



Andhra Pradesh State Skill Development Corporation



AWS CLOUD COMPUTING

CONFIGURATION OF THE PUBLIC SUBNET



Configuration of the Public Subnet



Public Subnet

Instances in Public Subnet can be accessible From the Internet, which means traffic From the Internet can hit a machine/Instance/Service in Public Subnet. You normally keep things like Load Balancers, Web Servers in Public Subnet. So when you create them, you add name Public in front of them to keep them separate from others and, it doesn't matter you enabled Auto-assign Public IPv4, but every time you choose the Public Subnet that you marked Public, you have to check or enable Auto-assign Public IPv4 option when you launch an instance of EC2 or RDS/Service. But it's better if you enable Auto-assign Public IPv4 and Subnet level to make it properly Public because this is the reason you are making difference and using Public-Private concept, so whenever you launch any service/Instance/machine in Public Subnet, it will be able to accessible From the Internet and To the Internet, means, You can hit the service/Instance over the Internet and can download updates and packages in the service/Instance as well.

Private Subnet

Instances in Private Subnet cannot be accessible From the Internet. E.g. you can put Database Server, Redis Server or these kinds of other services in a Private Subnet and no one can access it From the Internet. It would be accessible only via Instances/machines/Services in Public Subnet (Web server, ELB, etc). Because it doesn't have a Public IP enabled option and also, we marked it as Private for the specific use as explained, for security and unwanted access over the Internet. This is good for architecture level security to avoid loopholes. To access this Private Subnet services/Instances, you have to add allow rule in Security group and add proper routes in the route table.

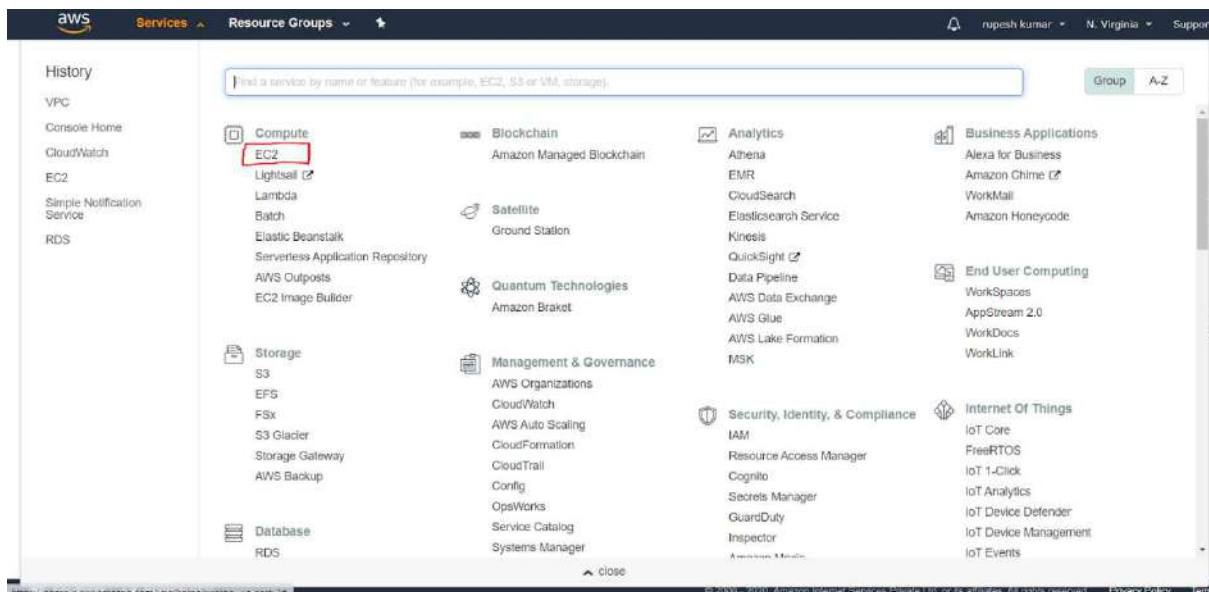
To Connect Linux Instance in Public Subnet

Launch Linux instance in public subnet → Example_pub_subnet

Open the AWS console

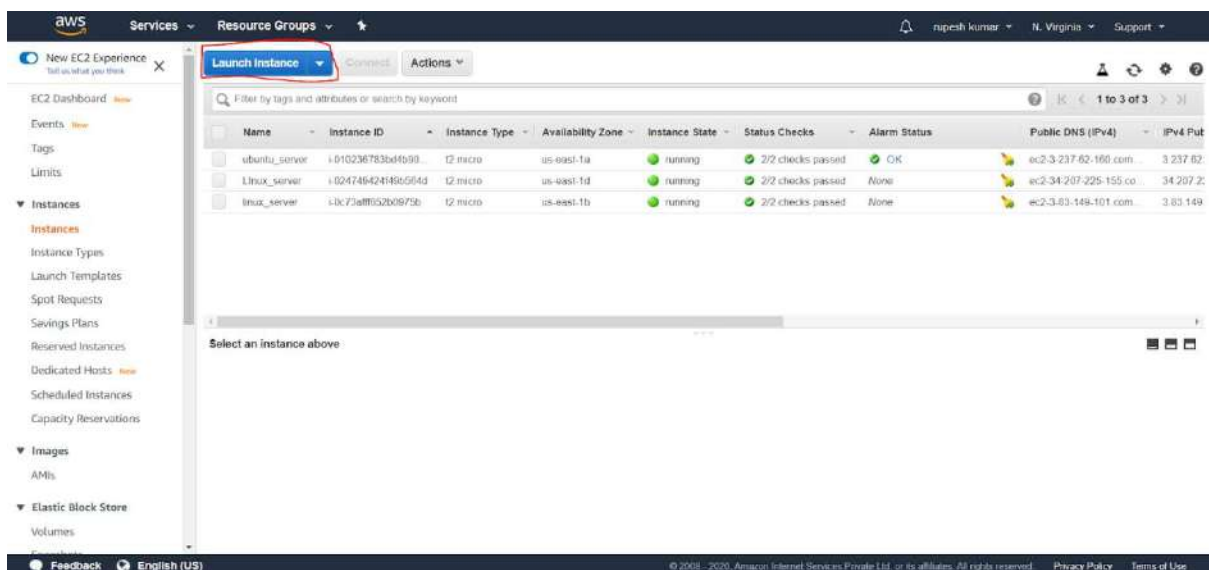
Click on Services

Select EC2



Click on Instances

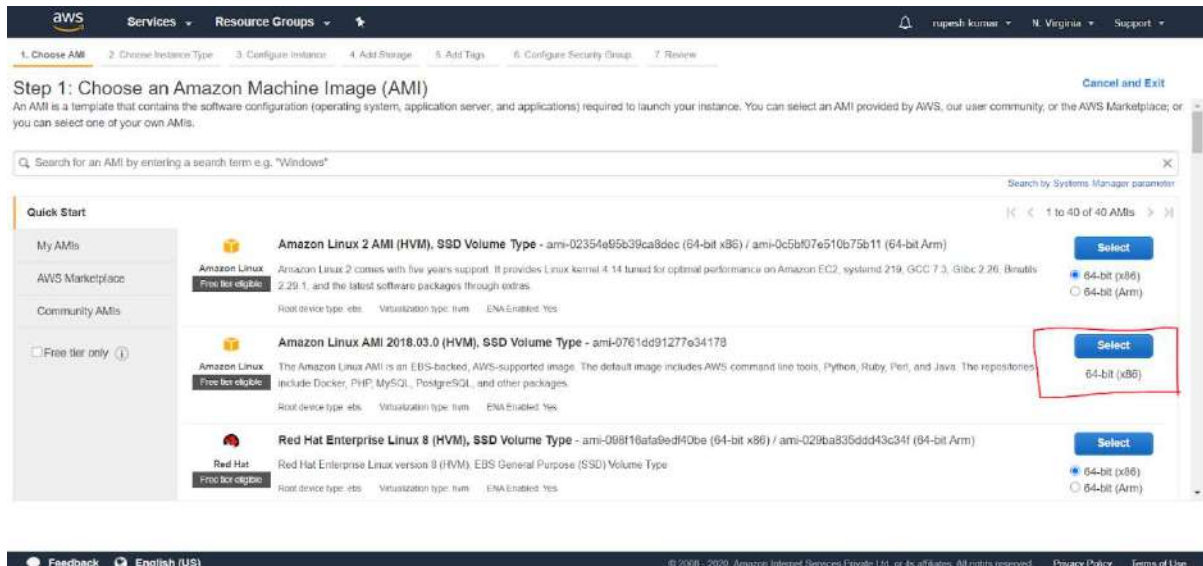
Click on “Launch Instance” button



On the “Choose an Amazon Machine Image (AMI)” page

Select AMI “**Amazon Linux AMI 2018.03.0 (HVM), SSD Volume Type** - ami-0761dd91277e34178”

Click on select button



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.

Search for an AMI by entering a search term e.g. "Windows"

Quick Start

- My AMIs
- AWS Marketplace
- Community AMIs

Free tier only (1)

Amazon Linux 2 AMI (HVM), SSD Volume Type - ami-02354e95b39ca8dec (64-bit x86) / ami-0c5b07e510b75b11 (64-bit Arm)

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 EBS-Enabled: Yes

Amazon Linux AMI 2018.03.0 (HVM), SSD Volume Type - ami-0761dd91277e34178

The Amazon Linux AMI is an EBS-backed, AWS-supported image. The default image includes AWS command line tools, Python, Ruby, Perl, and Java. The repositories include Docker, PHP, MySQL, PostgreSQL, and other packages.

Root device type: ebs Virtualization type: hvm EBS-Enabled: Yes

Red Hat Enterprise Linux 8 (HVM), SSD Volume Type - ami-098f16afa9edf40be (64-bit x86) / ami-029ba835dd43c34f (64-bit Arm)

Red Hat Enterprise Linux version 8 (HVM), EBS General Purpose (SSD) Volume Type

Root device type: ebs Virtualization type: hvm EBS-Enabled: Yes

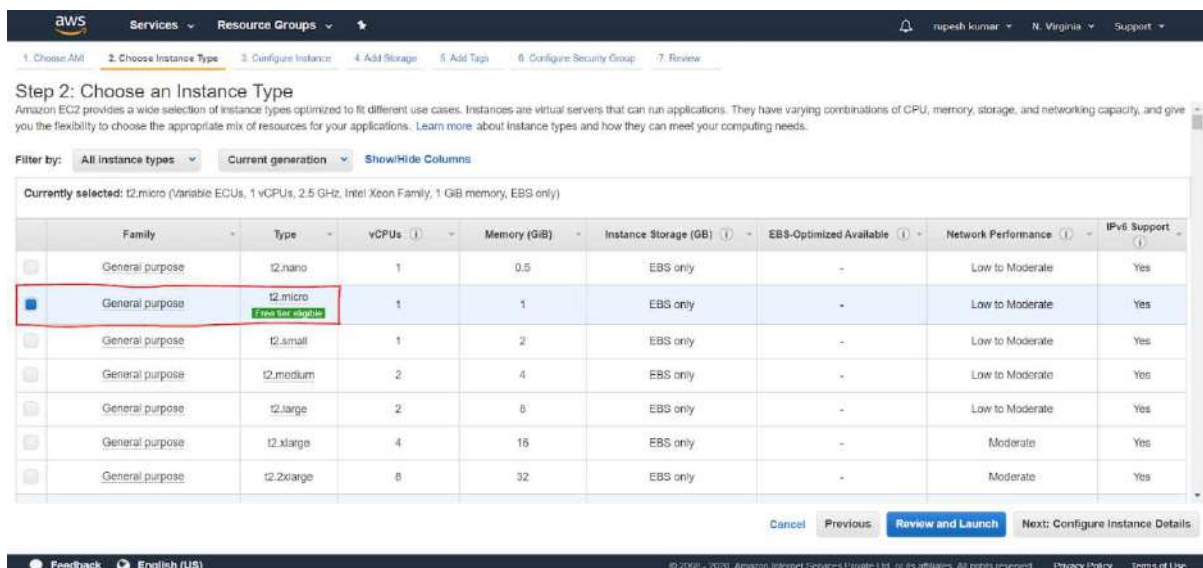
Feedback English (US) © 2009 - 2020 Amazon Internet Services Private Ltd. or its affiliates. All rights reserved. Privacy Policy Terms of Use

On the “Choose an instance Type” page

Select “General purpose”

Type → t2.micro

Click on “Next: Configure Instance Details”



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 types Current generation Show/Hide Columns

Currently selected: t2.micro (Variable ECUs, 1 vCPUs, 2.5 GHz, Intel Xeon Family, 1 GB memory, EBS only)

	Family	Type	vCPUs (1)	Memory (GiB)	Instance Storage (GB) (1)	EBS-Optimized Available (1)	Network Performance (1)	IPv6 Support (1)
<input type="checkbox"/>	General purpose	t2.nano	1	0.5	EBS only	-	Low to Moderate	Yes
<input checked="" type="checkbox"/>	General purpose	t2.micro	1	1	EBS only	-	Low to Moderate	Yes
<input type="checkbox"/>	General purpose	t2.small	1	2	EBS only	-	Low to Moderate	Yes
<input type="checkbox"/>	General purpose	t2.medium	2	4	EBS only	-	Low to Moderate	Yes
<input type="checkbox"/>	General purpose	t2.large	2	8	EBS only	-	Low to Moderate	Yes
<input type="checkbox"/>	General purpose	t2.xlarge	4	16	EBS only	-	Moderate	Yes
<input type="checkbox"/>	General purpose	t2.2xlarge	8	32	EBS only	-	Moderate	Yes

Cancel Previous Review and Launch Next: Configure Instance Details

Feedback English (US) © 2009 - 2020 Amazon Internet Services Private Ltd. or its affiliates. All rights reserved. Privacy Policy Terms of Use

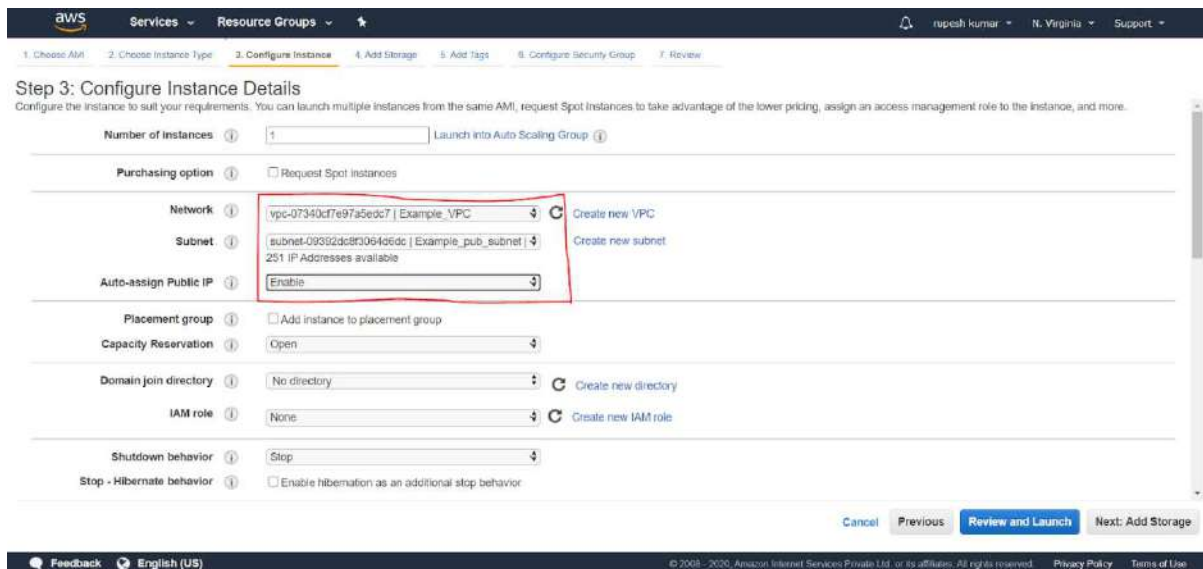
On the “Configuration Instance Details” page

Number of instances → 1

Network → Example_VPC

Subnet → Example_pub_subnet

Auto-assign Public IP → Enable

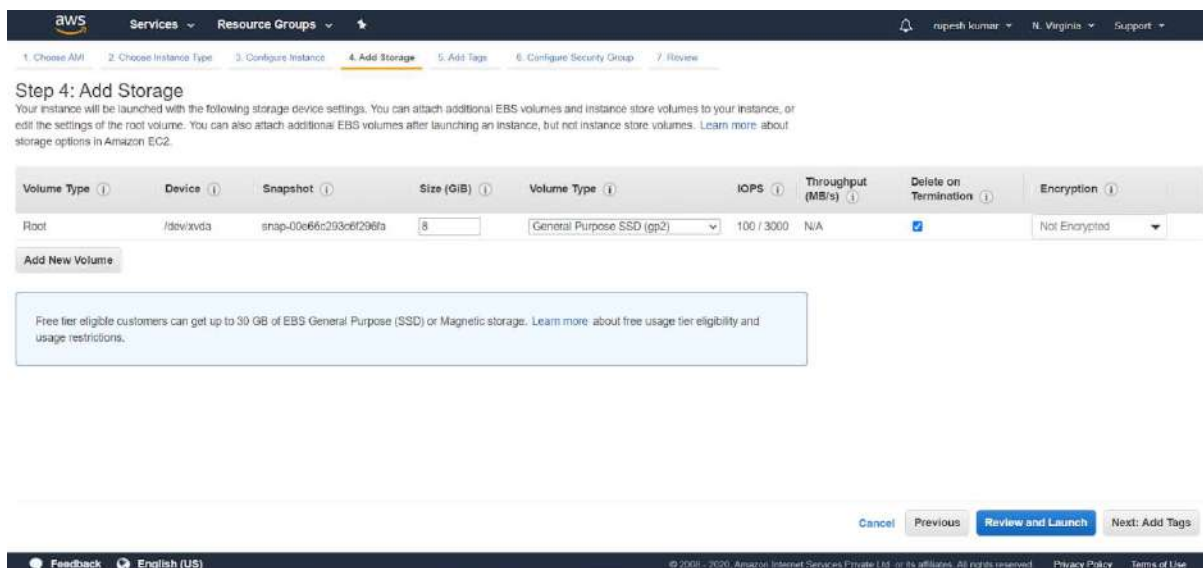


The screenshot shows the AWS Management Console 'Step 3: Configure Instance Details' page. The 'Number of instances' is set to 1. The 'Purchasing option' is 'On-Demand'. The 'Network' is set to 'vpc-07340cd7e97a5edc7 | Example_VPC'. The 'Subnet' is set to 'subnet-09392dc8f306426dc | Example_pub_subnet'. The 'Auto-assign Public IP' is set to 'Enable'. The 'Placement group' is 'Open'. The 'Domain join directory' is 'No directory'. The 'IAM role' is 'None'. The 'Shutdown behavior' is 'Stop'. The 'Stop - Hibernate behavior' is 'None'. The 'Review and Launch' button is highlighted.

On the “Add storage” page

Leave the values as default

Click on “Next: Add Tags” button



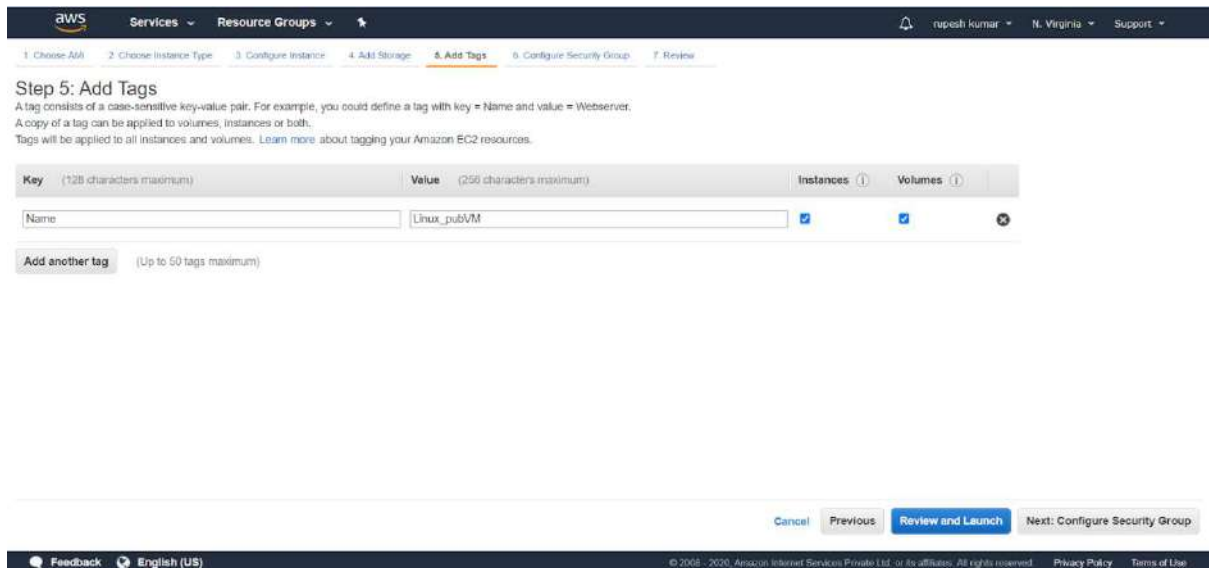
The screenshot shows the AWS Management Console 'Step 4: Add Storage' page. The 'Volume Type' is 'General Purpose SSD (gp2)'. The 'Device' is '/dev/xvda'. The 'Snapshot' is 'snap-00e66c293cbf296fa'. The 'Size (GiB)' is 8. The 'IOPS' is 100 / 3000. The 'Throughput (MB/s)' is N/A. The 'Delete on Termination' is checked. The 'Encryption' is 'Not Encrypted'. The 'Next: Add Tags' button is highlighted.

On the “Add Tags” page

Key → Name

Value → Linux_pubVM

Click on “Next Configuration Security Group” button



The screenshot shows the AWS Management Console 'Step 5: Add Tags' page. It includes a progress bar at the top with steps 1 through 7. The main heading is 'Step 5: Add Tags'. Below it, a text box explains that a tag is a case-sensitive key-value pair and can be applied to instances and volumes. A table for adding tags is visible with columns for Key, Value, Instances, and Volumes. One tag is added with Key 'Name' and Value 'Linux_pubVM'. At the bottom, there are buttons for 'Cancel', 'Previous', 'Review and Launch', and 'Next: Configure Security Group'.

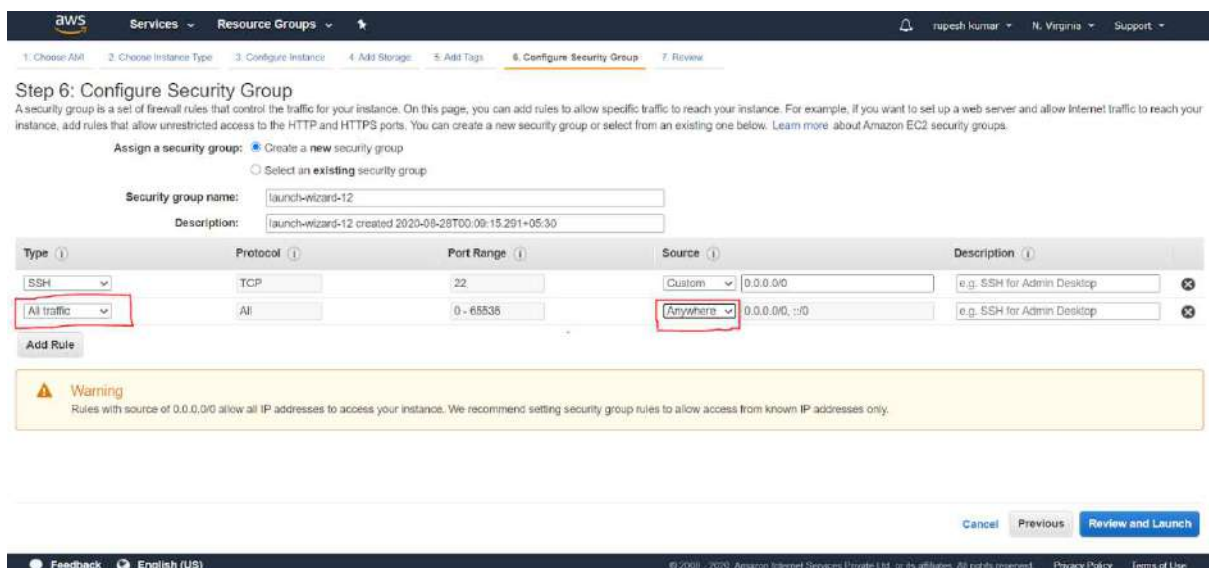
On the “Configuration Security Group” page

Assign a security group → Create a new security group

Click on Add rule and select All traffic

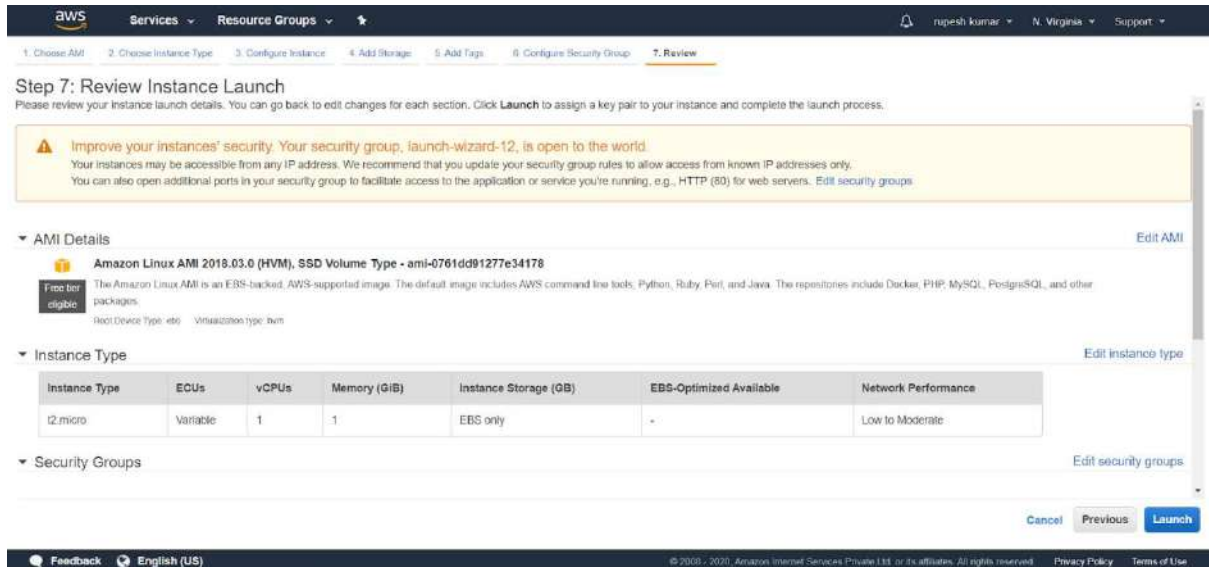
For All traffic rule types select the source to Anywhere

Click on Review and Launch button

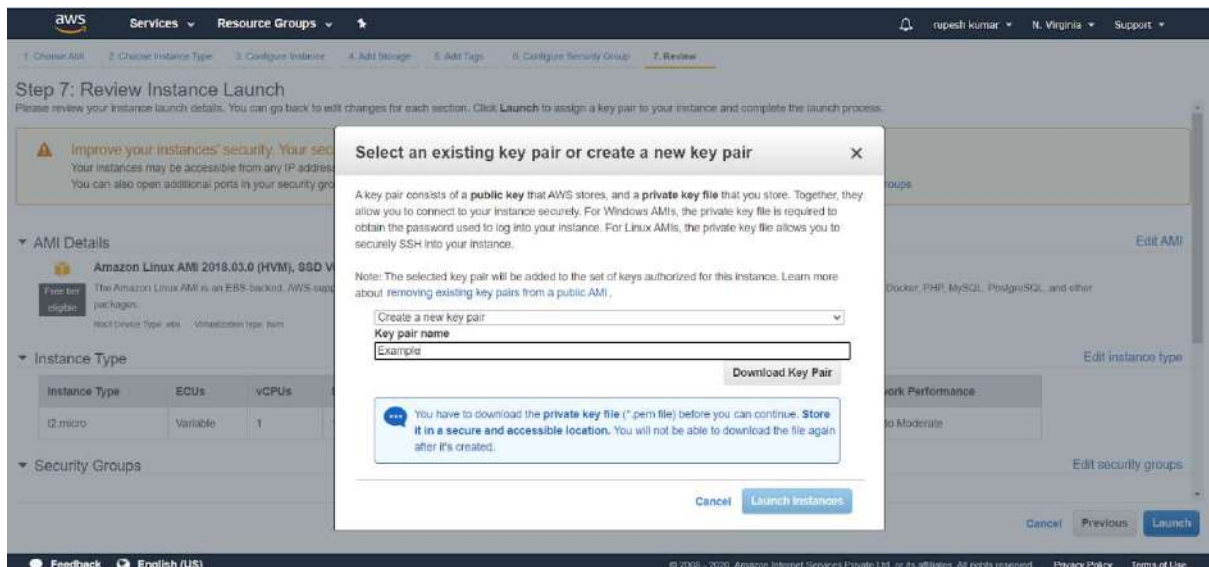


The screenshot shows the AWS Management Console 'Step 6: Configure Security Group' page. It includes a progress bar at the top with steps 1 through 7. The main heading is 'Step 6: Configure Security Group'. Below it, a text box explains that a security group is a set of firewall rules that control traffic to and from your instance. The 'Assign a security group' section shows 'Create a new security group' selected. The 'Security group name' is 'launch-wizard-12' and the 'Description' is 'launch-wizard-12 created 2020-08-28T00:09:15.291+05:30'. A table for adding rules is visible with columns for Type, Protocol, Port Range, Source, and Description. Two rules are added: one for SSH (Type: SSH, Protocol: TCP, Port Range: 22, Source: Custom 0.0.0.0/0) and one for All traffic (Type: All traffic, Protocol: All, Port Range: 0 - 65535, Source: Anywhere 0.0.0.0/0). At the bottom, there is a warning message about allowing access from known IP addresses only. At the very bottom, there are buttons for 'Cancel', 'Previous', 'Review and Launch', and 'Next: Configure Security Group'.

On the “Review Instance Launch” page
Click on Launch button



On the “Select an existing key pair or create a new key pair” page
Select Create a new key pair
Key pair name → Example
Click on the Download Key Pair
Click on “Launch Instance” button





Click the summary

Click on View Instance button

Launch Status

Get notified of estimated charges
Create billing alerts to get an email notification when estimated charges on your AWS bill exceed an amount you define (for example, if you exceed the free usage tier).

How to connect to your instances
Your instances are launching, and it may take a few minutes until they are in the **running** state, when they will be ready for you to use. Usage hours on your new instances will start immediately and continue to accrue until you stop or terminate your instances.
Click **View Instances** to monitor your instances' status. Once your instances are in the **running** state, you can **connect** to them from the instances screen. Find out [how to connect to your instances](#).

Here are some helpful resources to get you started

- How to connect to your Linux instance
- Learn about AWS Free Usage Tier
- Amazon EC2: User Guide
- Amazon EC2: Discussion Forum

While your instances are launching you can also

- Create status check alarms to be notified when these instances fail status checks. (Additional charges may apply)
- Create and attach additional EBS volumes. (Additional charges may apply)
- Manage security groups

[View Instances](#)

Feedback English (US) © 2009 - 2020, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved. Privacy Policy Terms of Use

Example.pem Show all X

Verification

Linux instance in public subnet is launched

AWS Services Resource Groups

New EC2 Experience

Launch Instance Connect Actions

Filter by tags and attributes or search by keyword

Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS (IPv4)
ubuntu_server	i-010236783ed4b90...	t2.micro	us-east-1a	running	2/2 checks passed	None	ec2-3-236-156-11
linux_server	i-0247494241f0b594d	t2.micro	us-east-1d	running	2/2 checks passed	None	ec2-34-207-225-
Example_pubVM	i-0b1015a396f601756	t2.micro	us-east-1d	initializing	initializing	None	
linux_server	i-0c73aff052b0675b	t2.micro	us-east-1b	running	2/2 checks passed	None	ec2-3-83-149-10

Instance: **i-0b1015a396f601756 (Example_pubVM)** Public IP: 54.160.204.45

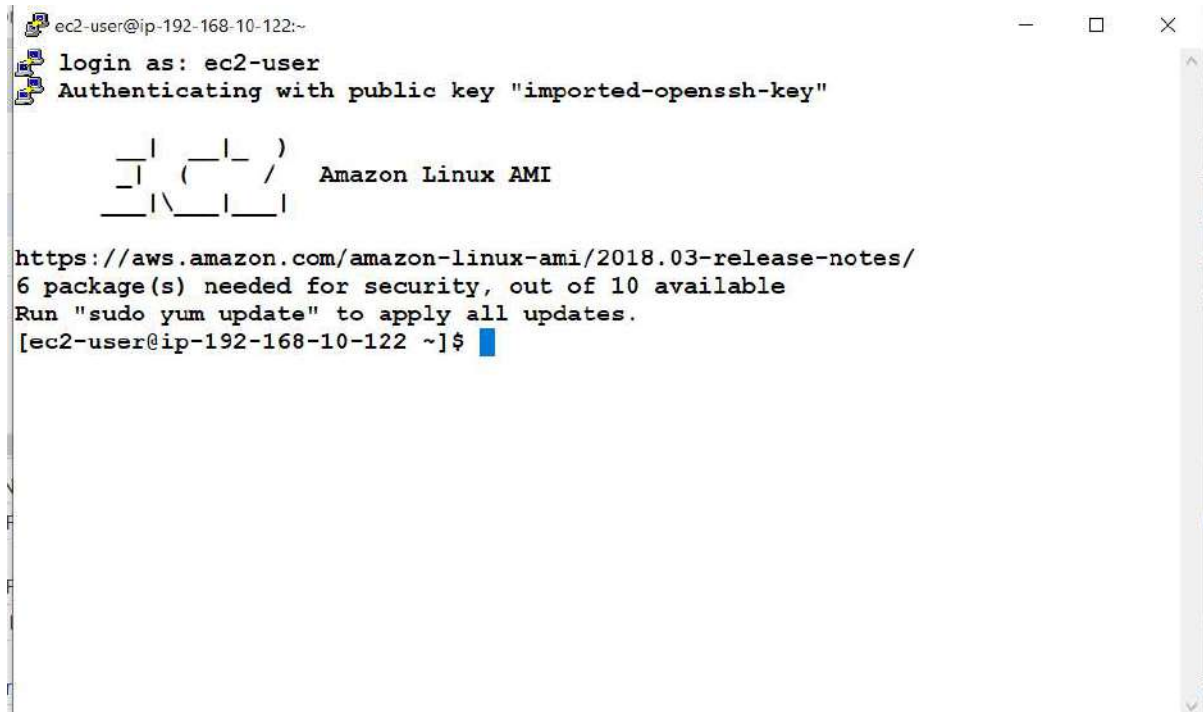
Description Status Checks Monitoring Tags

Instance ID: i-0b1015a396f601756
Instance state: running
Instance type: t2.micro
Finding: Opt-in to AWS Compute Optimizer for recommendations. [Learn more](#)
Private DNS: ip-102-166-10-122.ec2.internal
Private IP: 192.168.10.122
Public DNS (IPv4): 54.160.204.45
IPv4 Public IP: 54.160.204.45
IPv6 IPs: -
Elastic IPs: -
Availability zone: us-east-1d
Security groups: launch wizard, view inbound rules, view outbound rules

Feedback English (US) © 2009 - 2020, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved. Privacy Policy Terms of Use

Example.pem Show all X

Now connect instance in public subnet using ssh



```
ec2-user@ip-192-168-10-122:~  
login as: ec2-user  
Authenticating with public key "imported-openssh-key"  
  
  _ | _ | _ )  
  _ | ( _ | /  Amazon Linux AMI  
  _ | \ _ | _ |  
  
https://aws.amazon.com/amazon-linux-ami/2018.03-release-notes/  
6 package(s) needed for security, out of 10 available  
Run "sudo yum update" to apply all updates.  
[ec2-user@ip-192-168-10-122 ~]$
```

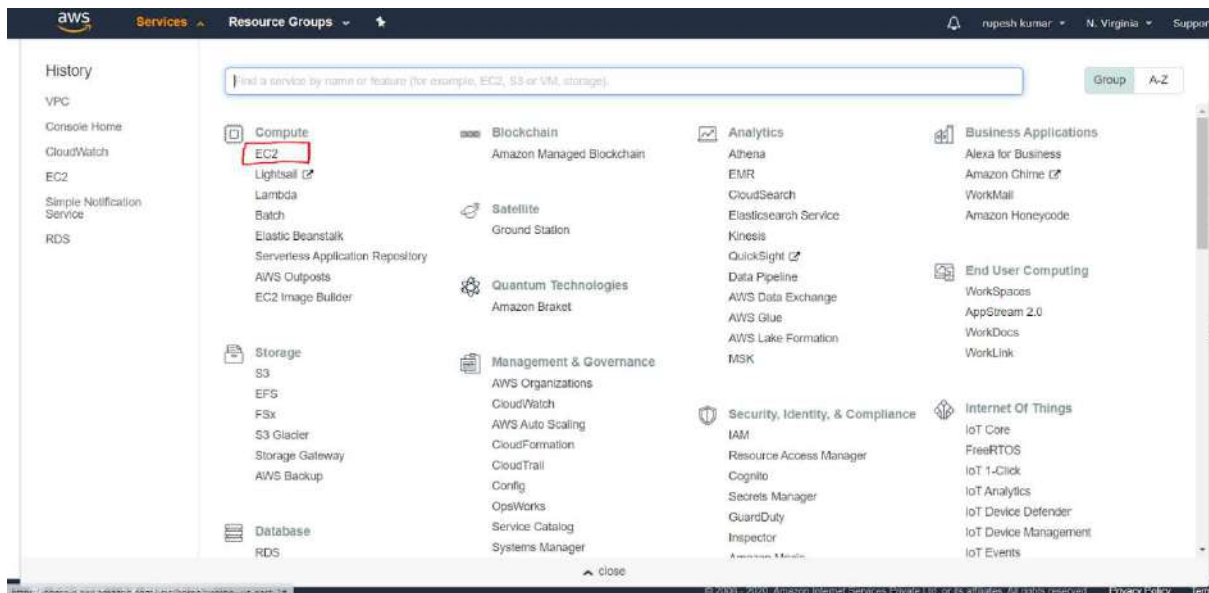
To Connect Linux Instance in Private Subnet

Launch Linux instance in private subnet → Example_pvt_sub

Open the AWS console

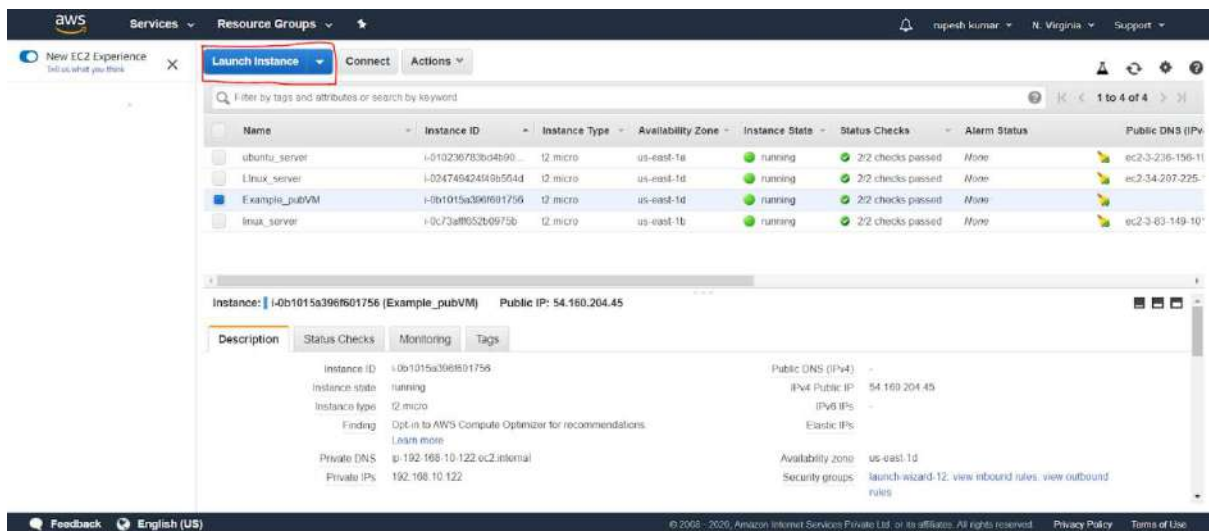
Click on Services

Click on EC2



Click on Instances

Click on “Launch Instance” button



On the “Choose an Amazon Machine Image (AMI)” page

Select AMI “Amazon Linux AMI 2018.03.0 (HVM), SSD Volume Type - ami-0761dd91277e34178”

Click on Select button

Step 1: Choose an Amazon Machine Image (AMI)

Quick Start

- My AMIs
- AWS Marketplace
- Community AMIs

☐ Free tier only

AMI	Architecture	Root device type	Virtualization type	ENI Enabled	Select
Amazon Linux 2 AMI (HVM), SSD Volume Type - ami-02354e95b39ca8dec (64-bit x86) / ami-0c5b07e510b75b11 (64-bit Arm)	64-bit x86	efs	hvm	Yes	Select
Amazon Linux 2018.03.0 (HVM), SSD Volume Type - ami-0761dd61277e34178	64-bit x86	efs	hvm	Yes	Select
Red Hat Enterprise Linux 8 (HVM), SSD Volume Type - ami-09616af69edf40be1 (64-bit x86) / ami-029ba835dd43c34f (64-bit Arm)	64-bit x86	efs	hvm	Yes	Select
SUSE Linux Enterprise Server 15 SP2 (HVM), SSD Volume Type - ami-0a782e324055d1cc0 (64-bit x86) / ami-06e0eaf39ca724d4 (64-bit Arm)	64-bit x86	efs	hvm	Yes	Select

On the “Choose an instance Type” page

Select “General purpose”

Type → t2.micro

Click on “Next: Configure Instance Details” button

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 types | Current generation | Show/Hide Columns

Currently selected: t2.micro (Variable ECUs, 1 vCPUs, 2.5 GHz, Intel Xeon Family, 1 GiB memory, EBS only)

Family	Type	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance	IPv6 Support
General purpose	t2.nano	1	0.5	EBS only	-	Low to Moderate	Yes
General purpose	t2.micro	1	1	EBS only	-	Low to Moderate	Yes
General purpose	t2.small	1	2	EBS only	-	Low to Moderate	Yes
General purpose	t2.medium	2	4	EBS only	-	Low to Moderate	Yes
General purpose	t2.large	2	8	EBS only	-	Low to Moderate	Yes
General purpose	t2.xlarge	4	16	EBS only	-	Moderate	Yes

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

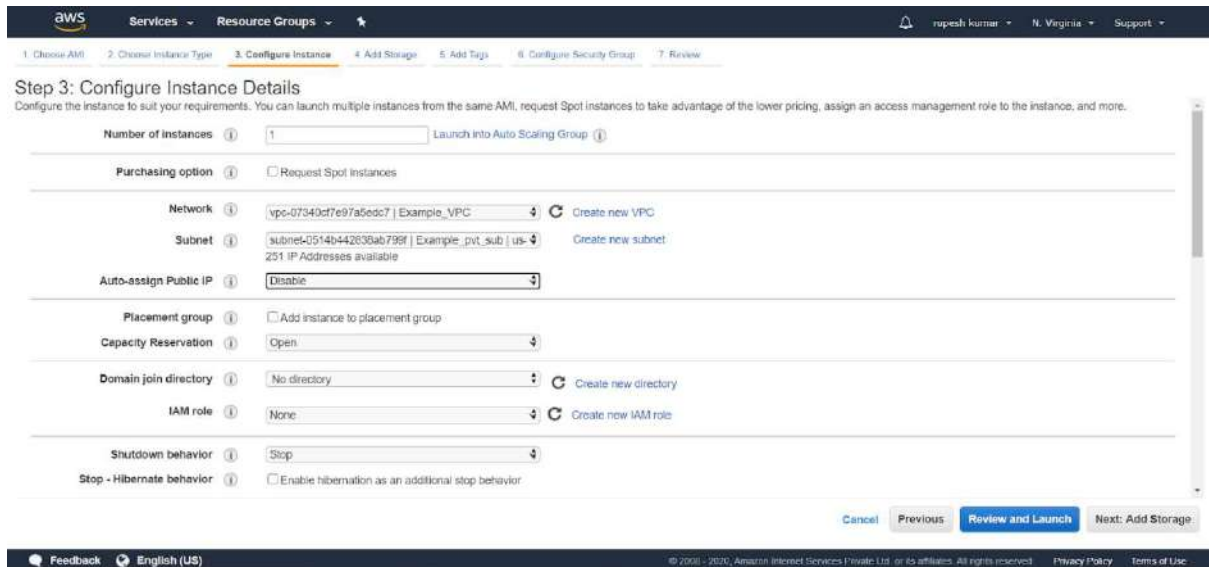
On the “Configuration Instance Details” page

Number of instances → 1

Network → Example_VPC

Subnet → Example_pvt_sub

Auto-assign Public IP → Disable



The screenshot shows the AWS Management Console interface for Step 3: Configure Instance Details. The navigation bar at the top includes the AWS logo, Services, Resource Groups, and user information (rupesh kumar, N. Virginia, Support). The progress bar indicates the current step is 3 of 7. The main content area is titled "Step 3: Configure Instance Details" and includes a sub-header: "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." The configuration options are as follows:

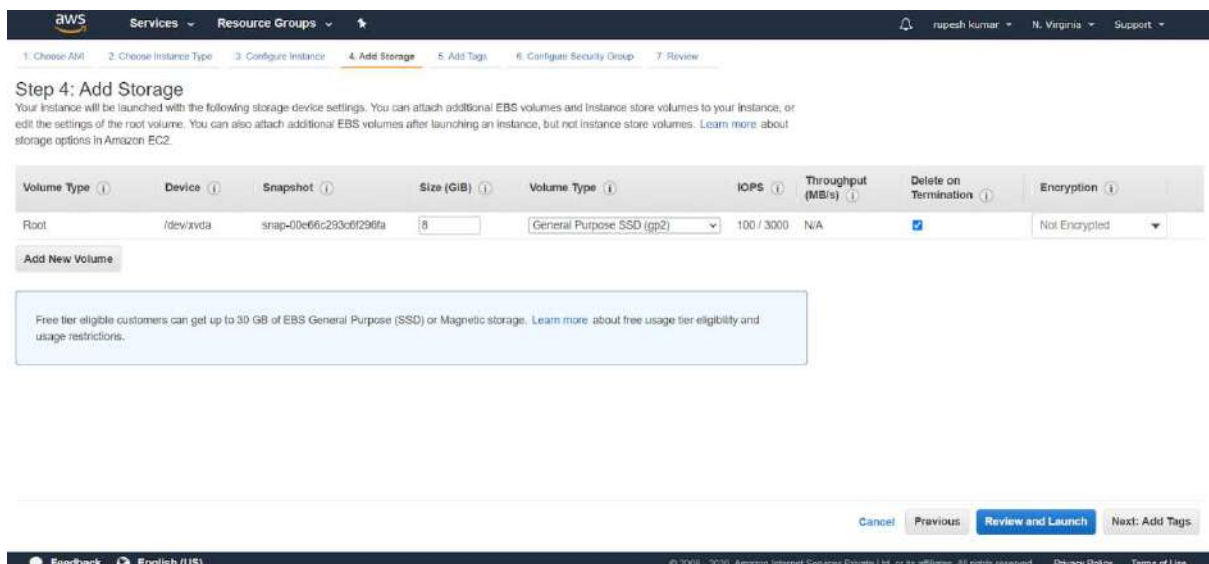
- Number of instances: 1 (with a link to "Launch into Auto Scaling Group")
- Purchasing option: ☐ Request Spot Instances
- Network: vpc-07340cd7e97a5edc7 | Example_VPC (with a "Create new VPC" link)
- Subnet: subnet-0514b442630ab799f | Example_pvt_sub | us- (with a "Create new subnet" link and "251 IP Addresses available")
- Auto-assign Public IP: Disable
- Placement group: ☐ Add instance to placement group
- Capacity Reservation: Open
- Domain join directory: No directory (with a "Create new directory" link)
- IAM role: None (with a "Create new IAM role" link)
- Shutdown behavior: Stop
- Stop - Hibernate behavior: ☐ Enable hibernation as an additional stop behavior

At the bottom, there are buttons for "Cancel", "Previous", "Review and Launch", and "Next: Add Storage". The footer includes "Feedback", "English (US)", and copyright information.

On the “Add Storage” page

Leave the value as default

Click on “Next: Add Tags” button

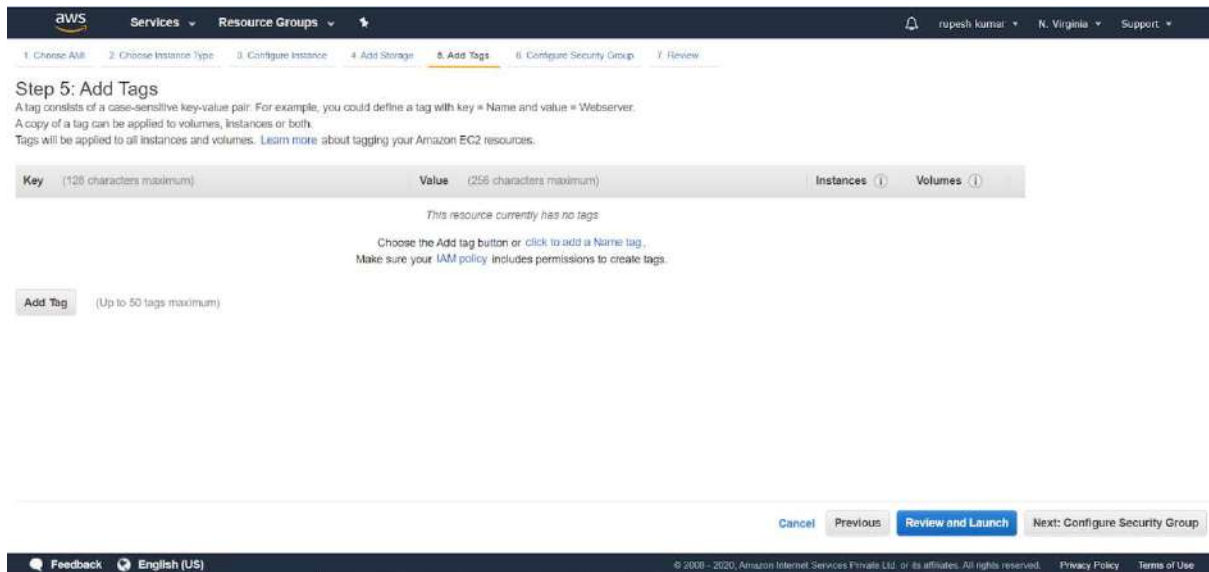


The screenshot shows the AWS Management Console interface for Step 4: Add Storage. The navigation bar and progress bar are consistent with the previous page. The main content area is titled "Step 4: Add Storage" and includes a sub-header: "Your instance will be launched with the following storage device 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." Below this, there is a table showing the root volume configuration:

Volume Type	Device	Snapshot	Size (GiB)	Volume Type	IOPS	Throughput (MiB/s)	Delete on Termination	Encryption
Root	/dev/xvda	snap-00e66c293cd0f296fa	8	General Purpose SSD (gp2)	100 / 3000	N/A	<input checked="" type="checkbox"/>	Not Encrypted

Below the table is an "Add New Volume" button. A note states: "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." At the bottom, there are buttons for "Cancel", "Previous", "Review and Launch", and "Next: Add Tags". The footer is also consistent with the previous page.

Click on Add Tag



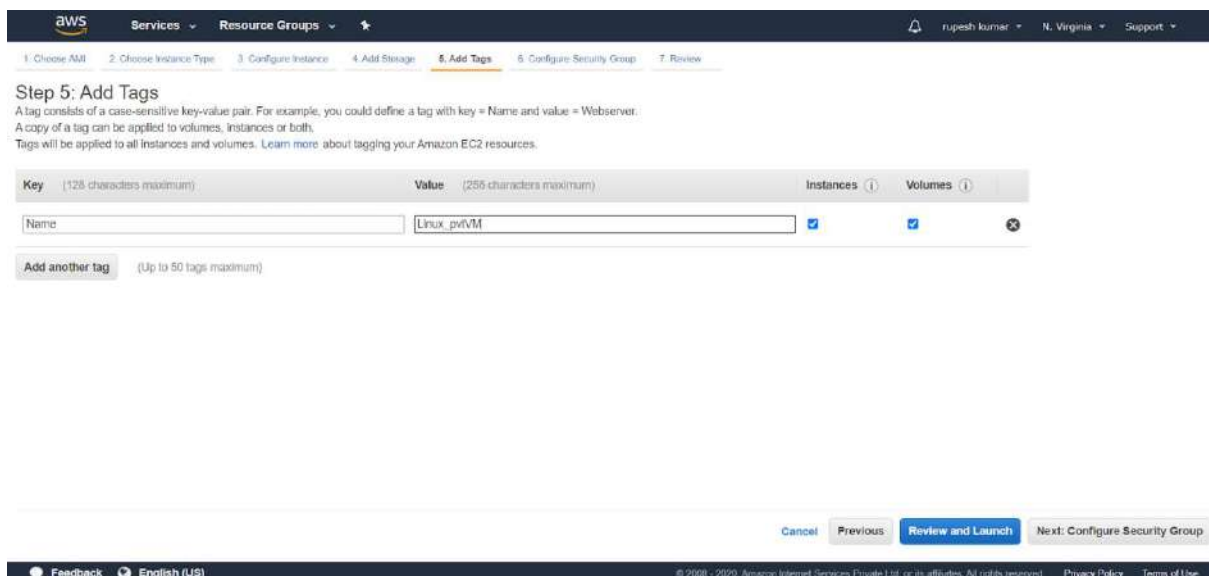
The screenshot shows the AWS console interface for the 'Add Tags' step. The top navigation bar includes the AWS logo, 'Services', 'Resource Groups', and user information. The breadcrumb trail shows the sequence: 1. Choose AMI, 2. Choose Instance Type, 3. Configure Instance, 4. Add Storage, 5. Add Tags (highlighted), 6. Configure Security Group, 7. Review. The main heading is 'Step 5: Add Tags'. Below it, a text block explains that a tag is a case-sensitive key-value pair and can be applied to instances and volumes. A table with two columns, 'Key' and 'Value', is shown, both with a '(128 characters maximum)' limit. Below the table, a message states 'This resource currently has no tags'. A button labeled 'Add Tag' is present, with a note '(Up to 50 tags maximum)'. At the bottom, there are buttons for 'Cancel', 'Previous', 'Review and Launch', and 'Next: Configure Security Group'. The footer includes 'Feedback', 'English (US)', and copyright information.

On the “Add Tags” page

Key → Name

Value → Example_pvtVM

Click on “Next: Configure Security Group” button



This screenshot shows the same AWS 'Add Tags' step, but with a tag added. The 'Key' field contains 'Name' and the 'Value' field contains 'Linux_pvtVM'. Checkmarks are visible in the 'Instances' and 'Volumes' columns, indicating the tag is applied to both. The 'Add another tag' button is still present. The bottom navigation bar and footer are identical to the previous screenshot.

On the “Configuration Security Group” page
 Assign a security group → Create a new security group
 Click on Add Rule and select all traffic
 For all traffic select the source to Anywhere
 Click on “Review and Launch” button

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:
 Description:

Type	Protocol	Port Range	Source	Description
SSH	TCP	22	Custom 0.0.0.0/0	e.g. SSH for Admin Desktop
All traffic	All	0 - 65535	Anywhere 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](#)

On the “Review Instance Launch” page
 Click on Launch button

Step 7: Review Instance Launch

▼ **AMI Details** [Edit AMI](#)
Amazon Linux AMI 2018.03.0 (HVM), SSD Volume Type - ami-0761dd91277e34178
 The Amazon Linux AMI is an EBS-backed, AWS-supported image. The default image includes AWS command line tools, Python, Ruby, Perl, and Java. The repositories include Docker, PHP, MySQL, PostgreSQL, and other packages.
 Root device type: ebs Virtualization type: hvm

▼ **Instance Type** [Edit instance type](#)

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

▼ **Security Groups** [Edit security groups](#)
 Security group name: launch-wizard-13
 Description: launch-wizard-13 created 2020-08-28T08:09:36.597+05:30

Type	Protocol	Port Range	Source	Description
SSH	TCP	22	0.0.0.0/0	
All traffic	All	All	0.0.0.0/0	

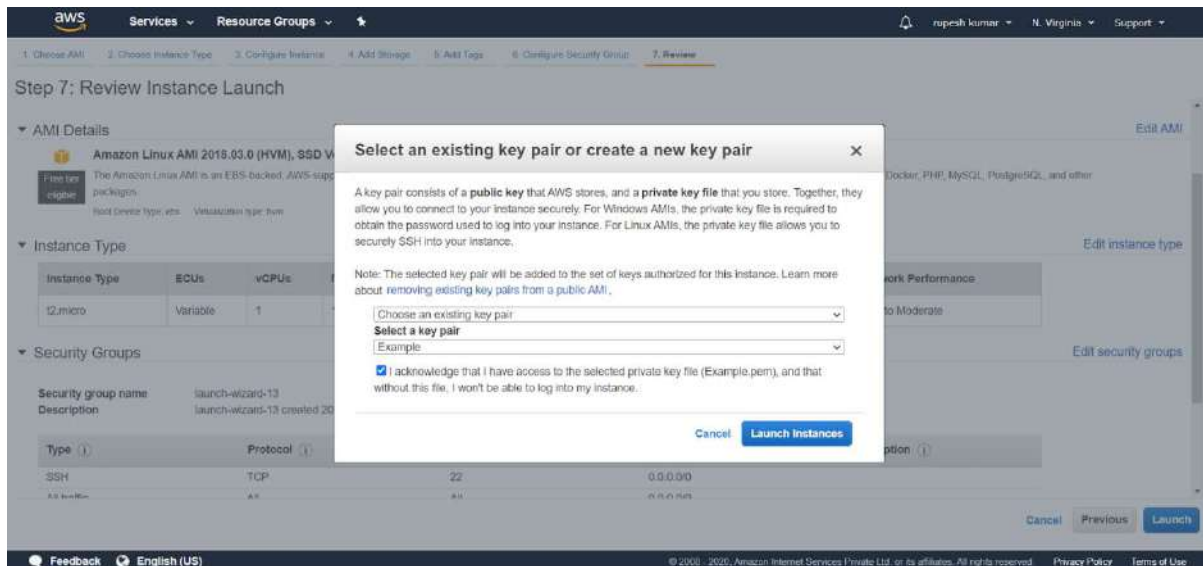
[Cancel](#) [Previous](#) [Launch](#)

On the “Select an existing key pair or create a new key pair” box

Select existing key pair

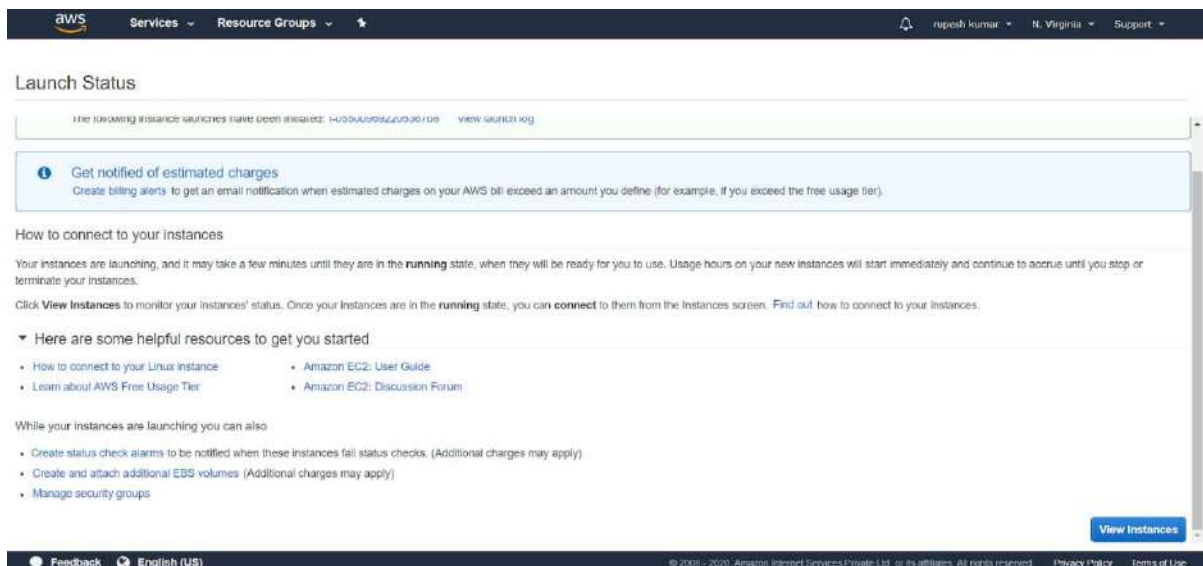
Key pair name → Example

Click on “Launch Instance” button



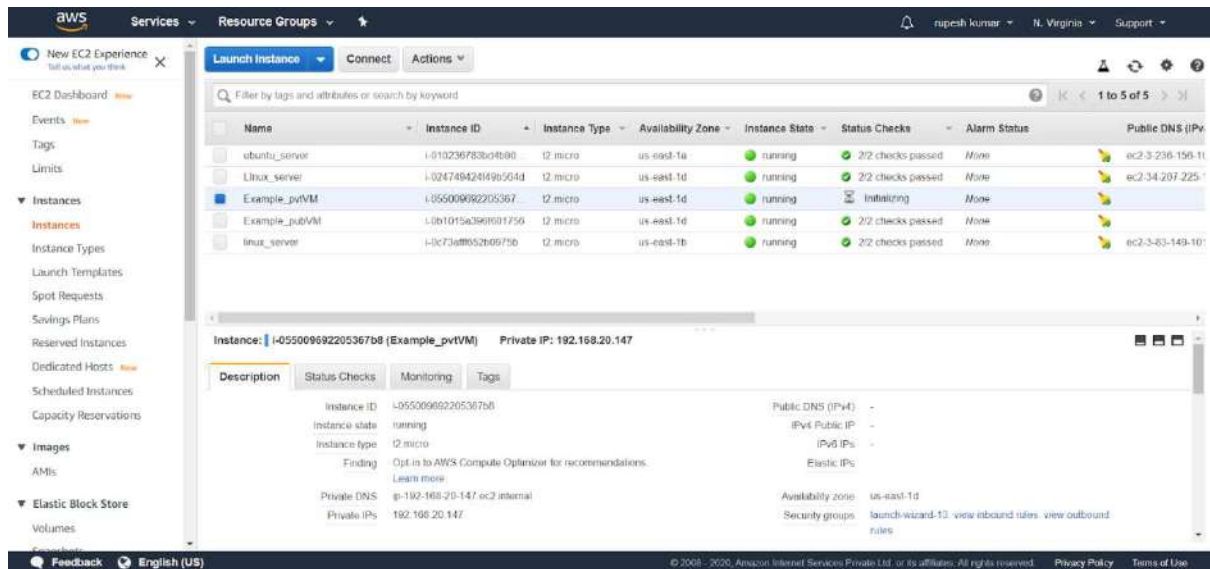
Check the summary

Click on View Instance button



Verification

Linux instance in public subnet is launched



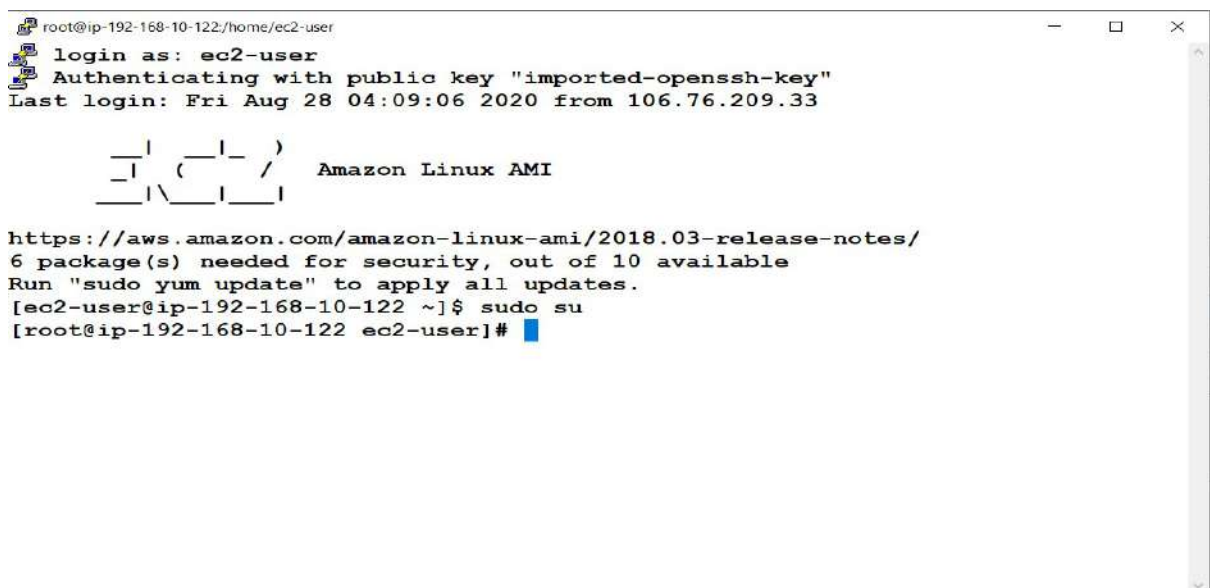
To connect to Linux private instance

First copy the key to Linux instance in public subnet

Now connect to Linux instance in public

Then connect to Linux instance in private

Connect to the Linux instance in public



Now connect to the Linux instance in private subnet using the private ip, the command is
Ssh -i ec2-user@192.168.20.147

```
root@ip-192-168-10-122:/home/ec2-user
login as: ec2-user
Authenticating with public key "imported-openssh-key"
Last login: Fri Aug 28 04:09:06 2020 from 106.76.209.33

 _ _ | _ _ | _ _ |
 _ | ( _ _ /   Amazon Linux AMI
 _ | \ _ _ | _ _ |

https://aws.amazon.com/amazon-linux-ami/2018.03-release-notes/
6 package(s) needed for security, out of 10 available
Run "sudo yum update" to apply all updates.
[ec2-user@ip-192-168-10-122 ~]$ sudo su
[root@ip-192-168-10-122 ec2-user]# ssh -i ec2-user@192.168.20.147
Warning: Identity file ec2-user@192.168.20.147 not accessible: No such file or d
irectory.
usage: ssh [-1246AaCfGgKkMnNqsTtVvXxYy] [-b bind_address] [-c cipher_spec]
[-D [bind_address:]port] [-E log_file] [-e escape_char]
[-F configfile] [-I pkcs11] [-i identity_file]
[-J [user@]host[:port]] [-L address] [-l login_name] [-m mac_spec]
[-O ctl_cmd] [-o option] [-p port] [-Q query_option] [-R address]
[-S ctl_path] [-W host:port] [-w local_tun[:remote_tun]]
[user@]hostname [command]
[root@ip-192-168-10-122 ec2-user]#
```

It gives the error message, now add the key pair related to the private subnet Linux instance and add the content of the keypair.
nano example.pem

```
root@ip-192-168-10-122:/home/ec2-user
GNU nano 2.5.3      File: example.pem

-----BEGIN RSA PRIVATE KEY-----
MIIEogIBAAKCAQEAKJaggB6GT06RLNSuOEtr3RW1SFJLFeFKwvLCHQauyptiZfaWU3IS+Eft2ef3
NGLw7Ks06B2n3ZpzzHoHhgdTIjh9N5p/UvsKf3vJml3WJ7Jwe2IV8qe7pa/ZdvUGk+PcF2VW2OtT
JQsofHBrjVZmaRE1xp45OUJKVNS1nRuHX1Xwc/Ck6vA9y1yYAV+OFNatPRRNIBk0nUy5JY0SZQkE
b/pW577eNnQ4jyYD6YHTd31A1xd+fe5LttGpYHg5PmAgNY0p2svKYSY3sLfqcVeDBanB9IVt5p/u
+3SFVoA4vIJI8kCCBAIdtxxwTzBcj9k3NHFCzUetPEyyOBUEVI6jzwIDAQABAoIBAG+FjtrJNusy
yW0PujU3j0HecTbAKqP8uoJ1Zd2niuhBQ3sr4DUKtrEEIErCM/0XF4ckYtFqF1Ep0Y100hrByZCK
i6J4qx5g/W7pFs3W9Nh3nKS+OfmJAIZChXRod6NYTC1Lg9oaYG4hFiAtQfepHaKKoLOJ9q9Afz7B
NjewxRB1daQ1W9kKkq1R4+1ZiY4IBj6edvGnv99HuZMzoSa8+MUXTRd2FNMmr04e5YjeNY8VfgTqc
HF1UG3E4BWOMDQA0UqJvHk+X8uq88gb8G37yJGI1j43bQ9+Pw2ikNMfR7kZrXEZbwG5agUbdjm5a
DrqDz3PJeJLo9qm8u0CQnab0LqkCgYEA6NppDG2vYoL++uDjEObwFA1G0qIOEv7QdEOJ/MMv+vdm
1kJd3Dh9kdYk17wjAvWlnznVp2/VeIFUREzDGX1ke4tLyjkz8TLL901UUQ2KuFssuxDRABtPpkYq
CQKs4UhdRYjtlwxHFGy+cvmy8J+Ergptuwrz7hIikTANbNH7jgUCgYEAnvYTjiVQJwgVHoH+a+ly
j8YzOQBbX7vUn4p6YSpwXRGRKefSqVQVTCQjXpb8NtRTH1By6gYonzw2sn+gpBNP83P7RLm+clMn
+81M1YpmijZxc2633NUIegsegmeStZgk9/zRs2+j/Cv95Cd80Y2NM5KmsjMN/KSuPk1V9ozPfsMC
gYAFta1Tv7DIQpwL/M20kWUUbqMOu0Ih10Me9whY1G3gmuEBMu03iQ4RYuh6F1fhpzyozgFCL3YMn
hExTrGowSri7Csxd9g8e//beZm7p2eIn6RxsrdniJ/ypABlxxR4GHCaCIvxcocsWfa5cz7ImFuvOe
7OSJ7JfUo56QKNUepvt7HQBGE+m0F+utL4KGFrrqoljiTiRayGbTbfXHGaA73FypzknkTo/n+Lge
SNmxVgPEtjanJ8sN1rcLRPmdS28f4pe0kaBU4ngqdpKEtolG8vQPuJfS9gA4X2sJZVAwGCARzYKU

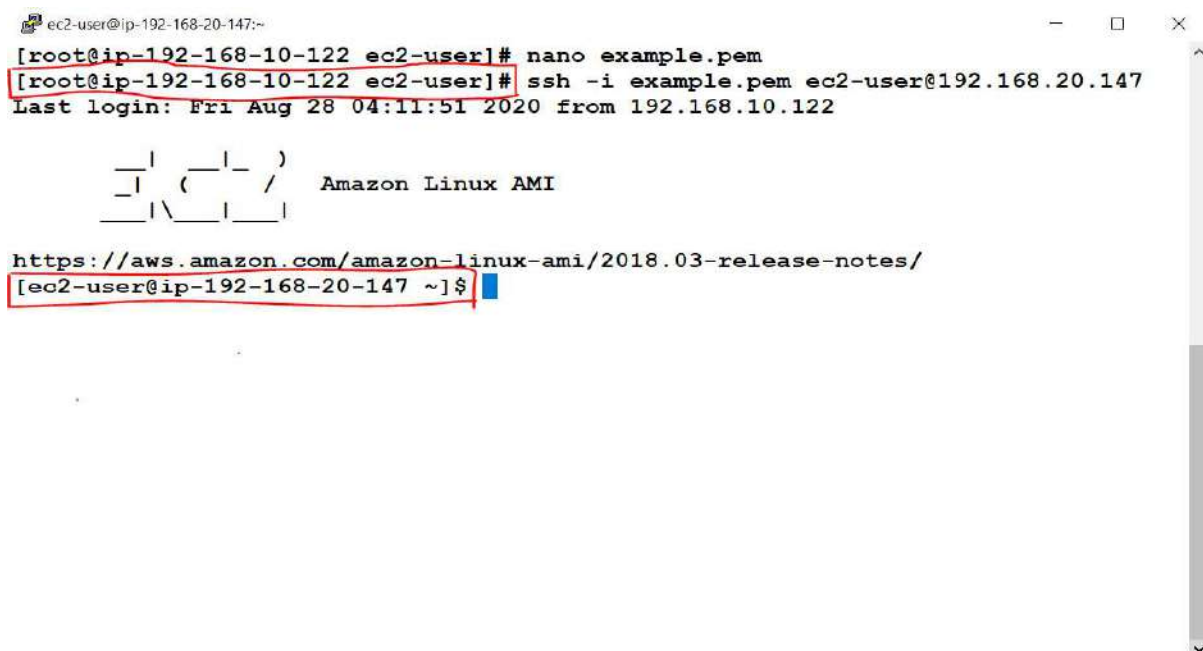
[ Read 23 lines ]
^G Get Help      ^O Write Out    ^W Where Is     ^K Cut Text     ^J Justify      ^C Cur Pos
^X Exit          ^R Read File    ^\ Replace      ^U Uncut Text   ^T To Spell     ^_ Go To Line
```

Now run the command to connect to the Linux instance in the private subnet



```
root@ip-192-168-10-122:/home/ec2-user
[root@ip-192-168-10-122 ec2-user]# nano example.pem
[root@ip-192-168-10-122 ec2-user]# ssh -i example.pem ec2-user@192.168.20.147
```

After running the command, the prompt was changed from the public subnet to the private subnet



```
ec2-user@ip-192-168-20-147:~
[root@ip-192-168-10-122 ec2-user]# nano example.pem
[root@ip-192-168-10-122 ec2-user]# ssh -i example.pem ec2-user@192.168.20.147
Last login: Fri Aug 28 04:11:51 2020 from 192.168.10.122

  _ _ | _ _ | _ )
 _ | ( _ _ /   Amazon Linux AMI
 _ | \ _ _ | _ |

https://aws.amazon.com/amazon-linux-ami/2018.03-release-notes/
[ec2-user@ip-192-168-20-147 ~]$
```

Now we are successfully connected to the private subnet.