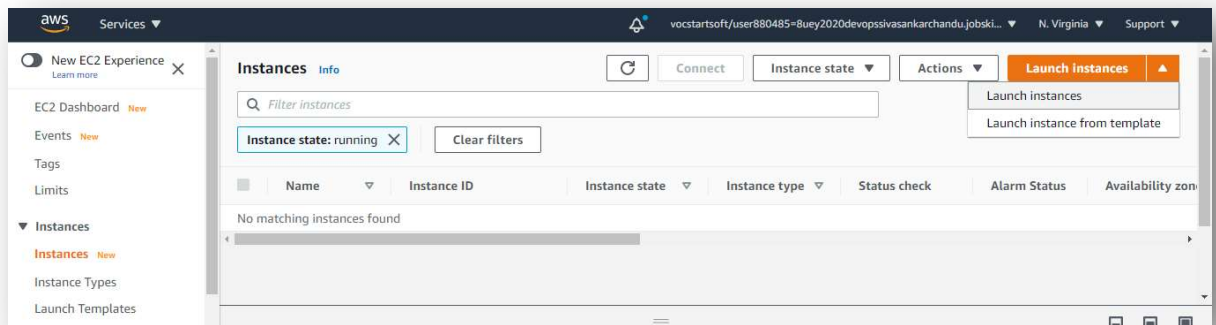
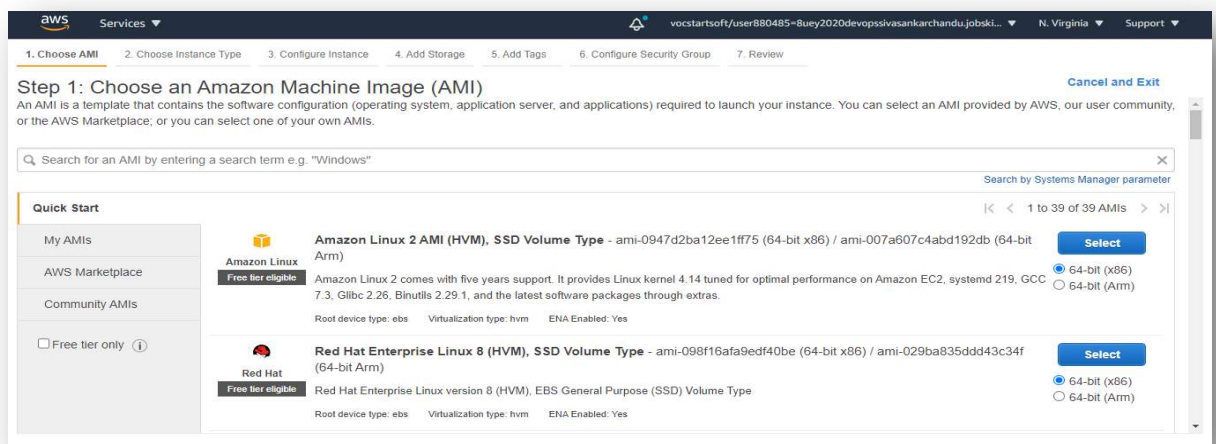


EC2 Instance (Amazon Linux) – Git Bash

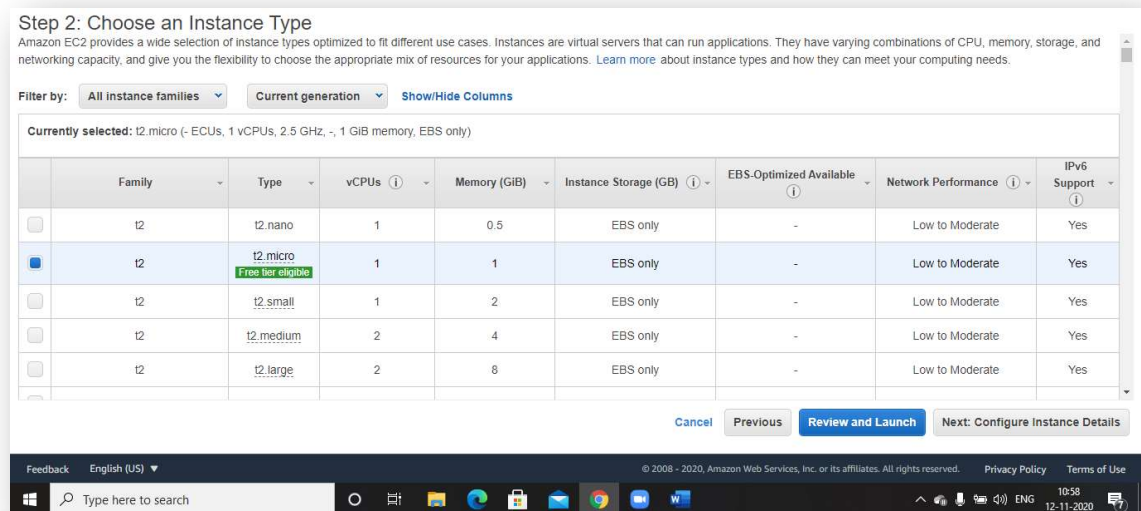
- First, Open AWS Console and Search for EC2 Service then click on EC2 service. We directed to EC2 page, click on running instances then we find below page. Now click on Launch Instances.



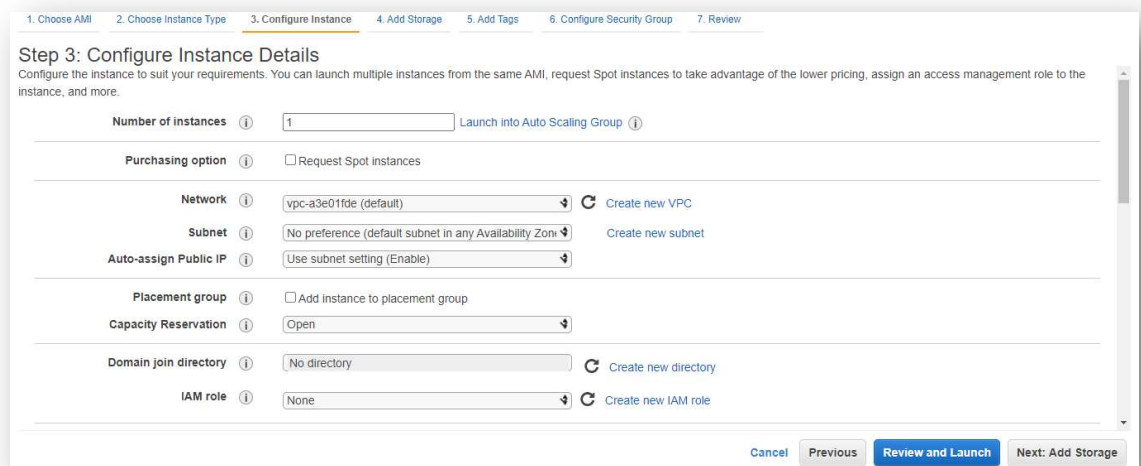
- Now select Amazon Machine Image (AMI), An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. You can select an AMI provided by AWS, our user community, or the AWS Marketplace; or you can select one of your own AMIs. Select Amazon Linux 2 AMI then forwarded to Next step.



- Now, choose 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. After choosing Instance type based on requirement click Next to Configure Instance Details.



- Now, based on requirement Select the VPC, Subnet, Public IP, shutdown behaviour, stop hibernate, Termination protection, Monitoring and Use bash code in User data to automatically install and run application. Click next to add storage.



- Now add storage to EC2 Instance. I am not adding any volume so I am going to next step.

The screenshot shows the 'Step 4: Add Storage' configuration page in the AWS Management Console. The page has a progress bar at the top with steps: 1. Choose AMI, 2. Choose Instance Type, 3. Configure Instance, 4. Add Storage (highlighted), 5. Add Tags, 6. Configure Security Group, and 7. Review. Below the progress bar, the title 'Step 4: Add Storage' is followed by explanatory text. A table lists the storage configuration for the root volume. The table has columns: Volume Type, Device, Snapshot, Size (GiB), Volume Type, IOPS, Throughput (MB/s), Delete on Termination, and Encryption. The root volume is configured with a size of 8 GiB, General Purpose SSD (gp2) volume type, 100 IOPS, 3000 MB/s throughput, and is set to 'Delete on Termination' and 'Not Encrypted'. Below the table is an 'Add New Volume' button and a blue informational box about free tier eligibility. At the bottom right are buttons for 'Cancel', 'Previous', 'Review and Launch', and 'Next: Add Tags'.

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

- Now assign some Tags to EC2 Instance. Key is Name and Value is First Instance then Click Next to Configure Security Group.

The screenshot shows the 'Step 5: Add Tags' configuration page in the AWS Management Console. The progress bar at the top highlights step 5. The title 'Step 5: Add Tags' is followed by explanatory text. Below the text is a table for adding tags. The table has columns: Key (128 characters maximum), Value (256 characters maximum), Instances, and Volumes. A tag is added with the key 'Name' and value 'First Instance', with checkboxes for 'Instances' and 'Volumes' both checked. Below the table is an 'Add another tag' button with the text '(Up to 50 tags maximum)'. At the bottom right are buttons for 'Cancel', 'Previous', 'Review and Launch', and 'Next: Configure Security Group'.

Key (128 characters maximum)	Value (256 characters maximum)	Instances	Volumes
Name	First Instance	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

- Now, we can select either new Security Group or existing Security Group. Security Group have set of firewall rules that controls the traffic of EC2 Instance. Select Security Group and Add new rule to it. Click next review the selected options and launch the EC2 Instance.

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:

Description:

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](#)

- Review the EC2 Instance.

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

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.

Warning
Improve your instances' security. Your security group, launch-wizard-4, 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](#)

▼ **AMI Details** [Edit AMI](#)

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.

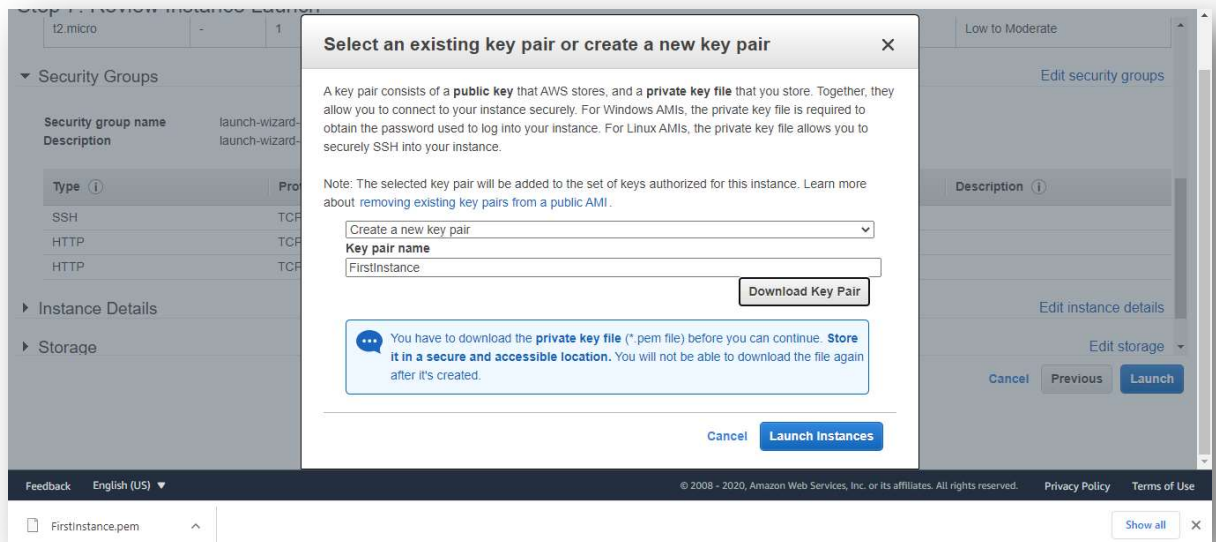
Root Device Type: ebs Virtualization type: hvm

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

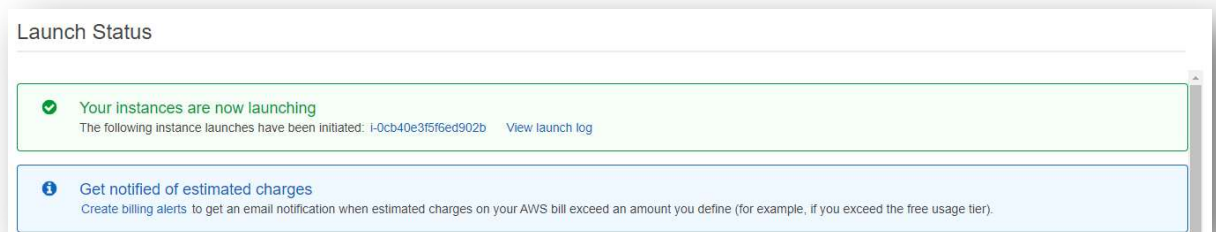
Instance Type	ECUs	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance
t2.micro	-	1	1	EBS only	-	Low to Moderate

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

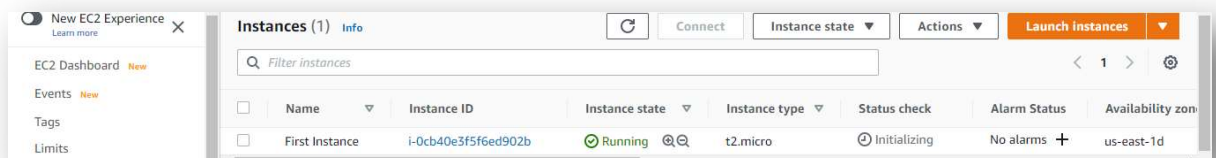
- After click on launch we select the Private key pair. Create new key pair and download it otherwise choose existing if you have any keypair. I am selecting new key pair and key pair downloaded. Now I am click on Launch Instances button to launch my Amazon Linux Instance.



- Launch status

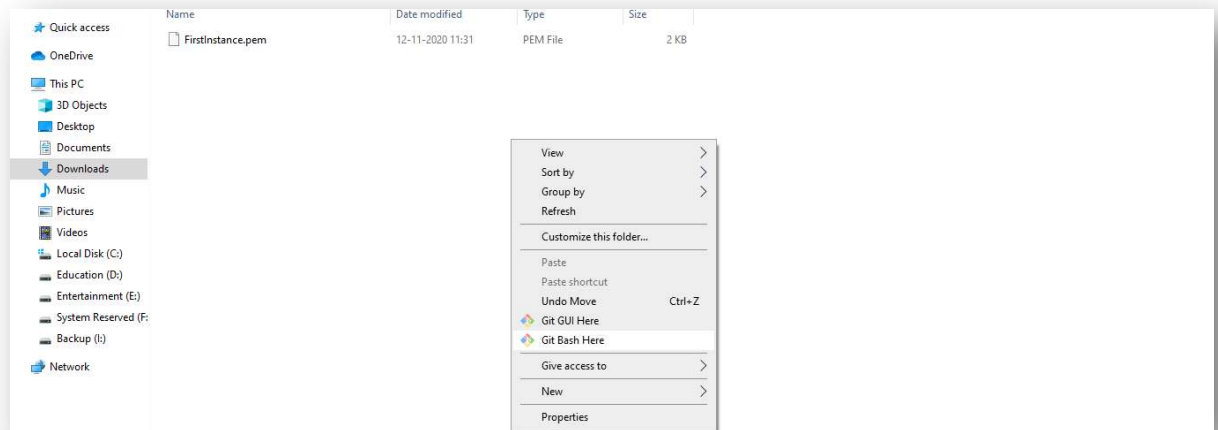


- My Instance state Running. Now connect instance through the Git Bash.

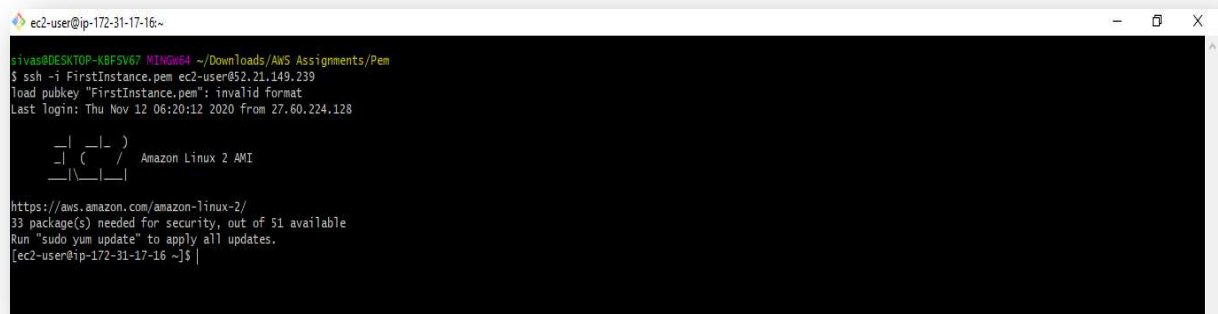


Connecting Instance through Git Bash...

- First Install Git Bash. After installing Git Bash open folder where the pem stored. Right Click on white space and then click on Git Bash here.



- Now paste the code (`ssh -i FirstInstance.pem ec2-user@PublicIP_Address`). Copy the Public IP address from EC2 Instance and Click Enter.



EC2 Instance Connected...

Now run “**sudo yum update**” to apply all the latest updates to EC2 Instance