



Archetype AI

AWS Marketplace Deployment Guide

Version: 1.1.3

Executive summary

Archetype exists to unlock the potential of physical AI through Newton, a Large Behavior Model (LBM) that understands and reasons about the physical world using multimodal sensor data. Built on advanced machine learning techniques, Newton processes and interprets data from any sensor type while ensuring data privacy and enabling edge deployment. With a focus on the trillion-dollar sensor economy, Newton is positioned to transform industries such as construction and manufacturing by providing new insights into their physical operations.

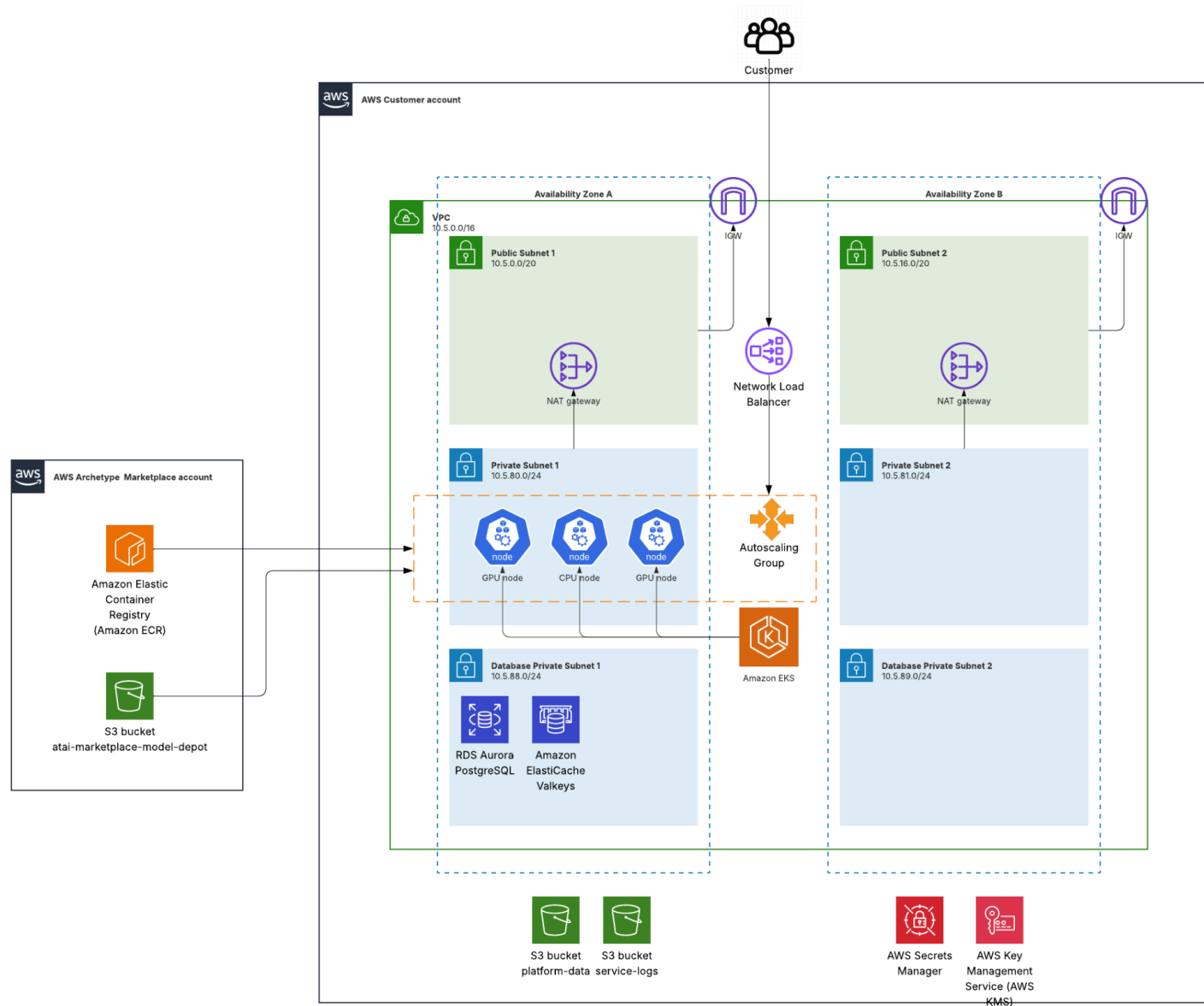
Table of contents

Document Version History	2
Executive summary	2
Table of contents	3
Infrastructure Overview	7
AWS infrastructure prerequisites	8
VPC configuration	8
Step 1: Create the VPC (Manual Configurration)	8
Step 1.1 Enable DNS Hostnames (if not already enabled)	9
Step 2: Create Internet Gateway	10
Step 3: Create Public Subnets (with correct CIDRs from the start)	11
3.1 Enable Auto-assign Public IPv4 for Public Subnets	12
Step 4: Create Private Subnets	13
Step 5: Create Database Subnets	14
Step 6: Allocate Elastic IP for NAT Gateway	15
Step 7: Create NAT Gateway	16
Step 8: Create and Configure Route Tables	17
Public Route Table:	17
Private Route Table:	19
Database Route Table:	21
Valkey clusters configuration	23
Prerequisites	23
Step 1: Create ElastiCache Subnet Group (if not existing)	23
Step 2: Create Security Group (if not existing)	26
Step 3: Create Parameter Group for Valkey 8.0	27
Step 3.1 After creation, edit the parameter group:	27
Step 4: Create Valkey Clusters	29
Cluster 1: registry	29
Cluster 2: access-manager	36
Cluster 3: api-events	37
Cluster 4: dfc (10 shards)	37
Cluster 5: file (10 shards)	38
Cluster 6: gpq (10 shards)	38
Cluster 7: health	39
Cluster 8: lens (10 shards)	40
Step 5: Store Auth Tokens in AWS Secrets Manager (Recommended)	41
Benefits of using Secrets Manager:	44
PostgreSQL database configuration	45
Prerequisites	45

Step 1: Create Database Subnet Group (if not already created)	45
Step 2: Create Security Group (if not existing)	46
Step 3: Create Aurora PostgreSQL Cluster	48
Step 4: Extra Database Configuration steps	55
Create databases	55
Create atai_dev database user	56
EKS cluster configuration	59
Prerequisites	59
Step 1: Create cluster IAM role	59
Step 2: Create cluster	61
Step 3: Update kubeconfig	67
Step 4: Get the default cluster security group	68
EKS managed node group configuration	69
Prerequisites	69
Step 1: Creating the Amazon EKS node IAM role	70
Step 2: Creating Node Security Group	72
Step 2.1 Add self rules to the node security group	73
Step 3: Launch templates	75
Step 3.1 CPU Node Group	75
Step 3.2 GPU Node Group	79
Step 4: Create CPU Node Group	83
Step 5: Create GPU Node Group	88
EKS configuration - Install the NVIDIA Device Plugin	93
Prerequisites	93
Step 1: Manual installation	93
S3 configuration	94
Step 1: Create the platform-data bucket	94
Step 2: Create the service logs bucket	97
Application Endpoints Configuration	99
Platform Architecture Overview	99
Required Public URLs	99
How Do I Expose My Services Publicly?	99
What is an Ingress?	99
How Do I Secure the Traffic?	100
How Do I Configure DNS?	100
Deployment Checklist	101
Before Helm Chart Installation:	101
Support	102
Prerequisites	103
Download the configuration files	103
Step 1: Kubernetes namespaces	104

Step 2: Kubernetes Service account for IAM roles (IRSA)	104
Step 3: Kubernetes secrets required for the atai-platform services	107
Step 3.1 Generate values for the IAM service secret	107
Helm chart installation	109
Prerequisites	109
Step 1: Installation	109
Getting Started with the Archetype Platform	111
Appendix	112
Bastion host configuration	112
Prerequisites	112
Step 1: Launch EC2 Instance (Bastion Host)	112
Step 2: Connect to your Bastion host	117
AWS Service Quotas	118
Running On-Demand G and VT instances	118
Running On-Demand Standard (A, C, D, H, I, M, R, T, Z) instances	120
EKS configuration - Install the NGINX ingress controller	121
Prerequisites	121
Step 1: Add helm repository	121
Step 2: Prepare configuration values	122
Step 2.1 High Availability Configuration (Optional)	123
Step 3: Install NGINX Ingress Controller	123
EKS configuration - Configure Cert-Manager and Let's Encrypt	124
MANUAL SETUP INSTRUCTIONS FOR EKS IRSA (IAM ROLES FOR SERVICE ACCOUNTS)	124
PREREQUISITES	124
STEP 1: VERIFY PREREQUISITES	124
STEP 2: ASSOCIATE IAM OIDC PROVIDER (IF NOT ALREADY DONE)	124
STEP 3: CREATE IAM POLICIES	125
POLICY 1: PLATFORM DATA ACCESS	125
POLICY 2: SERVICE LOGS ACCESS	126
POLICY 3: MODEL DEPOT ACCESS	126
STEP 4: CREATE KUBERNETES NAMESPACE	127
STEP 5: CREATE IAM ROLE FOR SERVICE ACCOUNT	128
Create an IAM role with the following configuration:	128
ATTACH POLICIES TO ROLE	129
STEP 6: CREATE KUBERNETES SERVICE ACCOUNT	129
STEP 7: VERIFICATION	129
SUMMARY OF CREATED RESOURCES	130
IMPORTANT NOTES	130

Infrastructure Overview



atai Infrastructure

AWS License Manager setup

This Marketplace product uses **AWS License Manager** for proper operation. You **must enable and configure AWS License Manager** in your account before deployment.

Failure to do so will prevent the product from functioning correctly. For more information on how to set up License Manager correctly, see [Appendix: License Manager Setup](#).

AWS infrastructure prerequisites

This document outlines the AWS infrastructure prerequisites that must be deployed before installing atai-platform helm chart.

VPC configuration

Create VPC with Public, Private, and Database Subnets

Step 1: Create the VPC (Manual Configuration)

1. Go to VPC Dashboard → Your VPCs → Create VPC
2. Select VPC only (not "VPC and more")
3. Configure:
 - a. Name tag: atai-platform-vpc
 - b. IPv4 CIDR block: 10.5.0.0/16
 - c. IPv6 CIDR block: No IPv6 CIDR block
 - d. Tenancy: Default
4. Click Create VPC

VPC > Your VPCs > Create VPC

CreateVpc

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.

atal-platform-vpc

IPv4 CIDR block [Info](#)
☒ IPv4 CIDR manual input ☐ IPAM-allocated IPv4 CIDR block

IPv4 CIDR
10.5.0.0/16
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

Tags

A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

Key **Value - optional** [Remove tag](#)

[Add tag](#)
You can add 49 more tags

[Cancel](#) [Preview code](#) [Create VPC](#)

Step 1.1 Enable DNS Hostnames (if not already enabled)

1. Go to Your VPCs → select your VPC
2. Actions → Edit VPC settings

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

☒	Name	VPC ID	State	Block Public...	IPv4 CIDR	IPv6 CIDR	DHCP opti...
☒	atal-platform-vpc	vpc-0a79faee3e664a31d	Available	Off	10.5.0.0/16	-	dopt-03a8172d4e5b9cd3f

Last updated 1 minute ago [Actions](#) [Create VPC](#)

- Create default VPC
- Create flow log
- Edit VPC settings**
- Edit CIDRs
- Manage middlebox routes
- Manage tags
- Delete VPC

3. Enable DNS hostnames and DNS resolution
4. Save

VPC > Your VPCs > vpc-0a79faee3e664a31d > Edit VPC settings

Edit VPC settings

VPC details [Info](#)

VPC ID
Name

DHCP settings [Info](#)
DHCP option set

DNS settings
☒ Enable DNS resolution [Info](#)
☒ Enable DNS hostnames [Info](#)

Network Address Usage metrics settings
☐ Enable Network Address Usage metrics [Info](#)

[Cancel](#) [Save](#)

Step 2: Create Internet Gateway

1. Go to VPC Dashboard → Go to Internet Gateways → Create internet gateway
2. Name tag: atai-platform-vpc-igw
3. Click Create internet gateway

The screenshot shows the 'Create internet gateway' page in the AWS VPC console. The breadcrumb trail is 'VPC > Internet gateways > Create internet gateway'. The page title is 'Create internet gateway' with an 'Info' icon. A descriptive text states: 'An internet gateway is a virtual router that connects a VPC to the internet. To create a new internet gateway specify the name for the gateway below.' The 'Internet gateway settings' section includes a 'Name tag' field with the value 'atai-platform-vpc-igw'. The 'Tags - optional' section shows a table with one tag: 'Name' with value 'atai-platform-vpc-igw'. At the bottom right, there are 'Cancel' and 'Create internet gateway' buttons.

4. Select it → Actions → Attach to VPC

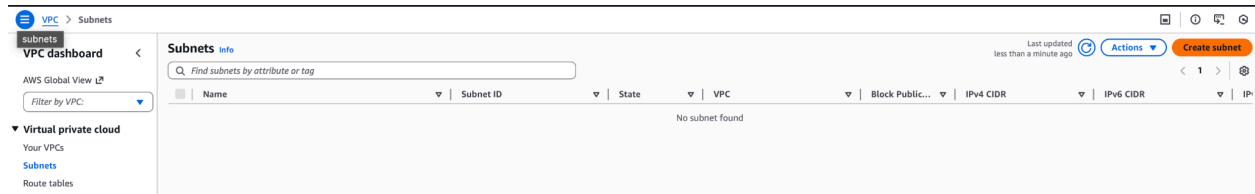
The screenshot shows the 'Internet gateway details' page for 'igw-05499c47a963a31f8 / atai-platform-vpc-igw'. A green notification bar at the top says: 'InternetGateway igw-05499c47a963a31f8 - atai-platform-vpc-igw. You can now attach to a VPC to enable the VPC to communicate with the internet.' The 'Details' section shows the 'Internet gateway ID' as 'igw-05499c47a963a31f8', 'State' as 'Detached', 'VPC ID' as '-', and 'Owner' as '716124474177'. The 'Tags (1)' section shows one tag: 'Name' with value 'atai-platform-vpc-igw'. On the right, the 'Actions' menu is open, showing 'Attach to VPC', 'Detach from VPC', 'Manage tags', and 'Delete'. At the bottom right, there are '< 1 >' and a settings icon.

5. Select your VPC → Attach internet gateway

The screenshot shows the 'Attach to VPC (igw-05499c47a963a31f8)' page in the AWS VPC console. The breadcrumb trail is 'VPC > Internet gateways > Attach to VPC (igw-05499c47a963a31f8)'. The page title is 'Attach to VPC (igw-05499c47a963a31f8)' with an 'Info' icon. The 'VPC' section states: 'Attach an internet gateway to a VPC to enable the VPC to communicate with the internet. Specify the VPC to attach below.' The 'Available VPCs' section shows a search bar with the value 'vpc-0a79faee3e664a31d'. At the bottom, there is an 'AWS Command Line Interface command' section. At the bottom right, there are 'Cancel' and 'Attach internet gateway' buttons.

Step 3: Create Public Subnets (with correct CIDRs from the start)

1. Go to VPC Dashboard → Go to Subnets → Create subnet



2. Subnet 1:
 - a. Select the VPC created in the Step 1
 - b. VPC: Select your VPC
 - c. Subnet name: atai-platform-vpc-public-us-west-2a
 - d. Availability Zone: us-west-2a (or your AZ1)
 - e. IPv4 CIDR block: 10.5.80.0/24
 - f. Click Create subnet

The screenshot shows the 'Create subnet' form in the AWS console. The 'VPC' section shows the selected VPC ID 'vpc-0a79faee5e664a31d (atai-platform-vpc)'. The 'Subnet settings' section includes: 'Subnet name' set to 'atai-platform-vpc-public-us-west-2a', 'Availability Zone' set to 'United States (Oregon) / usw2-az2 (us-west-2a)', 'IPv4 VPC CIDR block' set to '10.5.0.0/16', and 'IPv4 subnet CIDR block' set to '10.5.80.0/24'. The 'Tags' section shows a tag with key 'Name' and value 'atai-platform-vpc-public-us-west-2a'. At the bottom, there are 'Cancel' and 'Create subnet' buttons.

3. Subnet 2:
 - a. Select the VPC created in the Step 1
 - b. Subnet name: atai-platform-vpc-public-us-west-2b

- c. Availability Zone: us-west-2b (or your AZ2)
- d. IPv4 CIDR block: 10.5.81.0/24
- e. Click Create subnet

3.1 Enable Auto-assign Public IPv4 for Public Subnets

1. Go to VPC Dashboard → Go to Subnets → select each public subnet
2. Actions → Edit subnet settings

The screenshot shows the AWS VPC console 'Subnets' page. A table lists two subnets: 'atal-platform-vpc-public-us-west-2b' and 'atal-platform-vpc-public-us-west-2a'. Both are in the 'Available' state. The 'Actions' menu for the first subnet is open, showing options like 'View details', 'Create flow log', and 'Edit subnet settings'.

Name	Subnet ID	State	VPC	Block Public...	IPv4 CIDR
atal-platform-vpc-public-us-west-2b	subnet-0ebfa4aa83c5ae95b	Available	vpc-0a79faee3e664a31d atal-...	Off	10.5.81.0/24
atal-platform-vpc-public-us-west-2a	subnet-04434b04f54f5c276b	Available	vpc-0a79faee3e664a31d atal-...	Off	10.5.80.0/24

3. Check Enable auto-assign public IPv4 address
4. Save

The screenshot shows the 'Edit subnet settings' page for 'subnet-0ebfa4aa83c5ae95b'. The 'Auto-assign IP settings' section has 'Enable auto-assign public IPv4 address' checked. The 'Resource-based name (RBN) settings' section has 'IP name' selected under 'Hostname type'.

Subnet
Subnet ID: subnet-0ebfa4aa83c5ae95b
Name: atal-platform-vpc-public-us-west-2b

Auto-assign IP settings
Enable AWS to automatically assign a public IPv4 or IPv6 address to a new primary network interface for an instance in this subnet.
☒ Enable auto-assign public IPv4 address
☐ Enable auto-assign customer-owned IPv4 address

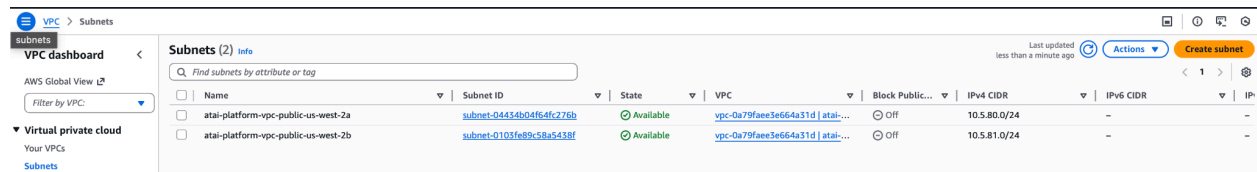
Resource-based name (RBN) settings
Specify the hostname type for EC2 instances in this subnet and optional RBN DNS query settings.
☐ Enable resource name DNS A record on launch
☐ Enable resource name DNS AAAA record on launch
Hostname type:
☐ Resource name
☒ IP name

DNS64 settings
Enable DNS64 to allow IPv6-only services in Amazon VPC to communicate with IPv4-only services and networks.
☐ Enable DNS64

5. Repeat the process for both public subnets.

Step 4: Create Private Subnets

1. Go to VPC Dashboard → Go to Subnets → Create subnet



2. Subnet 1:
 - a. Select the VPC created in the Step 1
 - b. Name: atai-platform-vpc-private-us-west-2a
 - c. Availability Zone: us-west-2a (or your AZ1)
 - d. CIDR: 10.5.0.0/20
 - e. Click Create subnet

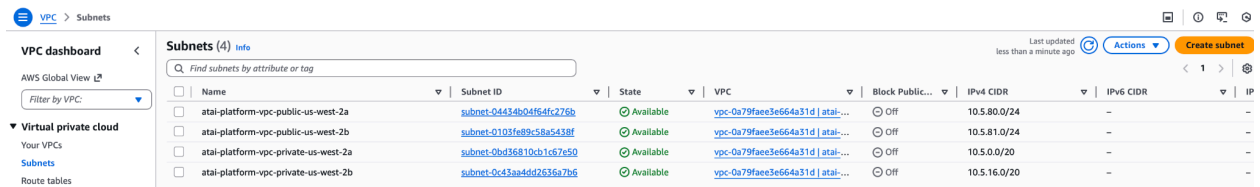
The screenshot shows the 'Create subnet' form in the AWS console. The 'VPC' section shows 'vpc-0a79faee3e664a31d (atai-platform-vpc)' selected. The 'Associated VPC CIDRs' section shows 'IPv4 CIDRs: 10.5.0.0/16'. The 'Subnet settings' section includes: 'Subnet name' (atai-platform-vpc-private-us-west-2a), 'Availability Zone' (United States (Oregon) / usw2-az2 (us-west-2a)), 'IPv4 VPC CIDR block' (10.5.0.0/16), and 'IPv4 subnet CIDR block' (10.5.0.0/20). The 'Tags' section shows a tag with key 'Name' and value 'atai-platform-vpc-private-us-west-2a'. At the bottom right, there are 'Cancel' and 'Create subnet' buttons.

- f. Subnet 2:
 - g. Select the VPC created in the Step 1
 - h. Name: atai-platform-vpc-private-us-west-2b
 - i. Availability Zone: us-west-2b (or your AZ2)
 - j. CIDR: 10.5.16.0/20

k. Click Create subnet

Step 5: Create Database Subnets

1. Go to VPC Dashboard → Go to Subnets → Create subnet



2. Subnet 1:

- Select the VPC created in the Step 1
- Name: atal-platform-vpc-database-us-west-2a
- Availability Zone: us-west-2a (or your AZ1)
- CIDR: 10.5.88.0/24
- Click Create subnet

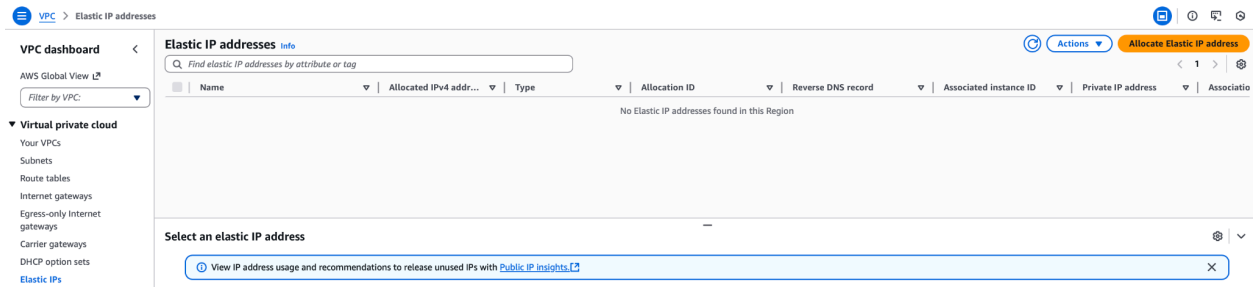
The screenshot shows the 'Create subnet' form in the AWS console. The 'VPC' section shows 'vpc-0a79faee3e664a31d (atal-platform-vpc)' selected. The 'Associated VPC CIDRs' section shows '10.5.0.0/16'. The 'Subnet settings' section is expanded, showing 'Subnet 1 of 1'. The 'Subnet name' field contains 'atal-platform-vpc-database-us-west-2a'. The 'Availability Zone' is set to 'United States (Oregon) / usw2-az2 (us-west-2a)'. The 'IPv4 VPC CIDR block' is '10.5.0.0/16'. The 'IPv4 subnet CIDR block' is '10.5.88.0/24'. The 'Tags' section is optional and shows a tag with key 'Name' and value 'atal-platform-vpc-database-us-west-2a'. A 'Create subnet' button is at the bottom right.

3. Subnet 2:

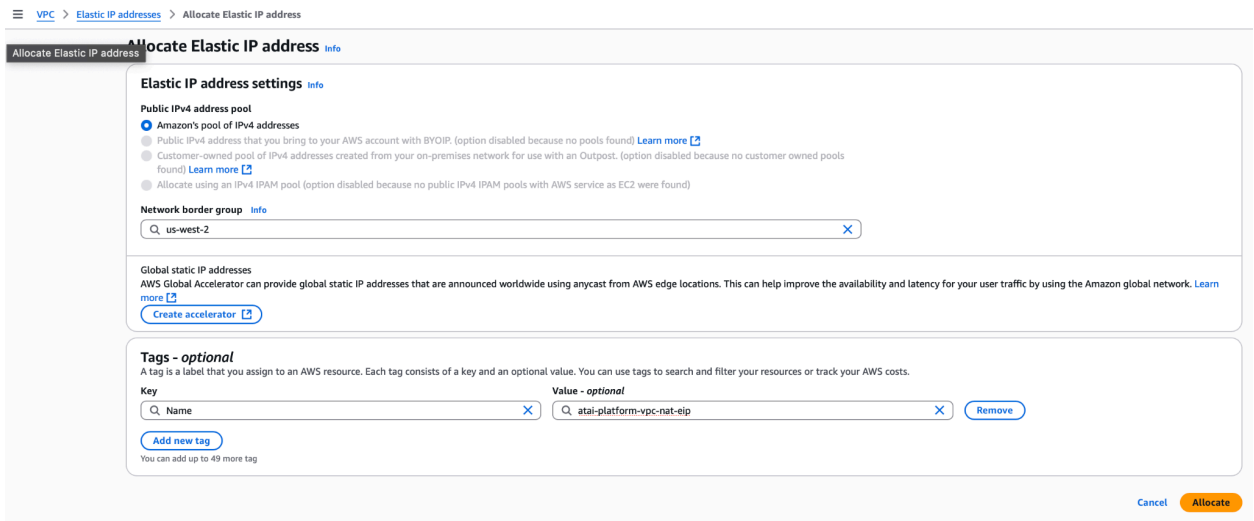
- Select the VPC created in the Step 1
- Name: atai-platform-vpc-database-us-west-2b
- Availability Zone: us-west-2b (or your AZ2)
- CIDR: 10.5.89.0/24
- Click Create subnet

Step 6: Allocate Elastic IP for NAT Gateway

- Go to VPC Dashboard → Go to Elastic IPs → Allocate Elastic IP address

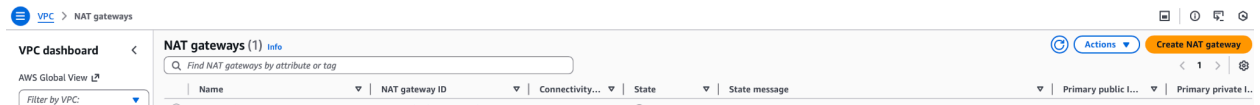


- Network border group: Select your region
- Public IPv4 address pool: Amazon's pool
- Add Name tag: atai-platform-vpc-nat-eip
- Click Allocate



Step 7: Create NAT Gateway

1. Go to VPC Dashboard → Go to NAT Gateways → Create NAT gateway



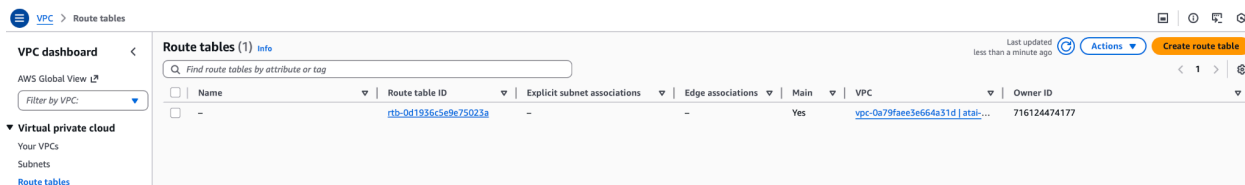
2. Name: atai-platform-vpc-nat
3. Subnet: Select first public subnet (10.5.80.0/24)
4. Elastic IP allocation ID: Select the EIP you just created in Step 6
5. Click Create NAT gateway (wait a few minutes)

A screenshot of the 'Create NAT gateway' form in the AWS VPC console. The form is titled 'Create NAT gateway Info' and includes a description: 'A highly available, managed Network Address Translation (NAT) service that instances in private subnets can use to connect to services in other VPCs, on-premises networks, or the internet.' The form is divided into sections: 'NAT gateway settings' and 'Tags'. In the 'NAT gateway settings' section, there are fields for 'Name - optional' (with value 'atai-platform-vpc-nat'), 'Subnet' (with value 'subnet-04434b04f64fc276b (atai-platform-vpc-public-us-west-2a)'), 'Connectivity type' (with 'Public' selected), and 'Elastic IP allocation ID' (with value 'eipalloc-0b989a3e8260adf7a (atai-platform-vpc-nat-eip)' and an 'Allocate Elastic IP' button). Below this is an 'Additional settings' section. The 'Tags' section includes a table with 'Key' and 'Value - optional' columns, showing a tag with key 'Name' and value 'atai-platform-vpc-nat'. There are buttons for 'Add new tag', 'Remove', 'Cancel', and 'Create NAT gateway'.

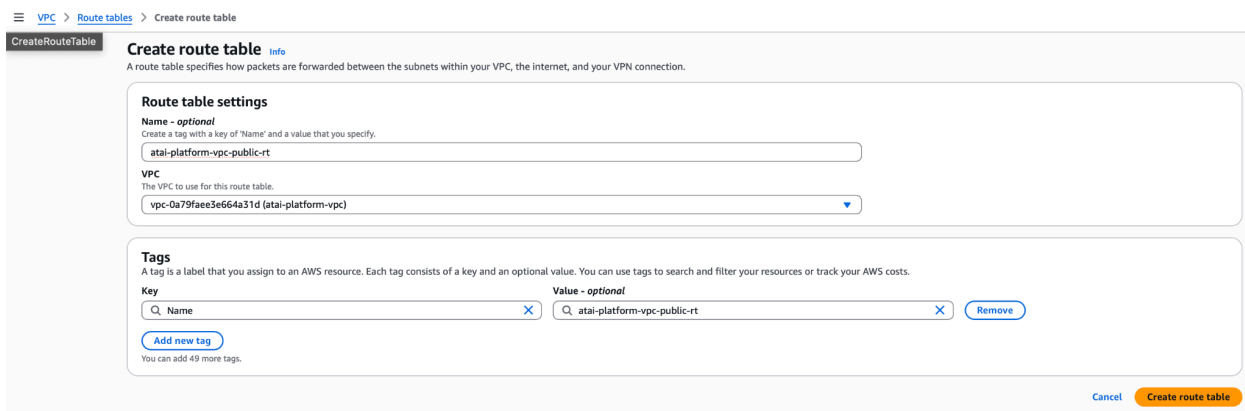
Step 8: Create and Configure Route Tables

Public Route Table:

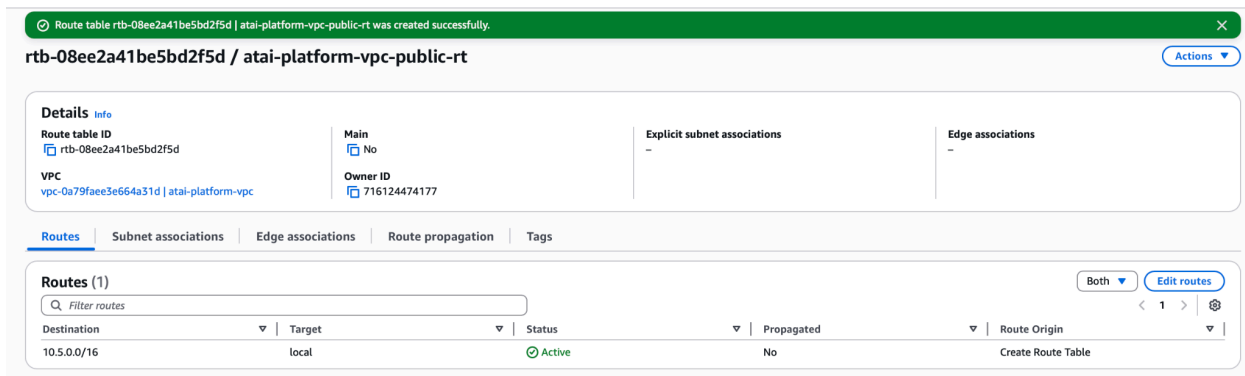
1. Go to VPC Dashboard → Route Tables → Create route table



2. Name: atai-platform-vpc-public-rt
3. VPC: Select your VPC from Step 1
4. Click Create route table



5. Routes → **Edit routes** → Add route:
 - a. Destination: 0.0.0.0/0
 - b. Target: Internet Gateway → select your IGW from Step 2
 - c. Save changes



VPCRoute tablesrtb-08ee2a41be5bd2f5dEdit routes

EditRoutes

Destination

10.5.0.0/16

Target

local

Status

Active

Propagated

No

Route Origin

CreateRouteTable

Q 0.0.0.0/0

Q local

Internet Gateway

Q igw-05499c47a963a31f8

Add route

Remove

Cancel

Preview

Save changes

6. Subnet associations → **Edit subnet associations:**
- a. Select both public subnets → Save associations

rtb-08ee2a41be5bd2f5d / atai-platform-vpc-public-rt

Actions

Details

Route table ID

rtb-08ee2a41be5bd2f5d

VPC

vpc-0a79faee3e664a31d | atai-platform-vpc

Main

No

Owner ID

716124474177

Explicit subnet associations

-

Edge associations

-

Routes

Subnet associations

Edge associations

Route propagation

Tags

Explicit subnet associations (0)

Edit subnet associations

Find subnet association

Name

Subnet ID

IPv4 CIDR

IPv6 CIDR

No subnet associations

You do not have any subnet associations.

VPCRoute tablesrtb-08ee2a41be5bd2f5dEdit subnet associations

Change which subnets are associated with this route table.

Available subnets (2/6)

Filter subnet associations

atai-platform-vpc-public-us-west-2a

subnet-04434b04f64fc276b

10.5.80.0/24

-

Main

atai-platform-vpc-private-us-west-2a

subnet-0bd36810cb1c57e50

10.5.0.0/20

-

Main

atai-platform-vpc-private-us-west-2b

subnet-0c43aa4dd2636a7b6

10.5.16.0/20

-

Main

atai-platform-vpc-database-us-west-2a

subnet-06819ecd03094dea7

10.5.88.0/24

-

Main

atai-platform-vpc-database-us-west-2b

subnet-06bd3873d34243b83

10.5.89.0/24

-

Main

atai-platform-vpc-public-us-west-2b

subnet-0ebfa4aa83c5ae95b

10.5.81.0/24

-

Main

Selected subnets

subnet-04434b04f64fc276b / atai-platform-vpc-public-us-west-2a

subnet-0ebfa4aa83c5ae95b / atai-platform-vpc-public-us-west-2b

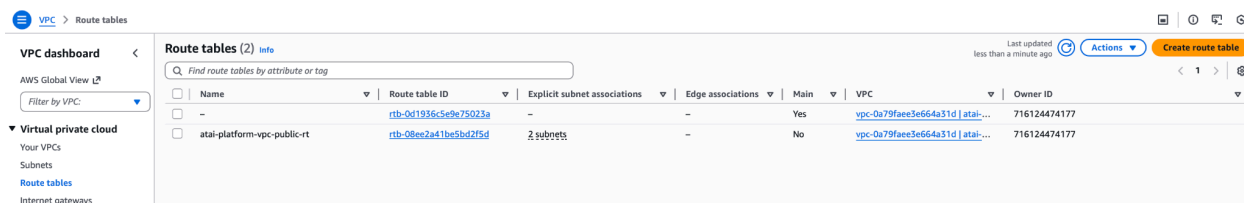
Cancel

Save associations

Archetype AI – Confidential 2026 ©

Private Route Table:

1. Go to VPC Dashboard → Route Tables → Create route table



2. Name: atai-platform-vpc-private-rt
3. VPC: Select your VPC from Step 1
4. 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.
atai-platform-vpc-private-rt

VPC
The VPC to use for this route table.
vpc-0a79faee3e664a31d (atai-platform-vpc)

Tags
A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

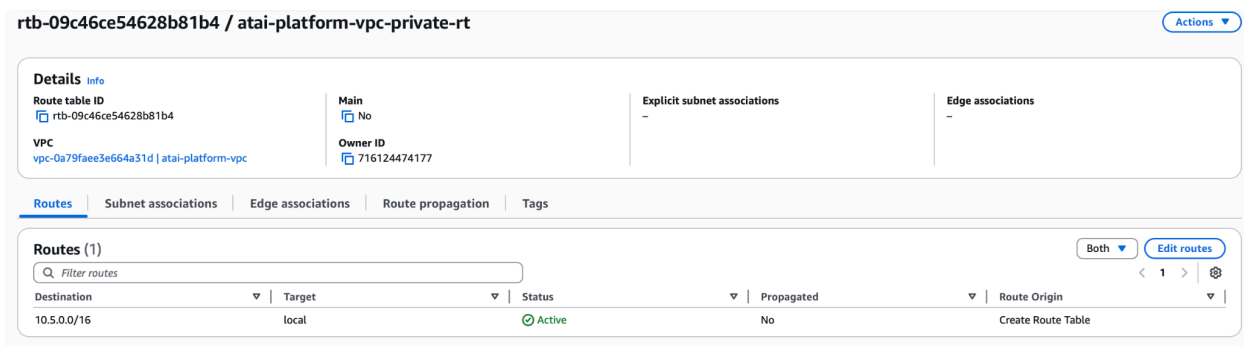
Key
Q Name X

Value - optional
Q atai-platform-vpc-private-rt X Remove

Add new tag
You can add 49 more tags.

Cancel Create route table

5. Routes → Edit routes → **Add route**:
 - a. Destination: 0.0.0.0/0
 - b. Target: NAT Gateway → select your NAT Gateway
 - c. Save changes



☰ VPC > Route tables > rtb-09c46ce54628b81b4 > Edit routes

Edit routes

Destination	Target	Status	Propagated	Route Origin
10.5.0.0/16	local	Active	No	CreateRouteTable
Q 0.0.0.0/0	Q local	-	No	CreateRoute
	Q NAT Gateway			
	Q nat-06cb295c04838edfc			

Add route

CancelPreviewSave changes

6. Subnet associations → Edit subnet associations:
- a. Select both private subnets → Save associations

rtb-09c46ce54628b81b4 / atai-platform-vpc-private-rt Actions

Details

Route table ID
rtb-09c46ce54628b81b4

VPC
vpc-0a79faec3e664a31d | atai-platform-vpc

Main
No

Owner ID
716124474177

Explicit subnet associations
-

Edge associations
-

Routes

Subnet associations

Edge associations

Route propagation

Tags

Explicit subnet associations (0)

Name	Subnet ID	IPv4 CIDR	IPv6 CIDR
No subnet associations			
You do not have any subnet associations.			

Edit subnet associations

< 1 > ⚙

☰ VPC > Route tables > rtb-09c46ce54628b81b4 > Edit subnet associations

Edit subnet associations

Change which subnets are associated with this route table.

Available subnets (2/6)

	Name	Subnet ID	IPv4 CIDR	IPv6 CIDR	Route
☐	atai-platform-vpc-public-us-west-2a	subnet-0434b04f64fc276b	10.5.80.0/24	-	rtb-06
☒	atai-platform-vpc-private-us-west-2a	subnet-0bd36810cb1c67e50	10.5.0.0/20	-	Main
☒	atai-platform-vpc-private-us-west-2b	subnet-0c43aa4dd2636a7b6	10.5.16.0/20	-	Main
☐	atai-platform-vpc-database-us-west-2a	subnet-06819ecd03094dea7	10.5.88.0/24	-	Main
☐	atai-platform-vpc-database-us-west-2b	subnet-06bd3873d34243b83	10.5.89.0/24	-	Main
☐	atai-platform-vpc-public-us-west-2b	subnet-0ebfa4aa83c5ae95b	10.5.81.0/24	-	rtb-06

Selected subnets

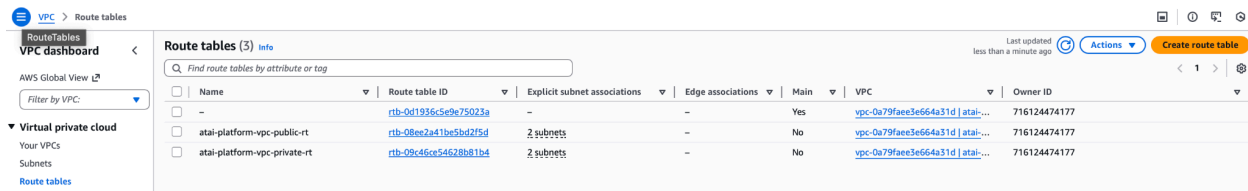
subnet-0bd36810cb1c67e50 / atai-platform-vpc-private-us-west-2a

subnet-0c43aa4dd2636a7b6 / atai-platform-vpc-private-us-west-2b

CancelSave associations

Database Route Table:

1. Go to VPC Dashboard → Route Tables → Create route table



2. Name: atai-platform-vpc-database-rt
3. VPC: Select your VPC from Step 1
4. Click Create route table

Create route table

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.
atai-platform-vpc-database-rt

VPC
The VPC to use for this route table.
vpc-0a79faee3e664a31d (atai-platform-vpc)

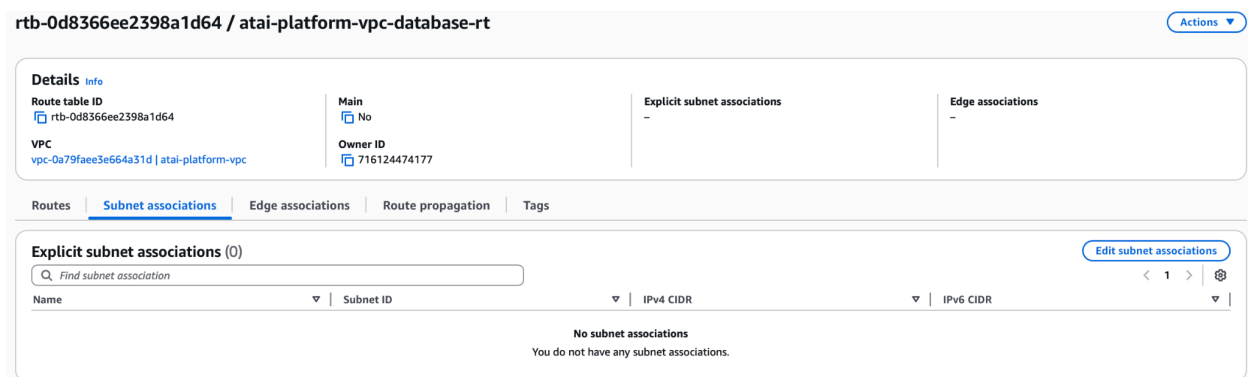
Tags
A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

Key
Name

Value - optional
atai-platform-vpc-database-rt

Create route table

5. (No extra routes needed - isolated)
6. Subnet associations → Edit subnet associations:
 - a. Select both database subnets → Save associations



VPCRoute tablesrtb-0d8566ee2398a1d64Edit subnet associations

EditRouteTableSubnetAssociations

Edit subnet associations

Change which subnets are associated with this route table.

Available subnets (2/6)

Filter subnet associations

	Name	Subnet ID	IPv4 CIDR	IPv6 CIDR	Route
☐	atal-platform-vpc-public-us-west-2a	subnet-04434b04f64fc276b	10.5.80.0/24	-	rtb-06
☐	atal-platform-vpc-private-us-west-2a	subnet-0bd36810cb1c67e50	10.5.0.0/20	-	rtb-05
☐	atal-platform-vpc-private-us-west-2b	subnet-0c43aa4dd2636a7b6	10.5.16.0/20	-	rtb-05
☒	atal-platform-vpc-database-us-west-2a	subnet-06819ecd03094dea7	10.5.88.0/24	-	Main (
☒	atal-platform-vpc-database-us-west-2b	subnet-06bd3873d34243b83	10.5.89.0/24	-	Main (
☐	atal-platform-vpc-public-us-west-2b	subnet-0ebfa4aa83c5ae95b	10.5.81.0/24	-	rtb-06

Selected subnets

subnet-06bd3873d34243b83 / atai-platform-vpc-database-us-west-2b X subnet-06819ecd03094dea7 / atai-platform-vpc-database-us-west-2a X

Cancel

Save associations

Valkey clusters configuration

Prerequisites

1. VPC with database subnets
2. Security group allowing access from EKS pods
3. Subnet group for database subnets

Step 1: Create ElastiCache Subnet Group

1. Go to ElastiCache → Subnet groups → Create subnet group

The screenshot shows the Amazon ElastiCache console. The left sidebar has a menu with 'Amazon ElastiCache' at the top, followed by 'Dashboard', 'Resources' (Valkey caches, Memcached caches, Redis OSS caches, Global datastores, Reserved nodes, Backups), and 'Configurations' (Subnet groups, Parameter groups, User management, User group management). The main content area shows the 'Subnet groups' page with a search bar, a table of subnet groups (empty), and a 'Create subnet group' button. A notification banner at the top says 'Amazon ElastiCache announces Valkey starting as low as \$6/month'.

2. Name: atai-db-subnet-group (or your name)
3. Description: Subnet group for Valkey clusters (single AZ for low latency)
4. VPC: Select your VPC

The screenshot shows the 'Create subnet group' form. It has a 'Subnet group settings' section with the following fields:

- Name:** atai-db-subnet-group
- Description - optional:** Subnet group for Valkey clusters
- VPC ID:** vpc-0a79faae3e664a31d (atai-platform-vpc)

Below these fields is a note: 'For Multi-AZ high availability mode, choose IDs for at least two subnets from two Availability Zones in the table below.' At the bottom, there is a 'Selected subnets (6)' table with columns: Availability Zone, Subnet ID, Subnet name, Outpost ID, CIDR block (IPv4), and CIDR block (IPv6). A 'Manage' button is next to the table.

5. In the section **Selected subnets** click on **Manage**
 - a. Availability Zones: Select only the first AZ (e.g., us-west-2a) and click on **Choose**
 - i. Important: Use only one AZ to reduce latency
 - ii. Note: EKS Managed node groups will use a private subnet in the same AZ (different subnet CIDR)

Manage subnets



Add or remove subnets from the table below.

Subnets (1/6)



< 1 2 >

☐	Availability Zone ▲	Subnet ID ▼	Subnet name ▼	Output
☐	us-west-2a	subnet-0bd36810cb1c67e50	atai-platform-vpc-private-us-west-2a	
☐	us-west-2a	subnet-04434b04f64fc276b	atai-platform-vpc-public-us-west-2a	
☒	us-west-2a	subnet-06819ecd03094dea7	atai-platform-vpc-database-us-west-2a	
☐	us-west-2b	subnet-0ebfa4aa83c5ae95b	atai-platform-vpc-public-us-west-2b	
☐	us-west-2b	subnet-06bd3873d34243b83	atai-platform-vpc-database-us-west-2b	

Cancel

Choose

6. Subnets: Select only the first database subnet (e.g., 10.5.88.0/24 in us-west-2a)
 - a. Do not select the second database subnet
7. Click Create

ElastiCache > Subnet groups > Create

Subnet group settings

A subnet group is a collection of subnets (typically private). Designate a subnet group for your clusters running in an Amazon Virtual Private Cloud (VPC) environment.

Name

The name can have up to 255 characters, and must not contain spaces.

Description - optional

VPC ID
The identifier for the VPC environment where your cluster is to run.
 [Create VPC](#)

For Multi-AZ high availability mode, choose IDs for at least two subnets from two Availability Zones in the table below.

Selected subnets (1) [Manage](#)

Availability Zone ▲	Subnet ID ▼	Subnet name ▼	Output ID ▼	CIDR block (IPv4) ▼	CIDR block (IPv6) ▼
us-west-2a	subnet-06819ecd03094dea7	atai-platform-vpc-database-us-west-2a		10.5.88.0/24	-

Tags
You can use tags to search and filter your subnet groups, or track your AWS costs.
No tags associated with the subnet group.
[Add new tag](#)
You can add 50 more tags.

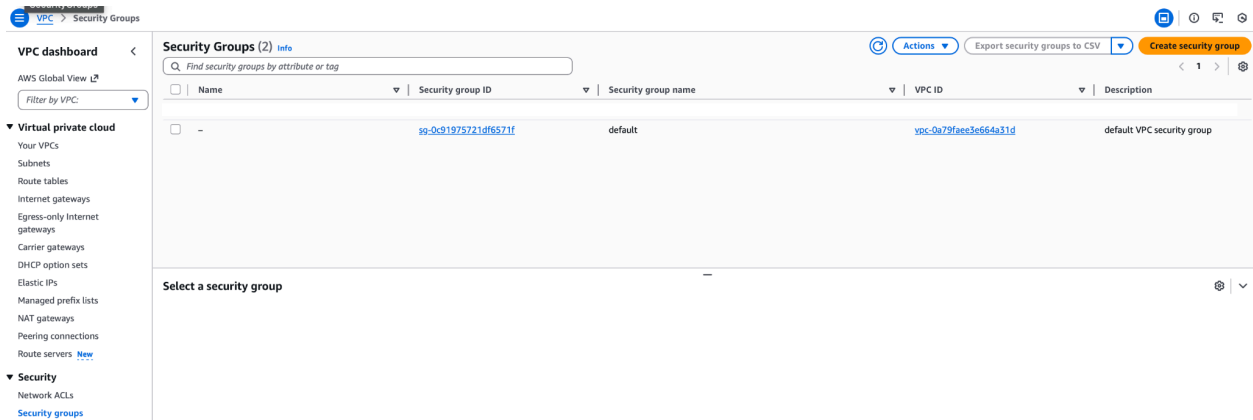
Cancel Create

Important Notes:

1. Single AZ deployment reduces cross-AZ network latency
2. All Valkey clusters will be created in this single AZ
3. EKS managed node groups should use a private subnet in the same AZ (e.g., 10.5.0.0/20 in us-west-2a) for optimal latency
4. Example: If you use us-west-2a for database subnet 10.5.88.0/24, use us-west-2a for private subnet 10.5.0.0/20 for node groups

Step 2: Create Security Group (if not existing)

1. Go to VPC → Security Groups → Create security group



2. Name: atai-valkey-sg (or your name)
3. Description: Security group for Valkey clusters
4. VPC: Select your VPC from section VPC configuration Step 1
5. Inbound rules: Add rule:
 - a. Type: Custom TCP
 - b. Port: 6379
 - c. Source: Custom → Enter your VPC CIDR (e.g., 10.5.0.0/16)
 - d. Description: Allow Valkey access from VPC

The screenshot shows the 'Create security group' form in the AWS Management Console. The form is divided into sections: Basic details, Inbound rules, Outbound rules, and Tags - optional. The 'Basic details' section includes fields for Security group name (atai-valkey-sg), Description (Security group for Valkey clusters), and VPC (vpc-0a79faee3e664a31d). The 'Inbound rules' section shows a rule being added with Type: Custom TCP, Protocol: TCP, Port range: 6379, Source: Custom (10.5.0.0/16), and Description: Allow Valkey access from VPC. The 'Outbound rules' section shows a rule being added with Type: All traffic, Protocol: All, Port range: All, Destination: Custom (0.0.0.0/0). The 'Tags - optional' section includes an 'Add new tag' button.

Note: For initial setup, opening to the VPC CIDR simplifies connectivity. Later, restrict to specific security groups (e.g., EKS node group security group) for tighter security.

- Click Create security group

Step 3: Create Parameter Group for Valkey 8.0

- Go to ElastiCache → Parameter groups → Create parameter group

Amazon ElastiCache announces Valkey starting as low as \$6/month
Introducing Valkey, an open source project that is a drop in replacement of Redis OSS, now available on ElastiCache, with up to 33% lower price than other engines. [Learn more](#)

Parameter groups (19)

Name	Family	Global	Description
default.memcached1.4	memcached1.4	No	Default parameter group for memcached1.4
default.memcached1.5	memcached1.5	No	Default parameter group for memcached1.5
default.memcached1.6	memcached1.6	No	Default parameter group for memcached1.6
default.redis2.6	redis2.6	No	Default parameter group for redis2.6
default.redis2.8	redis2.8	No	Default parameter group for redis2.8

- Group name: atai-valkey8-params
- Description: Parameter group for Valkey 8.0 clusters
- Parameter group family: valkey8

Create parameter group

Parameter group settings
Choose a parameter group. Amazon ElastiCache uses parameters to control the runtime properties of your nodes and clusters.

Name
atai-valkey8-params
The name can have up to 255 characters, and must not contain spaces.

Description
Parameter group for Valkey 8.0 clusters

Family
The family of the parameter group corresponding to an engine version.
valkey8

You can modify parameters after the parameter group is created.

Tags
You can use tags to search and filter your parameter groups, or track your AWS costs.
No tags associated with the parameter group.
[Add new tag](#)
You can add 50 more tags.

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

- Click Create

Step 3.1 After creation, edit the parameter group:

- Select the parameter group → Edit

Amazon ElastiCache announces Valkey starting as low as \$6/month
Introducing Valkey, an open source project that is a drop in replacement of Redis OSS, now available on ElastiCache, with up to 33% lower price than other engines. [Learn more](#)

Parameter groups (1/20)

Name	Family	Global	Description
atai-valkey8-params	valkey8	No	Parameter group for Valkey 8.0 clusters
default.memcached1.4	memcached1.4	No	Default parameter group for memcached1.4
default.memcached1.5	memcached1.5	No	Default parameter group for memcached1.5
default.memcached1.6	memcached1.6	No	Default parameter group for memcached1.6

2. Click on **Edit parameter values**

Amazon ElastiCache > Parameter groups > atai-valkey8-params

atai-valkey8-params

Parameter group settings

Name	Description	Family	Global
atai-valkey8-params	Parameter group for Valkey 8.0 clusters	valkey8	No

ARN: am:aws:elasticacheus-west-2:716124474177:parametergroup:atai-valkey8-params

Parameters (289)

Find parameters

Name	Allowed values	Is modifiable	Node type	Value	Source	Type	Change type	Description
act-pubsub-default	resetchannels,allch...	Yes	All	allchannels	system	string	immediate	Default pub...

3. In search bar type **cluster-enabled**

4. With the dropdown menu set the value to **yes**

atai-valkey8-params

Parameter group settings

Name	Description	Family	Global
atai-valkey8-params	Parameter group for Valkey 8.0 clusters	valkey8	No

ARN: am:aws:elasticacheus-west-2:716124474177:parametergroup:atai-valkey8-params

Parameters (1/59)

cluster-enabled

1 match

Name	Allowed values	Is modifiable	Node type	Value	Source	Type	Change type	Description
cluster-enabled	yes,no	Yes	All	yes	system	string	requires-reboot	Enable cluster mode

5. Click **Save changes**

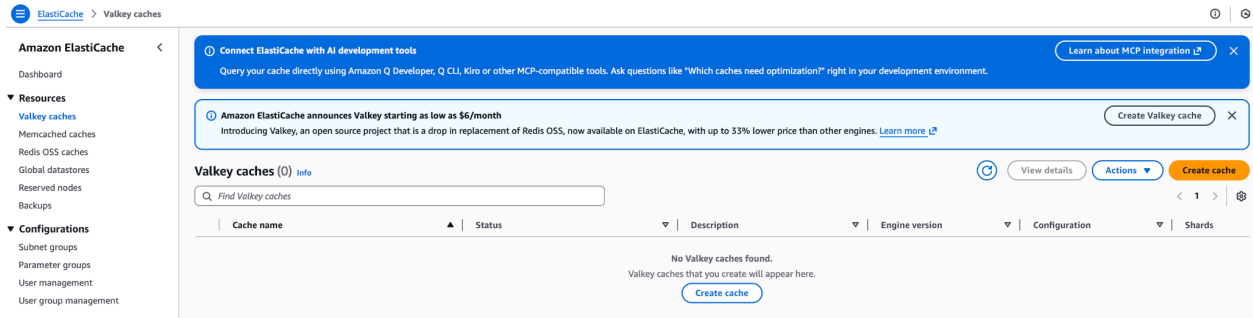
Important: **The cluster-enabled = yes parameter is required for cluster mode to work.** Without it, the clusters won't function properly in cluster mode, even if you enable cluster mode in the cluster settings.

Step 4: Create Valkey Clusters

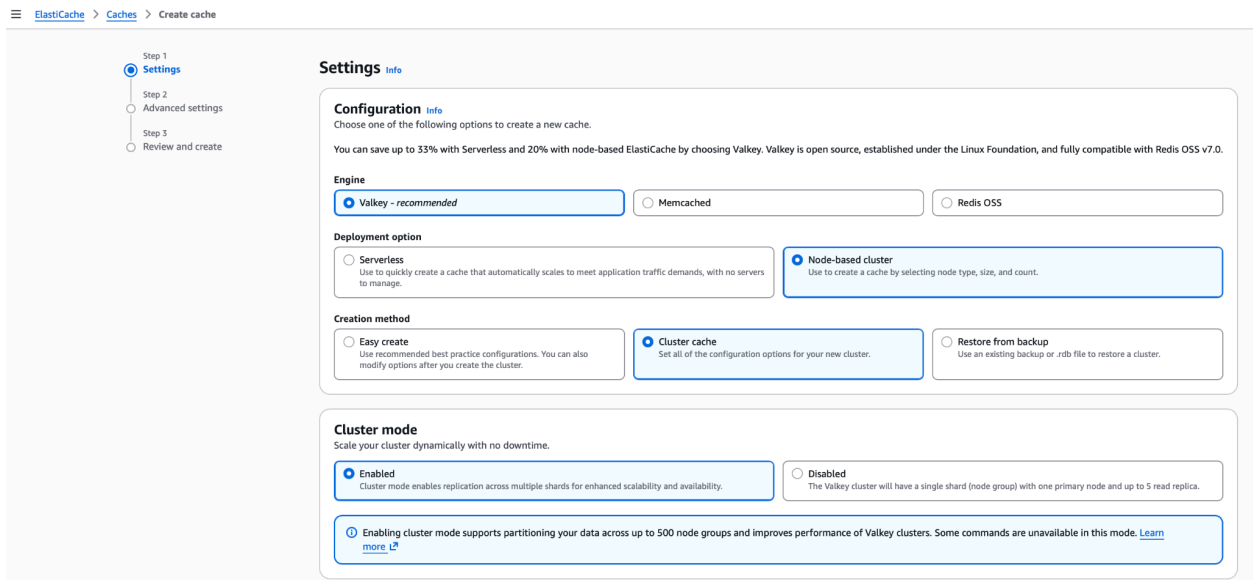
For each cluster, follow these steps.

Cluster 1: registry

1. ElastiCache → Valkey caches → **Create cache**



2. Engine: Valkey
3. Deployment option: Node-based cluster
4. Creation method: Cluster cache
5. Cluster mode: Enable



6. Cluster info
 - a. Name: atai-registry-service
 - b. Description: Registry Service Valkey cluster
7. Location
 - a. Location: AWS Cloud
 - b. Muti-AZ: Disable (to reduce latency)

Cluster info
 Use the following options to configure the cluster.

Name

The name can have up to 40 characters, and must not contain spaces.

Description - optional

Location
 Choose whether to host the cluster in the AWS Cloud or on premises.

Location

☒ **AWS Cloud**
Use the AWS Cloud for your ElastiCache instances.

☐ **On premises**
Create your ElastiCache instances on an Outpost (through AWS Outposts). You need to create a subnet ID on an Outpost first.

Multi-AZ
☐ **Enable**
Multi-AZ provides enhanced high availability through automatic failover to a read replica, cross AZs, in case of a primary node failover.

Auto-failover
☒ **Enable**
ElastiCache Auto Failover provides enhanced high availability through automatic failover to a read replica in case of a primary node failover.

ⓘ Disabling ElastiCache Multi-AZ on your cluster reduces your fault tolerance. In the unlikely event of an Availability Zone failure or loss of network connectivity, your cluster will become unavailable. [Learn more](#)

8. Cache setting
 - a. Engine version: 8.0
 - b. Port: 6379
 - c. Parameter group: Select the parameter group created in the Step 3 of this section
 - d. Node type: cache.t4g.small
 - e. Number of shards: 1
 - f. Replicas per shard: 0

cache.t4g.small

Cache settings
 Use the following options to configure the cluster.

Engine version
Version compatibility of the Valkey engine that will run on your nodes.

Port
The port number that nodes accept connections on.

Parameter groups
Parameter groups control the runtime properties of your nodes and clusters.
 ⓘ

Node type
The type of node to be deployed and its associated memory size.

1.37 GiB memory Up to 5 Gigabit network performance

Number of shards
Enter the number of shards in this cluster, from 1 to 500.

Replicas per shard
Enter the number of replicas for each shard, from 0 to 5.

ⓘ Multi-AZ can not be enabled when the number of replicas is set to 0. Select one or more replicas to enable Multi-AZ.

9. Connectivity:

- a. **Subnet groups:** Select your database subnet group (single AZ) from Step 1

Connectivity
Choose the IP version(s) this cluster will support. Then select an existing subnet group or create a new one.

Network type
Choose between IPv4, dual stack and IPv6

IPv4
Your resources will communicate only over the IPv4 protocol.

Subnet groups
☒ Choose existing subnet group ☐ Create a new subnet group

Subnet groups
A collection of subnets that you can designate for your clusters running in an Amazon VPC.
atal-db-subnet-group (vpc-0a79faee3e664a31d)

The subnet group that you chose supports only one Availability Zone. For high availability using Multi-AZ, modify this subnet group or choose another subnet group.

Associated subnets (1) [Modify](#)

Availability Zone	Subnet ID	CIDR block (IPv4)
us-west-2a	subnet-06819ecd03094dea7	10.5.88.0/24

- b. Use the default Availability Zone placements -> Next

Availability Zone placements
Use the following fields to configure placements for Availability Zones.

Slots and keyspaces
Distribution of the 16,384 cluster keyspace slots across shards.
Equal distribution

Availability Zone placements
By locating nodes in different Availability Zones, you reduce the chance that a failure in one Availability Zone, such as a power outage, will cause your entire system to fail. Choose **Specify Availability Zones** if you want to specify Availability Zones for cluster nodes.
No preference

Shards

Slots/keyspaces

Primary

Shard 1

Equal distribution

No preference

[Cancel](#) [Next](#)

10. Enable encryption at rest and Encryption in transit

Step 1
Settings

**Step 2
Advanced settings**

Step 3
Review and create

Advanced settings [Info](#)

Security
Use the following section to configure network security and data security for your cluster.

Encryption at rest
☒ **Enable**
Enables encryption of data stored on disk.

Encryption key
The master key that will be used to protect the key used to encrypt data at rest.
☒ **Default key**
An AWS owned key will be used for encryption.
☐ **Customer managed CMK**
Select a customer managed key.

Encryption in transit [Info](#)
☒ **Enable**
Enables encryption of data that moves between the service and client.

Transit encryption mode
Required

In Required mode, the cluster will support only encrypted TLS connections. Transit encryption mode can be modified after the cluster has been created. [Learn more](#)

11. For access control choose **AUTH default user access**

- Important: Save this auth token securely. It is required later to communicate with your Valkey cluster.
- Security recommendation: Create a unique auth token per Valkey cluster (do not reuse the same token across clusters).
- Store each token in a secure location such as AWS Secrets Manager.

Access control

Provides the ability to configure authenticating and authorizing access.

AUTH default user access

AUTH token

The AUTH token used for the cluster.

☐ Show token

At least 16 characters and a maximum of 128 characters, which can be any printable ASCII character except for ' ' (blank space), " (quotation mark), /, and @.

12. Security groups

- Click on **Manage**

Selected security groups (0)

A security group acts like a firewall that controls network access to your clusters.

Manage

Group ID



Name



No selected security groups

Add security groups by choosing the Manage button.

Manage

- Select the security group created in Step 2
- Click on **Choose**

Manage security groups



Security groups (1/2)



Find security groups

< 1 >



Security group ID



Security group name



Description



[sg-0a973800a58d354aa](#)

atai-valkey-sg

Security group for Valkey clusters



[sg-0c91975721df6571f](#)

default

default VPC security group

Cancel

Choose

13. Use the default configuration for **Backups, Maintenance, Logs and Tags.**

Backup
You can use backups to restore a cluster or seed a new cluster. The backup consists of the cluster's metadata, along with all of the data in the cluster.

☒ **Enable automatic backups**
Elasticache will automatically create a daily backup of a set of replicas.

Backup retention period
The number of days for which automated backups are retained before they're automatically deleted.

1

Backup window
The daily time range during which automatic backups start if they're enabled.

☒ **No preference**
☐ Specify backup window

Maintenance
Configure maintenance settings for the cluster.

Maintenance window
Specify the time range (UTC) for updates such as patching an operating system, updating drivers, and installing software or patches.

☒ **No preference**
☐ Specify maintenance window

Auto upgrade minor versions

☒ **Enable**
Automatically schedule cluster upgrade to the latest minor version, once it becomes available. Cluster upgrade will only be scheduled during the maintenance window.

Topic for Amazon SNS notification
Choose an SNS topic from the list, or enter the Amazon Resource Name (ARN) for an existing topic. If no topic is chosen, no notifications are sent.

Disable notifications

Logs
Specify whether to provide the Valkey slow logs or engine logs.

Slow logs

☐ **Enable**
Provide the slow log for queries that exceed a specified runtime.

Engine logs

☐ **Enable**
Provide the engine log for the cluster.

Tags
You can use tags to search and filter your clusters, or track your AWS costs.

No tags associated with the cluster.

Add new tag

You can add 50 more tags.

Cancel

Previous

Next

14. Click Next

15. Review your configuration

Step 1

Settings

Step 2

Advanced settings

Step 3

Review and create

Review and create

Step 1: Settings

Cluster info

Use the following options to configure the cluster.

Name

atai-access-manager-service

Description

Access Manager Service Valkey cluster

Location

Choose whether to host the cluster in the AWS Cloud or on premises.

Location

aws-cloud

Cluster mode

Enabled

Cache settings

Use the following options to configure the cluster.

Engine

Valkey

Engine version

8.0

Port

6379

Parameter groups

atai-valkey8-params

Node type

cache.t4g.small

Number of shards

1

Replicas per shard

0

Connectivity

Choose the IP version(s) this cluster will support. Then select an existing subnet group or create a new one.

Network type

IPv4

Subnet group

atai-db-subnet-group

Availability Zones

us-west-2a

Step 2: Advanced settings

Security

Use the following section to configure network security and data security for your cluster.

Security groups

sg-0a973800a58d354aa

Encryption at rest

Enabled

Encryption key

default-key

Encryption in transit

Enabled

Access control

redis-auth-token

AUTH token

UQwFeJvZ2KNpodmMWG7]

Transit encryption mode

Required

Backup

You can use backups to restore a cluster or seed a new cluster. The backup consists of the cluster's metadata, along with all of the data in the cluster.

Automatic backups

Enabled

Backup retention period

1 day

Backup window

No preference

Maintenance

Specify the time range (UTC) for updates such as patching an operating system, updating drivers, and installing software or patches.

Maintenance window

No preference

Auto upgrade minor versions

Enabled

Topic for Amazon SNS notification

SNS topic notifications disabled

Logs

Specify whether to provide the Valkey slow logs or engine logs.

Slow logs

Disabled

Engine logs

Disabled

Tags

You can use tags to search and filter your clusters, or track your AWS costs.

Key

Value

No tags found.
Tags that you create will appear here.

Cancel

Previous

Create

16. Click on **Create**

⚠ Store your Valkey host, and authtoken in a secure location. You will need them later in the *atai-platform* prerequisites.

Cluster 2: access-manager

Same as Cluster 1, but:

1. Name: atai-access-manager-service
2. Auth token: Generate a new unique token (different from previous clusters)

⚠ Store your Valkey host, and token in a secure location. You will need them later in the atai-platform prerequisites.

Cluster 3: jos

Same as Cluster 1, but:

3. Name: atai-jos-service
4. Auth token: Generate a new unique token (different from previous clusters)

⚠ Store your Valkey host, and token in a secure location. You will need them later in the atai-platform prerequisites.

Cluster 4: data-service (10 shards)

Same as Cluster 1, but:

1. Name: atai-data-service
2. Number of shards: 10 (instead of 1)
3. Replicas per shard: 0 (same for the other 10 shard caches)
4. Auth token: Generate a new unique token (different from previous clusters)

⚠ Store your Valkey host, and token in a secure location. You will need them later in the atai-platform prerequisites.

Cluster 5: gpq (10 shards)

Same as Cluster 1, but:

1. Name: atai-gpq-service
2. Number of shards: 10 (instead of 1)
3. Replicas per shard: 0 (same for the other 10 shard caches)
4. Auth token: Generate a new unique token (different from previous clusters)

⚠ Store your Valkey host, and token in a secure location. You will need them later in the atai-platform prerequisites.

Cluster 6: health

Same as Cluster 1, but:

1. Name: atai-health-service
2. Auth token: Generate a new unique token (different from previous clusters)

⚠ Store your Valkey host, and token in a secure location. You will need them later in the atai-platform prerequisites.

Cluster 7: lens (10 shards)

Same as Cluster 1, but:

1. Name: atai-lens-service
2. Number of shards: 10 (instead of 1)
3. Replicas per shard: 0 (same for the other 10 shard caches)
4. Auth token: Generate a new unique token (different from previous clusters)

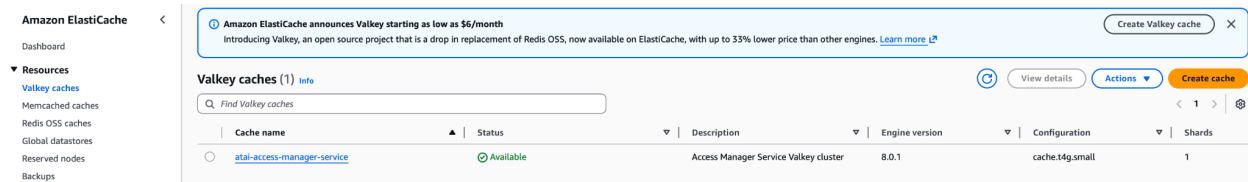
⚠ Store your Valkey host, and token in a secure location. You will need them later in the atai-platform prerequisites.

Step 5: Store Auth Tokens in AWS Secrets Manager (Recommended)

Note: This step is recommended but not mandatory. You can store credentials in Secrets Manager for easier access, better security, and integration with your applications. If you prefer to manage credentials differently, you can skip this step.

For each cluster, store the auth token and connection details:

1. Go to ElastiCache → Valkey caches → Click on your Valkey cache cluster



2. Copy the **Configuration endpoint**

ElastiCache > Valkey caches > atai-access-manager-service

Amazon ElastiCache

Dashboard

▼ **Resources**

Valkey caches

Memcached caches

Redis OSS caches

Global datastores

Reserved nodes

Backups

▼ **Configurations**

Subnet groups

Parameter groups

User management

User group management

Events

Service updates

atai-access-manager-service Info

▼ **Cluster details**

Cluster name atai-access-manager-service	Description Access Manager Service Valkey cluster	Node type cache.t4g.small
Engine Valkey	Engine version 8.0.1	Global datastore -
Update status Up to date	Cluster mode Enabled	Shards 1
Data tiering Disabled	Multi-AZ Disabled	Auto-failover Enabled
Encryption at rest Enabled	Parameter group atai-valkey8-params	Outpost ARN -
Configuration endpoint clustercfg.atai-access-manager-service.0w0z5r.usw2.ca che.amazonaws.com:6379	Primary endpoint -	Reader endpoint -
Data migration No active migrations		

3. Go to Secrets Manager → **Store a new secret**

AWS Secrets Manager

Secrets

Security, Identity and Compliance

AWS Secrets Manager

Easily rotate, manage and retrieve secrets throughout their lifecycle

AWS Secrets Manager helps you protect access to your applications, services and IT resources. You can easily rotate, manage and retrieve database credentials, API keys and other secrets throughout their lifecycles.

Get started

You can store database credentials or any other type of secret.

Store a new secret

4. Secret type: Other type of secret

5. Key/value pairs: Add:

- host: Configuration endpoint address (from cluster details, e.g., [atai-access-manager-service.xxxxx.cache.amazonaws.com](#))
- port: 6379
- auth_token: The auth token you set for this cluster

AWS Secrets Manager > Secrets > Store a new secret

Step 1
 ● Choose secret type
 ○ Step 2
 ○ Configure secret
 ○ Step 3 - optional
 ○ Configure rotation
 ○ Step 4
 ○ Review

Choose secret type

Secret type Info

☐ Credentials for Amazon RDS database
 ☐ Credentials for Amazon DocumentDB database
 ☐ Credentials for Amazon Redshift data warehouse

☐ Credentials for other database
 ☒ Other type of secret
API key, OAuth token, other.

Key/value pairs Info

Key/value | Plaintext

host	clustercfg.atai-access-manager-service.0w0z5r.usw2.cache.amazonaws.com	Remove
port	6379	Remove
auth	*****	Remove

[+ Add row](#)

Encryption key Info

You can encrypt using the KMS key that Secrets Manager creates or a customer-managed KMS key that you create.

aws/secretsmanager [Add new key](#)

[Cancel](#) [Next](#)

6. Secret name: atai/valkey/{service-name}

Examples:

- atai/valkey/access-manager
- atai/valkey/api-events
- atai/valkey/dfc
- etc.

7. Encryption key: Use AWS managed key (default) or your KMS key

8. Click Next → Next

AWS Secrets Manager > Secrets > Store a new secret

Step 1
 ○ Choose secret type
 ● Step 2
 ● Configure secret
 ○ Step 3 - optional
 ○ Configure rotation
 ○ Step 4
 ○ Review

Configure secret

Secret name and description Info

Secret name
 A descriptive name that helps you find your secret later.
 atai/valkey/access-manager
Secret name must only contain alphanumeric characters and the characters /, +, @.

Description - optional
 Credentials of the Access Manager Valkey
Maximum 250 characters.

Tags - optional
 No tags associated with the secret.
[Add](#)

Resource permissions - optional Info [Edit permissions](#)
 Add or edit a resource policy to access secrets across AWS accounts.

▶ **Replicate secret - optional**
 Create read-only replicas of your secret in other regions. Replica secrets incur a charge.

[Cancel](#) [Previous](#) [Next](#)

9. Do not configure rotation Click Next

Step 1

Choose secret type

Step 2

Configure secret

Step 3 - optional

Configure rotation

Step 4

Review

Configure rotation - optional

Configure automatic rotation [Info](#)

Configure AWS Secrets Manager to rotate this secret automatically.

☒ Automatic rotation

Rotation schedule [Info](#)

☒ Schedule expression builder

☐ schedule expression

Time unit

Hours

Hours

23

Window duration - optional

4h

Enter the time in hours.

☒ Rotate immediately when the secret is stored. The next rotation will begin on your schedule.

Rotation function [Info](#)

Lambda rotation function [Info](#)

Choose a Lambda function that can rotate this secret.

aws-controltower-NotificationForwarder

Create function [↗](#)

Cancel

Previous

Next

10. Review and click on **Store**

Step 1

Choose secret type

Step 2

Configure secret

Step 3 - optional

Configure rotation

Step 4

Review

Review

Secret type

Secret type

Other type of secret

Encryption key

aws/secretsmanager

Secret configuration

Secret name

ata/vailey/access-manager

Description

Credentials of the Access Manager Vailey

Tags

-

Resource permissions

-

Secret replication

Disabled

Rotation schedule

Automatic rotation

Disabled

Rotation schedule

-

Rotation function

Lambda rotation function

-

Secret that performs rotation

-

Sample code

Use these code samples to retrieve the secret in your application.

Java

JavaScript

C#

Python3

Ruby

Go

Rust

PHP

```
1 // Use this code snippet in your app.
2 // If you need more information about configurations or implementing the sample
3 // code, visit the AWS docs:
4 // https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/home.html
5
6 // Make sure to import the following packages in your code
7 // import software.amazon.awssdk.regions.Region;
8 // import software.amazon.awssdk.services.secretsmanager.SecretsManagerClient;
9 // import software.amazon.awssdk.services.secretsmanager.model.GetSecretValueRequest;
10 // import software.amazon.awssdk.services.secretsmanager.model.GetSecretValueResponse;
11
12 public static void getSecret() {
13
14     String secretName = "atal/valkey/access-manager";
15     Region region = Region.of("us-west-2");
16 }
```

Java Line 1, column 1 0 Errors: 0 0 Warnings: 0

[Download AWS SDK for Java](#)

Cancel

Previous

Store

11. Repeat for all 8 clusters with their respective:
- Configuration endpoint (host)
 - Auth token
 - Service name in the secret path

Benefits of using Secrets Manager:

- Applications can retrieve credentials programmatically
- Credentials are encrypted at rest
- Access is logged for audit purposes
- No need to hardcode credentials in application code

Alternative: If you skip this step, ensure you have the auth tokens and configuration endpoints stored securely elsewhere, as they are required for applications to connect to the Valkey clusters.

 Store your Valkey host, port, and token in a secure location. You will need them later in the *atal-platform* prerequisites.

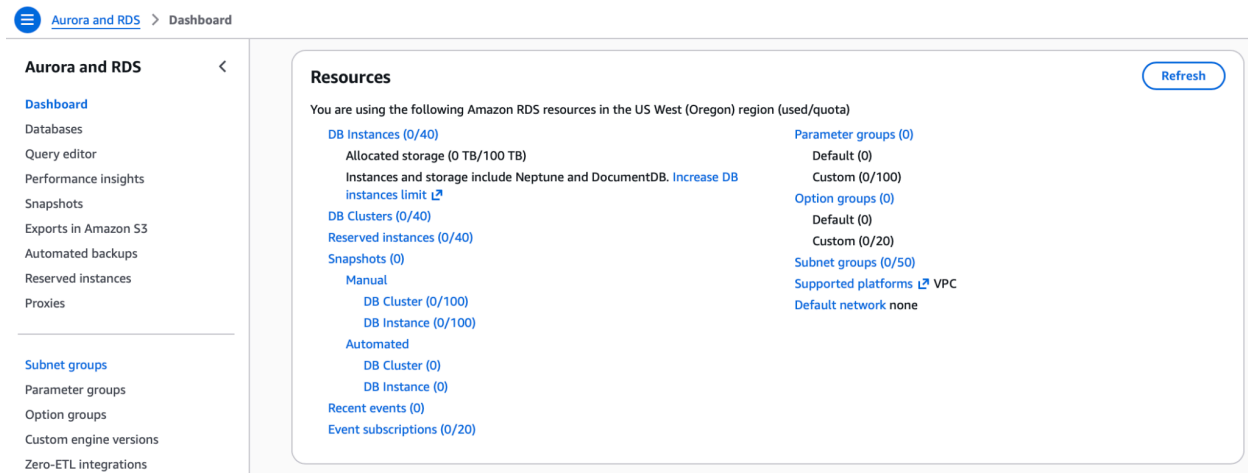
PostgreSQL database configuration

Prerequisites

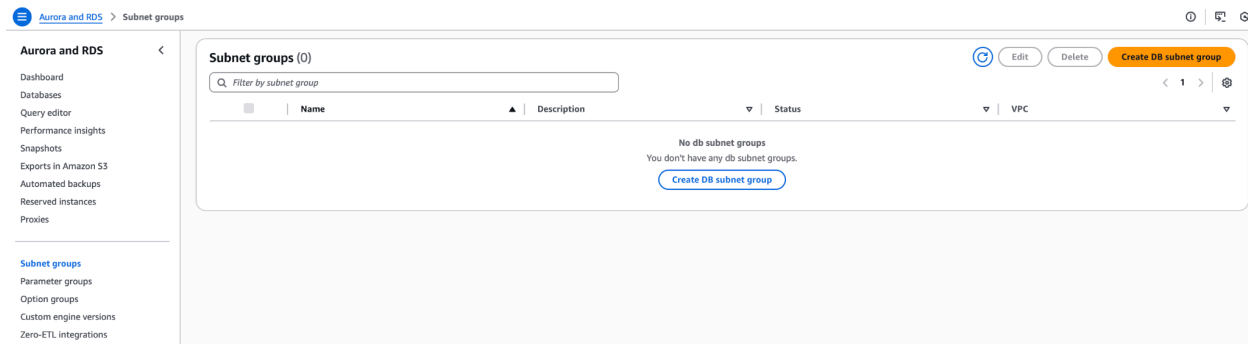
1. VPC with database subnets (at least 2 AZs required for Aurora subnet group)
2. Security group allowing access from EKS pods
3. Database subnet group

Step 1: Create Database Subnet Group

1. Go to RDS dashboard



2. Subnet groups → Create DB subnet group



3. Name: atai-db-subnet-group (or your name)
4. Description: Subnet group for Aurora PostgreSQL cluster
5. VPC: Select your VPC

☰ [Aurora and RDS](#) > [Subnet groups](#) > Create DB subnet group

Create DB subnet group

To create a new subnet group, give it a name and a description, and choose an existing VPC. You will then be able to add subnets related to that VPC.

Subnet group details

Name
You won't be able to modify the name after your subnet group has been created.

Must contain from 1 to 255 characters. Alphanumeric characters, spaces, hyphens, underscores, and periods are allowed.

Description

VPC
Choose a VPC identifier that corresponds to the subnets you want to use for your DB subnet group. You won't be able to choose a different VPC identifier after your subnet group has been created.

6 Subnets, 2 Availability Zones

6. Availability Zones: Select at least 2 AZs (AWS requirement for Aurora)
 - a. Note: Aurora requires at least 2 AZs for the subnet group, even if you deploy a single instance
 - b. Example: Select us-west-2a and us-west-2b
7. Subnets: Select your database subnets atai-platform-vpc-database-us-west-2a and atai-platform-vpc-database-us-west-2b (e.g., 10.5.88.0/24 in us-west-2a and 10.5.89.0/24 in us-west-2b)

Add subnets

Availability Zones
Choose the Availability Zones that include the subnets you want to add.

Choose an availability zone

us-west-2a × us-west-2b ×

Subnets
Choose the subnets that you want to add. The list includes the subnets in the selected Availability Zones.

Select subnets

atai-platform-vpc-database-us-west-2b × atai-platform-vpc-database-us-west-2a ×

Subnet ID: subnet-06bd5873d34243b83 CIDR: 10.5.89.0/24 Subnet ID: subnet-06819ecd03094dea7 CIDR: 10.5.88.0/24

ⓘ For Multi-AZ DB clusters, you must select 3 subnets in 3 different Availability Zones.

Subnets selected (2)

Availability zone	Subnet name	Subnet ID	CIDR block
us-west-2b	atai-platform-vpc-database-us-west-2b	subnet-06bd5873d34243b83	10.5.89.0/24
us-west-2a	atai-platform-vpc-database-us-west-2a	subnet-06819ecd03094dea7	10.5.88.0/24

Cancel Create

8. Click Create

Step 2: Create Security Group (if not existing)

1. Go to VPC → Security Groups → Create security group

☰ [VPC](#) > Security Groups

VPC dashboard < AWS Global View ⓘ Filter by VPC: ▼

▼ **Virtual private cloud**
Your VPCs
Subnets
Route tables
Internet gateways
Egress-only Internet gateways
Carrier gateways
DHCP option sets
Elastic IPs
Managed prefix lists
NAT gateways
Peering connections
Route servers [New](#)

▼ **Security**
Network ACLs
[Security groups](#)

Security Groups (2) ⓘ

Find security groups by attribute or tag

☐	Name	Security group ID	Security group name	VPC ID	Description
☐	-	sg-0c91975721df6571f	default	vpc-0a79faee3e664a31d	default VPC security group

Select a security group

2. Name: atai-rds-sg (or your name)
3. Description: Security group for Aurora PostgreSQL cluster
4. VPC: Select your VPC from section VPC configuration Step 1
5. Inbound rules: Add rule:
 - a. Type: PostgreSQL
 - b. Port: 5432
 - c. Source: Custom → Enter your VPC CIDR (e.g., 10.5.0.0/16)
 - d. Description: Allow PostgreSQL access from VPC

☰ VPC > Security Groups > Create security group

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, complete the fields below.

Basic details

Security group name Info

Name cannot be edited after creation.

Description Info

VPC Info

vpc-0a79faee3e664a31d (atai-platform-vpc) ▾

Inbound rules Info

Type Info	Protocol Info	Port range Info	Source Info	Description - optional Info
PostgreSQL ▾	TCP	5432	Custom ▾ 10.5.0.0/16 X 10.5.0.0/16 X	Allow PostgreSQL access from VPC Allow PostgreSQL access from VPC

[Add rule](#)

Outbound rules Info

Type Info	Protocol Info	Port range Info	Destination Info	Description - optional Info
All traffic ▾	All	All	Custom ▾ 0.0.0.0/0 X	

[Add rule](#)

⚠ Rules with destination of 0.0.0.0/0 or ::/0 allow your instances to send traffic to any IPv4 or IPv6 address. We recommend setting security group rules to be more restrictive and to only allow traffic to specific known IP addresses.

Tags - optional

A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

No tags associated with the resource.

[Add new tag](#)

You can add up to 50 more tags

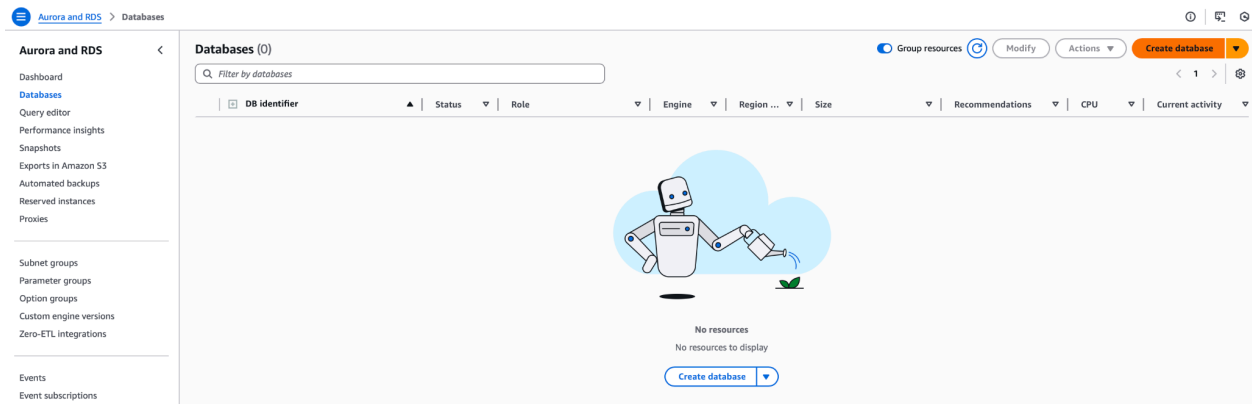
[Cancel](#) [Create security group](#)

Note: For initial setup, opening to the VPC CIDR simplifies connectivity. Later, restrict to specific security groups (e.g., EKS node group security group) for tighter security.

Click Create security group

Step 3: Create Aurora PostgreSQL Cluster

1. Go to RDS → Databases → Create database



2. Database creation method: Full configuration

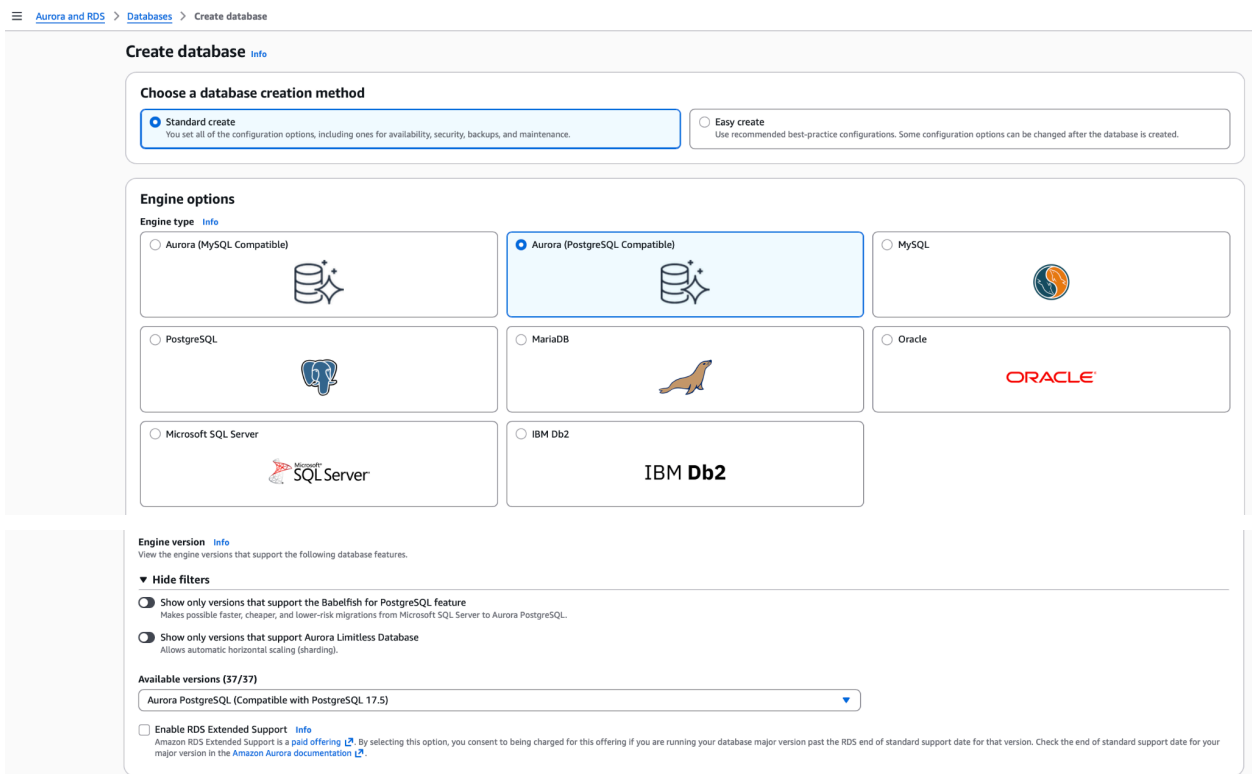
3. Engine options:

4. Engine type: Amazon Aurora

a. Edition: Amazon Aurora PostgreSQL-Compatible Edition

b. Available versions: Select Aurora PostgreSQL 17.5

c. Templates: Production (or Dev/Test for non-production)



Templates

Choose a sample template to meet your use case.

☒ Production

Use defaults for high availability and fast, consistent performance.

☐ Dev/Test

This instance is intended for development use outside of a production environment.

5. Settings:

- DB cluster identifier: atai-platform
- Master username: postgres (or your preferred username)
- Credentials management: Managed in AWS secrets manager

Settings

DB cluster identifier [Info](#)

Type a name for your DB cluster. The name must be unique across all DB clusters owned by your AWS account in the current AWS Region.

atai-platform

The DB cluster identifier is case-insensitive, but is stored as all lowercase (as in "mydbcluster"). Constraints: 1 to 63 alphanumeric characters or hyphens. First character must be a letter. Can't contain two consecutive hyphens. Can't end with a hyphen.

▼ Credentials Settings

Master username [Info](#)

Type a login ID for the master user of your DB instance.

postgres

1 to 16 alphanumeric characters. The first character must be a letter.

Credentials management

You can use AWS Secrets Manager or manage your master user credentials.

☒ Managed in AWS Secrets Manager - most secure

RDS generates a password for you and manages it throughout its lifecycle using AWS Secrets Manager.

☐ Self managed

Create your own password or have RDS create a password that you manage.

ⓘ If you manage the master user credentials in AWS Secrets Manager, additional charges apply. See [AWS Secrets Manager pricing](#). Additionally, some RDS features aren't supported. See [limitations](#).

Select the encryption key [Info](#)

You can encrypt using the KMS key that Secrets Manager creates or a customer managed KMS key that you create.

aws/secretsmanager (default)

[Add new key](#)

6. Cluster storage configuration

- Storage type: Aurora Standard

Cluster storage configuration [Info](#)

Choose the storage configuration for the Aurora DB cluster that best fits your application's price predictability and price performance needs.

Configuration options

Database instance, storage, and I/O charges vary depending on the configuration. [Learn more](#).

☐ Aurora I/O-Optimized

- Predictable pricing for all applications. Improved price performance for I/O-intensive applications (I/O costs >25% of total database costs).
- No additional charges for read/write I/O operations. DB instance and storage prices include I/O usage.

☒ Aurora Standard

- Cost-effective pricing for many applications with moderate I/O usage (I/O costs <25% of total database costs).
- Pay-per-request I/O charges apply. DB instance and storage prices don't include I/O usage.

7. Instance configuration

- DB instance class: Select your instance class (e.g., db.t4g.medium for dev, db.r7g.large for production)

Instance configuration

The DB instance configuration options below are limited to those supported by the engine that you selected above.

DB instance class [Info](#)

▼ Hide filters

☒ Include previous generation classes

☐ Serverless v2

☒ Memory optimized classes (includes r classes)

☐ Burstable classes (includes t classes)

☐ Optimized Reads classes

db.r7g.large

2 vCPUs 16 GiB RAM EBS Bandwidth: Up to 10,000 Mbps Network: Up to 12.5 Gbps

8. For Availability and Durability select Don't create an Aurora Replica

Availability & durability

Multi-AZ deployment [Info](#)

- ☐ Create an Aurora Replica or Reader node in a different AZ (recommended for scaled availability)
Creates an Aurora Replica for fast failover and high availability.
- ☒ Don't create an Aurora Replica

9. Connectivity:

- Virtual private cloud (VPC): Select your VPC
- DB subnet group: Select your database subnet group (created in Step 1)
- Publicly accessible: No (should be in private subnets)

Connectivity [Info](#)

Compute resource

Choose whether to set up a connection to a compute resource for this database. Setting up a connection will automatically change connectivity settings so that the compute resource can connect to this database.

- ☒ Don't connect to an EC2 compute resource
Don't set up a connection to a compute resource for this database. You can manually set up a connection to a compute resource later.
- ☐ Connect to an EC2 compute resource
Set up a connection to an EC2 compute resource for this database.

Network type [Info](#)

To use dual-stack mode, make sure that you associate an IPv6 CIDR block with a subnet in the VPC you specify.

- ☒ IPv4
Your resources can communicate only over the IPv4 addressing protocol.
- ☐ Dual-stack mode
Your resources can communicate over IPv4, IPv6, or both.

Virtual private cloud (VPC) [Info](#)

Choose the VPC. The VPC defines the virtual networking environment for this DB cluster.

atai-platform-vpc (vpc-0a79faee3e64a31d)
6 Subnets, 2 Availability Zones

Only VPCs with a corresponding DB subnet group are listed.

ⓘ After a database is created, you can't change its VPC.

DB subnet group [Info](#)

Choose the DB subnet group. The DB subnet group defines which subnets and IP ranges the DB cluster can use in the VPC that you selected.

atai-db-subnet-group
2 Subnets, 2 Availability Zones

Public access [Info](#)

- ☐ Yes
RDS assigns a public IP address to the cluster. Amazon EC2 instances and other resources outside of the VPC can connect to your cluster. Resources inside the VPC can also connect to the cluster. Choose one or more VPC security groups that specify which resources can connect to the cluster.
- ☒ No
RDS doesn't assign a public IP address to the cluster. Only Amazon EC2 instances and other resources inside the VPC can connect to your cluster. Choose one or more VPC security groups that specify which resources can connect to the cluster.

- VPC security group: Choose existing → Select atai-rds-sg
- Availability Zone: No preference (AWS will choose) or select your preferred AZ for the primary instance

VPC security group (firewall) [Info](#)

Choose one or more VPC security groups to allow access to your database. Make sure that the security group rules allow the appropriate incoming traffic.

- ☒ Choose existing
Choose existing VPC security groups
- ☐ Create new
Create new VPC security group

Existing VPC security groups

Choose one or more options

atai-rds-sg × default ×

Certificate authority - optional [Info](#)

Using a server certificate provides an extra layer of security by validating that the connection is being made to an Amazon database. It does so by checking the server certificate that is automatically installed on all databases that you provision.

rds-ca-rsa2048-g1 (default)
Expiry: May 24, 2061

If you don't select a certificate authority, RDS chooses one for you.

RDS Data API

- ☐ Enable the RDS Data API [Info](#)
Enable the SQL HTTP endpoint for the Data API. With this endpoint enabled, you can run SQL queries against this database over HTTP. You can do so by using the CLI, an AWS SDK, or the RDS query editor. For information about pricing, see [Amazon RDS pricing](#).

Note: For low latency, you can select the same AZ as your EKS node groups (e.g., us-west-2a)

10. Leave the default values for Read replica write forwarding, Tags, Babelfish settings and Database authentication.

Read replica write forwarding
☐ Turn on local write forwarding [Info](#)
Issues write operations from reader DB instances within the same DB cluster.

Tags - optional
A tag consists of a case-sensitive key-value pair.
No tags associated with the resource.
[Add new tag](#)
You can add up to 50 more tags.

Babelfish settings [Info](#)
☐ Turn on Babelfish
Makes possible faster, cheaper, and lower-risk migrations from Microsoft SQL Server to Aurora PostgreSQL.

Database authentication [Info](#)
Password authentication is always active for your database engine. You can also turn on additional authentication methods for your database below.
☐ IAM database authentication
Authenticates using IAM database authentication.
☐ Kerberos authentication
Authenticates using Kerberos authentication through an AWS Directory Service for Microsoft Active Directory.

11. Monitoring

a. Select Database insight - Standard

Monitoring [Info](#)
Choose monitoring tools for this database. Database Insights provides a combined view of Performance Insights and Enhanced Monitoring for your fleet of databases. Database Insights pricing is separate from RDS monthly estimates. See [Amazon CloudWatch pricing](#).

☐ Database Insights - Advanced

- Retains 15 months of performance history
- Fleet-level monitoring
- Integration with CloudWatch Application Signals

☒ Database Insights - Standard

- Retains 7 days of performance history, with the option to pay for the retention of up to 24 months of performance history

Performance Insights
☒ Enable Performance Insights
With Performance Insights dashboard, you can visualize the database load on your Amazon RDS DB Instance load and filter the load by waits, SQL statements, hosts, or users.

Retention period

7 days

AWS KMS key [Info](#)

(default) aws/rds

b. Disable Enhanced monitoring

c. Select PostgreSQL log under the Logs exports section

d. You can turn off DevOps Guru to avoid extra costs

Additional monitoring settings
Enhanced Monitoring, CloudWatch Logs and DevOps Guru

Enhanced Monitoring
☐ Enable Enhanced monitoring
Enabling Enhanced Monitoring metrics are useful when you want to see how different processes or threads use the CPU.

Log exports
Select the log types to publish to Amazon CloudWatch Logs
☐ iam-db-auth-error log
☐ instance log
☒ PostgreSQL log

IAM role
The following service-linked role is used for publishing logs to CloudWatch Logs.

RDS service-linked role

DevOps Guru
☐ Turn on DevOps Guru [Info](#)
DevOps Guru for RDS automatically detects performance anomalies for DB instances and provides recommendations.

12. Additional configuration:

- Initial database name: Leave blank (or enter a name if you want to create an initial database)
- DB cluster parameter group: default.aurora-postgresql17 (or create custom if needed)
- DB parameter group: default.aurora-postgresql17 (or create custom if needed)
- Backup retention period: 7 days (default, or your preference)
- Backup window: No preference (default, or set a specific window)
- Enable encryption: Enable encryption (it's enabled by default)
- Encryption key: Use AWS managed key (default) or your KMS key

▼ Additional configuration

Database options, encryption turned on, failover, backup turned on, backtrack turned off, maintenance, CloudWatch Logs, delete protection turned on.

Database options

Initial database name [Info](#)

If you do not specify a database name, Amazon RDS does not create a database.

DB cluster parameter group [Info](#)

default.aurora-postgresql17

▼

DB parameter group [Info](#)

default.aurora-postgresql17

▼

Option group [Info](#)

default.aurora-postgresql-17

▼

Failover priority

No preference

▼

Backup

Backup retention period [Info](#)

The number of days (1-35) for which automatic backups are kept.

7

▼

days

☒ Copy tags to snapshots

☒ Enable encryption

Choose to encrypt the given instance. Master key IDs and aliases appear in the list after they have been created using the AWS Key Management Service console. [Info](#)

AWS KMS key [Info](#)

(default) aws/rds

▼

Maintenance

Auto minor version upgrade [Info](#)

☒ Enable auto minor version upgrade

Enabling auto minor version upgrade will automatically upgrade your database minor version. For limitations and more details, see [Automatically upgrading the minor engine version documentation](#) [L2](#).

Maintenance window [Info](#)

Select the period you want pending modifications or maintenance applied to the database by Amazon RDS.

☐ Choose a window

☒ No preference

☒ Enable deletion protection

Protects the database from being deleted accidentally. While this option is enabled, you can't delete the database.

Estimated monthly costs

DB instance	402.96 USD
Storage	0.10 USD
Total	403.06 USD

This billing estimate is based on on-demand usage as described in [Amazon Aurora Pricing](#) [L2](#). Estimate does not consider reserved instance benefits and costs for instance storage, I/Os, or data transfer.

Estimate your monthly costs for the DB Instance using the [AWS Simple Monthly Calculator](#) [L2](#).

ⓘ You are responsible for ensuring that you have all of the necessary rights for any third-party products or services that you use with AWS services.

Cancel Create database

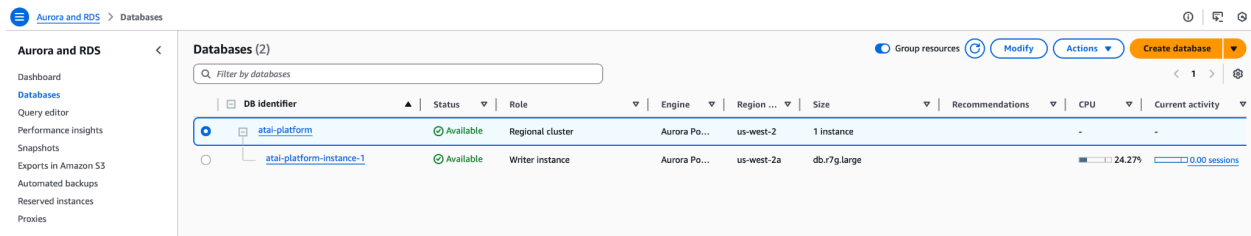
Note: No additional configurations are required — defaults work. You can customize backup retention, monitoring, and other settings if needed.

13. Click on **Create database**

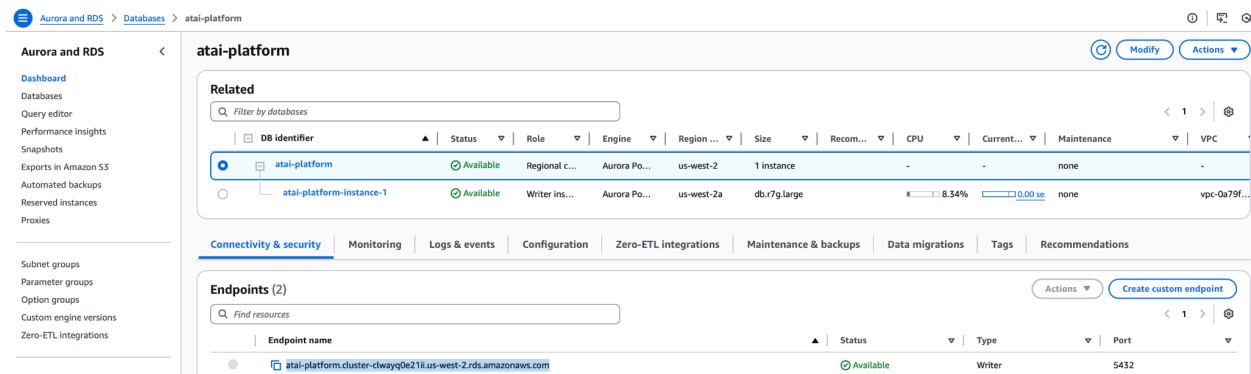
Archetype AI – Confidential 2026 ©

14. Get my database hostname

- After you create your database go to the RDS dashboard → **Databases**
- Select your database cluster



- Save the **Cluster endpoint** - will be needed for the helm values file

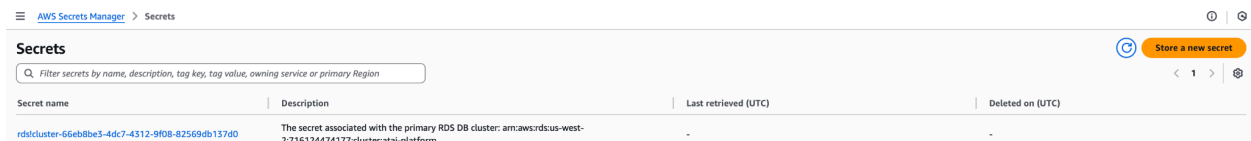


15. Get my database credentials

- Go to Secrets Manager → **Secrets**



- Click on the new RDS secret



- Click on **Get secret value**. You'll be able to see the user and password to connect to your RDS postgresQL.

rdscluster-66eb8be3-4dc7-4312-9f08-82569db137d0

ⓘ This secret was created by Amazon RDS (rds). Because this secret is managed by Amazon RDS (rds), you will not be able to modify the secret value. However, the secret may be modified in any other manner. [Learn more](#)

Secret details

Encryption key

aws/secretsmanager

Secret name

rdscluster-66eb8be3-4dc7-4312-9f08-82569db137d0

Secret ARN

arn:aws:secretsmanager:us-west-2:716124474177:secret:rdscluster-66eb8be3-4dc7-4312-9f08-82569db137d0-zOQsWh

Secret description

The secret associated with the primary RDS DB cluster: am:aws:rds:us-west-2:716124474177:cluster:atai-platform

Secret type

-

⌂ Actions

Overview

Rotation

Versions

Replication

Tags

Secret value

info

Retrieve and view the secret value.

Retrieve secret value

Step 4: Extra Database Configuration steps

Connect to your RDS, in our case we are using bastion host machine:

None

```
psql -h your-rds-endpoint.region.rds.amazonaws.com -U master_username -d postgres
```

Create databases

None

```
CREATE DATABASE experiment_logs_db;
CREATE DATABASE iam_db;
CREATE DATABASE platform_api_events_db;
CREATE DATABASE platform_eval_db;
CREATE DATABASE platform_lens_db;
CREATE DATABASE platform_logs_db;
CREATE DATABASE lens_db;
CREATE DATABASE data_service;
CREATE DATABASE jos_db;
```

```
postgres=> \l
```

Name	Owner	Encoding	Locale Provider	Collate	Ctype	Locale	ICU Rules	Access privileges
experiment_logs_db	postgres	UTF8	libc	en_US.UTF-8	en_US.UTF-8			=Tc/postgres + postgres=Ctc/postgres+ atai_dev=Ctc/postgres
iam_db	postgres	UTF8	libc	en_US.UTF-8	en_US.UTF-8			=Tc/postgres + postgres=Ctc/postgres+ atai_dev=Ctc/postgres
lens_db	postgres	UTF8	libc	en_US.UTF-8	en_US.UTF-8			=Tc/postgres + postgres=Ctc/postgres+ atai_dev=Ctc/postgres
platform_api_events_db	postgres	UTF8	libc	en_US.UTF-8	en_US.UTF-8			=Tc/postgres + postgres=Ctc/postgres+ atai_dev=Ctc/postgres
platform_eval_db	postgres	UTF8	libc	en_US.UTF-8	en_US.UTF-8			=Tc/postgres + postgres=Ctc/postgres+ atai_dev=Ctc/postgres
platform_lens_db	postgres	UTF8	libc	en_US.UTF-8	en_US.UTF-8			=Tc/postgres + postgres=Ctc/postgres+ atai_dev=Ctc/postgres
platform_logs_db	postgres	UTF8	libc	en_US.UTF-8	en_US.UTF-8			=Tc/postgres + postgres=Ctc/postgres+ atai_dev=Ctc/postgres
postgres	postgres	UTF8	libc	en_US.UTF-8	en_US.UTF-8			=Tc/postgres + postgres=Ctc/postgres+ atai_dev=Ctc/postgres
rdsadmin	rdsadmin	UTF8	libc	en_US.UTF-8	en_US.UTF-8			rdsadmin=Ctc/rdsadmin
template0	rdsadmin	UTF8	libc	en_US.UTF-8	en_US.UTF-8			=c/rdsadmin + rdsadmin=Ctc/rdsadmin
template1	postgres	UTF8	libc	en_US.UTF-8	en_US.UTF-8			=c/postgres + postgres=Ctc/postgres

(11 rows)

```
postgres=>
```

Create atai_dev database user

None

```
CREATE USER atai_dev WITH PASSWORD 'your_secure_password';
```

(10 rows)

postgres=> \du

List of roles	
Role name	Attributes
atai_dev	
postgres	Create role, Create DB Password valid until infinity +
rds_ad	Cannot login
rds_extension	No inheritance, Cannot login
rds_iam	Cannot login
rds_password	Cannot login
rds_replication	Cannot login
rds_superuser	Cannot login
rdsadmin	Superuser, Create role, Create DB, Replication, Bypass RLS+ Password valid until infinity
rdswriteforwarduser	No inheritance

postgres=>

 Store your atai_dev user and password in a secure location. You will need them later in the *atai-platform* prerequisites.

Then assign proper permissions to the new PostgreSQL user:

None

```
GRANT ALL PRIVILEGES ON DATABASE experiment_logs_db TO atai_dev;
GRANT ALL PRIVILEGES ON DATABASE iam_db TO atai_dev;
GRANT ALL PRIVILEGES ON DATABASE platform_api_events_db TO atai_dev;
GRANT ALL PRIVILEGES ON DATABASE platform_eval_db TO atai_dev;
GRANT ALL PRIVILEGES ON DATABASE platform_lens_db TO atai_dev;
GRANT ALL PRIVILEGES ON DATABASE platform_logs_db TO atai_dev;
GRANT ALL PRIVILEGES ON DATABASE lens_db TO atai_dev;
GRANT ALL PRIVILEGES ON DATABASE postgres TO atai_dev;
GRANT ALL PRIVILEGES ON DATABASE data_service TO atai_dev;
GRANT ALL PRIVILEGES ON DATABASE jos_db TO atai_dev;
```

Additional permissions required:

None

```
\c database_name (this should be done on all DBs)
```

```
GRANT USAGE ON SCHEMA public TO atai_dev;  
GRANT CREATE ON SCHEMA public TO atai_dev;  
GRANT ALL PRIVILEGES ON ALL TABLES IN SCHEMA public TO atai_dev;  
GRANT ALL PRIVILEGES ON ALL SEQUENCES IN SCHEMA public TO atai_dev; ALTER DEFAULT  
PRIVILEGES IN SCHEMA public GRANT ALL ON TABLES TO atai_dev;  
ALTER DEFAULT PRIVILEGES IN SCHEMA public GRANT ALL ON SEQUENCES TO atai_dev;  
GRANT ALL PRIVILEGES ON ALL FUNCTIONS IN SCHEMA public TO atai_dev; ALTER DEFAULT  
PRIVILEGES IN SCHEMA public GRANT ALL ON FUNCTIONS TO atai_dev;
```

Solve a common permission issue:

None

```
-- Issue:psycopg2.errors.InsufficientPrivilege:  
--   permission denied for schema public  
---  LINE 1: CREATE TABLE IF NOT EXISTS platform_lens_db(  
--   Fix: on that specific DB
```

```
GRANT CREATE ON SCHEMA public TO atai_dev;  
GRANT USAGE ON SCHEMA public TO atai_dev;
```

EKS cluster configuration

Prerequisites

1. An existing VPC and subnets that meet Amazon EKS requirements
2. At least two private subnets in your VPC:
 - a. Recommended naming:
 - i. atai-platform-vpc-private-us-west-2a,
 - ii. atai-platform-vpc-private-us-west-2b (or similar)
 - b. Recommended CIDR blocks: /20 per subnet
 - i. Example: 10.5.0.0/20 (subnet 1)
 - ii. Example: 10.5.16.0/20 (subnet 2)
 - c. Subnets must be in different Availability Zones
 - d. Subnets must have routes to NAT Gateway or Internet Gateway for outbound internet access
3. The kubectl command line tool v1.32-v1.34 is required.

4. Version 2.12.3 or later or version 1.27.160 or later of the AWS Command Line Interface (AWS CLI) installed and configured on your device.
5. An IAM principal with permissions to create and describe an Amazon EKS cluster

Step 1: Create cluster IAM role

1. If you already have a cluster IAM role, or you're going to create your cluster with eksctl, then you can skip this step. By default, eksctl creates a role for you.
2. Run the following command to create an IAM trust policy JSON file.

```
None
cat > eks-cluster-role-trust-policy.json << 'EOF'
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "eks.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
EOF
```

3. Create the Amazon EKS cluster IAM role. If necessary, preface eks-cluster-role-trust-policy.json with the path on your computer that you wrote the file to in the previous step. The command associates the trust policy that you created in the previous step to the role. To create an IAM role, the IAM principal that is creating the role must be assigned the iam:CreateRole action (permission).

```
None
aws iam create-role --role-name atai-platform-eks-cluster-role
--assume-role-policy-document file://"eks-cluster-role-trust-policy.json"
```

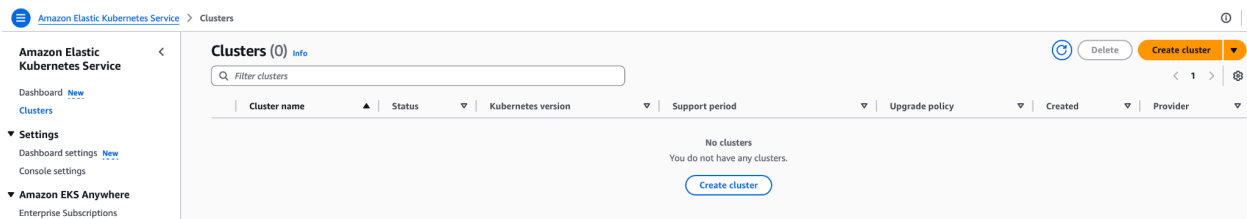
4. You can assign either the Amazon EKS managed policy or create your own custom policy. For the minimum permissions that you must use in your custom policy, see [Amazon EKS cluster IAM role](#). Attach the Amazon EKS managed policy named [AmazonEKSClusterPolicy](#) to the role. To attach an IAM policy to an IAM principal, the principal that is attaching the policy must be assigned one of the following IAM actions (permissions): iam:AttachUserPolicy or iam:AttachRolePolicy.

None

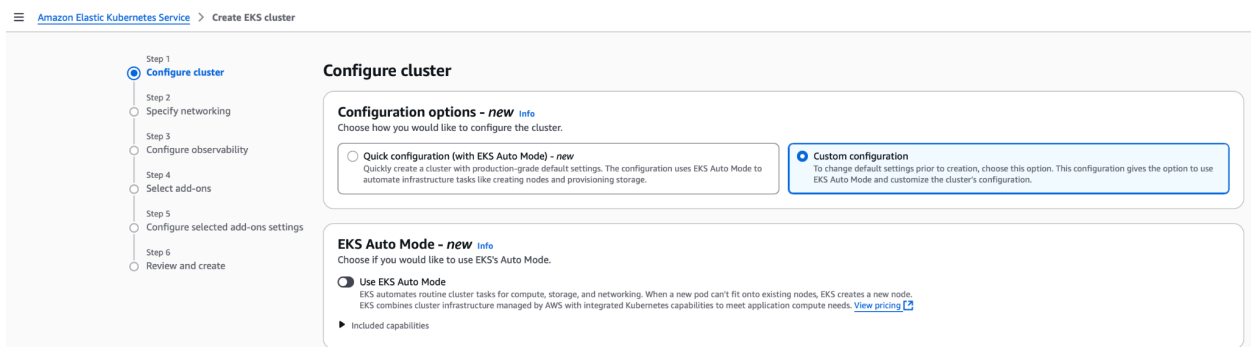
```
aws iam attach-role-policy --policy-arn
arn:aws:iam::aws:policy/AmazonEKSClusterPolicy --role-name
atai-platform-eks-cluster-role
```

Step 2: Create cluster

1. Open EKS Console → Click on **Create cluster**



2. Under Configuration options select Custom configuration
3. Under EKS Auto Mode, toggle Use EKS Auto Mode off.



4. On the Configure cluster page, enter the following fields:
 - a. Name: atai-platform

- b. Cluster IAM role – Choose the Amazon EKS cluster IAM role that you created in the Step 1 to allow the Kubernetes control plane to manage AWS resources on your behalf.
- c. Kubernetes version: 1.33
- d. Support type: Standard support

Cluster configuration [Info](#)

Name
Enter a unique name for this cluster. This property cannot be changed after the cluster is created.

The cluster name should begin with letter or digit and can have any of the following characters: the set of Unicode letters, digits, hyphens and underscores. Maximum length of 100.

Cluster IAM role [Info](#)
Select the Cluster IAM role to allow the Kubernetes control plane to manage AWS resources on your behalf. This cannot be changed after the cluster is created. To create a new custom role, follow the instructions in the [Amazon EKS User Guide](#).
 [Create recommended role](#)

Kubernetes version settings

Kubernetes version [Info](#)
Select Kubernetes version for this cluster.

Upgrade policy [Info](#)
Choose one of the following options. You can switch the setting later while the standard support period is in effect.

☒ **Standard support**
This option supports the Kubernetes version for 14 months after the release date. There is no additional cost. When standard support ends, your cluster will be auto upgraded to the next version.

☐ **Extended support**
This option supports the Kubernetes version for 26 months after the release date. The extended support period has an additional hourly cost that begins after the standard support period ends. When extended support ends, your cluster will be auto upgraded to the next version.

5. Cluster access

- a. Select Allow cluster administrator access
- b. Cluster authentication mode: EKS API and ConfigMap

Auto Mode Compute - new [Info](#)
Configure node management for your EKS cluster. EKS offers four compute options: EKS Auto Mode, EC2 Managed Node Groups, Fargate, and hybrid nodes. Node groups, Fargate profiles, and hybrid nodes are configured after cluster creation. You can also create self-managed nodes.

Compute configuration
If EKS Auto Mode is not managing compute resources, you need to create compute resources once the cluster is ready. We recommend creating a node group after cluster creation.
[View documentation](#)

Cluster access [Info](#)
Control how IAM principals can access this cluster.

Bootstrap cluster administrator access [Info](#)
Choose whether the IAM principal creating the cluster has Kubernetes cluster administrator access.

☒ **Allow cluster administrator access**
Allow cluster administrator access for your IAM principal.

☐ **Disallow cluster administrator access**
Disallow cluster administrator access for your IAM principal.

Cluster authentication mode [Info](#)
Configure which source the cluster will use for authenticated IAM principals.

☐ **EKS API**
The cluster will source authenticated IAM principals only from EKS access entry APIs.

☒ **EKS API and ConfigMap**
The cluster will source authenticated IAM principals from both EKS access entry APIs and the aws-auth ConfigMap.

6. Envelop encryption

- a. By default, AWS implements envelope encryption using an AWS owned key. Alternatively, you can setup your own customer managed key (CMK) and link this key by providing the CMK ARN when configuring your EKS cluster.

Envelope encryption info
Envelope encryption is applied to all Kubernetes API data.

☐ By default, AWS implements envelope encryption using an AWS owned key. Alternatively, you can setup your own customer managed key (CMK) and link this key by providing the CMK ARN when configuring your EKS cluster.

☒ Use your own AWS KMS key
After a cluster is created, you can migrate from using an AWS owned key to a customer managed key (CMK), but not vice versa.

7. Use the default configuration for ARC Zonal Shift (disabled by default).
(Optional) Enable Deletion protection
(Optional) Add tags
Click on **Next**

ARC Zonal shift info
Shift application traffic away from an impaired Availability Zone (AZ) in your EKS cluster. You can change this later.

☐ Enabled
EKS will register your cluster with ARC zonal shift to enable you to use zonal shift to shift application traffic away from an AZ.

☒ Disabled
EKS will not register your cluster with ARC zonal shift.

[Learn more](#)

Deletion protection
Deletion protection must be turned off to be able to delete a cluster. It can be turned on and off after the cluster is created.

☐ Turn on deletion protection
Deletion protection provides additional security against accidental cluster deletion.

Tags (0) info
No tags associated with the resource.

[Add new tag](#)
You can add up to 50 tags.

[Cancel](#)
[Next](#)

8. Networking
 - a. VPC: Select your VPC
 - b. Subnets: Select two private subnets:
 - i. atai-platform-vpc-private-us-west-2a (10.5.0.0/20)
 - ii. atai-platform-vpc-private-us-west-2b (10.5.16.0/20)
 - c. (Optional) Additional security groups: EKS automatically creates a cluster security group on cluster creation to facilitate communication between worker nodes and control plane but you can add others as needed.

After the cluster is created, you must **retrieve the security group ID** assigned by AWS. This ID is required when creating Launch Templates for your Managed Node Groups, to ensure proper networking and access between the control plane and the worker nodes. To learn how to get the security group automatically created by EKS got to Step 4: Get the default cluster security group

- d. Cluster IP address family: IPv4

Step 1
● Configure cluster

Step 2
● **Specify networking**

Step 3
○ Configure observability

Step 4
○ Select add-ons

Step 5
○ Configure selected add-ons settings

Step 6
○ Review and create

Specify networking

Networking [Info](#)
IP address family and service IP address range cannot be changed after cluster creation.

VPC [Info](#)
Select a VPC to use for your EKS cluster resources.

vpc-0a79faee3e664a31d | atai-platform-vpc

Subnets [Info](#)
Choose the subnets in your VPC where the control plane may place elastic network interfaces (ENIs) to facilitate communication with your cluster. To create a new subnet, go to the corresponding page in the [VPC console](#).

Select subnets

subnet-0bd36810cb1c67e50 | atai-platform-vpc-private-us-west-2a
us-west-2a 10.5.0.0/20

subnet-0c43aa4dd2636a7b6 | atai-platform-vpc-private-us-west-2b
us-west-2b 10.5.16.0/20

Additional security groups [Info](#)
EKS automatically creates a cluster security group on cluster creation to facilitate communication between worker nodes and control plane. Optionally, choose additional security groups to apply to the EKS-managed Elastic Network Interfaces that are created in your control plane subnets. To create a new security group, go to the corresponding page in the [VPC console](#).

Select security groups

Choose cluster IP address family [Info](#)
Specify the IP address type for pods and services in your cluster.

☒ IPv4

☐ IPv6

Configure Kubernetes service IP address block [Info](#)

☐ Specify the range from which cluster services will receive IP addresses.

Configure remote networks to enable hybrid nodes [Info](#)
EKS Hybrid Nodes enables you to use on-premises and edge infrastructure as nodes in EKS clusters.

☐ Specify the CIDR blocks for your on-premises environments that you will use for hybrid nodes.

9. Cluster endpoint access
 - a. Public and private
 - b. Click on **Next**

Cluster endpoint access [Info](#)
Configure access to the Kubernetes API server endpoint.

☐ Public
The cluster endpoint is accessible from outside of your VPC. Worker node traffic will leave your VPC to connect to the endpoint.

☒ **Public and private**
The cluster endpoint is accessible from outside of your VPC. Worker node traffic to the endpoint will stay within your VPC.

☐ Private
The cluster endpoint is only accessible through your VPC. Worker node traffic to the endpoint will stay within your VPC.

▼ **Advanced settings**

Add/edit sources to public access endpoint. [Info](#)

CIDR block

0.0.0.0/0

Remove

Add source

You can add up to 39 more items.

Cancel Previous **Next**

10. Configure observability page:
 - a. Control plane logging: Enable all:
 - i. API server
 - ii. Audit

- iii. Authenticator
- iv. Controller manager
- v. Scheduler

b. Prometheus metrics: Optional

The screenshot shows the 'Configure observability' step in the AWS EKS console. On the left, a vertical progress bar indicates the sequence of steps: Step 1 (Configure cluster), Step 2 (Specify networking), Step 3 (Configure observability - currently active), Step 4 (Select add-ons), Step 5 (Configure selected add-ons settings), and Step 6 (Review and create). The main content area is titled 'Configure observability' and contains two sections. The first section, 'About observability', has a sub-section 'Metrics' with two options: 'Send Prometheus metrics to Amazon Managed Service for Prometheus' (unchecked) and 'Send application and infrastructure telemetry to Amazon CloudWatch' (unchecked). The second section, 'Control plane logs', has a sub-section 'Send audit and diagnostic logs from the Amazon EKS control plane to CloudWatch Logs' with five checked options: 'API server', 'Audit', 'Authenticator', 'Controller manager', and 'Scheduler'. At the bottom right, there are three buttons: 'Cancel', 'Previous', and 'Next'.

Step 1
● Configure cluster

Step 2
● Specify networking

Step 3
● **Configure observability**

Step 4
○ Select add-ons

Step 5
○ Configure selected add-ons settings

Step 6
○ Review and create

Configure observability

► **About observability**

Metrics

Prometheus | [Info](#)

☐ Send Prometheus metrics to Amazon Managed Service for Prometheus
Monitor your application and infrastructure metrics with Amazon Managed Service for Prometheus. These metrics include system health and performance data.

CloudWatch | [Info](#)

☐ Send application and infrastructure telemetry to Amazon CloudWatch
Installs the Amazon CloudWatch Observability add-on to send application metrics from CloudWatch APM and infrastructure telemetry from CloudWatch Container Insights.

► **Services and telemetry included**

Control plane logs [Info](#)

Send audit and diagnostic logs from the Amazon EKS control plane to CloudWatch Logs.

☒ **API server**
Logs pertaining to API requests to the cluster.

☒ **Audit**
Logs pertaining to cluster access via the Kubernetes API.

☒ **Authenticator**
Logs pertaining to authentication requests into the cluster.

☒ **Controller manager**
Logs pertaining to state of cluster controllers.

☒ **Scheduler**
Logs pertaining to scheduling decisions.

[Cancel](#) [Previous](#) [Next](#)

11. Select add-ons: Select these add-ons:

- a. Amazon VPC CNI (required)
- b. CoreDNS (required)
- c. kube-proxy (required)

Click Next

Step 1
Configure cluster

Step 2
Specify networking

Step 3
Configure observability

Step 4
Select add-ons

Step 5
Configure selected add-ons settings

Step 6
Review and create

Select add-ons

Review the add-ons from multiple categories, then select add-ons to enhance your cluster.

AWS add-ons (19) [Info](#)



Node monitoring agent [Info](#)
 Enable automatic detection of node health issues.
Category
 observability
Compatible compute
 EC2, Hybrid Nodes

☐



CoreDNS [Info](#)
 Enable service discovery within your cluster.
Category
 networking
Compatible compute
 EC2, Hybrid Nodes, Fargate, EKS Auto Mode

☒



Amazon VPC CNI [Info](#)
 Enable pod networking within your cluster.
Category
 networking
Compatible compute
 EC2

☒



kube-proxy [Info](#)
 Enable service networking within your cluster.
Category
 networking
Compatible compute
 EC2, Hybrid Nodes

☒



SR-IOV Network Metrics Exporter
 Install SR-IOV Network Metrics Exporter to generate Prometheus metrics about SR-IOV (Single-Root I/O Virtualization) network devices in EKS bare metal environments.
Category
 observability
Compatible compute
 EC2

☐



Amazon FSx CSI driver
 Enable Amazon FSx for Lustre within your cluster.
Category
 storage
Compatible compute
 EC2, EKS Auto Mode

☐



Amazon EKS Pod Identity Agent [Info](#)
 Install EKS Pod Identity Agent to use EKS Pod Identity to grant AWS IAM permissions to pods through Kubernetes service accounts.
Category
 security
Compatible compute
 EC2, Hybrid Nodes

☒



Amazon CloudWatch Observability [Info](#)
 Install CloudWatch Agent and enable Container Insights and Application Signals within your cluster.
Category
 observability
Compatible compute
 EC2, Hybrid Nodes, EKS Auto Mode

☐



Mountpoint for Amazon S3 CSI Driver [Info](#)
 Enable Mountpoint for Amazon Simple Storage Service (S3) within your cluster.
Category
 storage
Compatible compute
 EC2, EKS Auto Mode
⚠ Add-on does not support EKS Pod Identity at this time. Please use IAM roles for service accounts (IRSA) with this add-on after the cluster is created.

☐



Amazon SageMaker HyperPod task governance
 Prioritize tasks, allocate compute resources, and maximize utilization.
Category
 policy-management
Compatible compute
 HyperPod

☐

12. Configure selected add-ons settings

- VPC CNI: v1.20.4-eksbuild.2 or Default/Current version
- CoreDNS: v1.12.1-eksbuild.2 or Default/Current version
- kube-proxy: v1.33.3-eksbuild.4 or Default/Current version

13. Review your cluster configuration and click on **Create**

Step 1

Configure cluster

Step 2

Specify networking

Step 3

Configure observability

Step 4

Select add-ons

Step 5

Configure selected add-ons settings

Step 6

Review and create

Review and create

Step 1: Cluster

Cluster configuration

Name

atal-platform

EKS Auto Mode

Disabled

Cluster IAM role

arn:aws:iam::716124474177:role/atal-platform-eks-cluster-role

Authentication mode

EKS API and ConfigMap

Kubernetes version

1.33

Upgrade policy

Standard support

Kubernetes cluster administrator access

Allow cluster administrator access

ARC Zonal shift

ARC Zonal shift

Disabled

Deletion protection

Deletion protection

Disabled

Tags (0)

Tags that you've added. Each tag consists of a key and an optional value.

< 1 >

Key

Value

No tags

This cluster does not have any tags.

Step 2: Networking

Networking

These properties cannot be changed after the cluster is created.

VPC

vpc-0a79faee3e664a31d

Subnets

subnet-0bd36810cb1c67e50

subnet-0c43aa4dd2636a7b6

Cluster IP address family

IPv4

Cluster endpoint access

API server endpoint access

Public and private

Public access source allowlist

0.0.0.0/0

Step 3: Observability

Edit

Control plane logs

API server

on

Authenticator

on

Scheduler

on

Audit

on

Controller manager

on

Step 4: Add-ons

Edit

Selected add-ons (4)

Find add-on

< 1 >

Add-on name	Type	Status
coredns	networking	Ready to install
eks-pod-identity-agent	security	Ready to install
kube-proxy	networking	Ready to install
vpc-cni	networking	Ready to install

Step 5: Versions

Edit

Selected add-ons version (4)

Find add-on versions

< 1 >

Add-on name	Version
coredns	v1.12.1-eksbuild.2
eks-pod-identity-agent	v1.3.9-eksbuild.5
kube-proxy	v1.33.3-eksbuild.4
vpc-cni	v1.20.4-eksbuild.2

EKS Pod Identity (1)

Find add-on versions

< 1 >

Add-on name	IAM role	Service account
vpc-cni	Not set	aws-node

Cancel

Previous

Create

Step 3: Update kubeconfig

1. Enable kubectl to communicate with your cluster by adding a new context to the kubectl config file.

None

```
aws eks update-kubeconfig --region region-code --name atai-platform
```

An example output is as follows.

None

```
Added new context arn:aws:eks:region-code:111122223333:cluster/atai-platform to
/home/username/.kube/config
```

2. Confirm communication with your cluster by running the following command.

None

```
kubectl get svc
```

An example output is as follows.

None

NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE
kubernetes	ClusterIP	10.100.0.1	<none>	443/TCP	28h

Step 4: Get the default cluster security group

After the cluster is created, you must **retrieve the security group ID** assigned by AWS. This ID is required when creating Launch Templates for your Managed Node Groups, to ensure proper networking and access between the control plane and the worker nodes.

You can determine the ID of your cluster security group in the AWS Management Console under the cluster's Networking section. Or, you can do so by running the following AWS CLI command.

None

```
aws eks describe-cluster --name atai-platform --query  
cluster.resourcesVpcConfig.clusterSecurityGroupId
```

EKS managed node group configuration

Prerequisites

1. EKS cluster is ACTIVE
2. IAM role for node groups created (see Step 1 below)

3. Security group for nodes (see Step 2 below)
4. An existing VPC and subnets that meet Amazon EKS requirements
5. One private subnets in your VPC:
 - a. Recommended naming:
 - i. atai-platform-vpc-private-us-west-2a,
 - b. Recommended CIDR blocks: /20 per subnet
 - i. Example: 10.5.0.0/20 (subnet 1)
 - c. Subnets must have routes to NAT Gateway or Internet Gateway for outbound internet access
6. The kubectl command line tool is required. The version can be the same as or up to one minor version earlier or later than the Kubernetes version of your cluster.
7. Version 2.12.3 or later or version 1.27.160 or later of the AWS Command Line Interface (AWS CLI) installed and configured on your device.
8. An IAM principal with permissions to create and describe an Amazon EKS cluster, and create Managed Node Groups.

Step 1: Creating the Amazon EKS node IAM role

1. Run the following command to create an IAM trust policy JSON file.

```
None
cat > node-role-trust-relationship.json << 'EOF'
{
  "Version": "2012-10-17",
```

```

    "Statement": [
      {
        "Effect": "Allow",
        "Action": [
          "sts:AssumeRole"
        ],
        "Principal": {
          "Service": [
            "ec2.amazonaws.com"
          ]
        }
      }
    ]
  }
}
EOF

```

2. Create the IAM role.

None

```

aws iam create-role \
  --role-name atai-platform-eks-node-role \
  --assume-role-policy-document file://"node-role-trust-relationship.json"

```

3. Attach three required IAM managed policies to the IAM role.

None

```

aws iam attach-role-policy \
  --policy-arn arn:aws:iam::aws:policy/AmazonEKSWorkerNodePolicy \
  --role-name atai-platform-eks-node-role

aws iam attach-role-policy \
  --policy-arn arn:aws:iam::aws:policy/AmazonEC2ContainerRegistryPullOnly \
  --role-name atai-platform-eks-node-role

aws iam attach-role-policy \
  --policy-arn arn:aws:iam::aws:policy/AmazonEC2ContainerRegistryReadOnly \
  --role-name atai-platform-eks-node-role

```

4. Attach the following IAM policy to the IAM role:

None

```
aws iam attach-role-policy \  
  --policy-arn arn:aws:iam::aws:policy/AmazonEKS_CNI_Policy \  
  --role-name atai-platform-eks-node-role
```

5. Attach the following IAM policy to the IAM role to allow SSH access to the nodes through AWS Session Manager (SSM) for debugging purposes.

None

```
aws iam attach-role-policy \  
  --policy-arn arn:aws:iam::aws:policy/AmazonSSMManagedInstanceCore \  
  --role-name atai-platform-eks-node-role
```

Step 2: Creating Node Security Group

6. Go to VPC → Security Groups → Create security group

The screenshot displays the AWS Management Console interface for the 'Security Groups' page. The left-hand navigation pane includes sections for 'Virtual private cloud' (with links to VPCs, Subnets, Route tables, Internet gateways, Egress-only Internet gateways, Carrier gateways, DHCP option sets, Elastic IPs, Managed prefix lists, NAT gateways, Peering connections, and Route servers) and 'Security' (with links to Network ACLs and Security groups). The main content area is titled 'Security Groups (2)' and features a search bar and a table of existing security groups. The table has columns for Name, Security group ID, Security group name, VPC ID, and Description. A single entry is shown: a security group named 'default' with ID 'sg-0c91975721df6571f' and VPC ID 'vpc-0a79facc3e664a31d'. Below the table, there is a 'Select a security group' section with a search bar and a list of security groups.

- Name: atai-platform-eks-node-sg (or your name)
- Description: Security group for EKS nodes
- VPC: Select your VPC from section VPC configuration Step 1

☰ VPC > Security Groups > Create security group

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, complete the fields below.

Basic details

Security group name Info

Name cannot be edited after creation.

Description Info

VPC Info

Inbound rules

This security group has no inbound rules.

[Add rule](#)

Outbound rules

Info

Type	Protocol	Port range	Destination	Description - optional	
All traffic	All	All	Custom	0.0.0.0/0	Delete

[Add rule](#)

Rules with destination of 0.0.0.0/0 or ::/0 allow your instances to send traffic to any IPv4 or IPv6 address. We recommend setting security group rules to be more restrictive and to only allow traffic to specific known IP addresses.

Tags - optional

A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs. No tags associated with the resource.

[Add new tag](#)

You can add up to 50 more tags

[CreateSecurityGroup](#) [Cancel](#) [Create security group](#)

10. Click on **Create security group**

Step 2.1 Add self rules to the node security group

A "self rule" in a security group allows traffic from the same security group.

- Select your security group configured in Step 2 and click on **Edit inbound rules**

🔔 Security group (sg-049eb8e8d3019b45f | atai-platform-eks-node-sg) was created successfully

[Details](#)

sg-049eb8e8d3019b45f - atai-platform-eks-node-sg Actions

Details

Security group name atai-platform-eks-node-sg	Security group ID sg-049eb8e8d3019b45f	Description Security group for EKS nodes	VPC ID vpc-0a79faee3e664a31d
Owner 716124474177	Inbound rules count 0 Permission entries	Outbound rules count 1 Permission entry	

[Inbound rules](#) | [Outbound rules](#) | [Sharing - new](#) | [VPC associations - new](#) | [Tags](#)

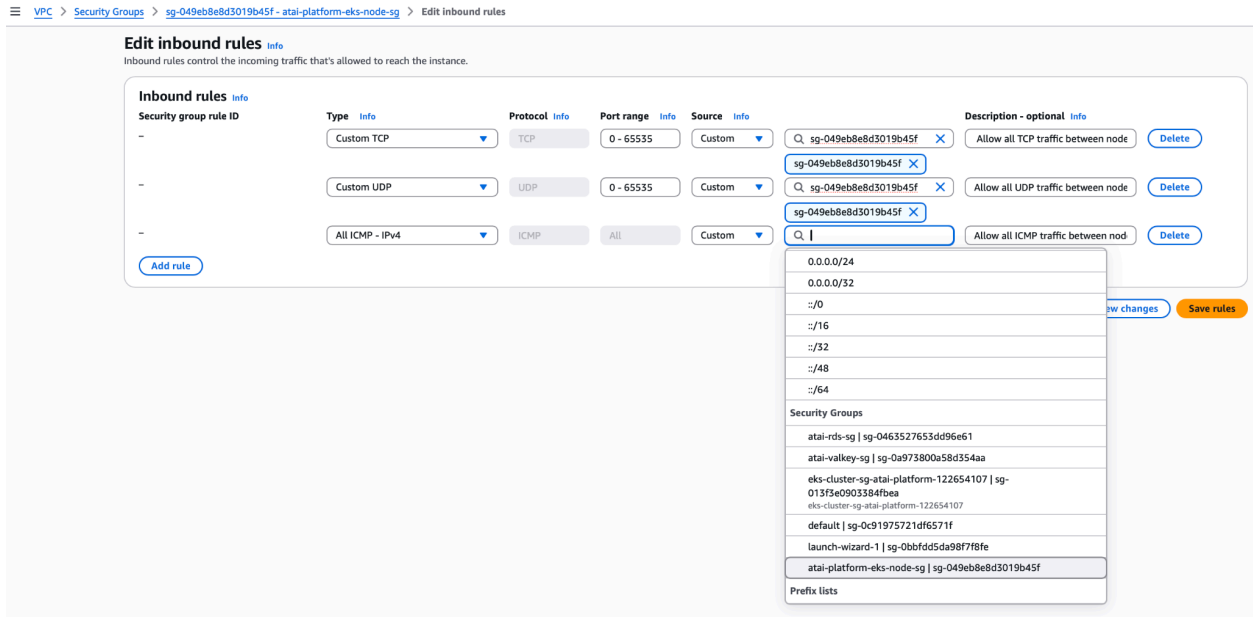
Inbound rules

Name	Security group rule ID	IP version	Type	Protocol	Port range	Source	Description
No security group rules found							

[Manage tags](#) [Edit inbound rules](#)

- Inbound rules: Add rule:

- a. Rule 1: All TCP
 - i. Type: All TCP
 - ii. Source: Custom
 - 1. In the Source field, select Custom (not CIDR blocks or IP addresses).
 - 2. Click on the blank textbox next to Source
 - 3. Navigate to Security groups.
 - 4. Select the same security group you're currently configuring.
 - iii. Description: Allow all TCP traffic between nodes
 - b. Rule 2: All UDP
 - i. Type: All UDP
 - ii. Source: Custom
 - 1. In the Source field, select Custom (not CIDR blocks or IP addresses).
 - 2. Click on the blank textbox next to Source
 - 3. Navigate to Security groups.
 - 4. Select the same security group you're currently configuring.
 - iii. Description: Allow all UDP traffic between nodes
 - c. Rule 3: ICMP
 - i. Type: All ICMP - IPv4
 - ii. Port: None
 - iii. Source: Custom
 - 1. In the Source field, select Custom (not CIDR blocks or IP addresses).
 - 2. Click on the blank textbox next to Source
 - 3. Navigate to Security groups.
 - 4. Select the same security group you're currently configuring.
 - iv. Description: Allow all ICMP traffic between nodes
3. Click on **Save rules**



Note: Creating Self-Referencing Security Group Rules

When adding ingress rules that allow traffic from the same security group (self-rule):

1. In the Source field, select Custom (not CIDR blocks or IP addresses).
2. Open the dropdown menu.
3. Navigate to Security groups.
4. Select the same security group you're currently configuring.

Step 3: Launch templates

Step 3.1 CPU Node Group

1. Go to EC2 → Launch template → **Create Launch Template**

EC2

Dashboard

AWS Global View [↗](#)

Events

▼ Instances

Instances

Instance Types

[Launch Templates](#)

Spot Requests

Savings Plans

Reserved Instances

Dedicated Hosts

Capacity Reservations

Capacity Manager [New](#)

▼ Images

AMIs

AMI Catalog

▼ Elastic Block Store

Volumes

Snapshots

Lifecycle Manager

▼ Network & Security

Security Groups

Elastic IPs

Compute

EC2 launch templates

Streamline, simplify and standardize instance launches

Use launch templates to automate instance launches, simplify permission policies, and enforce best practices across your organization. Save launch parameters in a template that can be used for on-demand launches and with managed services, including EC2 Auto Scaling and EC2 Fleet. Easily update your launch parameters by creating a new launch template version.

New launch template

[Create launch template](#)

Benefits and features

Streamline provisioning

Minimize steps to provision instances. With EC2 Auto Scaling, updates to a launch template can be automatically passed to an Auto Scaling group. [Learn more](#) [↗](#)

Simplify permissions

Create shorter, easier to manage IAM policies. [Learn more](#) [↗](#)

Governance

Ensure best practices are used across your organization. [Learn more](#) [↗](#)

Documentation

[Documentation](#) [↗](#)

[API reference](#) [↗](#)

2. Name: atai-cpu

[EC2](#) > [Launch templates](#) > Create launch template

Create launch template

Creating a launch template allows you to create a saved instance configuration that can be reused, shared and launched at a later time. Templates can have multiple versions.

Launch template name and description

Launch template name - *required*

atai-cpu

Must be unique to this account. Max 128 chars. No spaces or special characters like '&', '*', '@'.

Template version description

A prod webserver for MyApp

Max 255 chars

Auto Scaling guidance | [Info](#)

Select this if you intend to use this template with EC2 Auto Scaling

☐ Provide guidance to help me set up a template that I can use with EC2 Auto Scaling

► Template tags

► Source template

Launch template contents

Specify the details of your launch template below. Leaving a field blank will result in the field not being included in the launch template.

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

An AMI contains the operating system, application server, and applications for your instance. If you don't see a suitable AMI below, use the search field or choose **Browse more AMIs**.

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

Recents

Quick Start

☒ Don't include in launch template

☐ Recently launched

☐ Currently in use



Browse more AMIs

Including AMIs from
AWS, Marketplace and
the Community

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

Advanced

Instance type

Don't include in launch template

☒ 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

Don't include in launch template

[Create new key pair](#)

3. Network settings

- Select the security group created in the Step 2
- Select the default cluster security group that you get from the section **EKS cluster configuration - Step 4: Get the default cluster security group**.

▼ Network settings [Info](#)

Subnet [Info](#)

Don't include in launch template

[Create new subnet](#)

When you specify a subnet, a network interface is automatically added to your template.

Availability Zone [Info](#)

Don't include in launch template

[Enable additional zones](#)

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 reach your instance.

☒ Select existing security group

☐ Create security group

Security groups [Info](#)

Select security groups

atai-platform-eks-node-sg sg-049eb8e8d3019b45f ✕
VPC: vpc-0a79faee3e664a31d

eks-cluster-sg-atai-platform-122654107 sg-013f3e0903384fbea ✕
VPC: vpc-0a79faee3e664a31d

[Compare security group rules](#)

☐ Hide all selected

► Advanced network configuration

4. Storage (Volumes) - Click on **Add new volume**

▼ Storage (volumes) [Info](#)

No volume details are currently included in this template. Add a new volume to include it in the launch template

[Add new volume](#)

a. Volume 1 configuration

- i. Size: 100
- ii. Volume type: gp3
- iii. Device name:
 1. Open the dropdown menu and click on **Specify a custom value**
 2. Value: /dev/xvda

Specify a Device name value



Specifying a custom value allows you to create a template that can be used in other accounts

Device name

[Cancel](#)

[Save](#)

- iv. IOPS: 3000
- v. Throughput: 125

▼ Storage (volumes) [Info](#)

EBS Volumes

[Hide details](#)

▼ Volume 1 (Custom)

[Remove](#)

Storage type | [Info](#)
EBS

Device name - *required* | [Info](#)
/dev/xvda

Snapshot | [Info](#)
Don't include in launch template

Size (GiB) | [Info](#)
100

Volume type | [Info](#)
gp3

IOPS | [Info](#)
3000

Delete on termination | [Info](#)
Yes

Encrypted | [Info](#)
Don't include in launch template

KMS key | [Info](#)
Don't include in launch template

KMS keys are only applicable when encryption is set on this volume.

Throughput | [Info](#)
125

Volume initialization rate - *new, optional* | [Info](#)
Enter a value

Min: 100 MiB/s, Max: 300 MiB/s. Additional charges apply [L](#)

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



[Add new volume](#)

5. Resource tags

- a. Key: Name
- b. Value: atai-cpu
- c. Resource types
 - i. Instances
 - ii. Volumes
 - iii. Network Interfaces

The screenshot shows the 'Resource tags' configuration interface in the AWS Management Console. It features three main input fields: 'Key' with the value 'Name', 'Value' with the value 'atai-cpu', and 'Resource types' with a dropdown menu set to 'Select resource types'. Below the 'Resource types' dropdown, three buttons are visible: 'Instances', 'Volumes', and 'Network interfaces', each with an 'X' icon to remove it. A 'Remove' button is located to the right of the 'Resource types' dropdown. At the bottom left, there is an 'Add new tag' button and a note stating 'You can add up to 49 more tags.' A 'Hide all selected' checkbox is located below the 'Resource types' buttons.

6. Advanced details

- a. Metadata version: V2
- b. Metadata response hop limit: 1

The screenshot shows the 'Advanced details' configuration page for an AWS Launch Template. The page is divided into two main sections. The left section contains several configuration options: 'Metadata accessible' (set to 'Don't include in launch template'), 'Metadata IPv6 endpoint' (set to 'Don't include in launch template'), 'Metadata version' (set to 'V2 only (token required)'), 'Metadata response hop limit' (set to '1'), and 'Allow tags in metadata' (set to 'Don't include in launch template'). A warning message is displayed below these options: 'For V2 requests, you must include a session token in all instance metadata requests. Applications or agents that use V1 for instance metadata access will break.' The right section contains a 'Summary' of the configuration, listing the 'Software Image (AMI)', 'Virtual server type (instance type)', 'Firewall (security group)', and 'Storage (volumes)'. A 'Free tier' notice is also present, stating that in the first year of opening an AWS account, users get 750 hours per month of t2.micro instance usage (or t3.micro where t2.micro isn't available) when used with free tier AMIs, 750 hours per month of public IPv4 address usage, 30 GiB of EBS storage, 2 million I/Os, 1 GB of snapshots, and 100 GB of bandwidth to the internet. Data transfer charges are not included as part of the free tier allowance. At the bottom right, there are 'Cancel' and 'Create launch template' buttons.

7. Click on **Create Launch Template**

Step 3.2 GPU Node Group

8. Go to EC2 → Launch template → **Create Launch Template**

EC2

- Dashboard
- AWS Global View
- Events
- ▼ Instances
 - Instances
 - Instance Types
 - Launch Templates
 - Spot Requests
 - Savings Plans
 - Reserved Instances
 - Dedicated Hosts
 - Capacity Reservations
 - Capacity Manager
- ▼ Images
 - AMIs
 - AMI Catalog
- ▼ Elastic Block Store
 - Volumes
 - Snapshots
 - Lifecycle Manager
- ▼ Network & Security
 - Security Groups
 - Elastic IPs

Compute

EC2 launch templates

Streamline, simplify and standardize instance launches

Use launch templates to automate instance launches, simplify permission policies, and enforce best practices across your organization. Save launch parameters in a template that can be used for on-demand launches and with managed services, including EC2 Auto Scaling and EC2 Fleet. Easily update your launch parameters by creating a new launch template version.

New launch template

Create launch template

Benefits and features

Streamline provisioning

Minimize steps to provision instances. With EC2 Auto Scaling, updates to a launch template can be automatically passed to an Auto Scaling group. [Learn more](#)

Simplify permissions

Create shorter, easier to manage IAM policies. [Learn more](#)

Governance

Ensure best practices are used across your organization. [Learn more](#)

Documentation

[Documentation](#)

[API reference](#)

9. Name: atai-gpu

EC2 > Launch templates > Create launch template

Create launch template

Creating a launch template allows you to create a saved instance configuration that can be reused, shared and launched at a later time. Templates can have multiple versions.

Launch template name and description

Launch template name - *required*

Must be unique to this account. Max 128 chars. No spaces or special characters like '&', '*', '@'.

Template version description

Max 255 chars

Auto Scaling guidance | [Info](#)

Select this if you intend to use this template with EC2 Auto Scaling

☐ Provide guidance to help me set up a template that I can use with EC2 Auto Scaling

► **Template tags**

► **Source template**

Launch template contents

Specify the details of your launch template below. Leaving a field blank will result in the field not being included in the launch template.

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

An AMI contains the operating system, application server, and applications for your instance. If you don't see a suitable AMI below, use the search field or choose [Browse more AMIs](#).

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

[Recents](#)

[Quick Start](#)

☒ Don't include in launch template

☐ Recently launched

☐ Currently in use



[Browse more AMIs](#)

Including AMIs from
AWS, Marketplace and
the Community

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

[Advanced](#)

Instance type

Don't include in launch template

☒ 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

Don't include in launch template

[Create new key pair](#)

10. Network settings

- Select the security group created in the Step 2
- Select the default cluster security group that you get from the section EKS cluster configuration Step 4: Get the default cluster security group.

▼ Network settings [Info](#)

Subnet [Info](#)

Don't include in launch template

[Create new subnet](#)

When you specify a subnet, a network interface is automatically added to your template.

Availability Zone [Info](#)

Don't include in launch template

[Enable additional zones](#)

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 reach your instance.

☒ Select existing security group

☐ Create security group

Security groups [Info](#)

Select security groups

atai-platform-eks-node-sg sg-049eb8e8d3019b45f [×](#)
VPC: vpc-0a79faee3e664a31d

eks-cluster-sg-atai-platform-122654107 sg-013f3e0903384fbea [×](#)
VPC: vpc-0a79faee3e664a31d

[Compare security group rules](#)

[Hide all selected](#)

► Advanced network configuration

11. Storage (Volumes) - Click on **Add new volume**

▼ Storage (volumes) [Info](#)

No volume details are currently included in this template. Add a new volume to include it in the launch template

[Add new volume](#)

a. Volume 1 configuration

- i. Size: 100
- ii. Volume type: gp3
- iii. Device name:
 1. Open the dropdown menu and click on **Specify a custom value**
 2. Value: /dev/xvda

Specify a Device name value ✕

Specifying a custom value allows you to create a template that can be used in other accounts

Device name

[Cancel](#)

[Save](#)

- iv. IOPS: 3000
- v. Throughput: 125

▼ Storage (volumes) [Info](#)

EBS Volumes

[Hide details](#)

▼ Volume 1 (Custom)

[Remove](#)

Storage type [Info](#)

EBS

Device name - *required* [Info](#)

/dev/xvda

Snapshot [Info](#)

Don't include in launch template

Size (GiB) [Info](#)

100

Volume type [Info](#)

gp3

IOPS [Info](#)

3000

Delete on termination [Info](#)

Yes

Encrypted [Info](#)

Don't include in launch template

KMS key [Info](#)

Don't include in launch template

KMS keys are only applicable when encryption is set on this volume.

Throughput [Info](#)

125

Volume initialization rate - *new, optional* [Info](#)

Enter a value

Min: 100 MiB/s, Max: 300 MiB/s. Additional charges apply [L](#)

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

[Add new volume](#)

12. Resource tags

- a. Key: Name
- b. Value: atai-gpu
- c. Resource types
 - i. Instances
 - ii. Volumes
 - iii. Network Interfaces

The screenshot shows the 'Resource tags' configuration page in the AWS console. It features three input fields: 'Key' with the value 'Name', 'Value' with the value 'atai-gpu', and 'Resource types' with a dropdown menu set to 'Select resource types'. Below the dropdown are three buttons: 'Instances', 'Volumes', and 'Network interfaces', each with an 'X' icon to remove it. A 'Remove' button is also present. At the bottom left, there is an 'Add new tag' button and a note stating 'You can add up to 49 more tags.' A 'Hide all selected' checkbox is located below the resource type buttons.

13. Advanced details

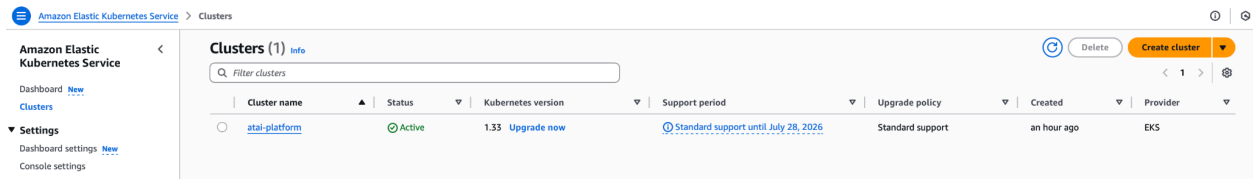
- a. Metadata version: V2
- b. Metadata response hop limit: 1

The screenshot shows the 'Create launch template' page in the AWS console, specifically the 'Advanced details' section. The page is divided into two main panels. The left panel contains several configuration options: 'Metadata accessible' (set to 'Don't include in launch template'), 'Metadata IPv6 endpoint' (set to 'Don't include in launch template'), 'Metadata version' (set to 'V2 only (token required)'), 'Metadata response hop limit' (set to '1'), and 'Allow tags in metadata' (set to 'Don't include in launch template'). A warning message states: 'For V2 requests, you must include a session token in all instance metadata requests. Applications or agents that use V1 for instance metadata access will break.' The right panel shows a 'Summary' section with details about the 'Software Image (AMI)', 'Virtual server type (instance type)', 'Firewall (security group)', and 'Storage (volumes)'. A 'Free tier' notification is displayed, stating: 'In your first year of opening an AWS account, you get 750 hours per month of t2.micro instance usage (or t3.micro where t2.micro isn't available) when used with free tier AMIs, 750 hours per month of public IPv4 address usage, 30 GiB of EBS storage, 2 million I/Os, 1 GB of snapshots, and 100 GB of bandwidth to the internet. Data transfer charges are not included as part of the free tier allowance.' At the bottom right, there are 'Cancel' and 'Create launch template' buttons.

14. Click on **Create Launch Template**

Step 4: Create CPU Node Group

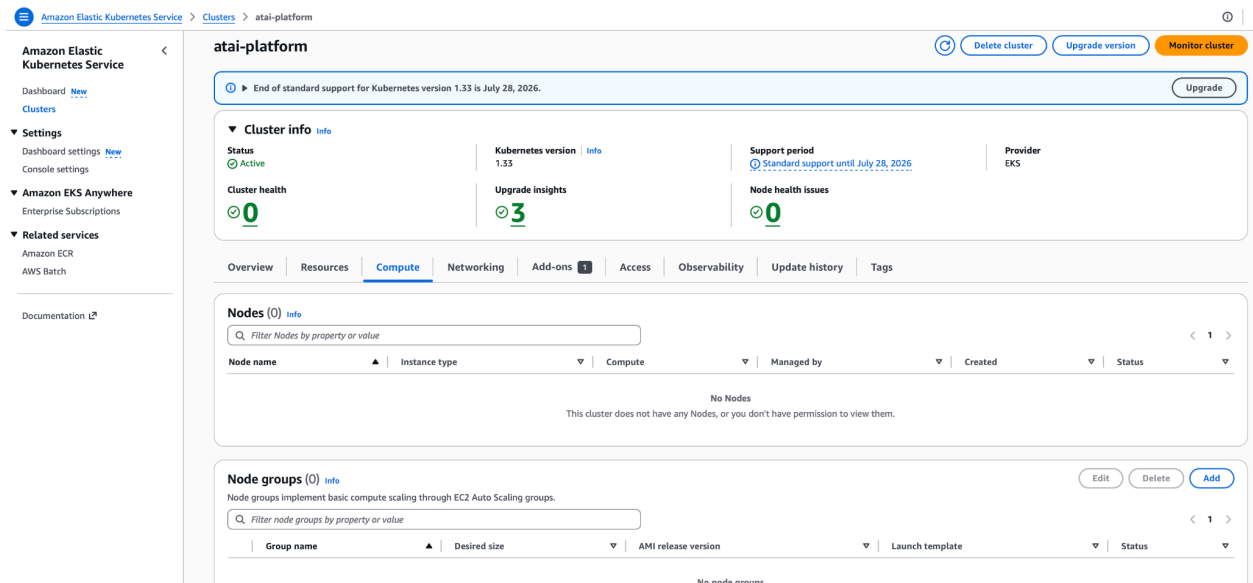
1. EKS Console → Your cluster



The screenshot shows the Amazon Elastic Kubernetes Service (EKS) console. The left sidebar contains the 'Amazon Elastic Kubernetes Service' logo and navigation links for 'Dashboard', 'Clusters', 'Settings', 'Amazon EKS Anywhere', and 'Related services'. The main content area is titled 'Clusters (1)' and displays a table with one cluster named 'atai-platform'. The cluster is in an 'Active' state, running Kubernetes version 1.33, with standard support until July 28, 2026. The table columns include Cluster name, Status, Kubernetes version, Support period, Upgrade policy, Created, and Provider.

Cluster name	Status	Kubernetes version	Support period	Upgrade policy	Created	Provider
atai-platform	Active	1.33	Standard support until July 28, 2026	Standard support	an hour ago	EKS

2. Select Compute tab → Node groups section → Add node group



The screenshot shows the 'atai-platform' cluster details in the EKS console. The left sidebar is the same as the previous screenshot. The main content area is titled 'atai-platform' and includes a warning banner about the end of standard support for Kubernetes version 1.33. Below this is a 'Cluster info' section with details on Status (Active), Kubernetes version (1.33), Support period (Standard support until July 28, 2026), and Provider (EKS). There are also sections for Cluster health, Upgrade insights, and Node health issues. The 'Compute' tab is selected, showing 'Nodes (0)' and 'Node groups (0)'. The 'Nodes' section indicates that the cluster does not have any nodes. The 'Node groups' section shows that there are no node groups and provides an 'Add' button to create a new one.

Cluster info

- Status: Active
- Kubernetes version: 1.33
- Support period: Standard support until July 28, 2026
- Provider: EKS

Cluster health: 0 issues

Upgrade insights: 3 insights

Node health issues: 0 issues

Nodes (0)

No Nodes
This cluster does not have any Nodes, or you don't have permission to view them.

Node groups (0)

No node groups

3. Configure node group:

- a. Node group name: atai-cpu
- b. Node IAM role: Select the role from Step 1
- c. Launch template: Select the Launch template created in the Step 3.1

Amazon Elastic Kubernetes Service > Clusters > atai-platform > Add node group

Step 1

Configure node group

Step 2

Set compute and scaling configuration

Step 3

Specify networking

Step 4

Review and create

Configure node group

A node group is a group of EC2 instances that supply compute capacity to your Amazon EKS cluster. You can add multiple node groups to your cluster.

Node group configuration

These properties cannot be changed after the node group is created.

Name
Assign a unique name for this node group.

The node group name should begin with letter or digit and can have any of the following characters: the set of Unicode letters, digits, hyphens and underscores. Maximum length of 63.

Node IAM role
Select the IAM role that will be used by the nodes. To create a new role, go to the [IAM console](#).
 [Create recommended role](#)

The selected role must not be used by a self-managed node group as this could lead to a service interruption upon managed node group deletion.

[Learn more](#)

Launch template

These properties cannot be changed after the node group is created.

☒ **Use launch template**
Configure this node group using an EC2 launch template.

Launch Template Name
To create a new launch template, go to the corresponding page in the [EC2 console](#).
 [Create launch template](#)

Launch template version
Select the launch template version.
 [Create launch template](#)

4. Configure Kubernetes labels and taints:

- a. Labels: Add label:
 - i. Key: archetypeai.io
 - ii. Value: cpu
- b. Taints: None (leave empty)

Kubernetes labels

Key **Value** [Remove label](#)

[Add label](#)

Remaining labels available to add: 49

Kubernetes taints

This node group does not have any taints.

[Add taint](#)

Remaining taints available to add: 50

Tags

No tags associated with the resource.

[Add new tag](#)

You can add up to 50 tags.

[Cancel](#) [Next](#)

5. Node group compute configuration
 - a. AMI type: Amazon Linux 2023 (AL2023_x86_64_STANDARD)
 - b. Capacity type: On-Demand
 - c. Instance types: m7i.4xlarge
 - d. Disk size: Specified in the Launch template

The screenshot shows the 'Set compute and scaling configuration' step in the AWS IAM console. On the left, a progress bar indicates four steps: Step 1 (Configure node group), Step 2 (Set compute and scaling configuration - currently active), Step 3 (Specify networking), and Step 4 (Review and create). The main content area is titled 'Set compute and scaling configuration' and contains the following sections:

- Node group compute configuration:** A note states 'These properties cannot be changed after the node group is created.'
- AMI type:** A dropdown menu is set to 'Amazon Linux 2023 (x86_64) Standard (AL2023_x86_64_STANDARD)'.
- Capacity type:** A dropdown menu is set to 'On-Demand'.
- Instance types:** A search bar is present with the text 'Enter an instance type'. Below it, a dropdown menu is open, showing 'm7i.4xlarge' as the selected option. Other details for this instance type are visible: 'vCPU: 16 vCPUs', 'Memory: 64 GiB', 'Network: Up to 12.5 Gigabit', 'Max ENI: 8', and 'Max IPs: 240'.
- Disk size:** A note states 'Select the size of the attached EBS volume for each node.' Below it, a dropdown menu is set to 'Specified in launch template'.

6. Node group scaling configuration:
 - a. Desired size: 5
 - b. Minimum size: 5
 - c. Maximum size: 5

The screenshot shows the 'Node group scaling configuration' section in the AWS IAM console. It contains three input fields for node counts, each followed by the word 'nodes':

- Desired size:** The input field contains the value '5'. A note below it states 'Set the desired number of nodes that the group should launch with initially.' and 'Desired node size must be greater than or equal to 0'.
- Minimum size:** The input field contains the value '5'. A note below it states 'Set the minimum number of nodes that the group can scale in to.' and 'Minimum node size must be greater than or equal to 0'.
- Maximum size:** The input field contains the value '5'. A note below it states 'Set the maximum number of nodes that the group can scale out to.' and 'Maximum node size must be greater than or equal to 1 and cannot be lower than the minimum size'.

7. Node group update configuration:
 - a. Maximum unavailable: Percentage
 - b. Value: 33%
 - c. Update strategy: Default

Node group update configuration
Info

Maximum unavailable
Set the maximum number or percentage of unavailable nodes to be tolerated during the node group version update.

☐ Number
Enter a number

☒ Percentage
Specify a percentage

Value
33 %
Percentage must be between 1 to 100.

Update strategy
☒ Default
☐ Minimal

8. Node group auto repair configuration: Disable

Node auto repair configuration
Info

When node auto repair is enabled, Amazon EKS continuously monitors the health of the nodes within a managed node group. This feature automatically detects and replaces nodes when issues occur.

The node auto repair feature reacts to the Ready condition of the kubelet and any node object manual deletions. It can detect more node conditions for repair when the node monitoring agent is also installed. [Go to Add-ons](#)

☐ Enable node auto repair

Cancel Previous Next

9. Node group network configuration

a. Subnets: Select your private subnets:

i. atai-platform-vpc-private-us-west-2a (10.5.0.0/20)

It's important to select the private subnet in the same AZs as your Valkey clusters and RDS PostgreSQL cluster.

Step 1
Configure node group
Step 2
Set compute and scaling configuration
Step 3
Specify networking
Step 4
Review and create

Specify networking

Node group network configuration
These properties cannot be changed after the node group is created.

Subnets
Info
Specify the subnets in your VPC where your nodes will run. To create a new subnet, go to the corresponding page in the [VPC console](#).

Select subnets
Clear selected subnets

subnet-0bd36810cb1c67e50 | atai-platform-vpc-private-us-west-2a
us-west-2a 10.5.0.0/20

EC2 Key Pair
Select an EC2 key pair to allow secure remote access to your nodes. To create a new EC2 key pair, go to the corresponding page in the [EC2 console](#).

Not specified in launch template

Without a key pair you will not be able to directly connect to nodes after they are created.

Cancel Previous Next

10. Review and click on **Create**
Wait for node group to be ACTIVE (5-10 minutes)

Step 1

Configure node group

Step 2

Set compute and scaling configuration

Step 3

Specify networking

Step 4

Review and create

Review and create

Step 1: Node group

Edit

Node group configuration

Name

atai-cpu

Node IAM role

arn:aws:iam::716124474177:role/atai-platform-eks-node-role

Kubernetes labels (1)

< 1 >

Key	Value
archetypeai.io	cpu

Kubernetes taints (0)

Filter by key, value or effect

< 1 >

Key	Value	Effect
No taints This node group does not have any Kubernetes taints.		

Tags (0)

Tags that you've added. Each tag consists of a key and an optional value.

< 1 >

Key	Value
No tags This node group does not have any tags.	

Step 2: Compute and scaling configuration

Edit

Node group compute configuration

Capacity type

On-Demand

Instance types

m6i.4xlarge

Disk size

Specified in launch template

AMI type

Amazon Linux 2023 (x86_64) Standard (AL2023_x86_64_STANDARD)

Node group scaling configuration

Desired size

1 node

Minimum size

1 node

Maximum size

5 nodes

Node group update configuration

Maximum unavailable

33 %

Update strategy

Default

Node auto repair configuration

Node auto repair

Disabled

Step 3: Networking

Edit

Node group network configuration

Subnets

subnet-0bd36810cb1c67e50

Configure remote access to nodes

off

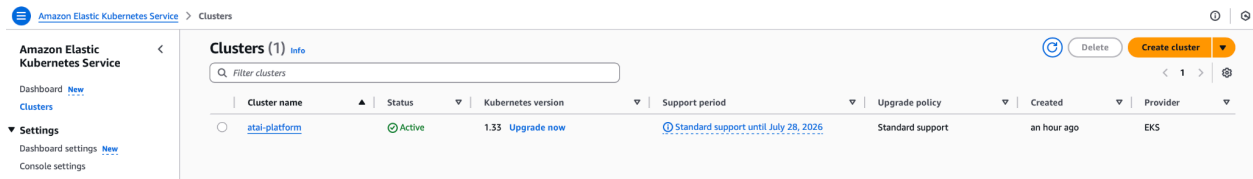
Cancel

Previous

Create

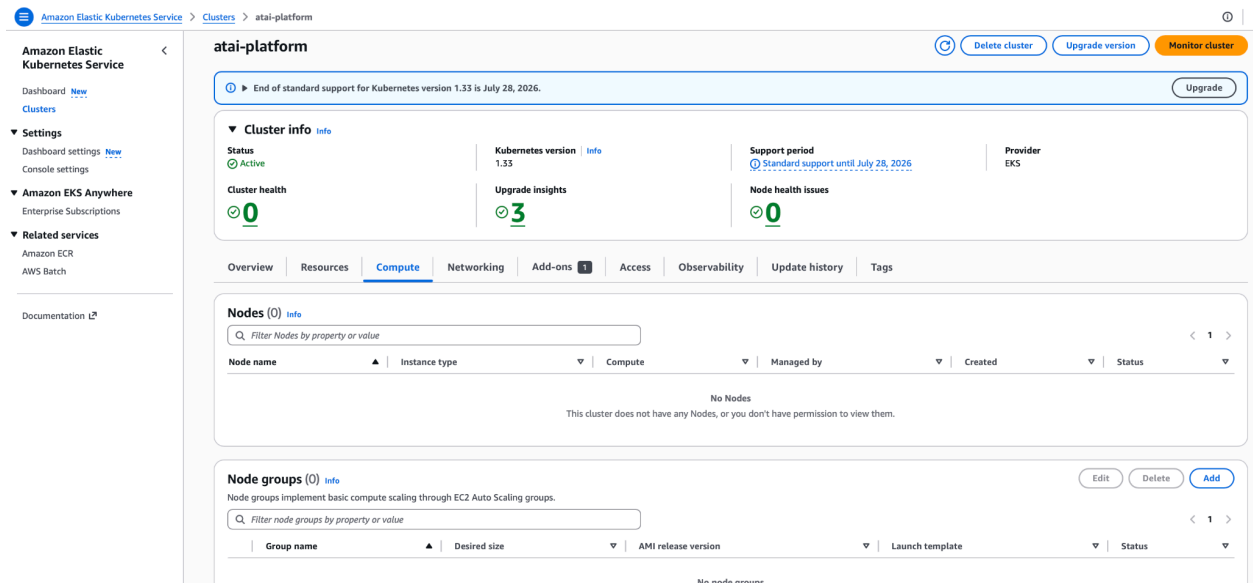
Step 5: Create GPU Node Group

1. EKS Console → Your cluster



The screenshot shows the Amazon Elastic Kubernetes Service (EKS) console. The left sidebar contains the 'Amazon Elastic Kubernetes Service' logo and navigation links for 'Dashboard', 'Clusters', 'Settings', 'Amazon EKS Anywhere', and 'Related services'. The main content area is titled 'Clusters (1)' and displays a table with the following columns: Cluster name, Status, Kubernetes version, Support period, Upgrade policy, Created, and Provider. A single cluster named 'atai-platform' is listed with a status of 'Active', version '1.33', and a support period ending on 'July 28, 2026'.

2. Select Compute tab → Node groups section → Add node group



The screenshot shows the 'atai-platform' cluster details page in the Amazon EKS console. The left sidebar is the same as the previous screenshot. The main content area has a breadcrumb trail: 'atai-platform'. Below the breadcrumb, there are buttons for 'Delete cluster', 'Upgrade version', and 'Monitor cluster'. A warning banner indicates that standard support for Kubernetes version 1.33 ends on July 28, 2026. The 'Cluster info' section shows the cluster is 'Active', version '1.33', and has 'Standard support until July 28, 2026'. The 'Node health' section shows 0 issues, 'Upgrade insights' shows 3, and 'Node health issues' shows 0. The 'Compute' tab is selected, showing 'Nodes (0)' and 'Node groups (0)'. The 'Nodes' section is empty, and the 'Node groups' section is also empty, with a message stating 'No node groups'.

3. Configure node group:

- Node group name: atai-gpu
- Node IAM role: Select the role from Step 1
- Launch template: Select the Launch template created in the Step 3.2

Amazon Elastic Kubernetes Service > Clusters > atai-platform > Add node group

Step 1

Configure node group

Step 2

Set compute and scaling configuration

Step 3

Specify networking

Step 4

Review and create

Configure node group Info

A node group is a group of EC2 instances that supply compute capacity to your Amazon EKS cluster. You can add multiple node groups to your cluster.

Node group configuration

These properties cannot be changed after the node group is created.

Name
Assign a unique name for this node group.

The node group name should begin with letter or digit and can have any of the following characters: the set of Unicode letters, digits, hyphens and underscores. Maximum length of 63.

Node IAM role Info
Select the IAM role that will be used by the nodes. To create a new role, go to the [IAM console](#).

Create recommended role

The selected role must not be used by a self-managed node group as this could lead to a service interruption upon managed node group deletion.

[Learn more](#)

Launch template Info

These properties cannot be changed after the node group is created.

☒ **Use launch template**
Configure this node group using an EC2 launch template.

Launch Template Name
To create a new launch template, go to the corresponding page in the [EC2 console](#).

Create launch template

Launch template version
Select the launch template version.

4. Configure Kubernetes labels and taints:

- Labels: Add label:
 - Key: archetypeai.io
 - Value: gpu
- Taints:
 - Key: nvidia.com/gpu
 - Value: present
 - Effect: NoSchedule

Kubernetes labels Info

Key	Value	
archetypeai.io	gpu	Remove label

Add label

Remaining labels available to add: 49

Kubernetes taints Info

Key	Value	Effect	
nvidia.com/gpu	present	NoSchedule	Remove taint

Add taint

Remaining taints available to add: 49

Tags Info

No tags associated with the resource.

Add new tag

You can add up to 50 tags.

Cancel

Next

5. Node group compute configuration

- AMI type: Amazon Linux 2023 (x86_64) Nvidia (AL2023_x86_64_NVIDIA)
- Capacity type: On-Demand
- Instance types: g6e.2xlarge
- Disk size: Specified in the Launch template

The screenshot shows the 'Set compute and scaling configuration' step in the AWS console. On the left, a progress bar indicates four steps: Step 1 (Configure node group), Step 2 (Set compute and scaling configuration - currently active), Step 3 (Specify networking), and Step 4 (Review and create). The main content area is titled 'Set compute and scaling configuration' and contains the following sections:

- Node group compute configuration:** A note states 'These properties cannot be changed after the node group is created.'
- AMI type:** A dropdown menu shows 'Amazon Linux 2023 (x86_64) Nvidia (AL2023_x86_64_NVIDIA)'.
- Capacity type:** A dropdown menu shows 'On-Demand'.
- Instance types:** A search bar contains 'g6e.2xlarge'. Below it, a tooltip displays details: 'g6e.2xlarge', 'vCPU: 8 vCPUs', 'Memory: 64 GiB', 'Network: Up to 20 Gigabit', 'Max ENI: 4', and 'Max IPs: 60'.
- Disk size:** A dropdown menu shows 'Specified in launch template'.

6. Node group scaling configuration:

- Desired size: 5
- Minimum size: 5
- Maximum size: 10

The screenshot shows the 'Node group scaling configuration' section in the AWS console. It contains three input fields for node counts, each followed by the text 'nodes':

- Desired size:** The input field contains '5'. A note below states: 'Set the desired number of nodes that the group should launch with initially. Desired node size must be greater than or equal to 0'.
- Minimum size:** The input field contains '5'. A note below states: 'Set the minimum number of nodes that the group can scale in to. Minimum node size must be greater than or equal to 0'.
- Maximum size:** The input field contains '10'. A note below states: 'Set the maximum number of nodes that the group can scale out to. Maximum node size must be greater than or equal to 1 and cannot be lower than the minimum size'.

7. Node group update configuration:

- Maximum unavailable: Percentage
- Value: 33%
- Update strategy: Default

Node group update configuration
Info

Maximum unavailable
Set the maximum number or percentage of unavailable nodes to be tolerated during the node group version update.

☐ Number
Enter a number

☒ Percentage
Specify a percentage

Value
33 %
Percentage must be between 1 to 100.

Update strategy
☒ Default
☐ Minimal

8. Node group auto repair configuration: Disable

Node auto repair configuration
Info

When node auto repair is enabled, Amazon EKS continuously monitors the health of the nodes within a managed node group. This feature automatically detects and replaces nodes when issues occur.

The node auto repair feature reacts to the Ready condition of the kubelet and any node object manual deletions. It can detect more node conditions for repair when the node monitoring agent is also installed. [Go to Add-ons](#)

☐ Enable node auto repair

Cancel Previous Next

9. Node group network configuration

a. Subnets: Select your private subnets:

i. atai-platform-vpc-private-us-west-2a (10.5.0.0/20)

It's important to select the private subnet in the same AZs as your Valkey clusters and RDS PostgreSQL cluster.

Step 1
Configure node group
Step 2
Set compute and scaling configuration
Step 3
Specify networking
Step 4
Review and create

Specify networking

Node group network configuration
These properties cannot be changed after the node group is created.

Subnets
Info
Specify the subnets in your VPC where your nodes will run. To create a new subnet, go to the corresponding page in the [VPC console](#).

Select subnets
Clear selected subnets

subnet-0bd36810cb1c67e50 | atai-platform-vpc-private-us-west-2a
us-west-2a 10.5.0.0/20

EC2 Key Pair
Select an EC2 key pair to allow secure remote access to your nodes. To create a new EC2 key pair, go to the corresponding page in the [EC2 console](#).

Not specified in launch template

Without a key pair you will not be able to directly connect to nodes after they are created.

Cancel Previous Next

10. Review and click on **Create**
Wait for node group to be ACTIVE (5-10 minutes)

Step 1

Configure node group

Step 2

Set compute and scaling configuration

Step 3

Specify networking

Step 4

Review and create

Review and create

Step 1: Node group

Node group configuration

Name

atai-gpu

Node IAM role

arn:aws:iam::716124474177:role/atai-platform-eks-node-role

Kubernetes labels (1)

< 1 >

Key	Value
archetypeai.io	gpu

Kubernetes taints (1)

Filter by key, value or effect

< 1 >

Key	Value	Effect
nvidia.com/gpu	present	NoSchedule

Tags (0)

Tags that you've added. Each tag consists of a key and an optional value.

< 1 >

Key	Value
No tags	
This node group does not have any tags.	

Step 2: Compute and scaling configuration

Node group compute configuration

Capacity type

On-Demand

Instance types

g6e.2xlarge

Disk size

Specified in launch template

AMI type

Amazon Linux 2023 (x86_64) Nvidia (AL2023_x86_64_NVIDIA)

Node group scaling configuration

Desired size

4 nodes

Minimum size

4 nodes

Maximum size

20 nodes

Node group update configuration

Maximum unavailable

33 %

Update strategy

Default

Node auto repair configuration

Node auto repair

Disabled

Step 3: Networking

Node group network configuration

Subnets

subnet-0bd36810cb1c67e50

Configure remote access to nodes

off

Cancel

Previous

Create

EKS configuration - Install the NVIDIA Device Plugin

The NVIDIA device plugin DaemonSet to enable GPU resource scheduling in Kubernetes.

Prerequisites

1. EKS cluster is ACTIVE
2. The kubectl command line tool is required. The version can be the same as or up to one minor version earlier or later than the Kubernetes version of your cluster.
3. An IAM principal with permissions to create and describe an Amazon EKS cluster

Step 1: Manual installation

1. Direct download from GitHub:

None

```
curl -L -o nvidia-device-plugin-v0.18.0.yml  
https://raw.githubusercontent.com/NVIDIA/k8s-device-plugin/v0.18.0/deployments/static/nvidia-device-plugin.yml
```

2. Then apply the Kubernetes manifest:

None

```
kubectl apply -f nvidia-device-plugin-v0.18.0.yml
```

3. Verify nodes are annotated with the nvidia label:

None

```
kubectl get nodes -o  
custom-columns="NAME:.metadata.name,GPU:.status.allocatable.nvidia\.com/gpu"
```

S3 configuration

Step 1: Create the platform-data bucket

1. Go to the S3 dashboard and click on **Create Bucket**
2. Make sure, you are located in your home AWS region

Amazon S3

General purpose buckets

Directory buckets

Table buckets

Vector buckets

Access Grants

Access Points (General Purpose)

General purpose buckets (3) Info

Buckets are containers for data stored in S3.

Find buckets by name

Name AWS Region Creation date

3. Bucket name: atai-<UNIQUE ID>-platform-data

Amazon S3 > Buckets > Create bucket

Create bucket Info

Buckets are containers for data stored in S3.

General configuration

AWS Region

US East (N. Virginia) us-east-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.

Bucket name Info

atai-example-platform-data

Bucket names must be 3 to 63 characters and unique within the global namespace. Bucket names must also begin and end with a letter or number. Valid characters are a-z, 0-9, periods (.), and hyphens (-). Learn more

Copy settings from existing bucket - optional

Only the bucket settings in the following configuration are copied.

Choose bucket

Format: s3://bucket/prefix

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.

Object Ownership

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.

Object Ownership

Bucket owner enforced

4. Keep the selection in the Block Public Access setting for the bucket

Block Public Access settings for this bucket

Public access is granted to buckets and objects through access control lists (ACLs), bucket policies, access point policies, or all. In order to ensure that public access to this bucket and its objects is blocked, turn on Block all public access. These settings apply only to this bucket and its access points. AWS recommends that you turn on Block all public access, but before applying any of these settings, ensure that your applications will work correctly without public access. If you require some level of public access to this bucket or objects within, you can customize the individual settings below to suit your specific storage use cases. Learn more

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

Tags - optional (0)

You can use bucket tags to track storage costs and organize buckets. Learn more

No tags associated with this bucket.

Add new tag

You can add up to 50 tags.

5. Use the default Encryption configuration

Default encryption [Info](#)
Server-side encryption is automatically applied to new objects stored in this bucket.

Encryption type [Info](#)
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](#). [L](#)

- ☒ 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)

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](#) [L](#)

- ☐ 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](#)

6. Click on **Create bucket**

⚠ Store your bucket endpoint in a secure location. You will need them later in the *atai-platform* prerequisites.

Step 2: Create the service logs bucket

1. Go to the S3 dashboard and click on **Create Bucket**
2. Make sure, you are located in your home AWS region

Amazon S3

Amazon S3 <

General purpose buckets
Directory buckets
Table buckets
Vector buckets
Access Grants
Access Points (General Purpose)

General purpose buckets [All AWS Regions](#) Directory buckets

General purpose buckets (3) [Info](#)

Buckets are containers for data stored in S3.

[Copy ARN](#) [Empty](#) [Delete](#) [Create bucket](#)

< 1 > [Settings](#)

Name	AWS Region	Creation date
------	------------	---------------

3. Bucket name: atai-<SOME-UNIQUE ID>-service-logs

Amazon S3 > Buckets > Create bucket

Create bucket [Info](#)
Buckets are containers for data stored in S3.

General configuration

AWS Region
US West (Oregon) us-west-2

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.

Bucket name [Info](#)

Bucket names must be 3 to 63 characters and unique within the global namespace. Bucket names must also begin and end with a letter or number. Valid characters are a-z, 0-9, periods (.), and hyphens (-). [Learn more](#) [L](#)

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

[Choose bucket](#)

Format: s3://bucket/prefix

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.

Object Ownership

- ☒ **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.

Object Ownership
Bucket owner enforced

4. Keep the selection in the **Block Public Access setting for the bucket**

5. Use the default Encryption configuration

Default encryption [Info](#)
Server-side encryption is automatically applied to new objects stored in this bucket.

Encryption type [Info](#)
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](#). [L2](#)

- ☒ 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)

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](#) [L2](#)

- ☐ 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](#)

6. Click on **Create bucket**

⚠ Store your bucket endpoint in a secure location. You will need them later in the *atai-platform* prerequisites.

Application Endpoints Configuration

Platform Architecture Overview

The Archetype platform consists of two main entry points that must be publicly accessible:

1. **Console:** The web-based user interface where users interact with the platform
2. **API:** The backend service that handles business logic and data processing

For proper operation, the API must be publicly accessible so the Console can communicate with it, and vice versa. This is a standard configuration for web applications where the frontend (Console) and backend (API) need to exchange data.

Required Public URLs

Once configured, the following URLs must be publicly accessible and secured:

1. **console-archetype.<your-domain>** - Console web application
2. **api-archetype.<your-domain>** - API backend service

Both URLs must be accessible over HTTPS with valid SSL certificates.

How Do I Expose My Services Publicly?

To make your Console and API services publicly accessible, you need to use a Kubernetes Ingress.

What is an Ingress?

An Ingress is a Kubernetes resource that defines routing rules for external traffic. It specifies which domain names should route to which services and on which paths. However, an Ingress resource by itself doesn't handle traffic, you need an Ingress Controller to process these rules and route the actual traffic.

An Ingress Controller acts as a reverse proxy that:

- Receives external HTTP/HTTPS traffic
- Reads Ingress resource rules
- Routes traffic to the appropriate backend services based on domain names and paths

Common Ingress Controllers: Popular Ingress Controller options include:

- AWS Load Balancer Controller - Creates AWS Application Load Balancers (ALB) or Network Load Balancers (NLB)
- NGINX Ingress Controller - A widely-used, feature-rich Ingress Controller
- Traefik - Another popular option

If you don't have an Ingress Controller configured: See Appendix [EKS configuration - Install the NGINX ingress controller](#).

How Do I Secure the Traffic?

Once your services are exposed via Ingress, it's important to secure the traffic with SSL/TLS certificates. This ensures all communications are encrypted and secure.

Common Certificate Solutions: Ingress Controllers typically integrate with certificate management solutions:

- Let's Encrypt through Cert-Manager - Automated, free SSL certificates that are automatically renewed
- AWS Certificate Manager (ACM) - Native AWS-managed certificates

Cert-Manager is a Kubernetes add-on that automatically provisions, renews, and manages SSL certificates. It can integrate with Let's Encrypt to obtain free certificates automatically, or work with AWS ACM for AWS-managed certificates.

If you need help configuring certificates: See Appendix: [EKS configuration - Configure Cert-Manager and Let's Encrypt](#)

How Do I Configure DNS?

After your Ingress Controller is configured, it typically creates an AWS Network Load Balancer (NLB) or Application Load Balancer (ALB) as the entry point for your services.

DNS Configuration Steps:

1. Your Ingress Controller creates an AWS load balancer (NLB or ALB)
2. The load balancer receives a public DNS name or IP address
3. You need to add DNS records that point your domain names to this load balancer

Required DNS Records:

1. Add the following DNS records (A records or CNAME records) pointing to the load balancer created by your Ingress solution:
 - a. **console-archetype.<your-domain>** → Points to the load balancer
 - b. **api-archetype.<your-domain>** → Points to the load balancer

The exact DNS configuration depends on your DNS provider and whether your load balancer provides a DNS name (use CNAME) or an IP address (use A record).

Deployment Checklist

Before Helm Chart Installation:

- ☐ **VPC and subnets** deployed with proper CIDR blocks
- ☐ **8 Valkey instances** created with correct names and versions
- ☐ **1 RDS PostgreSQL** instance created with Aurora engine
- ☐ **EKS cluster** deployed with Kubernetes 1.33
- ☐ **1 CPU node group** deployed with m7i.4xlarge instances
- ☐ **4 GPU node group** deployed with g6e.2xlarge instances and taints
- ☐ **S3 bucket** atai-{customer-prefix}-platform-data and atai-{customer-prefix}-service-logs created
- ☐ **Network connectivity** verified between pods and databases
- ☐ **k8s Ingress** solution installed in the EKS cluster

Support

For questions about infrastructure requirements, contact the Archetype team
(support@archetypeai.dev) for validation of instance types and scaling configurations.

atai-platform

Prerequisites

1. The kubectl command line tool is required. The version can be the same as or up to one minor version earlier or later than the Kubernetes version of your cluster.
2. The eksctl command line tool is required. For more information visit [Installation options for Eksctl](#).
3. Version 2.12.3 or later or version 1.27.160 or later of the AWS Command Line Interface (AWS CLI) installed and configured on your device.
4. An IAM principal with permissions to create and describe an Amazon EKS cluster

Download the deploy kit files

Download and extract the deploy kit, which contains the configuration files and scripts required for the atai-platform configuration:

```
Shell
curl -O
https://archetypeai-marketplace-assets.s3.us-west-2.amazonaws.com/atai-platform-deploy-kit.tar.gz && \
tar -xzf atai-platform-deploy-kit.tar.gz
```

This will create an **atai-platform-deploy-kit/** directory with the following structure:

```
Shell
atai-platform-deploy-kit/
├── scripts/
│   ├── 6.create-irsa.sh
│   └── policies/
│       ├── platform-data-access.json.tpl
│       ├── service-logs-access.json.tpl
│       └── model-depot-access.json
└── templates/
    └── values.yaml
```

Step 1: Kubernetes namespaces

1. Create the namespace to install all the components of the atai-platform

```
Shell
$ kubectl create namespace atai-platform
```

Step 2: Kubernetes Service account for IAM roles (IRSA)

1. Associated an IAM OIDC provider

Shell

```
$ eksctl utils associate-iam-oidc-provider \
  --region <AWS_REGION> \
  --cluster atai-platform \
  --approve
```

2. Run the setup script to create IAM policies, the IAM role, and the Kubernetes service account.

Before running, gather the following values:

Parameter	Description
CUSTOMER_NAME	Short identifier for your organization (max 15 characters). Used as a prefix for IAM resource names.
EKS_CLUSTER_NAME	Name of the EKS cluster created in the previous section.
PLATFORM_DATA_BUCKET_NAME	Name of the S3 bucket created in "S3 configuration". Bucket name only (do not include s3:// or the full ARN).
SERVICE_LOGS_BUCKET_NAME	Name of the S3 bucket created in "S3 configuration". Bucket name only (do not include s3:// or the full ARN).

Set the variables and run the script:

Shell

```
CUSTOMER_NAME="<CUSTOMER_NAME>" \
EKS_CLUSTER_NAME="<EKS_CLUSTER_NAME>" \
PLATFORM_DATA_BUCKET_NAME="<PLATFORM_DATA_BUCKET_NAME>" \
SERVICE_LOGS_BUCKET_NAME="<SERVICE_LOGS_BUCKET_NAME>" \
&& cd atai-platform-deploy-kit/scripts \
&& ./6.create-irsa.sh \
  --region us-west-2 \
  --customer-name "$CUSTOMER_NAME" \
  --cluster-name "$EKS_CLUSTER_NAME" \
  --platform-data-bucket "$PLATFORM_DATA_BUCKET_NAME" \
```

```
--service-logs-bucket "$SERVICE_LOGS_BUCKET_NAME"
```

Upon successful completion, you should see output similar to:

```
Shell
[INFO] Setup completed successfully!
[INFO]
[INFO] Summary:
[INFO]   - Created namespace: atai-platform
[INFO]   - Created 3 IAM policies
[INFO]   - Using S3 buckets:
[INFO]     * Platform Data: atai-<CUSTOMER_NAME>-platform-data
[INFO]     * Service Logs:  atai-<CUSTOMER_NAME>-service-logs
[INFO]   - Created IAM role: atai-platform-role
[INFO]   - Created service account: atai-platform-sa
```

Step 3: Kubernetes secrets required for the atai-platform services

Step 3.1 Generate values for the IAM service secret

Master Key (**IAM_MASTER_KEY**)

Used to authenticate **administrative operations**, such as creating and managing organizations and keys.

Purpose:

- Solves the *bootstrapping problem* — allows you to populate an empty database.

To generate:

```
None
openssl rand -base64 32
```

Server Salt (**IAM_SERVER_SALT**)

Used as the **salt input for Argon2** when hashing API keys.

Key Points:

- Must remain **persistent** for the lifetime of the service.
(Changing it invalidates all existing keys.)
- Leaking it doesn't immediately compromise security **if keys are service-generated** (not manually uploaded).

To generate:

None

```
openssl rand -base64 48
```

After generation stores the secret in a secure place

IAM db name (**IAM_DB_NAME**)

Postgres database that was previously created called iam_db in the section PostgreSQL database configuration Step 4: Extra Database Configuration steps.

IAM db user (**IAM_DB_USER**)

atai_dev postgres user that was created in the PostgreSQL database configuration Step 4: Extra Database Configuration steps.

IAM db password (**IAM_DB_PASSWORD**)

atai_dev postgres user password that was created in the PostgreSQL database configuration Step 4: Extra Database Configuration steps.

IAM db port (**IAM_DB_PORT**)

Port of your PostgreSQL instance

IAM db host (**IAM_DB_HOST**)

Host of your PostgreSQL instance

 Store your user and password in a secure location. You will need them later in the *atai-platform* prerequisites.

Helm chart installation

Prerequisites

1. The kubectl command line tool is required. The version can be the same as or up to one minor version earlier or later than the Kubernetes version of your cluster.
2. Version 2.12.3 or later or version 1.27.160 or later of the AWS Command Line Interface (AWS CLI) installed and configured on your device.
3. Helm 3.20.0 or higher. Learn more about [Helm installation here](#).

Step 1: Installation

1. Retrieve an authentication token and authenticate your clients. Enter the AWS CLI:

```
None
aws ecr get-login-password \
  --region us-west-2 | helm registry login \
  --username AWS \
  --password-stdin 337756366293.dkr.ecr.us-west-2.amazonaws.com
```

2. Create a tmp folder:

```
None
mkdir awsmc-chart && cd awsmc-chart
```

3. Create your **values.yaml** from the template included in the deploy kit:

```
Shell
cp atai-platform-deploy-kit/templates/values.yaml values.yaml
```

Note: Open **values.yaml** and follow the instructions at the top of the file to replace all placeholders with your actual configuration values. Each placeholder includes inline comments with format examples to guide you.

4. Install the **atai-platform** helm chart:

```
Shell
$ helm upgrade atai-platform atai-platform-1.0.3-163-a6c5863.tgz \
```

```
--namespace atai-platform \  
--values values.yaml \  
--install
```

```
Release "atai-platform" does not exist. Installing it now.  
I1111 11:06:28.291728 35840 warnings.go:110] "Warning:  
spec.template.spec.containers[0].env[9]: hides previous definition of  
\"REDIS_USE_INSECURE_TLS\", which may be dropped when using apply"  
NAME: atai-platform  
LAST DEPLOYED: Tue Nov 11 11:06:23 2025  
NAMESPACE: atai-platform  
STATUS: deployed  
REVISION: 1  
TEST SUITE: None
```

Getting Started with the Archetype Platform

To get started with the Archetype Platform, please visit following documentation page:
<https://docs.archetypeai.app/introduction/overview>

Appendix

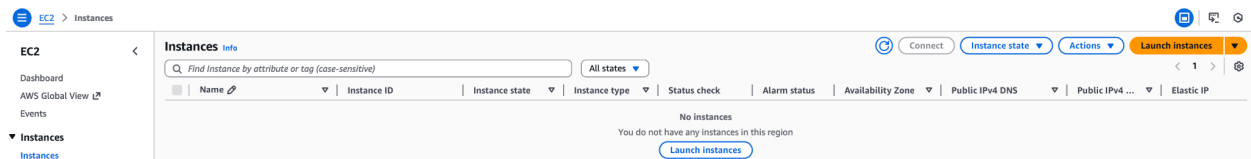
Bastion host configuration

Prerequisites

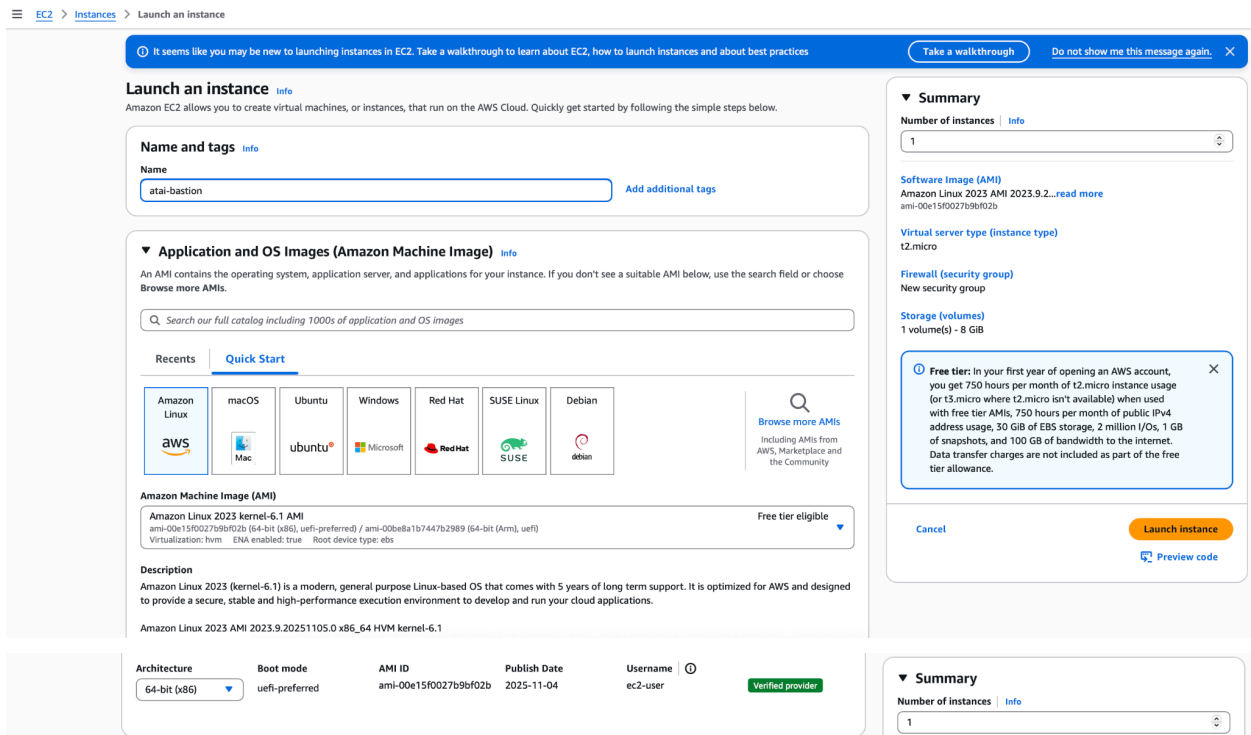
1. VPC with public subnet
2. Internet Gateway attached to VPC
3. Route table configured for public subnet

Step 1: Launch EC2 Instance (Bastion Host)

1. Go to EC2 → Instances → Launch instance



2. Name: atai-bastion
3. Application and OS Images (Amazon Machine Image):
 - a. Search for: Amazon Linux 2023 AMI
 - b. Select: Amazon Linux 2023 AMI (x86_64, HVM, kernel 6.1)



4. Instance type: Select your instance type (e.g., t3.medium for dev, t3.large for production)

Launch an instance

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

Instance type

t3.large
Family: t3 2 vCPU 8 GiB Memory Current generation: true On-Demand RHEL base pricing: 0.112 USD per Hour
On-Demand Ubuntu Pro base pricing: 0.0867 USD per Hour On-Demand Linux base pricing: 0.0832 USD per Hour
On-Demand SUSE base pricing: 0.1395 USD per Hour On-Demand Windows base pricing: 0.1108 USD per Hour

☒ All generations

[Compare instance types](#)

Additional costs apply for AMIs with pre-installed software

5. Select an existing key pair or click on **Create a new key pair**

Launch an instance

▼ 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*

Select

[Create new key pair](#)

6. Assign a new atai-bastion-key, then click on **Create key pair**

Create key pair

Key pair name

Key pairs allow you to connect to your instance securely.

atai-bastion-key

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

Key pair type

☒ RSA
RSA encrypted private and public key pair

☐ ED25519
ED25519 encrypted private and public key pair

Private key file format

☒ .pem
For use with OpenSSH

☐ .ppk
For use with PuTTY

⚠ When prompted, store the private key in a secure and accessible location on your computer. You will need it later to connect to your instance. [Learn more](#)

Cancel

Create key pair

Archetype AI – Confidential 2026 ©

7. Network settings:

- VPF: Select your VPC
- Subnet: Select a public subnet (e.g., atai-platform-vpc-public-us-west-2a)
- Auto-assign public IP: Enable (or use Elastic IP from Step 4)

Launch an instance

▼ Network settings

VPC - required

vpc-0a79faee3e664a31d (atai-platform-vpc)

10.5.0.0/16

Subnet

subnet-04434b04f64fc276b

atai-platform-vpc-public-us-west-2a

VPC: vpc-0a79faee3e664a31d Owner: 716124474177 Availability Zone: us-west-2a (usw2-az2)
Zone type: Availability Zone IP addresses available: 250 CIDR: 10.5.80.0/24

Auto-assign public IP

Enable

Additional charges apply when outside of free tier allowance

- Firewall (security groups): Select **Create security group**
 - By default AWS will add an SSH rule

Launch an instance

Firewall (security groups)

Create security group

Select existing security group

Security group name - required

launch-wizard-1

This security group will be added to all network interfaces. The name can't be edited after the security group is created. Max length is 255 characters. Valid characters: a-z, A-Z, 0-9, spaces, and ._-:/[]@!+=&~:~\$*

Description - required

launch-wizard-1 created 2025-11-07T19:48:21.016Z

Inbound Security Group Rules

▼ Security group rule 1 (TCP, 22, 0.0.0.0/0)

Type

ssh

Protocol

TCP

Port range

22

Source type

Anywhere

Source

0.0.0.0/0

Description - optional

e.g. SSH for admin desktop

Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.

Add security group rule

► Advanced network configuration

Archetype AI – Confidential 2026 ©

8. Configure storage

- Default (8 GiB gp3)
- Recommended 25 GB, but you can adjust based on your requirements

▼ Configure storage [Info](#)

Advanced

1x 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

9. Advanced details (expand):

- Metadata accessible: Enable
- Metadata version: V2 only (token required) (IMDSv2)
- Metadata token response hop limit: 2

▼ Advanced details [Info](#)

Domain join directory [Info](#)

↕ [Create new directory ↗](#)

IAM instance profile [Info](#)

↕ [Create new IAM profile ↗](#)

Metadata accessible [Info](#)

↕

Metadata IPv6 endpoint [Info](#)

↕

Metadata version [Info](#)

↕

ⓘ For V2 requests, you must include a session token in all instance metadata requests. Applications or agents that use V1 for instance metadata access will break.

Metadata response hop limit [Info](#)

↕

Allow tags in metadata [Info](#)

↕

d. User data: Paste the following script:

None

```
#!/bin/bash
```

```
# Update system packages
sudo dnf update -y
```

```
# Install essential packages
sudo dnf install -y \
    htop \
    vim \
    git \
    wget \
    unzip \
    jq \
    postgresql17 \
    redis6
```

```
# Install AWS CLI v2
curl "https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip" -o "awscliv2.zip"
unzip awscliv2.zip
sudo ./aws/install
rm -rf aws awscliv2.zip
```

User data - optional Info

Upload a file with your user data or enter it in the field.

[Choose file](#)

```
sudo dnf install -y \
    htop \
    vim \
    git \
    wget \
    unzip \
    jq \
    postgresql17 \
    redis6

# Install AWS CLI v2
curl "https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip" -o "awscliv2.zip"
unzip awscliv2.zip
sudo ./aws/install
rm -rf aws awscliv2.zip
```

☐ User data has already been base64 encoded

Free tier: In your first year of opening an AWS account, you get 750 hours per month of t2.micro instance usage (or t3.micro where t2.micro isn't available) when used with free tier AMIs, 750 hours per month of public IPv4 address usage, 30 GiB of EBS storage, 2 million I/Os, 1 GB of snapshots, and 100 GB of bandwidth to the internet. Data transfer charges are not included as part of the free tier allowance.

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

10. Click on **Launch instance**.

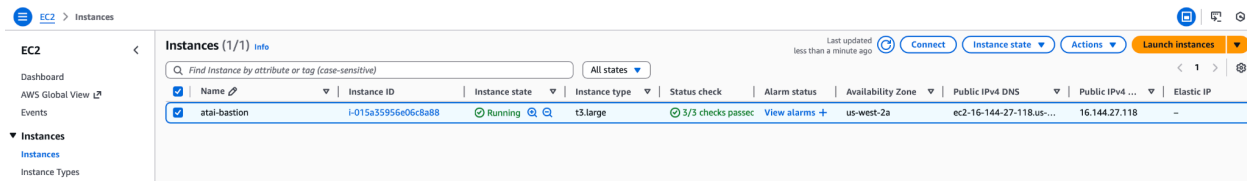
EC2 > Instances > Launch an instance

Success
Successfully initiated launch of instance (i-015a35956e06c8a88)

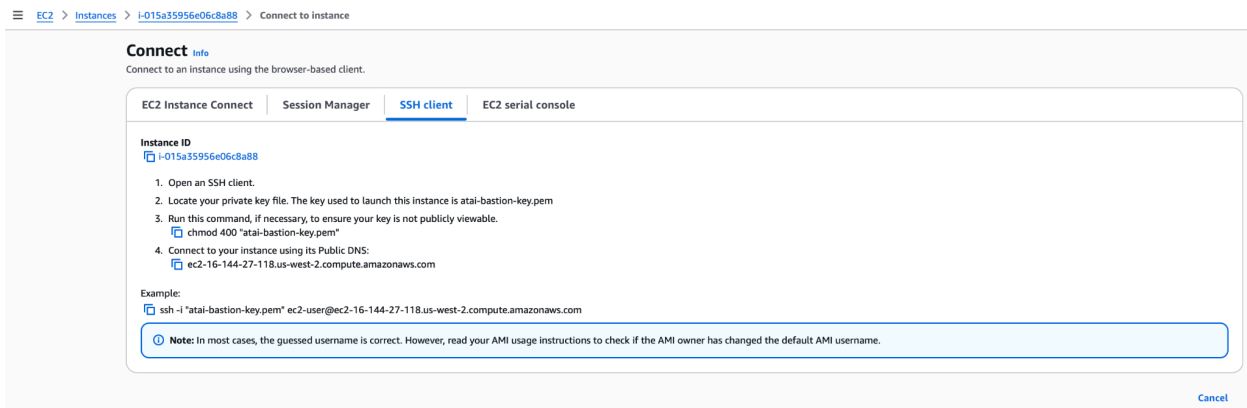
► Launch log

Step 2: Connect to your Bastion host

1. Go to EC2 → Instances → Select your atai-bastion instance and click on **Connect**



2. Click on the tab **SSH client**



- a. Open an SSH client.
- b. Locate your private key file. The key used to launch this instance is atai-bastion-key.pem
- c. Run this command, if necessary, to ensure your key is not publicly viewable.

None

```
chmod 400 "atai-bastion-key.pem"
```

- d. Connect to your instance using its Public DNS e.g.:

None

```
ec2-16-144-27-118.us-west-2.compute.amazonaws.com
```

Example

None

```
ssh -i "atai-bastion-key.pem"
ec2-user@ec2-16-144-27-118.us-west-2.compute.amazonaws.com
```


AWS Service Quotas

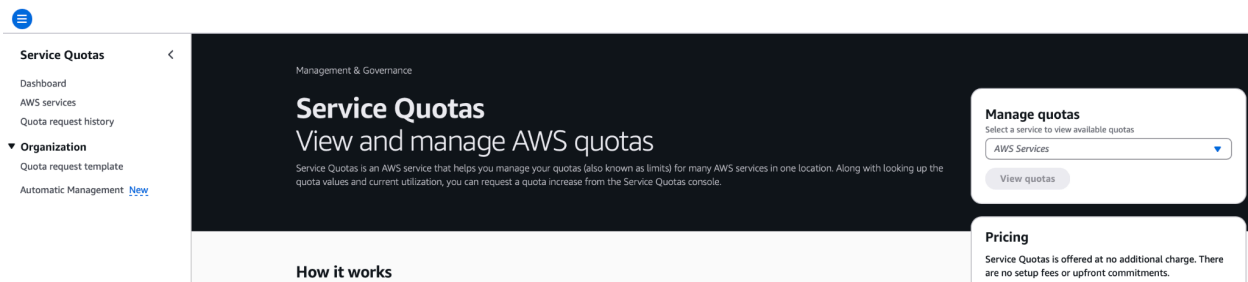
Running On-Demand G and VT instances

Requirements:

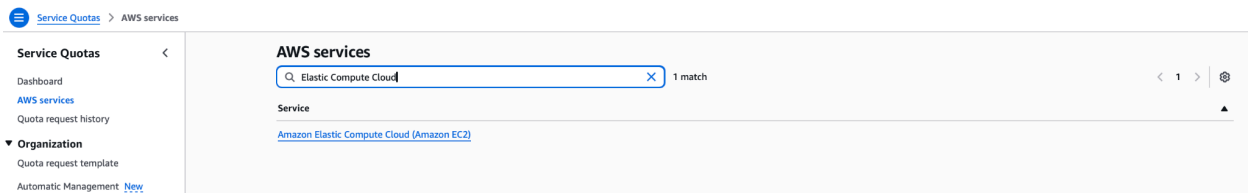
- Instance Type: g6e.2xlarge
- vCPUs per instance: 8 vCPUs
- Minimum instances required: 10
- Total vCPUs needed: $10 \times 8 = 80$ vCPUs

Service Quota to check:

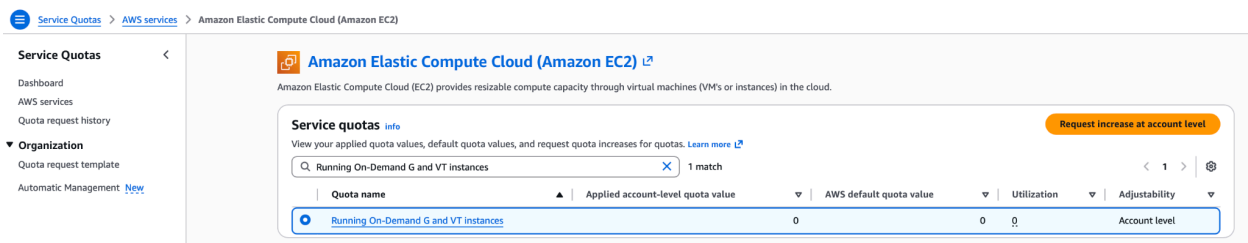
3. Go to AWS Service Quotas → In the left panel click on **AWS Services**



4. In the search bar type AWS Elastic Compute Cloud



5. In the search bar type **Running On-Demand G and VT instances**, select the quota and click on **Request increase at account level**.



6. Set the **Increase account value** to 80 vCPUs and click on **Request**

Request quota increase: Running On-Demand G and VT instances

**Description**

Maximum number of vCPUs assigned to the Running On-Demand G and VT instances.

Requested for

Account (716124474177)

Region

United States (Oregon) us-west-2

Increase quota value

Enter in the total amount that you want the quota to be.

Must be a number greater than your current quota value of 0

Utilization

0

Approvals: For some services, smaller increases are automatically approved, while larger requests are submitted to AWS Support.

Approval timeline: AWS Support can approve, deny, or partially approve your requests. Larger increase requests take more time to process and assess while we work with the service team.

[Cancel](#)

[View quota details](#)

[Request](#)

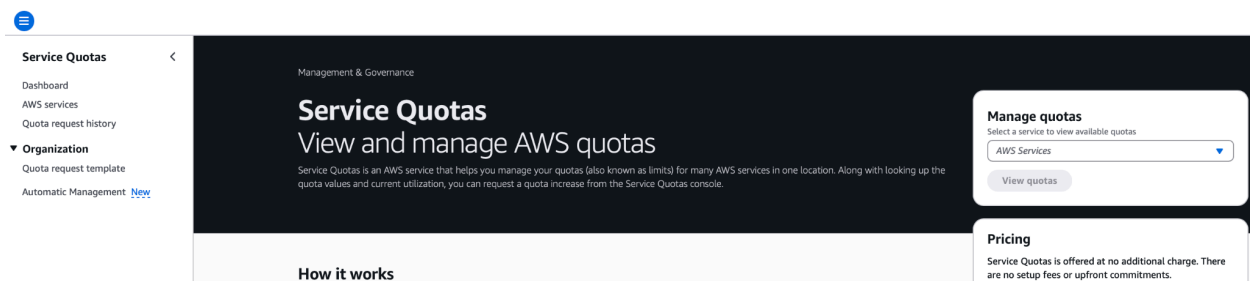
Running On-Demand Standard (A, C, D, H, I, M, R, T, Z) instances

Requirements:

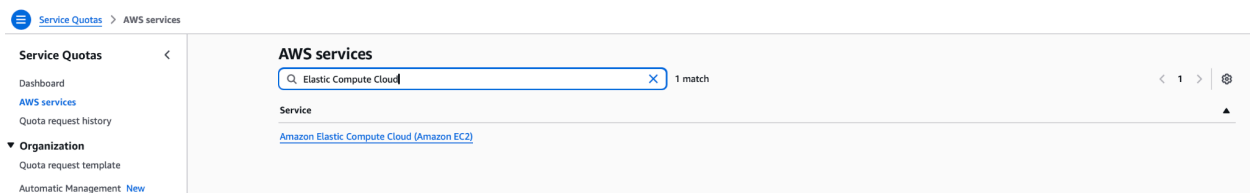
- Instance Type: m6i.4xlarge
- vCPUs per instance: 16 vCPUs
- Minimum instances required: 6
- Total vCPUs needed: $6 \times 16 = 96$ vCPUs

Service Quota to check:

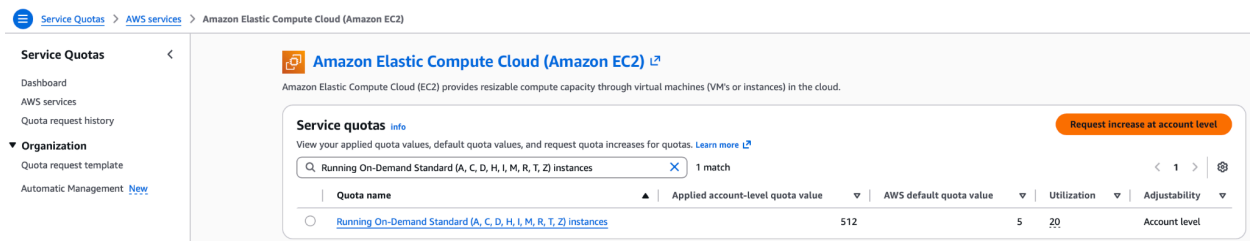
1. Go to AWS Service Quotas → In the left panel click on **AWS Services**



2. In the search bar type AWS Elastic Compute Cloud



3. In the search bar type **Running On-Demand Standard (A, C, D, H, I, M, R, T, Z) instances**, select the quota and click on **Request increase at account level**.



4. Set the **Increase account value** to 96 vCPUs and click on **Request**

Note: If your existing quota for Standard Instances (A, C, D, H, I, M, R, T, Z family) already exceeds 96 vCPUs, no increase is needed for this family.

EKS configuration - Install the NGINX ingress controller

The **atai-platform** Helm chart already bundles the NGINX Ingress Controller as a dependency subchart. There is no need to install a separate Helm chart or add external repositories.

Prerequisites

Before installing NGINX Ingress Controller, ensure you have:

1. EKS Cluster - Your Kubernetes cluster must be running and accessible
2. Public Subnets - You need the subnet IDs where the load balancer will be created
3. VPC ID - The VPC ID where your cluster is running

Enabling the controller

In your values.yaml, locate the **ingress-nginx** section and set enabled to **true**:

```
None
ingress-nginx:
  enabled: true    # change from false to true
```

Then replace the two placeholders in the same section:

Placeholder	Example
<PLACEHOLDER_NLB_SECURITY_GROUP_ID>	sg 0a1b2c3d4e5f6g7h8
<PLACEHOLDER_PUBLIC_SUBNET_IDS>	subnet xxxxxxxxxx, subnet yyyyyyyyyy

The controller will be deployed automatically when you run **helm upgrade --install**.

Note: Already have an NGINX Ingress Controller? Keep **ingress-nginx.enabled: false** and update the **className** field in each ingress block (**api-service-backend**, **console-2-service-frontend**, **iam-service-backend**) to match your existing IngressClass name.

EKS configuration - Configure Cert-Manager and Let's Encrypt

The atai-platform Helm chart already bundles cert-manager as a dependency subchart. There is no need to install a separate Helm chart or add external repositories.

Enabling cert-manager

In your values.yaml, locate the **cert-manager** section and set enabled to **true**:

```
None
cert-manager:
  enabled: true    # change from false to true
```

Cert-manager will be deployed automatically when you run **helm upgrade --install**. No additional placeholders are required.

Create a ClusterIssuer for Let's Encrypt

After the Helm install completes, create a ClusterIssuer to enable automatic SSL certificate provisioning:

```
None
apiVersion: cert-manager.io/v1
kind: ClusterIssuer
metadata:
  name: letsencrypt-prod
spec:
  acme:
    server: https://acme-v02.api.letsencrypt.org/directory
    email: <YOUR_EMAIL>
    privateKeySecretRef:
      name: letsencrypt-prod
    solvers:
      - http01:
          ingress:
            class: atai-nginx
```

Replace **<YOUR_EMAIL>** with a valid email address for certificate expiration notifications.

Apply it:

Shell

```
kubectl apply -f cluster-issuer.yaml
```

MANUAL SETUP INSTRUCTIONS FOR EKS IRSA (IAM ROLES FOR SERVICE ACCOUNTS)

PREREQUISITES

Before starting, gather the following information:

- AWS Region: (e.g., us-west-2)
- Customer Name: Maximum 15 characters (e.g., acme)
- EKS Cluster Name: (e.g., atai-platform)
- Platform Data Bucket Name: (e.g., atai-acme-platform-data)
- Service Logs Bucket Name: (e.g., atai-acme-service-logs)
- AWS Account ID: Your 12-digit AWS account ID

STEP 1: VERIFY PREREQUISITES

1. Confirm the EKS cluster exists and is accessible
2. Verify both S3 buckets exist:
 - Platform Data Bucket
 - Service Logs Bucket
3. Ensure the EKS cluster has an OIDC provider associated (required for IRSA)

STEP 2: ASSOCIATE IAM OIDC PROVIDER (IF NOT ALREADY DONE)

If your cluster doesn't already have an OIDC provider associated:

1. Navigate to EKS Console → Select your cluster
2. Go to the Configuration tab
3. Under OpenID Connect provider URL, note the OIDC provider URL
4. In IAM Console → Identity Providers, verify an identity provider exists for this OIDC URL
5. If not, create one:
 - Provider Type: OpenID Connect
 - Provider URL: (from EKS cluster)
 - Audience: sts.amazonaws.com

STEP 3: CREATE IAM POLICIES

You need to create THREE IAM policies. You'll need the policy JSON documents from the policies/ directory.

POLICY 1: PLATFORM DATA ACCESS

Policy Name: atai-{customer-name}-platform-data-access

Policy Document (substitute {PLATFORM_DATA_BUCKET_ARN}):

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:ListBucket",
        "s3:GetBucketLocation"
      ],
      "Resource": "arn:aws:s3:::{platform-data-bucket-name}"
    },
    {
      "Effect": "Allow",
      "Action": [
        "s3:GetObject",
        "s3:PutObject",
        "s3:DeleteObject"
      ],
      "Resource": "arn:aws:s3:::{platform-data-bucket-name}/*"
    }
  ]
}
```

POLICY 2: SERVICE LOGS ACCESS

Policy Name: atai-{customer-name}-service-logs-access

Policy Document (substitute {SERVICE_LOGS_BUCKET_ARN}):

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:ListBucket",
        "s3:GetBucketLocation"
      ],
      "Resource": "arn:aws:s3:::{service-logs-bucket-name}"
    },
    {
      "Effect": "Allow",
      "Action": [
        "s3:GetObject",
        "s3:PutObject",
        "s3:DeleteObject"
      ],
      "Resource": "arn:aws:s3:::{service-logs-bucket-name}/*"
    }
  ]
}
```

POLICY 3: MODEL DEPOT ACCESS

Policy Name: atai-{customer-name}-model-depot-access

Policy Document (uses fixed bucket: atai-marketplace-model-depot):


```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:ListBucket",
        "s3:GetBucketLocation"
      ],
      "Resource": "arn:aws:s3:::atai-marketplace-model-depot"
    },
    {
      "Effect": "Allow",
      "Action": [
        "s3:GetObject"
      ],
      "Resource": "arn:aws:s3:::atai-marketplace-model-depot/*"
    }
  ]
}
```

NOTE: Record the ARN of each policy after creation. The format will be:
 arn:aws:iam::{account-id}:policy/{policy-name}

STEP 4: CREATE KUBERNETES NAMESPACE

Create a namespace in your EKS cluster:

Namespace Name: atai-platform

Using kubectl:

```
kubectl create namespace atai-platform
```

Or apply via YAML:

```
apiVersion: v1
kind: namespace
metadata:
  name: atai-platform
```

STEP 5: CREATE IAM ROLE FOR SERVICE ACCOUNT

Create an IAM role with the following configuration:

Role Name: atai-platform-role

Trust Relationship (substitute values):

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Federated":
"arn:aws:iam::{account-id}:oidc-provider/oidc.eks.{region}.amazonaws.com/id/{oidc-id}"
      },
      "Action": "sts:AssumeRoleWithWebIdentity",
      "Condition": {
        "StringEquals": {
          "oidc.eks.{region}.amazonaws.com/id/{oidc-id}:sub":
"system:serviceaccount:atai-platform:atai-platform-sa",
          "oidc.eks.{region}.amazonaws.com/id/{oidc-id}:aud":
"sts.amazonaws.com"
        }
      }
    }
  ]
}
```

Where:

- {account-id}: Your AWS account ID
- {region}: Your AWS region
- {oidc-id}: The OIDC provider ID (find in EKS cluster details)

ATTACH POLICIES TO ROLE

Attach the following SIX policies to the IAM role:

1. `arn:aws:iam::{account-id}:policy/atai-{customer-name}-platform-data-access`
2. `arn:aws:iam::{account-id}:policy/atai-{customer-name}-service-logs-access`
3. `arn:aws:iam::{account-id}:policy/atai-{customer-name}-model-depot-access`
4. `arn:aws:iam::aws:policy/AWSMarketplaceMeteringFullAccess` (AWS managed)
5. `arn:aws:iam::aws:policy/AWSMarketplaceMeteringRegisterUsage` (AWS managed)
6. `arn:aws:iam::aws:policy/service-role/AWSLicenseManagerConsumptionPolicy` (AWS managed)

STEP 6: CREATE KUBERNETES SERVICE ACCOUNT

Create a Kubernetes service account with the IAM role annotation:

Service Account Name: `atai-platform-sa`

Namespace: `atai-platform`

Apply the following YAML (substitute `{role-arn}`):

```
apiVersion: v1
kind: ServiceAccount
metadata:
  name: atai-platform-sa
  namespace: atai-platform
  annotations:
    eks.amazonaws.com/role-arn:
arn:aws:iam::{account-id}:role/atai-platform-role
```

STEP 7: VERIFICATION

Verify the setup is complete:

1. Check IAM Policies exist: Navigate to IAM → Policies and confirm all 3 custom policies are created
2. Check IAM Role exists: Navigate to IAM → Roles and verify `atai-platform-role` has 6 policies attached

3. Check Kubernetes resources:

```
kubectl get namespace atai-platform  
kubectl get serviceaccount atai-platform-sa -n atai-platform  
kubectl describe serviceaccount atai-platform-sa -n atai-platform
```

4. Verify the service account has the correct annotation with the IAM role ARN

SUMMARY OF CREATED RESOURCES

- Namespace: atai-platform
- Service Account: atai-platform-sa
- IAM Role: atai-platform-role
- IAM Policies:
 - atai-{customer-name}-platform-data-access
 - atai-{customer-name}-service-logs-access
 - atai-{customer-name}-model-depot-access

IMPORTANT NOTES

- Replace {customer-name} with your actual customer name throughout
- The customer name must be 15 characters or less
- The OIDC provider ID can be found in your EKS cluster's configuration page
- If policies already exist, you can create a new version instead of creating a new policy
- The service account annotation `eks.amazonaws.com/role-arn` is critical for IRSA to work