

Name : Harsh Jadhav
Class : D15B
RollNo : 23
Subject : Adv. DevOps No. 03

The screenshot shows the 'Launch an instance' page in the AWS Management Console. The page is divided into two main sections: 'Name and tags' and 'Application and OS Images (Amazon Machine Image)'. The 'Name and tags' section has a text input field for the instance name, which is currently set to 'Master'. The 'Application and OS Images' section shows a list of AMIs, with 'Amazon Linux 2023 AMI 2023.5.2' selected. The 'Summary' section on the right provides a overview of the configuration: 1 instance, Amazon Linux 2023 AMI 2023.5.2, t2.micro instance type, New security group, and 1 volume of 8 GiB. At the bottom right, there are 'Cancel' and 'Launch instance' buttons.

Launch an instance

Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.

Name and tags

Name: Master

Application and OS Images (Amazon Machine Image)

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

Summary

Number of instances: 1

Software Image (AMI): Amazon Linux 2023 AMI 2023.5.2...read more

Virtual server type (instance type): t2.micro

Firewall (security group): New security group

Storage (volumes): 1 volume(s) - 8 GiB

Cancel Launch instance

The screenshot shows the 'Instances' page in the AWS Management Console. It displays a list of 4 instances. The table columns are: Name, Instance ID, Instance state, Instance type, Status check, Alarm status, and Availability Zone. The instances are: aws-cloud9-ha..., worker 1, worker 2, and Master. All instances are in the 'Running' state.

Instances (4)

Last updated less than a minute ago

Find Instance by attribute or tag (case-sensitive)

Instance state = running

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone
aws-cloud9-ha...	i-03cb63f91dc3417d9	Running	t2.micro	2/2 checks passed	View alarms	us-east-1b
worker 1	i-0c9fe8c234dfde7ad	Running	t2.micro	2/2 checks passed	View alarms	us-east-1a
worker 2	i-096c78b3e09db6c04	Running	t2.micro	2/2 checks passed	View alarms	us-east-1a
Master	i-000b3fe94ba8d2b2e	Running	t2.micro	2/2 checks passed	View alarms	us-east-1d

The screenshot shows the 'Edit inbound rules' page in the AWS Management Console. It displays a table of inbound rules for a security group. The table columns are: Security group rule ID, Type, Protocol, Port range, Source, and Description - optional. There is one rule with ID 'sgr-0f205f80f3604fa4a', Type 'SSH', Protocol 'TCP', Port range '22', and Source '0.0.0.0/0'. There is an 'Add rule' button at the bottom left.

Edit inbound rules

Inbound rules control the incoming traffic that's allowed to reach the instance.

Inbound rules

Security group rule ID	Type	Protocol	Port range	Source	Description - optional
sgr-0f205f80f3604fa4a	SSH	TCP	22	0.0.0.0/0	

Add rule

aws

Services

Search

[Alt+S]

⌘

⏏

?

⚙

N. Virginia

voclabs/user3389085=d2022.harsh.jadhav@ves.ac.in @ 8996-7863...

EC2

>

Instances

>

i-096c78b3e09db6c04

>

Connect to instance

Connect to instance [Info](#)


Connect to your instance i-096c78b3e09db6c04 (worker 2) using any of these options

EC2 Instance Connect

Session Manager


SSH client

EC2 serial console

**Port 22 (SSH) is open to all IPv4 addresses**

Port 22 (SSH) is currently open to all IPv4 addresses, indicated by **0.0.0.0/0** in the inbound rule in [your security group](#). For increased security, consider restricting access to only the EC2 Instance Connect service IP addresses for your Region: 18.206.107.24/29. [Learn more](#).

Instance ID

 i-096c78b3e09db6c04 (worker 2)

Connection Type

☒ **Connect using EC2 Instance Connect**

Connect using the EC2 Instance Connect browser-based client, with a public IPv4 or IPv6 address.

☐ **Connect using EC2 Instance Connect Endpoint**

Connect using the EC2 Instance Connect browser-based client, with a private IPv4 address and a VPC endpoint.

The screenshot shows a terminal session on an Amazon Linux 2023 EC2 instance. The user first checks the metadata expiration date, which is 0:10:27 ago on Tue Sep 24 14:17:05 2024. Then, they install Docker using yum. A detailed output follows, listing the packages to be installed, their dependencies, and the versions. Finally, the user verifies the Docker version.

```
[ec2-user@ip-172-31-3-250 ~]$ sudo su
[root@ip-172-31-3-250 ec2-user]# yum install docker -y
Last metadata expiration check: 0:10:27 ago on Tue Sep 24 14:17:05 2024.
Dependencies resolved.
=====================================================================================================================
 Package                                Architecture          Version                Repository
=====================================================================================================================
Installing:
docker                                 x86_64                 25.0.6-1.amzn2023.0.2  amazonlinux
Installing dependencies:
containerd                            x86_64                 1.7.20-1.amzn2023.0.1  amazonlinux
iptables-libns                         x86_64                 1.8.8-3.amzn2023.0.2   amazonlinux
iptables-nft                          x86_64                 1.8.8-3.amzn2023.0.2   amazonlinux
Installed:
containerd-1.7.20-1.amzn2023.0.1.x86_64    docker-25.0.6-1.amzn2023.0.2.x86_64    iptables-libns-1.8.8-3.amzn2023.0.2.x86_64
iptables-nft-1.8.8-3.amzn2023.0.2.x86_64    libcgrouper-3.0-1.amzn2023.0.1.x86_64    libnetfilter_conntrack-1.0.8-2.amzn2023.0.2.x86_64
libnftnl-1.0.1-19.amzn2023.0.2.x86_64        libnftnl-1.2.2-2.amzn2023.0.2.x86_64     pigz-2.5-1.amzn2023.0.3.x86_64
runc-1.1.13-1.amzn2023.0.1.x86_64
Complete!
[root@ip-172-31-3-250 ec2-user]# docker --version
Docker version 25.0.5, build 5dc9bcc
[root@ip-172-31-3-250 ec2-user]#
```

```
[ec2-user@ip-172-31-34-122 ~]$ kubectl version
kubectl version --client
kubelet --version
kubeadm version: &version.Info{Major:"1", Minor:"26", GitVersion:"v1.26.1", GitCommit:"8f94681cd294aa8cfd3407b8191f6c70214973a4", GitTreeState:"clean", BuildDate:"2023-01-18T15:56:50Z", GoVersion:"go1.19.5", Compiler:"gc", Platform:"linux/amd64"}
WARNING: This version information is deprecated and will be replaced with the output from kubectl version --short. Use --output=yaml|json to get the full version.
Client Version: version.Info{Major:"1", Minor:"26", GitVersion:"v1.26.1", GitCommit:"8f94681cd294aa8cfd3407b8191f6c70214973a4", GitTreeState:"clean", BuildDate:"2023-01-18T15:58:16Z", GoVersion:"go1.19.5", Compiler:"gc", Platform:"linux/amd64"}
Kustomize Version: v4.5.7
Kubernetes v1.26.1
```