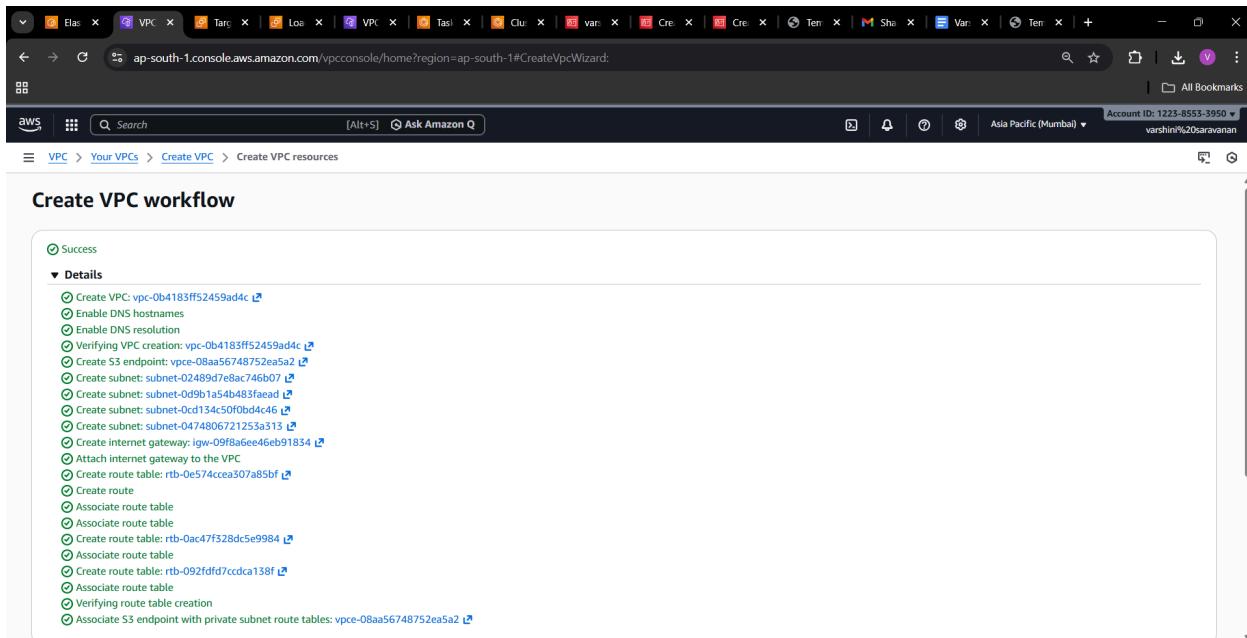


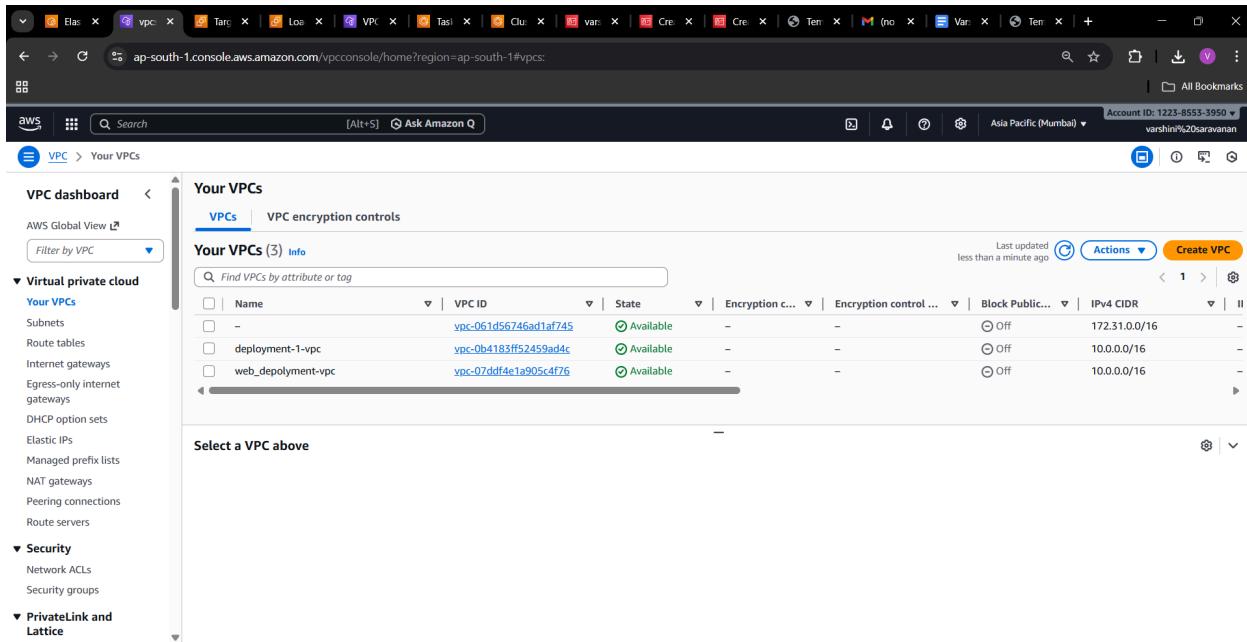
PROJECT 2 REST API

VPC Creation



The screenshot shows the 'Create VPC workflow' page after a successful creation. It lists 27 steps, all of which have been completed successfully, indicated by green checkmarks. The steps include creating the VPC itself, enabling DNS hostnames and resolution, verifying the creation, creating S3 endpoints, subnets, route tables, and internet gateways, and associating route tables.

Name	VPC ID	State	Encryption control	Block Public	IPv4 CIDR
-	vpce-0b4183ff52459ad4c	Available	-	Off	172.31.0.0/16
deployment-1-vpc	vpce-0b4183ff52459ad4c	Available	-	Off	10.0.0.0/16
web_deployment-vpc	vpce-07ddff4e1a905c4f76	Available	-	Off	10.0.0.0/16



The screenshot shows the 'Your VPCs' section of the VPC dashboard. It displays three VPCs: '-', 'deployment-1-vpc', and 'web_deployment-vpc'. Each VPC is listed with its name, VPC ID, state (Available), encryption controls, block public setting, and IPv4 CIDR range. The 'Actions' button is visible for each row.

Launching Instance EC2

The screenshot shows the initial step of launching an EC2 instance. The user has entered 'deployment-1' as the instance name. The summary panel indicates 1 instance is being launched with the Canonical, Ubuntu 24.04 AMI.

Name and tags
Name: deployment-1

Application and OS Images (Amazon Machine Image)
Search: Search our full catalog including 1000s of application and OS images
Recent OS Images: Amazon Linux, macOS, Ubuntu, Windows, Red Hat, SUSE Linux, Debian
Quick Start: Mac OS X, Microsoft, Red Hat, SUSE, Debian

Summary
Number of instances: 1
Software Image (AMI): Canonical, Ubuntu, 24.04, amd64...
Virtual server type (instance type): t3.micro
Firewall (security group): New security group
Storage (volumes): 1 volume(s) - 8 GiB
Launch instance

Additional costs apply for AMIs with pre-installed software

The user has selected 'deployment-1-key' as the key pair. The summary panel remains the same.

Key pair (login)
Key pair name - required: deployment-1-key

Network settings
VPC - required: Info
Subnet: Info
Auto-assign public IP: Info
Firewall (security groups): Info

Summary
Number of instances: 1
Software Image (AMI): Canonical, Ubuntu, 24.04, amd64...
Virtual server type (instance type): t3.micro
Firewall (security group): New security group
Storage (volumes): 1 volume(s) - 8 GiB
Launch instance

© 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Screenshot of the AWS EC2 Instances Launch wizard step 2: Configure Instance Details.

Description - required: deployment-1 created 2025-12-26T07:19:50.346Z

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, Description: e.g. SSH for admin desktop
- Security group rule 2 (TCP, 80, 0.0.0.0/0)
Type: HTTP, Protocol: TCP, Port range: 80
Source type: Anywhere, Description: e.g. SSH for admin desktop
- Security group rule 3 (TCP, 443, 0.0.0.0/0)

Summary

- Number of instances: 1
- Software Image (AMI): Canonical, Ubuntu, 24.04, amd64... (read more)
- Virtual server type (instance type): t3.micro
- Firewall (security group): New security group
- Storage (volumes): 1 volume(s) - 8 GiB

Actions: Cancel, Launch instance, Preview code

Screenshot of the AWS EC2 Instances Launch wizard step 3: Launch instance.

Success: Successfully initiated launch of instance (i-0a778f8488907c266)

Launch log

Next Steps

What would you like to do next with this instance, for example "create alarm" or "create backup"?

Create billing usage alerts: To manage costs and avoid surprise bills, set up email notifications for billing usage thresholds. (Create billing alerts)

Connect to your instance: Once your instance is running, log into it from your local computer. (Connect to instance)

Connect an RDS database: Configure the connection between an EC2 instance and a database to allow traffic flow between them. (Connect an RDS database)

Create EBS snapshot policy: Create a policy that automates the creation, retention, and deletion of EBS snapshots. (Create EBS snapshot policy)

Manage detailed monitoring: Enable or disable detailed monitoring for the instance. If you enable detailed monitoring, the Amazon EC2 Metrics agent will collect detailed monitoring data for your instance. (Learn more)

Create Load Balancer: Create a application, network gateway or classic Elastic Load Balancer.

Create AWS budget: AWS Budgets allows you to create budgets, forecast spend, and take action on your costs and usage from a single place.

Manage CloudWatch alarms: Create or update Amazon CloudWatch alarms for the instance.

Actions: 1, 2, 3, 4, 5, 6, >

The screenshot shows the AWS EC2 Instances page. On the left, there's a navigation sidebar with sections like Dashboard, AWS Global View, Events, Instances (selected), Images, and Elastic Block Store. The main content area displays 'Instances (1/2) Info' for two instances:

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public
web-deploy...	i-02e2492c1cd081703	Running	t3.micro	3/3 checks passed	View alarms +	ap-south-1a	ec2-13-127-104-73.ap...	13.127
deployment-1	i-0a778f8488907c266	Running	t3.micro	3/3 checks passed	View alarms +	ap-south-1a	ec2-43-205-143-206.ap...	43.205

Below the table, the details for instance **i-0a778f8488907c266 (deployment-1)** are shown. The **Status and alarms** tab is selected. It includes sections for Status checks, Metrics, and Alarms, each with a search bar.

CONNECT TO EC2

Screenshot of the AWS EC2 Connect interface showing the connection setup for instance i-0a778f8488907c266.

EC2 Instance Connect

Connection type: Connect using a Public IP (43.205.143.206)

Username: ubuntu

Note: In most cases, the default username, ubuntu, is correct. However, read your AMI usage instructions to check if the AMI owner has changed the default AMI username.

System information as of Fri Dec 26 07:37:21 UTC 2025

```
System load: 0.0 Temperature: -273.1 C
Usage of /: 25.8% of 6.71GB Processes: 109
Memory usage: 22% Users logged in: 0
Swap usage: 0% IPv4 address for ens5: 10.0.2.36

Expanded Security Maintenance for Applications is not enabled.
0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

The list of available updates is more than a week old.
To check for new updates run: sudo apt update

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-10-0-2-36:~$
```

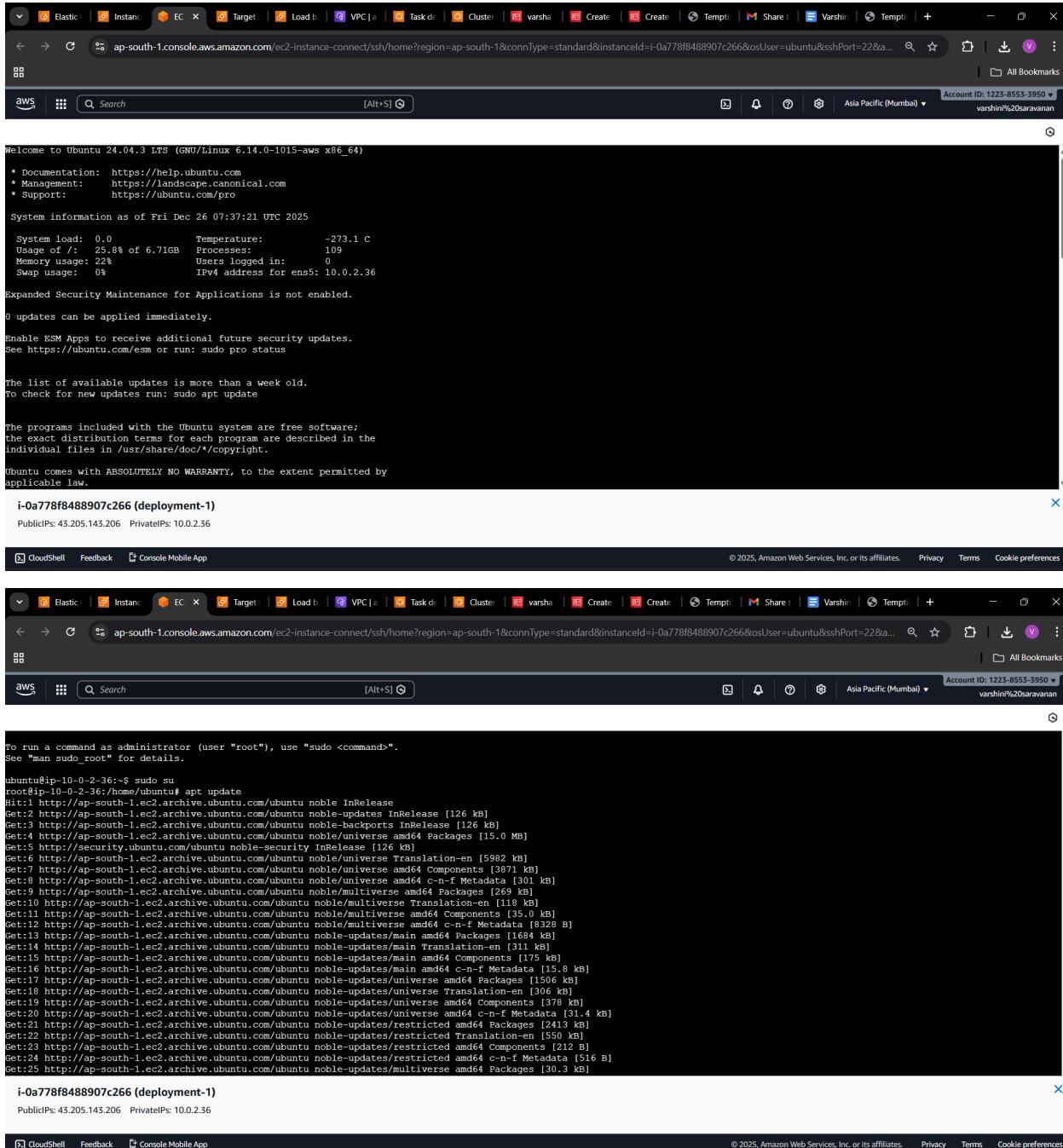
i-0a778f8488907c266 (deployment-1)

PublicIPs: 43.205.143.206 PrivateIPs: 10.0.2.36

CloudShell Feedback Console Mobile App

© 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Update EC2



```
Welcome to Ubuntu 24.04.3 LTS (GNU/Linux 6.14.0-1015-aws x86_64)

 * Documentation: https://help.ubuntu.com
 * Management: https://landscape.canonical.com
 * Support: https://ubuntu.com/pro

System information as of Fri Dec 26 07:37:21 UTC 2025

System load: 0.0 Temperature: -273.1 C
Usage of /: 25.8% of 6.71GB Processes: 109
Memory usage: 22% Users logged in: 0
Swap usage: 0% IPv4 address for ens5: 10.0.2.36

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

The list of available updates is more than a week old.
To check for new updates run: sudo apt update

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

i-0a778f8488907c266 (deployment-1)
PublicIPs: 43.205.143.206 PrivateIPs: 10.0.2.36

CloudShell Feedback Console Mobile App
```

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

```
ubuntu@ip-10-0-2-36:~$ sudo su
root@ip-10-0-2-36:/home/ubuntu# apt update
Hit:1 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Get:2 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Get:3 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease [126 kB]
Get:4 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Packages [15.0 MB]
Get:5 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]
Get:6 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/universe Translation-en [5992 kB]
Get:7 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Components [387 kB]
Get:8 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Metadata [301 kB]
Get:9 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 Packages [265 kB]
Get:10 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse Translation-en [118 kB]
Get:11 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 Components [35.0 kB]
Get:12 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 c-n-f Metadata [8328 B]
Get:13 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 Packages [1684 kB]
Get:14 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main Translation-en [311 kB]
Get:15 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 Components [175 kB]
Get:16 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 c-n-f Metadata [15.8 kB]
Get:17 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe amd64 Packages [1506 kB]
Get:18 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe Translation-en [300 kB]
Get:19 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe amd64 Components [378 kB]
Get:20 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe amd64 c-n-f Metadata [10.4 kB]
Get:21 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates/restricted amd64 Packages [260 kB]
Get:22 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates/restricted Translation-en [550 kB]
Get:23 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates/restricted amd64 Components [212 B]
Get:24 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates/restricted amd64 c-n-f Metadata [516 B]
Get:25 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates/multiverse amd64 Packages [30.3 kB]
```

i-0a778f8488907c266 (deployment-1)
PublicIPs: 43.205.143.206 PrivateIPs: 10.0.2.36

```
Get:44 http://security.ubuntu.com/ubuntu noble-security/main amd64 c-n-f Metadata [9504 B]
Get:45 http://security.ubuntu.com/ubuntu noble-security/universe amd64 Packages [212 B]
Get:46 http://security.ubuntu.com/ubuntu noble-security/universe Translation-en [507 kB]
Get:47 http://security.ubuntu.com/ubuntu noble-security/universe amd64 Components [71.5 kB]
Get:48 http://security.ubuntu.com/ubuntu noble-security/universe amd64 c-n-f Metadata [19.4 kB]
Get:49 http://security.ubuntu.com/ubuntu noble-security/restricted amd64 Packages [2286 kB]
Get:50 http://security.ubuntu.com/ubuntu noble-security/restricted Translation-en [523 kB]
Get:51 http://security.ubuntu.com/ubuntu noble-security/restricted amd64 Components [212 B]
Get:52 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 Packages [274 kB]
Get:53 http://security.ubuntu.com/ubuntu noble-security/multiverse Translation-en [5956 B]
Get:54 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 Components [208 B]
Get:55 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 c-n-f Metadata [384 B]
Fetched 39.2 MB in 6s (6253 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
68 packages can be upgraded. Run 'apt list --upgradable' to see them.
root@ip-10-0-2-36:/home/ubuntu# sudo apt install -y ca-certificates curl gnupg lsb-release
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
ca-certificates is already the newest version (20240203).
ca-certificates set to manually installed.
curl is already the newest version (8.5.0-2ubuntu10.6).
curl set to manually installed.
gnupg is already the newest version (2.4.4-2ubuntu17.3).
gnupg set to manually installed.
lsb-release is already the newest version (12.0-2).
lsb-release set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 68 not upgraded.
root@ip-10-0-2-36:/home/ubuntu#
i-0a77bf8488907c266 (deployment-1)
PublicIPs: 43.205.143.206 PrivateIPs: 10.0.2.36
```

CloudShell Feedback Console Mobile App © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences Account ID: 1233-8553-3950 varshini%20saravanan

Install Docker Engine

```
Get:4 https://download.docker.com/linux/ubuntu noble InRelease [48.5 kB]
Get:5 https://download.docker.com/linux/ubuntu noble/stable amd64 Packages [41.1 kB]
Hit:6 http://security.ubuntu.com/ubuntu noble-security InRelease
Reading package lists... (187 kB)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
68 packages can be upgraded. Run 'apt list --upgradable' to see them.
root@ip-10-0-2-36:/home/ubuntu# apt install -y docker-ce docker-ce-cli containerd.io
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  docker-buildx-plugin docker-ce-rootless-extras docker-compose-plugin libslirp0 pigz slirp4netns
Suggested packages:
  cgroupsfs-mount | cgroup-lite docker-model-plugin
The following NEW packages will be installed:
  containerd.io docker-buildx-plugin docker-ce docker-ce-rootless-extras docker-compose-plugin libslirp0 pigz slirp4netns
0 upgraded, 9 newly installed, 0 to remove and 68 not upgraded.
Need to get 91.3 MB of archives.
After this operation, 364 MB of additional disk space will be used.
Get:1 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 pigz amd64 2.8-1 [65.6 kB]
Get:2 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 libslirp0 amd64 4.7.0-1ubuntu3 [63.8 kB]
Get:3 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 slirp4netns amd64 1.2.1-1build2 [34.9 kB]
Get:4 https://download.docker.com/linux/ubuntu/noble/stable amd64 containerd.io amd64 2.2.1-1ubuntu.24.04-noble [23.4 MB]
Get:5 https://download.docker.com/linux/ubuntu/noble/stable amd64 docker-ce-clm amd64 5:29.1.3-1ubuntu.24.04-noble [16.3 MB]
Get:6 https://download.docker.com/linux/ubuntu/noble/stable amd64 docker-ce amd64 5:29.1.3-1ubuntu.24.04-noble [21.0 MB]
Get:7 https://download.docker.com/linux/ubuntu/noble/stable amd64 docker-buildx-plugin amd64 0.30.1-1ubuntu.24.04-noble [16.4 MB]
Get:8 https://download.docker.com/linux/ubuntu/noble/stable amd64 docker-ce-rootless-extras amd64 5:29.1.3-1ubuntu.24.04-noble [6383 kB]
Get:9 https://download.docker.com/linux/ubuntu/noble/stable amd64 docker-compose-plugin amd64 5.0.0-1ubuntu.24.04-noble [7709 kB]
[##########################################################################] Selecting previous or unselected package containerd.io.
Created symlink /etc/systemd/system/multi-user.target.wants/containerd.service → /usr/lib/systemd/system/containerd.service.
i-0a77bf8488907c266 (deployment-1)
PublicIPs: 43.205.143.206 PrivateIPs: 10.0.2.36
```

Enabling Docker Engine

AWS CloudShell interface showing the configuration of the Docker engine. The terminal window displays the following commands and output:

```
System load: 0.05 Temperature: -273.1 C
Usage of /: 34.7% of 6.71GB Processes: 120
Memory usage: 30% Users logged in: 1
Swap usage: 0% IPv4 address for ens5: 10.0.2.36

Expanded Security Maintenance for Applications is not enabled.

74 updates can be applied immediately.
28 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

Last login: Fri Dec 26 07:37:21 2025 from 13.233.177.3
ubuntu@ip-10-0-2-36:~$ sudo apt update
root@ip-10-0-2-36:/home/ubuntu# apt update
Hit:1 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Hit:2 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease
Hit:3 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease
Hit:4 https://download.docker.com/linux/ubuntu noble InRelease
Hit:5 https://security.ubuntu.com/ubuntu noble-security InRelease
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
68 packages can be upgraded. Run 'apt list --upgradable' to see them.
root@ip-10-0-2-36:/home/ubuntu# systemctl start docker
root@ip-10-0-2-36:/home/ubuntu# systemctl enable docker
Synchronizing state of docker.service with SysV service script with /usr/lib/systemd/systemd-sysv-install.
Executing: /usr/lib/systemd/systemd-sysv-install enable docker
root@ip-10-0-2-36:/home/ubuntu#
```

i-0a778f8488907c266 (deployment-1)

PublicIPs: 43.205.143.206 PrivateIPs: 10.0.2.36

Checking Status of Engine

AWS CloudShell interface showing the status of the Docker engine. The terminal window displays the following command and output:

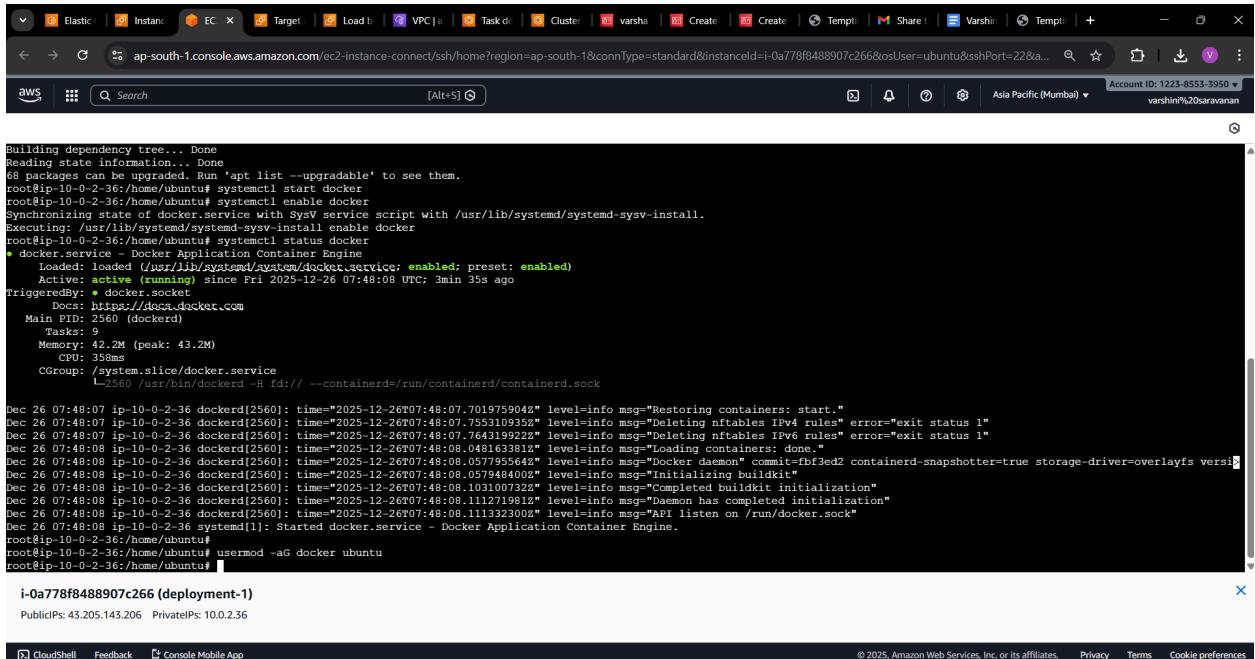
```
root@ip-10-0-2-36:/home/ubuntu# systemctl status docker
● docker.service - Docker Application Container Engine
   Loaded: loaded (/usr/lib/systemd/system/docker.service; enabled; preset: enabled)
   Active: active (running) since Fri 2025-12-26 07:48:08 UTC; 3min 35s ago
     Docs: https://docs.docker.com
 Main PID: 2560 (dockerd)
   Tasks: 2
    Memory: 42.2M (pssK: 43.2M)
      CPU: 356ms
     CGroup: /system.slice/docker.service
             └─2560 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock

Dec 26 07:48:07 ip-10-0-2-36 dockerd[2560]: time="2025-12-26T07:48:07.701975904Z" level=info msg="Restoring containers: start."
Dec 26 07:48:07 ip-10-0-2-36 dockerd[2560]: time="2025-12-26T07:48:07.755310935Z" level=info msg="Deleting nftables IPv4 rules" error="exit status 1"
Dec 26 07:48:07 ip-10-0-2-36 dockerd[2560]: time="2025-12-26T07:48:07.764319922Z" level=info msg="Deleting nftables IPv6 rules" error="exit status 1"
Dec 26 07:48:08 ip-10-0-2-36 dockerd[2560]: time="2025-12-26T07:48:08.048163381Z" level=info msg="Loading containers: done."
Dec 26 07:48:08 ip-10-0-2-36 dockerd[2560]: time="2025-12-26T07:48:08.057795564Z" level=info msg="Docker daemon" commit=fbff3ed2 containerd-snapshotter=true storage-driver=overlayfs version=v2025.12.0+git.20251207.48:08.057795564
Dec 26 07:48:08 ip-10-0-2-36 dockerd[2560]: time="2025-12-26T07:48:08.084006Z" level=info msg="Initializing buildkit"
Dec 26 07:48:08 ip-10-0-2-36 dockerd[2560]: time="2025-12-26T07:48:08.103100Z" level=info msg="Completed buildkit initialization"
Dec 26 07:48:08 ip-10-0-2-36 dockerd[2560]: time="2025-12-26T07:48:08.111271981Z" level=info msg="Docker daemon has completed initialization"
Dec 26 07:48:08 ip-10-0-2-36 dockerd[2560]: time="2025-12-26T07:48:08.111332300Z" level=info msg="API listen on /run/docker.sock"
Dec 26 07:48:08 system[1]: Started docker.service - Docker Application Container Engine.
```

lines 1-22/22 (END)

i-0a778f8488907c266 (deployment-1)

PublicIPs: 43.205.143.206 PrivateIPs: 10.0.2.36

AWS CloudShell interface showing terminal output for Docker setup.

```
Building dependency tree... Done
Reading state information... Done
68 packages upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
Run 'apt list --upgradable' to see them.
root@ip-10-0-2-36:/home/ubuntu# systemctl start docker
root@ip-10-0-2-36:/home/ubuntu# systemctl enable docker
Synchronizing state of docker.service with SysV service script with /usr/lib/systemd/systemd-sysv-install.
Executing: /usr/lib/systemd/systemd-sysv-install enable docker
root@ip-10-0-2-36:/home/ubuntu# systemctl status docker
● docker.service - Docker Application Container Engine
   Loaded: loaded (/usr/lib/systemd/system/docker.service; enabled; preset: enabled)
     Active: active (running) since Fri 2025-12-26 07:48:08 UTC; 3min 35s ago
TriggeredBy: • docker.socket
   Docs: https://docs.docker.com
      Main PID: 2560 (dockerd)
        Tasks: 1
       Memory: 42.2M (peak: 43.2M)
          CPU: 358ms
         CGroup: /system.slice/docker.service
                  └─2560 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock

Dec 26 07:48:07 ip-10-0-2-36 dockerd[2560]: time="2025-12-26T07:48:07.701975904Z" level=info msg="Restoring containers: start."
Dec 26 07:48:07 ip-10-0-2-36 dockerd[2560]: time="2025-12-26T07:48:07.755310935Z" level=info msg="Deleting nftables IPv4 rules" error="exit status 1"
Dec 26 07:48:07 ip-10-0-2-36 dockerd[2560]: time="2025-12-26T07:48:07.764319922Z" level=info msg="Deleting nftables IPv6 rules" error="exit status 1"
Dec 26 07:48:08 ip-10-0-2-36 dockerd[2560]: time="2025-12-26T07:48:08.046163381Z" level=info msg="Loading containers: done."
Dec 26 07:48:08 ip-10-0-2-36 dockerd[2560]: time="2025-12-26T07:48:08.057955642Z" level=info msg="Docker daemon" commit=fbffed2 containerd-snapshotter=true storage-driver=overlayfs version=v2025.12.0+0.20251226-1
Dec 26 07:48:08 ip-10-0-2-36 dockerd[2560]: time="2025-12-26T07:48:08.057948400Z" level=info msg="Initializing buildkit"
Dec 26 07:48:08 ip-10-0-2-36 dockerd[2560]: time="2025-12-26T07:48:08.103100732Z" level=info msg="Completed buildkit initialization"
Dec 26 07:48:08 ip-10-0-2-36 dockerd[2560]: time="2025-12-26T07:48:08.111271981Z" level=info msg="Daemon has completed initialization"
Dec 26 07:48:08 ip-10-0-2-36 dockerd[2560]: time="2025-12-26T07:48:08.111332300Z" level=info msg="API listen on /run/docker.sock"
Dec 26 07:48:08 ip-10-0-2-36 [systemd]: Started docker.service - Docker Application Container Engine.
root@ip-10-0-2-36:/home/ubuntu#
root@ip-10-0-2-36:/home/ubuntu# usermod -aG docker ubuntu
root@ip-10-0-2-36:/home/ubuntu#
```

i-0a778f8488907c266 (deployment-1)

PublicIPs: 43.205.143.206 PrivateIPs: 10.0.2.36

© 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Disconnect and Reconnect

The screenshot shows the AWS EC2 Instance Connect interface. At the top, it displays the instance ID: i-0a778f8488907c266. Below this, there are tabs for 'EC2 Instance Connect', 'Session Manager', 'SSH client', and 'EC2 serial console'. Under 'Instance ID', the instance ID is listed. In the 'Connection type' section, 'Connect using a Public IP' is selected. The 'Username' field contains 'ubuntu'. A note at the bottom states: 'Note: In most cases, the default username, ubuntu, is correct. However, read your AMI usage instructions to check if the AMI owner has changed the default AMI username.' At the bottom right are 'Cancel' and 'Connect' buttons.

The screenshot shows a terminal session in the AWS CloudShell. The terminal output is as follows:

```
Welcome to Ubuntu 24.04.3 LTS (GNU/Linux 6.14.0-1015-aws x86_64)

 * Documentation: https://help.ubuntu.com
 * Management: https://landscape.canonical.com
 * Support: https://ubuntu.com/pro

System information as of Fri Dec 26 07:53:51 UTC 2025

System load: 0.0 Temperature: -273.1 C
Usage of /: 34.7% of 6.71GB Processes: 115
Memory usage: 23% Users logged in: 0
Swap usage: 0% IPv4 address for ens5: 10.0.2.36

Expanded Security Maintenance for Applications is not enabled.

94 updates can be applied immediately.
88 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable
Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

Last login: Fri Dec 26 07:50:07 2025 from 13.233.177.5
ubuntu@ip-10-0-2-36:~$
```

At the bottom, it shows the instance ID: i-0a778f8488907c266 (deployment-1) and its public and private IPs: 43.205.143.206 and 10.0.2.36 respectively.

Verifying Docker Installation

System information as of Fri Dec 26 07:53:51 UTC 2025

```
System load: 0.0      Temperature: -273.1 C
Usage of /: 34.7% of 6.71GB  Processes: 115
Memory usage: 29%      Users logged in: 0
Swap usage: 0%         IPv4 address for ens5: 10.0.2.36
```

Expanded Security Maintenance for Applications is not enabled.

74 updates can be applied immediately.
28 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See <https://ubuntu.com/esm> or run: sudo pro status

```
Last login: Fri Dec 26 07:50:07 2025 from 13.233.177.5
ubuntu@ip-10-0-2-36:~$ sudo su
root@ip-10-0-2-36:/home/ubuntu# apt update
Hit:1 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Hit:2 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease
Hit:3 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease
Hit:4 https://download.docker.com/linux/ubuntu noble InRelease
Hit:5 https://security.ubuntu.com/ubuntu noble-security InRelease
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
68 packages can be upgraded. Run 'apt list --upgradable' to see them.
root@ip-10-0-2-36:/home/ubuntu# docker --version
Docker version 29.1.3, build f52614d
root@ip-10-0-2-36:/home/ubuntu#
```

i-0a778f8488907c266 (deployment-1)

PublicIPs: 43.205.143.206 PrivateIPs: 10.0.2.36

CloudShell Feedback Console Mobile App © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Making Directory for Rest API

System information as of Fri Dec 26 07:50:07 2025 from 13.233.177.5

```
System load: 0.0      Temperature: -273.1 C
Usage of /: 34.7% of 6.71GB  Processes: 115
Memory usage: 29%      Users logged in: 0
Swap usage: 0%         IPv4 address for ens5: 10.0.2.36
```

Expanded Security Maintenance for Applications is not enabled.

74 updates can be applied immediately.
28 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See <https://ubuntu.com/esm> or run: sudo pro status

```
Last login: Fri Dec 26 07:50:07 2025 from 13.233.177.5
ubuntu@ip-10-0-2-36:~$ sudo su
root@ip-10-0-2-36:/home/ubuntu# apt update
Hit:1 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Hit:2 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease
Hit:3 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease
Hit:4 https://download.docker.com/linux/ubuntu noble InRelease
Hit:5 https://security.ubuntu.com/ubuntu noble-security InRelease
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
68 packages can be upgraded. Run 'apt list --upgradable' to see them.
root@ip-10-0-2-36:/home/ubuntu# docker --version
Docker version 29.1.3, build f52614d
root@ip-10-0-2-36:/home/ubuntu# mkdir rest-api
root@ip-10-0-2-36:/home/ubuntu# cd rest-api
root@ip-10-0-2-36:/home/ubuntu/rest-api#
```

i-0a778f8488907c266 (deployment-1)

PublicIPs: 43.205.143.206 PrivateIPs: 10.0.2.36

CloudShell Feedback Console Mobile App © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Installing Node.js

```
Setting up node-deep-equal (2.2.3+-cs43,15.94-1) ...
Setting up libjs-util (0.12.5+-cs3,0.10-1) ...
Setting up node-babel-plugin-lodash (3.3.4+-cs2.0.1-7) ...
Setting up node-jest-runtimebundle (29.6.2-ds1+cs73.45.28-7) ...
Setting up node-sass (16.3.7-ds1+cs50.9.19-4) ...
Setting up node-util (0.12.5+-1.0.10-1) ...
Setting up webpack (5.76.1+dfsg1+cs17.16.16-1) ...
Setting up node-assert (2.0.0+-cs3.9.8-2) ...
Setting up node-css-loader (6.8.1+-cs14.0.17-1) ...
Setting up node-parse-json (5.2.0+-cs5.1.7-1) ...
Setting up npm (9.2.0-ds1-2) ...
Processing triggers for libc-bin (2.39-0ubuntu8.6) ...
Processing triggers for man-db (2.12.0-4build2) ...
Processing triggers for sgml-base (1.31) ...
Setting up libfontconfig1:amd64 (2.19.0-1.1ubuntu2) ...
Setting up liblxml2:amd64 (2.10.3-6.1build1) ...
Setting up libxslt1.1:amd64 (2.1.3-3.1ubuntu5) ...
Setting up xl11-utils (7.74.6build2) ...
Setting up libc-devtools (2.39-0ubuntu8.6) ...
Processing triggers for libc-bin (2.39-0ubuntu8.6) ...
Scanning processes...
Scanning linux images...

Running kernel seems to be up-to-date.

No services need to be restarted.

No containers need to be restarted.

No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.

root@ip-10-0-2-36:/home/ubuntu/rest-api#
```

i-0a778f8488907c266 (deployment-1)
PublicIPs: 43.205.143.206 PrivateIPs: 10.0.2.36

[CloudShell](#) [Feedback](#) [Console Mobile App](#)

Initialize Node Project

```
Setting up libc-devtools (2.39-0ubuntu8.6) ...
Processing triggers for libc-bin (2.39-0ubuntu8.6) ...
Scanning processes...
Scanning linux images...

Running kernel seems to be up-to-date.

No services need to be restarted.

No containers need to be restarted.

No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.

root@ip-10-0-2-36:/home/ubuntu/rest-api# npm init -y

Wrote to /home/ubuntu/rest-api/package.json:

{
  "name": "rest-api",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "keywords": [],
  "author": "",
  "license": "ISC"
}

root@ip-10-0-2-36:/home/ubuntu/rest-api#
```

i-0a778f8488907c266 (deployment-1)
PublicIPs: 43.205.143.206 PrivateIPs: 10.0.2.36

[CloudShell](#) [Feedback](#) [Console Mobile App](#)

Install Express

No containers need to be restarted.
No user sessions are running outdated binaries.
No VM guests are running outdated hypervisor (qemu) binaries on this host.
root@ip-10-0-2-36:/home/ubuntu/rest-api# npm init -y
Wrote to /home/ubuntu/rest-api/package.json:
{
 "name": "rest-api",
 "version": "1.0.0",
 "description": "",
 "main": "index.js",
 "scripts": {
 "test": "echo \"Error: no test specified\" & exit 1"
 },
 "keywords": [],
 "author": "",
 "license": "ISC"
}

root@ip-10-0-2-36:/home/ubuntu/rest-api# npm install express
added 65 packages, and audited 66 packages in 3s
22 packages are looking for funding
 run `npm fund` for details
found 0 vulnerabilities
root@ip-10-0-2-36:/home/ubuntu/rest-api#

i-0a778f8488907c266 (deployment-1)
Public IPs: 43.205.143.206 Private IPs: 10.0.2.36

CloudShell Feedback Console Mobile App © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

API File Creation

```
const express = require("express");
const app = express();

app.use(express.json());

app.get("/", (req, res) => {
  res.json({ message: "REST API is running successfully" });
});

app.get("/health", (req, res) => {
  res.status(200).json({ status: "UP" });
});

app.get("/api/users", (req, res) => {
  res.json([
    { id: 1, name: "Avin" },
    { id: 2, name: "Devops User" }
  ]);
});

const PORT = 3000;
app.listen(PORT, () => {
  console.log(`Server running on port ${PORT}`);
});
```

-- INSERT --

i-0a778f8488907c266 (deployment-1)
Public IPs: 43.205.143.206 Private IPs: 10.0.2.36

CloudShell Feedback Console Mobile App © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Testing API Local

Account ID: 123456789012

```
"name": "rest-api",
"version": "1.0.0",
"description": "",
"main": "index.js",
"scripts": {
  "test": "echo \"Error: no test specified\" && exit 1"
},
"keywords": [],
"author": "",
"license": "ISC"
}

root@ip-10-0-2-36:/home/ubuntu/rest-api# npm install express
added 65 packages, and audited 66 packages in 3s

22 packages are looking for funding
  run 'npm fund' for details

found 0 vulnerabilities
root@ip-10-0-2-36:/home/ubuntu/rest-api# vim app.js
root@ip-10-0-2-36:/home/ubuntu/rest-api# node app.js
Server running on port 3000
^C
root@ip-10-0-2-36:/home/ubuntu/rest-api# node app.js
Server running on port 3000

root@ip-10-0-2-36:/home/ubuntu/rest-api# curl http://localhost:3000/health
curl: (7) Failed to connect to localhost port 3000 after 0 ms: Couldn't connect to server
root@ip-10-0-2-36:/home/ubuntu/rest-api# node app.js
Server running on port 3000
```

Creating DockerFile

```
"scripts": {
  "test": "echo \"Error: no test specified\" && exit 1"
},
"keywords": [],
"author": "",
"license": "ISC"
}

root@ip-10-0-2-36:/home/ubuntu/rest-api# npm install express
added 65 packages, and audited 66 packages in 3s

22 packages are looking for funding
  run 'npm fund' for details

found 0 vulnerabilities
root@ip-10-0-2-36:/home/ubuntu/rest-api# vim app.js
root@ip-10-0-2-36:/home/ubuntu/rest-api# node app.js
Server running on port 3000
^C
root@ip-10-0-2-36:/home/ubuntu/rest-api# node app.js
Server running on port 3000
^C
root@ip-10-0-2-36:/home/ubuntu/rest-api# curl http://localhost:3000/health
curl: (7) Failed to connect to localhost port 3000 after 0 ms: Couldn't connect to server
root@ip-10-0-2-36:/home/ubuntu/rest-api# node app.js
Server running on port 3000
^C
root@ip-10-0-2-36:/home/ubuntu/rest-api# cd ~/rest-api
bash: cd: /root/rest-api: No such file or directory
root@ip-10-0-2-36:/home/ubuntu/rest-api# vim dockerfile
root@ip-10-0-2-36:/home/ubuntu/rest-api# [REDACTED]
```

The screenshot shows a CloudShell terminal window with the following Dockerfile content:

```
FROM node:18-alpine
WORKDIR /app
COPY package*.json .
RUN npm install --only=production
COPY . .
EXPOSE 3000
CMD ["node", "app.js"]
```

The terminal also displays deployment information for a specific instance:

-- INSERT --

i-0a778f8488907c266 (deployment-1)
Public IPs: 43.205.143.206 Private IPs: 10.0.2.36

CloudShell Feedback Console Mobile App

© 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Building Docker Image

```
[+] Building 9.0s (8/9)
=> [internal] load build definition from Dockerfile
[+] Building 10.5s (10/10) FINISHED
=> [internal] load build definition from Dockerfile
=> => transferring dockerfile: 178B
=> [internal] load metadata for docker.io/library/node:18-alpine
=> [internal] load .dockerignore
=> transferring context: 2B
[+] [1/5] FROM docker.io/library/node:18-alpine@sha256:8d421d663b4c28f3d3ebc498332f249011d118945588d0a35cb9bc4b8ca09d9e
=> => resolve docker.io/library/node:18-alpine@sha256:8d421d663b4c28f3d3ebc498332f249011d118945588d0a35cb9bc4b8ca09d9e
=> => sha256:25ff2da83641908f65c3a7d48d0409d61b2ccfaab220b9ea70b90df5a2e0549 446B
=> => sha256:1ea94c89ee5c0920c6340ab0ed4db64919e6eda1937f430bd47f026702d6e3 1.26MB
=> => sha256:f10117141fd8345b520319a5994c2d0a094a19a4b4cefeaf161b2b54200 40.09MB / 40.01MB
=> => sha256:f10117141fd8345b520319a5994c2d0a094a19a4b4cefeaf161b2b54200 40.09MB / 40.01MB
=> => extracting sha256:f10117141fd8345b520319a5994c2d0a094a19a4b4cefeaf161b2b54200
=> => extracting sha256:d7d1de34b5c203d162901e6b8994ch23049ae049a0eabc4feaf161b2b5a3d0e
=> => extracting sha256:1e5a94c89ee5c0826c540ab0d4db6491c96da01937f430bd47f02d6702d6e3
=> => extracting sha256:1e5a94c89ee5c0826c540ab0d4db6491c96da01937f430bd47f02d6702d6e3
=> [internal] load build context
=> transferring context: 2.14MB
(2/5) WORKDIR /app
(3/5) COPY package*.json .
(4/5) RUN npm install --only=production
=> (5/5) COPY . .
=> exporting to image
=> => exporting layers
=> => exporting manifest sha256:76aa6391dad6d16d218ce824209ab5fb00593a18011a6a98905bbddbf0lab0a14
=> => exporting manifest sha256:12cd45623cd7d74d4413df9e57cheksf9662f12ef0e6faeb1fc215fcfaa610
=> => exporting attestation manifest sha256:1609041b68a78a4cf5f92e0a4c6cf2cccaf02f0cd2fb2b0f708311ab492dd02
=> => exporting manifest list sha256:c22697cfaff00e68b17d4ee3820b2c60a464f9b3ccb28cc32a832f7ddd6cc2cf
=> => naming to docker.io/library/rest-api:v1
=> => unpacking to docker.io/library/rest-api:v1
root@ip-10-0-2-36:/home/ubuntu/rest-api#
```



```

=> (4/5) RUN npm install --only-production
=> (5/5) COPY .
=> exporting image
=> exporting layers
=> >> exporting manifest sha256:76aa6391dad6d16d219ce824209ab5fb0583a18011a6a998905bbddbf0lab0a14
=> >> exporting config sha256:1:2c4d8623cdc7d4443df9e57feba9e62f2e0e6acb3edaf1fc215cfccfa5610
=> >> exporting attestation manifest sha256:1609041b69a78a44c5ff92e0a4c5f2ceca8f2fcdedfb2b0f708311ab492dd02
=> >> exporting manifest list sha256:5c32697c7af90ee88b17d4ee38209bc60a84f9b3ccb286c32a832f7ddd6cc2cf
=> >> naming to docker.io/library/rest-api:v1
=> >> unpacking to docker.io/library/rest-api:v1
root@ip-10-0-2-36:/home/ubuntu/rest-api# docker images
e
IMAGE          ID             SIZE      CREATED        LAST ACTIVE
rest-api:v1    5c32697c7af8  194MB     46.9MB   2023-07-10T10:08:23Z
root@ip-10-0-2-36:/home/ubuntu/rest-api# docker run -d -p 3000:3000 rest-api:v1
f87ef72a964362b002179e82d8fafeba0db47224960c003030b97e6fd20671
root@ip-10-0-2-36:/home/ubuntu/rest-api# docker run -d -p 3000:3000 rest-api:v1
84b746d1741fc5cc551dc646b9172e1d2846420a39bdf7e22c618203e54e0801
docker: Error response from daemon: failed to set up container networking: driver failed programming external connectivity on endpoint nifty_solomon (9c6fb7a6358d3e1a10f08e035cb8bec0f46e
6ee974eb937d8d31cd0ae3ff4ba): Bind for 0.0.0.0:3000 failed: port is already allocated
Run 'docker run --help' for more information
root@ip-10-0-2-36:/home/ubuntu/rest-api# docker run -d -p 3000:3000 rest-api:v1
29c62157e36bae7a03795d3e12e43f1ec575c938ce44e5c6e0ff8e32f6e504
docker: Error response from daemon: failed to set up container networking: driver failed programming external connectivity on endpoint flamboyant_mayer (dcca920ab1ed7c4dfb1bb6075e515bc35
0952f129d99002f28ef50c53466f6929): Bind for 0.0.0.0:3000 failed: port is already allocated
Run 'docker run --help' for more information
root@ip-10-0-2-36:/home/ubuntu/rest-api# docker ps
CONTAINER ID        IMAGE               COMMAND                  CREATED             STATUS              PORTS
f87ef72a964362b002179e82d8fafeba0db47224960c003030b97e6fd20671
root@ip-10-0-2-36:/home/ubuntu/rest-api# docker run -d -p 3000:3000 rest-api:v1
84b746d1741fc5cc551dc646b9172e1d2846420a39bdf7e22c618203e54e0801
docker: Error response from daemon: failed to set up container networking: driver failed programming external connectivity on endpoint nifty_solomon (9c6fb7a6358d3e1a10f08e035cb8bec0f46e
6ee974eb937d8d31cd0ae3ff4ba): Bind for 0.0.0.0:3000 failed: port is already allocated
Run 'docker run --help' for more information
root@ip-10-0-2-36:/home/ubuntu/rest-api# docker run -d -p 3000:3000 rest-api:v1
29c62157e36bae7a03795d3e12e43f1ec575c938ce44e5c6e0ff8e32f6e504
docker: Error response from daemon: failed to set up container networking: driver failed programming external connectivity on endpoint flamboyant_mayer (dcca920ab1ed7c4dfb1bb6075e515bc35
0952f129d99002f28ef50c53466f6929): Bind for 0.0.0.0:3000 failed: port is already allocated
Run 'docker run --help' for more information
root@ip-10-0-2-36:/home/ubuntu/rest-api# docker ps
CONTAINER ID        IMAGE               COMMAND                  CREATED             STATUS              PORTS
f87ef72a964362b002179e82d8fafeba0db47224960c003030b97e6fd20671
root@ip-10-0-2-36:/home/ubuntu/rest-api# curl http://localhost:3000/health
{"status": "UP"}root@ip-10-0-2-36:/home/ubuntu/rest-api#
```

i-0a778f8488907c266 (deployment-1)

PublicIPs: 43.205.143.206 PrivateIPs: 10.0.2.36

Test Containerization API

```

=> (5/5) COPY .
=> exporting image
=> exporting layers
=> >> exporting manifest sha256:76aa6391dad6d16d219ce824209ab5fb0583a18011a6a998905bbddbf0lab0a14
=> >> exporting config sha256:1:2c4d8623cdc7d4443df9e57feba9e62f2e0e6acb3edaf1fc215cfccfa5610
=> >> exporting attestation manifest sha256:1609041b69a78a44c5ff92e0a4c5f2ceca8f2fcdedfb2b0f708311ab492dd02
=> >> exporting manifest list sha256:5c32697c7af90ee88b17d4ee38209bc60a84f9b3ccb286c32a832f7ddd6cc2cf
=> >> naming to docker.io/library/rest-api:v1
=> >> unpacking to docker.io/library/rest-api:v1
root@ip-10-0-2-36:/home/ubuntu/rest-api# docker images
e
IMAGE          ID             SIZE      CREATED        LAST ACTIVE
rest-api:v1    5c32697c7af8  194MB     46.9MB   2023-07-10T10:08:23Z
root@ip-10-0-2-36:/home/ubuntu/rest-api# docker run -d -p 3000:3000 rest-api:v1
f87ef72a964362b002179e82d8fafeba0db47224960c003030b97e6fd20671
root@ip-10-0-2-36:/home/ubuntu/rest-api# docker run -d -p 3000:3000 rest-api:v1
84b746d1741fc5cc551dc646b9172e1d2846420a39bdf7e22c618203e54e0801
docker: Error response from daemon: failed to set up container networking: driver failed programming external connectivity on endpoint nifty_solomon (9c6fb7a6358d3e1a10f08e035cb8bec0f46e
6ee974eb937d8d31cd0ae3ff4ba): Bind for 0.0.0.0:3000 failed: port is already allocated
Run 'docker run --help' for more information
root@ip-10-0-2-36:/home/ubuntu/rest-api# docker run -d -p 3000:3000 rest-api:v1
29c62157e36bae7a03795d3e12e43f1ec575c938ce44e5c6e0ff8e32f6e504
docker: Error response from daemon: failed to set up container networking: driver failed programming external connectivity on endpoint flamboyant_mayer (dcca920ab1ed7c4dfb1bb6075e515bc35
0952f129d99002f28ef50c53466f6929): Bind for 0.0.0.0:3000 failed: port is already allocated
Run 'docker run --help' for more information
root@ip-10-0-2-36:/home/ubuntu/rest-api# docker ps
CONTAINER ID        IMAGE               COMMAND                  CREATED             STATUS              PORTS
f87ef72a964362b002179e82d8fafeba0db47224960c003030b97e6fd20671
root@ip-10-0-2-36:/home/ubuntu/rest-api# curl http://localhost:3000/health
{"status": "UP"}root@ip-10-0-2-36:/home/ubuntu/rest-api#
```

i-0a778f8488907c266 (deployment-1)

PublicIPs: 43.205.143.206 PrivateIPs: 10.0.2.36

Creating Repository

Screenshot of the AWS ECR Private Registry Repositories page:

The page shows a single repository named "web-deployment".

Repository name	URI	Created at	Tag immutability	Encryption type
web-deployment	122385533950.dkr.ecr.ap-south-1.amazonaws.com/web-deployment	December 26, 2025, 10:29:43 (UTC+05:50)	Mutable	AES-256

Screenshot of the AWS ECR Create private repository page:

The "General settings" section shows a repository name of "deployment-1".

The "Image tag settings" section includes options for image tag mutability (set to "Mutable") and mutable tag exclusions.

The "Encryption settings" section is present but empty.

The screenshot shows the AWS ECR console interface. On the left, there's a navigation sidebar for Amazon Elastic Container Service with sections for Private registry (Repositories, Features & Settings) and Public registry (Repositories, Settings). Below that are links for ECR public gallery, Amazon ECS, and Amazon EKS. At the bottom of the sidebar are Getting started and Documentation links. The main content area is titled "Private repositories (2)". It displays two entries: "deployment-1" and "web-deployment". Each entry has a URI, creation date, tag immutability (Mutable), and encryption type (AES-256). A banner at the top of the main content area informs users about managed signing, stating "Managed signing now available" and "Automatically sign your container images upon push to verify authenticity and ensure supply chain security. Configure image signing". There are buttons for "View push commands", "Delete", "Actions", and "Create repository". The bottom of the page includes CloudShell, Feedback, and Console Mobile App links, along with standard footer links for Privacy, Terms, and Cookie preferences.

Installing AWS CLI

The screenshot shows an AWS CloudShell terminal window. The URL in the address bar is `ap-south-1.console.aws.amazon.com/ec2-instance-connect/ssh/home?region=ap-south-1&connType=standard&instanceId=i-0a77fb488907c266&osUser=ubuntu&sshPort=22&address...`. The terminal output shows the installation of the AWS CLI. It starts with creating directories and files in the `aws/dist/awscli/customizations/wizard` and `aws/dist/awscli/_data` paths. It then inflates several files from the `prompt_toolkit` package, including `LICENSE`, `AUTHORS.rst`, and `METADATA`. Finally, it installs the AWS CLI using `sudo ./aws/install`. The terminal concludes with the AWS CLI version information: `aws-cli/2.32.23 Python/3.13.11 Linux/6.14.0-1015-aws exe/x86_64.ubuntu.24`.

Creating user

The screenshot shows the AWS Identity and Access Management (IAM) service in the AWS Management Console. The URL is `us-east-1.console.aws.amazon.com/iam/home?region=ap-south-1#/users`. The left sidebar shows the navigation menu for IAM, with 'Users' selected. The main content area displays a table titled 'Users (1) Info'. The table has one row for a user named 'varsha'. The columns include 'User name' (varsha), 'Path' (/), 'Group' (0), 'Last activity' (3 hours ago), 'MFA' (-), 'Password age' (-), 'Console last sign-in' (-), and 'Access key last used' (-). There are buttons for 'Delete' and 'Create user' at the top right of the table.

User name	Path	Group	Last activity	MFA	Password age	Console last sign-in	Access key last used
varsha	/	0	3 hours ago	-	-	-	-

us-east-1.console.aws.amazon.com/iam/home?region=ap-south-1#/users/create

IAM > Users > Create user

Review and create

Review your choices. After you create the user, you can view and download the autogenerated password, if enabled.

User details

User name	varsha1	Console password type	None	Require password reset	No
-----------	---------	-----------------------	------	------------------------	----

Permissions summary

Name	Type	Used as
AmazonEC2ContainerRegistryFullAccess	AWS managed	Permissions policy
AmazonEC2FullAccess	AWS managed	Permissions policy
AmazonECS_FullAccess	AWS managed	Permissions policy

Tags - optional

Tags are key-value pairs you can add to AWS resources to help identify, organize, or search for resources. Choose any tags you want to associate with this user.

No tags associated with the resource.

Add new tag

You can add up to 50 more tags.

Create user

us-east-1.console.aws.amazon.com/iam/home?region=ap-south-1#/users

IAM > Users

User created successfully

You can view and download the user's password and email instructions for signing in to the AWS Management Console.

View user

Users (2) Info

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

User name	Path	Groups	Last activity	MFA	Password age	Console last sign-in	Access key ID
varsha	/	0	3 hours ago	-	-	-	Active - AKIARY7V7B...
varsha1	/	0	-	-	-	-	-

The screenshot shows the AWS IAM User Details page for a user named 'varsha1'. A prominent green success message box at the top left states 'User created successfully' and provides instructions for viewing and downloading password and email instructions. Below this, the user's details are listed under the 'Summary' tab, including the ARN, console access status, and last sign-in information. The 'Permissions' tab is selected, showing a list of attached policies, with one policy named 'AmazonEC2ContainerRegistryFullAccess' visible. The sidebar on the left contains navigation links for Access management, Access reports, and other IAM features.

The screenshot shows the second step of the 'Create access key' wizard for the user 'varsha1'. It asks for the use case for the new access key. The 'Command Line Interface (CLI)' option is selected, with a note explaining it enables AWS CLI access. Other options include 'Local code', 'Application running on an AWS compute service', 'Third-party service', 'Application running outside AWS', and 'Other'. A 'Confirmation' section at the bottom contains a checkbox for accepting recommendations to use AWS CLI V2 or CloudShell, which is checked. The sidebar on the left shows the current step: 'Step 2 - optional' with 'Set description tag' and 'Step 3 Retrieve access keys'.

Creating Access Key

```
System load: 0.08      Temperature: -273.1 C
Usage of /: 61.8% of 6.71GB  Processes: 118
Memory usage: 39%      Users logged in: 0
Swap usage: 0%          IPv4 address for ens5: 10.0.2.36

Expanded Security Maintenance for Applications is not enabled.

71 updates can be applied immediately.
27 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

Last login: Fri Dec 26 08:32:07 2025 from 13.233.177.4
ubuntu@ip-10-0-2-36:~$ sudo su
root@ip-10-0-2-36:~/home/ubuntu# aws configure
AWS Access Key ID [*****4YTR]: AKIARY7V7BP7JQ7L4YTR
AWS Secret Access Key [*****4YTR]: 99D3dQwz4sIPq7WD2lWffzDIDt4QsgBcodPoW+u
Default region name [ap-south-1]: ap-south-1
Default output format [json]: json
root@ip-10-0-2-36:~/home/ubuntu# aws ecr get-login-password --region ap-south-1 \
| docker login --username AWS --password-stdin 122385533950.dkr.ecr.ap-south-1.amazonaws.com/deployment-1
WARNING! Your credentials are stored unencrypted in '/root/.docker/config.json'.
Configure a credential helper to remove this warning. See
https://docs.docker.com/go/credential-store/
Login Succeeded
root@ip-10-0-2-36:~/home/ubuntu# i-0a778f8488907c266 (deployment-1)
PublicIPs: 43.205.143.206 PrivateIPs: 10.0.2.36
```

CloudShell Feedback Console Mobile App © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

```
System load: 0.08      Temperature: -273.1 C
Usage of /: 61.8% of 6.71GB  Processes: 118
Memory usage: 39%      Users logged in: 0
Swap usage: 0%          IPv4 address for ens5: 10.0.2.36

Expanded Security Maintenance for Applications is not enabled.

71 updates can be applied immediately.
27 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

Last login: Fri Dec 26 08:32:07 2025 from 13.233.177.4
ubuntu@ip-10-0-2-36:~$ sudo su
root@ip-10-0-2-36:~/home/ubuntu# aws configure
AWS Access Key ID [*****4YTR]: AKIARY7V7BP7JQ7L4YTR
AWS Secret Access Key [*****4YTR]: 99D3dQwz4sIPq7WD2lWffzDIDt4QsgBcodPoW+u
Default region name [ap-south-1]: ap-south-1
Default output format [json]: json
root@ip-10-0-2-36:~/home/ubuntu# aws ecr get-login-password --region ap-south-1 \
| docker login --username AWS --password-stdin 122385533950.dkr.ecr.ap-south-1.amazonaws.com/deployment-1
WARNING! Your credentials are stored unencrypted in '/root/.docker/config.json'.
Configure a credential helper to remove this warning. See
https://docs.docker.com/go/credential-store/
Login Succeeded
root@ip-10-0-2-36:~/home/ubuntu# i-0a778f8488907c266 (deployment-1)
PublicIPs: 43.205.143.206 PrivateIPs: 10.0.2.36
```

CloudShell Feedback Console Mobile App © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences