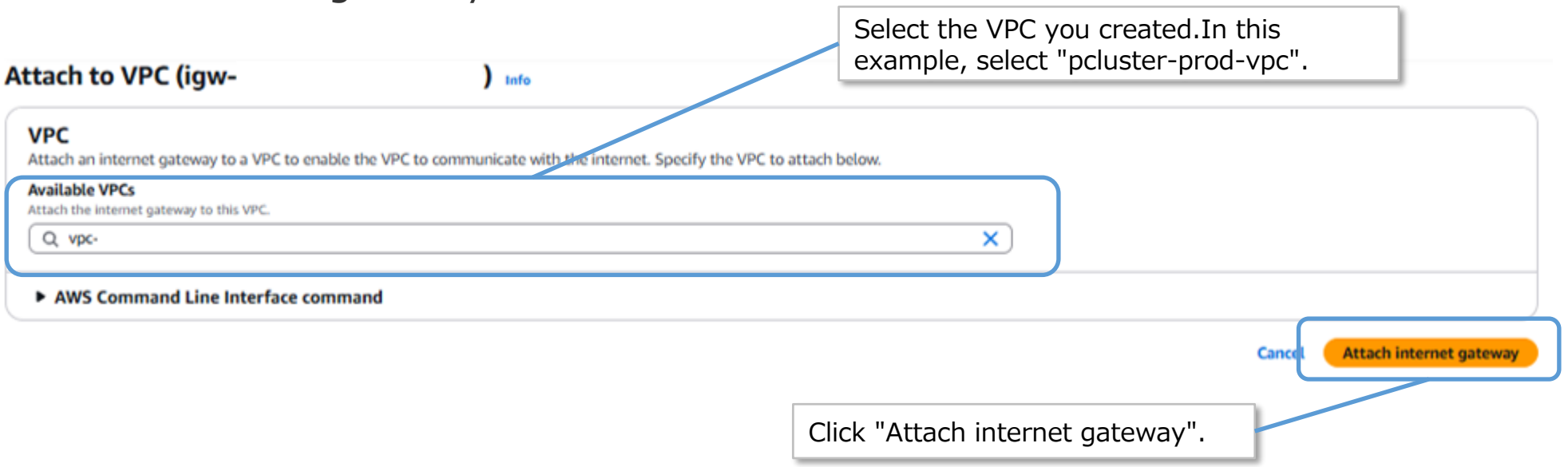


6.3 Click "Actions" in the upper-right corner, then select "Attach to VPC".

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".



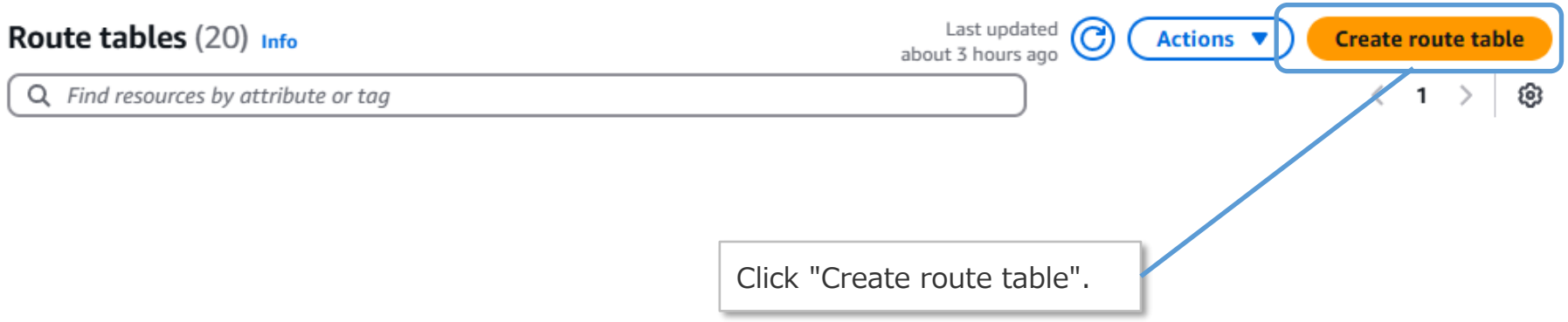
6.5 The Internet Gateway is successfully attached to the VPC when the following message appears on the screen.

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

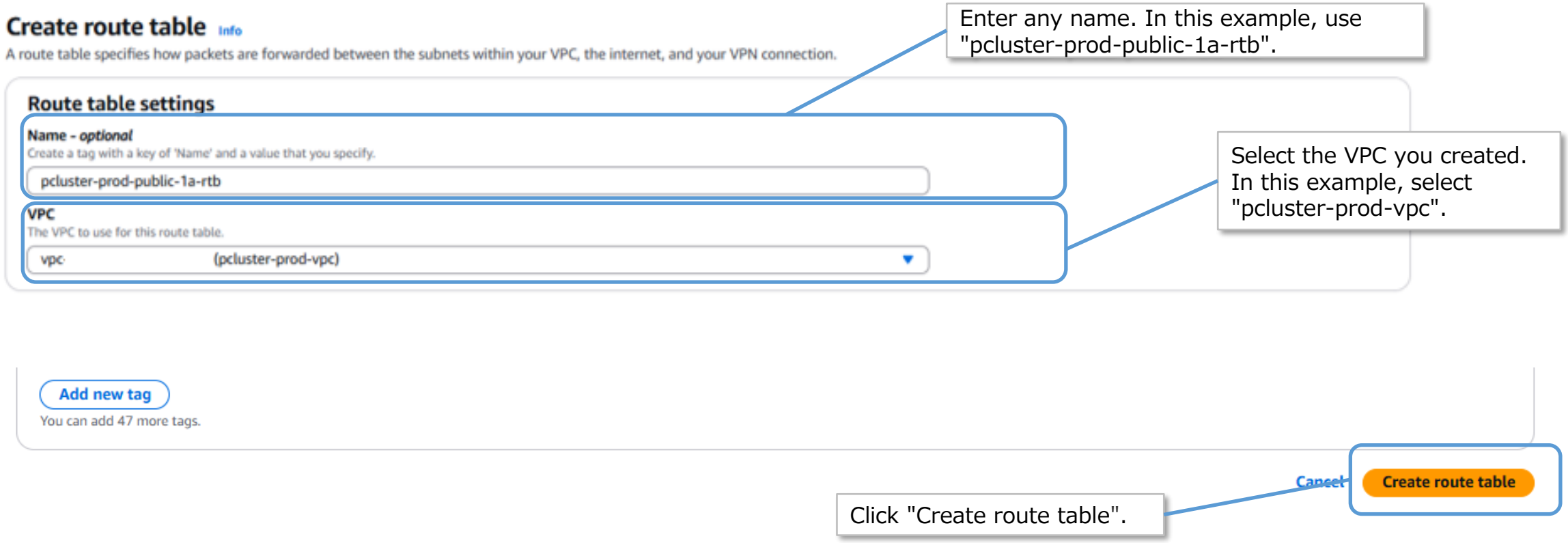
7.2 Click "Create route table".



Network Configuration

7. Creating Route Table (2/3)

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



Network Configuration

7. Creating Route Table (3/3)

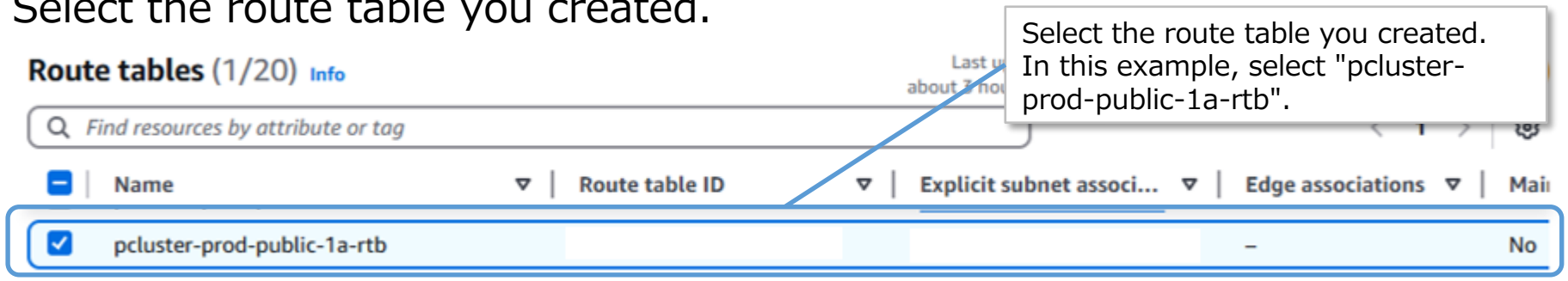
7.4 The route table creation is complete when the following message appears on the screen.

Network Configuration

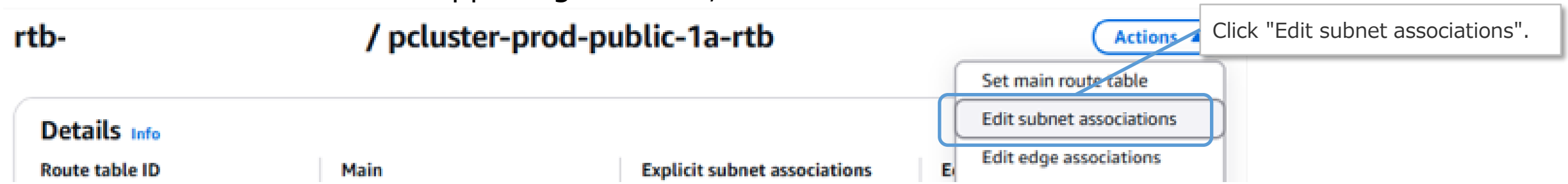
8. Associating Route Table (1/2)

8.1 In the AWS Management Console, navigate to the Route Tables list page.
https://ap-northeast-1.console.aws.amazon.com/vpcconsole/home?region=ap-northeast-1#RouteTables

8.2 Select the route table you created.



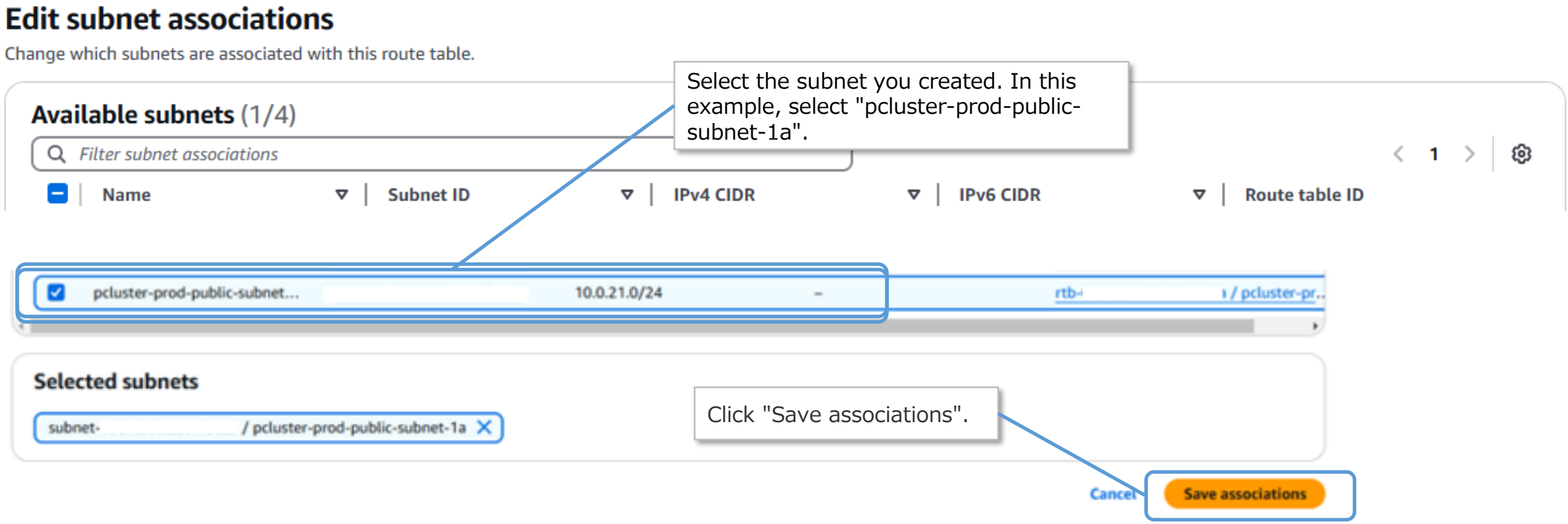
8.3 Click "Actions" in the upper-right corner, then select "Edit subnet associations".



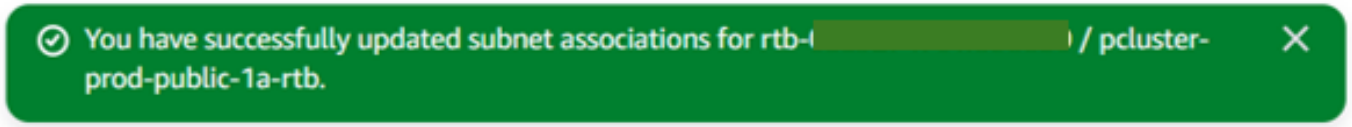
Network Configuration

8. Associating Route Table (2/2)

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



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



Network Configuration

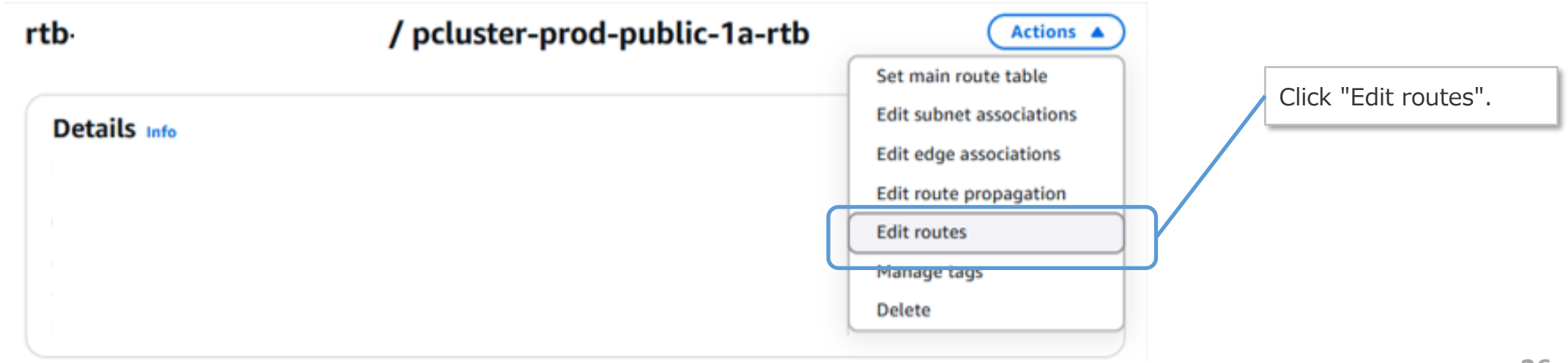
9. Adding Route Entry (1/3)

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

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

9.2 Select the route table you created.

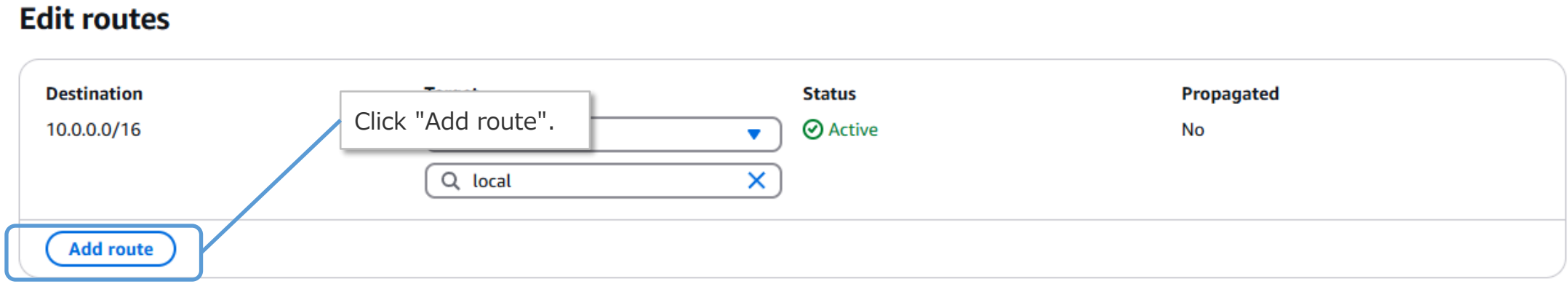
9.3 In the upper-right corner, click "Actions", then select "Edit routes".



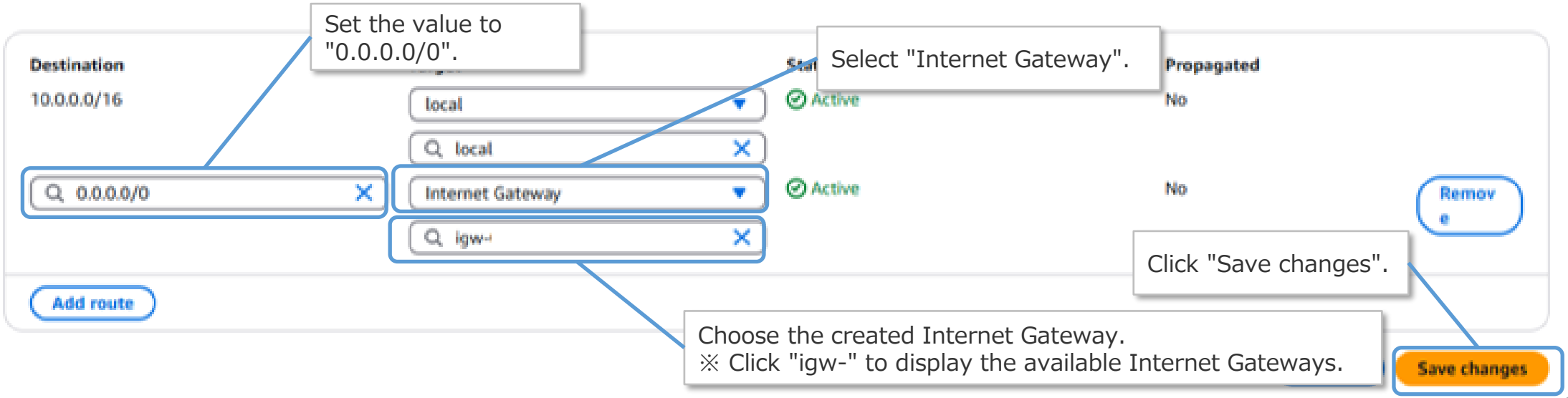
Network Configuration

9. Adding Route Entry (2/3)

9.4 Click "Add route".



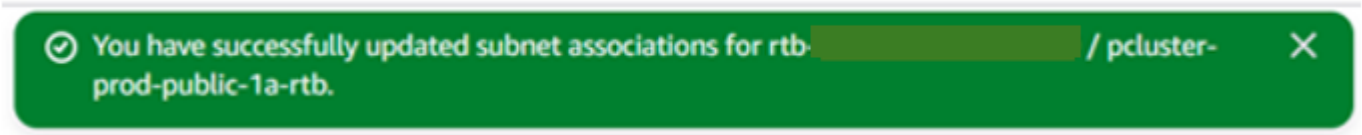
9.5 "Configure the parameters as follows, then click "Save changes".



Network Configuration

9. Adding Route Entry (3/3)

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

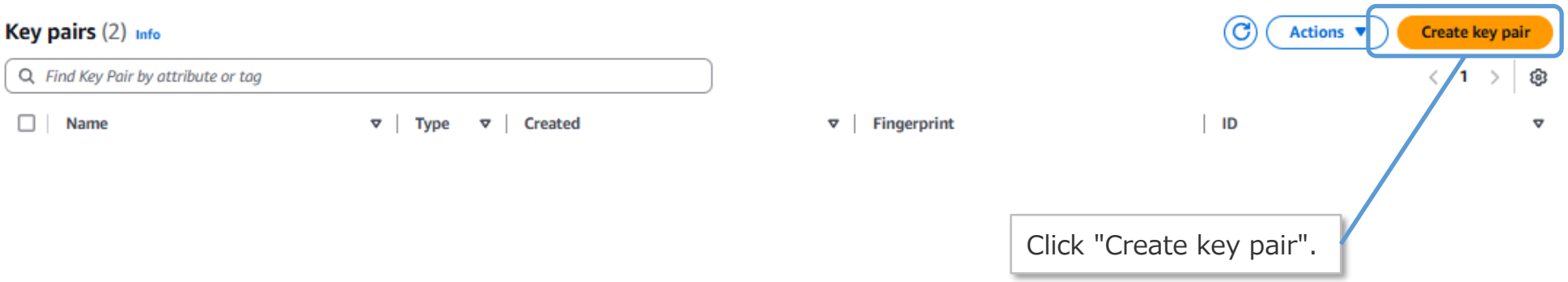


Network Configuration

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

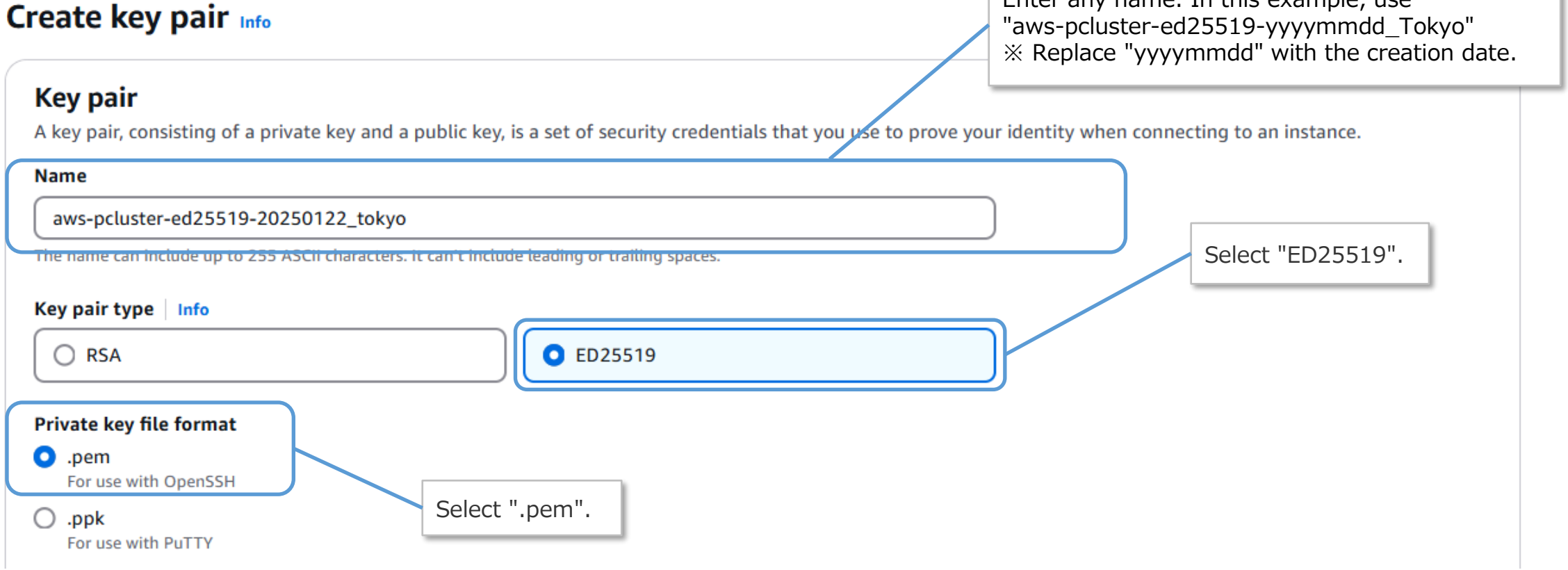
10.2 Click "Create key pair".



Network Configuration

10. Creating Key Pair (2/3)

10.3 Configure the parameters as follows.



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.

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

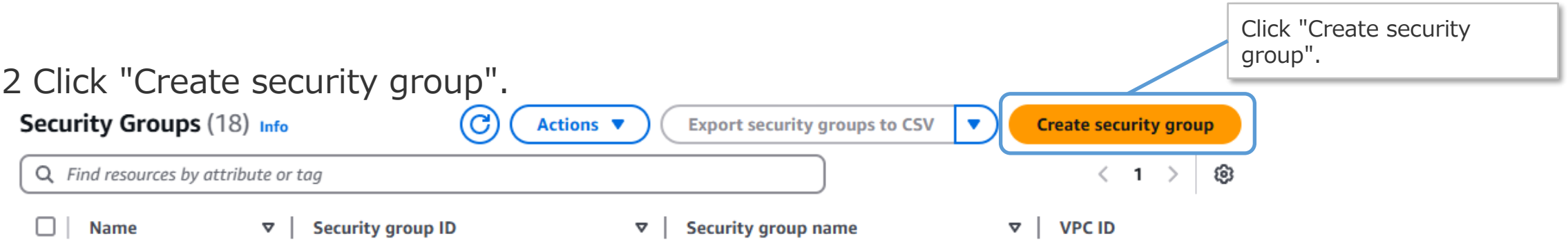
Network Configuration

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

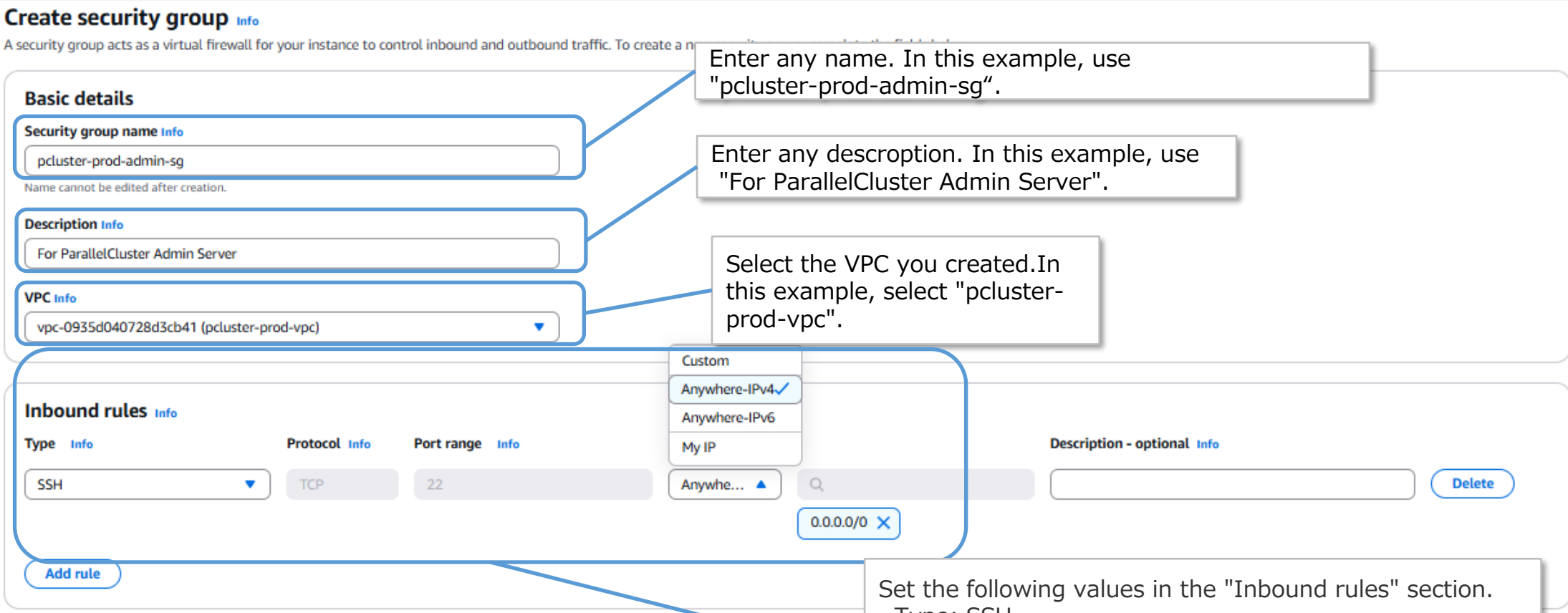
11.2 Click "Create security group".



Network Configuration

11. Creating Security Group (2/4)

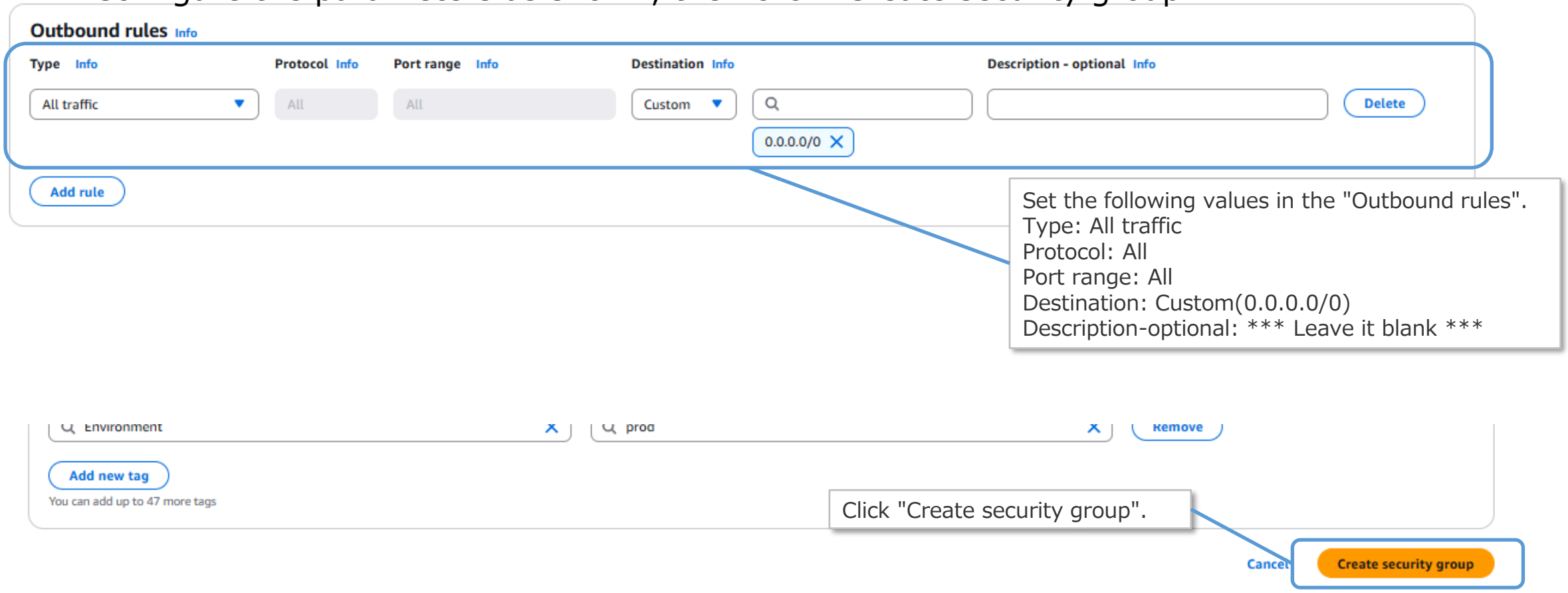
11.3 Configure the parameters as follows.



Network Configuration

11. Creating Security Group (3/4)

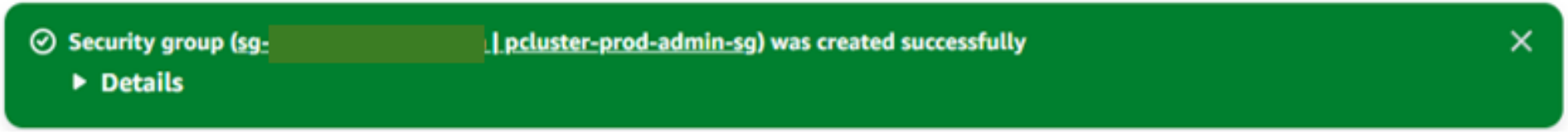
11.4 Configure the parameters as shown, then click "Create security group".



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

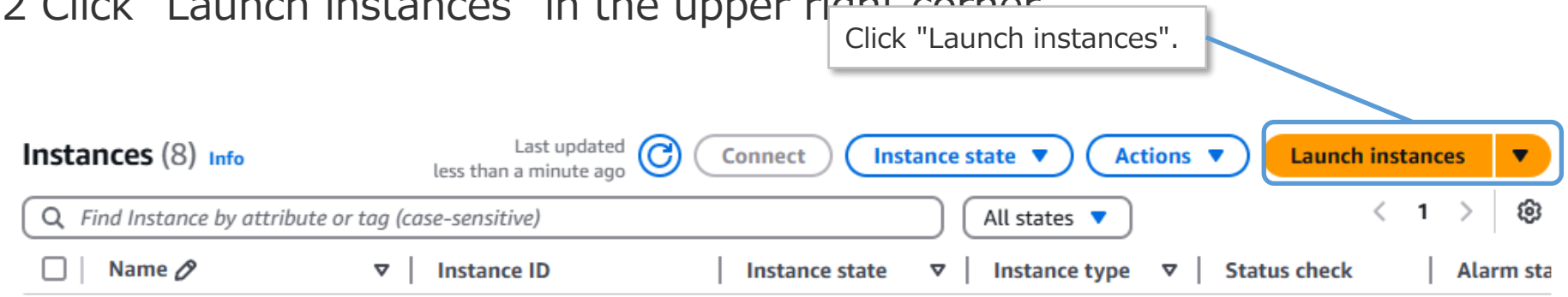
Creating ParallelCluster Setup Instance

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

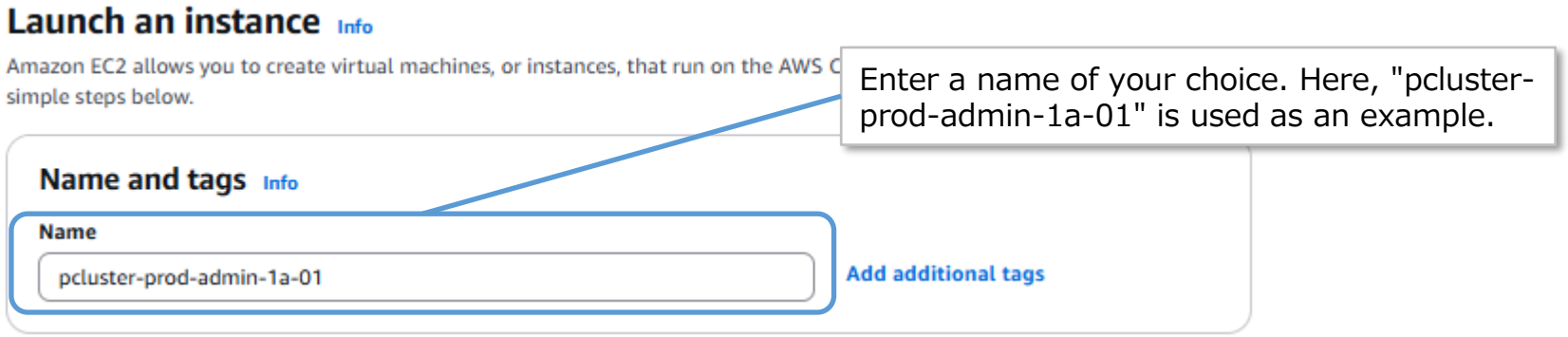
1.2 Click "Launch instances" in the upper right corner.



Creating ParallelCluster Setup Instance

1. Creating EC2 Instance (2/8)

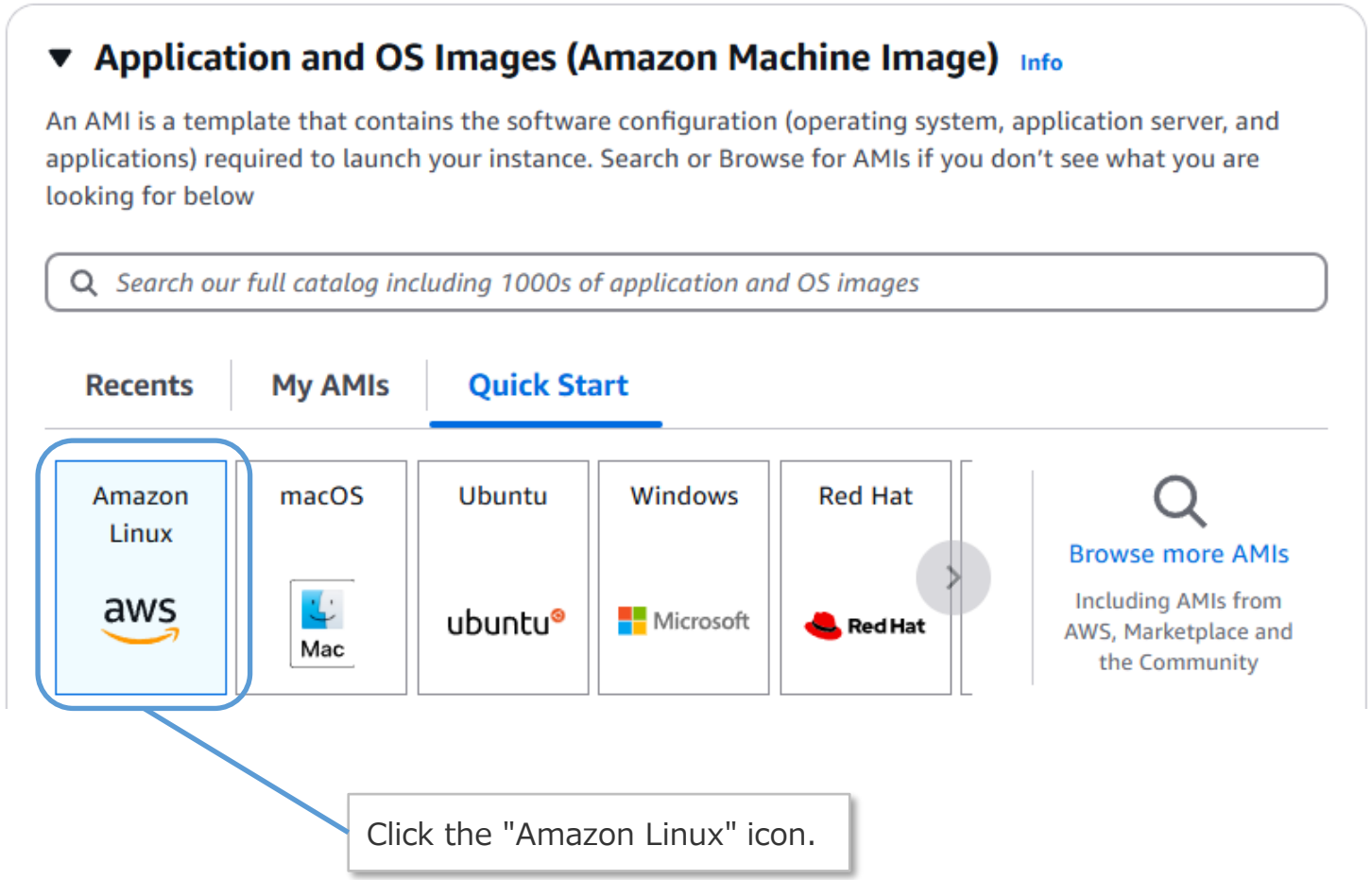
1.3 Configure the parameters as follows.



Creating ParallelCluster Setup Instance

1. Creating EC2 Instance (3/8)

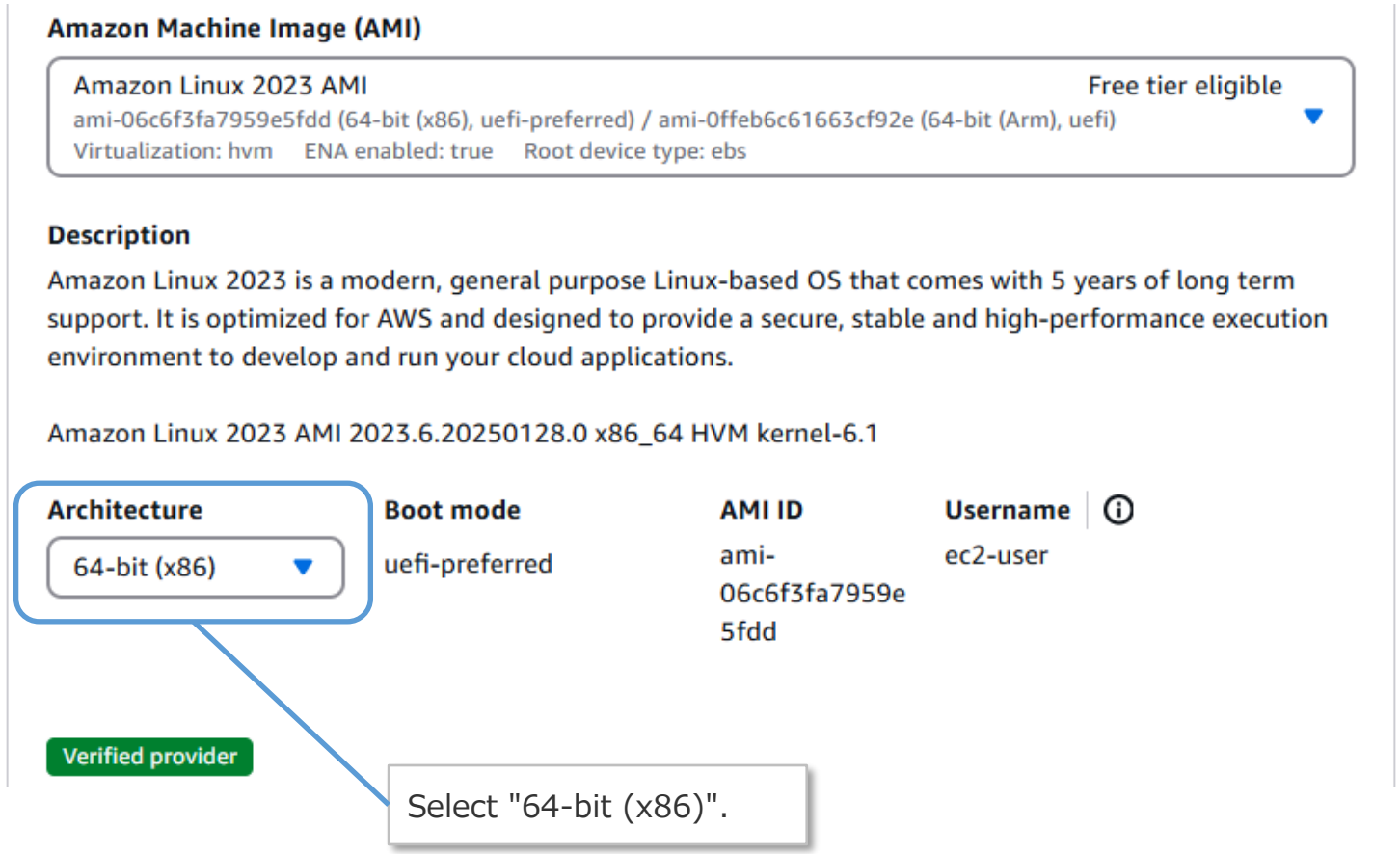
1.4 Configure the parameters as follows.



Creating ParallelCluster Setup Instance

1. Creating EC2 Instance (4/8)

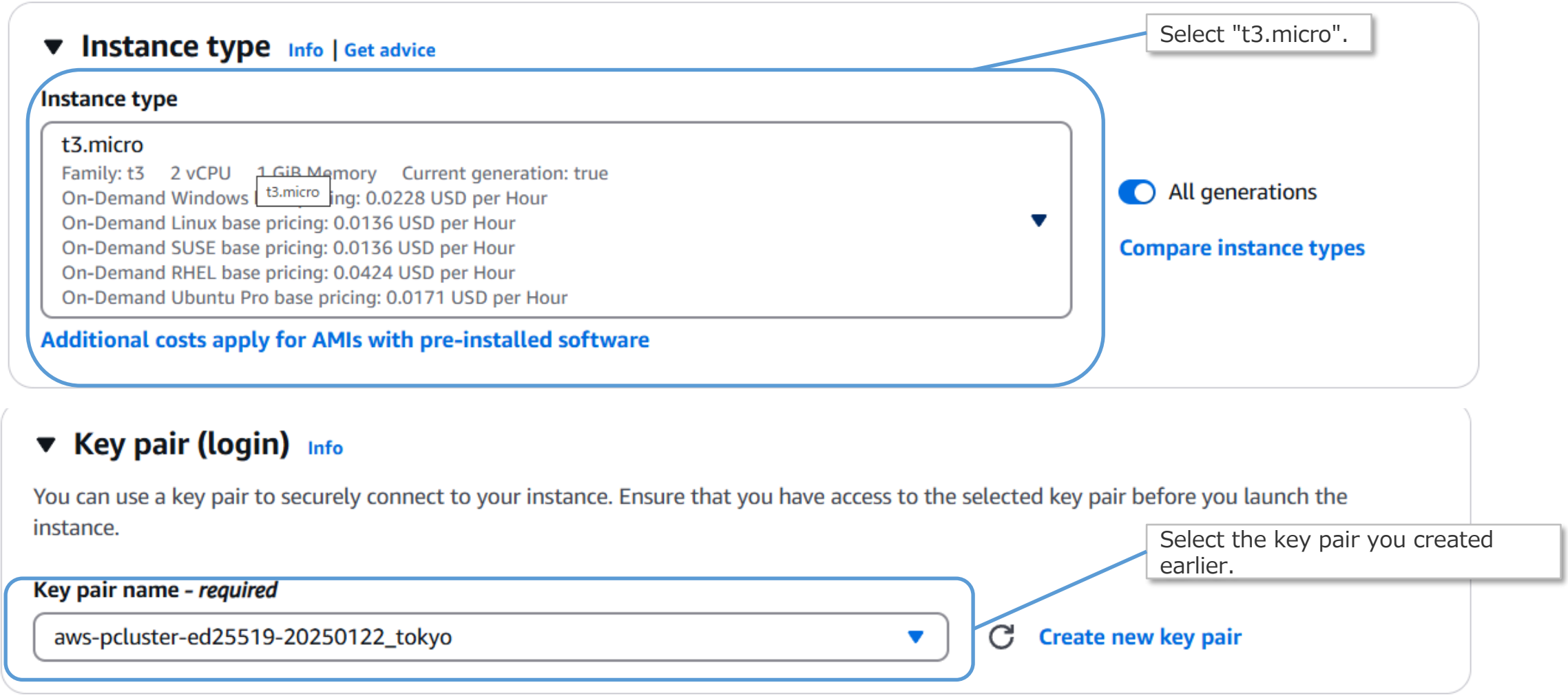
1.5 Configure the parameters as follows.



Creating ParallelCluster Setup Instance

1. Creating EC2 Instance (5/8)

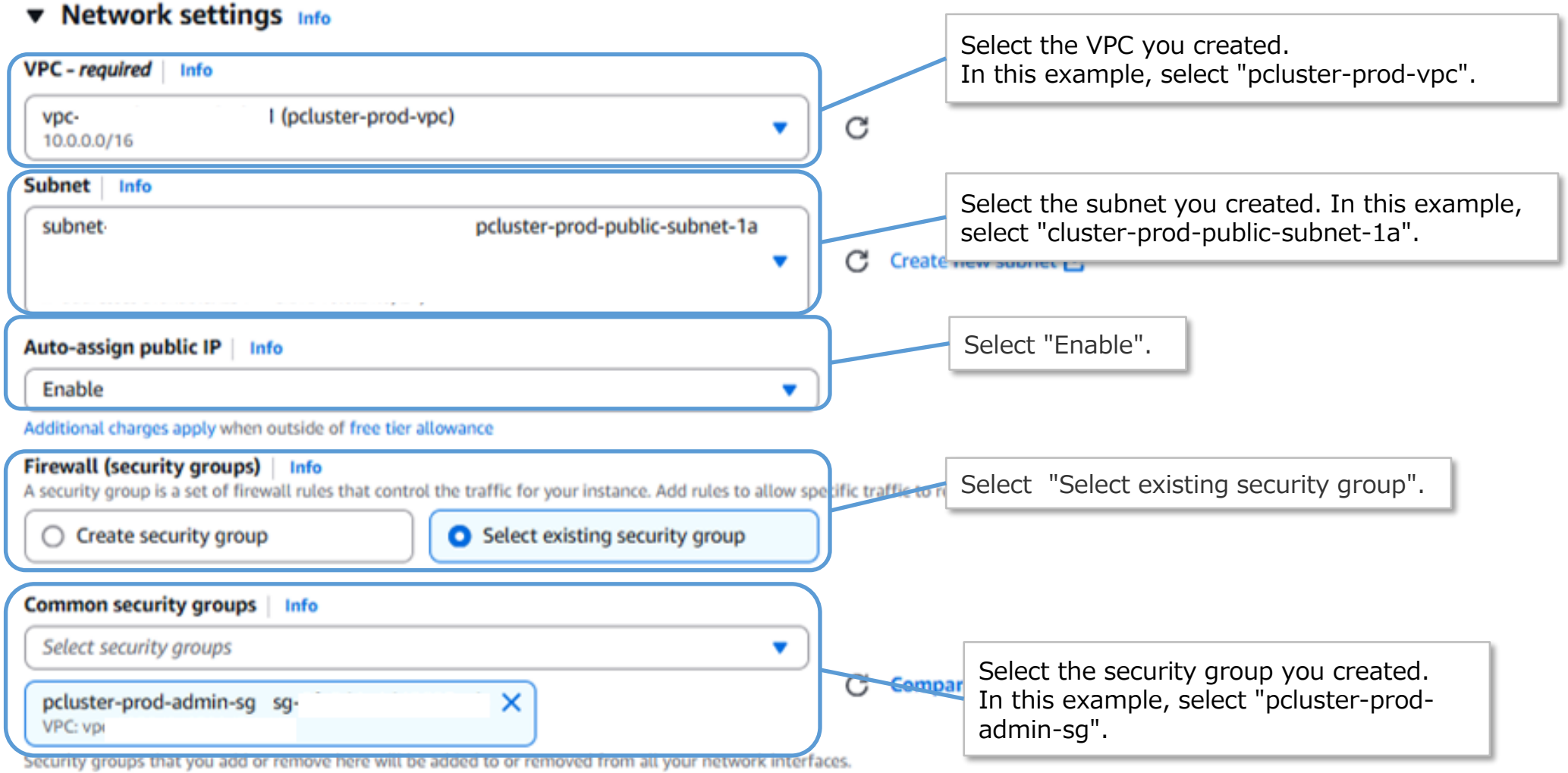
1.6 Configure the parameters as follows.



Creating ParallelCluster Setup Instance

1. Creating EC2 Instance (6/8)

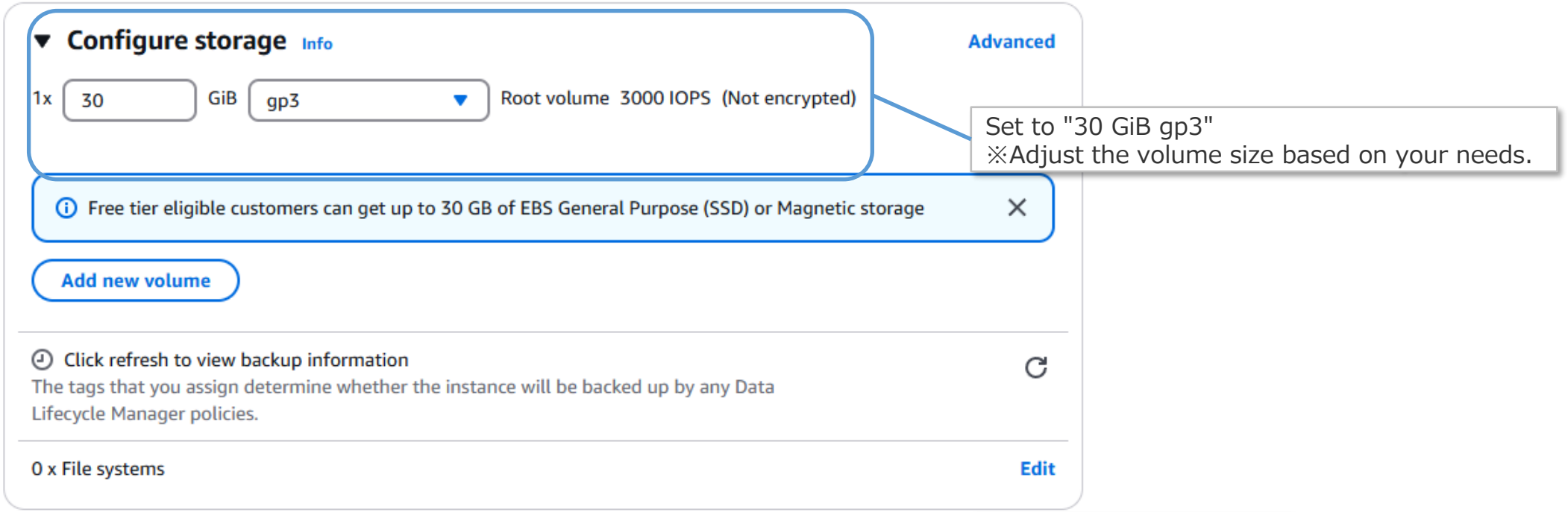
1.7 Configure the parameters as follows.



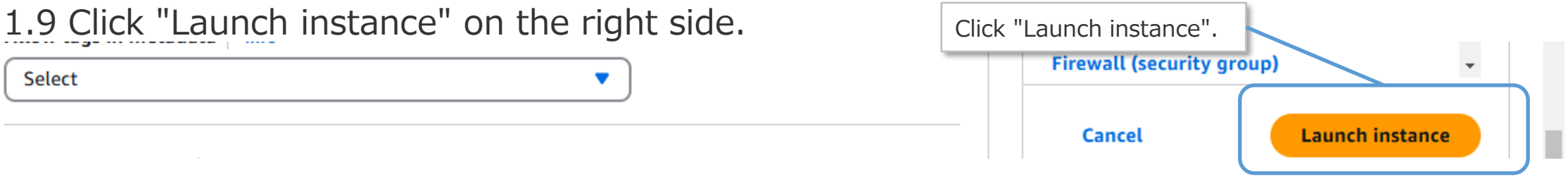
Creating ParallelCluster Setup Instance

1. Creating EC2 Instance (7/8)

1.8 Configure the parameters as follows.



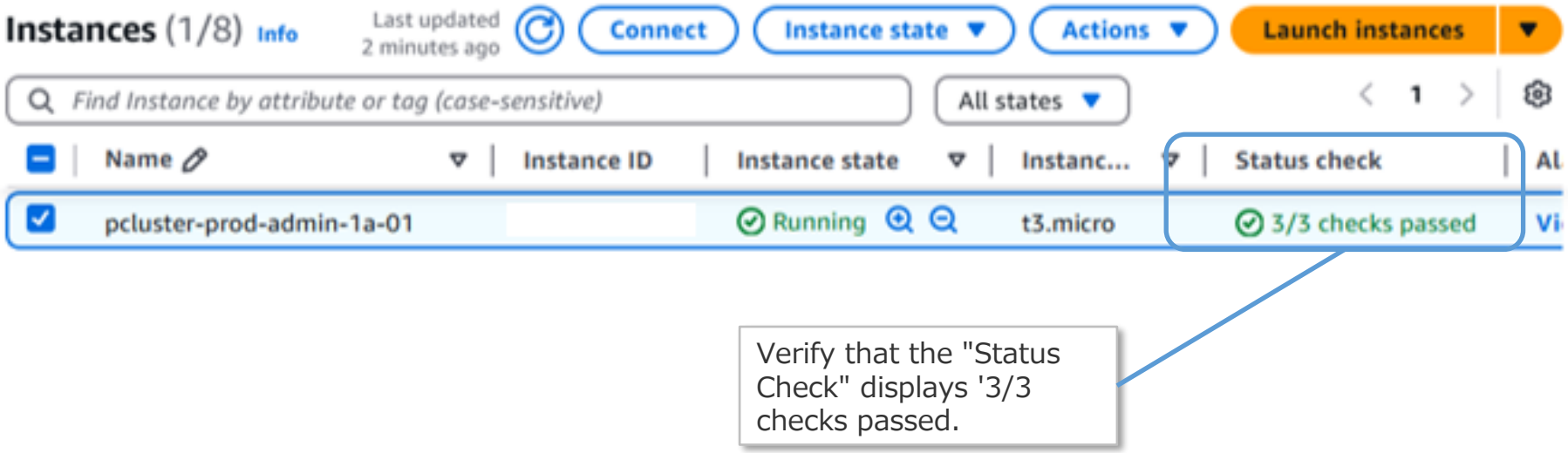
1.9 Click "Launch instance" on the right side.



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.



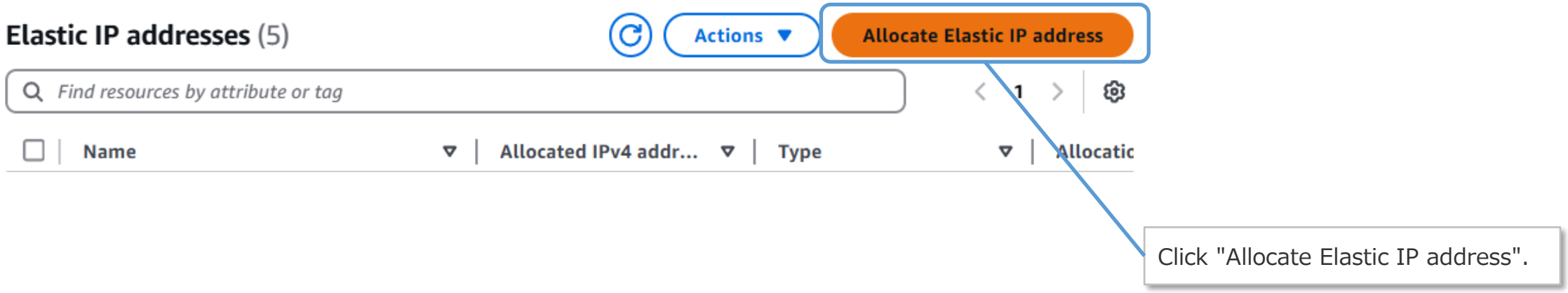
Creating ParallelCluster Setup Instance

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

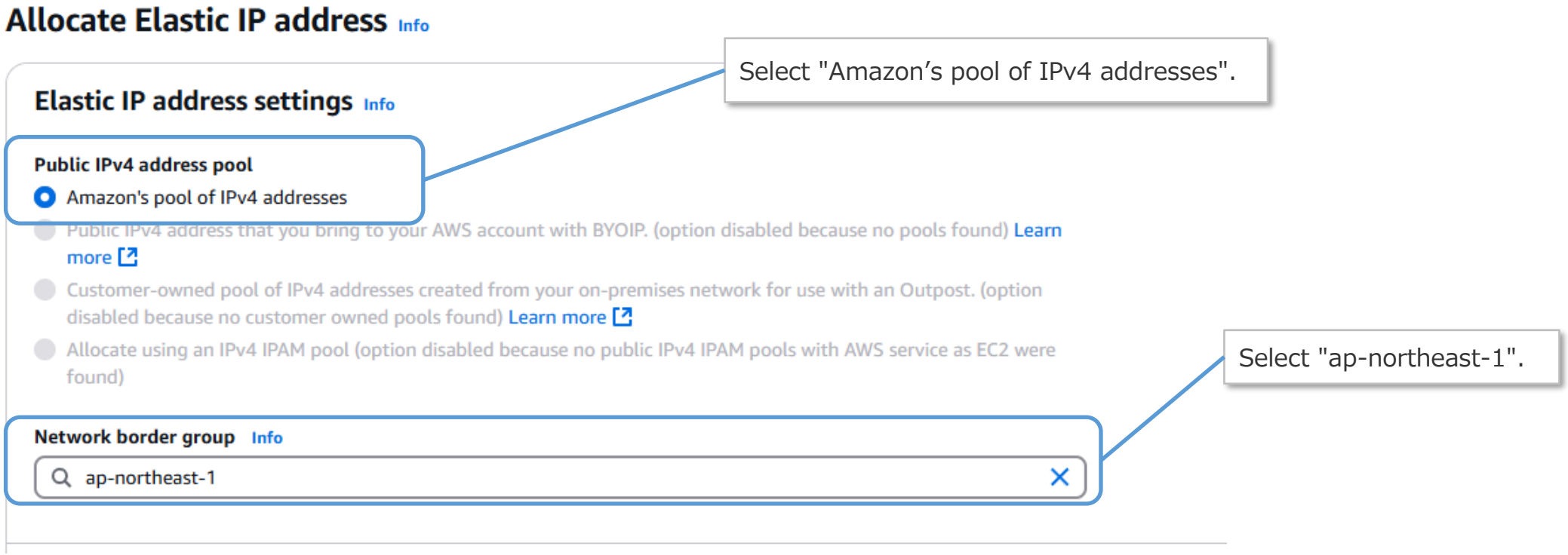
2.2 Click "Allocate Elastic IP address" in the upper right corner.



Creating ParallelCluster Setup Instance

2. Allocating Elastic IP Address (2/3)

2.3 Configure the parameters as follows.



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.

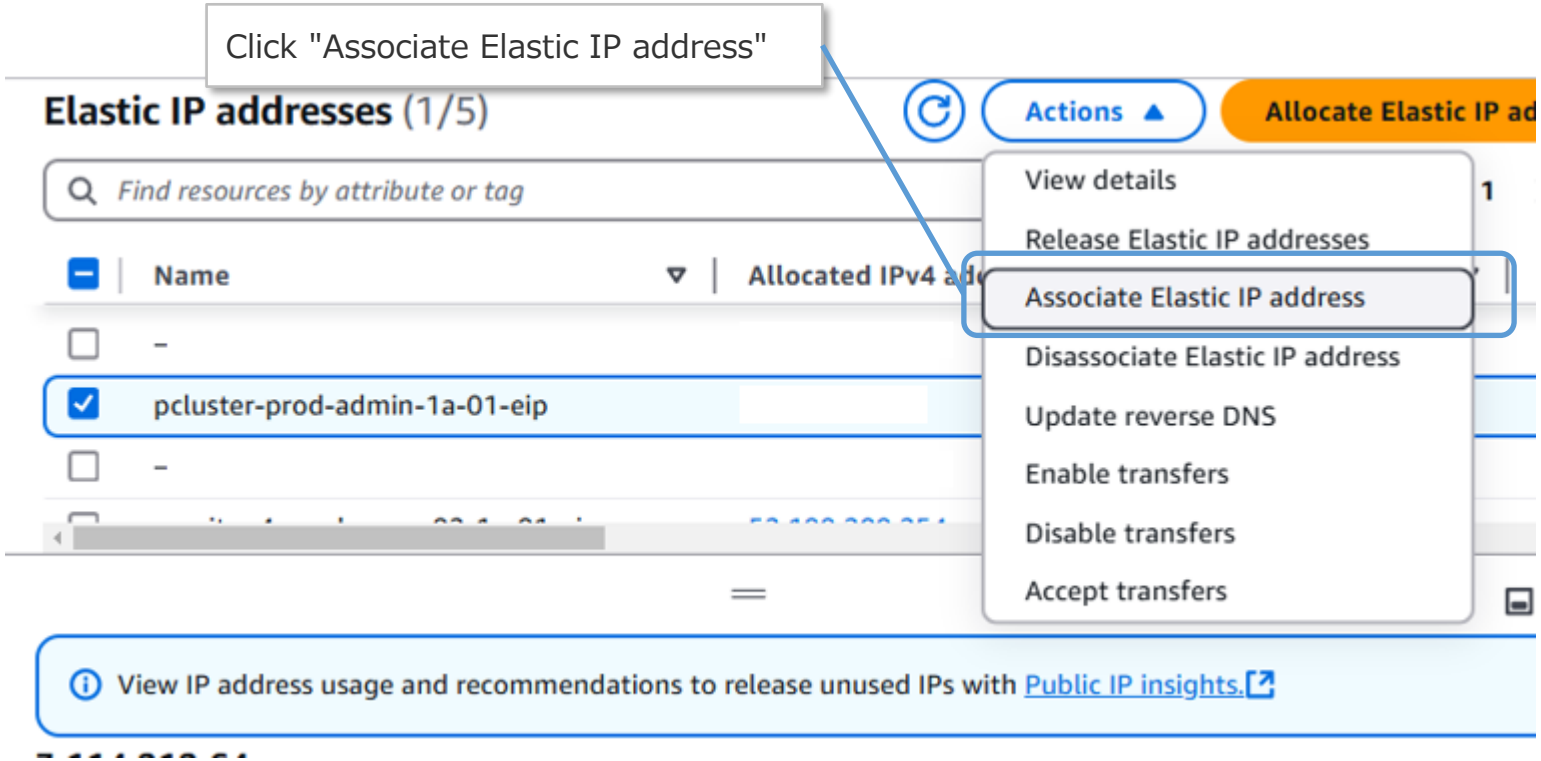
Creating ParallelCluster Setup Instance

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

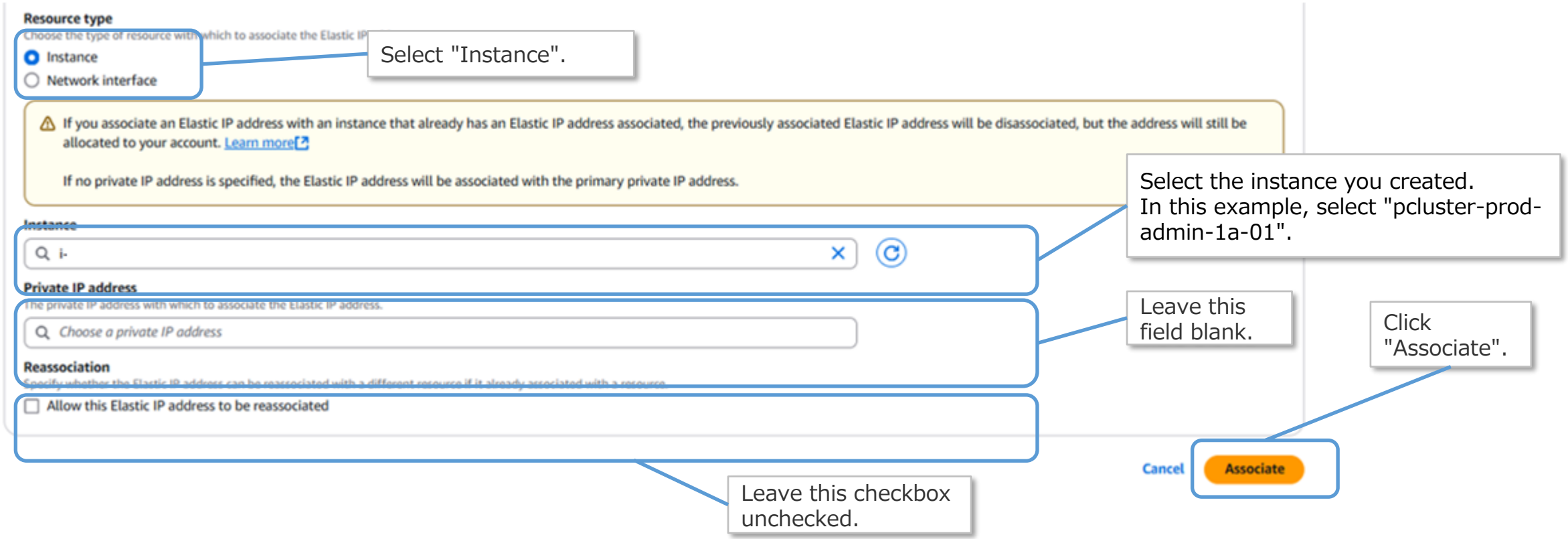
3.2 Select the EIP to assign to the EC2 instance, then click "Actions" in the top right corner and choose "Associate Elastic IP address".



Creating ParallelCluster Setup Instance

3. Associating Elastic IP Address (2/2)

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



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

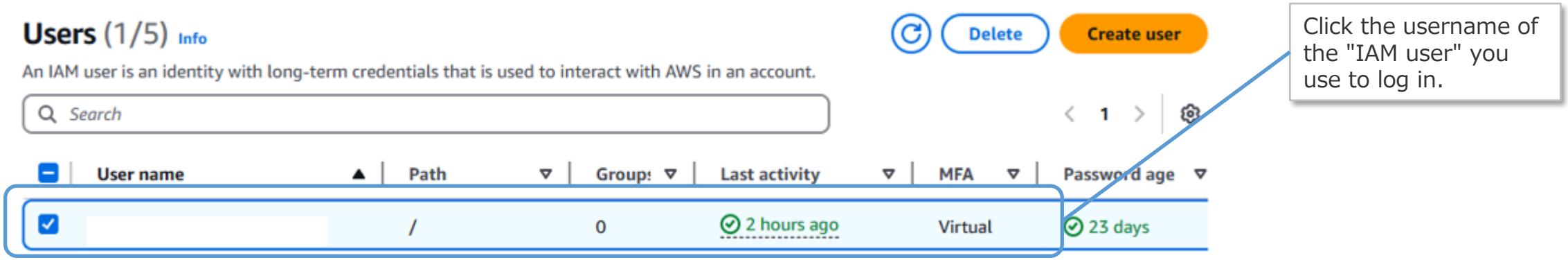
Creating ParallelCluster Setup Instance

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.

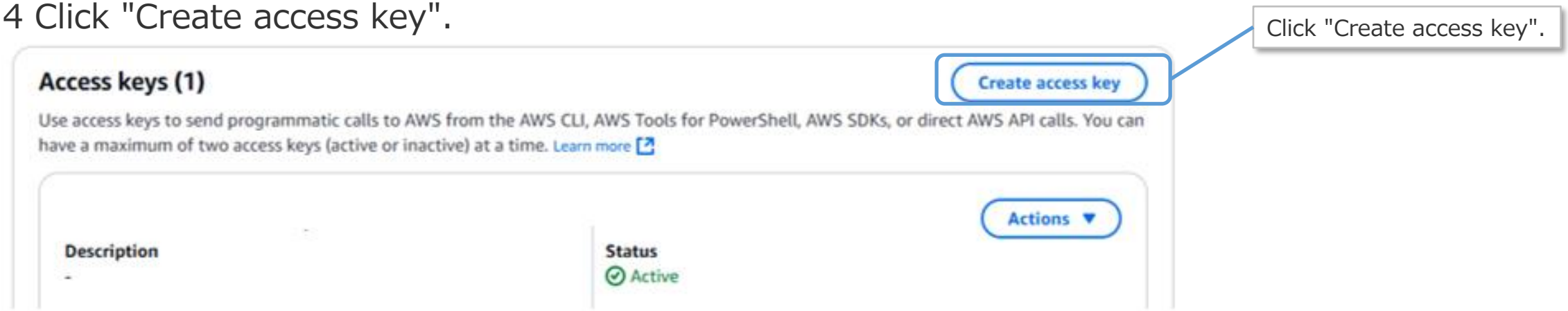


Creating ParallelCluster Setup Instance

4. Creating Access Key (2/4)

4.3 Click "Security credentials".

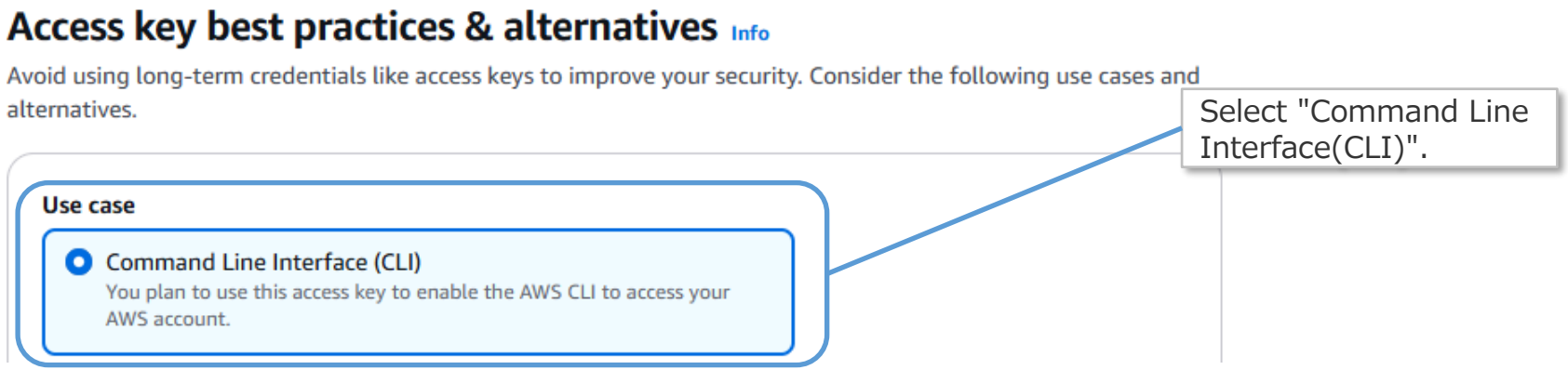
4.4 Click "Create access key".



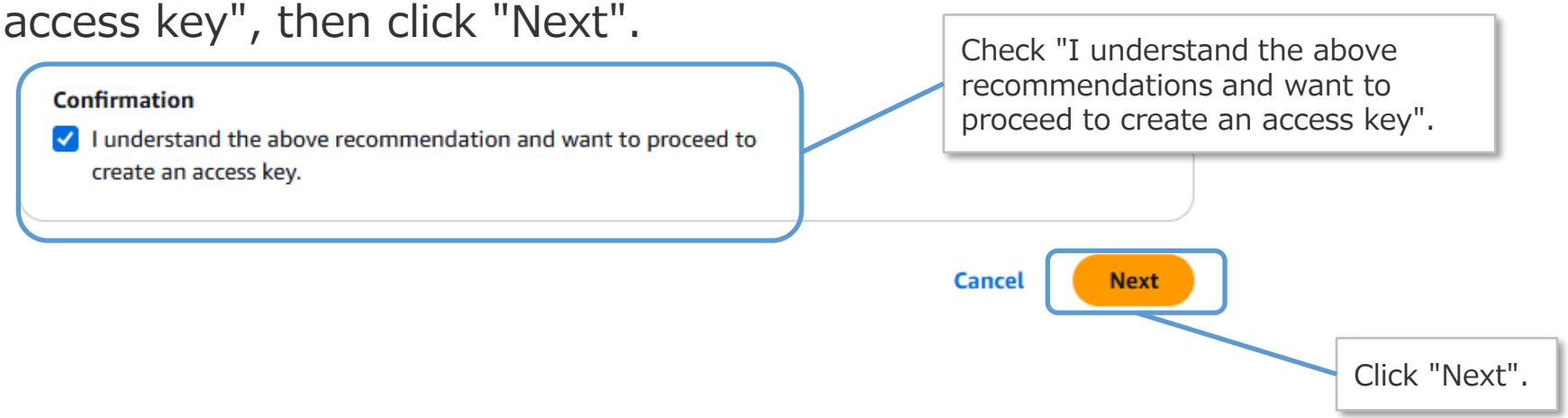
Creating ParallelCluster Setup Instance

4. Creating Access Key (3/4)

4.5 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".

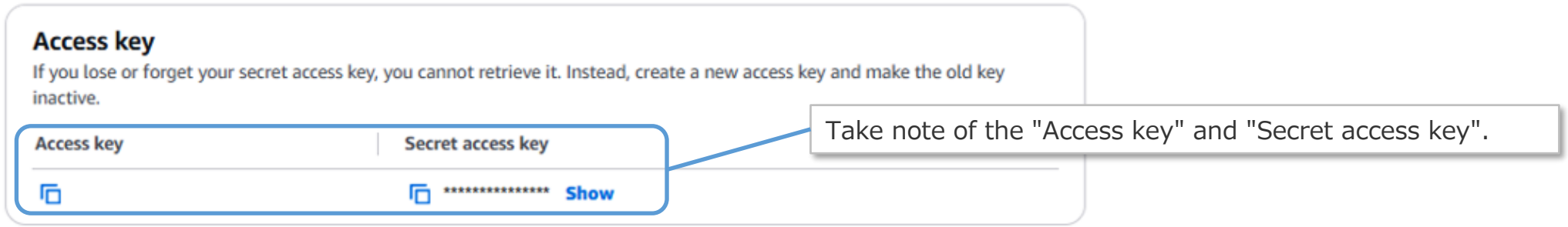


Creating ParallelCluster Setup Instance

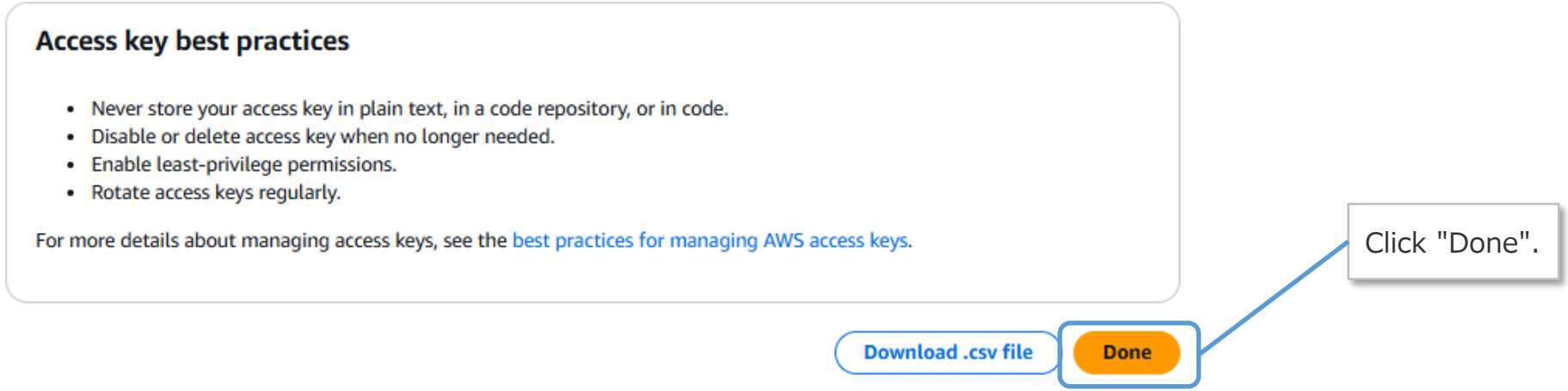
4. Creating Access Key (4/4)

4.7 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.



4.9 Click "Done".



Creating ParallelCluster Setup Instance

Logging into the ParallelCluster Setup Instance

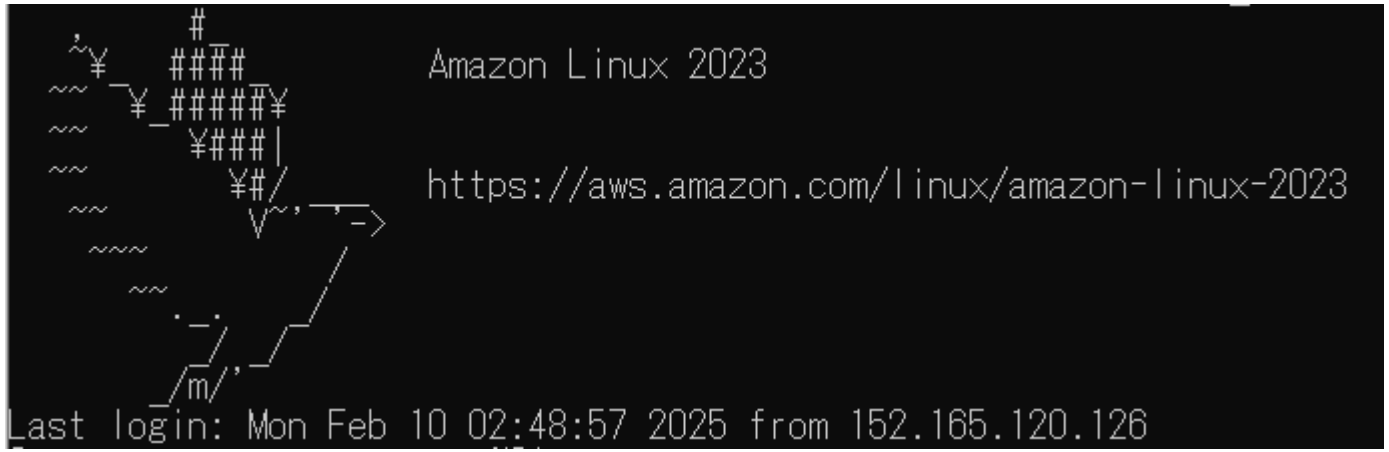
To log into the ParallelCluster instance, establish an SSH connection to its public IP address.

1. Run the following command to connect to the ParallelCluster execution environment via SSH.

```
$ ssh -i [SSH Authentication Key] [User Name]@[Public IP Address]
```

SSH Connection Configuration	
Host Name	Public IP Address of the ParallelCluster Setup Instance
User Name	ec2-user
Password	*** Leave it blank ***
SSH Authentication Key	Example:"aws-pcluster-ed25519-xxxxxxxx_tokyo.pem" ※The private key created in "Create Key Pair" section

2. Login is successful when the following message appears in the terminal.



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

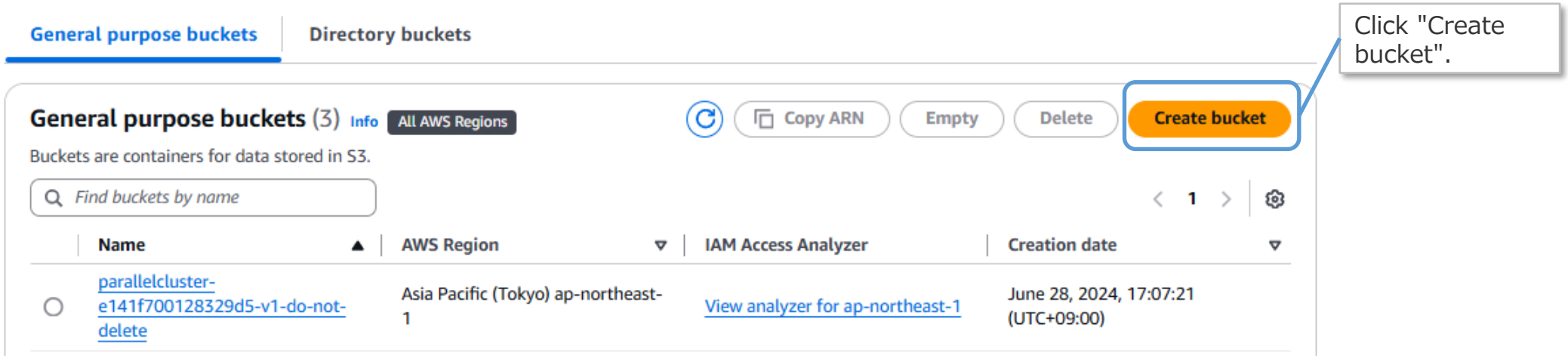
In the cluster environment set up according to this manual, an S3 bucket serves as long-term storage for computational results and other data. This section explains how to create an S3 bucket.

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

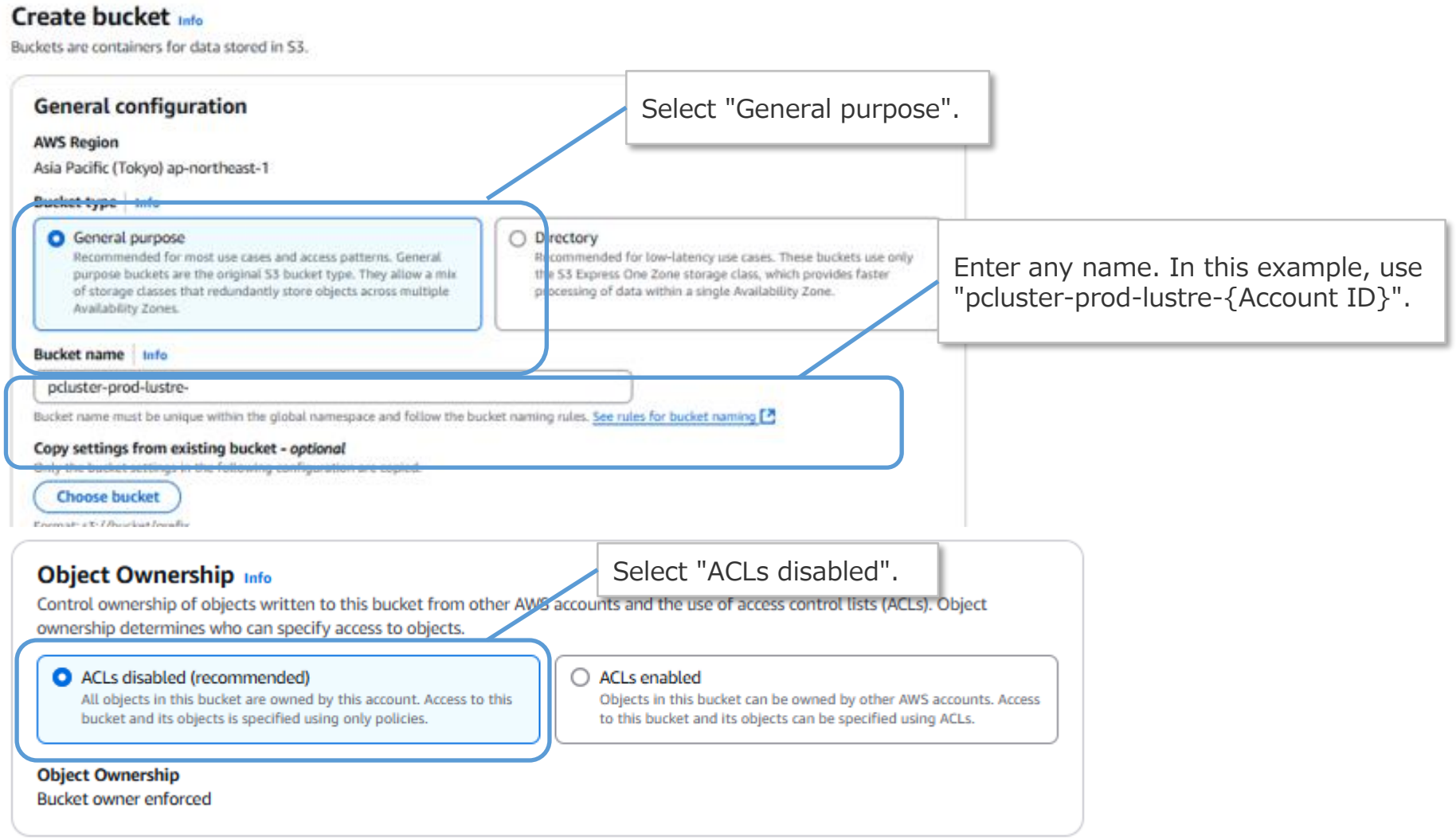
1.2 Click "Create bucket".



Creating External Storage

1. Creating S3 Bucket (2/4)

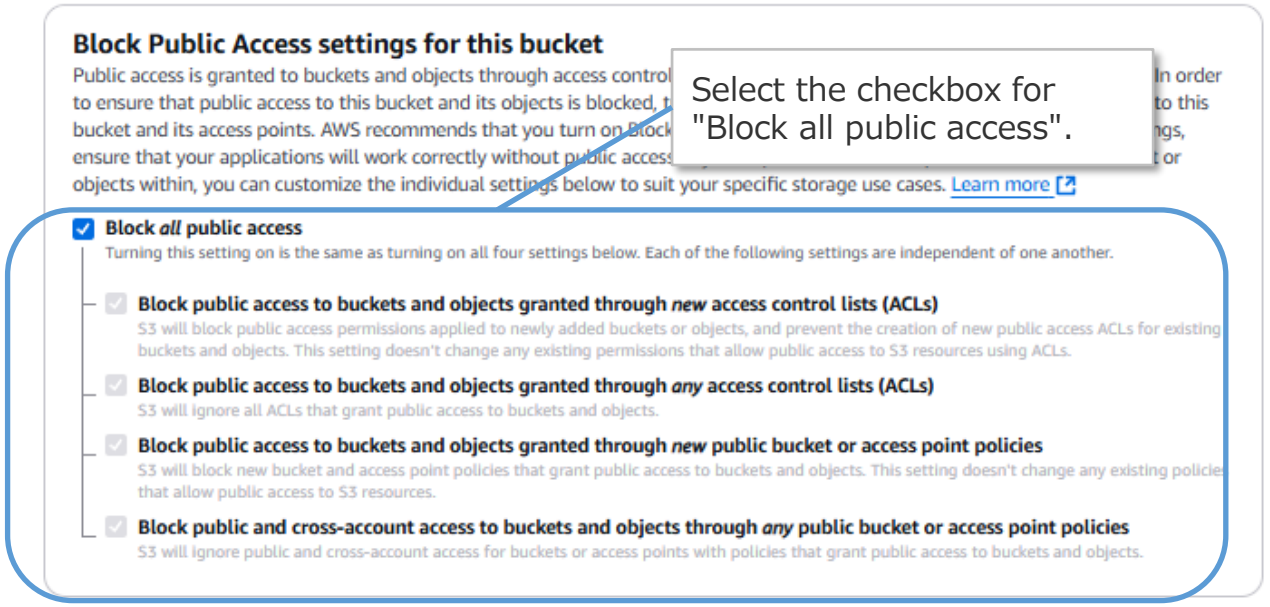
1.3 Configure the parameters as follows.



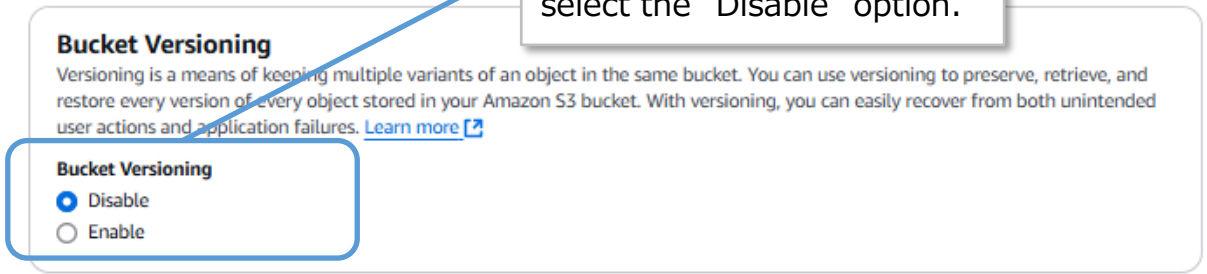
Creating External Storage

1. Creating S3 Bucket (3/4)

1.4 Configure the parameters as follows.



Select the checkbox for "Block all public access".

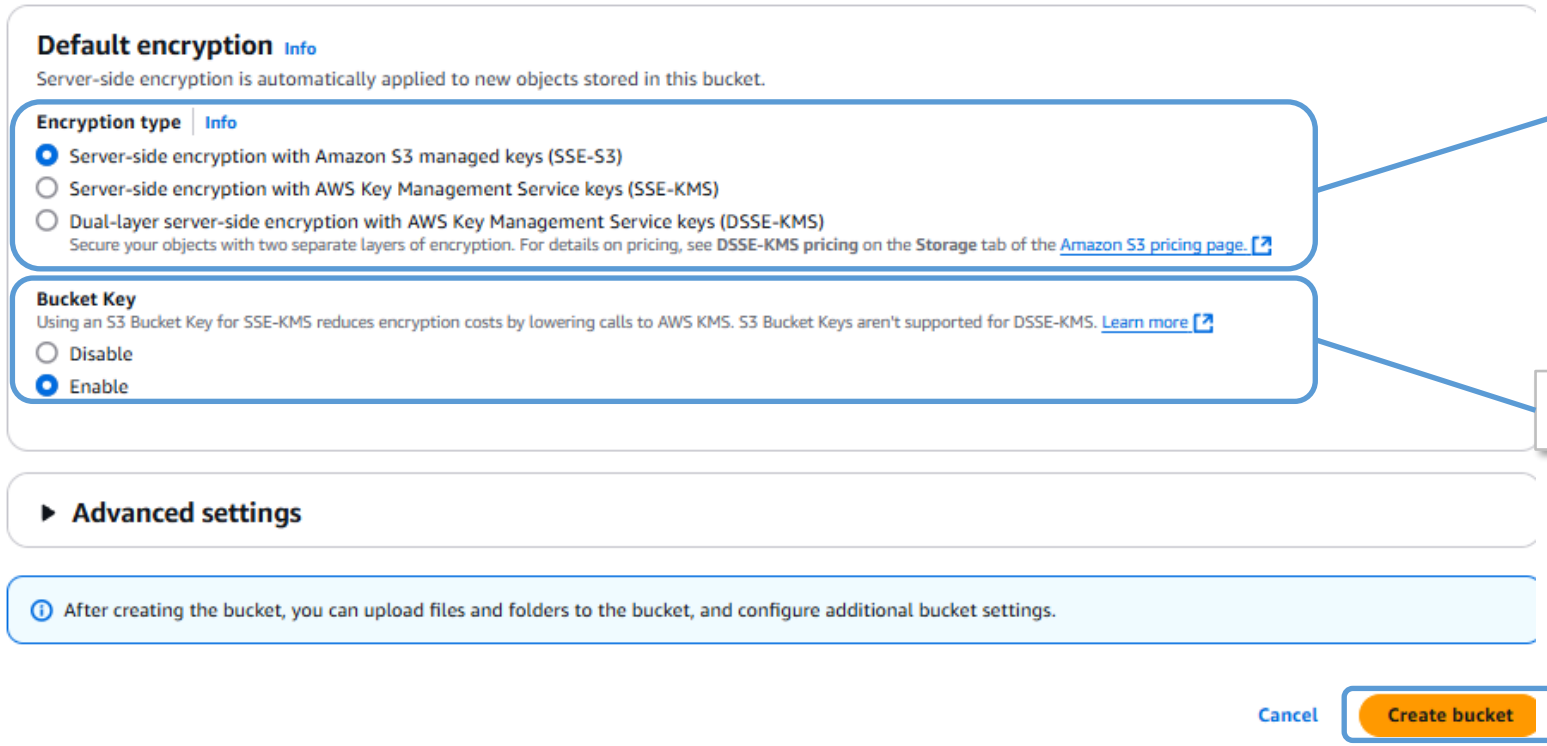


Configure this setting as needed. In this example, select the "Disable" option.

Creating External Storage

1. Creating S3 Bucket (4/4)

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

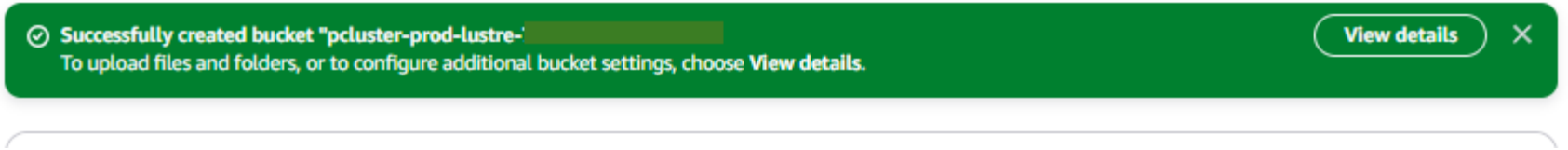


Select "Server-side encryption with Amazon S3 managed keys (SSE-S3)".

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.



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".

```
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[illegible]
Scheduling:
  Scheduler: slurm
  SlurmQueues:
  - Name: queue1
    ComputeResources:
    - Name: m7g2xlarge
      Instances:
      - InstanceType: m7g.2xlarge
      MinCount: 0
      MaxCount: 10
    Networking:
      SubnetIds:
      - subnet-0747376860b49e9b6
```

```
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}
```

Configure the storage size as needed.

Configure storage for the Head Node. Set the storage size to at least 64GB to accommodate downloading the "Virtual Fugaku" container.

Configure Shared Storage

Specify the name of the S3 bucket that you created.
Example: pcluster-prod-lustre-{Account ID}

Configuring shared storage in this file will automatically set it up and integrate it with the cluster.

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 runs 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.

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.

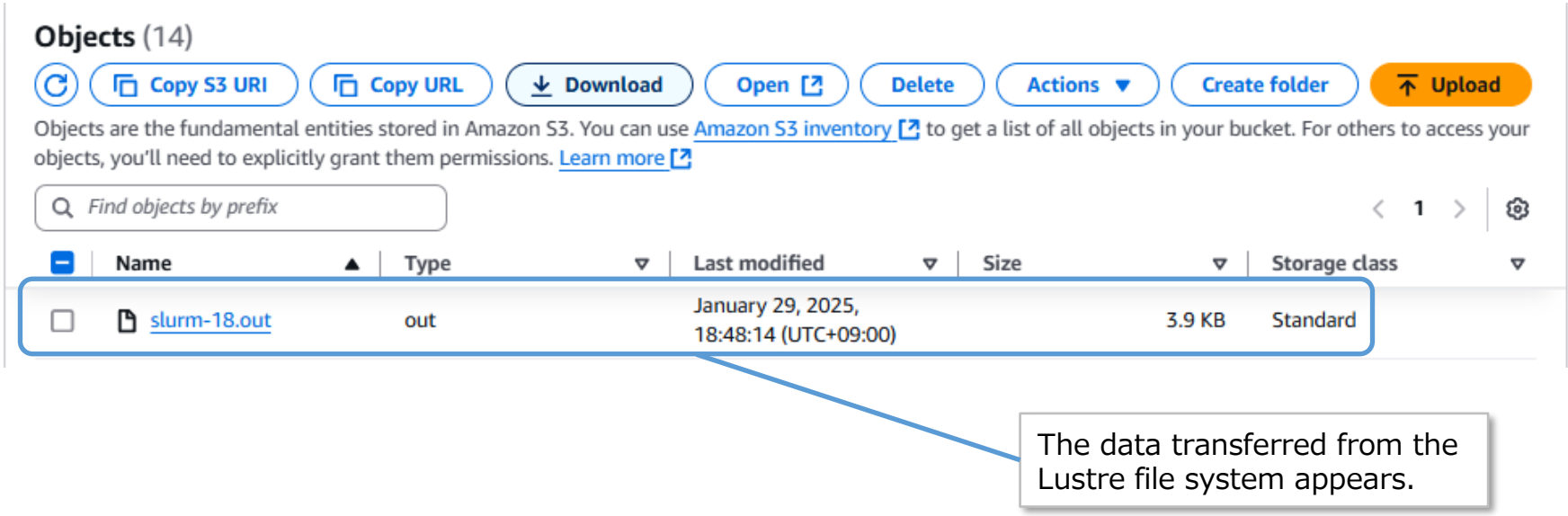
https://ap-northeast-1.console.aws.amazon.com/s3/buckets?region=ap-northeast-1&bucketType=general

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

Exporting to External Storage

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

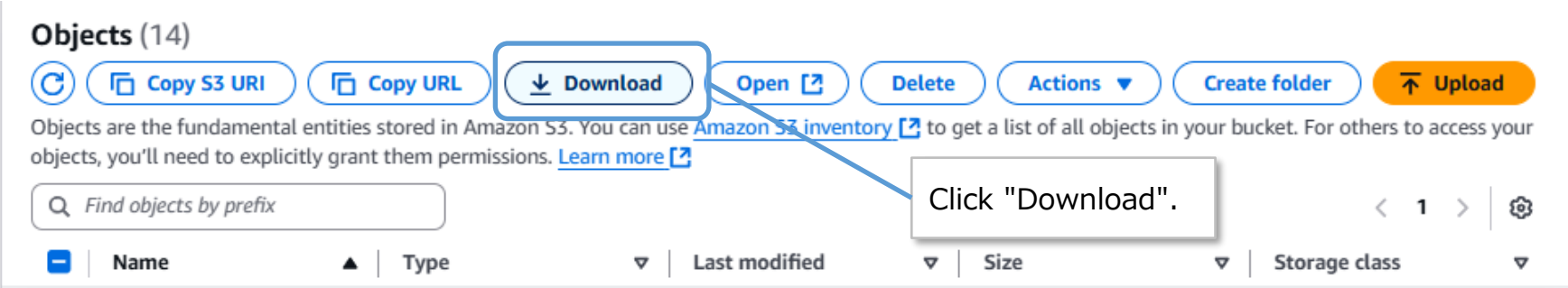
2.3 Confirm that the data from the Lustre file system is stored in the S3 bucket.



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

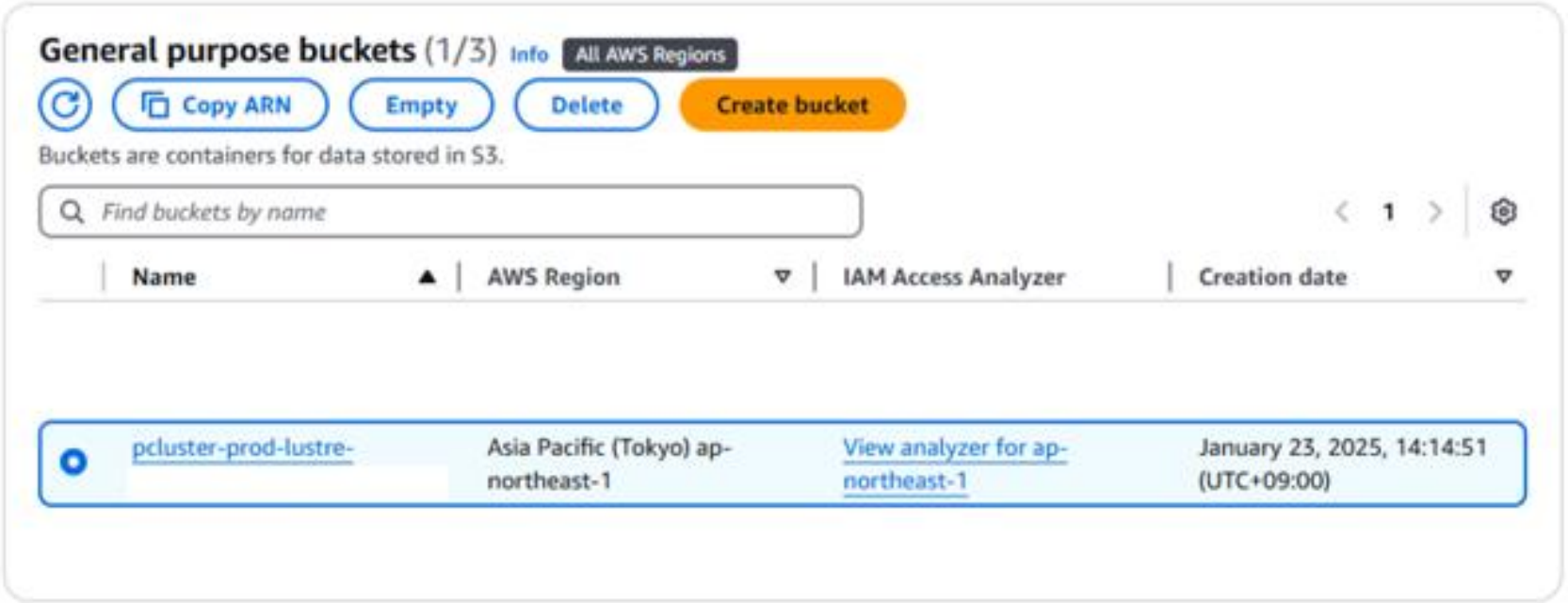
Data Transfer from Local Environment to the Cluster

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

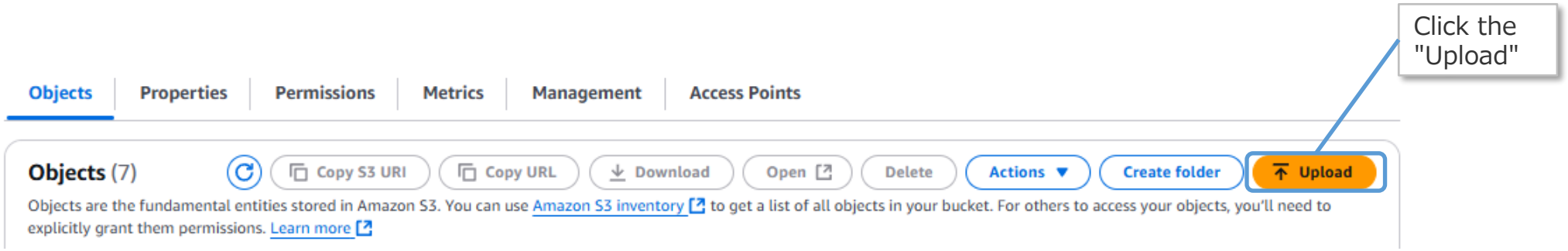
1.2 Click the S3 bucket name that was created on the "Creating External Storage" section.



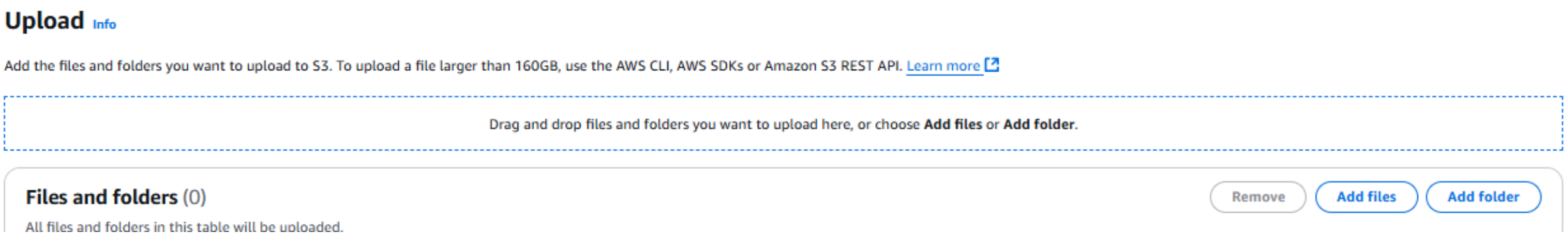
Data Transfer from Local Environment to the Cluster

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.

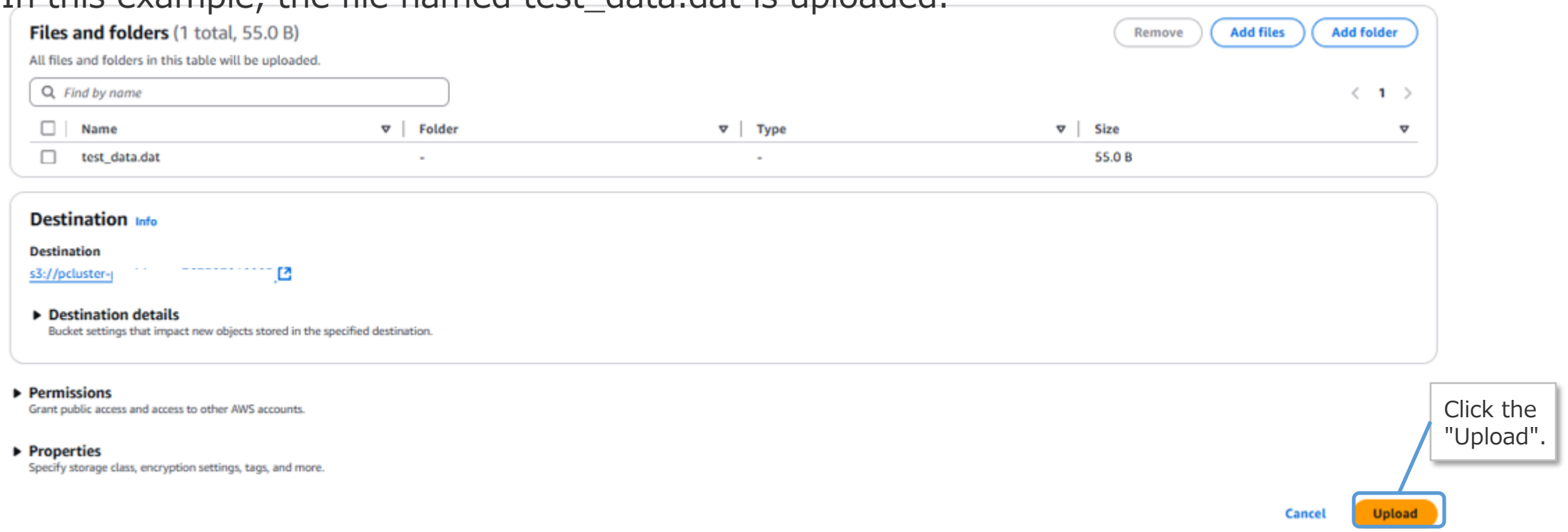


Data Transfer from Local Environment to the Cluster

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.



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

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.r-ccs.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
mpiexec --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
mpiexec --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
mpiexec --use-hwthread-cpus -n ${SLURM_NTASKS} singularity run ${SIFFILE} scale-rm run_R20kmDX500m.conf
#Post processing
cp sample/net2g_R20kmDX500m.conf ./net2g_R20kmDX500m.conf
mpiexec --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.

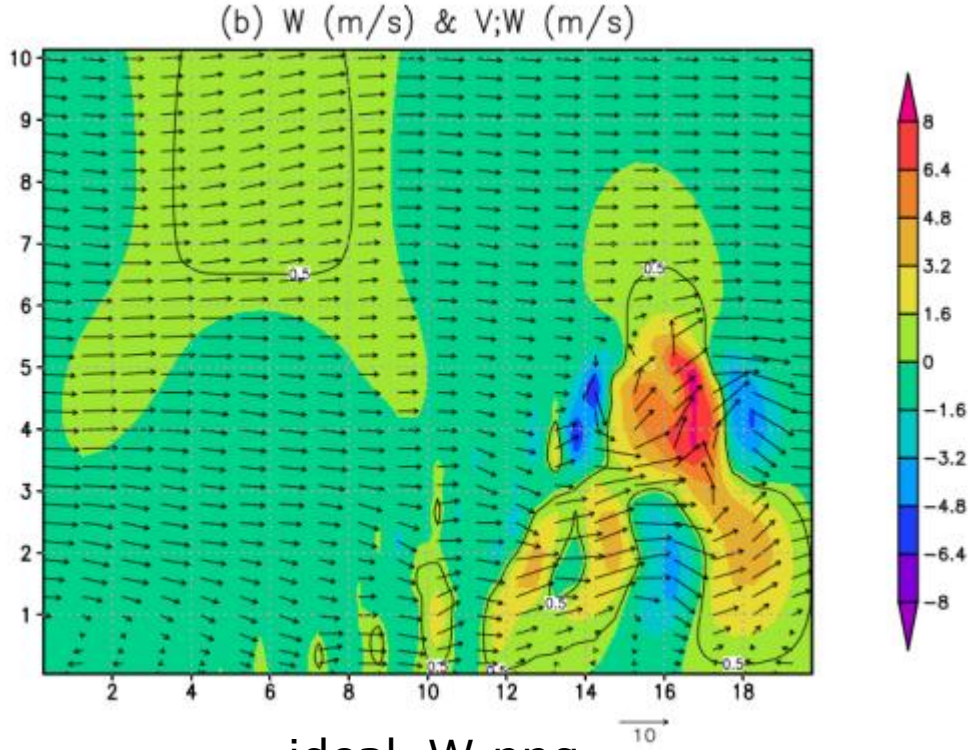
- QHYD_d01z-3d.ctl
- QHYD_d01z-3d.grd
- V_d01z-3d.ctl
- V_d01z-3d.grd
- W_d01z-3d.ctl
- W_d01z-3d.grd

Execution of SCALE

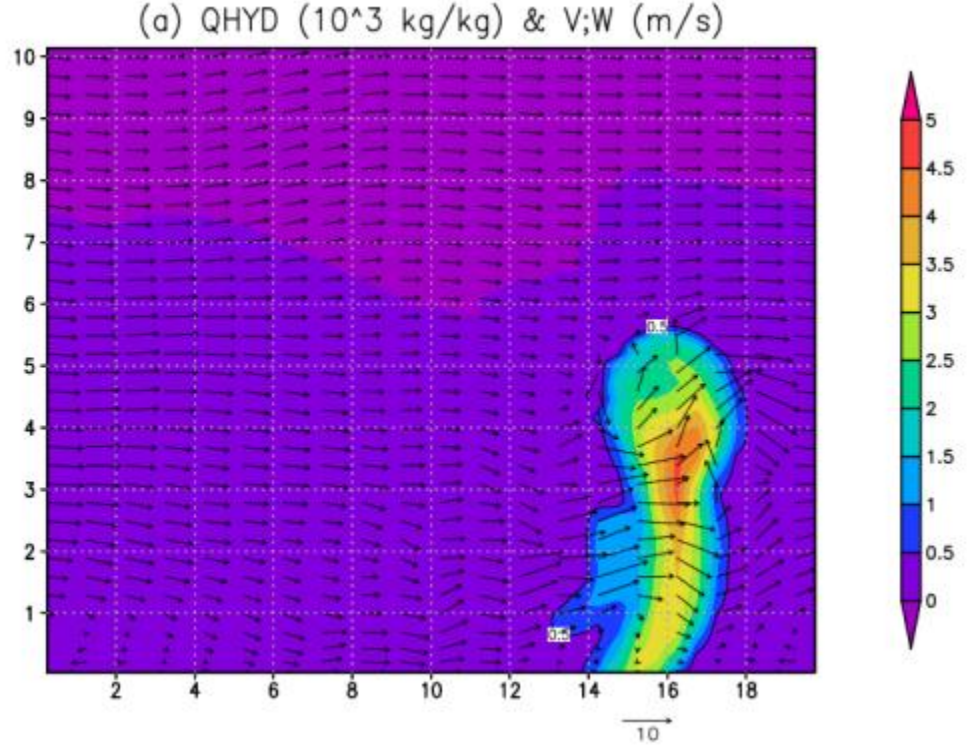
1. Execution Process (4/4)

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.



ideal_W.png



ideal_qhyd.png

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-xxxxxxxxxxxxx",
-------------------------------------------------------
```

"parallelclusternetworking-pubpriv-xxxxxxxxxxxxx" 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-xxxxxxxxxxxxx" with the stack name you recorded earlier.

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

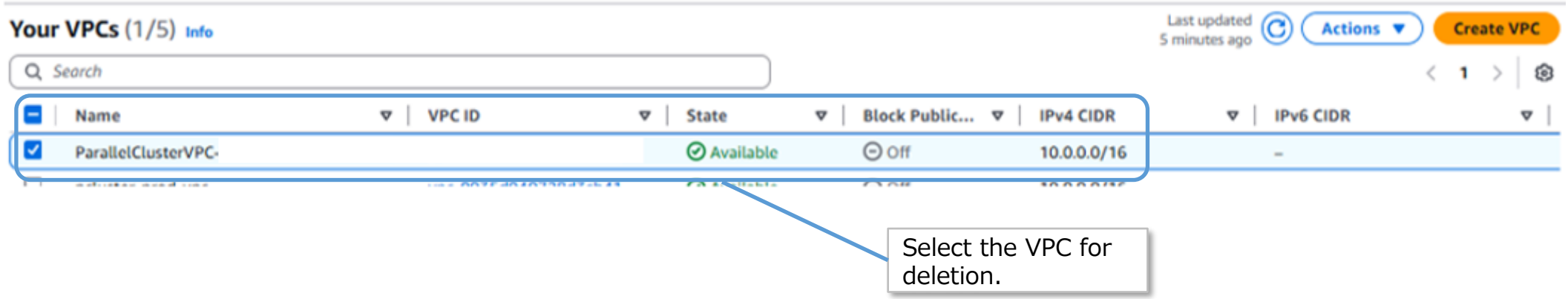
(1) Deleting the Cluster Environment

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/vpcconsole/home?region=ap-northeast-1#vpcs

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

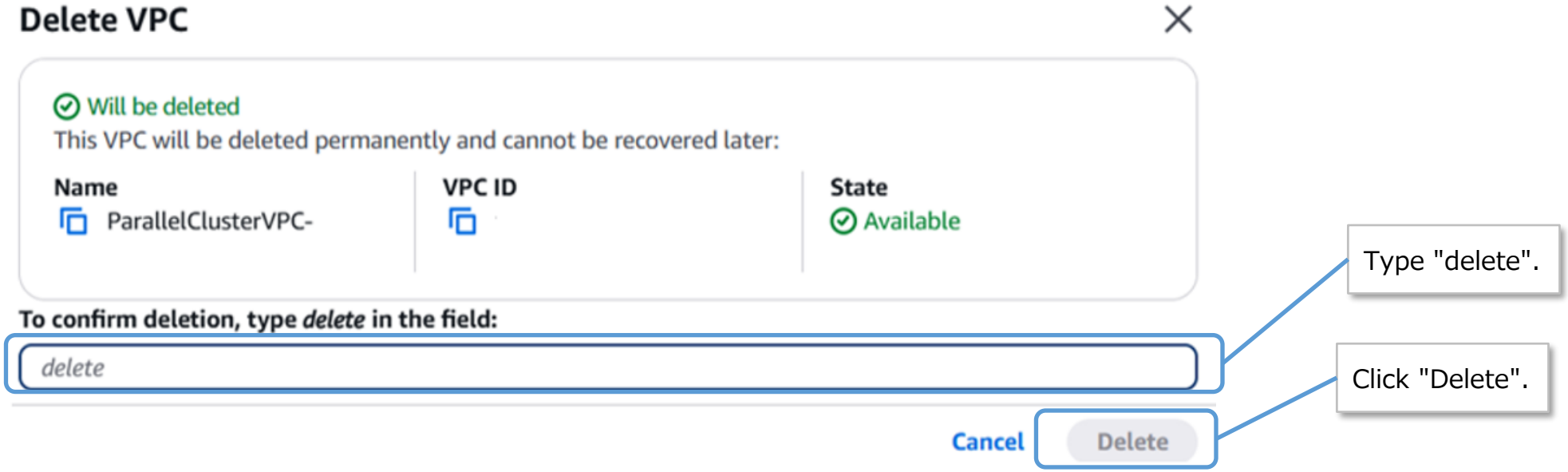


(1) Deleting the Cluster Environment

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.



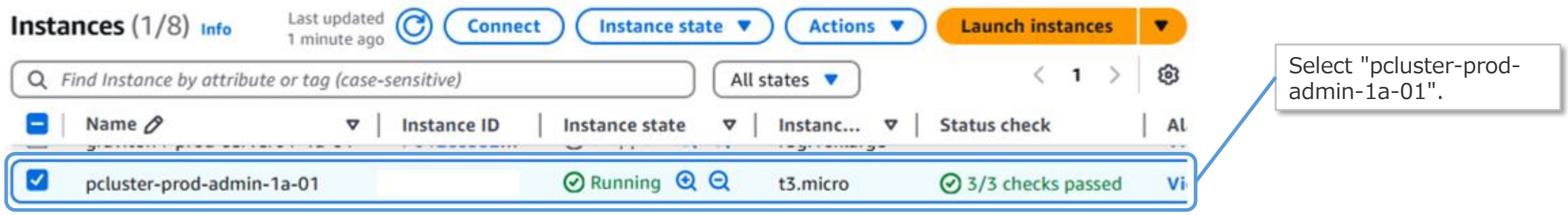
(1) Deleting the Cluster Environment

3. Deleting ParallelCluster Setup Instance (1/2)

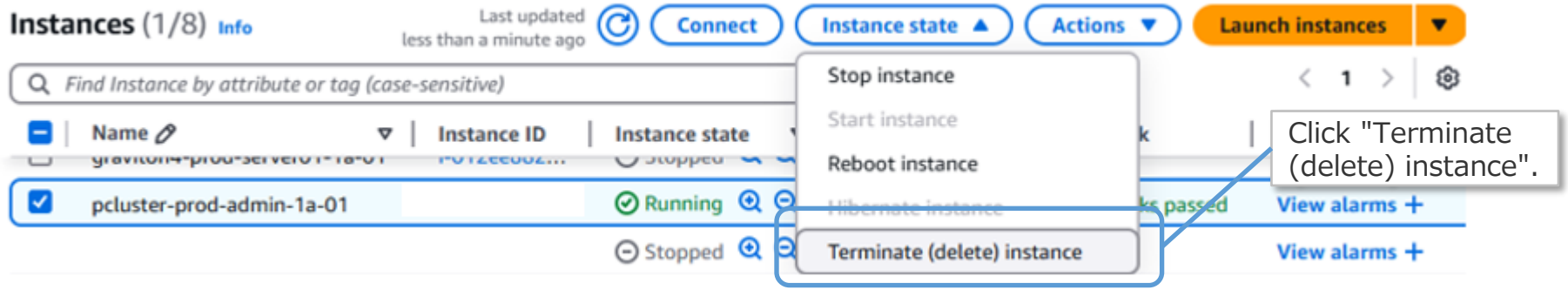
3.1 In the AWS Management Console, navigate to the EC2 Instances page.

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

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



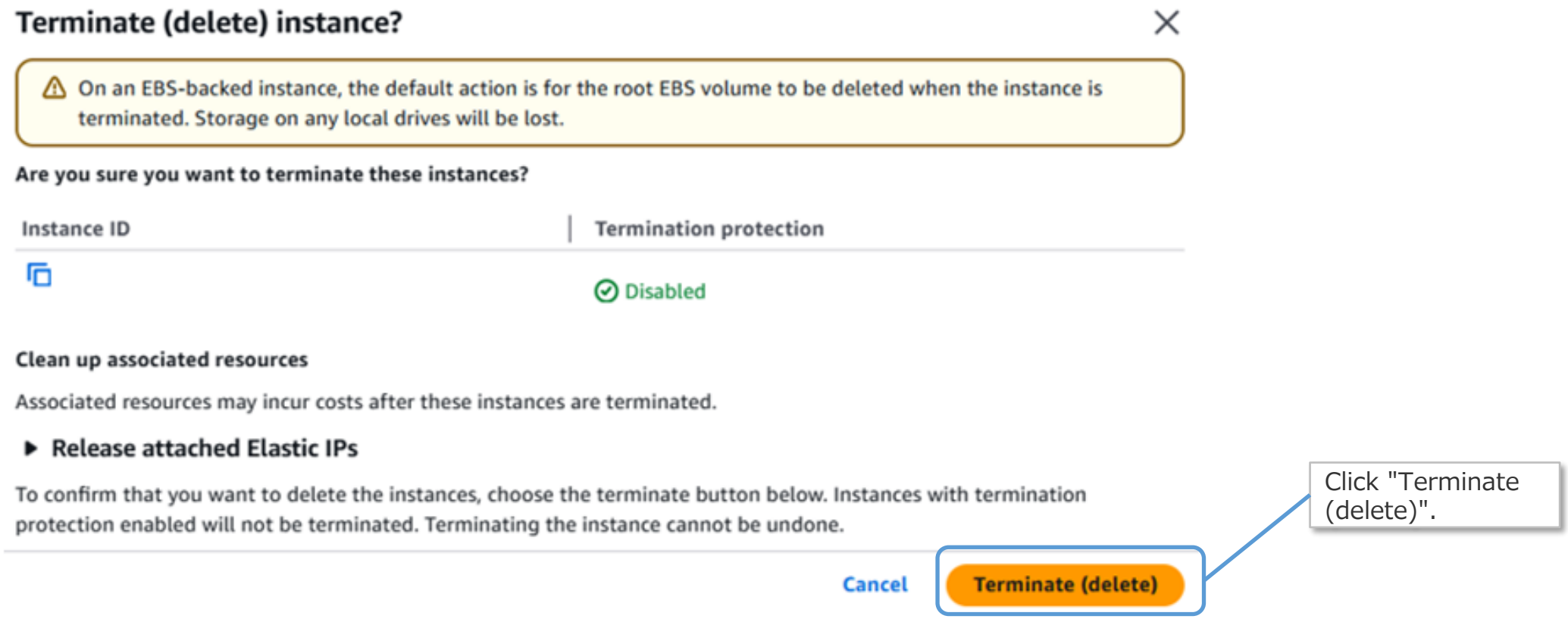
3.3 Click "Instance State" in the top-right corner, then select "Terminate (delete) instance".



(1) Deleting the Cluster Environment

3. Deleting ParallelCluster Setup Instance (2/2)

3.4 Click "Terminate (Delete)".



(1) Deleting the Cluster Environment

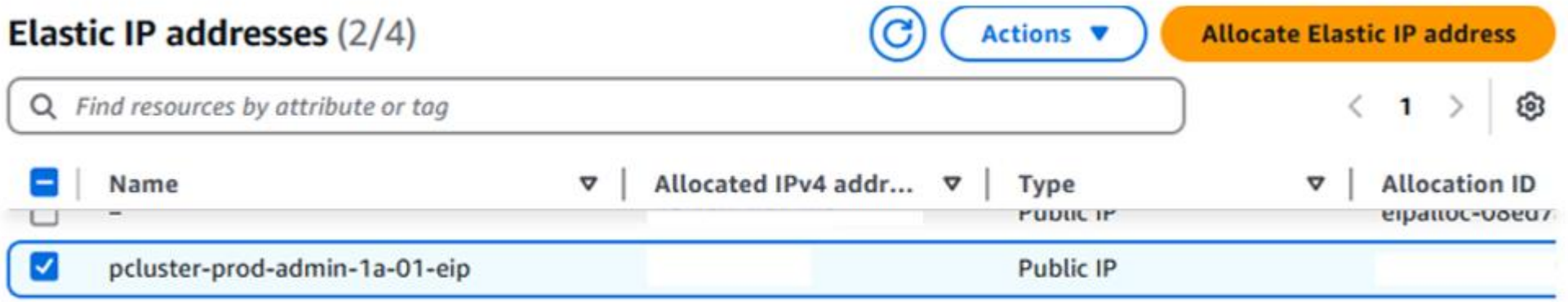
4. Releasing Static IP Address (1/3)

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

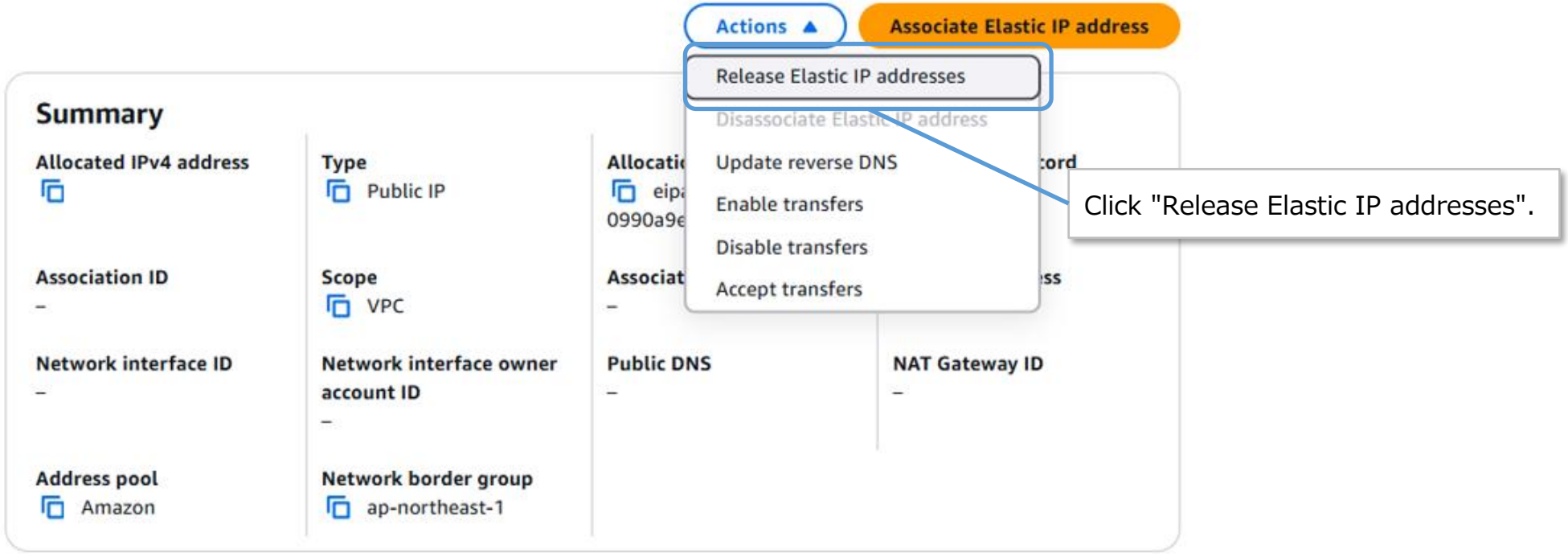
4.2 Select the Elastic IP address to be released.



(1) Deleting the Cluster Environment

4. Releasing Static IP Address (2/3)

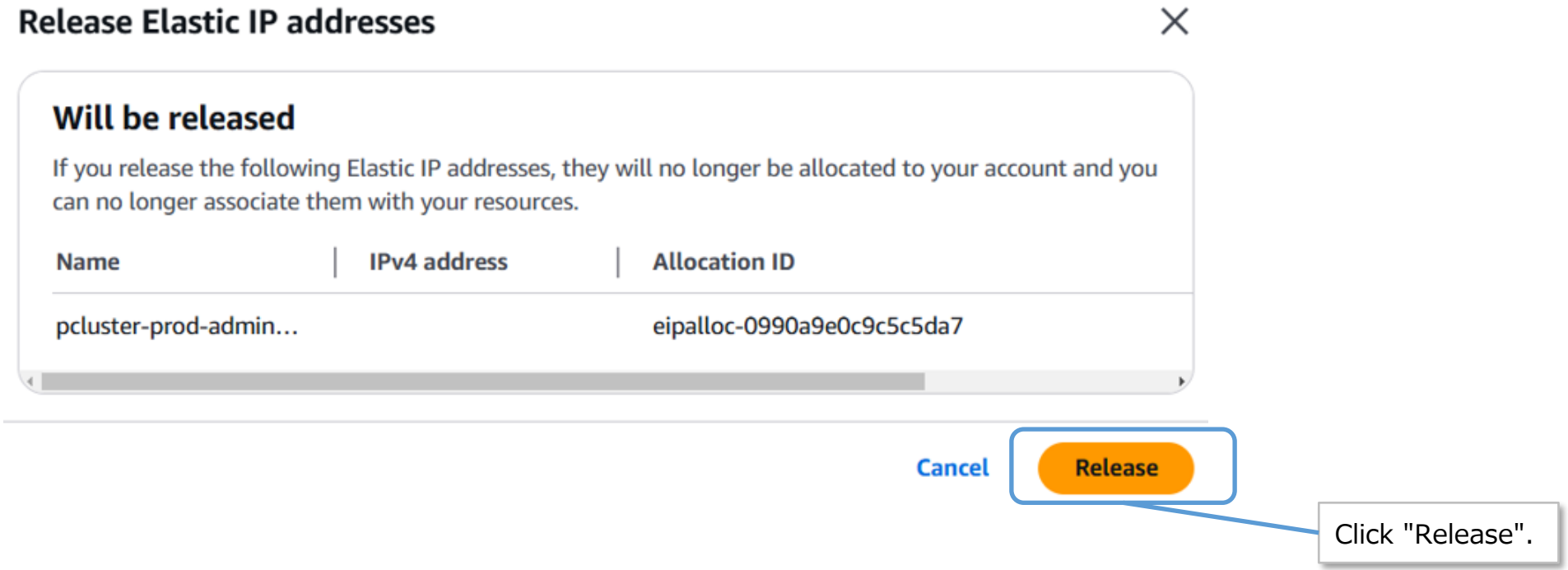
4.3 Click the "Actions" in the top-right corner, and choose "Release Elastic IP addresses".



(1) Deleting the Cluster Environment

4. Releasing Static IP Address (3/3)

4.4 Click "Release".



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

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 (lammps)
- 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)
- Quantume 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)
/home
/opt/parallelcluster/shared
/opt/slurm (No write permissions)