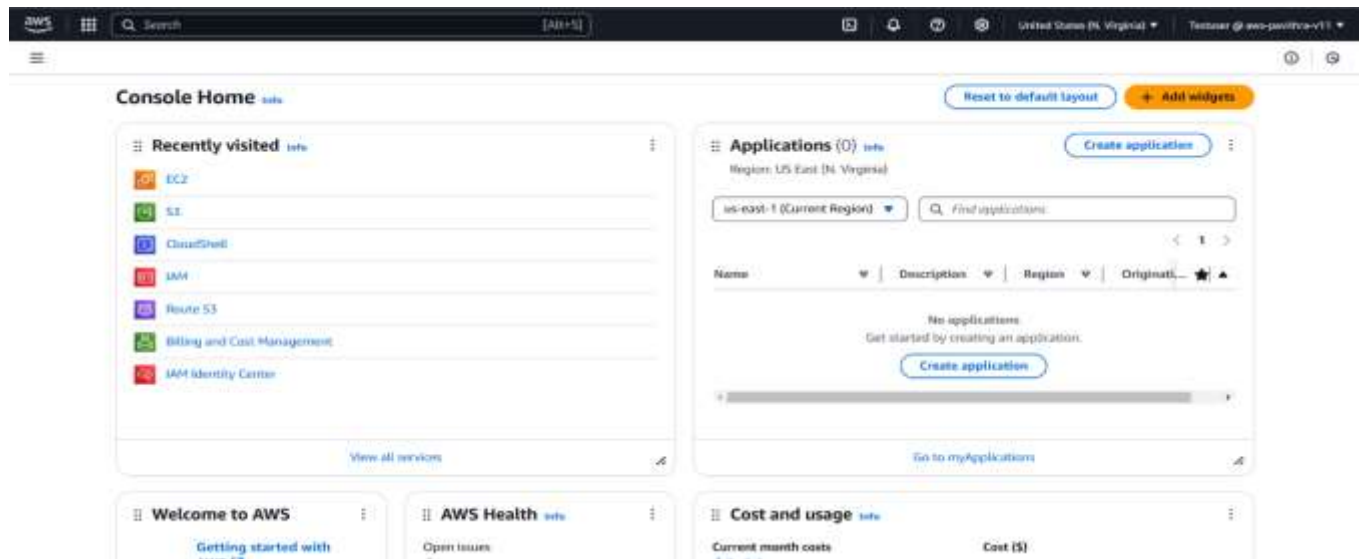
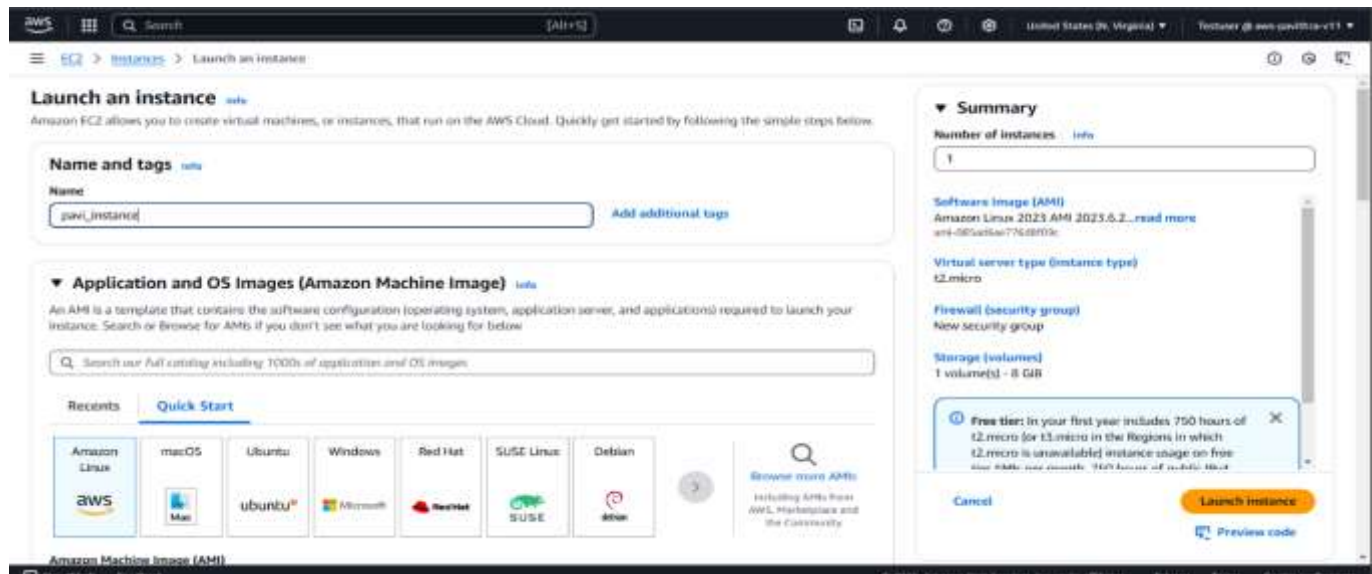


Creating a Linux EC2 Instance on AWS

Step 1: Log In to AWS Management Console



Step 2: Navigate to the EC2 Dashboard



Step 3: Configure Instance Details

Step 3.1: Choose an Amazon Machine Image (AMI)

Search

[AMI+5]

United States (N. Virginia) | Tester @ aws-pentire-v11

EC2 > Instances > Launch an instance

Amazon Linux

macOS

Ubuntu

Windows

Red Hat

SUSE Linux

Debian

Browse more AMIs

including AMIs from AWS Marketplace and the Community

Amazon Machine Image (AMI)

Amazon Linux 2023 AMI
ami-085ad3ae775cd8b09c (64-bit (x86), x86_64-preferred) / ami-085ad3ae775cd8b09c (64-bit (ARM), aarch64)
Virtualization: hvm | EBS: standard | Root: device: /dev/xvda

Free tier eligible

Description

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

Amazon Linux 2023 AMI 2023.6.20250203.1 x86_64 HVM kernel-6.1

Architecture

64-bit (x86)

Boot mode

uefi-preferred

AMI ID

ami-085ad3ae775cd8b09c

Username

ec2-user

Verified provider

Instance type

Info | Get advice

Instance type

t2.micro

Free tier eligible

Summary

Number of instances

Info

1

t2.micro

Firewall (security group)

New security group

Storage (volumes)

1 volume(s) - 8 GiB

Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 750 hours of public IPv4 address usage per month, 30 GiB of EBS storage, 2 million I/Os, 1 GiB of snapshots, and 100 GiB of bandwidth to the internet.

Cancel

Launch instance

Preview code

Step 3.2: Choose an Instance Type

Search

[AMI+5]

United States (N. Virginia) | Tester @ aws-pentire-v11

EC2 > Instances > Launch an instance

Instance type

Info | Get advice

Instance type

t2.micro

Free tier eligible

Family: t2 | 1 vCPU | 1 GiB Memory | Current generation hvm

On-Demand Windows base pricing: 0.0162 USD per Hour

On-Demand Ubuntu Pro base pricing: 0.0134 USD per Hour

On-Demand SUSE base pricing: 0.0110 USD per Hour

On-Demand RHEL base pricing: 0.026 USD per Hour

On-Demand Linux base pricing: 0.0110 USD per Hour

Additional costs apply for AMIs with pre-installed software.

All generations

Compare instance types

Key pair (login)

Info

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

Key pair name - required

gencf_first_key

Create new key pair

Network settings

Info

Network

Info

vpc-09a86c566a0ed0962

Subnet

Info

No preference (Default subnet in any availability zone)

Edit

Summary

Number of instances

Info

1

t2.micro

Firewall (security group)

New security group

Storage (volumes)

1 volume(s) - 8 GiB

Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 750 hours of public IPv4 address usage per month, 30 GiB of EBS storage, 2 million I/Os, 1 GiB of snapshots, and 100 GiB of bandwidth to the internet.

Cancel

Launch instance

Preview code

Step 3.3: Configure Instance Details

Search

[AMI+5]

United States (N. Virginia) | Tester @ aws-pentire-v11

EC2 > Instances > Launch an instance

Amazon Linux

macOS

Ubuntu

Windows

Red Hat

SUSE Linux

Debian

Browse more AMIs

including AMIs from AWS Marketplace and the Community

Amazon Machine Image (AMI)

Amazon Linux 2023 AMI
ami-085ad3ae775cd8b09c (64-bit (x86), x86_64-preferred) / ami-085ad3ae775cd8b09c (64-bit (ARM), aarch64)
Virtualization: hvm | EBS: standard | Root: device: /dev/xvda

Free tier eligible

Description

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

Amazon Linux 2023 AMI 2023.6.20250203.1 x86_64 HVM kernel-6.1

Architecture

64-bit (x86)

Boot mode

uefi-preferred

AMI ID

ami-085ad3ae775cd8b09c

Username

ec2-user

Verified provider

Instance type

Info | Get advice

Instance type

t2.micro

Free tier eligible

Summary

Number of instances

Info

1

t2.micro

Firewall (security group)

New security group

Storage (volumes)

1 volume(s) - 8 GiB

Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 750 hours of public IPv4 address usage per month, 30 GiB of EBS storage, 2 million I/Os, 1 GiB of snapshots, and 100 GiB of bandwidth to the internet.

Cancel

Launch instance

Preview code

Step 4: Add Storage

The screenshot shows the 'Launch an instance' page in the AWS Management Console, specifically the 'Configure storage' step. The interface includes a top navigation bar with the AWS logo, search bar, and user information. The main content area is divided into two panels. The left panel, titled 'Configure storage', shows the 'Root volume' as 'gp5' with a size of '8 GiB' and '3000 IOPS (Not encrypted)'. It also includes a warning about security group rules, a 'Free tier' notice, and a 'Click refresh to view backup information' button. The right panel, titled 'Summary', shows the 'Number of instances' as '1', the instance type as 't2.micro', and the 'Storage (volumes)' as '1 volume(s) - 8 GiB'. It also includes a 'Free tier' notice and 'Launch Instance' and 'Preview code' buttons.

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.

Configure storage [info](#) [Advanced](#)

1x 8 GiB gp5 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 volumes](#)

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](#)

Advanced details [info](#)

Summary

Number of instances [info](#)
1

t2.micro

Firewall (security group)
New security group

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

Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 750 hours of public IPv4 address usage per month, 30 GB of EBS storage, 2 million I/Os, 1 GB of snapshots, and 100 GB of bandwidth to the internet.

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

Step 5: Add Tags (Optional)

Step 6: Configure Security Group

The screenshot shows the 'Launch an instance' page in the AWS Management Console, specifically the 'Configure security group' step. The interface includes a top navigation bar with the AWS logo, search bar, and user information. The main content area is divided into two panels. The left panel, titled 'network settings', shows the 'Network' as 'vpc-0fa86c566a0ed0962', the 'Subnet' as 'No preference (Default subnet in any availability zone)', and the 'Auto-assign public IP' as 'Enable'. It also includes a 'Firewall (security groups)' section with a 'Create security group' button and a 'Select existing security group' button. The right panel, titled 'Summary', shows the 'Number of instances' as '1', the instance type as 't2.micro', and the 'Storage (volumes)' as '1 volume(s) - 8 GiB'. It also includes a 'Free tier' notice and 'Launch Instance' and 'Preview code' buttons.

network settings [info](#) [edit](#)

Network [info](#)
vpc-0fa86c566a0ed0962

Subnet [info](#)
No preference (Default subnet in any availability zone)

Auto-assign public IP [info](#)
Enable

Additional charges apply when opt-in to free tier allowances

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.

[Create security group](#) [Select existing security group](#)

We'll create a new security group called 'launch-wizard-4' with the following rules:

☒ Allow SSH traffic from [Anywhere](#) [0.0.0.0/0](#)

☐ Allow HTTPS traffic from the internet
To set up an endpoint, for example when creating a web server

☒ Allow HTTP traffic from the internet
To set up an endpoint, for example when creating a web server

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.

Summary

Number of instances [info](#)
1

t2.micro

Firewall (security group)
New security group

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

Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 750 hours of public IPv4 address usage per month, 30 GB of EBS storage, 2 million I/Os, 1 GB of snapshots, and 100 GB of bandwidth to the internet.

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

Step 7: Launch Instance

Instance type

Instance type

t2.micro

Family: t2 1 vCPU 1 GB memory Current generation: true

On-Demand Windows base pricing: 0.0162 USD per Hour

On-Demand Ubuntu Pro base pricing: 0.0134 USD per Hour

On-Demand SUSE base pricing: 0.0116 USD per Hour

On-Demand Linux base pricing: 0.0116 USD per Hour

On-Demand RHEL base pricing: 0.026 USD per Hour

Free tier eligible

All generations

Compare instance types

Key pair (login)

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

pevi_first_key

Create new key pair

Network settings

Network

vpc-03a86c566a0ed0962

Subnet

No preference (Default subnet in any availability zone)

Edit

Summary

Number of instances

1

t2.micro

Firewall (security group)

New security group

Storage (volumes)

1 volume(s) - 8 GB

Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier. AMIs per month, 750 hours of public IPv4 address usage per month, 50 GB of EBS storage, 2 million I/Os, 1 GB of snapshots, and 100 GB of bandwidth to the internet.

Cancel

Launch instance

Preview code

Step 8: Verify Your Instance

Dashboard

EC2 Global View

Events

Instances

Instance Types

Launch Templates

Spot Requests

Savings Plans

Reserved Instances

Dedicated Hosts

Capacity Reservations

Images

AMIs

AMI Catalog

Elastic Block Store

Volumes

Snapshots

Lifecycle Manager

Network & Security

Instances (2)

Last updated: less than a minute ago

Connect

Instance state

Actions

Launch instances

Find instance by attribute or tag (case-sensitive)

All states

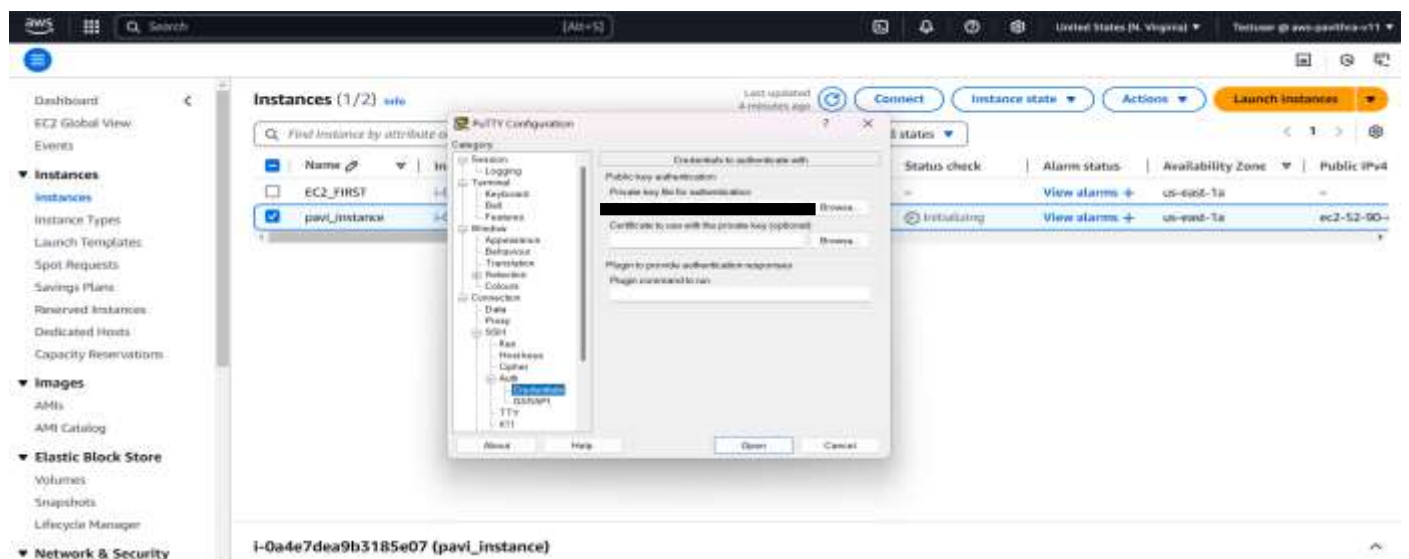
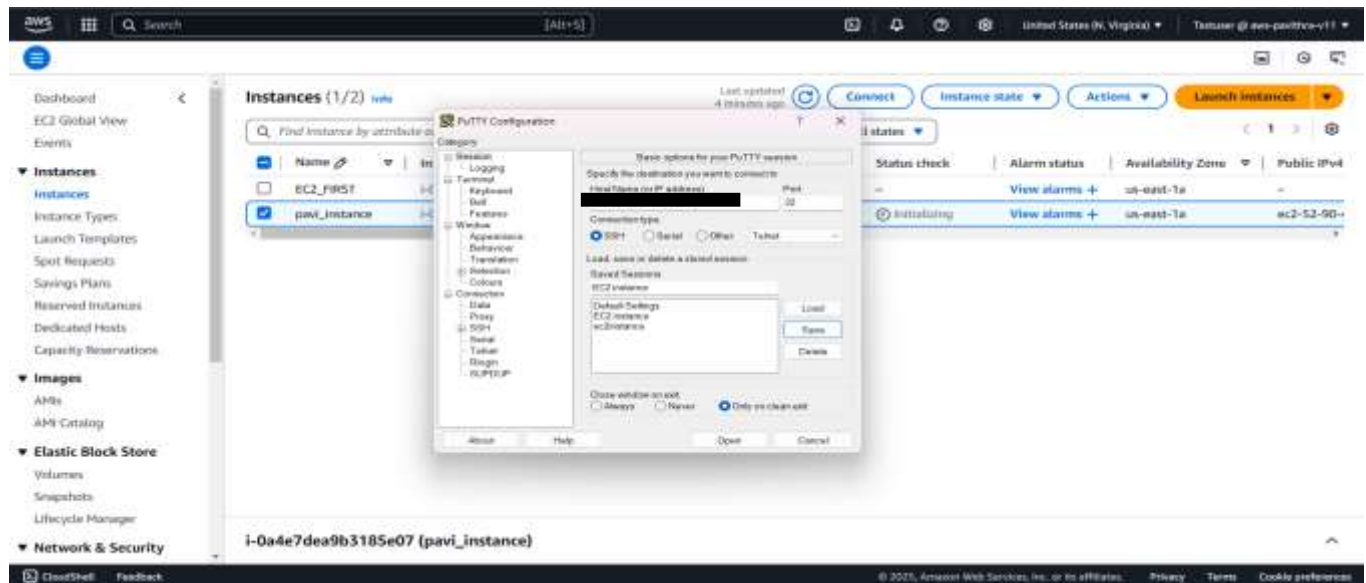
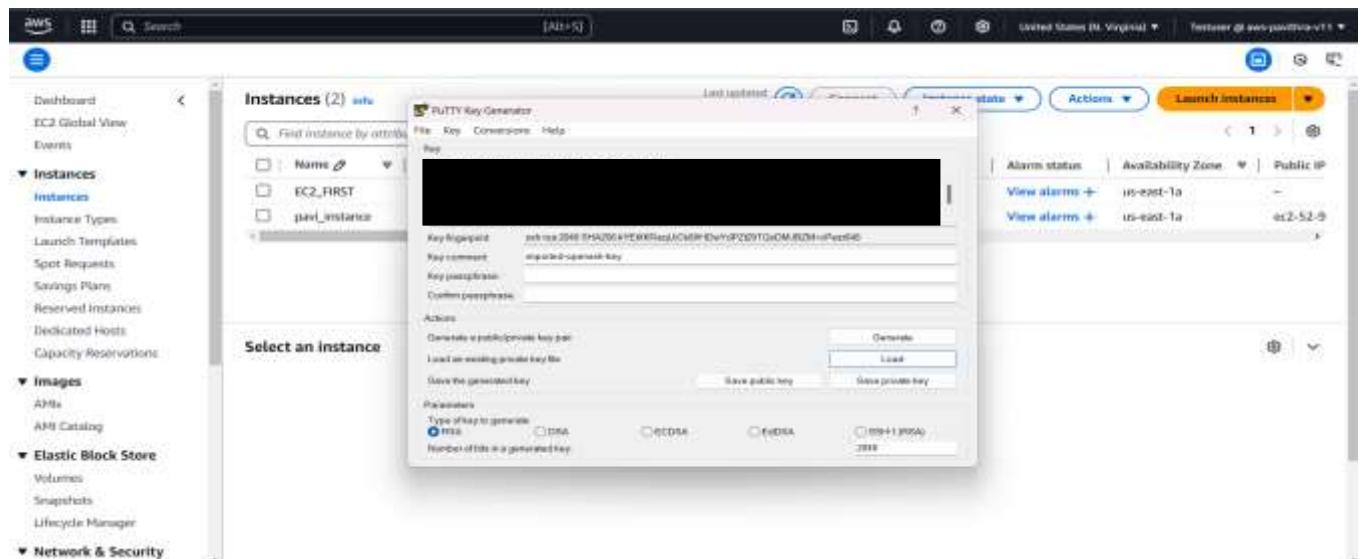
	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IP
<input type="checkbox"/>	EC2_FIRST		Terminated	t2.micro	-	View alarms	us-east-1a	-
<input type="checkbox"/>	pevi_instance		Running	t2.micro	Initializing	View alarms	us-east-1a	ec2-52-9

Select an instance

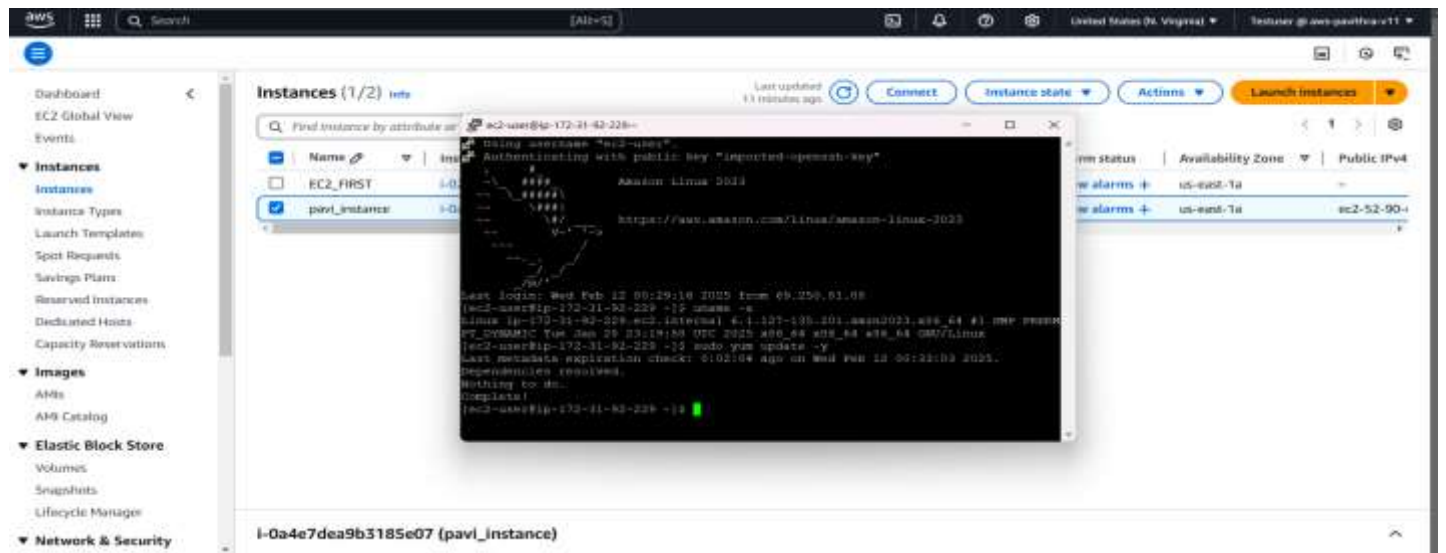
Step 9.1: Connect Using an SSH Client

The screenshot displays the AWS Management Console interface. On the left, the navigation menu includes sections like Dashboard, EC2 Global View, Events, Instances, Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations, Images, AMIs, AMI Catalog, Elastic Block Store, and Network & Security. The main area shows the 'Instances' page for the 'us-east-1' region. A table lists instances, with 'ec2-44-2' selected. The instance details panel on the right shows the instance is a Linux instance with a public IP of 44.201.73.18. The console output of the instance shows the user 'ec2-user' logging in and running commands to check the system information and update the system.

The screenshot displays the AWS Management Console interface. On the left, the navigation menu includes sections for Dashboard, EC2 Global View, Events, Instances, Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations, Images, AMIs, AMI Catalog, Elastic Block Store, Volumes, Snapshots, Lifecycle Manager, and Network & Security. The main content area shows the 'Instances (2)' page. A modal window titled 'Putty Key Generator' is open, showing the 'Actions' tab where a public key is being generated. The background interface includes a search bar, a list of instances with columns for Name, Alarm status, Availability Zone, and Public IP, and a 'Launch instances' button.



Step 10: Verify Connection



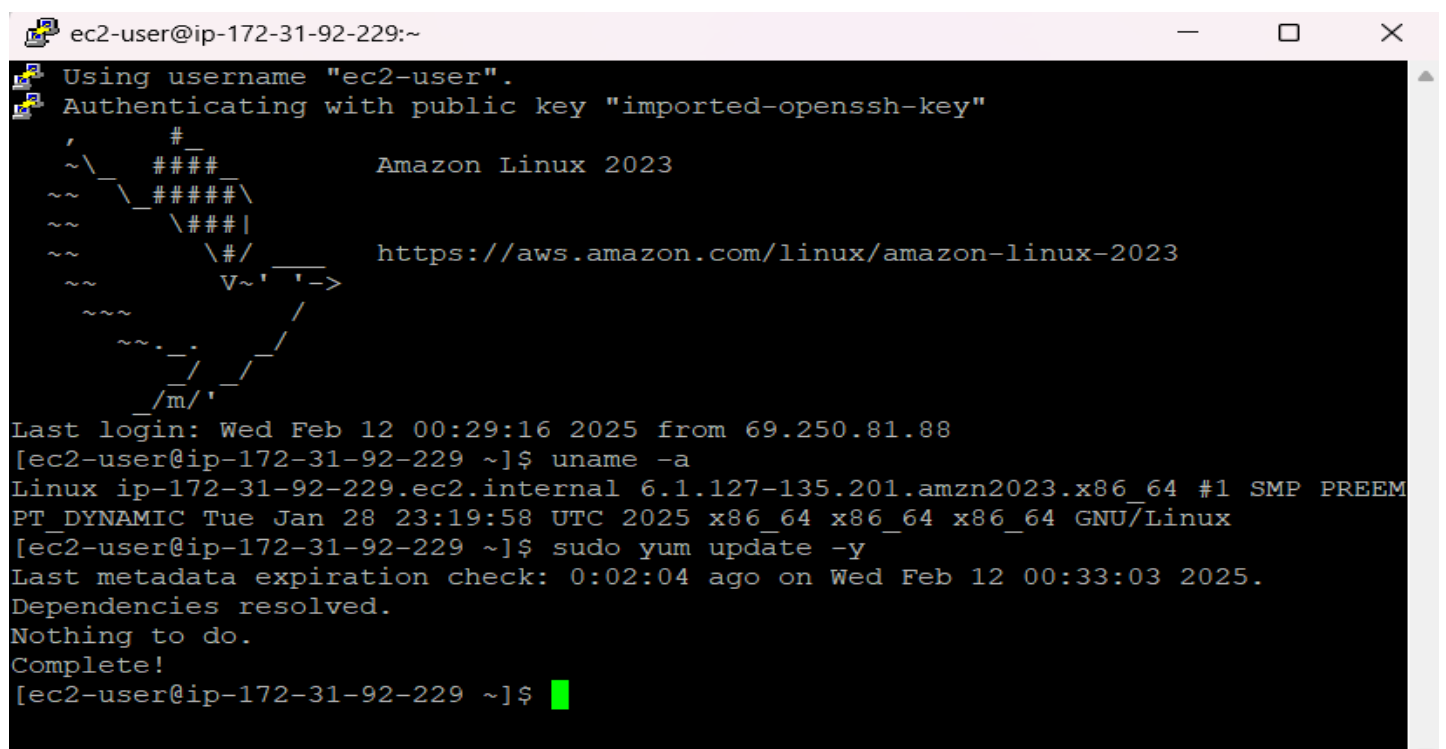
The screenshot shows the AWS Management Console with the EC2 Instances page. A terminal window is open, displaying the login process for an Amazon Linux 2023 instance. The terminal shows the AWS CLI command to connect to the instance using a public key, followed by the login banner and the execution of 'uname -a' and 'sudo yum update -y'.

Instances (1/2)

Name	Instance ID	State	Availability Zone	Public IPv4
EC2_FIRST	i-0a4e7dea9b3185e07	Running	us-east-1a	ec2-52-90-

```
ec2-user@ip-172-31-92-229:~$ awscli --profile testuser --region us-east-1 --key-file imported-openssh-key ec2-ssh --ip-addr 52.90.10.100 --key-file imported-openssh-key
Using username "ec2-user".
Authenticating with public key "imported-openssh-key"
#####
Amazon Linux 2023
#####
https://aws.amazon.com/linux/amazon-linux-2023

Last login: Wed Feb 12 00:29:16 2025 from 69.250.81.88
[ec2-user@ip-172-31-92-229 ~]$ uname -a
Linux ip-172-31-92-229.ec2.internal 6.1.127-135.201.amzn2023.x86_64 #1 SMP PREEMPT_DYNAMIC Tue Jan 28 23:19:58 UTC 2025 x86_64 x86_64 GNU/Linux
[ec2-user@ip-172-31-92-229 ~]$ sudo yum update -y
Last metadata expiration check: 0:02:04 ago on Wed Feb 12 00:33:03 2025.
Dependencies resolved.
Nothing to do.
Complete!
[ec2-user@ip-172-31-92-229 ~]$
```

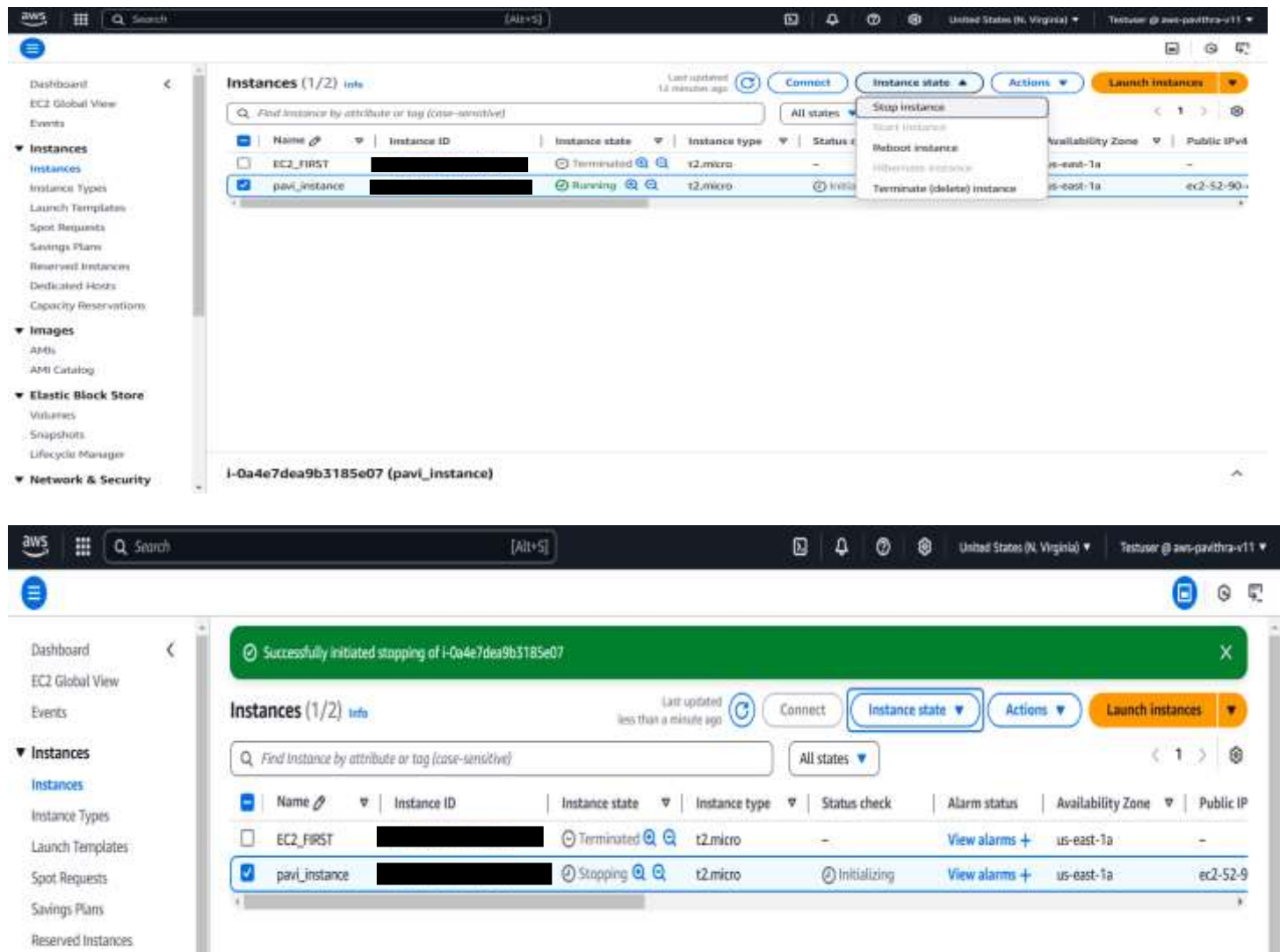


The screenshot shows a terminal window with the following output:

```
ec2-user@ip-172-31-92-229:~$ awscli --profile testuser --region us-east-1 --key-file imported-openssh-key ec2-ssh --ip-addr 52.90.10.100 --key-file imported-openssh-key
Using username "ec2-user".
Authenticating with public key "imported-openssh-key"
#####
Amazon Linux 2023
#####
https://aws.amazon.com/linux/amazon-linux-2023

Last login: Wed Feb 12 00:29:16 2025 from 69.250.81.88
[ec2-user@ip-172-31-92-229 ~]$ uname -a
Linux ip-172-31-92-229.ec2.internal 6.1.127-135.201.amzn2023.x86_64 #1 SMP PREEMPT_DYNAMIC Tue Jan 28 23:19:58 UTC 2025 x86_64 x86_64 GNU/Linux
[ec2-user@ip-172-31-92-229 ~]$ sudo yum update -y
Last metadata expiration check: 0:02:04 ago on Wed Feb 12 00:33:03 2025.
Dependencies resolved.
Nothing to do.
Complete!
[ec2-user@ip-172-31-92-229 ~]$
```

Step 11: Clean Up (Optional)



A brief report summarizing

Step 1: Log In to AWS Management Console

- Access the **AWS Management Console** and sign in.
- Navigate to the **EC2 Dashboard**.

Step 2: Configure EC2 Instance

- **Choose Amazon Machine Image (AMI):** Selected **Amazon Linux** as the operating system.
- **Select Instance Type:** Chose an instance type based on performance needs.
- **Configure Instance Details:** Ensured the instance is properly set up with networking options.

Step 3: Add Storage & Tags

- **Define Storage:** Kept the default storage settings.
- **Add Tags (Optional):** Assigned a tag to easily identify the instance.

Step 4: Configure Security Group

- Configured **security rules** to allow access for:
 - **SSH (Port 22)** – Enabled secure remote login.

- **HTTP (Port 80)** – Allowed web traffic access.

Step 5: Launch and Verify Instance

- Reviewed configurations and **launched the instance**.
- Created and downloaded a **key pair (.pem file)** for secure SSH access.
- Verified that the instance **changed to Running status** in the EC2 dashboard.

Step 6: Connect to the Instance

Option 1: SSH Client (Linux/Mac/Windows Terminal)

1. Open a terminal and navigate to the directory containing the .pem file.
2. Connect using the following command:
3. `ssh -i <your-key.pem> ec2-user@<public-ip>`
4. Accepted the host key fingerprint and successfully logged in.

Option 2: PuTTY (Windows Users)

1. Converted the .pem file to .ppk format using PuTTYgen.
2. Opened **PuTTY**, entered the **Public IP** of the instance like `ec2-user@<public-ip>`
3. Under **Connection** → **SSH** → **Auth->credentials**, loaded the .ppk key and connected.

Step 7: Verify the Connection

- After logging in, ran system verification commands:
- `uname -a`
- `lsb_release -a`
- `sudo yum update -y`
- Ensured connectivity and verified that **Amazon Linux** was running correctly.

Step 8: Clean Up (Optional)

- If the instance is no longer needed, it can be **terminated** from the AWS EC2 dashboard to prevent additional charges.