St. Francis Institute of Technology, Mumbai-400 103

Department Of Information Technology

A.Y. 2024-2025

Class: TE-ITA/B, Semester: V

Subject: **DevOps Lab**

Experiment – 10: To learn Pull based Software Configuration Management and provisioning tools using Puppet.

1. **Aim:** To install and Configure Pull based Software Configuration Management and provisioning tools using Puppet.
2. **Objectives:** Aim of this experiment is that, the students will learn:
   - To Synthesize software configuration and provisioning using Puppet
   - To Build and operate a scalable automation system.
3. **Outcomes:** After study of this experiment, the students will learn following:
   - Architecture of Puppet
   - Puppet Master Slave Communication
   - Configuring Puppet Master and Agent on Linux machines
4. **Prerequisite:** None
5. **Requirements:** AWS account, putty, Personal Computer, Windows operating system, Internet Connection, Microsoft Word.
6. **Pre-Experiment Exercise:**

   Brief Theory: Refer shared material
7. **Laboratory Exercise**

   A. Procedure:

   a. Answer the following:
   - What is Puppet? Enlist its features.
   - Explain architecture of Puppet with a diagram.

   b. Refer the shared material and do online research to answer following:

   i. Mention steps for creating 2 EC2 instances with Ubuntu OS on AWS for creating master and slave machines. Attach screenshots for the same.

   ii. Commands used to configure master and slave machines using putty.

   Commands to run on puppet Master (which is one of the EC2 instances)
   - sudo apt-get update :This command will update the packages
   - wget https://apt.puppetlabs.com/puppet-release-bionic.deb :This command will download the puppet folder
   - sudo dpkg -i puppet-release-bionic.deb :This command will unzip the puppet folder
   - sudo apt-get install puppetmaster : This command will install the puppet master
   - apt policy puppetmaster : This command will verify puppet master after installation
   - sudo systemctl status puppet-master.service: This command will check status of puppet master service
   - sudo nano /etc/default/puppet-master: This command will fine tune some settings….

- Add this line in the puppet master file: JAVA_ARGS="-Xms512m -Xmx512m"
  This command will change the memory allocation to 512MB
- sudo systemctl restart puppet-master.service : This command will restart puppet master after the recent changes
- sudo ufw allow 8140/tcp : This command will open TCP port for puppet to communicate
- sudo nano /etc/hosts : This command will open hosts file for entering master's IP address
- sudo puppet cert list : This command will show puppet agent's certificate received for signing
- sudo puppet cert sign –all : This command will sign the received certificate

Commands to run on slave node/ puppet agent (which is the other EC2 instance)

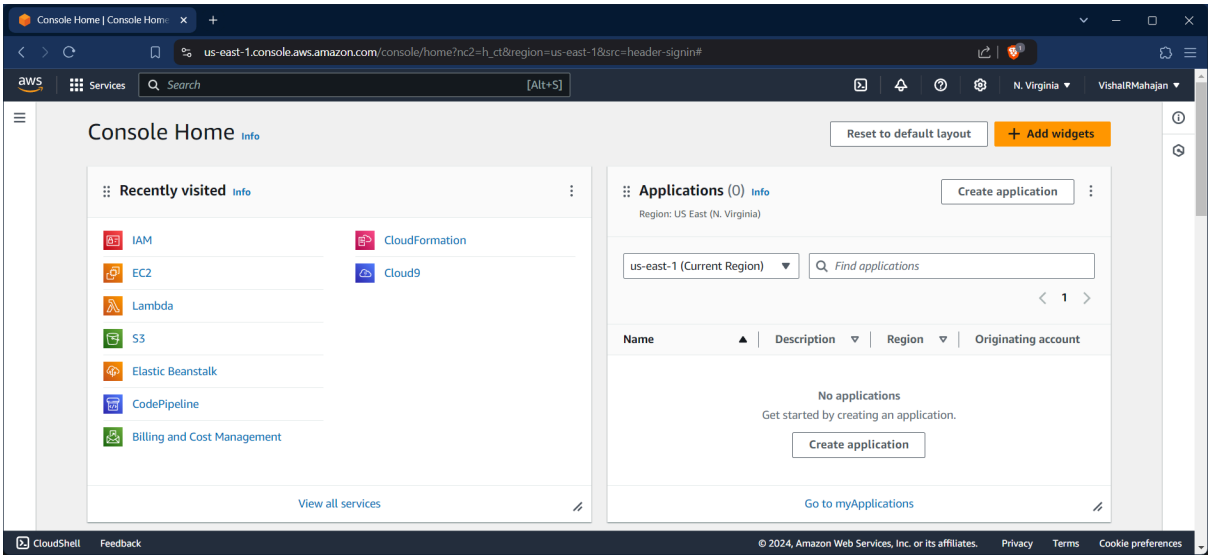
- sudo apt-get update: This command will update the packages
- wget https://apt.puppetlabs.com/puppet-release-bionic.deb This command will download the puppet folder
- sudo dpkg -i puppet-release-bionic.deb :This command will unzip the puppet folder
- sudo apt-get install puppet :This command will install the puppet agent
- sudo nano /etc/hosts :This command will open hosts file for entering master's IP address
- sudo systemctl start puppet :This command will start the puppet agent
- sudo systemctl enable puppet :This command will enable the puppet agent
- sudo puppet agent --test :This command will test communication between puppet master

8. **Post-Experiments Exercise**
   A. **Extended Theory:**
      Nil
   B. **Questions:**
      - Explain the two types of configuration management approaches.
      - How does the connection between puppet master server and puppet agent nodes happen?
   C. **Conclusion:**
      - Write what was performed in the experiment.
      - Write the significance of the topic studied in the experiment.
   D. **References:**
      https://www.edureka.co/blog/puppet-tutorial/
      https://www.simplilearn.com/puppet-tutorial-article

Creating EC2 Instances (Master and Slave)

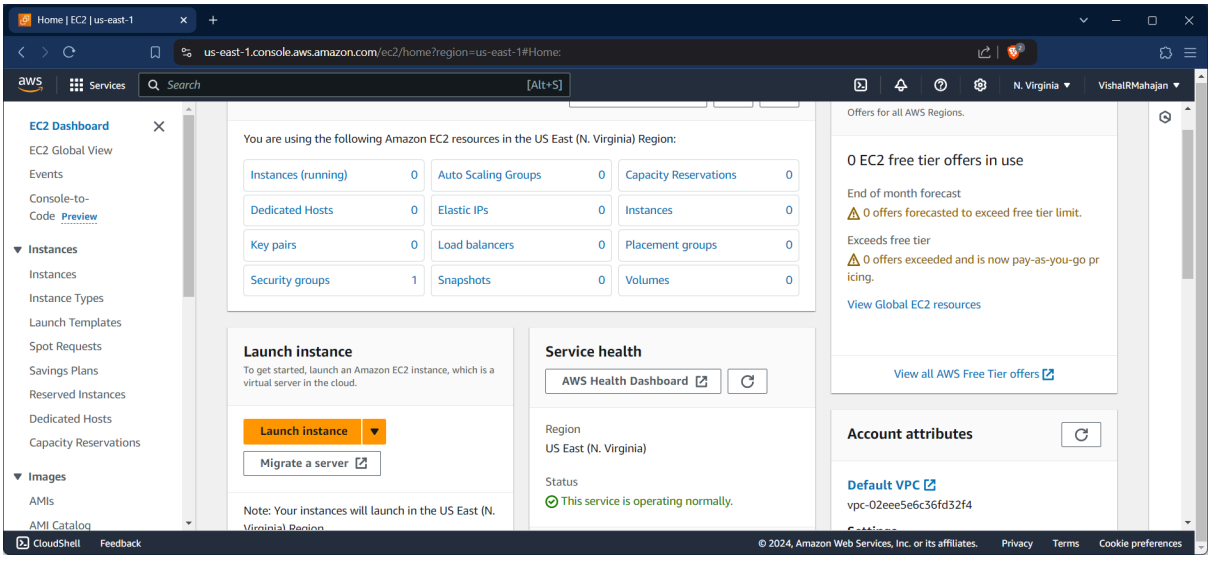
Step 1: Login to AWS Management Console

1. Go to aws.amazon.com.
2. Log in with your AWS credentials.

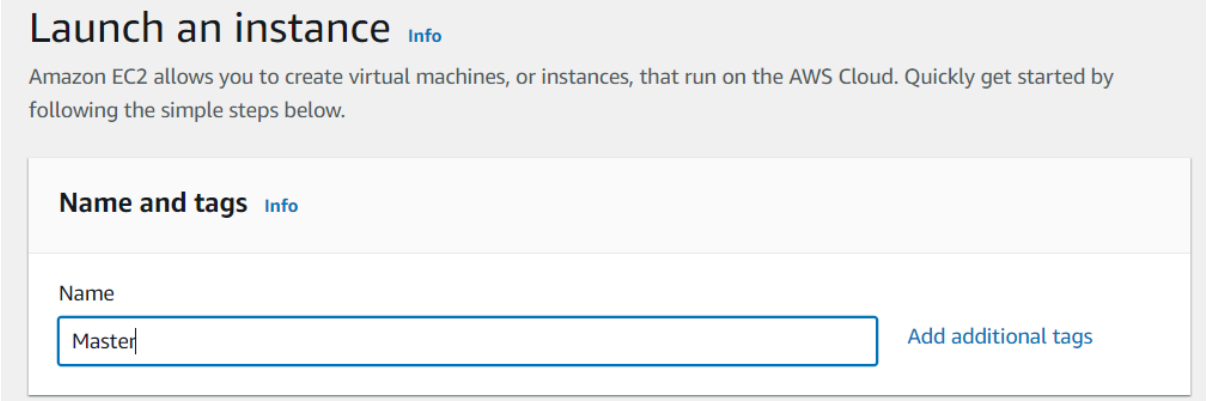


Step 2: Launch the Master EC2 Instance

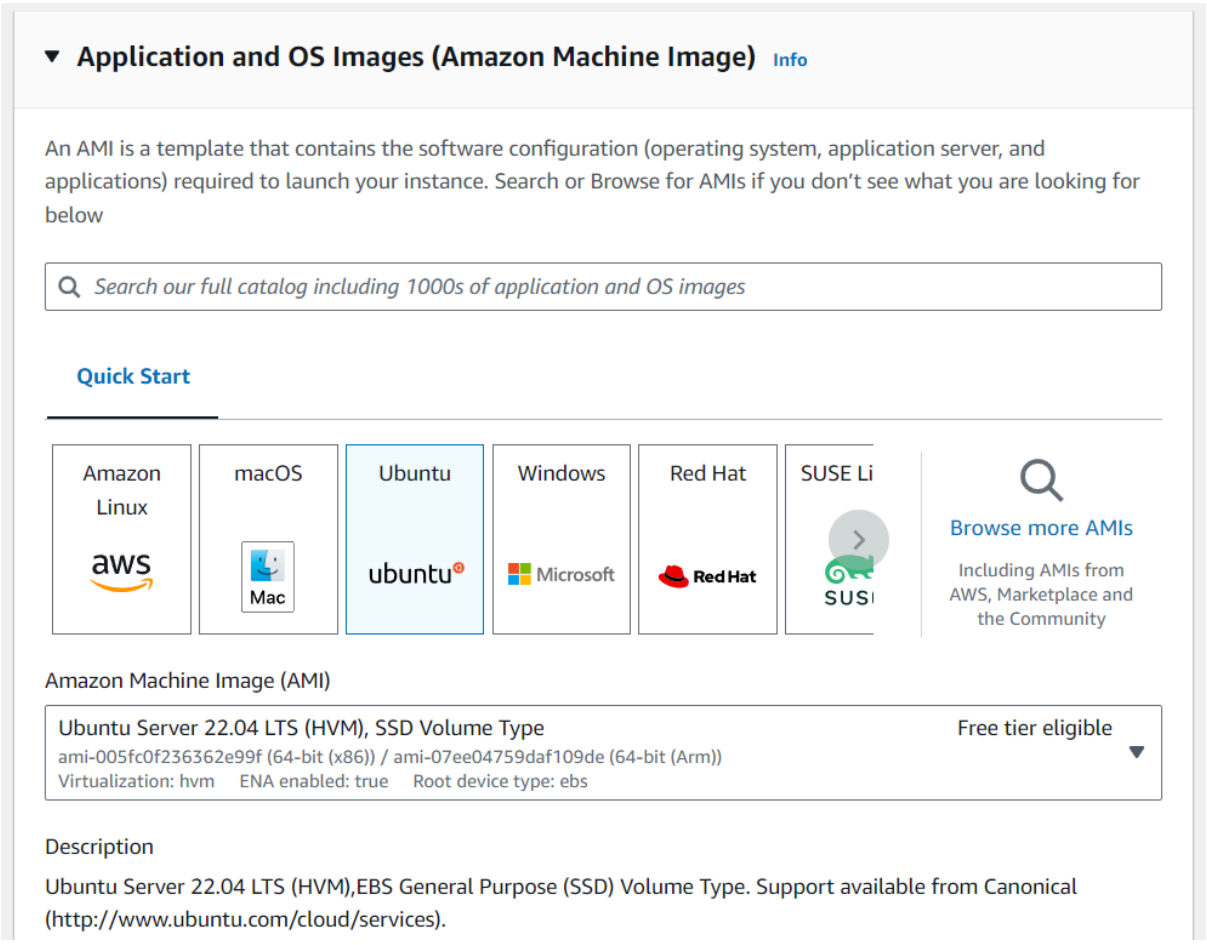
1. **Go to EC2 Dashboard**:
   - In the search bar at the top, type **EC2** and select it.
2. **Click "Launch Instance"**:
   - On the EC2 Dashboard, click the **Launch Instance** button.



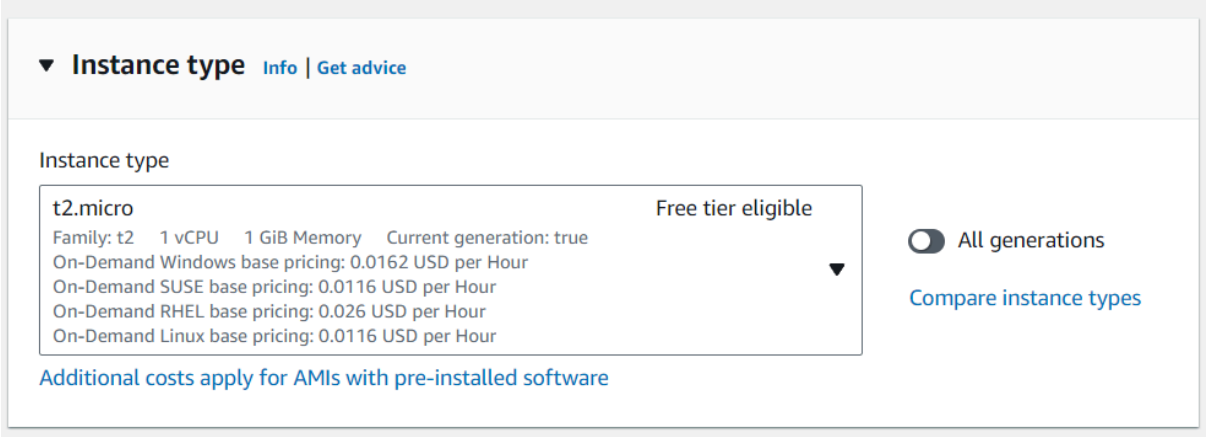
3. **Name Your Instance**:
    - In the **Name and Tags** section, set the **Name** as `Master`.



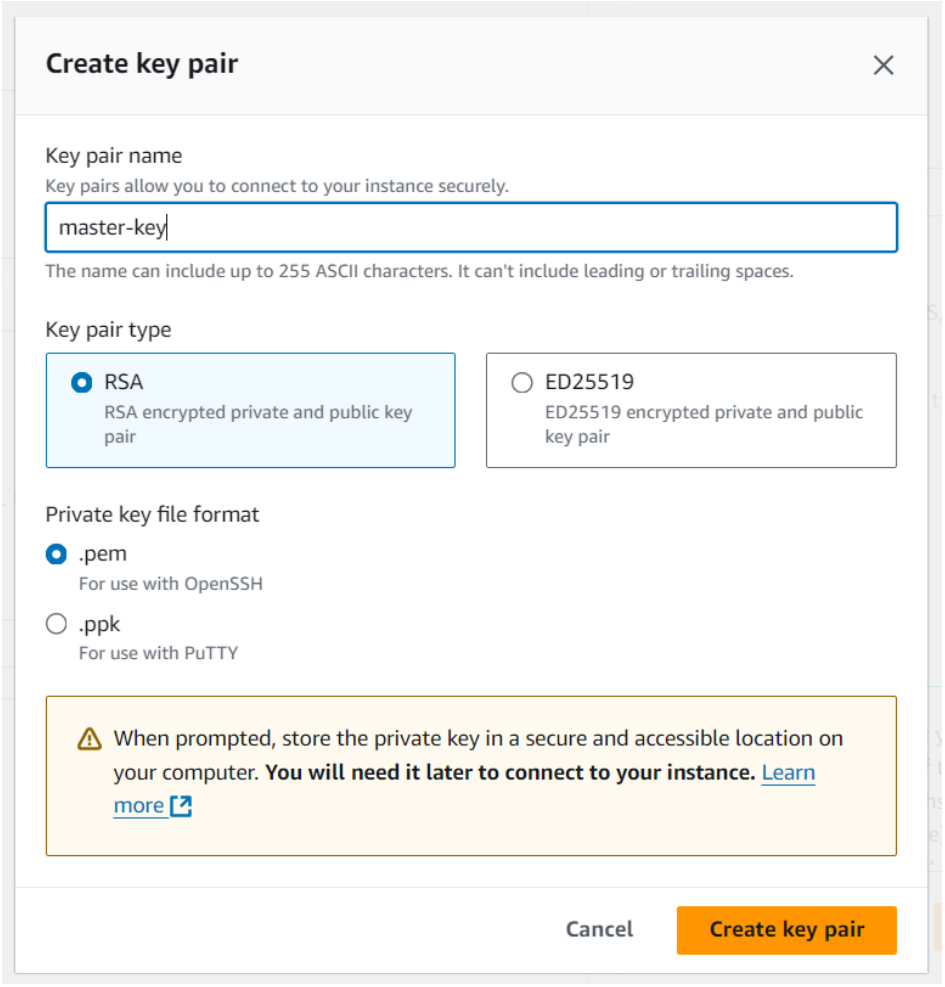
4. **Choose AMI (Amazon Machine Image)**:
    - Select **Ubuntu Server 22.04 LTS (Free tier eligible)**.



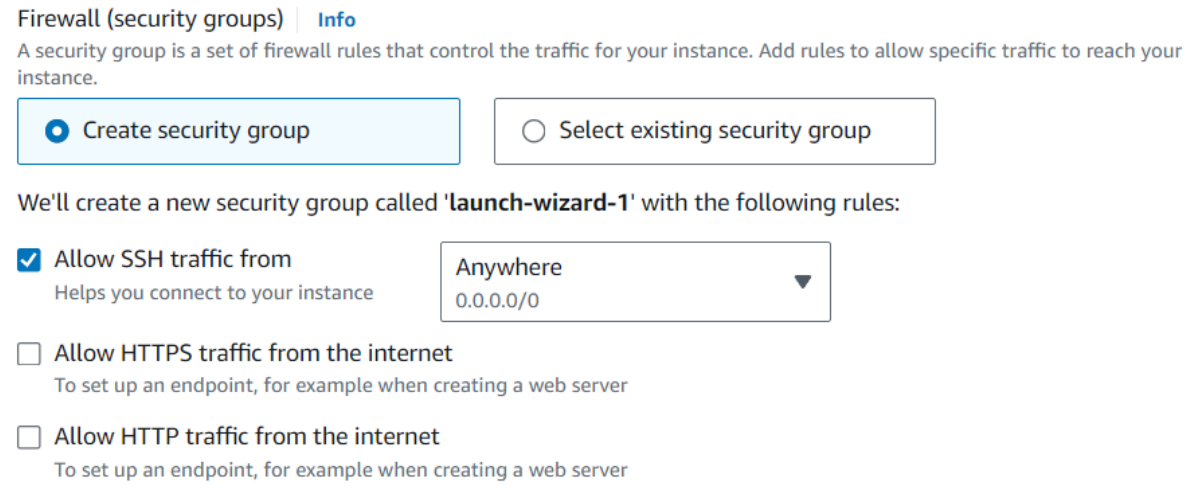
5. **Select Instance Type**:
    - Select **t2.micro** (Free Tier Eligible).



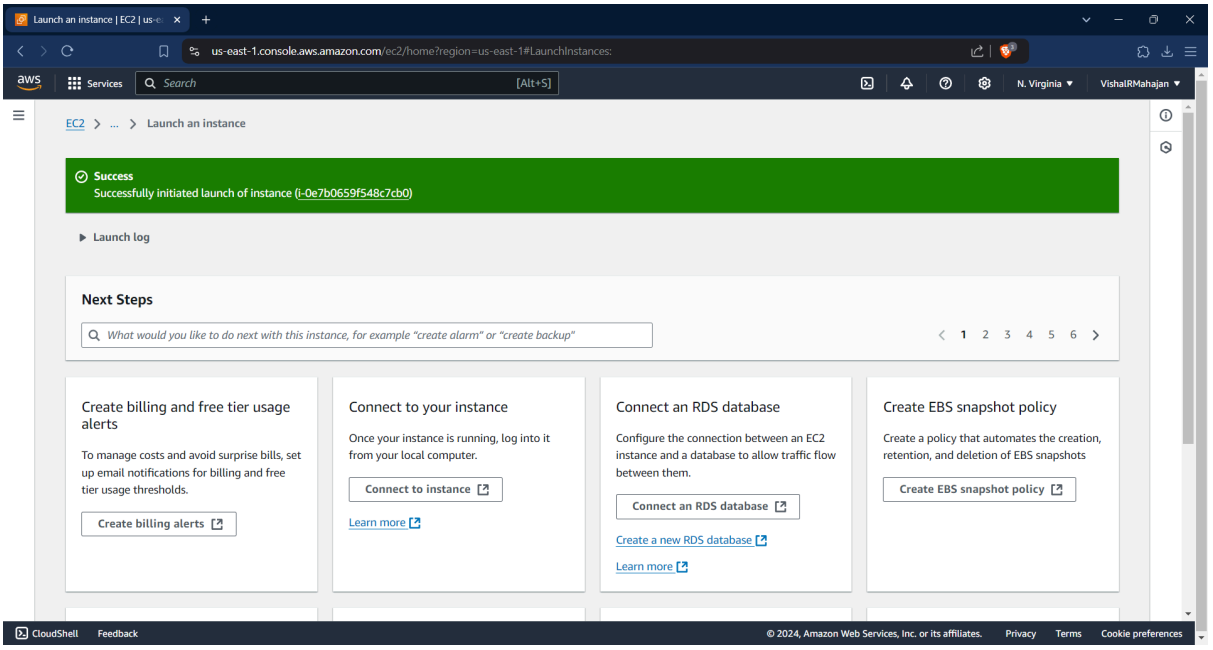
6. **Key Pair (Login)**:
    - Either **create a new key pair** (e.g., master-key.pem) or choose an existing key pair.
    - **Download the key pair** and save it securely. You'll use it to SSH into the instance.



7. **Security Group Settings**:
    - Ensure **SSH (port 22)** is allowed. AWS will pre-configure this for you.

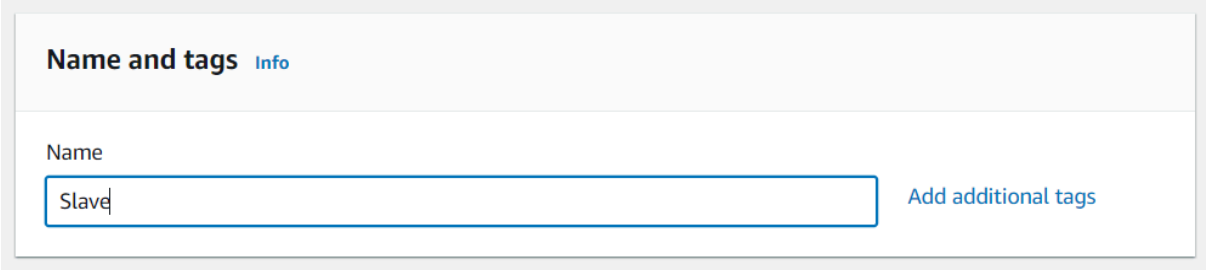


8. **Launch Instance**:
    - Click **Launch Instance** to create the Master EC2 instance.

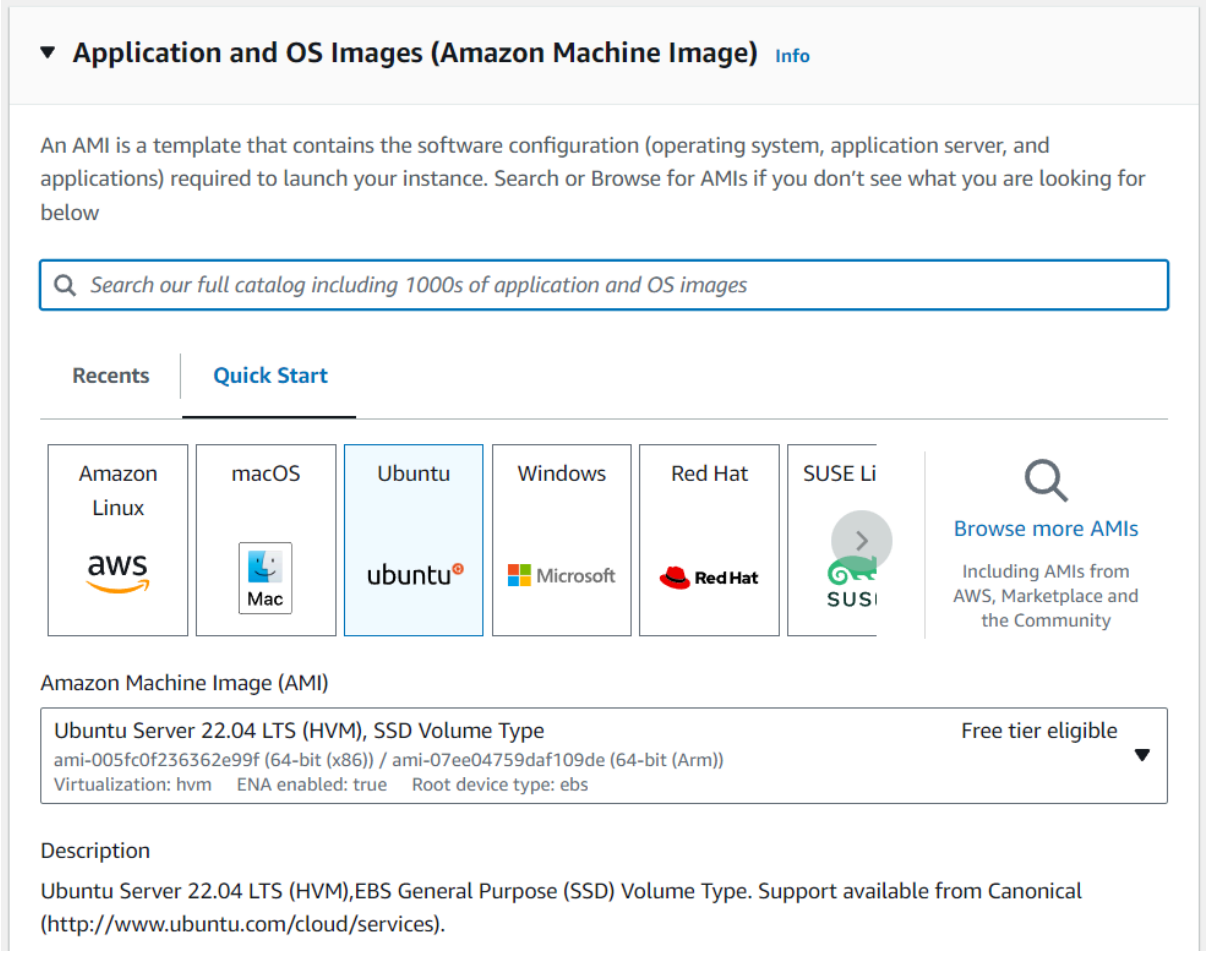


Step 3: Launch the Slave EC2 Instance

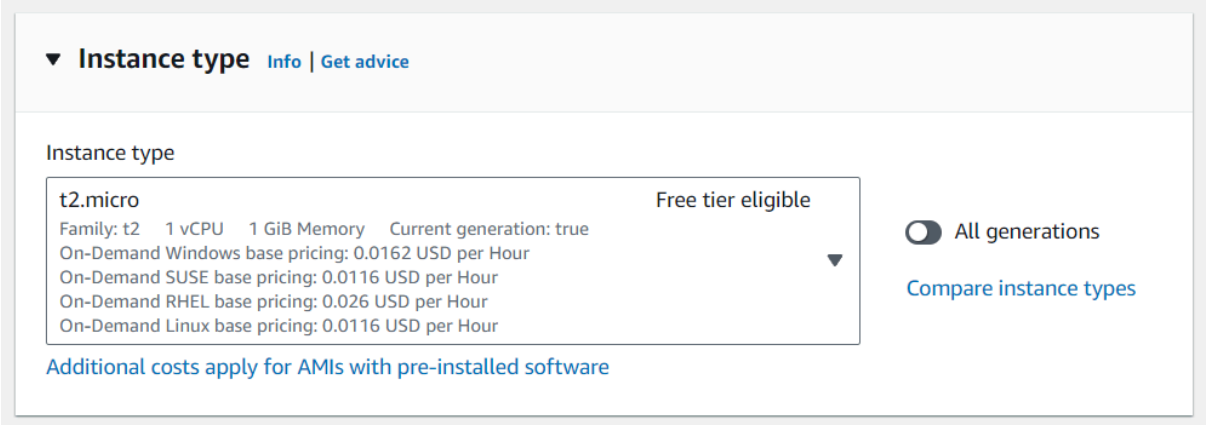
1. **Click "Launch Instance" Again**:
    - From the EC2 Dashboard, click **Launch Instance** again to create the Slave instance.
    -
2. **Name Your Instance**:
    - Set the **Name** as `Slave` in the **Name and Tags** section.



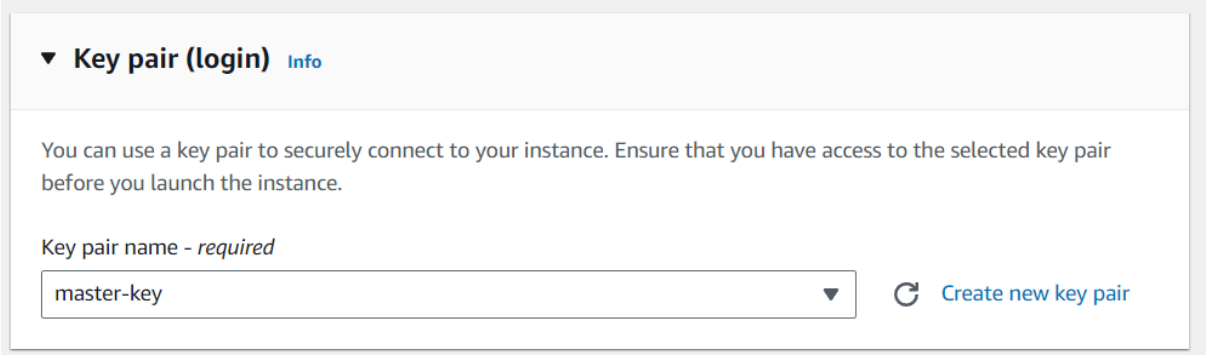
3. **Choose AMI (Amazon Machine Image)**:
    - Select **Ubuntu Server 22.04 LTS (Free tier eligible)**.



4. **Select Instance Type**:
    - Again, select **t2.micro**.



5. **Key Pair (Login)**:
    - Use the same key pair you created for the Master instance (or create a new one).
    - Ensure you download the key pair if you created a new one.



6. **Security Group Settings**:
    - Allow **SSH (port 22)** so that you can connect to the Slave instance.

Firewall (security groups) | Info

A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

(•) Create security group ( ) Select existing security group

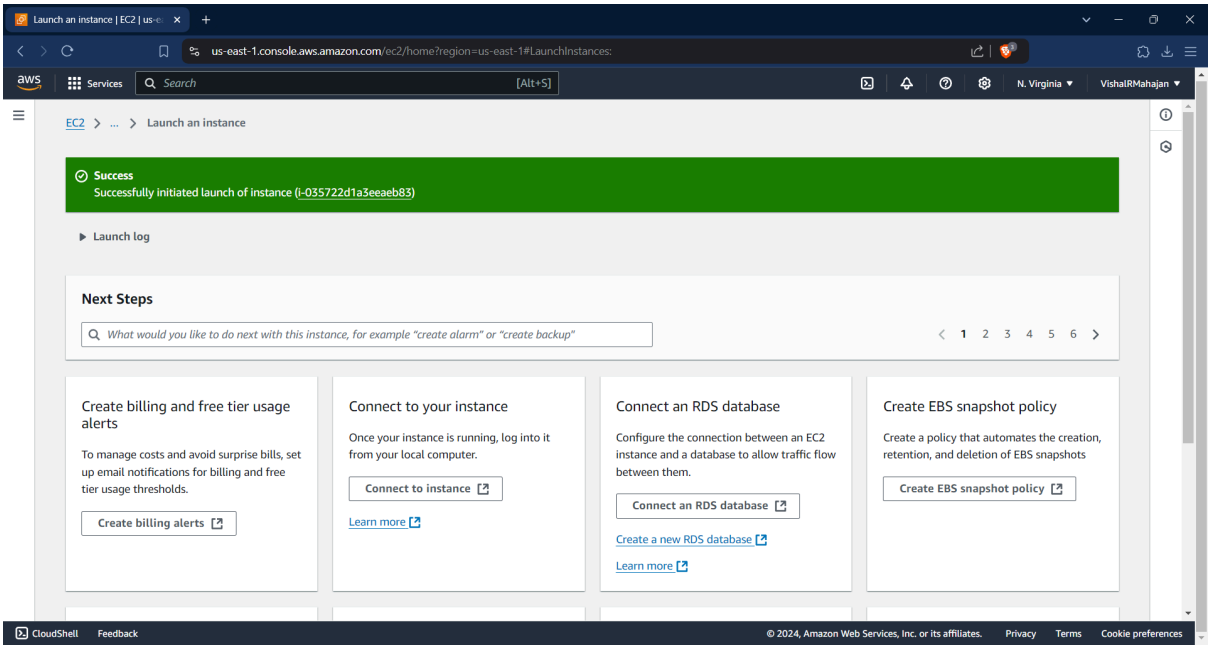
We'll create a new security group called '**launch-wizard-2**' with the following rules:

[x] Allow SSH traffic from
Helps you connect to your instance
Anywhere
0.0.0.0/0

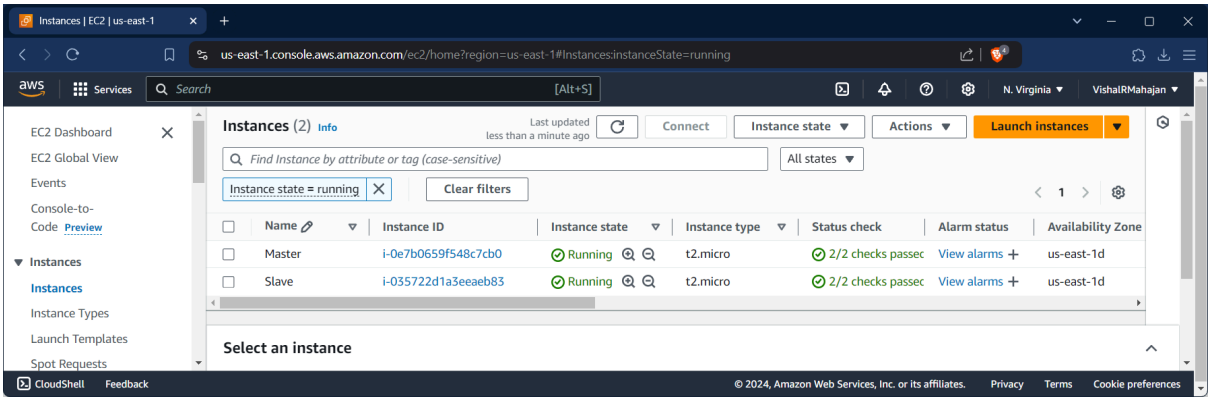
[ ] Allow HTTPS traffic from the internet
To set up an endpoint, for example when creating a web server

[ ] Allow HTTP traffic from the internet
To set up an endpoint, for example when creating a web server

7. **Launch Instance**:
   - Click **Launch Instance** to create the Slave EC2 instance.



EC Dashboard Showing Both Master and Slave Instances



Commands to run on puppet Master (which is one of the EC2 instances)

- sudo apt-get update :This command will update the packages
- wget https://apt.puppetlabs.com/puppet-release-bionic.deb :This command will download the puppet folder
- sudo dpkg -i puppet-release-bionic.deb :This command will unzip the puppet folder
- sudo apt-get install puppetmaster : This command will install the puppet master
- apt policy puppetmaster : This command will verify puppet master after installation
- sudo systemctl status puppet-master.service: This command will check status of puppet master service
- sudo nano /etc/default/puppet-master: This command will fine tune some settings….
- Add this line in the puppet master file: JAVA_ARGS="-Xms512m -Xmx512m".This command will change the memory allocation to 512MB
- sudo systemctl restart puppet-master.service : This command will restart puppet master after the recent changes
- sudo ufw allow 8140/tcp : This command will open TCP port for puppet to communicate
- sudo nano /etc/hosts : This command will open hosts file for entering master's IP address
- sudo puppet cert list : This command will show puppet agent's certificate received for signing
- sudo puppet cert sign –all : This command will sign the received certificate

Commands to run on slave node/ puppet agent (which is the other EC2 instance)

- sudo apt-get update: This command will update the packages
- wget https://apt.puppetlabs.com/puppet-release-bionic.deb This command will download the puppet folder
- sudo dpkg -i puppet-release-bionic.deb :This command will unzip the puppet folder
- sudo apt-get install puppet :This command will install the puppet agent
- sudo nano /etc/hosts :This command will open hosts file for entering master's IP address
- sudo systemctl start puppet :This command will start the puppet agent
- sudo systemctl enable puppet :This command will enable the puppet agent
- sudo puppet agent --test :This command will test communication between puppet master