

Virtual Private Cloud

Launching instances by using own VPC and Connecting Public instance to Private Instance

- To Create VPC go to AWS Console -> Services -> VPC and then click on VPCs

The screenshot shows the AWS VPC Service Health page. On the left, there's a sidebar with 'New VPC Experience' and a 'VPC Dashboard' section containing links for Your VPCs, Subnets, Route Tables, Internet Gateways, Egress Only Internet Gateways, Carrier Gateways, DHCP Options Sets, Elastic IPs, Managed Prefix Lists, Endpoints, Endpoint Services, NAT Gateways, and Peering Connections. The main area has tabs for 'Launch VPC Wizard' (highlighted in orange) and 'Launch EC2 Instances'. A note says 'Your Instances will launch in the US East (N. Virginia) region.' Below this is a 'Resources by Region' section with tables for VPCs, Subnets, Route Tables, Internet Gateways, Egress-only Internet Gateways, NAT Gateways, and Peering Connections, all listed under the N. Virginia region. To the right is a 'Service Health' panel with a table for 'Amazon EC2 - US East (N. Virginia)' showing 'Service is operating normally'. It also includes sections for 'Settings' (Zones, Console Experiments), 'Additional Information' (VPC Documentation, All VPC Resources, Forums, Report an Issue), and 'Transit Gateway Network Manager'.

- Click on create VPC button to create VPC

The screenshot shows the 'Your VPCs' page. The sidebar is identical to the previous one. The main area has a 'Your VPCs' table with columns for Name, VPC ID, State, IPv4 CIDR, and IPv6 CIDR. A 'Create VPC' button is located at the top right of the table. Below the table, a message says 'Select a VPC above' with three small icons for copy, cut, and paste.

- Give name Tag to VPC and IPv4 CIDR block range to VPC then click on Create VPC button.

Create VPC Info

A VPC is an isolated portion of the AWS cloud populated by AWS objects, such as Amazon EC2 instances.

VPC settings

Name tag - optional
Creates a tag with a key of 'Name' and a value that you specify.

IPv4 CIDR block Info
10.0.0.0/16

IPv6 CIDR block Info
 No IPv6 CIDR block
 Amazon-provided IPv6 CIDR block
 IPv6 CIDR owned by me

Tenancy Info
Default

Tags
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.

Key	Value - optional
<input type="text" value="Name"/>	<input type="text" value="MyVPC"/>

Add new tag
You can add 49 more tags.

Create VPC

- VPC Created.

You successfully created `vpc-050c4db9f25644fee / MyVPC`

VPC > Your VPCs > `vpc-050c4db9f25644fee`

Actions ▾

Details			
VPC ID <input type="text" value="vpc-050c4db9f25644fee"/>	State Available	DNS hostnames Disabled	DNS resolution Enabled
Tenancy Default	DHCP options set <code>dopt-7ede4604</code>	Route table <code>rtb-0b6f9ab3921c87e88</code>	Network ACL <code>acl-0a8eb0a9799f13d0a</code>
Default VPC No	IPv4 CIDR 10.0.0.0/16	IPv6 pool -	IPv6 CIDR (Network border group) -
Owner ID <input type="text" value="129516552661"/>			

- Now, Create Internet Gateway and attach it to VPC. First Open Internet Gateway Page.

The screenshot shows the AWS VPC Internet Gateways page. On the left, there's a sidebar with 'New VPC Experience' and a 'Select a VPC' dropdown. The main area has a table titled 'Internet gateways (1/1)'. It contains one row with the following data:

Name	Internet gateway ID	State	VPC ID
-	igw-39230142	Attached	vpc-a3e01fde

- Give name to Internet Gateway and click on create Internet Gateway

The screenshot shows the 'Create internet gateway' wizard. It has two main sections: 'Internet gateway settings' and 'Tags - optional'.

Internet gateway settings: A 'Name tag' field contains 'MyIG'.

Tags - optional: A table shows a single tag: 'Key' is 'Name' and 'Value - optional' is 'MyIG'. There are 'Add new tag' and 'Remove' buttons, and a note that 49 more tags can be added.

At the bottom are 'Cancel' and 'Create internet gateway' buttons.

- Internet Gateway created

The screenshot shows the AWS VPC Internet Gateways page again. A green banner at the top says 'The following internet gateway was created: igw-00452c93a981f9f68. You can now attach to a VPC to enable the VPC to communicate with the internet.' Below it, the main table shows the newly created gateway:

Internet gateway ID	State	VPC ID	Owner
igw-00452c93a981f9f68	Detached	-	129516552661

- Attaching IGW to VPC

The following internet gateway was created: igw-00452c93a981f9f68 . You can now attach to a VPC to enable the VPC to communicate with the Internet.

igw-00452c93a981f9f68 / MyIG

Internet gateway ID	State	VPC ID	Owner
igw-00452c93a981f9f68	Detached	-	129516552661

Actions

- Attach to VPC
- Detach from VPC
- Manage tags
- Delete

- Select VPC to attach IGW then click attach

Attach to VPC (igw-00452c93a981f9f68)

VPC
Attach an internet gateway to a VPC to enable the VPC to communicate with the internet. Specify the VPC to attach below.

Available VPCs
Attach the internet gateway to this VPC.

Select a VPC
vpc-050c4db9f25644fee - MyVPC

▶ AWS Command Line Interface command

Cancel **Attach internet gateway**

- IGW Attached to VPC

Internet gateway igw-00452c93a981f9f68 successfully attached to vpc-050c4db9f25644fee

igw-00452c93a981f9f68 / MyIG

Internet gateway ID	State	VPC ID	Owner
igw-00452c93a981f9f68	Attached	vpc-050c4db9f25644fee MyVPC	129516552661

- After attaching IGW to VPC open Subnets to created subnets and to attach subnets to VPC. Click Create Subnet button to create subnets

The screenshot shows the AWS VPC Subnets list page. On the left, there's a sidebar with 'New VPC Experience' and a search bar for 'vpc-050c4...'. The main area is titled 'Subnets (6) Info' with a 'Filter subnets' input field. A table lists six subnets:

Name	Subnet ID	State	VPC	IPv4 CIDR
-	subnet-356bc853	Available	vpc-a3e01fde	172.31.0.0/20
-	subnet-1ae4245	Available	vpc-a3e01fde	172.31.32.0/20
-	subnet-cec1b283	Available	vpc-a3e01fde	172.31.16.0/20
-	subnet-0de58003	Available	vpc-a3e01fde	172.31.64.0/20
-	subnet-45cc6564	Available	vpc-a3e01fde	172.31.80.0/20
-	subnet-4453a475	Available	vpc-a3e01fde	172.31.48.0/20

- Select VPC and Give name to Subnet. Select Availability Zone and IPv4 CIDR range then click Create Subnet Button to create subnet.

Create subnet

VPC

VPC ID: vpc-050c4db8125644fe (MyVPC)

Associated VPC CIDRs: 10.0.0.0/16

Subnet settings

Specify the CIDR blocks and Availability Zone for the subnet..

Subnet 1 of 1

Subnet name: MySubnet1

Availability Zone: US East (N. Virginia) / us-east-1a

IPv4 CIDR block: 10.0.1.0/24

Tags - optional:

Key: Name Value: MySubnet1

Add new tag

Create subnet

- Six subnets are created. I am created each Availability Zone has one subnet.

Subnet Name	Subnet ID	Status	VPC ID	CIDR Range
MySubnet6	subnet-01b75d186c42bd6c5	Available	vpc-050c4db9f25644fee	10.0.6.0/2
MySubnet5	subnet-0b1efc1adea6724bd	Available	vpc-050c4db9f25644fee	10.0.5.0/2
MySubnet3	subnet-0bd81e84a410aedf7	Available	vpc-050c4db9f25644fee	10.0.3.0/2
MySubnet1	subnet-0326249ae5495b711	Available	vpc-050c4db9f25644fee	10.0.1.0/2
MySubnet2	subnet-0e0a0f85878ec462d	Available	vpc-050c4db9f25644fee	10.0.2.0/2
MySubnet4	subnet-08546e0d11c35eda5	Available	vpc-050c4db9f25644fee	10.0.4.0/2

- After subnets, Create route tables. Click Create route table button to create route table.

Name	Route Table ID	Explicit subnet association	Edge associations	Main	VPC ID
	rtb-0b619ab3921c87e88	-	-	Yes	vpc-050c4db9f25644fee ...
	rtb-926d17ec	-	-	Yes	vpc-a3e01fde

- Give name and select VPC then click on Create button to create route table. I am creating two route tables one is PublicRT and another one is PrivateRT.

Create route table

A route table specifies how packets are forwarded between the subnets within your VPC, the internet, and your VPN connection.

Name tag: Info

VPC*: Actions Cancel Create

Key (128 characters maximum) | Value (256 characters maximum)

Name	<input type="text" value="PublicRT"/> Remove
------	---

Add Tag 49 remaining (Up to 50 tags maximum)

* Required

- PublicRT route table created.

Create route table

The following Route Table was created:

Route Table ID: rtb-03f4e5db9456938fe

Close

- Now creating another route table called PrivateRT.

Route Tables > Create route table

Create route table

A route table specifies how packets are forwarded between the subnets within your VPC, the internet, and your VPN connection.

Name tag	PrivateRT	
VPC*	vpc-050c4db9f25644fee	
Key (128 characters maximum) Value (256 characters maximum)		
<input type="text" value="Name"/> <input type="text" value="PrivateRT"/>		
<input type="button" value="Add Tag"/> 49 remaining (Up to 50 tags maximum)		
<small>* Required</small>		<input type="button" value="Cancel"/> <input type="button" value="Create"/>

- PrivateRT route table created.

Route Tables > Create route table

Create route table

The following Route Table was created:

Route Table ID	rtb-013931ac13ebbb784
----------------	-----------------------

- Two route tables are shown in below image.

New VPC Experience [Learn more](#)

[Create route table](#) [Actions ▾](#)

Filter by tags and attributes or search by keyword

Name	Route Table ID	Explicit subnet association	Edge associations	Main	VPC ID
PrivateRT	rtb-013931ac13ebbb784	-	-	No	vpc-050c4db9f25644fee ...
PublicRT	rtb-03f4e5db9456938fe	-	-	No	vpc-050c4db9f25644fee ...
	rtb-0b6f9ab3921c87e88	-	-	Yes	vpc-050c4db9f25644fee ...
	rtb-926d17ec	-	-	Yes	vpc-a3e01fde ...

- Click on Edit routes to edit routes.

Name	Route Table ID	Explicit subnet association	Edge associations	Main	VPC ID
PrivateRT	rtb-013931ac13ebbb784	-	-	No	vpc-050c4db9f25644fee ...
PublicRT	rtb-03f4e5db9456938fe	-	-	No	vpc-050c4db9f25644fee ...
	rtb-0b6f9ab3921c87e88	-	-	Yes	vpc-050c4db9f25644fee ...
	rtb-926d17ec	-	-	Yes	vpc-a3e01fde ...

Route Table: rtb-03f4e5db9456938fe

Summary Routes Subnet Associations Edge Associations Route Propagation Tags

Edit routes

Destination	Target	Status	Propagated
10.0.0.0/16	local	active	No

- Add 0.0.0.0/0 destination and select IGW on Target field. Save routes and go back. It is only for PublicRT.

Destination	Target	Status	Propagated
10.0.0.0/16	local	active	No
0.0.0.0/0	igw-00452c93a981ff68		No

Add route

* Required

Cancel **Save routes**

- Routes saved.

Routes successfully edited

- Add subnets to route table. I am attaching three subnets to PublicRT.

- Now see three subnets are attached to PublicRT.

- Two subnets are attached to PrivateRT.

- Three subnets IP settings Modified to auto assign IP

Subnets (1/12) Info

Name	Subnet ID	Status	VPC	Actions
MySubnet0	subnet-01efc1adea6724bd	Available	vpc-050	View details Create flow log Modify auto-assign IP settings Edit IPv6 CIDRs Edit network ACL association Edit route table association Share subnet Manage tags Delete subnet
MySubnet5	subnet-0b1e84a410aedf7	Available	vpc-050	
MySubnet3	subnet-0bd81e84a410aedf7	Available	vpc-050	
MySubnet1	subnet-0326249ae5495b711	Available	vpc-050	View details Create flow log Modify auto-assign IP settings Edit IPv6 CIDRs Edit network ACL association Edit route table association Share subnet Manage tags Delete subnet
MySubnet2	subnet-0e0a0f85878ec462d	Available	vpc-050	
MySubnet4	subnet-08546e0d11c35eda5	Available	vpc-050	

Modify auto-assign IP settings [Info](#)

Enable the auto-assign IP address setting to automatically request a public IPv4 or IPv6 address for a new network interface in this subnet.

Settings

Subnet ID	subnet-0326249ae5495b711
Auto-assign IPv4	Info
<input checked="" type="checkbox"/> Enable auto-assign public IPv4 address	
Auto-assign customer-owned IPv4 address	Info
<input type="checkbox"/> Enable auto-assign customer-owned IPv4 address	Option disabled because no customer owned pools found.

Cancel **Save**

You have successfully modified auto-assign IP settings.

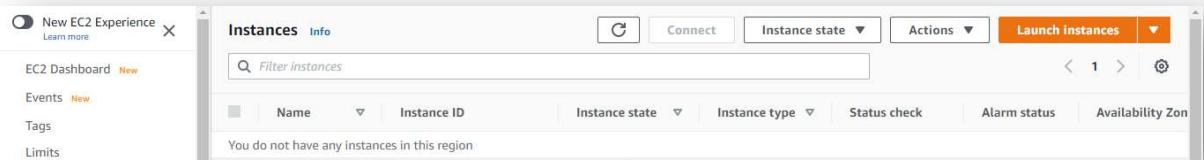
- Public IPv4 address

Subnets (1/1) Info

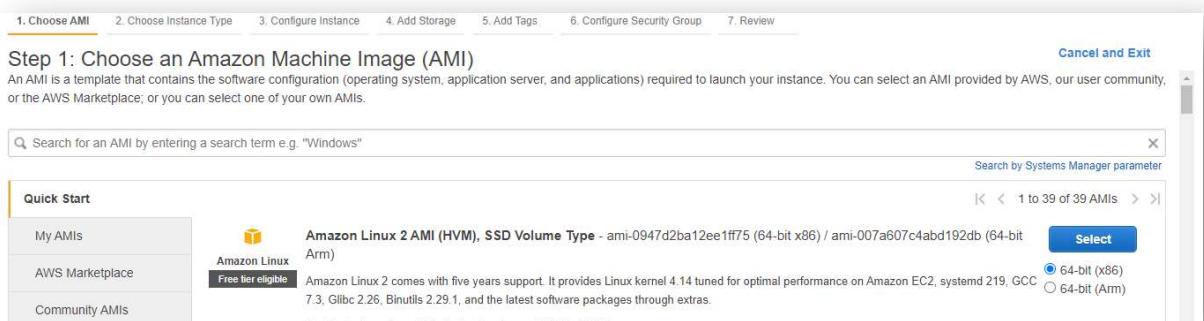
Name	Subnet ID	State	VPC	IPv4 CIDR
MySubnet1	subnet-0326249ae5495b711	Available	vpc-050	10.0.1.0/24

- Creating public instance.

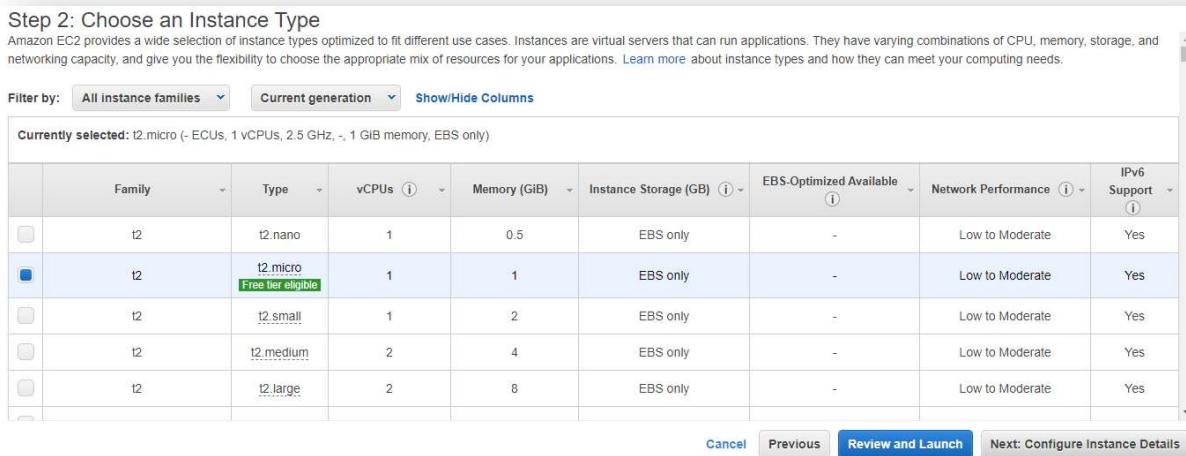
I. Launch Instance



II. Choose Amazon Linux AMI



III. Choose Instance Type



IV. Configure Instance Details. Select VPC i.e., MyVPC and select subnet 1 the Automatically Auto-assign Public IP Enabled.

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: 1

Purchasing option: Request Spot Instances

Network: vpc-050c4db9f25644fe | MyVPC

Subnet: subnet-0326249ae5495b711 | MySubnet1 | us-east-1

Auto-assign Public IP: Use subnet setting (Enable)

Placement group: Add instance to placement group

Capacity Reservation: Open

Domain join directory: No directory

IAM role: None

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

V. Keep everything as default and go to next.

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-0299d083f0ce6cd12	8	General Purpose SSD (gp2)	100 / 3000	N/A	<input checked="" 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.

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

VI. Add name tag Name :: Public Instance

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
Name	Public Instance	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Add another tag (Up to 50 tags maximum)

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

VII. Create new Security Group and add new HTTP protocol.

The screenshot shows the 'Step 6: Configure Security Group' page of the AWS instance creation wizard. The top navigation bar includes links for Choose AMI, Choose Instance Type, Configure Instance, Add Storage, Add Tags, Configure Security Group (which is highlighted), and Review. A warning message at the top states: '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.' Below this, there are two radio button options: 'Create a new security group' (selected) and 'Select an existing security group'. A 'Security group name' field contains 'launch-wizard-7' and a 'Description' field contains 'launch-wizard-7 created 2020-11-13T12:20:39.420+05:30'. A table lists two rules: one for SSH (TCP port 22) and another for HTTP (TCP port 80). An 'Add Rule' button is available. A warning box at the bottom left says: 'Warning: 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.' At the bottom right are 'Cancel', 'Previous', and 'Review and Launch' buttons.

VIII. Review the Instance and Launch the Instance.

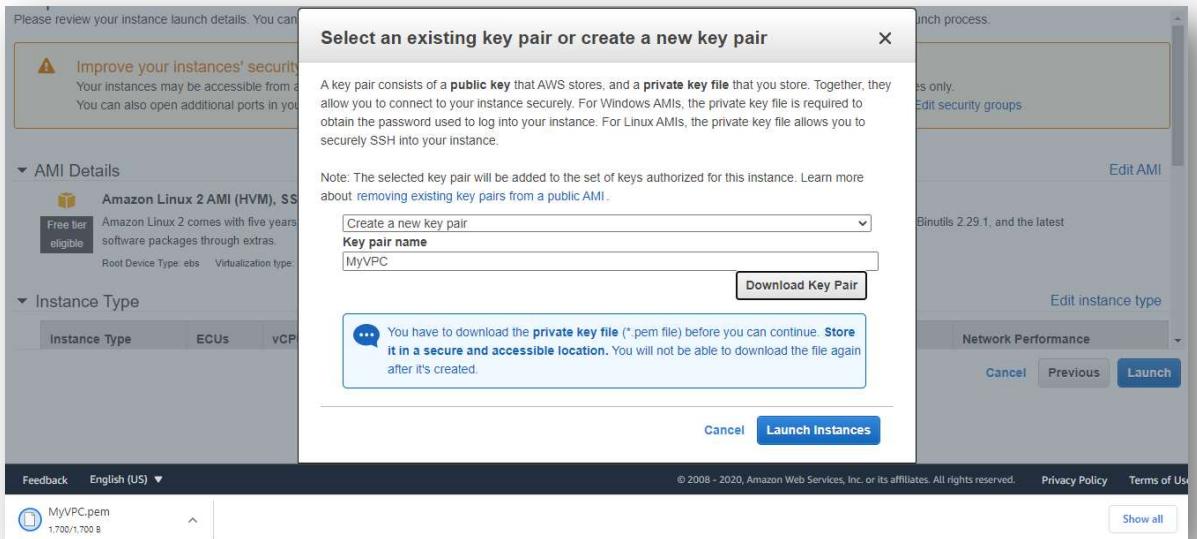
The screenshot shows the 'Step 7: Review Instance Launch' page of the AWS instance creation wizard. The top navigation bar includes links for Choose AMI, Choose Instance Type, Configure Instance, Add Storage, Add Tags, Configure Security Group, and Review (highlighted). A warning message at the top says: 'Please review your instance launch details. You can go back to edit changes for each section. Click Launch to assign a key pair to your instance and complete the launch process.' Below this, a yellow box contains a warning: '⚠ Improve your instances' security. Your security group, launch-wizard-7, is open to the world. Your instances may be accessible from any IP address. We recommend that you update your security group rules to allow access from known IP addresses only. You can also open additional ports in your security group to facilitate access to the application or service you're running, e.g., HTTP (80) for web servers. Edit security groups'.

Below the warning, there are two expandable sections: 'AMI Details' and 'Instance Type'. The 'AMI Details' section shows 'Amazon Linux 2 AMI (HVM), SSD Volume Type - ami-0947d2ba12ee1ff77' and indicates it is 'Free tier eligible'. The 'Instance Type' section shows a table for the 't2.micro' instance type:

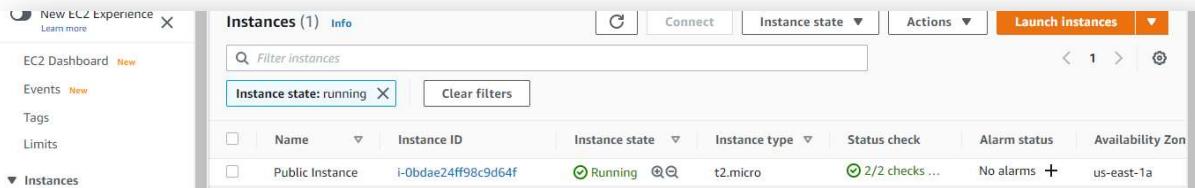
Instance Type	ECUs	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance
t2.micro	-	1	1	EBS only	-	Low to Moderate

At the bottom right are 'Cancel', 'Previous', and 'Launch' buttons.

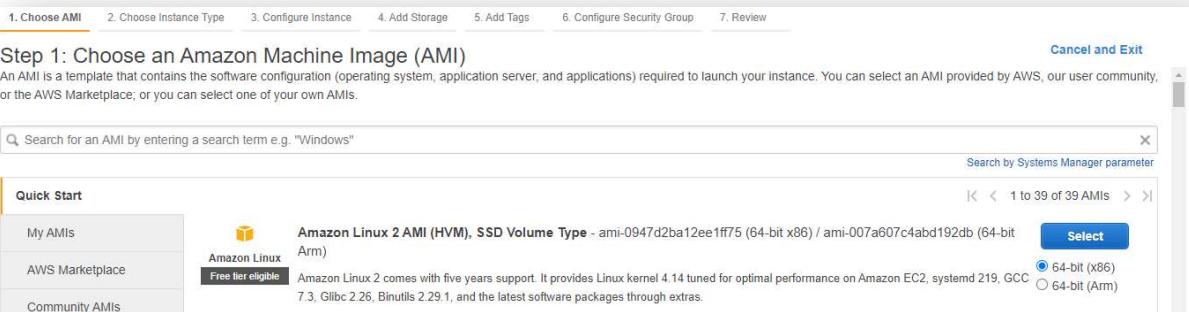
IX. Select Key pair either new or existing and download the key pair if it is new. Click Launch instances.



X. Public Instance Launched.



- Now creating new instance i.e., Private Instance
- I. Select Amazon Linux AMI.



II. Choose Instance Type t2.micro.

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	1	1	EBS only	-	Low to Moderate	Yes
<input type="checkbox"/>	t2	t2.small	1	2	EBS only	-	Low to Moderate	Yes
<input type="checkbox"/>	t2	t2.medium	2	4	EBS only	-	Low to Moderate	Yes

Cancel Previous Review and Launch Next: Configure Instance Details

III. Now select MyVPC in Network then select Subnet 4 or 5. After selecting subnet 4 or 5 automatically Auto-assign Public IP goes to Disable state.

Step 3: Configure Instance Details

Number of instances: 1 Launch into Auto Scaling Group ⓘ

Purchasing option: Request Spot instances

Network: vpc-050c4db9f25644fee | MyVPC Create new VPC

Subnet: subnet-08546e0d11c35eda5 | MySubnet4 | us-east-1 Create new subnet
251 IP Addresses available

Auto-assign Public IP: Use subnet setting (Disable)

Placement group: Add instance to placement group

Capacity Reservation: Open

Domain join directory: No directory Create new directory

IAM role: None Create new IAM role

CPU options: Specify CPU options

Cancel Previous Review and Launch Next: Add Storage

IV. Keep everything default and go to next step

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-0299d083f0ce6cd12	8	General Purpose SSD (gp2)	100 / 3000	N/A	<input checked="" 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**

V. Add name tag Name : Private Instance

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
Name		Private Instance		<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**

VI. Add http protocol and go to next

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: launch-wizard-8
Description: launch-wizard-8 created 2020-11-13T12:28:41.874+05:30

Type	Protocol	Port Range	Source	Description
SSH	TCP	22	Custom 0.0.0.0/0	e.g. SSH for Admin Desktop
HTTP	TCP	80	Custom 0.0.0.0/0	e.g. SSH for Admin Desktop

Add Rule

Warning
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.

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

VII. Review the Instance and click on Launch

Step 7: Review Instance Launch

Please review your instance launch details. You can go back to edit changes for each section. Click **Launch** to assign a key pair to your instance and complete the launch process.

AMI Details

Amazon Linux 2 AMI (HVM), SSD Volume Type - ami-0947d2ba12ee1ff75

Free tier eligible Amazon Linux 2 comes with five years support. It provides Linux kernel 4.14 tuned for optimal performance on Amazon EC2, systemd 219, GCC 7.3, Glibc 2.26, Binutils 2.29.1, and the latest software packages through extras.

Root Device Type: ebs Virtualization type: hvm

Instance Type

Instance Type	ECUs	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance
t2.micro	-	1	1	EBS only	-	Low to Moderate

Launch

VIII. Select previous key pair and Launch Instance

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.

Choose an existing key pair
Select a key pair
MyVPC

I acknowledge that I have access to the selected private key file (MyVPC.pem), and that without this file, I won't be able to log into my instance.

Launch Instances

IX. Instance Launched

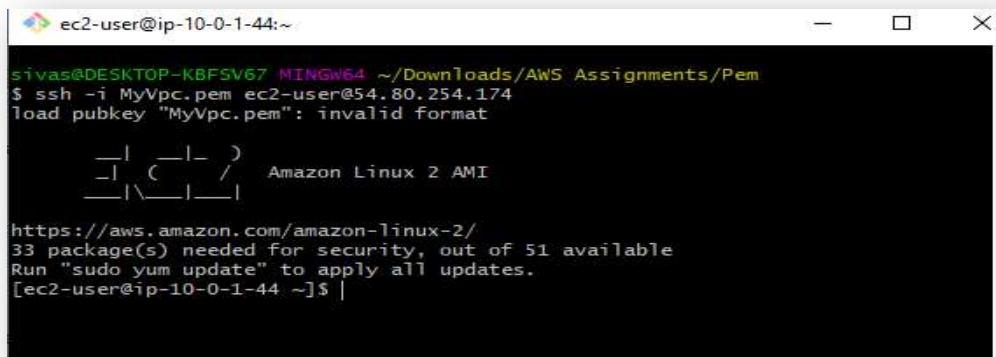
Instances (2) Info

Actions ▾ **Launch instances** ▾

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone
Public Instance	i-0bdae24ff98c9d64f	Running	t2.micro	2/2 checks ...	No alarms +	us-east-1a
Private Insta...	i-0958bc48163b5de01	Running	t2.micro	-	No alarms +	us-east-1d

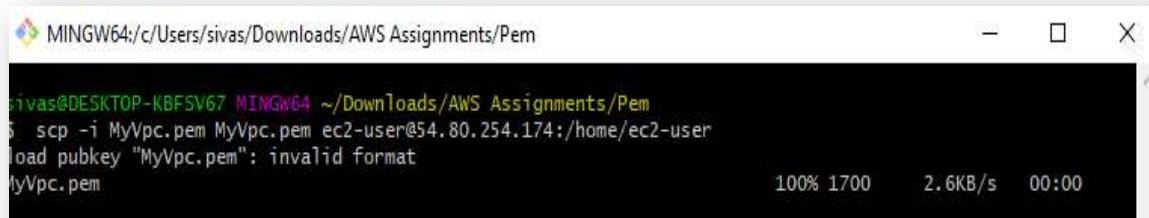
Connecting Private Instance Through Public Instance...

- First Login into Public instance through git



```
ec2-user@ip-10-0-1-44:~ sivas@DESKTOP-KBF5V67 MINGW64 ~/Downloads/AWS Assignments/Pem $ ssh -i MyVpc.pem ec2-user@54.80.254.174 load pubkey "MyVpc.pem": invalid format  
_ _ | _ _ / Amazon Linux 2 AMI  
_ _ \_\_ | _ _ https://aws.amazon.com/amazon-linux-2/  
33 package(s) needed for security, out of 51 available  
Run "sudo yum update" to apply all updates.  
[ec2-user@ip-10-0-1-44 ~]$ |
```

- Use another GitBash window to copy pem file to EC2 instance home directory.



```
MINGW64:/c/Users/sivas/Downloads/AWS Assignments/Pem sivas@DESKTOP-KBF5V67 MINGW64 ~/Downloads/AWS Assignments/Pem $ scp -i MyVpc.pem MyVpc.pem ec2-user@54.80.254.174:/home/ec2-user load pubkey "MyVpc.pem": invalid format  
MyVpc.pem 100% 1700 2.6KB/s 00:00
```

- Key pair Copied into EC2 instance.



```
ec2-user@ip-10-0-1-44:~ [ec2-user@ip-10-0-1-44 ~]$ ls MyVpc.pem [ec2-user@ip-10-0-1-44 ~]$
```

- Again login into Public EC2 instance, change pem file permissions. Now use same command to connect Private instance but use Private IP address of Private Instance instead of Public IP address of Public Instance.

```

ec2-user@ip-10-0-4-141:~$ ls
MyVpc.pem
[ec2-user@ip-10-0-1-44 ~]$ ssh -i MyVpc.pem ec2-user@10.0.4.141
The authenticity of host '10.0.4.141 (10.0.4.141)' can't be established.
ECDSA key fingerprint is SHA256:jkVcmIpz5cyHoMb3afZP8S8kfC5OxkMWlzbNrs7X4.
ECDSA key fingerprint is MD5:bc:d9:ff:6e:1a:9a:73:74:1c:ef:92:f8:13:a4:c0:11.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '10.0.4.141' (ECDSA) to the list of known hosts.
@@@@@@@WARNING: UNPROTECTED PRIVATE KEY FILE! @
Permissions 0644 for 'MyVpc.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 "MyVpc.pem": bad permissions
Permission denied (publickey,gssapi-keyex,gssapi-with-mic).
[ec2-user@ip-10-0-1-44 ~]$ chmod 400 MyVpc.pem
[ec2-user@ip-10-0-1-44 ~]$ ssh -i MyVpc.pem ec2-user@10.0.4.141
[ec2-user@ip-10-0-1-44 ~]$

```

The terminal window shows the user navigating through their home directory (~). They run an 'ls' command to list files, showing 'MyVpc.pem'. They then attempt to connect to an EC2 instance at '10.0.4.141' using SSH with the private key 'MyVpc.pem'. The connection fails due to a warning about the file being 'UNPROTECTED PRIVATE KEY FILE!' and having 'Permissions 0644'. The user then runs 'chmod 400 MyVpc.pem' to change the permissions to '0400', which makes the connection succeed. Finally, they run another 'ssh' command to connect to the instance again, this time successfully.