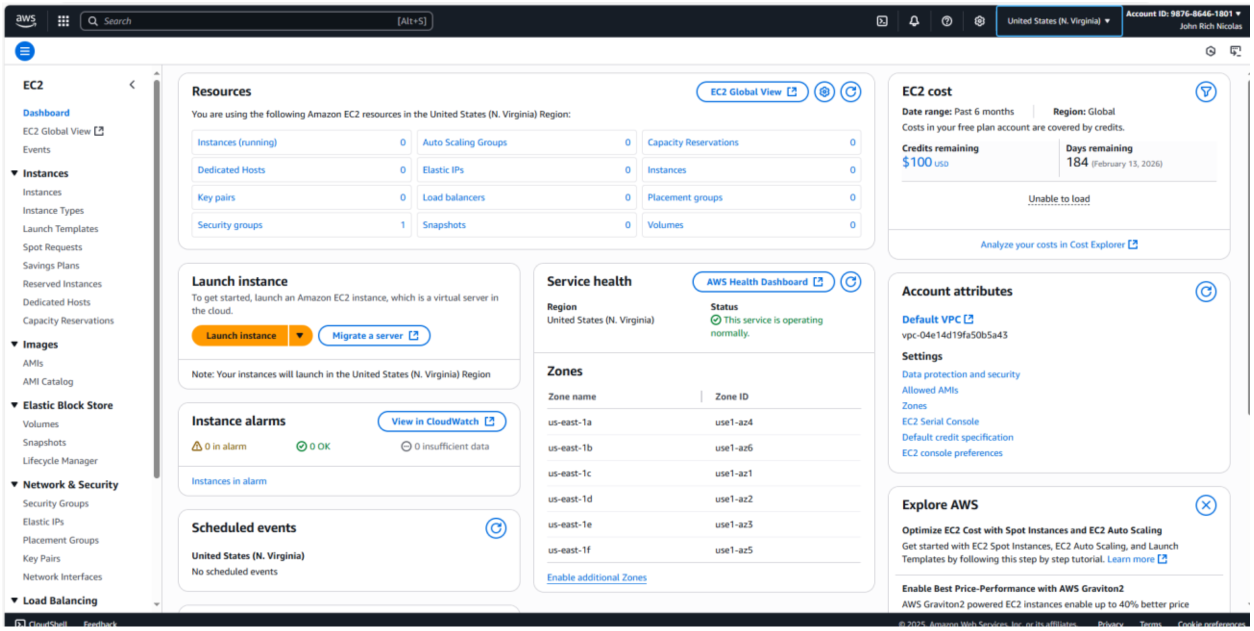
John Rich Nicolas | CLOUD ARCHITECTURE
DAY 1 | AWS

CREATING A KEY PAIR

STEP 1 & STEP 2


Search [Alt+S] United States (N. Virginia) ▾ Account ID: 9876-8646-1801 ▾ John Rich Nicolas

EC2
Dashboard
EC2 Global View
Events
▾ Instances
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
Security Groups
Elastic IPs
Placement Groups
Key Pairs
Network Interfaces
▾ Load Balancing

Resources EC2 Global View

You are using the following Amazon EC2 resources in the United States (N. Virginia) Region:

Instances (running)	0	Auto Scaling Groups	0	Capacity Reservations	0
Dedicated Hosts	0	Elastic IPs	0	Instances	0
Key pairs	0	Load balancers	0	Placement groups	0
Security groups	1	Snapshots	0	Volumes	0

EC2 cost
Date range: Past 6 months Region: Global
Costs in your free plan account are covered by credits.
Credits remaining $100 USD
Days remaining 184 (February 13, 2026)
Unable to load
Analyze your costs in Cost Explorer

Launch instance
To get started, launch an Amazon EC2 instance, which is a virtual server in the cloud.
Launch instance ▾ Migrate a server
Note: Your instances will launch in the United States (N. Virginia) Region

Service health AWS Health Dashboard
Region: United States (N. Virginia)
Status: This service is operating normally.

Zones

Zone name	Zone ID
us-east-1a	use1-az4
us-east-1b	use1-az6
us-east-1c	use1-az1
us-east-1d	use1-az2
us-east-1e	use1-az3
us-east-1f	use1-az5

Enable additional Zones

Account attributes
Default VPC
vpc-04e14d19fa50b5a43
Settings
Data protection and security
Allowed AMIs
Zones
EC2 Serial Console
Default credit specification
EC2 console preferences

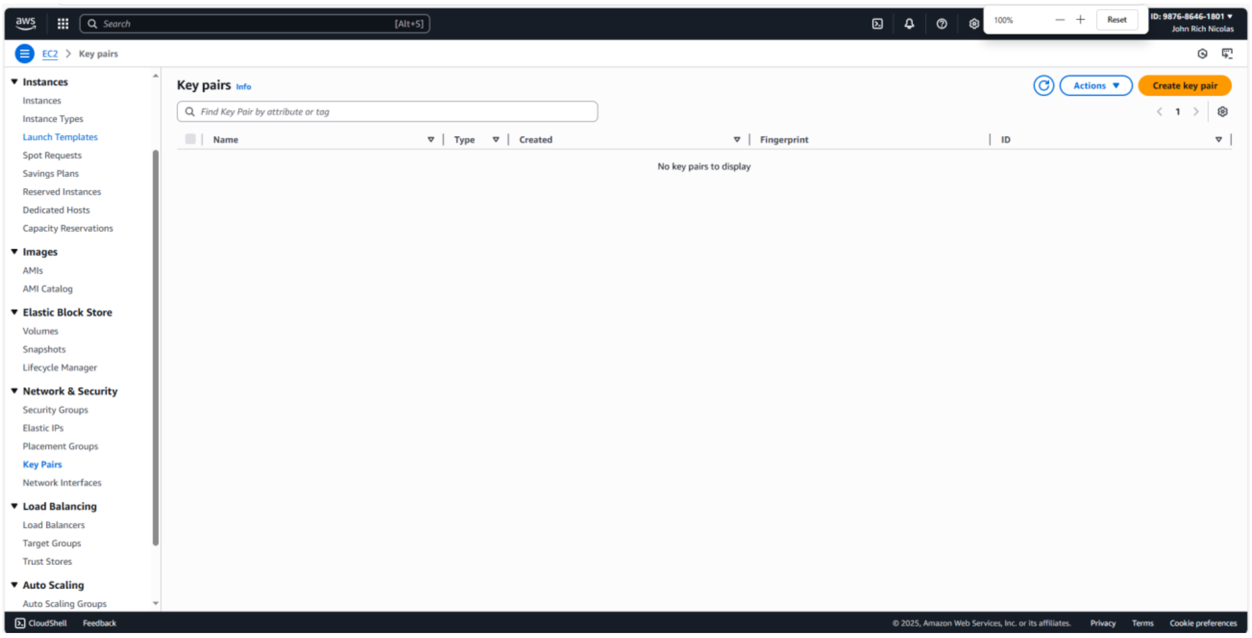
Instance alarms View in CloudWatch
0 in alarm 0 OK 0 insufficient data
Instances in alarm

Scheduled events
United States (N. Virginia)
No scheduled events

Explore AWS
Optimize EC2 Cost with Spot Instances and EC2 Auto Scaling
Get started with EC2 Spot Instances, EC2 Auto Scaling, and Launch Templates by following this step by step tutorial. Learn more
Enable Best Price-Performance with AWS Graviton2
AWS Graviton2 powered EC2 instances enable up to 40% better price

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

STEP 3


Search [Alt+S] 100% — + Reset ID: 9876-8646-1801 ▾ John Rich Nicolas

EC2 > Key pairs

▾ Instances
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
Security Groups
Elastic IPs
Placement Groups
Key Pairs
Network Interfaces
▾ Load Balancing
Load Balancers
Target Groups
Trust Stores
▾ Auto Scaling
Auto Scaling Groups

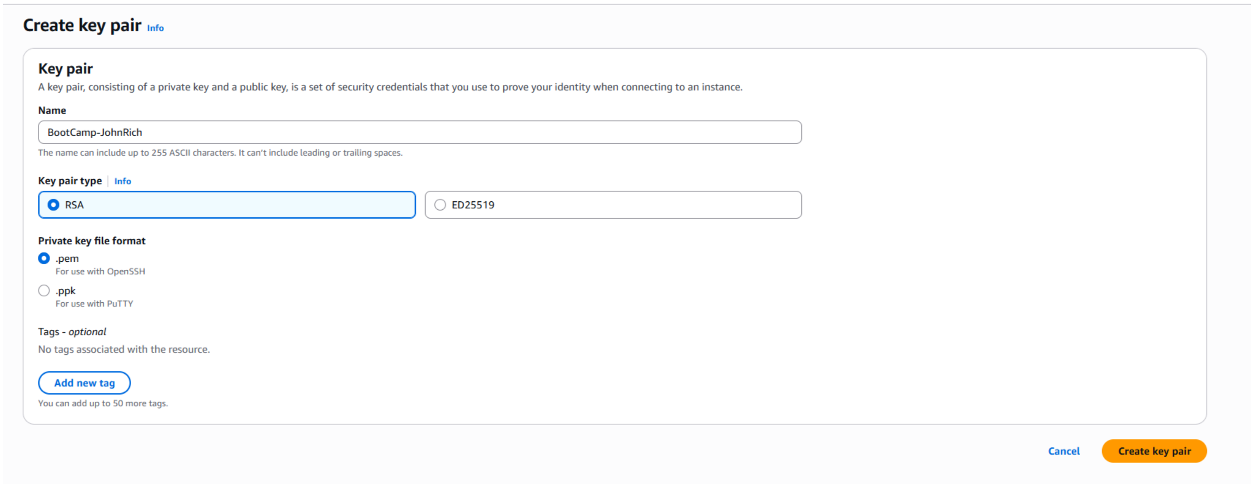
Key pairs Info Actions ▾ Create key pair
Find Key Pair by attribute or tag

Name	Type	Created	Fingerprint	ID

No key pairs to display

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

STEP 4 -8



Create key pair Info

Key pair

A key pair, consisting of a private key and a public key, is a set of security credentials that you use to prove your identity when connecting to an instance.

Name: BootCamp-JohnRich

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

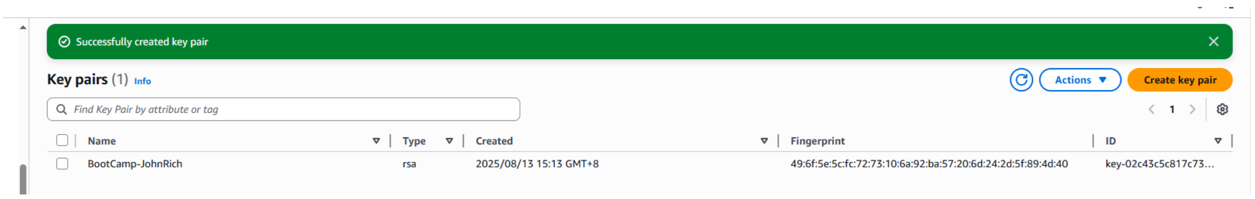
Key pair type: [✓ RSA] ED25519

Private key file format:
[✓ .pem] For use with OpenSSH
.ppk For use with PuTTY

Tags - *optional*

No tags associated with the resource.

You can add up to 50 more tags.



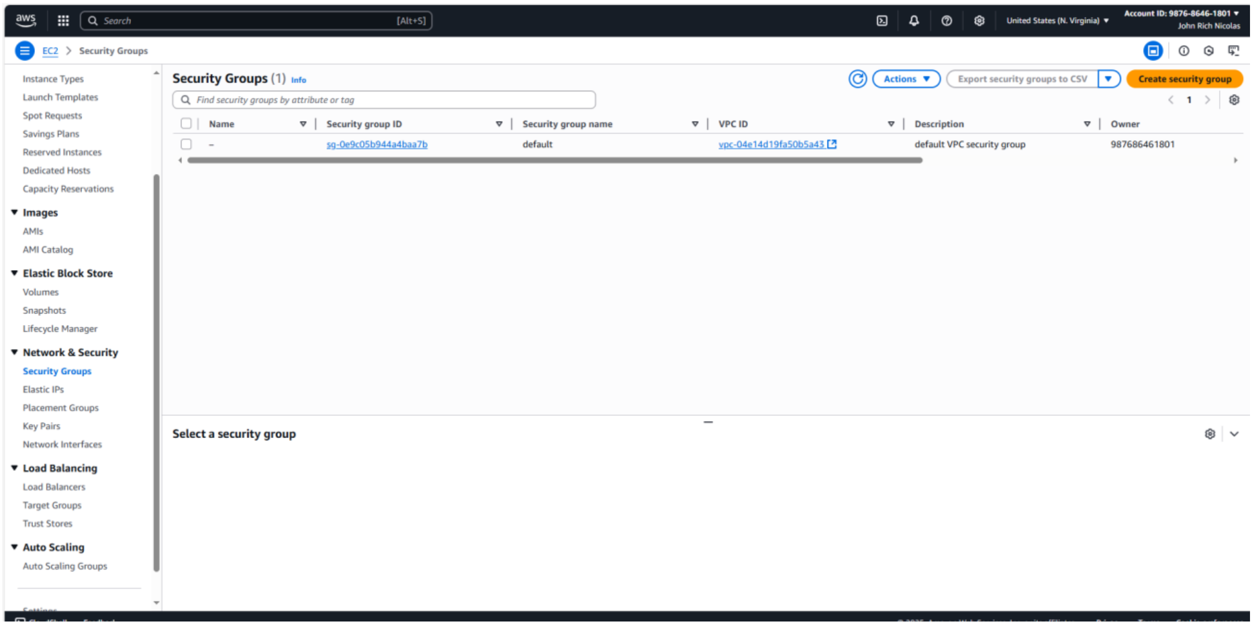
Successfully created key pair

Key pairs (1) Info

Name	Type	Created	Fingerprint	ID
BootCamp-JohnRich	rsa	2025/08/13 15:13 GMT+8	49:6f:5e:5c:fc:72:73:10:6a:92:ba:57:20:6d:24:2d:5f:89:4d:40	key-02c43c5c817c73...

Create a Security Group (Linux and Windows)

STEP 1 – 4

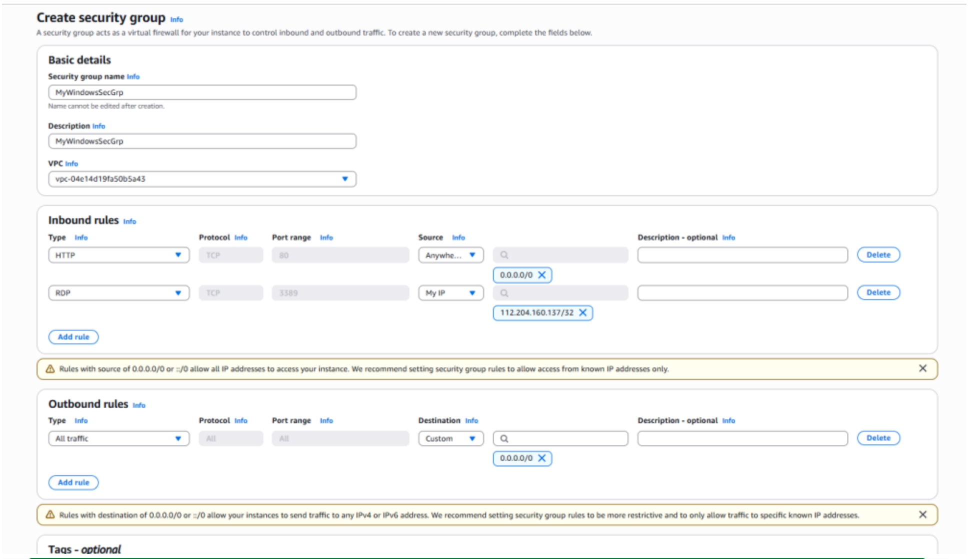


Security Groups (1) Info

Name	Security group ID	Security group name	VPC ID	Description	Owner
–	sg-0e9c05b944a4baa7b	default	vpc-04e14d19fa50b5a43	default VPC security group	987686461801

Select a security group

RESULTS WINDOWS



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: MyWindowsSecGrp
Name cannot be edited after creation.

Description: MyWindowsSecGrp

VPC: vpc-04e14d19fa50b5a43

Inbound rules

Type	Protocol	Port range	Source		Description - optional
HTTP	TCP	80	Anywhe...	0.0.0.0/0	
RDP	TCP	3389	My IP	112.204.160.137/32	

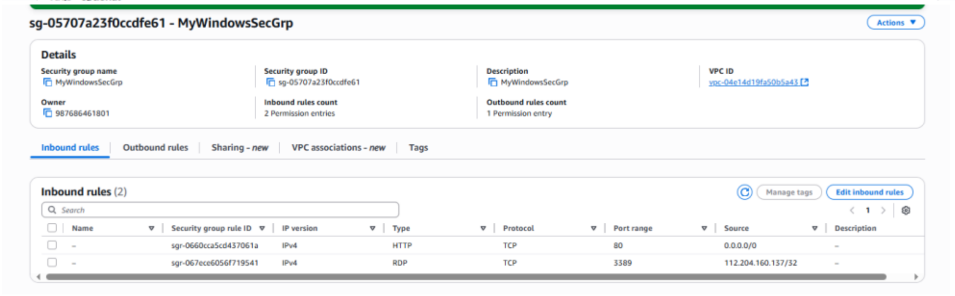
Rules with source of 0.0.0.0/0 or ::/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.

Outbound rules

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

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*



sg-05707a23f0ccdfe61 - MyWindowsSecGrp

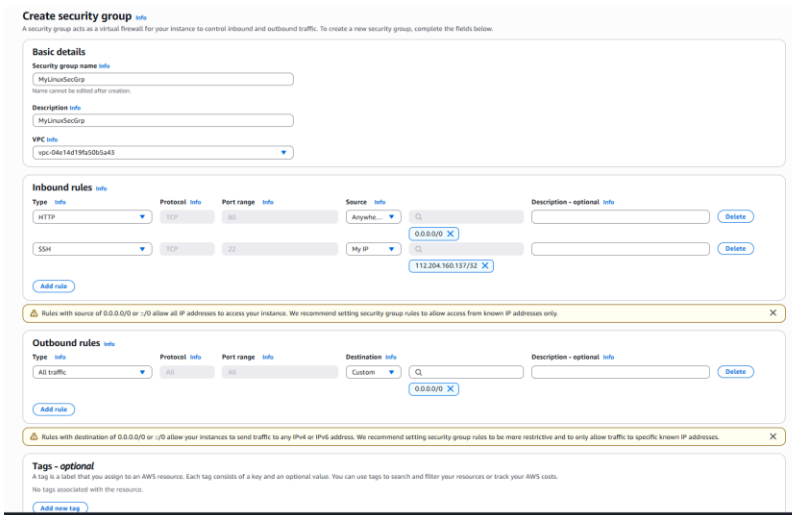
Details

Security group name: MyWindowsSecGrp	Security group ID: sg-05707a23f0ccdfe61	Description: MyWindowsSecGrp	VPC ID: vpc-04e14d19fa50b5a43
Owner: 987686461801	Inbound rules count: 2 Permission entries	Outbound rules count: 1 Permission entry	

Inbound rules (2)

Name	Security group rule ID	IP version	Type	Protocol	Port range	Source	Description
–	sgr-0660cca5cd437061a	IPv4	HTTP	TCP	80	0.0.0.0/0	–
–	sgr-067ece6056f719541	IPv4	RDP	TCP	3389	112.204.160.137/32	–

RESULTS LINUX



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: MyLinuxSecGrp
Name cannot be edited after creation.

Description: MyLinuxSecGrp

VPC: vpc-04e14d19fa50b5a43

Inbound rules

Type	Protocol	Port range	Source		Description - optional
HTTP	TCP	80	Anywhe...	0.0.0.0/0	
SSH	TCP	22	My IP	112.204.160.137/32	

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

Outbound rules

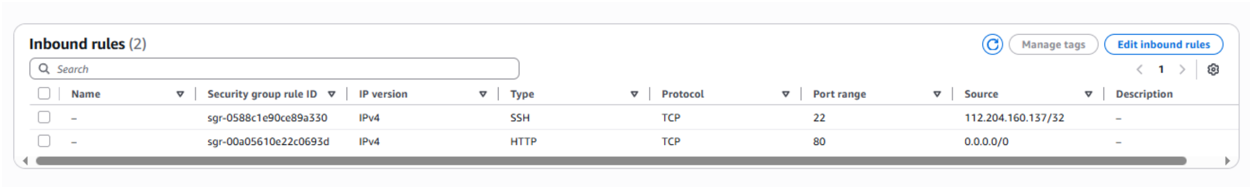
Type	Protocol	Port range	Destination		Description - optional
All traffic	All	All	Custom	0.0.0.0/0	

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.

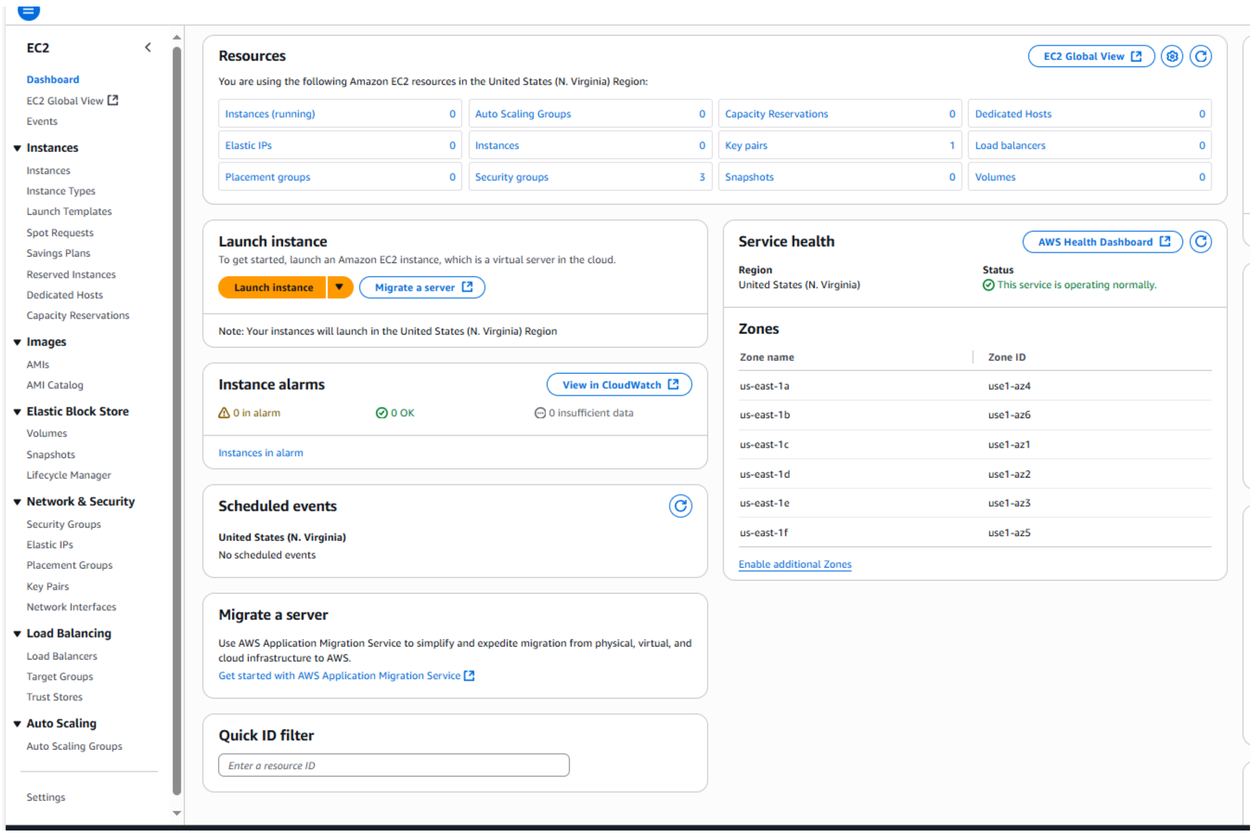


Inbound rules (2)

Name	Security group rule ID	IP version	Type	Protocol	Port range	Source	Description
–	sgr-0588c1e90ce89a330	IPv4	SSH	TCP	22	112.204.160.137/32	–
–	sgr-00a05610e22c0693d	IPv4	HTTP	TCP	80	0.0.0.0/0	–

John Rich Nicolas CLOUD ARCHITECTURE
DAY 1 AWS

LAUNCHING AN INSTANCE – LINUX



Resources

You are using the following Amazon EC2 resources in the United States (N. Virginia) Region:

Instances (running)	0	Auto Scaling Groups	0	Capacity Reservations	0	Dedicated Hosts	0
Elastic IPs	0	Instances	0	Key pairs	1	Load balancers	0
Placement groups	0	Security groups	3	Snapshots	0	Volumes	0

Launch instance

To get started, launch an Amazon EC2 instance, which is a virtual server in the cloud.

Note: Your instances will launch in the United States (N. Virginia) Region

Instance alarms

0 in alarm 0 OK 0 insufficient data

Instances in alarm

Scheduled events

United States (N. Virginia)
No scheduled events

Migrate a server

Use AWS Application Migration Service to simplify and expedite migration from physical, virtual, and cloud infrastructure to AWS.
Get started with AWS Application Migration Service

Quick ID filter

Service health

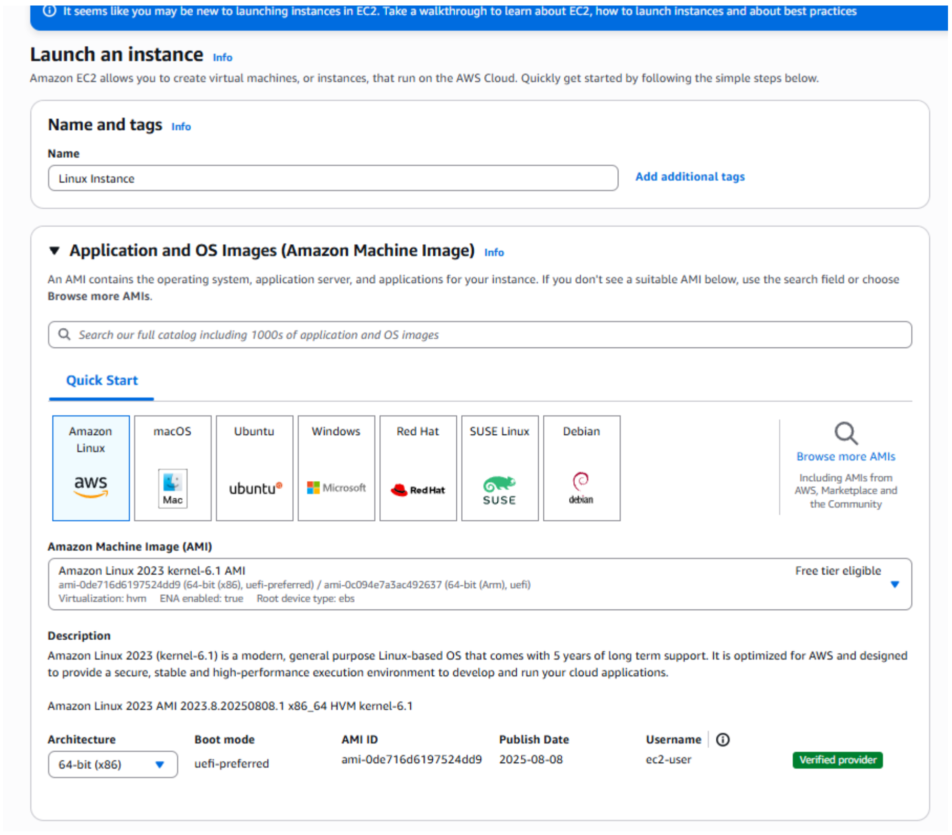
Region: United States (N. Virginia)
Status: This service is operating normally.

Zones

Zone name	Zone ID
us-east-1a	use1-az4
us-east-1b	use1-az6
us-east-1c	use1-az1
us-east-1d	use1-az2
us-east-1e	use1-az3
us-east-1f	use1-az5

Enable additional Zones

STEP 2



ⓘ It seems like you may be new to launching instances in EC2. Take a walkthrough to learn about EC2, how to launch instances and about best practices

Launch an instance Info

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 Info

Name

Linux Instance

Add additional tags

▼ 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**.

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

Quick Start

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)

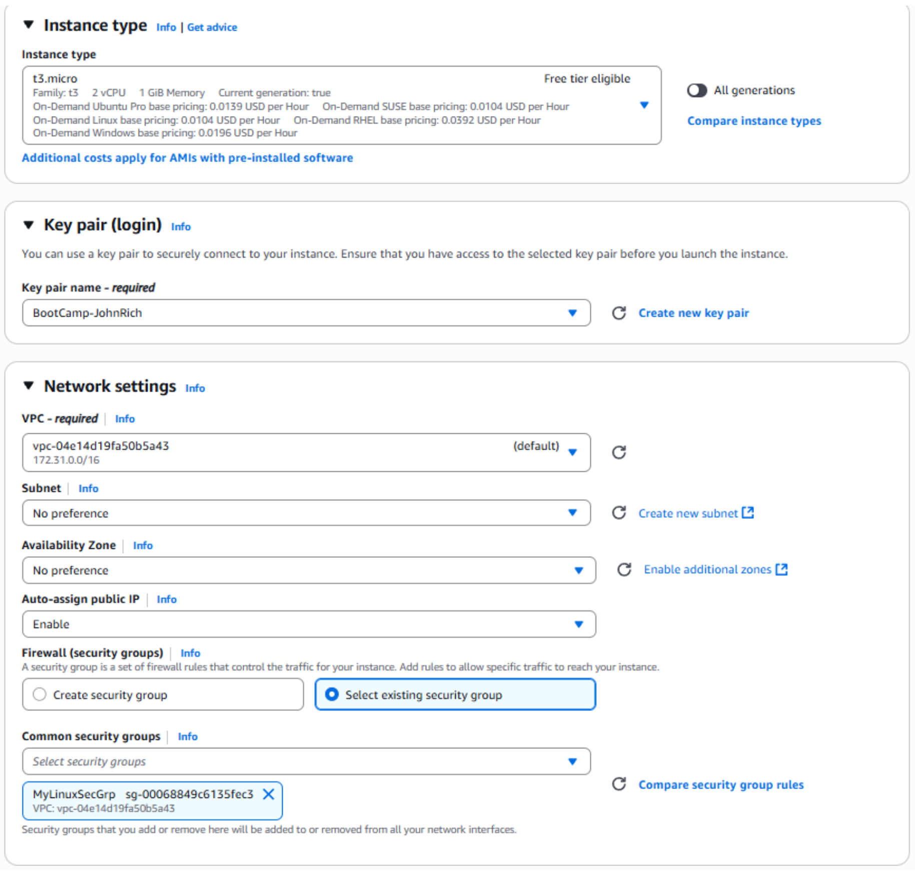
Amazon Linux 2023 kernel-6.1 AMI — Free tier eligible
ami-0de716d6197524dd9 (64-bit (x86), uefi-preferred) / ami-0c094e7a3ac492637 (64-bit (Arm), uefi)
Virtualization: hvm   ENA enabled: true   Root device type: ebs

Description

Amazon Linux 2023 (kernel-6.1) 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 2023.8.20250808.1 x86_64 HVM kernel-6.1

Architecture	Boot mode	AMI ID	Publish Date	Username	
64-bit (x86)	uefi-preferred	ami-0de716d6197524dd9	2025-08-08	ec2-user	Verified provider



▼ Instance type Info | Get advice

Instance type

t3.micro — Free tier eligible
Family: t3   2 vCPU   1 GiB Memory   Current generation: true
On-Demand Ubuntu Pro base pricing: 0.0139 USD per Hour   On-Demand SUSE base pricing: 0.0104 USD per Hour
On-Demand Linux base pricing: 0.0104 USD per Hour   On-Demand RHEL base pricing: 0.0392 USD per Hour
On-Demand Windows base pricing: 0.0196 USD per Hour

All generations

Compare instance types

Additional costs apply for AMIs with pre-installed software

▼ 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*

BootCamp-JohnRich

Create new key pair

▼ Network settings Info

VPC - *required* | Info

vpc-04e14d19fa50b5a43 (default)
172.31.0.0/16

Subnet | Info

No preference

Create new subnet

Availability Zone | Info

No preference

Enable additional zones

Auto-assign public IP | Info

Enable

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]

Common security groups | Info

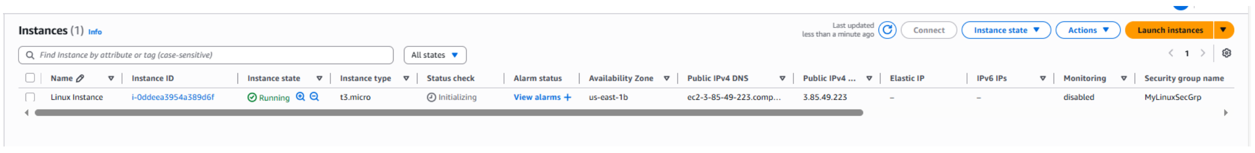
Select security groups

MyLinuxSecGrp sg-00068849c6135fec3
VPC: vpc-04e14d19fa50b5a43

Compare security group rules

Security groups that you add or remove here will be added to or removed from all your network interfaces.

INSTANCE CONFIMATION

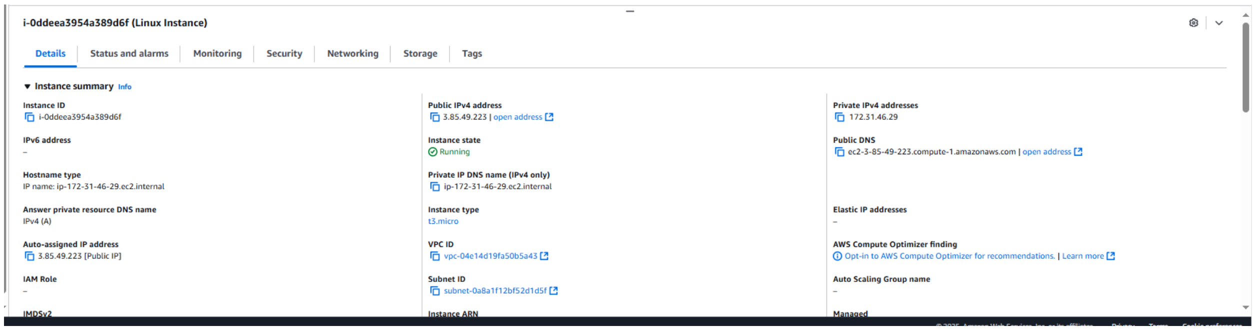


Instances (1) Info

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 IPv4 DNS	Public IPv4 ...	Elastic IP	IPv6 IPs	Monitoring	Security group name
Linux Instance	i-0ddeea3954a389d6f	Running	t3.micro	Initializing	View alarms +	us-east-1b	ec2-3-85-49-223.comp...	3.85.49.223	–	–	disabled	MyLinuxSecGrp



i-0ddeea3954a389d6f (Linux Instance)

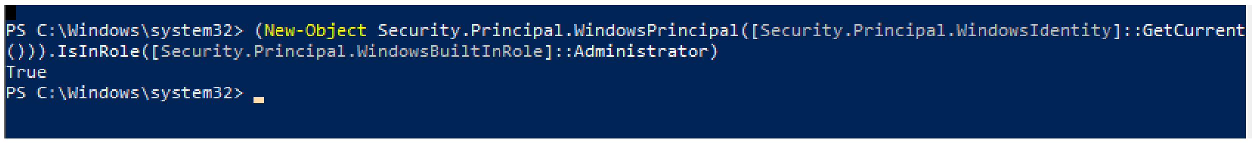
Details | Status and alarms | Monitoring | Security | Networking | Storage | Tags

Instance summary Info

Instance ID i-0ddeea3954a389d6f	**Public IPv4 address** 3.85.49.223 | open address	**Private IPv4 addresses** 172.31.46.29
IPv6 address –	**Instance state** Running	**Public DNS** ec2-3-85-49-223.compute-1.amazonaws.com | open address
Hostname type IP name: ip-172-31-46-29.ec2.internal	**Private IP DNS name (IPv4 only)** ip-172-31-46-29.ec2.internal	
Answer private resource DNS name IPv4 (A)	**Instance type** t3.micro	**Elastic IP addresses** –
Auto-assigned IP address 3.85.49.223 [Public IP]	**VPC ID** vpc-04e14d19fa50b5a43	**AWS Compute Optimizer finding** Opt-in to AWS Compute Optimizer for recommendations. | Learn more
IAM Role –	**Subnet ID** subnet-0a8a1f12bf52d1d5f	**Auto Scaling Group name** –
IMDSv2	**Instance ARN**	**Managed**

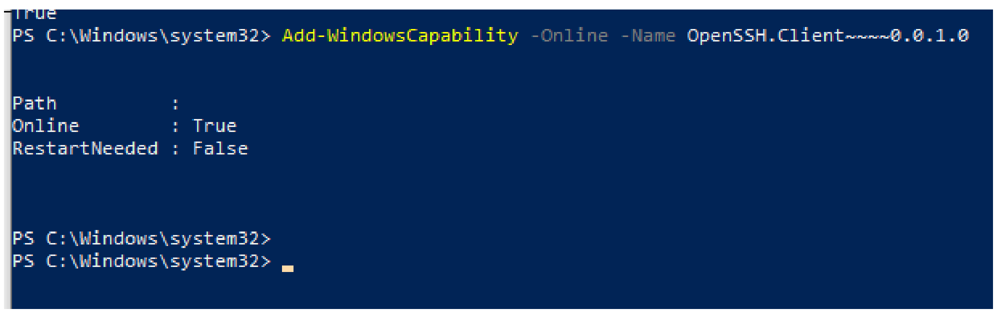
Connect to your Linux instance from Windows using OpenSSH

STEP 1 – CONFIRMATION



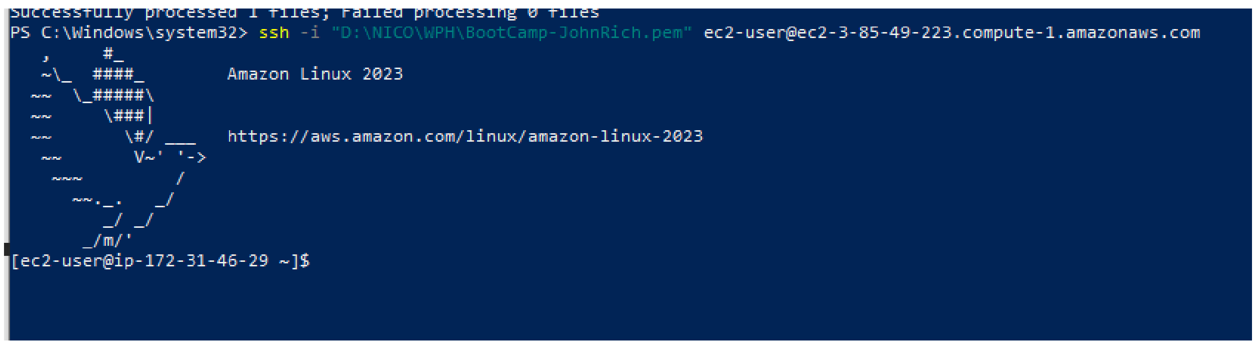
```
PS C:\Windows\system32> (New-Object Security.Principal.WindowsPrincipal([Security.Principal.WindowsIdentity]::GetCurrent
())).IsInRole([Security.Principal.WindowsBuiltInRole]::Administrator)
True
PS C:\Windows\system32> _
```

INSTALLATION OF OPENSSH



```
True
PS C:\Windows\system32> Add-WindowsCapability -Online -Name OpenSSH.Client~~~~0.0.1.0


Path          :
Online        : True
RestartNeeded : False



PS C:\Windows\system32>
PS C:\Windows\system32> _
```

SSH COMMAND

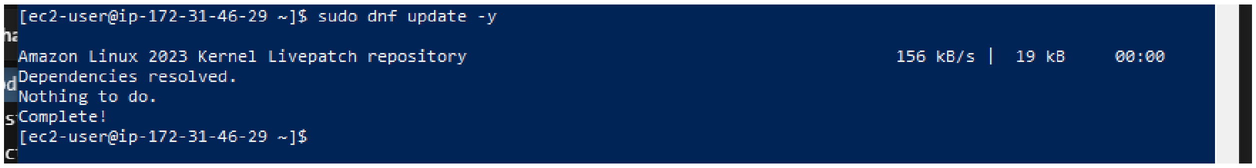


```
Successfully processed 1 files; Failed processing 0 files
PS C:\Windows\system32> ssh -i "D:\NICO\WPH\BootCamp-JohnRich.pem" ec2-user@ec2-3-85-49-223.compute-1.amazonaws.com
   ,     #_
   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
[ec2-user@ip-172-31-46-29 ~]$
```

John Rich Nicolas
DAY 1

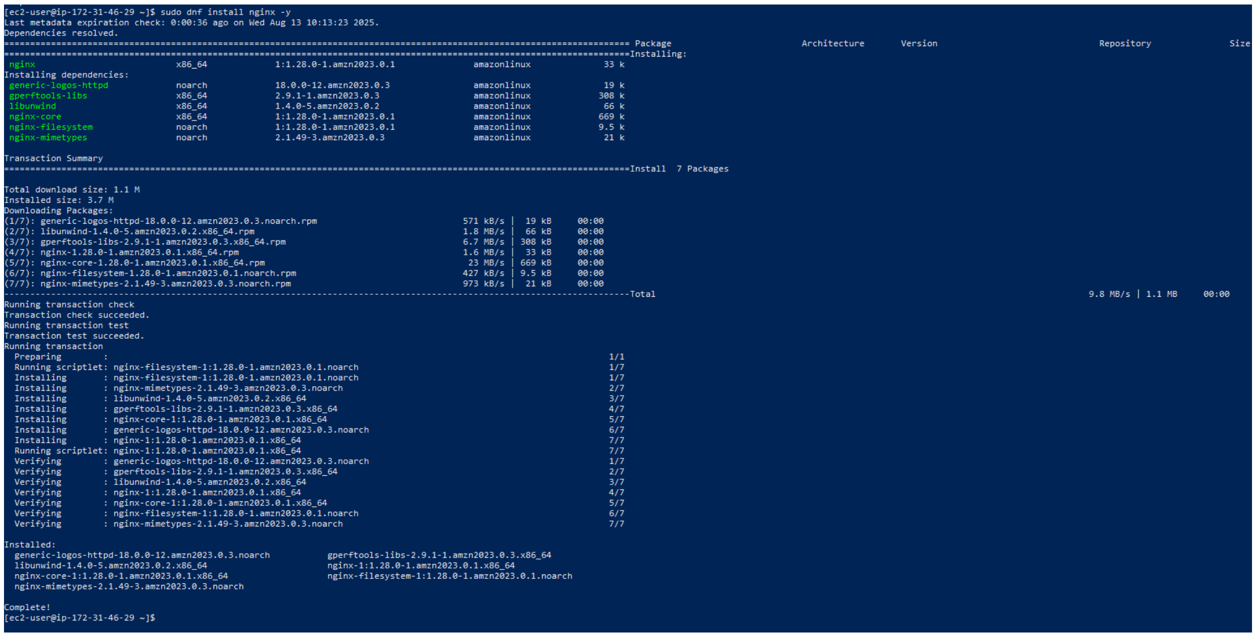
CLOUD ARCHITECTURE
AWS

SUDO DNF UPDATE -Y



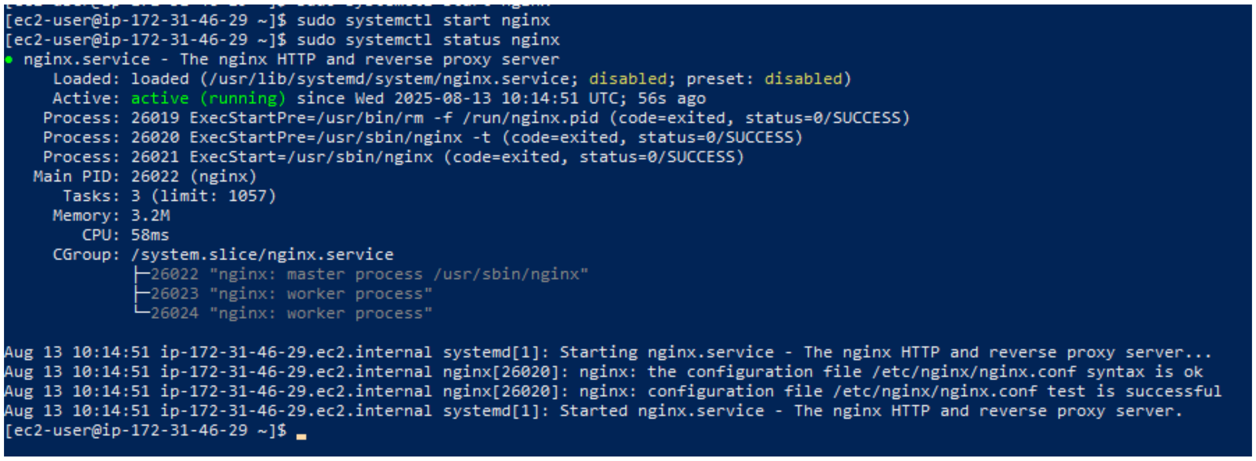
```
[ec2-user@ip-172-31-46-29 ~]$ sudo dnf update -y

Amazon Linux 2023 Kernel Livepatch repository                               156 kB/s |  19 kB     00:00
Dependencies resolved.
Nothing to do.
Complete!
[ec2-user@ip-172-31-46-29 ~]$
```

sudo dnf install nginx -y



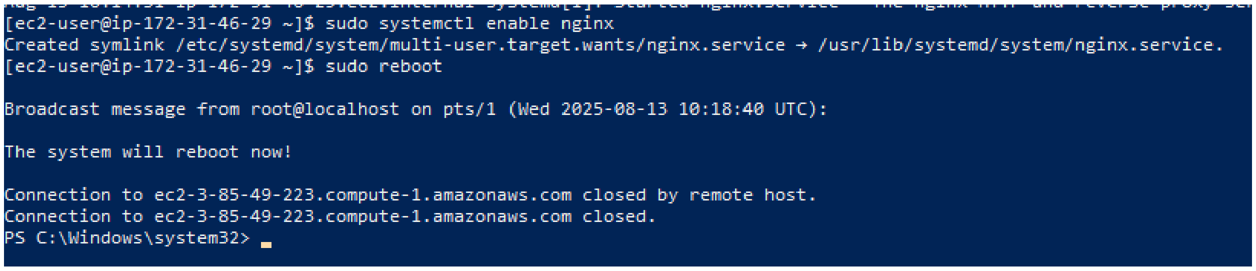
```
[ec2-user@ip-172-31-46-29 ~]$ sudo dnf install nginx -y
Last metadata expiration check: 0:00:36 ago on Wed Aug 13 10:13:23 2025.
Dependencies resolved.
======================================================================================================================== Package                        Architecture       Version                               Repository              Size
========================================================================================================================Installing:
 nginx                          x86_64             1:1.28.0-1.amzn2023.0.1               amazonlinux              33 k
Installing dependencies:
 generic-logos-httpd            noarch             18.0.0-12.amzn2023.0.3                amazonlinux              19 k
 gperftools-libs                x86_64             2.9.1-1.amzn2023.0.3                  amazonlinux             308 k
 libunwind                      x86_64             1.4.0-5.amzn2023.0.2                  amazonlinux              66 k
 nginx-core                     x86_64             1:1.28.0-1.amzn2023.0.1               amazonlinux             669 k
 nginx-filesystem               noarch             1:1.28.0-1.amzn2023.0.1               amazonlinux             9.5 k
 nginx-mimetypes                noarch             2.1.49-3.amzn2023.0.3                 amazonlinux              21 k

Transaction Summary
========================================================================================================================Install  7 Packages

Total download size: 1.1 M
Installed size: 3.7 M
Downloading Packages:
(1/7): generic-logos-httpd-18.0.0-12.amzn2023.0.3.noarch.rpm                           571 kB/s |  19 kB     00:00
(2/7): libunwind-1.4.0-5.amzn2023.0.2.x86_64.rpm                                       1.8 MB/s |  66 kB     00:00
(3/7): gperftools-libs-2.9.1-1.amzn2023.0.3.x86_64.rpm                                 6.7 MB/s | 308 kB     00:00
(4/7): nginx-1.28.0-1.amzn2023.0.1.x86_64.rpm                                          1.6 MB/s |  33 kB     00:00
(5/7): nginx-core-1.28.0-1.amzn2023.0.1.x86_64.rpm                                      23 MB/s | 669 kB     00:00
(6/7): nginx-filesystem-1.28.0-1.amzn2023.0.1.noarch.rpm                               427 kB/s | 9.5 kB     00:00
(7/7): nginx-mimetypes-2.1.49-3.amzn2023.0.3.noarch.rpm                                973 kB/s |  21 kB     00:00
------------------------------------------------------------------------------------------------------------------------Total                                                  9.8 MB/s | 1.1 MB     00:00
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
  Preparing        :                                                                                                 1/1
  Running scriptlet: nginx-filesystem-1:1.28.0-1.amzn2023.0.1.noarch                                                 1/7
  Installing       : nginx-filesystem-1:1.28.0-1.amzn2023.0.1.noarch                                                 1/7
  Installing       : nginx-mimetypes-2.1.49-3.amzn2023.0.3.noarch                                                    2/7
  Installing       : libunwind-1.4.0-5.amzn2023.0.2.x86_64                                                           3/7
  Installing       : gperftools-libs-2.9.1-1.amzn2023.0.3.x86_64                                                     4/7
  Installing       : nginx-core-1:1.28.0-1.amzn2023.0.1.x86_64                                                       5/7
  Installing       : generic-logos-httpd-18.0.0-12.amzn2023.0.3.noarch                                               6/7
  Installing       : nginx-1:1.28.0-1.amzn2023.0.1.x86_64                                                            7/7
  Running scriptlet: nginx-1:1.28.0-1.amzn2023.0.1.x86_64                                                            7/7
  Verifying        : generic-logos-httpd-18.0.0-12.amzn2023.0.3.noarch                                               1/7
  Verifying        : gperftools-libs-2.9.1-1.amzn2023.0.3.x86_64                                                     2/7
  Verifying        : libunwind-1.4.0-5.amzn2023.0.2.x86_64                                                           3/7
  Verifying        : nginx-1:1.28.0-1.amzn2023.0.1.x86_64                                                            4/7
  Verifying        : nginx-core-1:1.28.0-1.amzn2023.0.1.x86_64                                                       5/7
  Verifying        : nginx-filesystem-1:1.28.0-1.amzn2023.0.1.noarch                                                 6/7
  Verifying        : nginx-mimetypes-2.1.49-3.amzn2023.0.3.noarch                                                    7/7

Installed:
  generic-logos-httpd-18.0.0-12.amzn2023.0.3.noarch          gperftools-libs-2.9.1-1.amzn2023.0.3.x86_64
  libunwind-1.4.0-5.amzn2023.0.2.x86_64                      nginx-1:1.28.0-1.amzn2023.0.1.x86_64
  nginx-core-1:1.28.0-1.amzn2023.0.1.x86_64                  nginx-filesystem-1:1.28.0-1.amzn2023.0.1.noarch
  nginx-mimetypes-2.1.49-3.amzn2023.0.3.noarch

Complete!
[ec2-user@ip-172-31-46-29 ~]$
```

sudo systemctl start nginx & sudo systemctl status nginx



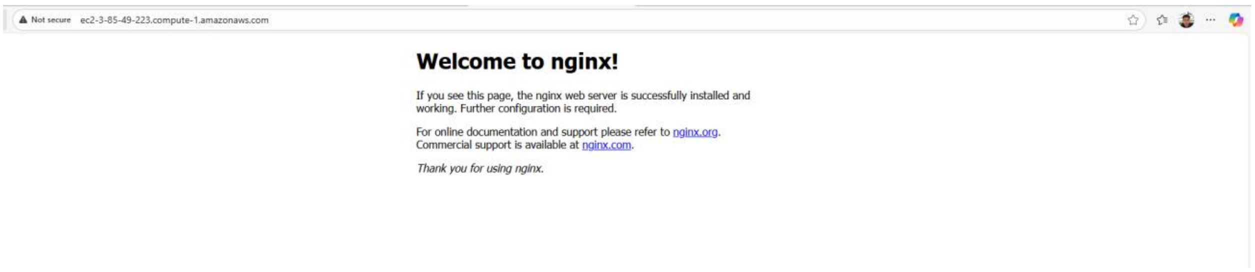
```
[ec2-user@ip-172-31-46-29 ~]$ sudo systemctl start nginx
[ec2-user@ip-172-31-46-29 ~]$ sudo systemctl status nginx
● nginx.service - The nginx HTTP and reverse proxy server
     Loaded: loaded (/usr/lib/systemd/system/nginx.service; disabled; preset: disabled)
     Active: active (running) since Wed 2025-08-13 10:14:51 UTC; 56s ago
    Process: 26019 ExecStartPre=/usr/bin/rm -f /run/nginx.pid (code=exited, status=0/SUCCESS)
    Process: 26020 ExecStartPre=/usr/sbin/nginx -t (code=exited, status=0/SUCCESS)
    Process: 26021 ExecStart=/usr/sbin/nginx (code=exited, status=0/SUCCESS)
   Main PID: 26022 (nginx)
      Tasks: 3 (limit: 1057)
     Memory: 3.2M
        CPU: 58ms
     CGroup: /system.slice/nginx.service
             ├─26022 "nginx: master process /usr/sbin/nginx"
             ├─26023 "nginx: worker process"
             └─26024 "nginx: worker process"

Aug 13 10:14:51 ip-172-31-46-29.ec2.internal systemd[1]: Starting nginx.service - The nginx HTTP and reverse proxy server...
Aug 13 10:14:51 ip-172-31-46-29.ec2.internal nginx[26020]: nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
Aug 13 10:14:51 ip-172-31-46-29.ec2.internal nginx[26020]: nginx: configuration file /etc/nginx/nginx.conf test is successful
Aug 13 10:14:51 ip-172-31-46-29.ec2.internal systemd[1]: Started nginx.service - The nginx HTTP and reverse proxy server.
[ec2-user@ip-172-31-46-29 ~]$ _
```

sudo systemctl enable nginx & sudo reboot



```
[ec2-user@ip-172-31-46-29 ~]$ sudo systemctl enable nginx
Created symlink /etc/systemd/system/multi-user.target.wants/nginx.service → /usr/lib/systemd/system/nginx.service.
[ec2-user@ip-172-31-46-29 ~]$ sudo reboot

Broadcast message from root@localhost on pts/1 (Wed 2025-08-13 10:18:40 UTC):

The system will reboot now!

Connection to ec2-3-85-49-223.compute-1.amazonaws.com closed by remote host.
Connection to ec2-3-85-49-223.compute-1.amazonaws.com closed.
PS C:\Windows\system32>
```

CHECK IF ITS RUNNING



Not secure   ec2-3-85-49-223.compute-1.amazonaws.com

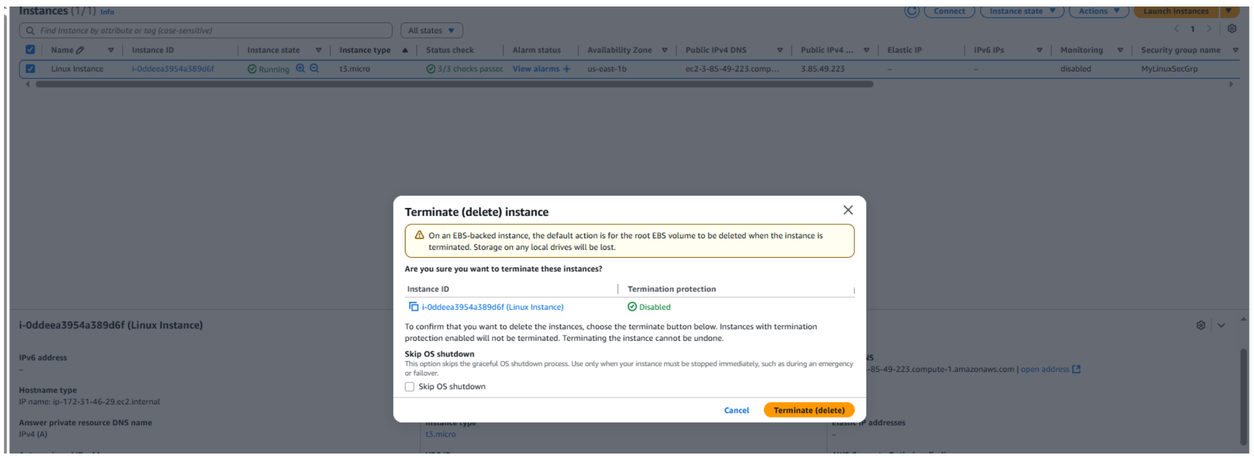
Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.

CLEAN UP INSTANCE



Instances (1/1) Info

Find Instance by attribute or tag (case-sensitive)

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 ...	Elastic IP	IPv6 IPs	Monitoring	Security group name
Linux Instance	i-0ddeea3954a389d6f	Running	t3.micro	3/3 checks passec	View alarms +	us-east-1b	ec2-3-85-49-223.comp...	3.85.49.223	–	–	disabled	MyLinuxSecGrp

i-0ddeea3954a389d6f (Linux Instance)

IPv6 address
–

Hostname type
IP name: ip-172-31-46-29.ec2.internal

Answer private resource DNS name
IPv4 (A)

Terminate (delete) instance

On an EBS-backed instance, the default action is for the root EBS volume to be deleted when the instance is terminated. Storage on any local drives will be lost.

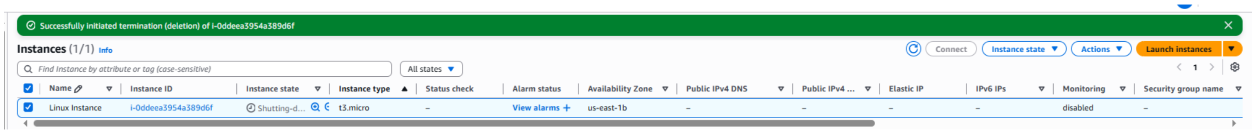
Are you sure you want to terminate these instances?

Instance ID	Termination protection
i-0ddeea3954a389d6f (Linux Instance)	Disabled

To confirm that you want to delete the instances, choose the terminate button below. Instances with termination protection enabled will not be terminated. Terminating the instance cannot be undone.

Skip OS shutdown
This option skips the graceful OS shutdown process. Use only when your instance must be stopped immediately, such as during an emergency or failover.

☐ Skip OS shutdown



Successfully initiated termination (deletion) of i-0ddeea3954a389d6f

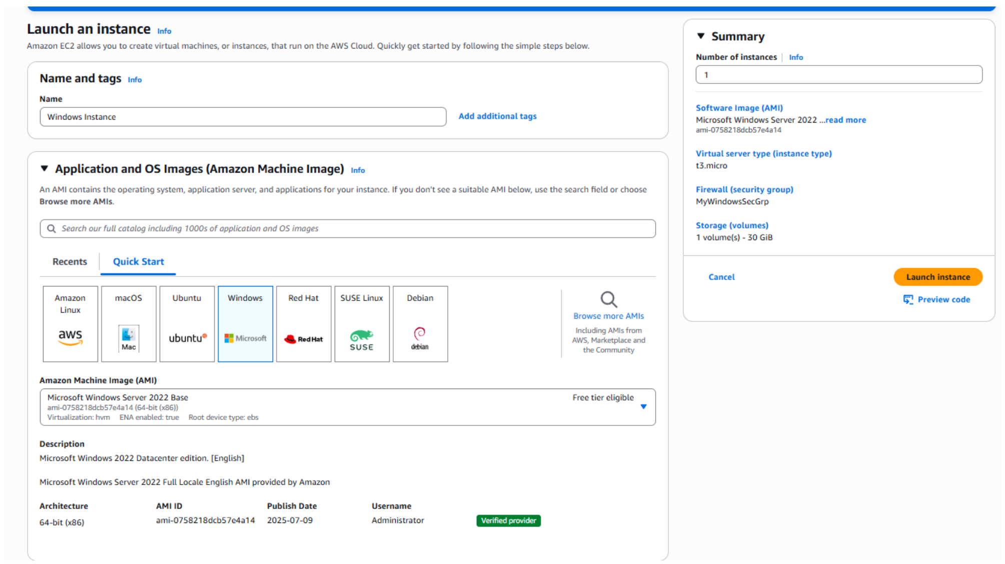
Instances (1/1) Info

Find Instance by attribute or tag (case-sensitive)

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 ...	Elastic IP	IPv6 IPs	Monitoring	Security group name
Linux Instance	i-0ddeea3954a389d6f	Shutting-d...	t3.micro	–	View alarms +	us-east-1b	–	–	–	–	disabled	–

John Rich Nicolas CLOUD ARCHITECTURE
DAY 1 AWS

LAUNCH AN INSTANCE – WINDOWS



Launch an instance Info

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 Info

Name: Windows Instance

Add additional tags

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**.

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

Recents | Quick Start

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)

Microsoft Windows Server 2022 Base — Free tier eligible
ami-0758218dcb57e4a14 (64-bit (x86))
Virtualization: hvm ENA enabled: true Root device type: ebs

Description

Microsoft Windows 2022 Datacenter edition. [English]

Microsoft Windows Server 2022 Full Locale English AMI provided by Amazon

Architecture	AMI ID	Publish Date	Username	
64-bit (x86)	ami-0758218dcb57e4a14	2025-07-09	Administrator	Verified provider

Summary

Number of instances | Info: 1

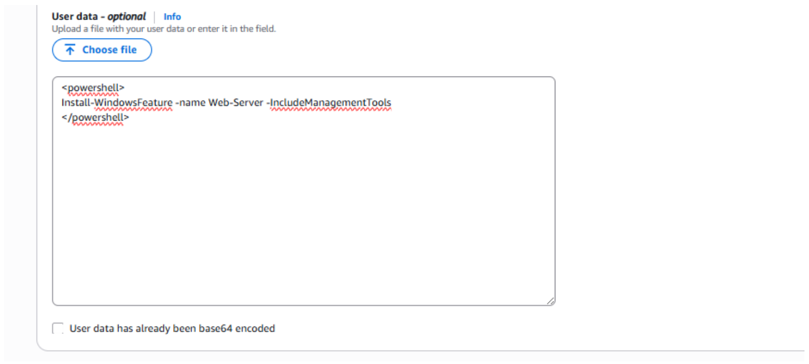
Software Image (AMI)
Microsoft Windows Server 2022 ...read more
ami-0758218dcb57e4a14

Virtual server type (instance type)
t3.micro

Firewall (security group)
MyWindowsSecGrp

Storage (volumes)
1 volume(s) - 30 GiB

Cancel | Launch instance | Preview code



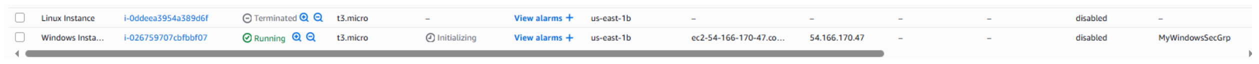
User data - *optional* | Info

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

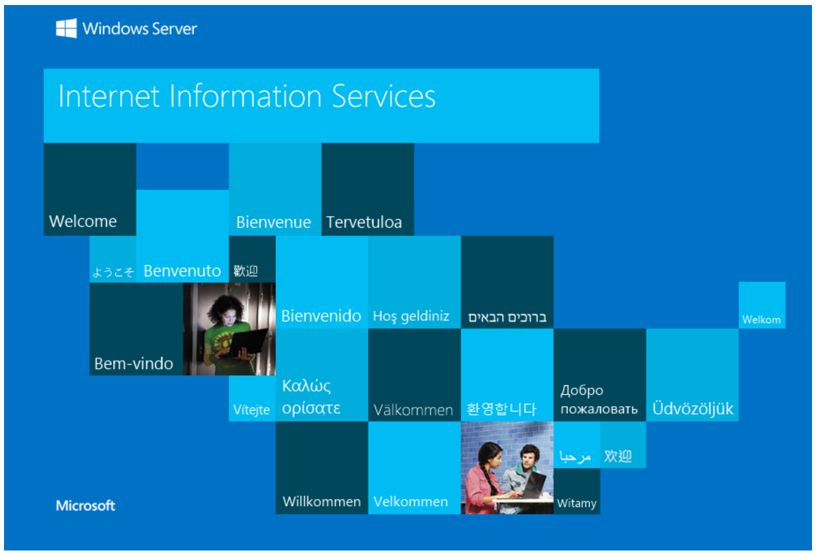
Choose file

```
<powershell>
Install-WindowsFeature -name Web-Server -IncludeManagementTools
</powershell>
```

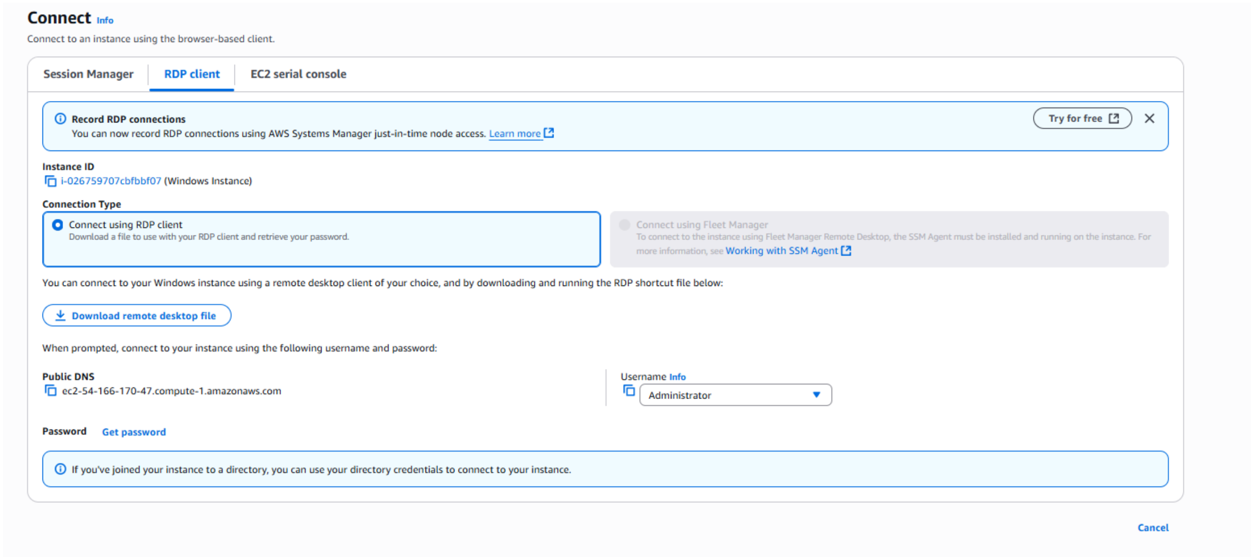
☐ User data has already been base64 encoded



Linux Instance	i-0ddeea3954a389d6f	Terminated	t3.micro	–	View alarms +	us-east-1b	–	–	–	–	disabled	–
Windows Insta...	i-026759707cbfbbf07	Running	t3.micro	Initializing	View alarms +	us-east-1b	ec2-54-166-170-47.co...	54.166.170.47	–	–	disabled	MyWindowsSecGrp



CONNECTING TO RDP


Connect Info

Connect to an instance using the browser-based client.

Record RDP connections
You can now record RDP connections using AWS Systems Manager just-in-time node access. Learn more

Instance ID
i-026759707cbfbbf07 (Windows Instance)

Connection Type

- Connect using RDP client
  Download a file to use with your RDP client and retrieve your password.
- Connect using Fleet Manager
  To connect to the instance using Fleet Manager Remote Desktop, the SSM Agent must be installed and running on the instance. For more information, see Working with SSM Agent

You can connect to your Windows instance using a remote desktop client of your choice, and by downloading and running the RDP shortcut file below:

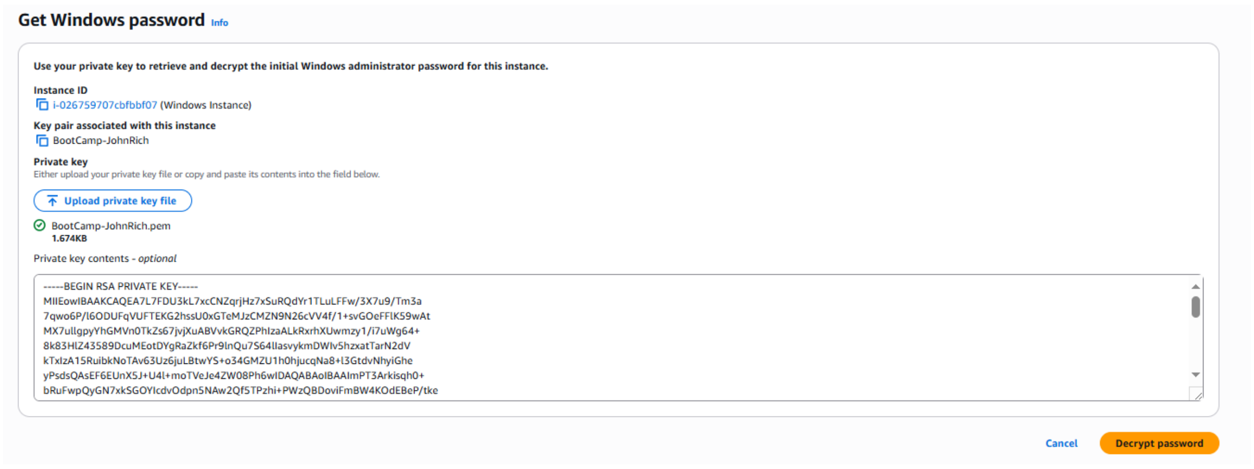
When prompted, connect to your instance using the following username and password:

Public DNS
ec2-54-166-170-47.compute-1.amazonaws.com

Username Info
Administrator

Password Get password

If you've joined your instance to a directory, you can use your directory credentials to connect to your instance.


Get Windows password Info

Use your private key to retrieve and decrypt the initial Windows administrator password for this instance.

Instance ID
i-026759707cbfbbf07 (Windows Instance)

Key pair associated with this instance
BootCamp-JohnRich

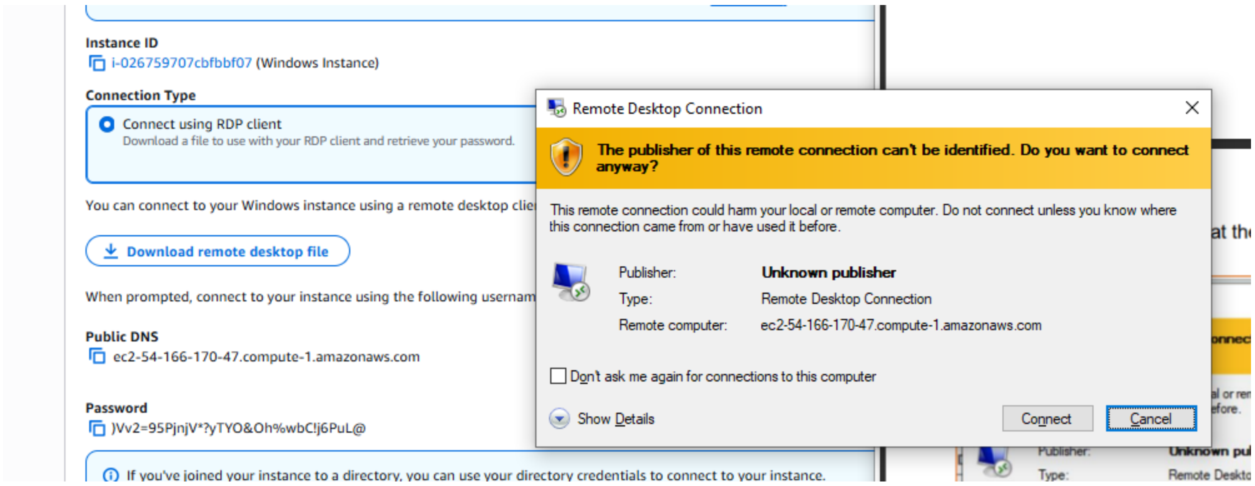
Private key
Either upload your private key file or copy and paste its contents into the field below.

BootCamp-JohnRich.pem
1.674KB

Private key contents - *optional*

```
-----BEGIN RSA PRIVATE KEY-----
MIIEowIBAAKCAQEA7L7FDU3kL7xcCNZqrjHz7xSuRQdYr1TLuLFFw/3X7u9/Tm3a
7qwo6P/l6ODUFqVUFTEKG2hssU0xGTeMJzCMZN9N26cVV4f/1+svGOeFFlK59wAt
MX7ullgpyYhGMVn0TkZs67jvjXuABVvkGRQZPhIzaALkRxrhXUwmzy1/i7uWg64+
8k83HlZ43589DcuMEotDYgRaZkf6Pr9lnQu7S64llasvykmDWIv5hzxatTarN2dV
kTxIzA15RuibkNoTAv63Uz6juLBtwYS+o34GMZU1h0hjucqNa8+l3GtdvNhyiGhe
yPsdsQAsEF6EUnX5J+U4l+moTVeJe4ZW08Ph6wIDAQABAoIBAAImPT3Arkisqh0+
bRuFwpQyGN7xkSGOYIcdvOdpn5NAw2Qf5TPzhi+PWzQBDoviFmBW4KOdEBeP/tke
```

CONNECT


Instance ID
i-026759707cbfbbf07 (Windows Instance)

Connection Type

- Connect using RDP client
  Download a file to use with your RDP client and retrieve your password.

You can connect to your Windows instance using a remote desktop clie

When prompted, connect to your instance using the following usernam

Public DNS
ec2-54-166-170-47.compute-1.amazonaws.com

Password
)Vv2=95PjnjV*?yTYO&Oh%wbC!j6PuL@

If you've joined your instance to a directory, you can use your directory credentials to connect to your instance.

Remote Desktop Connection

The publisher of this remote connection can't be identified. Do you want to connect anyway?

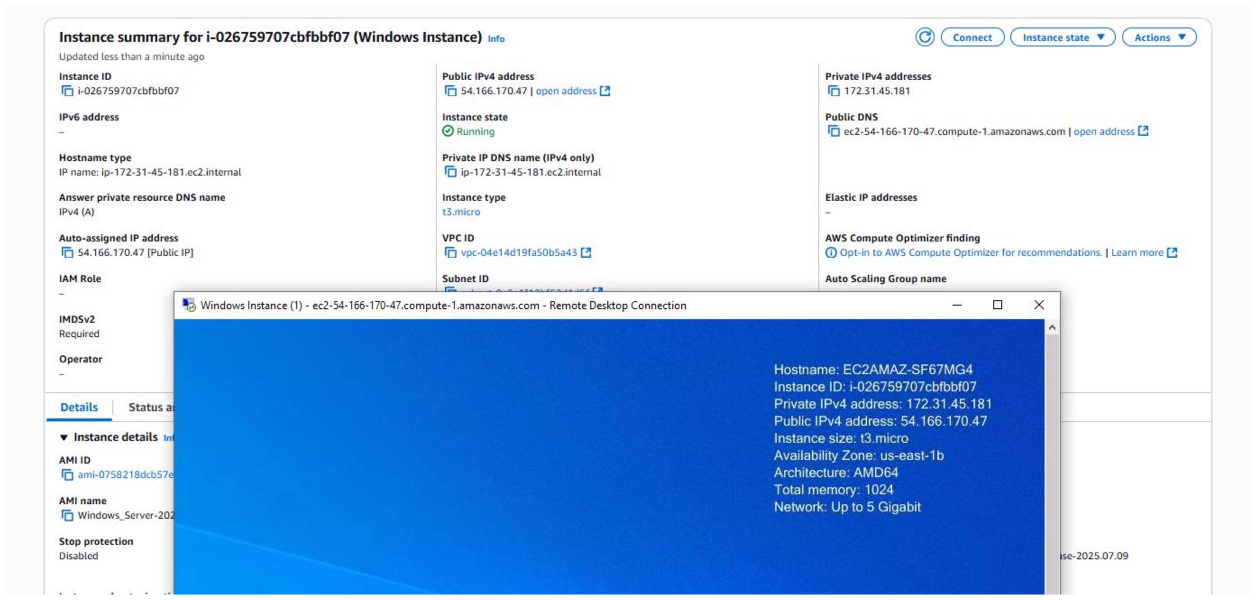
This remote connection could harm your local or remote computer. Do not connect unless you know where this connection came from or have used it before.

Publisher:	**Unknown publisher**
Type:	Remote Desktop Connection
Remote computer:	ec2-54-166-170-47.compute-1.amazonaws.com

☐ Don't ask me again for connections to this computer

John Rich Nicolas
DAY 1

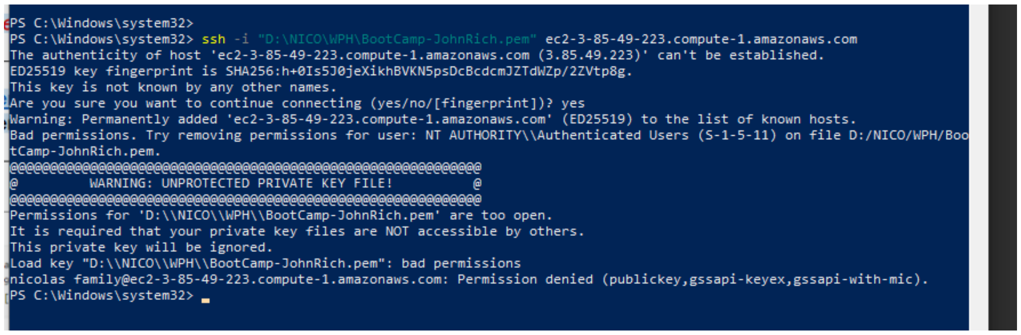
CLOUD ARCHITECTURE
AWS



```
Instance summary for i-026759707cbfbbf07 (Windows Instance) Info
Updated less than a minute ago

Instance ID                          Public IPv4 address                       Private IPv4 addresses
i-026759707cbfbbf07                  54.166.170.47 | open address              172.31.45.181

IPv6 address                         Instance state                            Public DNS
–                                    Running                                   ec2-54-166-170-47.compute-1.amazonaws.com | open address

Hostname type                        Private IP DNS name (IPv4 only)
IP name: ip-172-31-45-181.ec2.internal  ip-172-31-45-181.ec2.internal

Answer private resource DNS name     Instance type                             Elastic IP addresses
IPv4 (A)                             t3.micro                                  –

Auto-assigned IP address             VPC ID                                    AWS Compute Optimizer finding
54.166.170.47 [Public IP]            vpc-04e14d19fa50b5a43                     Opt-in to AWS Compute Optimizer for recommendations. | Learn more

IAM Role                             Subnet ID                                 Auto Scaling Group name
–

IMDSv2
Required

Operator
–

Details | Status a

Instance details Inf
AMI ID
ami-0758218dcb57e

AMI name
Windows_Server-202                                                             se-2025.07.09

Stop protection
Disabled
```

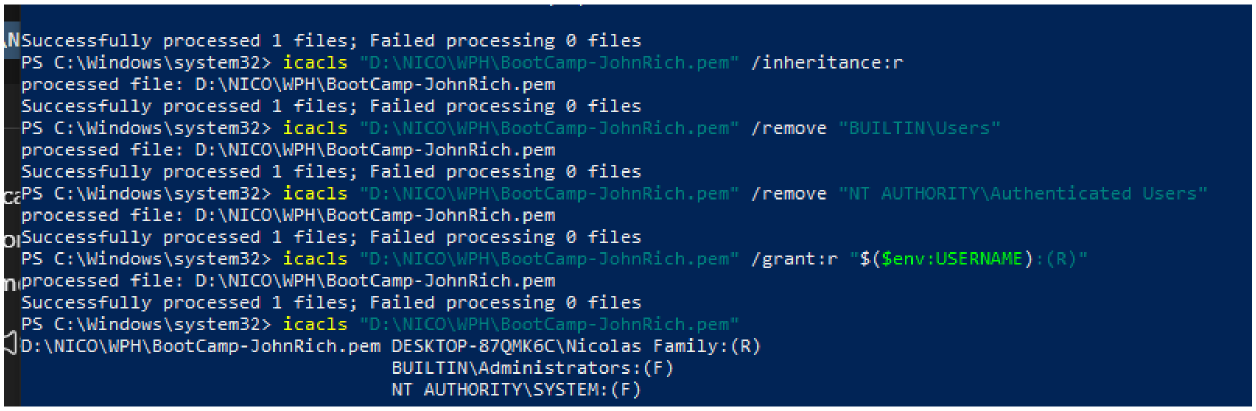
```
Windows Instance (1) - ec2-54-166-170-47.compute-1.amazonaws.com - Remote Desktop Connection

Hostname: EC2AMAZ-SF67MG4
Instance ID: i-026759707cbfbbf07
Private IPv4 address: 172.31.45.181
Public IPv4 address: 54.166.170.47
Instance size: t3.micro
Availability Zone: us-east-1b
Architecture: AMD64
Total memory: 1024
Network: Up to 5 Gigabit
```

ERROR ENCOUNTERED



```
PS C:\Windows\system32>
PS C:\Windows\system32> ssh -i "D:\NICO\WPH\BootCamp-JohnRich.pem" ec2-3-85-49-223.compute-1.amazonaws.com
The authenticity of host 'ec2-3-85-49-223.compute-1.amazonaws.com (3.85.49.223)' can't be established.
ED25519 key fingerprint is SHA256:h+0Is5J0jeXikhBVKN5psDcBcdcmJZTdWZp/2ZVtp8g.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-3-85-49-223.compute-1.amazonaws.com' (ED25519) to the list of known hosts.
Bad permissions. Try removing permissions for user: NT AUTHORITY\\Authenticated Users (S-1-5-11) on file D:/NICO/WPH/Boo
tCamp-JohnRich.pem.
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@         WARNING: UNPROTECTED PRIVATE KEY FILE!          @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
Permissions for 'D:\\NICO\\WPH\\BootCamp-JohnRich.pem' are too open.
It is required that your private key files are NOT accessible by others.
This private key will be ignored.
Load key "D:\\NICO\\WPH\\BootCamp-JohnRich.pem": bad permissions
nicolas family@ec2-3-85-49-223.compute-1.amazonaws.com: Permission denied (publickey,gssapi-keyex,gssapi-with-mic).
PS C:\Windows\system32> _
```

SOLUTION



```
Successfully processed 1 files; Failed processing 0 files
PS C:\Windows\system32> icacls "D:\NICO\WPH\BootCamp-JohnRich.pem" /inheritance:r
processed file: D:\NICO\WPH\BootCamp-JohnRich.pem
Successfully processed 1 files; Failed processing 0 files
PS C:\Windows\system32> icacls "D:\NICO\WPH\BootCamp-JohnRich.pem" /remove "BUILTIN\Users"
processed file: D:\NICO\WPH\BootCamp-JohnRich.pem
Successfully processed 1 files; Failed processing 0 files
PS C:\Windows\system32> icacls "D:\NICO\WPH\BootCamp-JohnRich.pem" /remove "NT AUTHORITY\Authenticated Users"
processed file: D:\NICO\WPH\BootCamp-JohnRich.pem
Successfully processed 1 files; Failed processing 0 files
PS C:\Windows\system32> icacls "D:\NICO\WPH\BootCamp-JohnRich.pem" /grant:r "$($env:USERNAME):(R)"
processed file: D:\NICO\WPH\BootCamp-JohnRich.pem
Successfully processed 1 files; Failed processing 0 files
PS C:\Windows\system32> icacls "D:\NICO\WPH\BootCamp-JohnRich.pem"
D:\NICO\WPH\BootCamp-JohnRich.pem DESKTOP-87QMK6C\Nicolas Family:(R)
                                  BUILTIN\Administrators:(F)
                                  NT AUTHORITY\SYSTEM:(F)
```

John Rich Nicolas CLOUD ARCHITECTURE
DAY 1 AWS

REFLECTION

Amazon EC2 (Elastic Compute Cloud) is a service in AWS that lets you create and run virtual machines in the cloud. In this activity, the goal was to create and access both Linux and Windows instances so I could get hands-on experience with different operating systems in EC2. Before starting, I had to prepare some prerequisites like creating a key pair, setting up security groups for both Linux and Windows, installing an SSH client for Linux access, and making sure I had Remote Desktop Protocol (RDP) for Windows. I also had to make sure I was in the correct AWS region so everything I set up would be consistent.

Part of the process was generating a key pair, making sure I selected the right format (.pem) for Linux, downloading it, and saving it securely. For Linux, I learned that I needed to set the correct file permissions before connecting, otherwise it wouldn't work. I also had to create separate security groups—one for Linux with an inbound rule for SSH (port 22) and one for Windows with an inbound rule for RDP (port 3389). Both security groups also had outbound rules that allowed all traffic so the instances could send and receive data.

When launching the Windows Server 2022 instance, I attached it to the Windows security group and added a user data script in the advanced settings to install IIS Web Server automatically. After it launched, I was able to connect to it through RDP and see that IIS was installed. Finally, to avoid unnecessary costs, I made sure to properly terminate both my Linux and Windows instances after testing.

One of the challenges I faced was when I tried connecting to my Linux instance but got an error saying my private key file had "bad permissions." The error also mentioned removing permissions for NT AUTHORITY\Authenticated Users on my .pem file. I solved this by adjusting the file permissions so it wasn't accessible to other users, and after that, I was able to connect successfully.

Overall, the whole process was time-consuming and honestly overwhelming because there was just too much information to keep track of. On top of that, I was nervous to continue at some points because I borrowed my sister's credit card for my AWS account and I was scared that if I made a mistake, I might accidentally get charged. This activity taught me not only how to launch and connect to instances, but also how to troubleshoot common connection problems—while being careful with every step so I wouldn't get billed unexpectedly.