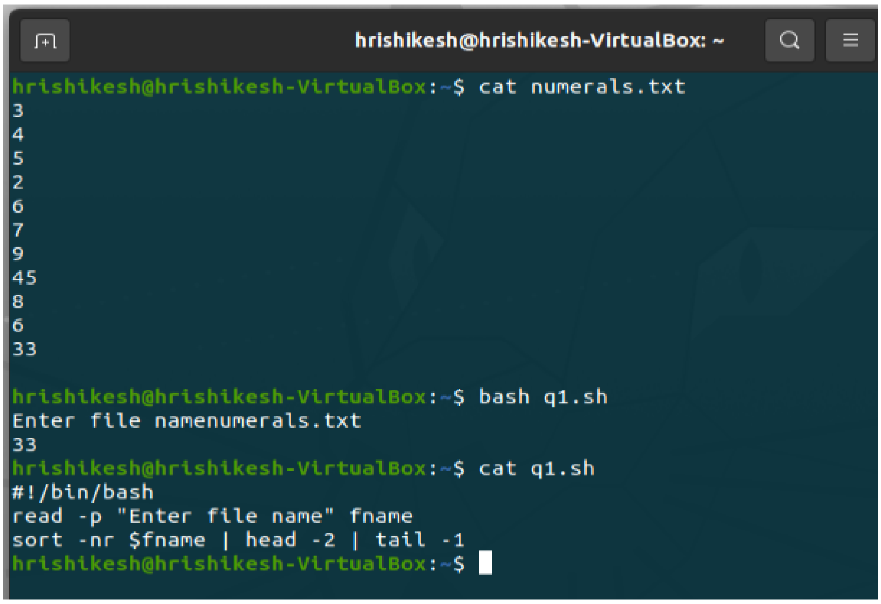
Hrishikesh Tonge | 230340325015
Set B

Linux/Shell

1. Shell program to find second largest number in the list.

    #!/bin/bash
    read -p “Enter filename: “ fname
    #Reads the file name and stores in fname
    sort -nr $fname | head -2 | tail -1
    #To sort the file using numerical values and then reverse the sorted list and then print the second highest value using head and tail



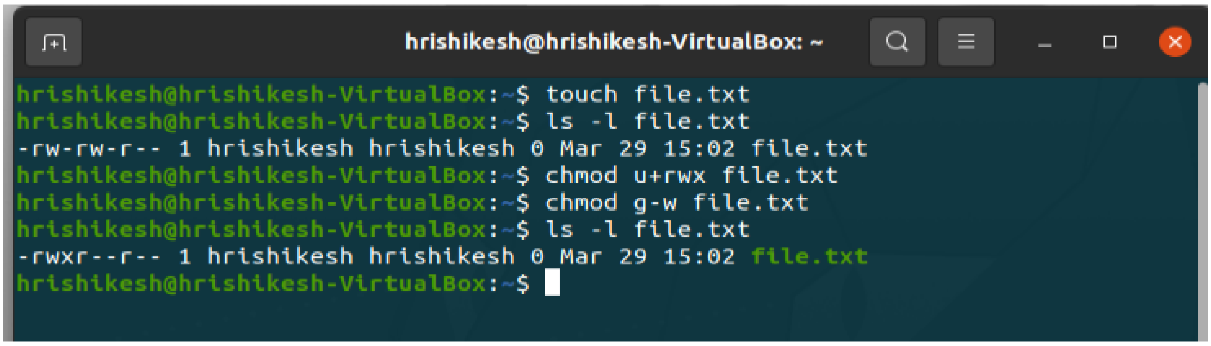
```
hrishikesh@hrishikesh-VirtualBox:~$ cat numerals.txt
3
4
5
2
6
7
9
45
8
6
33

hrishikesh@hrishikesh-VirtualBox:~$ bash q1.sh
Enter file namenumerals.txt
33
hrishikesh@hrishikesh-VirtualBox:~$ cat q1.sh
#!/bin/bash
read -p "Enter file name" fname
sort -nr $fname | head -2 | tail -1
hrishikesh@hrishikesh-VirtualBox:~$ 
```

2. chmod

    touch file.txt #created file
    ls -l file.txt

    chmod u+rwx #give owner read write execute
    chmod g-w #removed write permissions for group

    ls -l file.txt #check file permissions

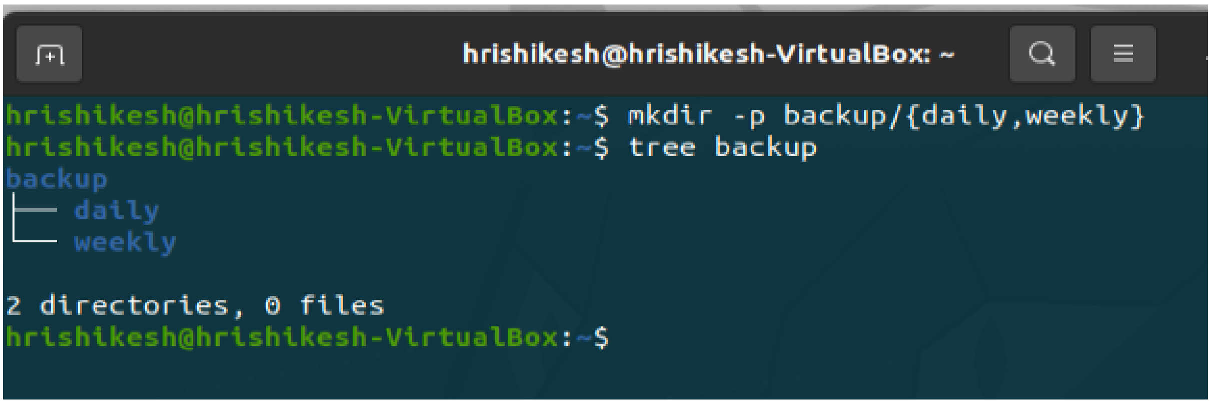


```
hrishikesh@hrishikesh-VirtualBox:~$ touch file.txt
hrishikesh@hrishikesh-VirtualBox:~$ ls -l file.txt
-rw-rw-r-- 1 hrishikesh hrishikesh 0 Mar 29 15:02 file.txt
hrishikesh@hrishikesh-VirtualBox:~$ chmod u+rwx file.txt
hrishikesh@hrishikesh-VirtualBox:~$ chmod g-w file.txt
hrishikesh@hrishikesh-VirtualBox:~$ ls -l file.txt
-rwxr--r-- 1 hrishikesh hrishikesh 0 Mar 29 15:02 file.txt
hrishikesh@hrishikesh-VirtualBox:~$ 
```

3. Directory

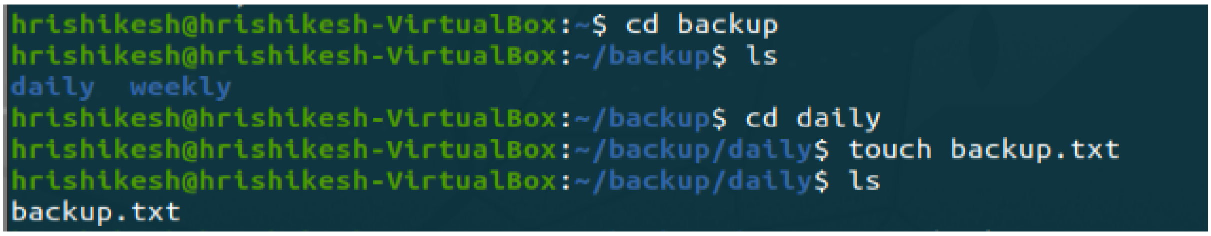
   mkdir -p backup/{daily,weekly} #created directory backup with daily, weekly as subdirectories

   tree backup



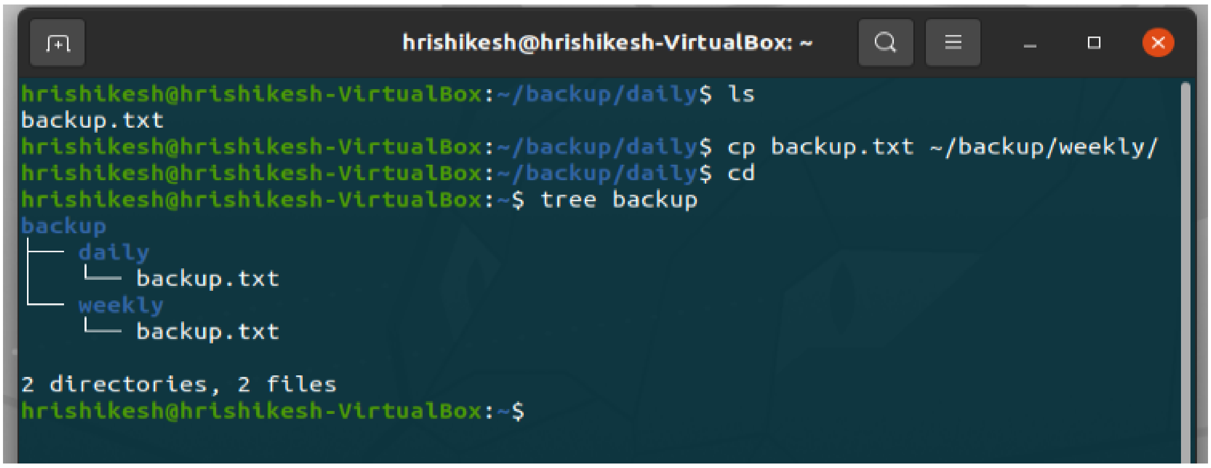
```
hrishikesh@hrishikesh-VirtualBox:~$ mkdir -p backup/{daily,weekly}
hrishikesh@hrishikesh-VirtualBox:~$ tree backup
backup
├── daily
└── weekly

2 directories, 0 files
hrishikesh@hrishikesh-VirtualBox:~$
```

cd backup
ls
cd daily
touch backup.txt #created backup.txt file under daily
ls



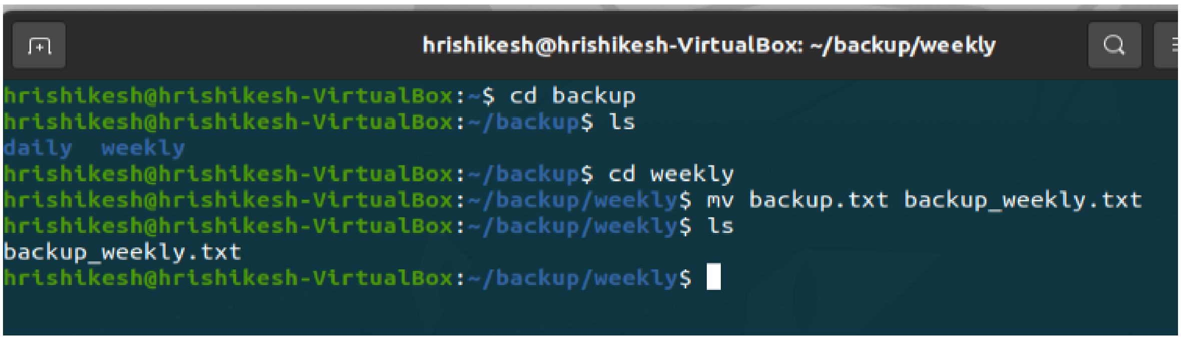
```
hrishikesh@hrishikesh-VirtualBox:~$ cd backup
hrishikesh@hrishikesh-VirtualBox:~/backup$ ls
daily  weekly
hrishikesh@hrishikesh-VirtualBox:~/backup$ cd daily
hrishikesh@hrishikesh-VirtualBox:~/backup/daily$ touch backup.txt
hrishikesh@hrishikesh-VirtualBox:~/backup/daily$ ls
backup.txt
```

ls
cp backup.txt ~/backup/weekly #copied backup.txt to backup/weekly
cd
tree backup



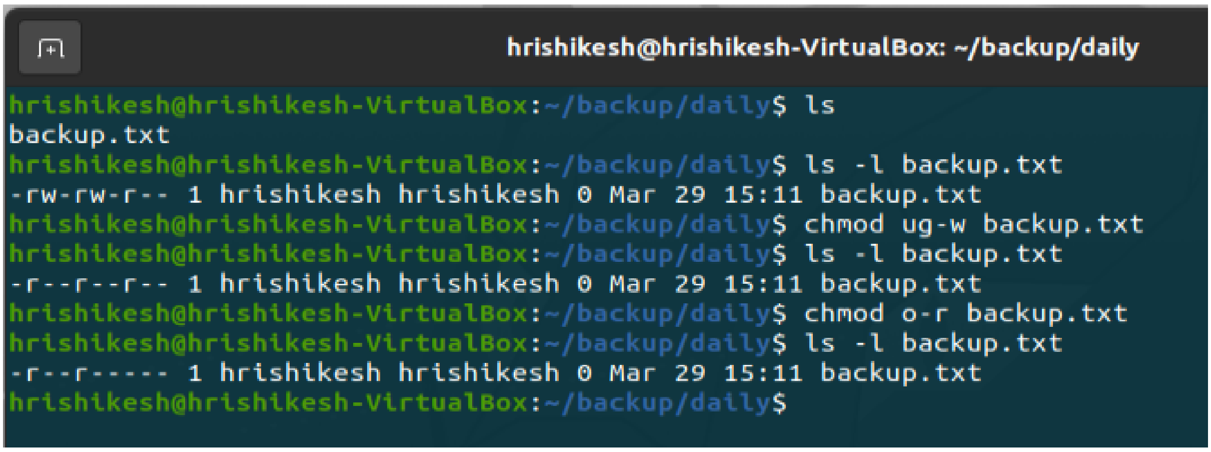
```
hrishikesh@hrishikesh-VirtualBox:~/backup/daily$ ls
backup.txt
hrishikesh@hrishikesh-VirtualBox:~/backup/daily$ cp backup.txt ~/backup/weekly/
hrishikesh@hrishikesh-VirtualBox:~/backup/daily$ cd
hrishikesh@hrishikesh-VirtualBox:~$ tree backup
backup
├── daily
│   └── backup.txt
└── weekly
    └── backup.txt

2 directories, 2 files
hrishikesh@hrishikesh-VirtualBox:~$
```

cd backup
ls
cd weekly
mv backup.txt backup_weekly.txt #renamed backup.txt to backup_weekly.txt which is under weekly
ls



```
hrishikesh@hrishikesh-VirtualBox:~$ cd backup
hrishikesh@hrishikesh-VirtualBox:~/backup$ ls
daily  weekly
hrishikesh@hrishikesh-VirtualBox:~/backup$ cd weekly
hrishikesh@hrishikesh-VirtualBox:~/backup/weekly$ mv backup.txt backup_weekly.txt
hrishikesh@hrishikesh-VirtualBox:~/backup/weekly$ ls
backup_weekly.txt
hrishikesh@hrishikesh-VirtualBox:~/backup/weekly$ 
```

ls
ls -l backup.txt #checked for existing permissions of file backup.txt
chmod ug-w backup.txt #removed write permissions for user and group
ls -l backup.txt
chmod o-r backup.txt #removed all permissions for others
ls -l backup.txt



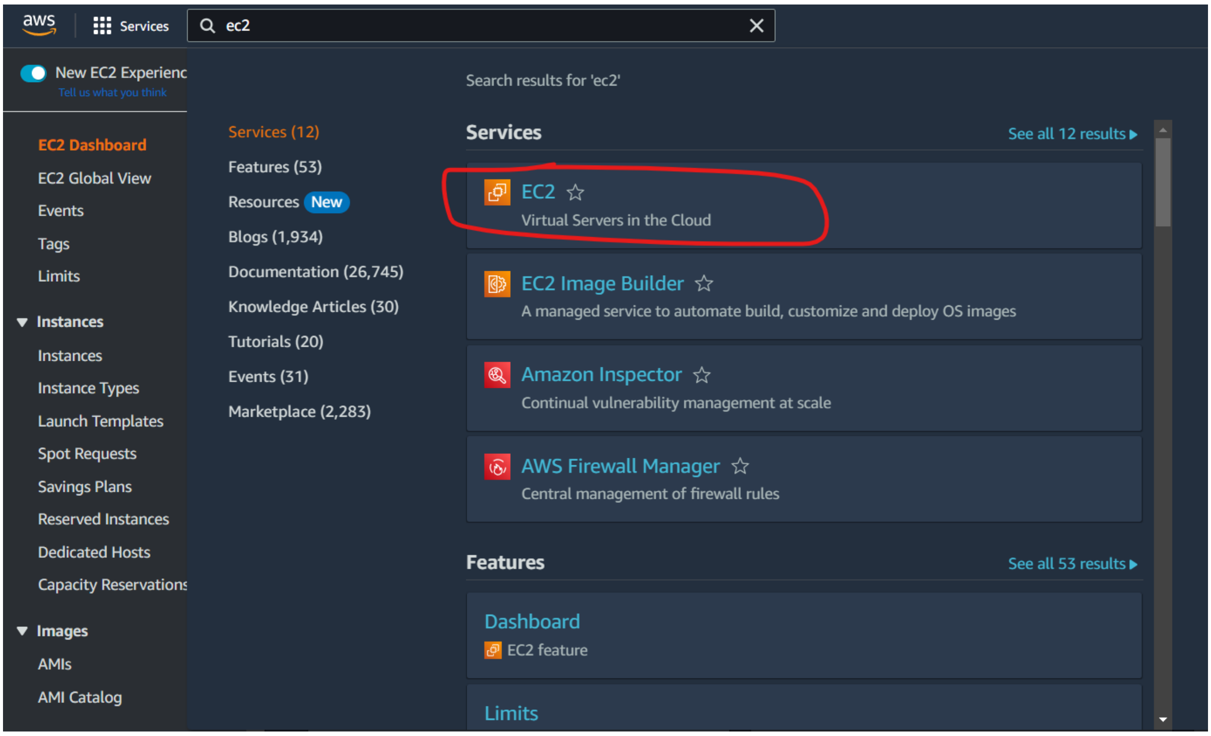
```
hrishikesh@hrishikesh-VirtualBox:~/backup/daily$ ls
backup.txt
hrishikesh@hrishikesh-VirtualBox:~/backup/daily$ ls -l backup.txt
-rw-rw-r-- 1 hrishikesh hrishikesh 0 Mar 29 15:11 backup.txt
hrishikesh@hrishikesh-VirtualBox:~/backup/daily$ chmod ug-w backup.txt
hrishikesh@hrishikesh-VirtualBox:~/backup/daily$ ls -l backup.txt
-r--r--r-- 1 hrishikesh hrishikesh 0 Mar 29 15:11 backup.txt
hrishikesh@hrishikesh-VirtualBox:~/backup/daily$ chmod o-r backup.txt
hrishikesh@hrishikesh-VirtualBox:~/backup/daily$ ls -l backup.txt
-r--r----- 1 hrishikesh hrishikesh 0 Mar 29 15:11 backup.txt
hrishikesh@hrishikesh-VirtualBox:~/backup/daily$
```

Hrishikesh Tonge | 230340325015
Set B

Cloud

4. Create a Linux EC2

Search for ec2 instance in services



Search results for 'ec2'

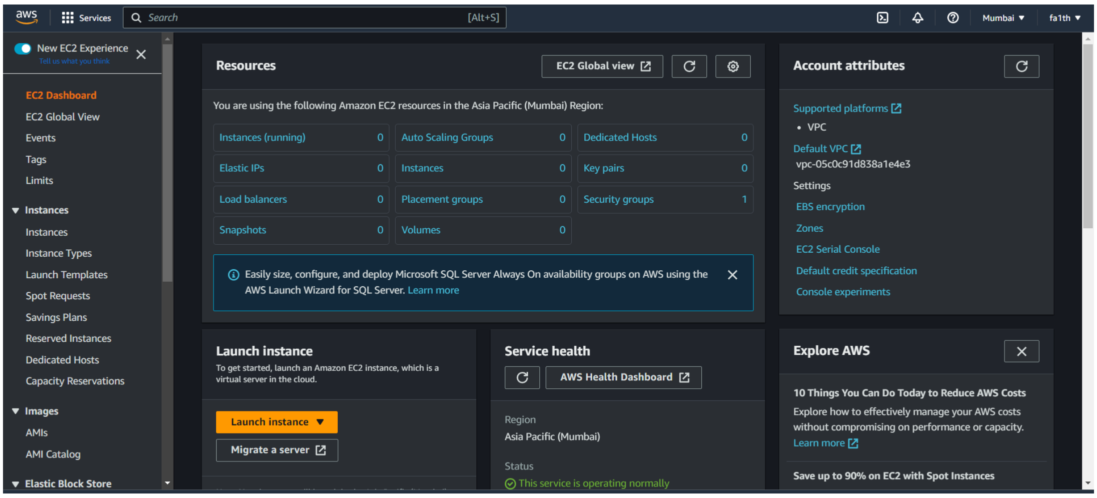
EC2 — Virtual Servers in the Cloud

EC2 Image Builder — A managed service to automate build, customize and deploy OS images

Amazon Inspector — Continual vulnerability management at scale

AWS Firewall Manager — Central management of firewall rules

EC2 Dashboard



Resources

You are using the following Amazon EC2 resources in the Asia Pacific (Mumbai) Region:

Instances (running)	0	Auto Scaling Groups	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		

Easily size, configure, and deploy Microsoft SQL Server Always On availability groups on AWS using the AWS Launch Wizard for SQL Server. Learn more

Account attributes

Supported platforms
- VPC

Default VPC
vpc-05c0c91d838a1e4e3

Launch instance

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

Service health

Region
Asia Pacific (Mumbai)

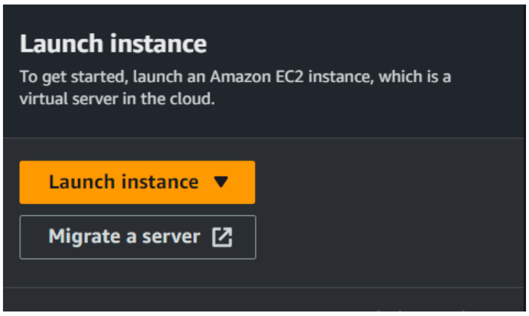
Status
This service is operating normally

Explore AWS

10 Things You Can Do Today to Reduce AWS Costs
Explore how to effectively manage your AWS costs without compromising on performance or capacity.
Learn more

Save up to 90% on EC2 with Spot Instances

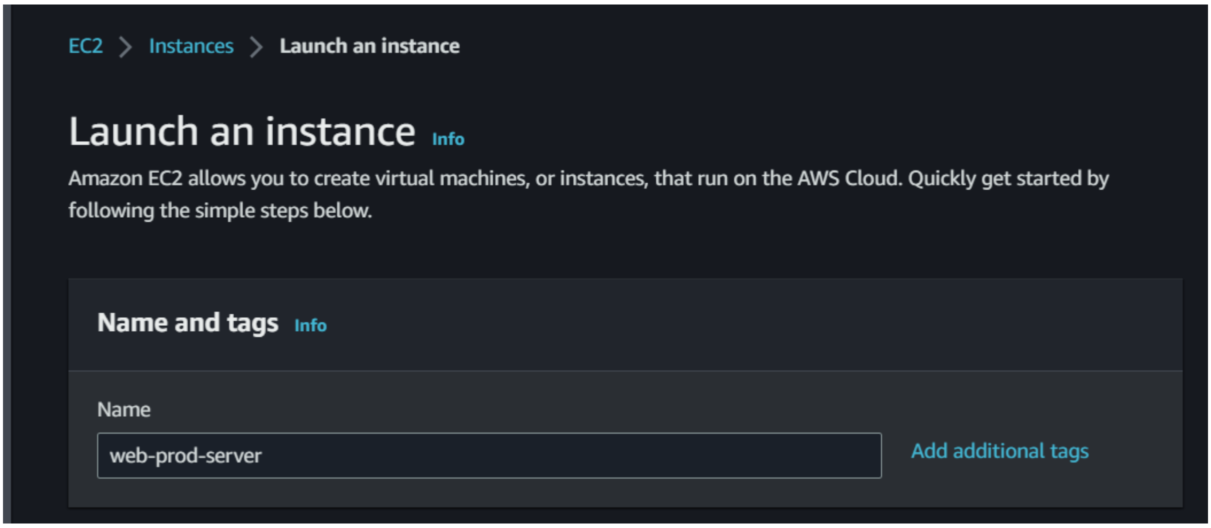
Launch Instance



Launch instance

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

Suitable name for the instance



EC2 > Instances > Launch an instance

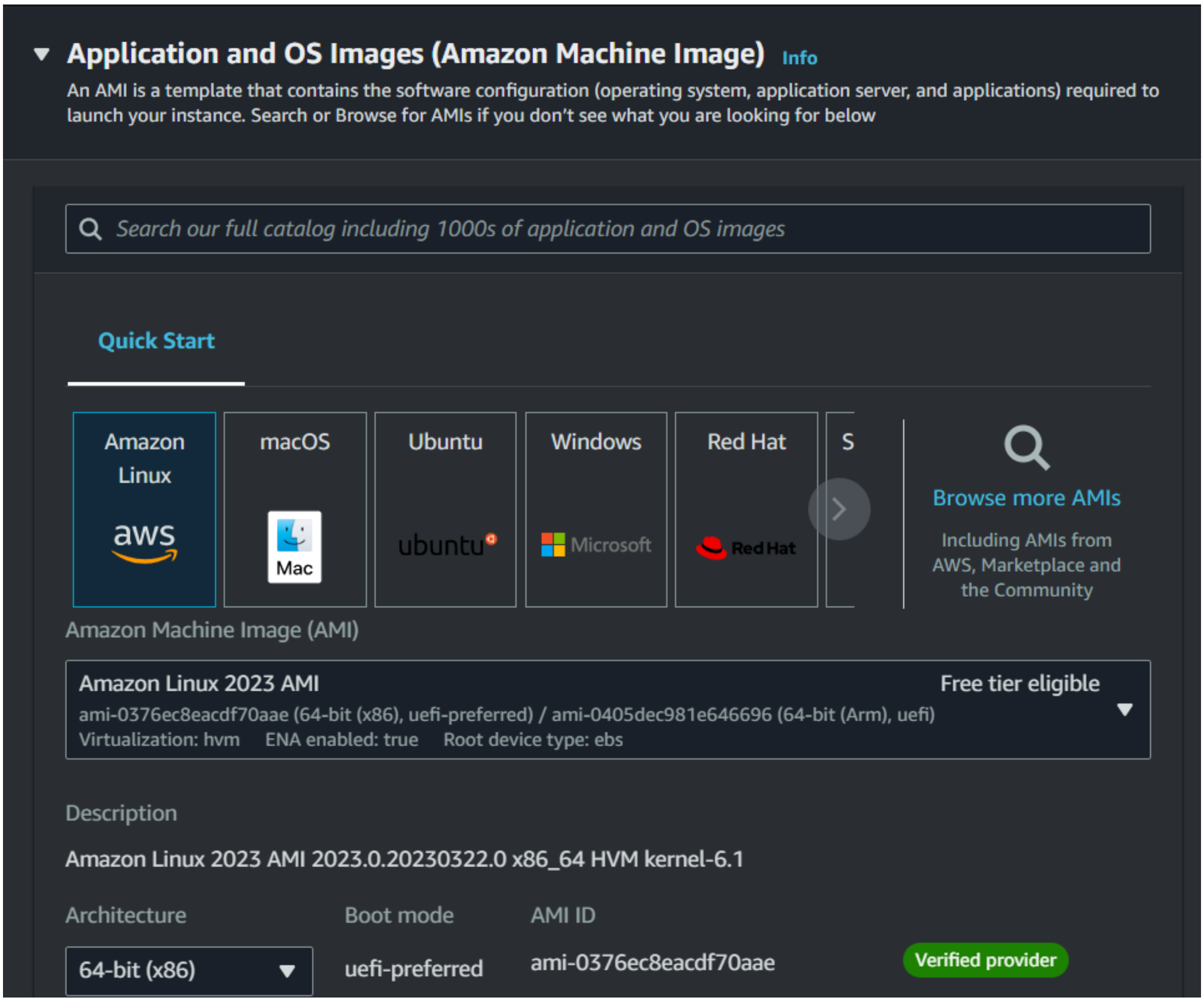
Launch an instance Info

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: web-prod-server

Selecting the OS and architecture.
Here we are selecting **Amazon Linux 64 bit architecture**



▼ Application and OS Images (Amazon Machine Image) Info

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

Search our full catalog including 1000s of application and OS images
Quick Start
Amazon Linux
aws
macOS
Mac
Ubuntu
ubuntu
Windows
Microsoft
Red Hat
Red Hat
S
Browse more AMIs
Including AMIs from AWS, Marketplace and the Community

Amazon Machine Image (AMI)

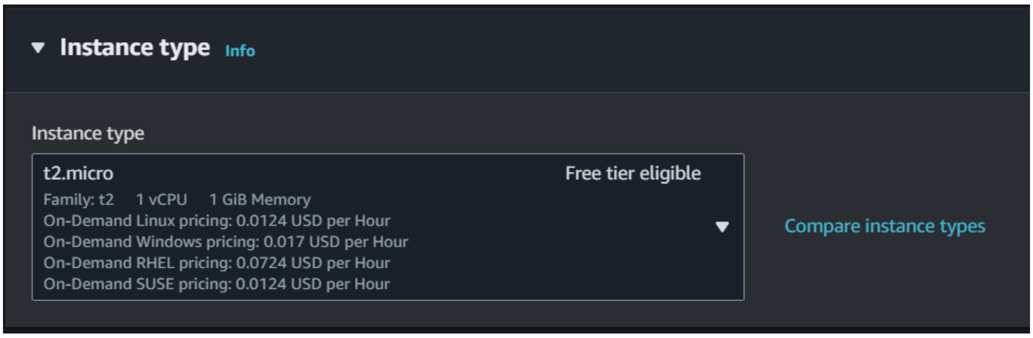
Amazon Linux 2023 AMI — Free tier eligible
ami-0376ec8eacdf70aae (64-bit (x86), uefi-preferred) / ami-0405dec981e646696 (64-bit (Arm), uefi)
Virtualization: hvm ENA enabled: true Root device type: ebs

Description

Amazon Linux 2023 AMI 2023.0.20230322.0 x86_64 HVM kernel-6.1

Architecture	Boot mode	AMI ID	
64-bit (x86) ▼	uefi-preferred	ami-0376ec8eacdf70aae	Verified provider

Instance type as per the requirement
Here t2.micro
1 vCPU and GiB Memory

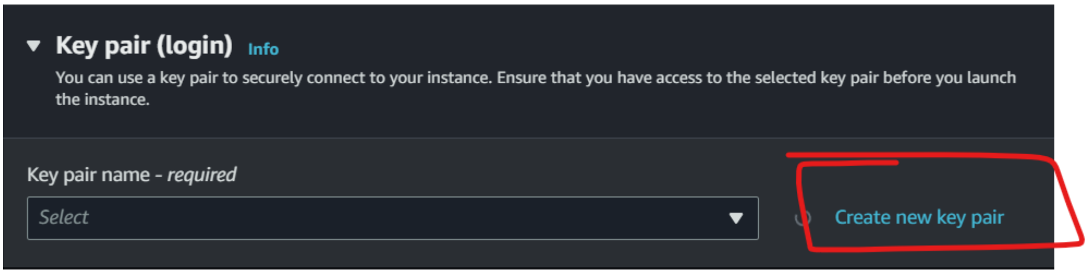


▼ Instance type Info

Instance type

t2.micro — Free tier eligible
Family: t2 1 vCPU 1 GiB Memory
On-Demand Linux pricing: 0.0124 USD per Hour
On-Demand Windows pricing: 0.017 USD per Hour
On-Demand RHEL pricing: 0.0724 USD per Hour
On-Demand SUSE pricing: 0.0124 USD per Hour

Now we need key pair for authentication
So we create new key



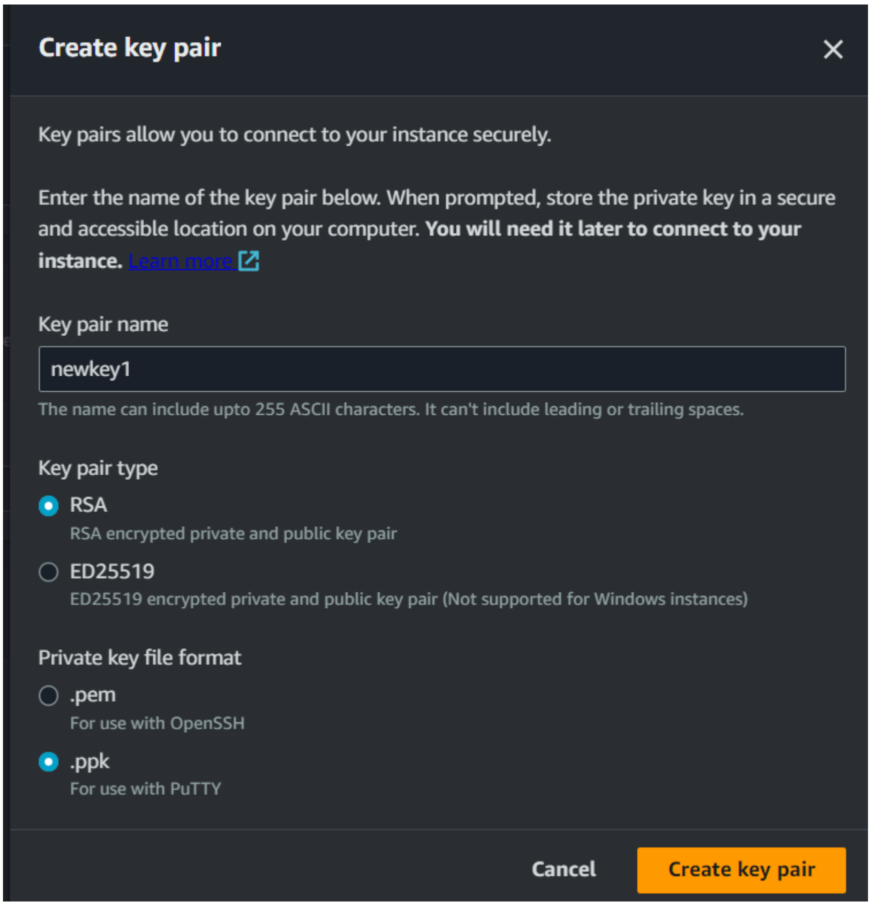
▼ Key pair (login) Info

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

Key pair name - *required*

Select

New key name and type **.ppk** since we have to connect using putty.



Create key pair

Key pairs allow you to connect to your instance securely.

Enter the name of the key pair below. When prompted, store the private key in a secure and accessible location on your computer. **You will need it later to connect to your instance.** Learn more

Key pair name

newkey1

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

Key pair type

● RSA
RSA encrypted private and public key pair

○ ED25519
ED25519 encrypted private and public key pair (Not supported for Windows instances)

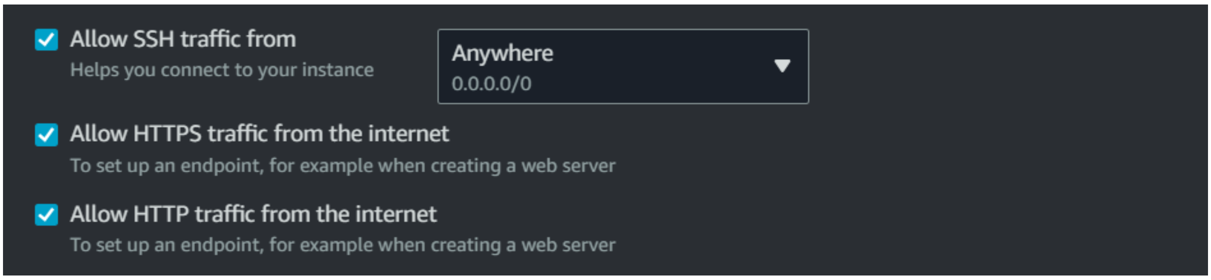
Private key file format

○ .pem
For use with OpenSSH

● .ppk
For use with PuTTY

Key will be downloaded.

Now we need to allow traffic from anywhere



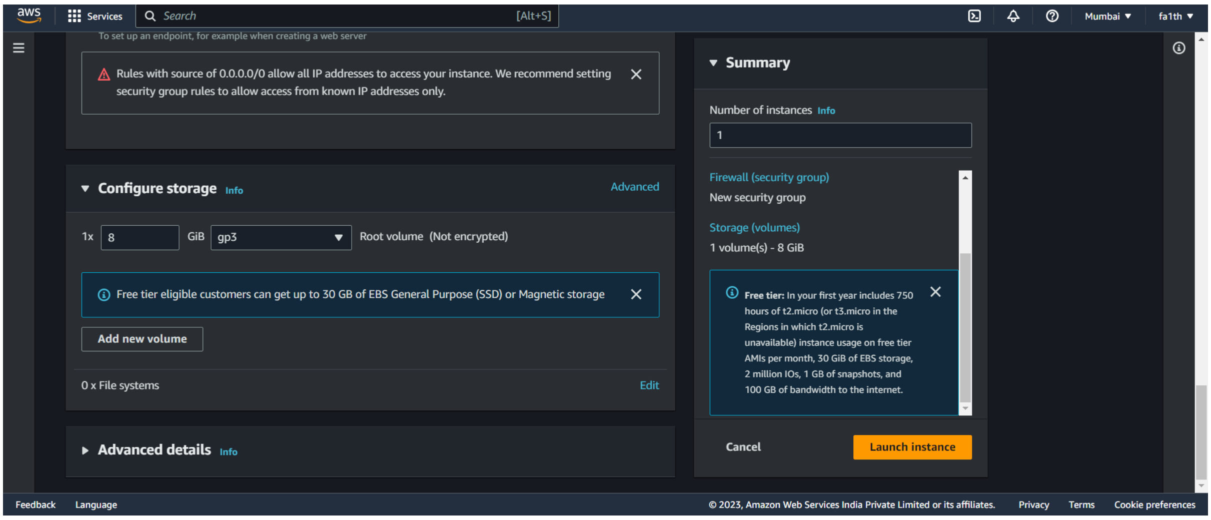
☑ Allow SSH traffic from
Helps you connect to your instance

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

Now Launch instance



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.

▼ Configure storage Info — Advanced

1x 8 GiB gp3 Root volume (Not encrypted)

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

Add new volume

0 x File systems — Edit

▶ Advanced details Info

▼ Summary

Number of instances Info: 1

Firewall (security group)
New security group

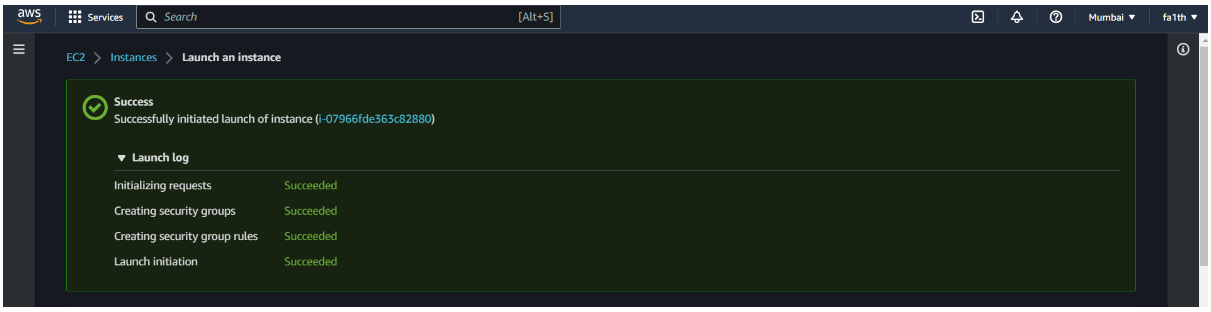
Storage (volumes)
1 volume(s) - 8 GiB

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, 30 GiB of EBS storage, 2 million IOs, 1 GB of snapshots, and 100 GB of bandwidth to the internet.

Cancel — Launch instance

Feedback
Language
© 2023, Amazon Web Services India Private Limited or its affiliates.
Privacy
Terms
Cookie preferences

Instance successfully created



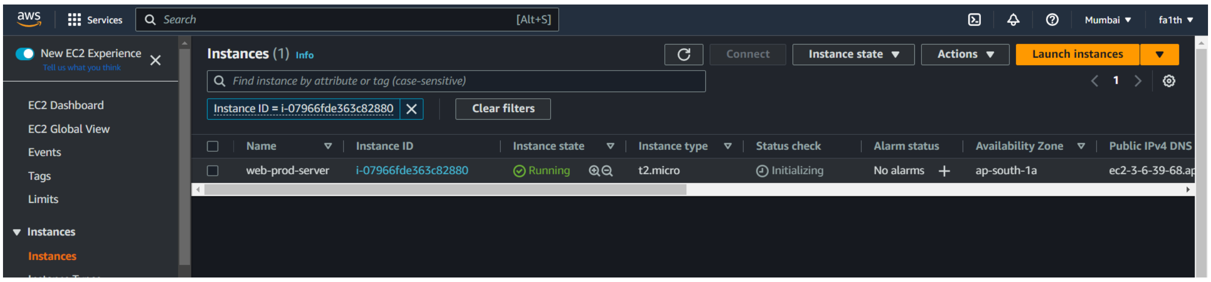
EC2 > Instances > Launch an instance

Success
Successfully initiated launch of instance (i-07966fde363c82880)

▼ Launch log

Initializing requests	Succeeded
Creating security groups	Succeeded
Creating security group rules	Succeeded
Launch initiation	Succeeded

Now we click on instance id and it will redirect to console



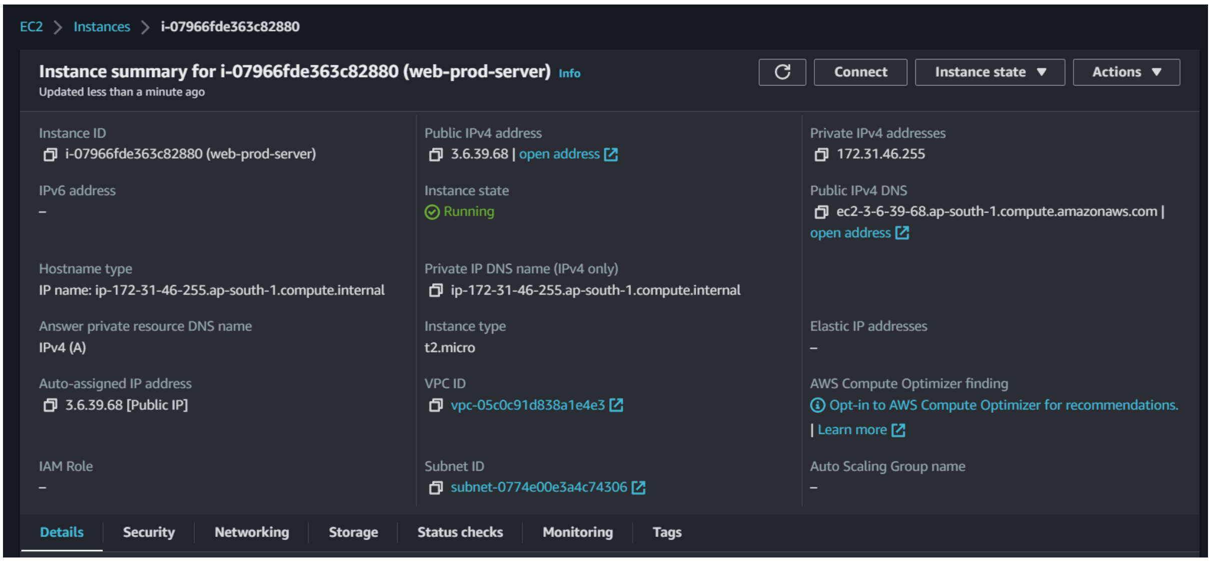
Instances (1) Info

Find instance by attribute or tag (case-sensitive)

Instance ID = i-07966fde363c82880 — Clear filters

☐	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS
☐	web-prod-server	i-07966fde363c82880	Running	t2.micro	Initializing	No alarms	ap-south-1a	ec2-3-6-39-68.ap

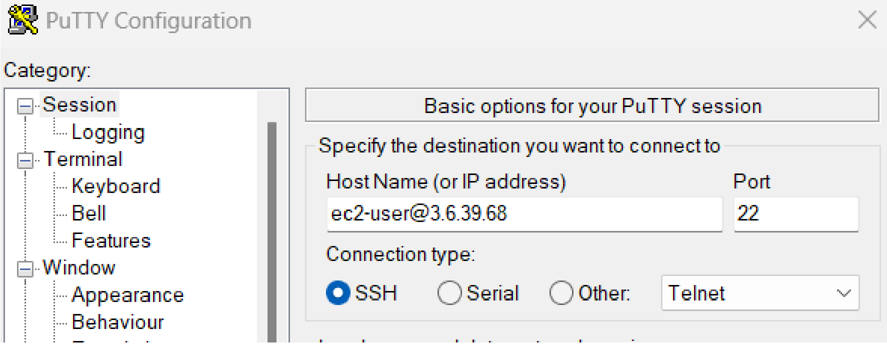
Hrishikesh Tonge | 230340325015
Set B



Instance summary for i-07966fde363c82880 (web-prod-server) Info
Updated less than a minute ago

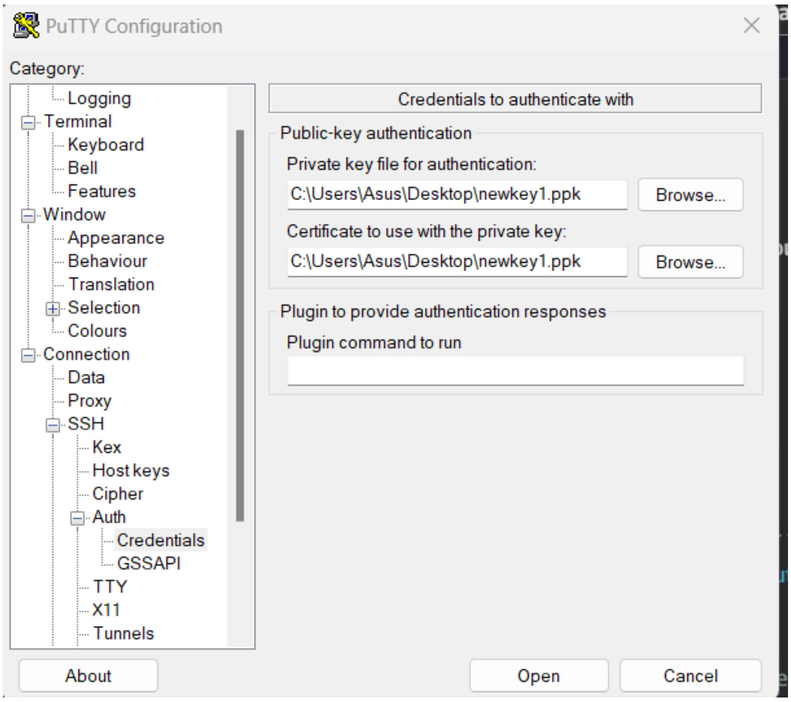
Instance ID: i-07966fde363c82880 (web-prod-server)
Public IPv4 address: 3.6.39.68 | open address
Private IPv4 addresses: 172.31.46.255
IPv6 address: –
Instance state: Running
Public IPv4 DNS: ec2-3-6-39-68.ap-south-1.compute.amazonaws.com | open address
Hostname type: IP name: ip-172-31-46-255.ap-south-1.compute.internal
Private IP DNS name (IPv4 only): ip-172-31-46-255.ap-south-1.compute.internal
Answer private resource DNS name: IPv4 (A)
Instance type: t2.micro
Elastic IP addresses: –
Auto-assigned IP address: 3.6.39.68 [Public IP]
VPC ID: vpc-05c0c91d838a1e4e3
AWS Compute Optimizer finding: Opt-in to AWS Compute Optimizer for recommendations. | Learn more
IAM Role: –
Subnet ID: subnet-0774e00e3a4c74306
Auto Scaling Group name: –

Now we open Putty on our local machine
We enter the user name and public ip of the instance and select ssh



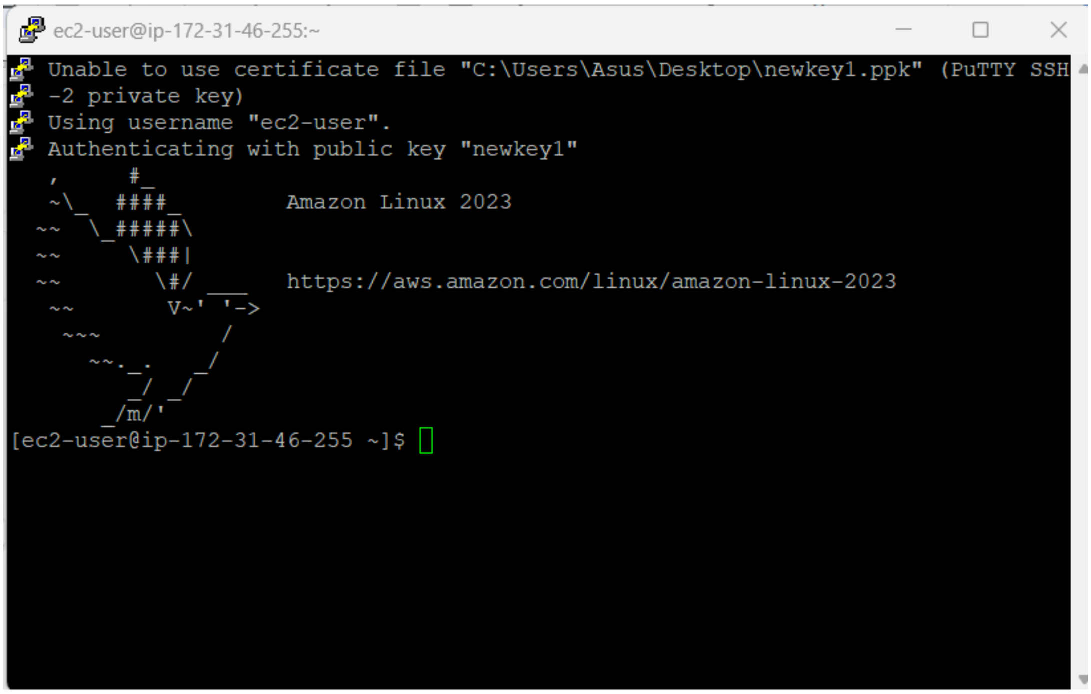
Basic options for your PuTTY session
Specify the destination you want to connect to
Host Name (or IP address): ec2-user@3.6.39.68
Port: 22
Connection type: ● SSH ○ Serial ○ Other: Telnet

SSH > Auth > Credentials



Credentials to authenticate with
Public-key authentication
Private key file for authentication: C:\Users\Asus\Desktop\newkey1.ppk
Certificate to use with the private key: C:\Users\Asus\Desktop\newkey1.ppk
Plugin to provide authentication responses
Plugin command to run:

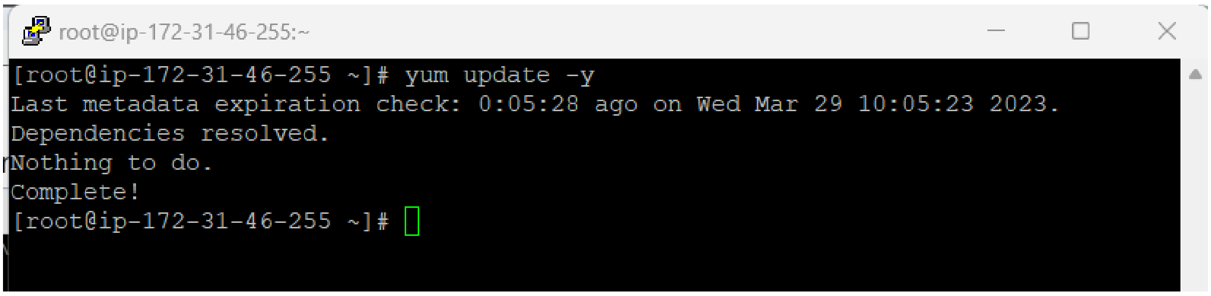
Click open and we will be inside our instance (Amazon Linux)



```
ec2-user@ip-172-31-46-255:~
Unable to use certificate file "C:\Users\Asus\Desktop\newkey1.ppk" (PuTTY SSH
-2 private key)
Using username "ec2-user".
Authenticating with public key "newkey1"
   ,     #_
   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
[ec2-user@ip-172-31-46-255 ~]$
```

Check for any updates

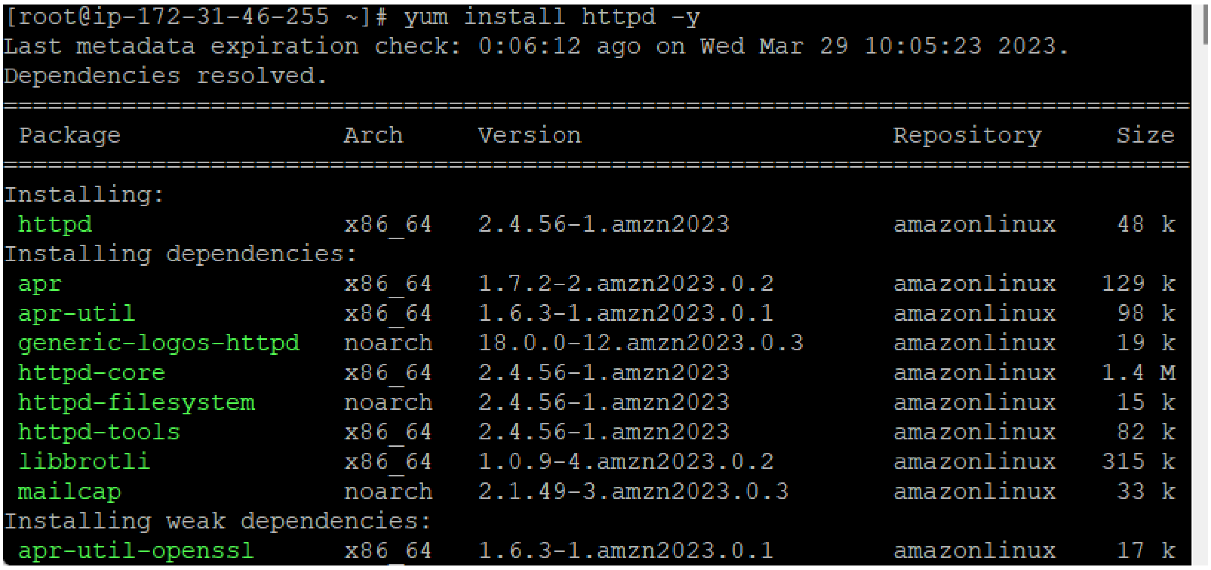
yum update -y



```
root@ip-172-31-46-255:~
[root@ip-172-31-46-255 ~]# yum update -y
Last metadata expiration check: 0:05:28 ago on Wed Mar 29 10:05:23 2023.
Dependencies resolved.
Nothing to do.
Complete!
[root@ip-172-31-46-255 ~]#
```

Now we need the apache http server to host our webpage

yum install httpd -y

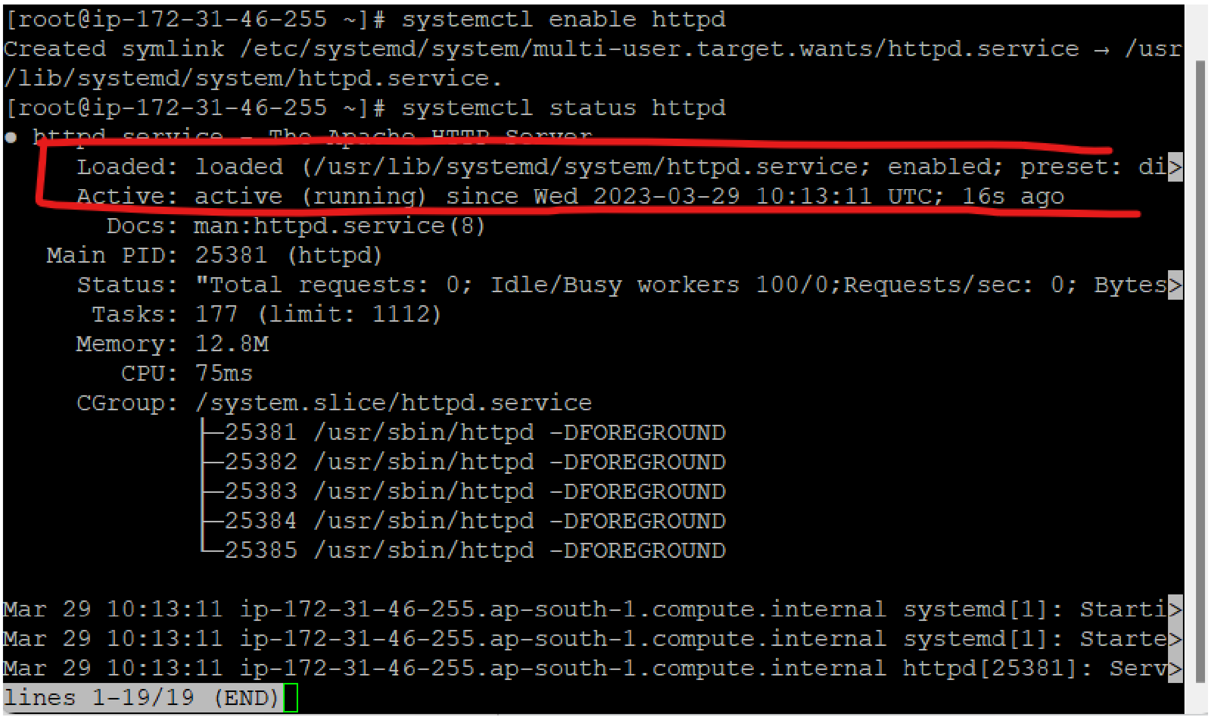


```
[root@ip-172-31-46-255 ~]# yum install httpd -y
Last metadata expiration check: 0:06:12 ago on Wed Mar 29 10:05:23 2023.
Dependencies resolved.
================================================================================
 Package                Arch     Version                     Repository    Size
================================================================================
Installing:
 httpd                  x86_64   2.4.56-1.amzn2023           amazonlinux   48 k
Installing dependencies:
 apr                    x86_64   1.7.2-2.amzn2023.0.2        amazonlinux  129 k
 apr-util               x86_64   1.6.3-1.amzn2023.0.1        amazonlinux   98 k
 generic-logos-httpd    noarch   18.0.0-12.amzn2023.0.3      amazonlinux   19 k
 httpd-core             x86_64   2.4.56-1.amzn2023           amazonlinux  1.4 M
 httpd-filesystem       noarch   2.4.56-1.amzn2023           amazonlinux   15 k
 httpd-tools            x86_64   2.4.56-1.amzn2023           amazonlinux   82 k
 libbrotli              x86_64   1.0.9-4.amzn2023.0.2        amazonlinux  315 k
 mailcap                noarch   2.1.49-3.amzn2023.0.3       amazonlinux   33 k
Installing weak dependencies:
 apr-util-openssl       x86_64   1.6.3-1.amzn2023.0.1        amazonlinux   17 k
```

After installing the service, we need to start and enable the httpd service

systemctl start httpd
systemctl enable httpd
systemctl status httpd

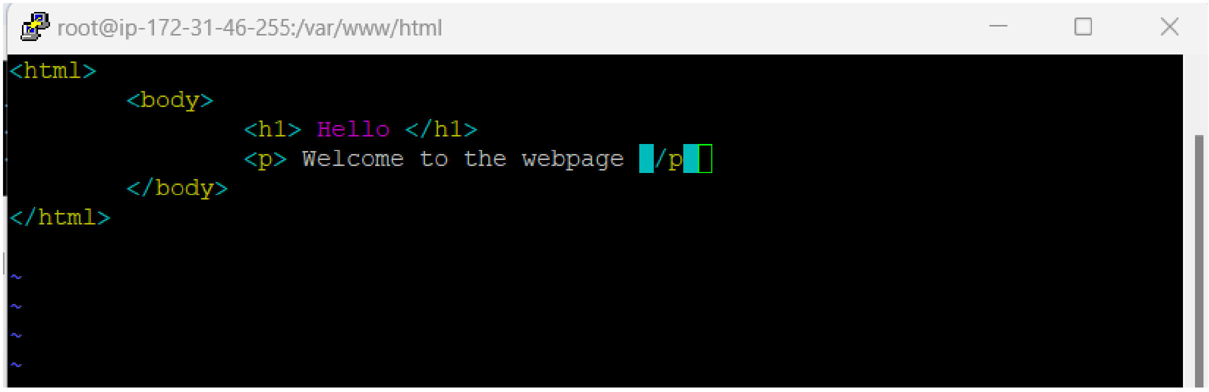
Thus the status of httpd is active and running



```
[root@ip-172-31-46-255 ~]# systemctl enable httpd
Created symlink /etc/systemd/system/multi-user.target.wants/httpd.service → /usr
/lib/systemd/system/httpd.service.
[root@ip-172-31-46-255 ~]# systemctl status httpd
● httpd.service - The Apache HTTP Server
     Loaded: loaded (/usr/lib/systemd/system/httpd.service; enabled; preset: di>
     Active: active (running) since Wed 2023-03-29 10:13:11 UTC; 16s ago
       Docs: man:httpd.service(8)
   Main PID: 25381 (httpd)
     Status: "Total requests: 0; Idle/Busy workers 100/0;Requests/sec: 0; Bytes>
      Tasks: 177 (limit: 1112)
     Memory: 12.8M
        CPU: 75ms
     CGroup: /system.slice/httpd.service
             ├─25381 /usr/sbin/httpd -DFOREGROUND
             ├─25382 /usr/sbin/httpd -DFOREGROUND
             ├─25383 /usr/sbin/httpd -DFOREGROUND
             ├─25384 /usr/sbin/httpd -DFOREGROUND
             └─25385 /usr/sbin/httpd -DFOREGROUND

Mar 29 10:13:11 ip-172-31-46-255.ap-south-1.compute.internal systemd[1]: Starti>
Mar 29 10:13:11 ip-172-31-46-255.ap-south-1.compute.internal systemd[1]: Starte>
Mar 29 10:13:11 ip-172-31-46-255.ap-south-1.compute.internal httpd[25381]: Serv>
lines 1-19/19 (END)
```

Now we need to create a html page. For that we need to go into /var/www/html

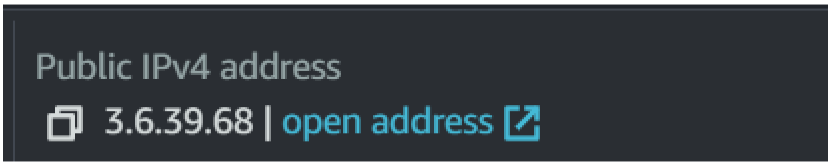
cd /var/www/html
vim index.html



root@ip-172-31-46-255:/var/www/html

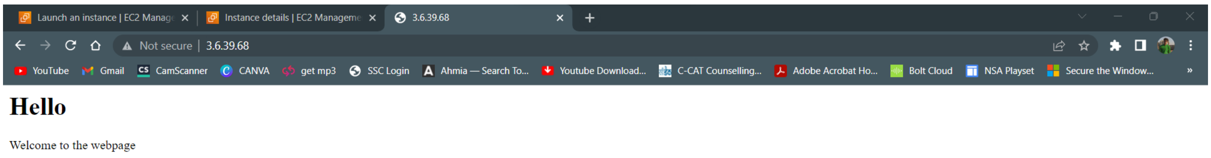
```
<html>
        <body>
                <h1> Hello </h1>
                <p> Welcome to the webpage </p
        </body>
</html>
~
~
~
~
```

Copy the public IP



Public IPv4 address
3.6.39.68 | open address

Paste in another tab and it will show the contents of html file

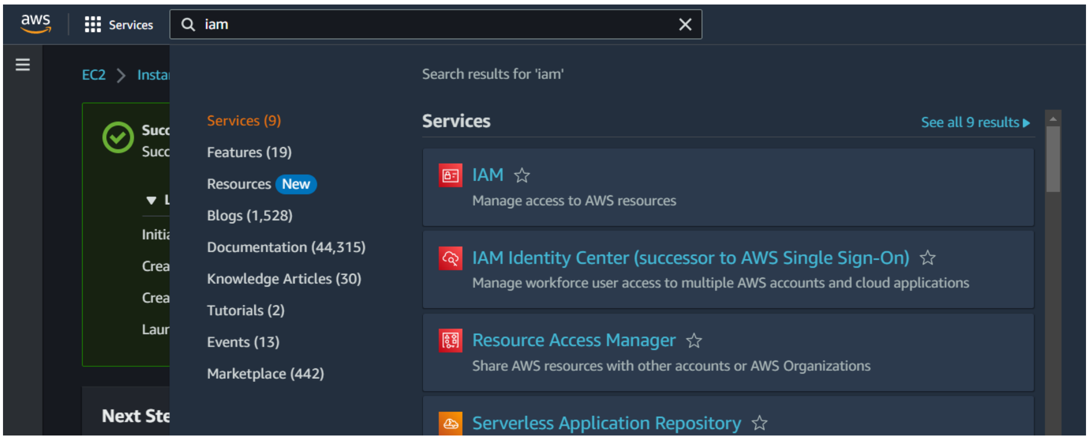


Hello

Welcome to the webpage

5. IAM

Search IAM in services



Search results for 'iam'

Services

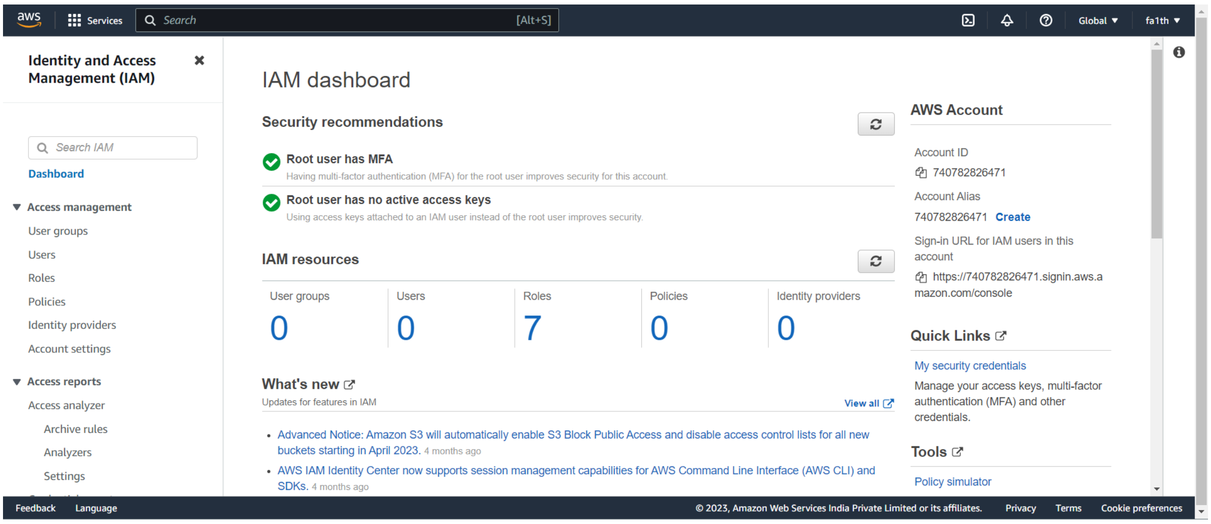
IAM ☆
Manage access to AWS resources

IAM Identity Center (successor to AWS Single Sign-On) ☆
Manage workforce user access to multiple AWS accounts and cloud applications

Resource Access Manager ☆
Share AWS resources with other accounts or AWS Organizations

Serverless Application Repository ☆

IAM Dashboard will open up



IAM dashboard

Security recommendations

✓ Root user has MFA
Having multi-factor authentication (MFA) for the root user improves security for this account.

✓ Root user has no active access keys
Using access keys attached to an IAM user instead of the root user improves security.

IAM resources

User groups	Users	Roles	Policies	Identity providers
0	0	7	0	0

What's new
Updates for features in IAM
View all

- Advanced Notice: Amazon S3 will automatically enable S3 Block Public Access and disable access control lists for all new buckets starting in April 2023. 4 months ago
- AWS IAM Identity Center now supports session management capabilities for AWS Command Line Interface (AWS CLI) and SDKs. 4 months ago

AWS Account

Account ID
740782826471

Account Alias
740782826471 Create

Sign-in URL for IAM users in this account
https://740782826471.signin.aws.amazon.com/console

Quick Links

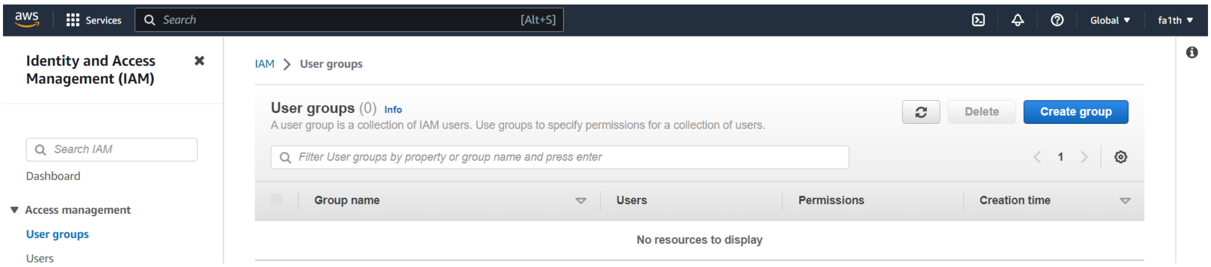
My security credentials
Manage your access keys, multi-factor authentication (MFA) and other credentials.

Tools

Policy simulator

© 2023, Amazon Web Services India Private Limited or its affiliates.

Then we click on groups and create new group - devops

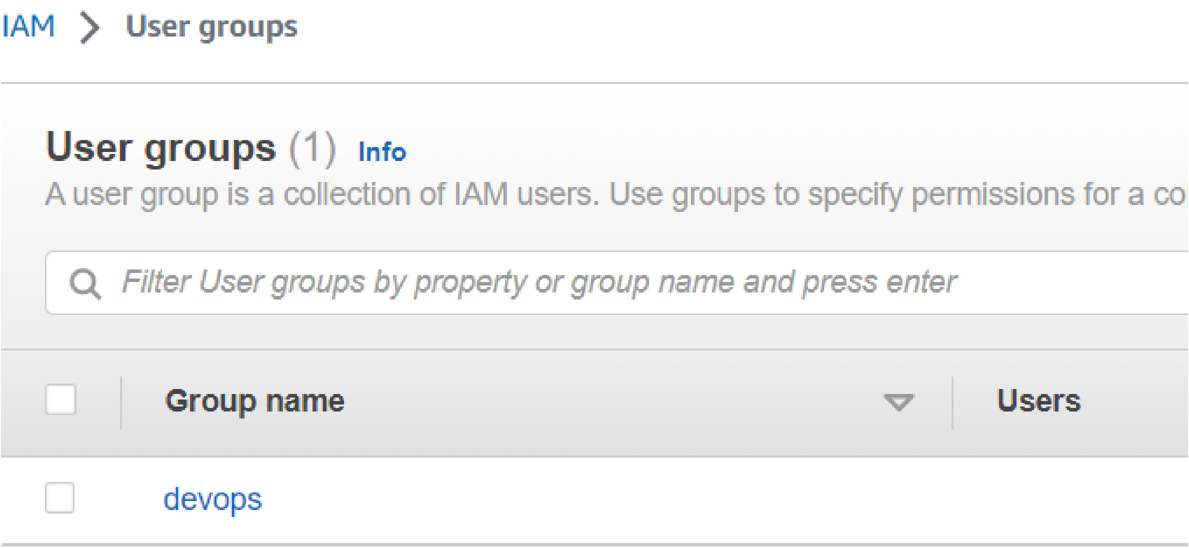


User groups (0) Info
A user group is a collection of IAM users. Use groups to specify permissions for a collection of users.

Filter User groups by property or group name and press enter

Group name	Users	Permissions	Creation time

No resources to display


IAM > User groups

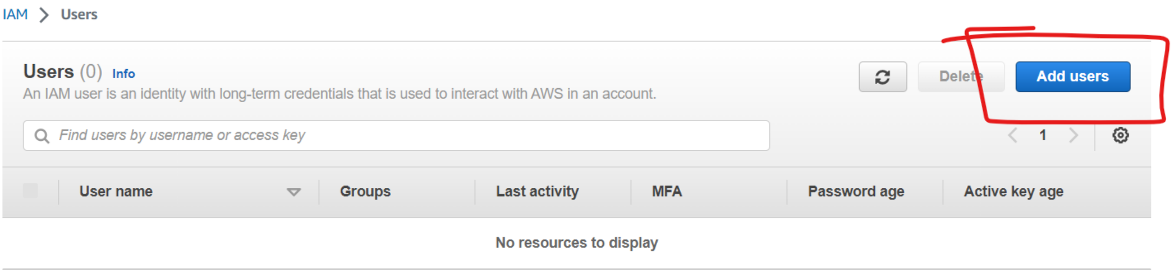
User groups (1) Info

A user group is a collection of IAM users. Use groups to specify permissions for a co

Filter User groups by property or group name and press enter

□	Group name	Users
□	devops	

Create user


IAM > Users

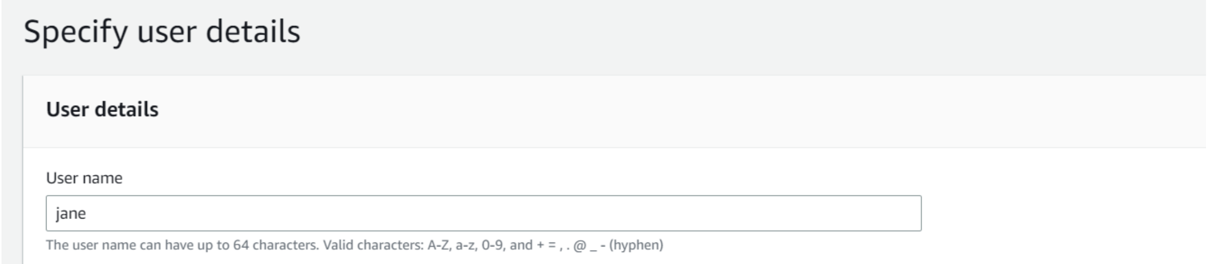
Users (0) Info

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

□	User name	Groups	Last activity	MFA	Password age	Active key age

No resources to display

Specify the user details

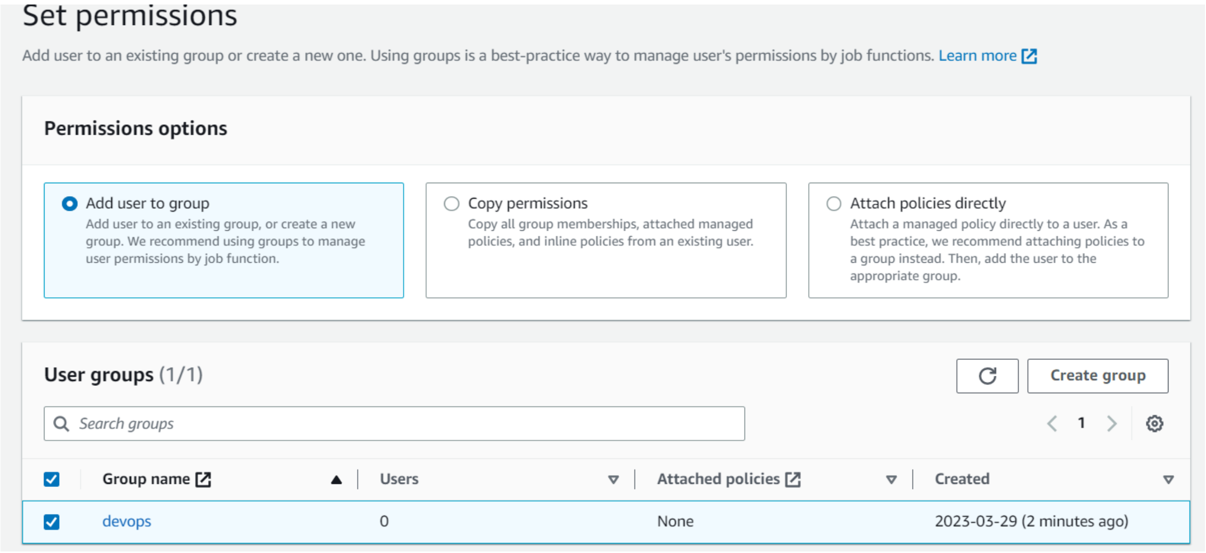

Specify user details

User details

User name: jane

The user name can have up to 64 characters. Valid characters: A-Z, a-z, 0-9, and + = , . @ _ - (hyphen)

Add the user to the group as well



Set permissions

Add user to an existing group or create a new one. Using groups is a best-practice way to manage user's permissions by job functions. Learn more

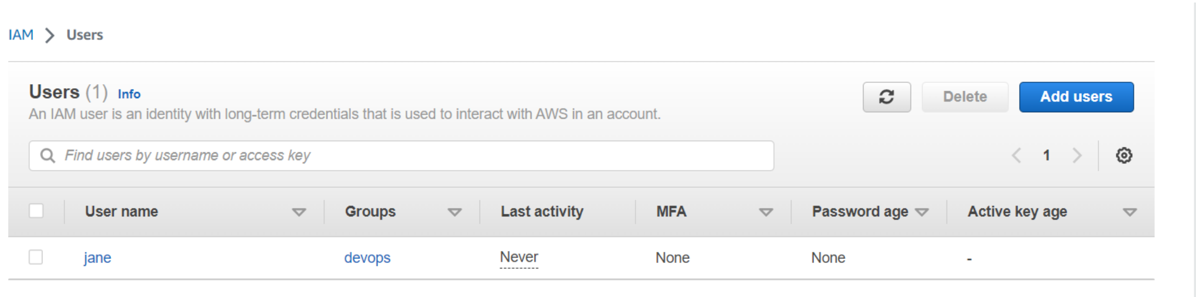
Permissions options

- ● Add user to group
  Add user to an existing group, or create a new group. We recommend using groups to manage user permissions by job function.
- ○ Copy permissions
  Copy all group memberships, attached managed policies, and inline policies from an existing user.
- ○ Attach policies directly
  Attach a managed policy directly to a user. As a best practice, we recommend attaching policies to a group instead. Then, add the user to the appropriate group.

User groups (1/1)

☑	Group name	Users	Attached policies	Created
☑	devops	0	None	2023-03-29 (2 minutes ago)

User is created and added to the group

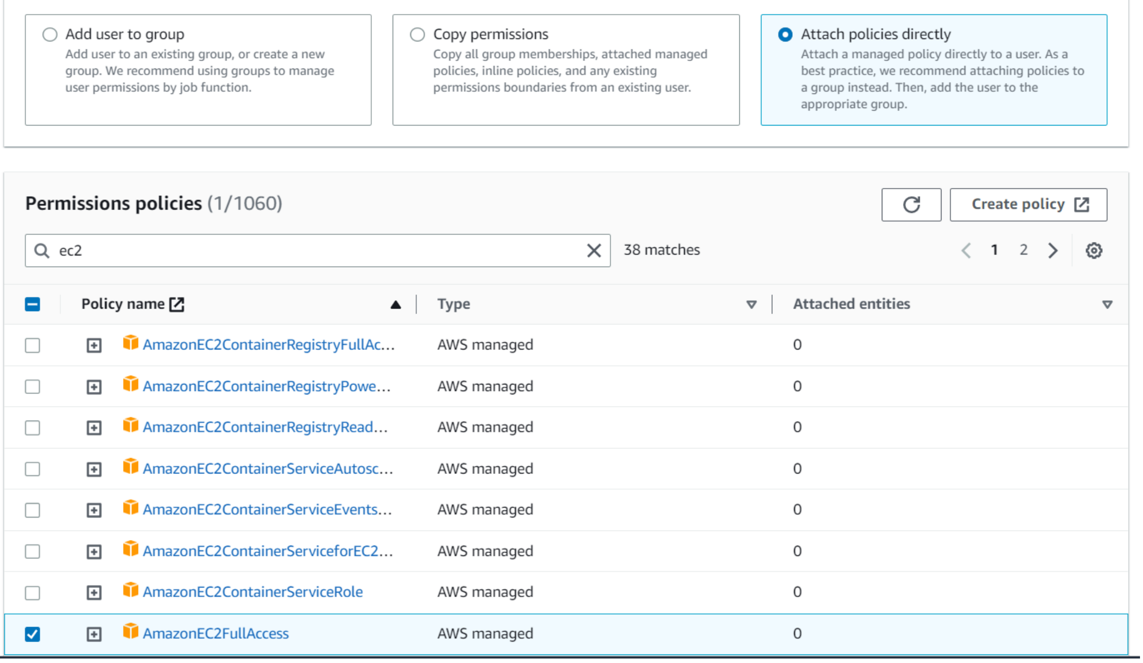


Users (1) Info

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

□	User name	Groups	Last activity	MFA	Password age	Active key age
□	jane	devops	Never	None	None	-

Click on the username and we attach policies giving full access to EC2



- ○ Add user to group
  Add user to an existing group, or create a new group. We recommend using groups to manage user permissions by job function.
- ○ Copy permissions
  Copy all group memberships, attached managed policies, inline policies, and any existing permissions boundaries from an existing user.
- ● Attach policies directly
  Attach a managed policy directly to a user. As a best practice, we recommend attaching policies to a group instead. Then, add the user to the appropriate group.

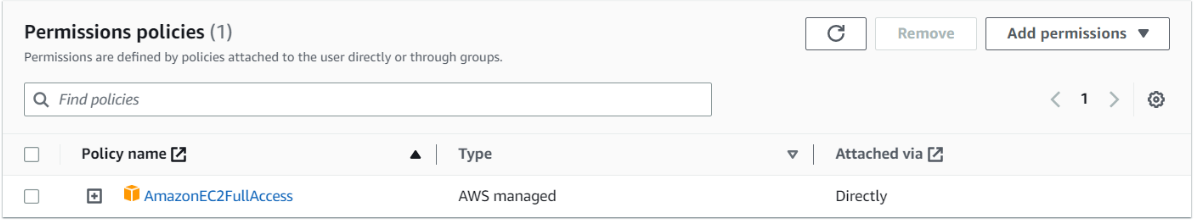
Permissions policies (1/1060)

Search: ec2 — 38 matches

☐	Policy name	Type	Attached entities
□	AmazonEC2ContainerRegistryFullAc...	AWS managed	0
□	AmazonEC2ContainerRegistryPowe...	AWS managed	0
□	AmazonEC2ContainerRegistryRead...	AWS managed	0
□	AmazonEC2ContainerServiceAutosc...	AWS managed	0
□	AmazonEC2ContainerServiceEvents...	AWS managed	0
□	AmazonEC2ContainerServiceforEC2...	AWS managed	0
□	AmazonEC2ContainerServiceRole	AWS managed	0
☑	AmazonEC2FullAccess	AWS managed	0

Thus, policy has been added

Hrishikesh Tonge | 230340325015
Set B

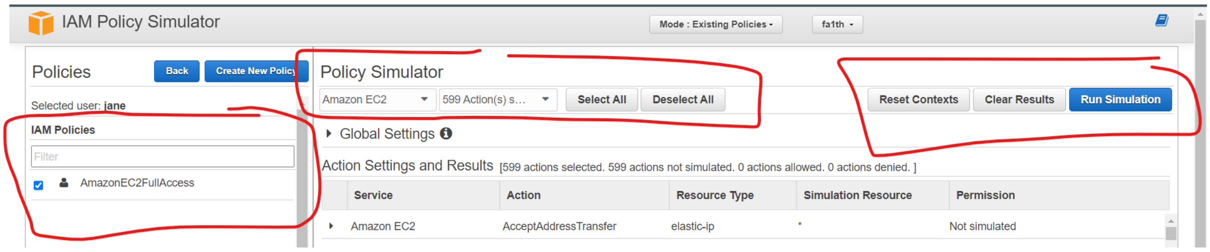


Permissions policies (1)

Permissions are defined by policies attached to the user directly or through groups.

□	Policy name	Type	Attached via
□	⊞ AmazonEC2FullAccess	AWS managed	Directly

Now we go to IAM Policy Simulator
And click on the user we created - jane
Here we can see the assigned policy to the user.



Selected user: jane

IAM Policies

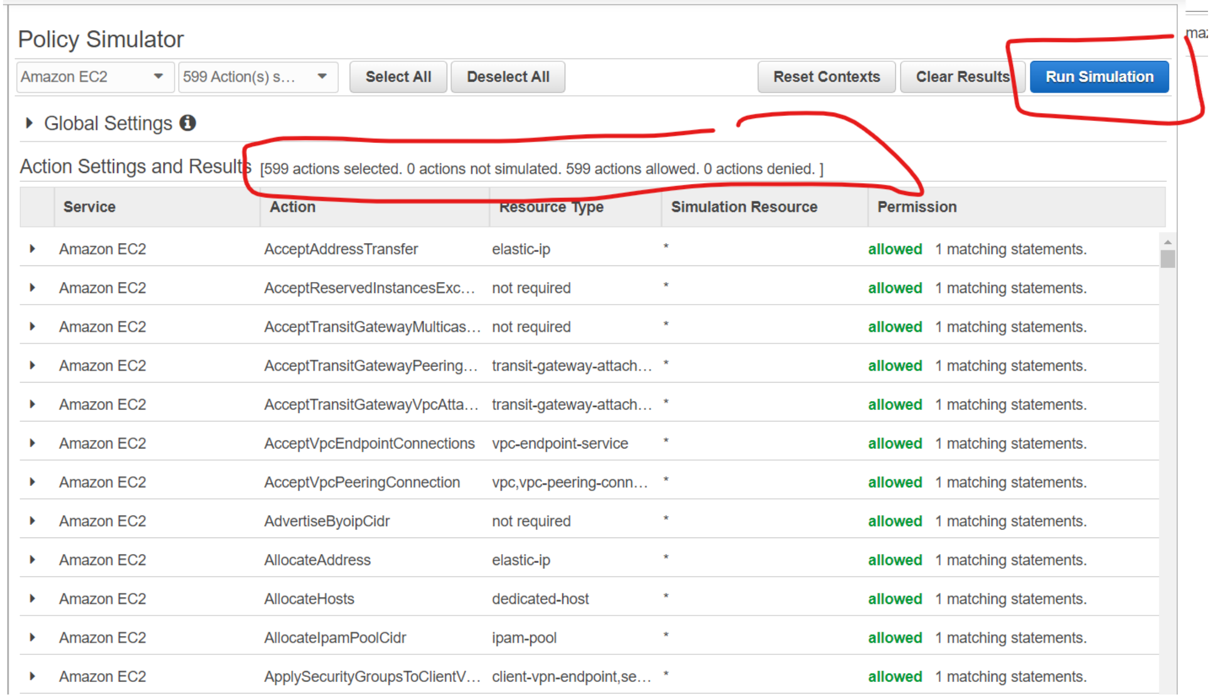
☑ AmazonEC2FullAccess

Amazon EC2 | 599 Action(s) s...

Action Settings and Results [599 actions selected. 599 actions not simulated. 0 actions allowed. 0 actions denied.]

Service	Action	Resource Type	Simulation Resource	Permission
Amazon EC2	AcceptAddressTransfer	elastic-ip	*	Not simulated

Once we click simulate. All the selected actions will be simulated and allowed to the user.



Amazon EC2 | 599 Action(s) s...

Action Settings and Results [599 actions selected. 0 actions not simulated. 599 actions allowed. 0 actions denied.]

Service	Action	Resource Type	Simulation Resource	Permission
Amazon EC2	AcceptAddressTransfer	elastic-ip	*	allowed 1 matching statements.
Amazon EC2	AcceptReservedInstancesExc...	not required	*	allowed 1 matching statements.
Amazon EC2	AcceptTransitGatewayMulticas...	not required	*	allowed 1 matching statements.
Amazon EC2	AcceptTransitGatewayPeering...	transit-gateway-attach...	*	allowed 1 matching statements.
Amazon EC2	AcceptTransitGatewayVpcAtta...	transit-gateway-attach...	*	allowed 1 matching statements.
Amazon EC2	AcceptVpcEndpointConnections	vpc-endpoint-service	*	allowed 1 matching statements.
Amazon EC2	AcceptVpcPeeringConnection	vpc,vpc-peering-conn...	*	allowed 1 matching statements.
Amazon EC2	AdvertiseByoipCidr	not required	*	allowed 1 matching statements.
Amazon EC2	AllocateAddress	elastic-ip	*	allowed 1 matching statements.
Amazon EC2	AllocateHosts	dedicated-host	*	allowed 1 matching statements.
Amazon EC2	AllocateIpamPoolCidr	ipam-pool	*	allowed 1 matching statements.
Amazon EC2	ApplySecurityGroupsToClientV...	client-vpn-endpoint,se...	*	allowed 1 matching statements.

Thus we have verified user permissions, all actions allowed because we have given EC2 full access.