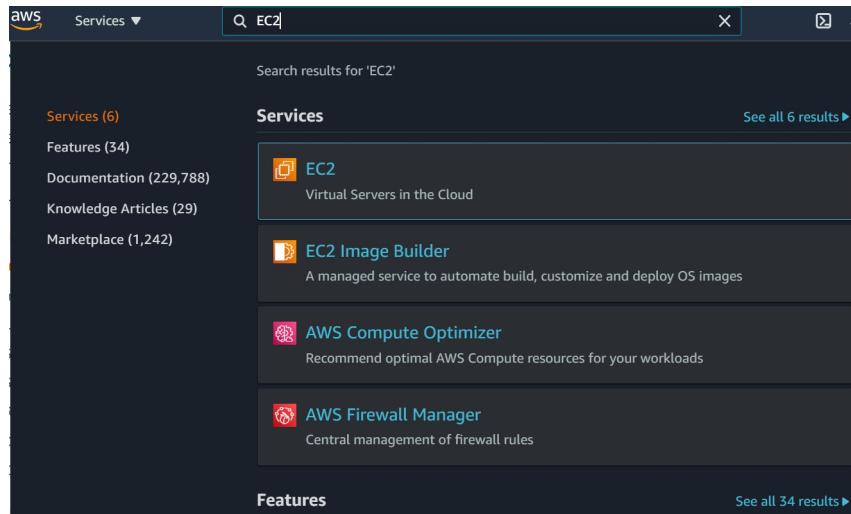


AWS_EC2

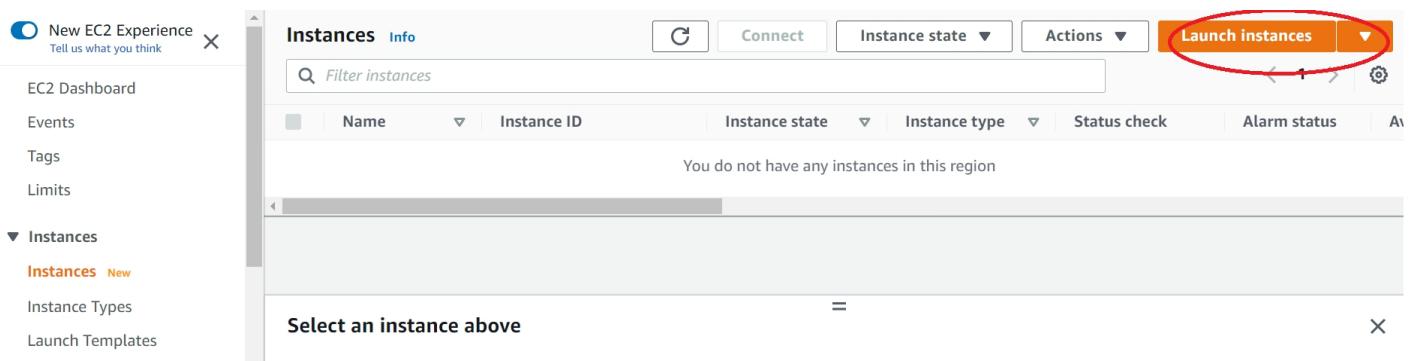
EC2 in action: Creating an EC2 instance with and without bootstrapping and attaching ENI's and playing with hibernate

performed by: **Rohit Sah**
Registration id- **SIRSS2129**

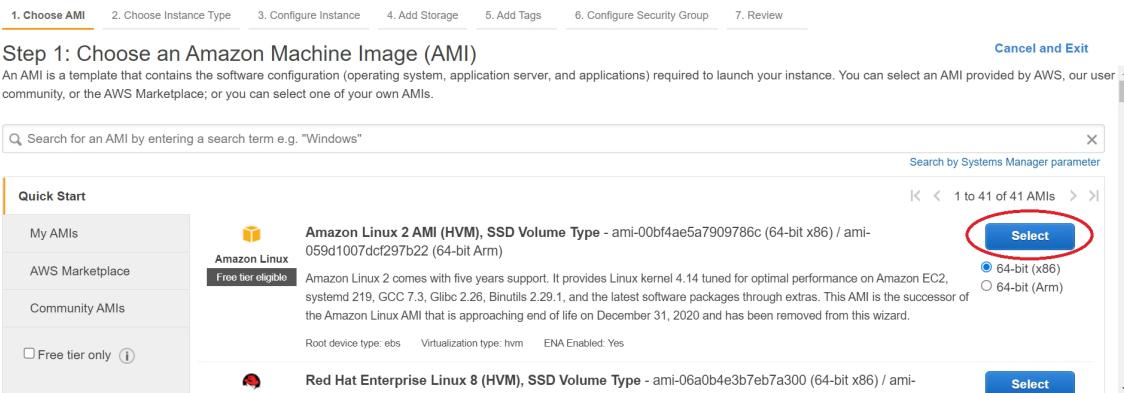
First go to the EC2 section



Then Click on Launch Instance



Select the AMI(Amazon Machine Image) that you want



Select the Instance type or the machine that you want

1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Add Tags 6. Configure Security Group 7. Review

Step 2: Choose an Instance Type

Amazon EC2 provides a wide selection of instance types optimized to fit different use cases. Instances are virtual servers that can run applications. They have varying combinations of CPU, memory, storage, and networking capacity, and give you the flexibility to choose the appropriate mix of resources for your applications. [Learn more](#) about instance types and how they can meet your computing needs.

Filter by: All instance families ▾ Current generation ▾ Show/Hide Columns

Currently selected: t2.micro (- ECUs, 1 vCPUs, 2.5 GHz, -, 1 GiB memory, EBS only)

	Family	Type	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance	IPv6 Support
<input type="checkbox"/>	t2	t2.nano	1	0.5	EBS only	-	Low to Moderate	Yes
<input checked="" type="checkbox"/>	t2	t2.micro Free tier eligible	1	1	EBS only	-	Low to Moderate	Yes
<input type="checkbox"/>	t2	t2.small	1	2	EBS only	-	Low to Moderate	Yes

Cancel Previous Review and Launch **Next: Configure Instance Details**

Then configure the instance details

1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Add Tags 6. Configure Security Group 7. Review

Step 3: Configure Instance Details

Configure the instance to suit your requirements. You can launch multiple instances from the same AMI, request Spot instances to take advantage of the lower pricing, assign an access management role to the instance, and more.

Number of instances Launch into Auto Scaling Group

Purchasing option Request Spot Instances

Network

Subnet

Auto-assign Public IP Use subnet setting (Enable)

Placement group Add instance to placement group

Capacity Reservation

Domain join directory

IAM role

Shutdown behavior

Stop - Hibernate behavior Enable hibernation as an additional stop behavior

Enable termination protection Protect against accidental termination

Monitoring Enable CloudWatch detailed monitoring
Additional charges apply.

Tenancy Additional charges will apply for dedicated tenancy.

Credit specification Unlimited

Cancel Previous Review and Launch **Next: Add Storage**

Select the storage details

1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Add Tags 6. Configure Security Group 7. Review

Step 4: Add Storage

Your instance will be launched with the following storage device settings. You can attach additional EBS volumes and instance store volumes to your instance, or edit the settings of the root volume. You can also attach additional EBS volumes after launching an instance, but not instance store volumes. [Learn more](#) about storage options in Amazon EC2.

Volume Type	Device	Snapshot	Size (GiB)	Volume Type	IOPS	Throughput (MB/s)	Delete on Termination	Encryption
Root	/dev/xvda	snap-039680514ac8c6b1a	<input type="text" value="8"/>	General Purpose SSD (gp2)	100 / 3000	N/A	<input checked="" type="checkbox"/>	<input type="checkbox"/> Not Encrypted

Add New Volume

Free tier eligible customers can get up to 30 GB of EBS General Purpose (SSD) or Magnetic storage. [Learn more](#) about free usage tier eligibility and usage restrictions.

Cancel Previous Review and Launch **Next: Add Tags**

Add a tag to the EC2 machine but it is optional

1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Add Tags 6. Configure Security Group 7. Review

Step 5: Add Tags

A tag consists of a case-sensitive key-value pair. For example, you could define a tag with key = Name and value = Webserver.

A copy of a tag can be applied to volumes, instances or both.

Tags will be applied to all instances and volumes. [Learn more](#) about tagging your Amazon EC2 resources.

Key	(128 characters maximum)	Value	(256 characters maximum)	Instances	Volumes	Network Interfaces
Machine	Test			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

[Add another tag](#) (Up to 50 tags maximum)

Cancel Previous **Review and Launch** Next: Configure Security Group

Now the most important part is to configure the security groups, here we are setting an SSH from our own IP and HTTP from everywhere so everyone can access the machine through the IP but only we can SSH

1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Add Tags 6. Configure Security Group 7. Review

Step 6: Configure Security Group

A security group is a set of firewall rules that control the traffic for your instance. On this page, you can add rules to allow specific traffic to reach your instance. For example, if you want to set up a web server and allow Internet traffic to reach your instance, add rules that allow unrestricted access to the HTTP and HTTPS ports. You can create a new security group or select from an existing one below.

[Learn more](#) about Amazon EC2 security groups.

Assign a security group: Create a new security group
 Select an existing security group

Security group name: SGforEC2

Description: securityGroupForEC2

Type	Protocol	Port Range	Source	Description
SSH	TCP	22	My IP <input style="border: none; background-color: transparent; font-size: small;" type="button" value="..."/> 223.238.122.133/32	sshForAdmin
HTTP	TCP	80	Anywhere <input style="border: none; background-color: transparent; font-size: small;" type="button" value="..."/> 0.0.0.0/0, ::/0	HttpForEveryone

[Add Rule](#)

Cancel Previous **Review and Launch**

Now we set a key pair and download it for ssh into our instance from our terminal

1. Choose AMI 2. Choose Instance Type

Step 7: Review Instance Launch

Please review your instance launch details.

AMI Details

Amazon Linux 2 AMI (HVM)

Free tier eligible

Amazon Linux 2 comes with IPv4 and IPv6 support, kernel version 2.29.1, and the latest software packages.

Root Device Type: ebs Virtualization Type: HVM

Instance Type

Instance Type	ECUs
t2.micro	-

Security Groups

Select an existing key pair or create a new key pair

A key pair consists of a **public key** that AWS stores, and a **private key file** that you store. Together, they allow you to connect to your instance securely. For Windows AMIs, the private key file is required to obtain the password used to log into your instance. For Linux AMIs, the private key file allows you to securely SSH into your instance.

Note: The selected key pair will be added to the set of keys authorized for this instance. Learn more about [removing existing key pairs from a public AMI](#).

You have to download the **private key file** (*.pem file) before you can continue. **Store it in a secure and accessible location**. You will not be able to download the file again after it's created.

Cancel **Launch Instances**

We can see the listed instance that we just created

The screenshot shows the AWS EC2 Instances page. On the left sidebar, under 'Instances', 'Instances' is selected. In the main content area, there is one instance listed:

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Avg.
-	i-018efec048689dff4	Running	t2.micro	Initializing	No alarms	ap-

A modal window titled 'Select an instance above' is open at the bottom of the list.

Now to ssh from our local machine to the EC2 instance we got the terminal and enter the command as shown below with the user as ec2-user@user and at the public IP of the instance

```
\Desktop>ssh -i "ec2_test_instance.pem" ec2-user@13.127.243.87
[ec2-user@ip-172-31-36-66 ~]$
```

Now we will launch a web server from our instance so first we install all the dependencies

```
[ec2-user@ip-172-31-36-66 ~]$ sudo su
[root@ip-172-31-36-66 ec2-user]# yum update -y
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
amzn2-core
Resolving Dependencies
--> Running transaction check
---> Package chrony.x86_64 0:4.0-3.amzn2.0.1 will be updated
---> Package chrony.x86_64 0:4.0-3.amzn2.0.2 will be an update
---> Package grub2.x86_64 1:2.02-35.amzn2.0.4 will be obsoleted
---> Package grub2.x86_64 1:2.06-2.amzn2.0.1 will be obsoleting
---> Package grub2-common.noarch 1:2.02-35.amzn2.0.4 will be updated
---> Package grub2-common.noarch 1:2.06-2.amzn2.0.1 will be an update
---> Package grub2-pc.x86_64 1:2.02-35.amzn2.0.4 will be updated
---> Package grub2-pc.x86_64 1:2.06-2.amzn2.0.1 will be obsoleting
---> Package grub2-pc-modules.noarch 1:2.02-35.amzn2.0.4 will be updated
---> Package grub2-pc-modules.noarch 1:2.06-2.amzn2.0.1 will be an update
---> Package grub2-tools.x86_64 1:2.02-35.amzn2.0.4 will be obsoleted
---> Package grub2-tools.x86_64 1:2.06-2.amzn2.0.1 will be obsoleting
---> Package grub2-tools-efi.x86_64 1:2.06-2.amzn2.0.1 will be obsoleting
```

```

python2-rpm.x86_64 0:4.11.3-40.amzn2.0.6           rpm.x86_64 0:4.11.3-40.amzn2.0.6           rpm-build-libs.x86_64 0:4.11.3-4
rpm-libs.x86_64 0:4.11.3-40.amzn2.0.6             rpm-plugin-systemd-inhibit.x86_64 0:4.11.3-40.amzn2.0.6
Replaced:
grub2.x86_64 1:2.02-35.amzn2.0.4                  grub2-tools.x86_64 1:2.02-35.amzn2.0.4

Complete!
[root@ip-172-31-36-66 ec2-user]#
[root@ip-172-31-36-66 ec2-user]#
[root@ip-172-31-36-66 ec2-user]# yum install -y httpd.x86_64
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
Resolving Dependencies
--> Running transaction check
--> Package httpd.x86_64 0:2.4.48-2.amzn2 will be installed
--> Processing Dependency: httpd-tools = 2.4.48-2.amzn2 for package: httpd-2.4.48-2.amzn2.x86_64
--> Processing Dependency: httpd-filesystem = 2.4.48-2.amzn2 for package: httpd-2.4.48-2.amzn2.x86_64
--> Processing Dependency: system-logos-httpd for package: httpd-2.4.48-2.amzn2.x86_64
--> Processing Dependency: mod_http2 for package: httpd-2.4.48-2.amzn2.x86_64
--> Processing Dependency: httpd-filesystem for package: httpd-2.4.48-2.amzn2.x86_64
--> Processing Dependency: /etc/mime.types for package: httpd-2.4.48-2.amzn2.x86_64
--> Processing Dependency: libaprutil-1.so.0()(64bit) for package: httpd-2.4.48-2.amzn2.x86_64
--> Processing Dependency: libapr-1.so.0()(64bit) for package: httpd-2.4.48-2.amzn2.x86_64
--> Running transaction check
--> Package apr.x86_64 0:1.6.3-5.amzn2.0.2 will be installed
--> Package apr-util.x86_64 0:1.6.1-5.amzn2.0.2 will be installed
--> Processing Dependency: apr-util-bdb(x86-64) = 1.6.1-5.amzn2.0.2 for package: apr-util-1.6.1-5.amzn2.0.2.x86_64
--> Package generic-logos-httpd.noarch 0:18.0.0-4.amzn2 will be installed
--> Package httpd-filesystem.noarch 0:2.4.48-2.amzn2 will be installed
--> Package httpd-tools.x86_64 0:2.4.48-2.amzn2 will be installed
--> Package mailcap.noarch 0:2.1.41-2.amzn2 will be installed
--> Package mod_http2.x86_64 0:1.15.19-1.amzn2.0.1 will be installed
--> Running transaction check
--> Package apr-util-bdb.x86_64 0:1.6.1-5.amzn2.0.2 will be installed
--> Finished Dependency Resolution

```

```

root@ip-172-31-36-66:/home/ec2-user
[root@ip-172-31-36-66 ec2-user]#
[root@ip-172-31-36-66 ec2-user]# systemctl start httpd.service
[root@ip-172-31-36-66 ec2-user]#
[root@ip-172-31-36-66 ec2-user]# systemctl enable httpd.service
Created symlink from /etc/systemd/system/multi-user.target.wants/httpd.service to /usr/lib/systemd/system/httpd.service.
[root@ip-172-31-36-66 ec2-user]#
[root@ip-172-31-36-66 ec2-user]# echo "Hello World from $(hostname -f)" > /var/www/html/index.html
[root@ip-172-31-36-66 ec2-user]#
[root@ip-172-31-36-66 ec2-user]#
[root@ip-172-31-36-66 ec2-user]#

```

Now as we have installed and configured our web server we can view it on the HTTP rule from the browser



Hello World from ip-172-31-36-66.ap-south-1.compute.internal

Now accessing the same site using bootstrapping

Following the same steps from the previous topic

The screenshot shows the AWS search interface with 'EC2' typed into the search bar. The results are categorized under 'Services' and 'Features'. Under 'Services', 'EC2' is highlighted with a blue box, described as 'Virtual Servers in the Cloud'. Other services listed include 'EC2 Image Builder', 'AWS Compute Optimizer', and 'AWS Firewall Manager'. Under 'Features', there is a link to 'See all 34 results'.

The screenshot shows the AWS EC2 Instances page. On the left, there's a sidebar with options like 'EC2 Dashboard', 'Events', 'Tags', 'Limits', and 'Instances' (which is expanded to show 'Instances New', 'Instance Types', and 'Launch Templates'). The main area shows a table with columns: Name, Instance ID, Instance state, Instance type, Status check, and Alarm status. A message at the top says 'You do not have any instances in this region'. At the bottom, it says 'Select an instance above'.

1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Add Tags 6. Configure Security Group 7. Review

Step 1: Choose an Amazon Machine Image (AMI)

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. You can select an AMI provided by AWS, our user community, or the AWS Marketplace; or you can select one of your own AMIs.

The screenshot shows the 'Step 1: Choose an Amazon Machine Image (AMI)' wizard. At the top, there's a search bar with placeholder text 'Search for an AMI by entering a search term e.g. "Windows"'. Below it, a 'Quick Start' sidebar lists 'My AMIs', 'AWS Marketplace', 'Community AMIs', and a checked 'Free tier only' checkbox. The main area displays two AMI options: 'Amazon Linux 2 AMI (HVM), SSD Volume Type - ami-00bf4ae5a7909786c (64-bit x86) / ami-059d1007dcf297b22 (64-bit Arm)' and 'Red Hat Enterprise Linux 8 (HVM), SSD Volume Type - ami-06a0b4e3b7eb7a300 (64-bit x86) / ami-06b-a0b-0b4-e3b-7eb-7a300 (64-bit Arm)'. Each option has a 'Select' button to its right. The 'Amazon Linux 2' entry also includes a note about five years support and being the successor of the Amazon Linux AMI.

Step 2: Choose an Instance Type

Amazon EC2 provides a wide selection of instance types optimized to fit different use cases. Instances are virtual servers that can run applications. They have varying combinations of CPU, memory, storage, and networking capacity, and give you the flexibility to choose the appropriate mix of resources for your applications. [Learn more](#) about instance types and how they can meet your computing needs.

Filter by: All instance families ▾ Current generation ▾ Show/Hide Columns

Currently selected: t2.micro (- ECUs, 1 vCPUs, 2.5 GHz, -, 1 GiB memory, EBS only)

	Family	Type	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance	IPv6 Support
<input type="checkbox"/>	t2	t2.nano	1	0.5	EBS only	-	Low to Moderate	Yes
<input checked="" type="checkbox"/>	t2	t2.micro <small>Free tier eligible</small>	1	1	EBS only	-	Low to Moderate	Yes
<input type="checkbox"/>	t2	t2.small	1	2	EBS only	-	Low to Moderate	Yes

Cancel Previous Review and Launch **Next: Configure Instance Details**

Now the main part where we are using the bootstrapping method or the user-data part here we will add all the commands that are to be executed when the machine boots p will all the commands in a new line

Step 3: Configure Instance Details

Advanced Details

Enclave Enable

Metadata accessible Enabled

Metadata version V1 and V2 (token optional)

Metadata token response hop limit 1

User data As text As file Input is already base64 encoded

```
#!/bin/bash
sudo su
yum update -y
yum install -y httpd.x86_64
systemctl start httpd.service
systemctl enable httpd.service
echo "Hello World from $(hostname -f)" > /var/www/html/index.html
```

Cancel Previous Review and Launch **Next: Add Storage**

Using the same security group details as before

Step 6: Configure Security Group

A security group is a set of firewall rules that control the traffic for your instance. On this page, you can add rules to allow specific traffic to reach your instance. For example, if you want to set up a web server and allow Internet traffic to reach your instance, add rules that allow unrestricted access to the HTTP and HTTPS ports. You can create a new security group or select from an existing one below. [Learn more](#) about Amazon EC2 security groups.

Assign a security group: Create a new security group Select an existing security group

Security Group ID	Name	Description	Actions
<input type="checkbox"/> sg-3342b44f	default	default VPC security group	Copy to new
<input checked="" type="checkbox"/> sg-042f3e95e8c375179	launch-wizard-1	launch-wizard-1 created 2021-07-26T21:38:47.358+05:30	Copy to new

Type	Protocol	Port Range	Source	Description
HTTP	TCP	80	0.0.0.0/0	HttpForEveryone
HTTP	TCP	80	::/0	HttpForEveryone
SSH	TCP	22	223.238.122.133/32	sshForAdmin

Cancel Previous **Review and Launch**

We can verify these when we ssh into our instance and hit **httpd -v** we can see that all our services have been installed already that was because of the bootstrapping that we used while launching the instance

```
ssh -i "ec2_test_instance.pem" ec2-user@65.0.95.84
The authenticity of host '65.0.95.84 (65.0.95.84)' can't be established.
ECDSA key fingerprint is SHA256:1qDTa3JtxGaUtryfcSJm2Qhnu3TPiG6cGBzgQGJ5LmI.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '65.0.95.84' (ECDSA) to the list of known hosts.

[ec2-user@ip-172-31-46-242 ~]$ httpd -v
Server version: Apache/2.4.48 ()
Server built: Jun 25 2021 18:53:37
[ec2-user@ip-172-31-46-242 ~]$
```

Now we will be playing with the ENI and we will be creating a new ENI and will be attaching it to our instance and verify it

Snapshots
Lifecycle Manager New

Network & Security

- Security Groups
- Elastic IPs
- Placement Groups
- Key Pairs
- Network Interfaces**

Load Balancing

Network interfaces (1) [Info](#)

Name	Network interface ID	Subnet ID	VPC ID	Availability Zone
-	eni-025e890f974c63426	subnet-6574810e	vpc-fd905f96	ap-south-1a

Actions [Create network interface](#)

Now when we check the ENI attached to instance before we only see one ENI that is 172.31.46.242

```
ssh -i "ec2_test_instance.pem" ec2-user@65.0.95.84
Last login: Mon Jul 26 16:54:33 2021 from 223.238.122.133

[ec2-user@ip-172-31-46-242 ~]$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 9001
    inet 172.31.46.242 netmask 255.255.240.0 broadcast 172.31.47.255
        ether 02:5a:84:61:0c:d8 txqueuelen 1000  (Ethernet)
        RX packets 66816 bytes 96356334 (91.8 MiB)
        RX errors 0 dropped 0 overruns 0 frame 0
        TX packets 5020 bytes 351337 (343.1 KiB)
        TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
        loop txqueuelen 1000  (Local Loopback)
        RX packets 32 bytes 2592 (2.5 KiB)
        RX errors 0 dropped 0 overruns 0 frame 0
        TX packets 32 bytes 2592 (2.5 KiB)
        TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

[ec2-user@ip-172-31-46-242 ~]$
```

Now to create an ENI we migrate into the ENI section from the EC2 dashboard and fill the details a our requirement

≡ **Create network interface**

An elastic network interface is a logical networking component in a VPC that represents a virtual network card.

Details [Info](#)

Description - *optional*
A descriptive name for the network interface.

Subnet
The subnet in which to create the network interface.
 X C

Private IPv4 address
The private IPv4 address to assign to the network interface.
 Auto-assign
 Custom

Elastic Fabric Adapter
 Enable

Security groups (1/2) [Info](#)

Find resources < 1 > ⚙

≡ Enable

Security groups (1/2) [Info](#)

Find resources < 1 > ⚙

<input type="checkbox"/>	Group ID	▲ Group name	▼ Description
<input checked="" type="checkbox"/>	sg-042f3e95e8c375179	launch-wizard-1	launch-wizard-1 created 2021-07-2...
<input type="checkbox"/>	sg-3342b44f	default	default VPC security group

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.

[Add new tag](#)

You can add 50 more tags

Cancel **Create network interface**

We can see that we have successfully created an ENI

The screenshot shows the AWS EC2 Network Interfaces page. A green header bar at the top indicates "Successfully created network interface eni-0cad2a96ea1fab9a9". The main table lists two network interfaces:

Name	Network interface ID	Subnet ID	VPC ID	Availability Zone
-	eni-025e890f974c63426	subnet-6574810e	vpc-fd905f96	ap-south-1a
<input checked="" type="checkbox"/>	eni-0cad2a96ea1fab9a9	subnet-6574810e	vpc-fd905f96	ap-south-1a

Below the table, a section titled "Network interface: eni-0cad2a96ea1fab9a9" shows tabs for "Details", "Flow logs", and "Tags".

Now to attach the ENI to the instance click the attach option and select the instance that we want to attach with

The screenshot shows the same AWS EC2 Network Interfaces page as before, but with a different context. The ENI "eni-0cad2a96ea1fab9a9" is selected. The "Actions" dropdown menu is open, and the "Attach" option is highlighted.

We can view the instance details and see the ENI that are attached to it

The screenshot shows the AWS EC2 Instances page. A green header bar at the top indicates "Successfully created network interface eni-0cad2a96ea1fab9a9". The main table lists instances:

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Ava...
-	i-018efec048689dff4	Terminated	t2.micro	-	No alarms	ap...
<input checked="" type="checkbox"/>	i-0d9bb88e37fd361fc	Running	t2.micro	2/2 checks passed	No alarms	ap...

Below the table, a section titled "Instance: i-0d9bb88e37fd361fc" shows a table of network interfaces:

Interface ID	Description	Public IPv4 address	Private IPv4 address	Private IPv4 DNS
eni-025e890f974c63...	-	65.0.95.84	172.31.46.242	ip-172-31-46-242.ap...
eni-0cad2a96ea1fab...	test_eni_for_ec2	-	172.31.35.222	ip-172-31-35-222.ap...

Now to verify it we can also see the ENI that we have attached to instance by ssh and hitting the command `ifconfig` to see the ENIs and we can see that there is a new ENI with IP 172.31.35.222

```
[ec2-user@ip-172-31-46-242:~]
    inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop txqueuelen 1000  (Local Loopback)
        RX packets 32 bytes 2592 (2.5 KiB)
        RX errors 0 dropped 0 overruns 0 frame 0
        TX packets 32 bytes 2592 (2.5 KiB)
        TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

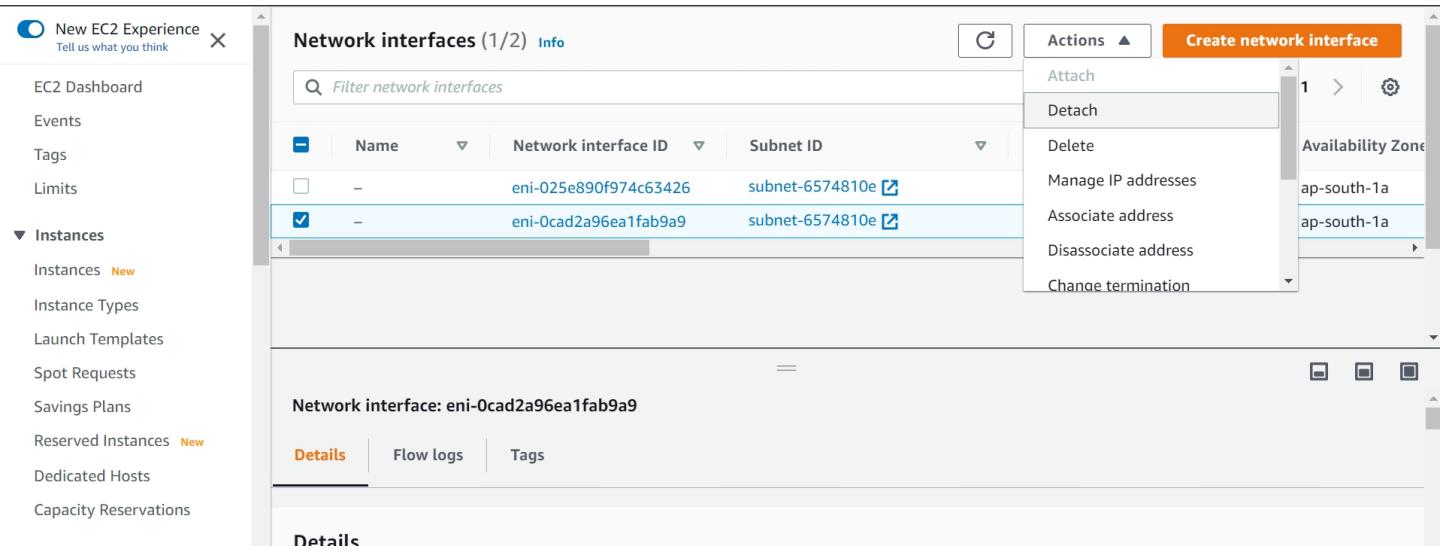
[ec2-user@ip-172-31-46-242 ~]$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 9001
    inet 172.31.46.242 netmask 255.255.240.0 broadcast 172.31.47.255
    inet6 fe80::5a:84ff:fe61:cd8  prefixlen 64  scopeid 0x20<link>
        ether 02:5a:84:61:0c:d8  txqueuelen 1000  (Ethernet)
        RX packets 66988 bytes 96375673 (91.9 MiB)
        RX errors 0 dropped 0 overruns 0 frame 0
        TX packets 5201 bytes 371332 (362.6 KiB)
        TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

eth1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 9001
    inet 172.31.35.222 netmask 255.255.240.0 broadcast 172.31.47.255
    inet6 fe80::8a:7bff:fe43:de40  prefixlen 64  scopeid 0x20<link>
        ether 02:8a:7b:43:de:40  txqueuelen 1000  (Ethernet)
        RX packets 13 bytes 1530 (1.4 KiB)
        RX errors 0 dropped 0 overruns 0 frame 0
        TX packets 16 bytes 2008 (1.9 KiB)
        TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

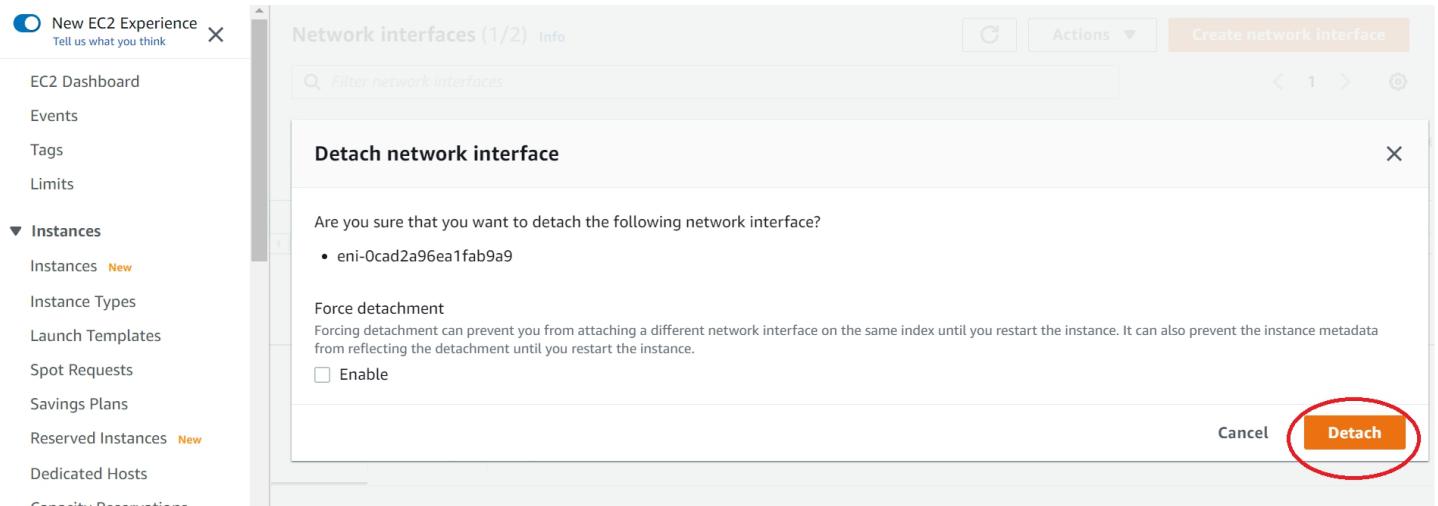
lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop txqueuelen 1000  (Local Loopback)
        RX packets 32 bytes 2592 (2.5 KiB)
        RX errors 0 dropped 0 overruns 0 frame 0
        TX packets 32 bytes 2592 (2.5 KiB)
        TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

[ec2-user@ip-172-31-46-242 ~]$
```

To detach the NEI we can hit the Detach option like this



And confirm the detachment for the ENI



To verify it we can again check it through the command line

```
c:\ ec2-user@ip-172-31-46-242:~  
      ether 02:8a:7b:43:de:40 txqueuelen 1000 (Ethernet)  
        RX packets 13 bytes 1530 (1.4 KiB)  
        RX errors 0 dropped 0 overruns 0 frame 0  
        TX packets 16 bytes 2008 (1.9 KiB)  
        TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0  
  
lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536  
      inet 127.0.0.1 netmask 255.0.0.0  
      inet6 ::1 prefixlen 128 scopeid 0x10<host>  
      loop txqueuelen 1000 (Local Loopback)  
        RX packets 32 bytes 2592 (2.5 KiB)  
        RX errors 0 dropped 0 overruns 0 frame 0  
        TX packets 32 bytes 2592 (2.5 KiB)  
        TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0  
  
[ec2-user@ip-172-31-46-242 ~]$  
[ec2-user@ip-172-31-46-242 ~]$ ifconfig  
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 9001  
      inet 172.31.46.242 netmask 255.255.240.0 broadcast 172.31.47.255  
      inet6 fe80::5a:84ff:fe61:cd8 prefixlen 64 scopeid 0x20<link>  
      ether 02:5a:84:61:0c:d8 txqueuelen 1000 (Ethernet)  
        RX packets 67030 bytes 96379994 (91.9 MiB)  
        RX errors 0 dropped 0 overruns 0 frame 0  
        TX packets 5250 bytes 378324 (369.4 KiB)  
        TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0  
  
lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536  
      inet 127.0.0.1 netmask 255.0.0.0  
      inet6 ::1 prefixlen 128 scopeid 0x10<host>  
      loop txqueuelen 1000 (Local Loopback)  
        RX packets 32 bytes 2592 (2.5 KiB)  
        RX errors 0 dropped 0 overruns 0 frame 0  
        TX packets 32 bytes 2592 (2.5 KiB)  
        TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0  
  
[ec2-user@ip-172-31-46-242 ~]$  
[ec2-user@ip-172-31-46-242 ~]$  
[ec2-user@ip-172-31-46-242 ~]$
```

Now we will playing with the hibernate feature of the EC2 instance, so we will keep all the setting same but in the instance details part we will hit/check the checkbox of Stop-hibernate behaviour for hibernate to work

Step 3: Configure Instance Details

Capacity Reservation

Domain join directory

IAM role

Shutdown behavior

Stop - Hibernate behavior To enable hibernation, space is allocated on the root volume to store the instance memory (RAM). Make sure that the root volume is large enough to store the RAM contents and accommodate your expected usage, e.g. OS, applications. To use hibernation, the root volume must be an encrypted EBS volume. [Learn more](#)

Enable termination protection Protect against accidental termination

Cancel **Previous** **Review and Launch** **Next: Add Storage**

We set up the storage details as hibernate uses encrypted EBS volumes so we have to encrypt the volume that we are creating

Step 4: Add Storage

Your instance will be launched with the following storage device settings. You can attach additional EBS volumes and instance store volumes to your instance, or edit the settings of the root volume. You can also attach additional EBS volumes after launching an instance, but not instance store volumes. [Learn more](#) about storage options in Amazon EC2.

Volume Type	Device	Snapshot	Size (GiB)	Volume Type	IOPS	Throughput (MB/s)	Delete on Termination	Encryption
Root	/dev/xvda	snap-039680514ac8c6b1a	8	General Purpose SSD (gp2)	100 / 3000	N/A	<input checked="" type="checkbox"/>	<input type="button" value="Not Encrypt"/>

Add New Volume

Free tier eligible customers can get up to 30 GB of EBS General Purpose (SSD) or Magnetic storage. [Learn more](#) about free usage tier eligibility and usage restrictions.

Cancel **Previous** **Review and Launch** **Next: Add Tags**

Using the same security group

Step 6: Configure Security Group

A security group is a set of network rules that control traffic for your instance. On this page, you can add rules to allow specific traffic to reach your instance. For example, if you want to set up a web server and allow Internet traffic to reach your instance, add rules that allow unrestricted access to the HTTP and HTTPS ports. You can create a new security group or select from an existing one below. [Learn more](#) about Amazon EC2 security groups.

Assign a security group: Create a new security group Select an existing security group

Security Group ID	Name	Description	Actions
<input type="checkbox"/> sg-3342b44f	default	default VPC security group	<input type="button" value="Copy to new"/>
<input checked="" type="checkbox"/> sg-042f3e95e8c375179	launch-wizard-1	launch-wizard-1 created 2021-07-26T21:38:47.358+05:30	<input type="button" value="Copy to new"/>

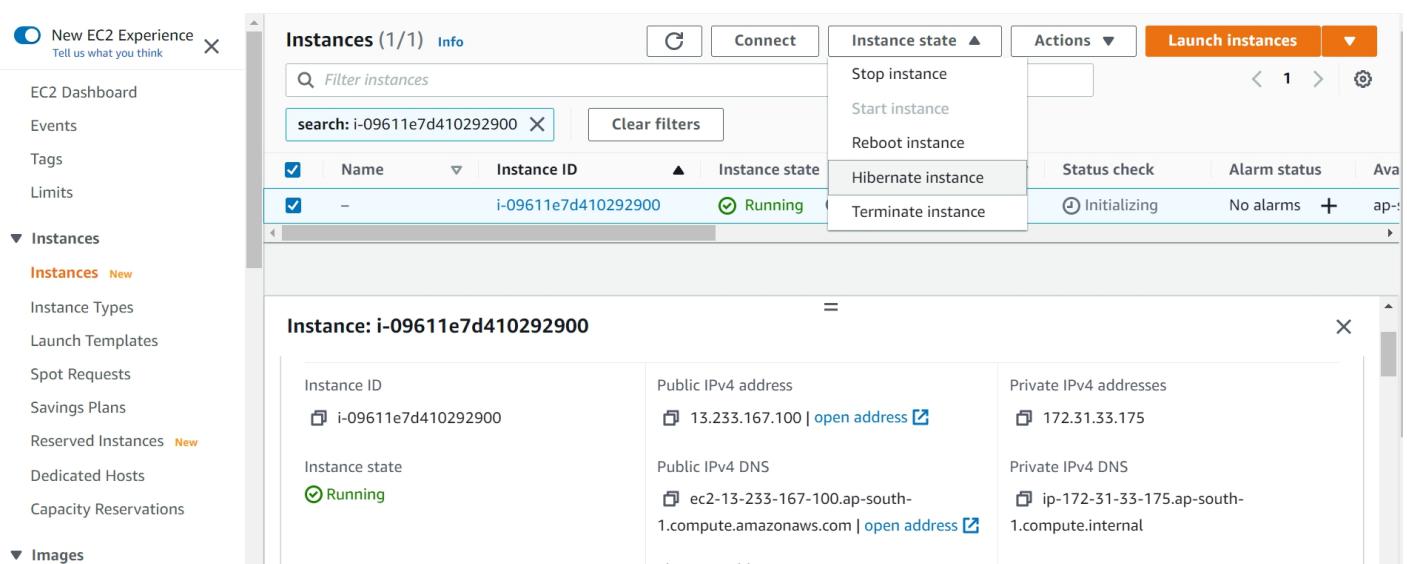
Type	Protocol	Port Range	Source	Description
HTTP	TCP	80	0.0.0.0/0	HttpForEveryone
HTTP	TCP	80	::/0	HttpForEveryone
SSH	TCP	22	223.238.122.133/32	sshForAdmin

Cancel **Previous** **Review and Launch**

Now we hit some commands to acquire some data so that we can see that when we use hibernate we can again regain the data of the instance

```
ec2-user@ip-172-31-33-175:~  
  
ssh -i "ec2_test_instance.pem" ec2-user@13.233.167.100  
The authenticity of host '13.233.167.100 (13.233.167.100)' can't be established.  
ECDSA key fingerprint is SHA256:CycBg2R3hwUs9zjVLWaZQQ01Z8XQxDz0K7cEX1dVQfI.  
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes  
Warning: Permanently added '13.233.167.100' (ECDSA) to the list of known hosts.  
  
_ _| _|_)  
_ | ( _ / Amazon Linux 2 AMI  
_ \_\_|_  
  
https://aws.amazon.com/amazon-linux-2/  
16 package(s) needed for security, out of 18 available  
Run "sudo yum update" to apply all updates.  
[ec2-user@ip-172-31-33-175 ~]$  
[ec2-user@ip-172-31-33-175 ~]$  
[ec2-user@ip-172-31-33-175 ~]$ uptime  
17:30:37 up 1 min, 1 user, load average: 1.67, 0.70, 0.25  
[ec2-user@ip-172-31-33-175 ~]$  
[ec2-user@ip-172-31-33-175 ~]$ uptime  
17:30:40 up 1 min, 1 user, load average: 1.67, 0.70, 0.25  
[ec2-user@ip-172-31-33-175 ~]$  
[ec2-user@ip-172-31-33-175 ~]$ uptime  
17:30:52 up 1 min, 1 user, load average: 1.30, 0.66, 0.25  
[ec2-user@ip-172-31-33-175 ~]$  
[ec2-user@ip-172-31-33-175 ~]$ uptime  
17:31:04 up 1 min, 1 user, load average: 1.10, 0.64, 0.25  
[ec2-user@ip-172-31-33-175 ~]$  
[ec2-user@ip-172-31-33-175 ~]$ uptime  
17:32:14 up 2 min, 1 user, load average: 0.34, 0.50, 0.23  
[ec2-user@ip-172-31-33-175 ~]$  
[ec2-user@ip-172-31-33-175 ~]$
```

Now we hit the hibernate option to hibernate the instance



The screenshot shows the AWS EC2 Instances page. On the left sidebar, under the 'Instances' section, 'Instances' is selected. The main area displays a table titled 'Instances (1/1)'. The table has columns for 'Name', 'Instance ID', and 'Instance state'. One row is visible, showing 'i-09611e7d410292900' as the Instance ID and 'Running' as the Instance state. To the right of the table, there is a detailed view for the instance 'i-09611e7d410292900'. This view includes sections for 'Instance ID' (i-09611e7d410292900), 'Public IPv4 address' (13.233.167.100), 'Private IPv4 addresses' (172.31.33.175), 'Instance state' (Running), 'Public IPv4 DNS' (ec2-13-233-167-100.ap-south-1.compute.amazonaws.com), and 'Private IPv4 DNS' (ip-172-31-33-175.ap-south-1.compute.internal). The 'Actions' dropdown menu is open, showing options: Stop instance, Start instance, Reboot instance, Hibernate instance (which is highlighted in blue), and Terminate instance.

Now again when we again start the instance and SSH into the instance and check the uptime and we can get our data back or we can say that the RAM is regained

```
ec2-user@ip-172-31-33-175:~  
  
      ssh -i "ec2_test_instance.pem" ec2-user@13.233.92.216  
The authenticity of host '13.233.92.216 (13.233.92.216)' can't be established.  
ECDSA key fingerprint is SHA256:CycBg2R3hwUs9zjVLWaZQQ01Z8XQxDz0K7cEX1dVQfI.  
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes  
Warning: Permanently added '13.233.92.216' (ECDSA) to the list of known hosts.  
Last login: Mon Jul 26 17:30:32 2021 from 223.238.122.133  
  
_ _|_ ( _|_ )  
_ | / Amazon Linux 2 AMI  
_ \_||_|  
  
https://aws.amazon.com/amazon-linux-2/  
16 package(s) needed for security, out of 18 available  
Run "sudo yum update" to apply all updates.  
[ec2-user@ip-172-31-33-175 ~]$  
[ec2-user@ip-172-31-33-175 ~]$  
[ec2-user@ip-172-31-33-175 ~]$ uptime  
17:36:08 up 6 min, 2 users, load average: 0.04, 0.32, 0.19  
[ec2-user@ip-172-31-33-175 ~]$  
[ec2-user@ip-172-31-33-175 ~]$  
[ec2-user@ip-172-31-33-175 ~]$
```

