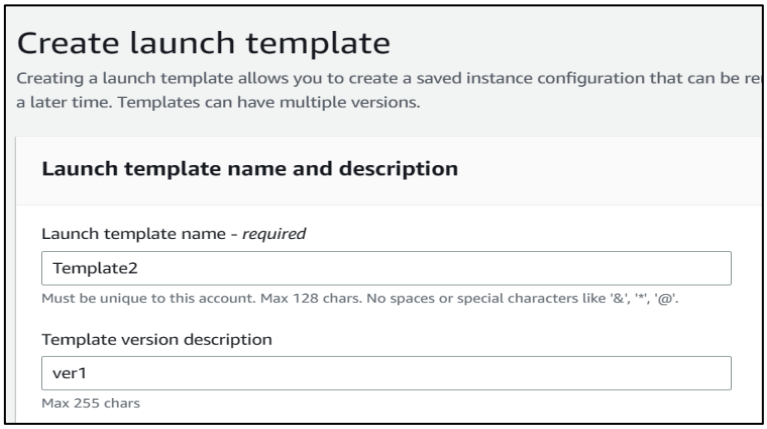
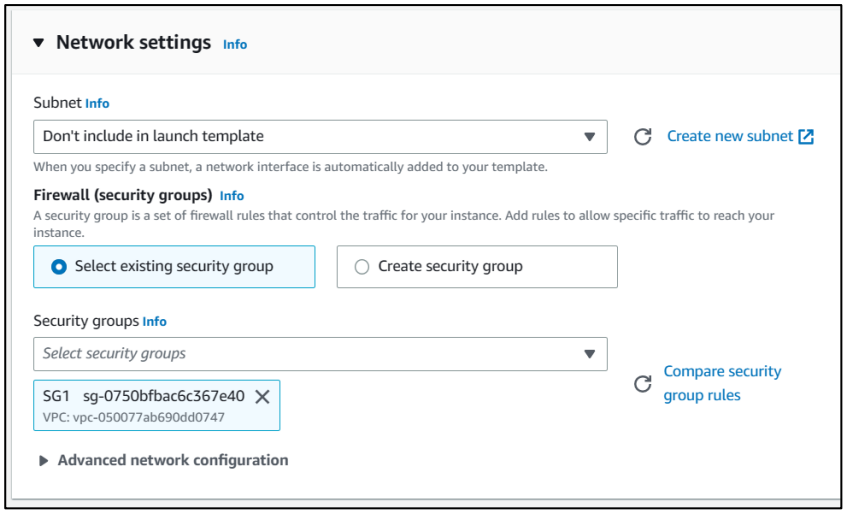
Assignment 11: Build Scaling plans in AWS that balance load on different EC2 instances.

1. Creating Launch Template
   a) In the left panel, under instances, go to **Launch Templates** option and click on **Create launch template.**
   b) Give name and description of the template and select the checkbox. Under **Application and OS Images** under **Quick Start** select **Ubuntu**, under **Instance type** select **t2.micro**, create a key pair.

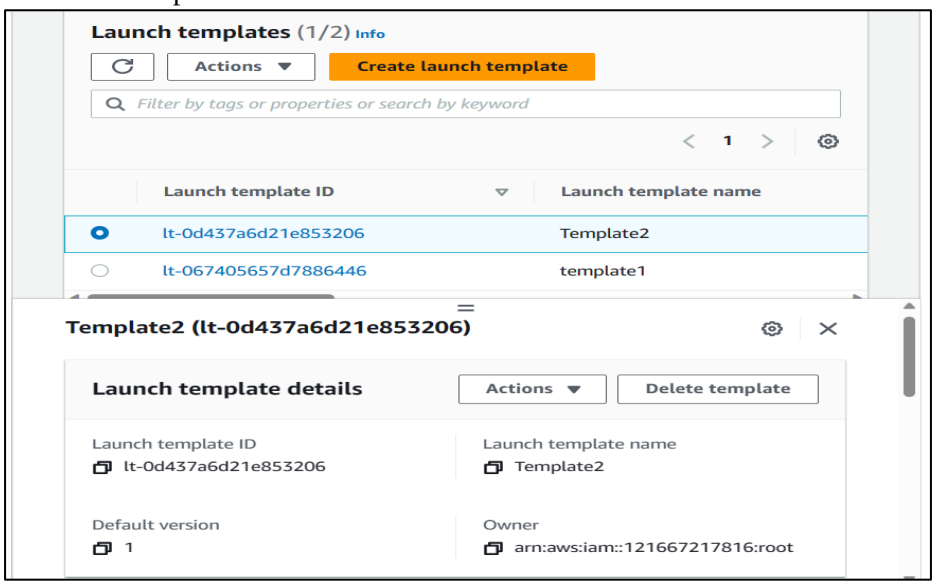


   c) Under **Network Settings**, select the **Security Group** created before.

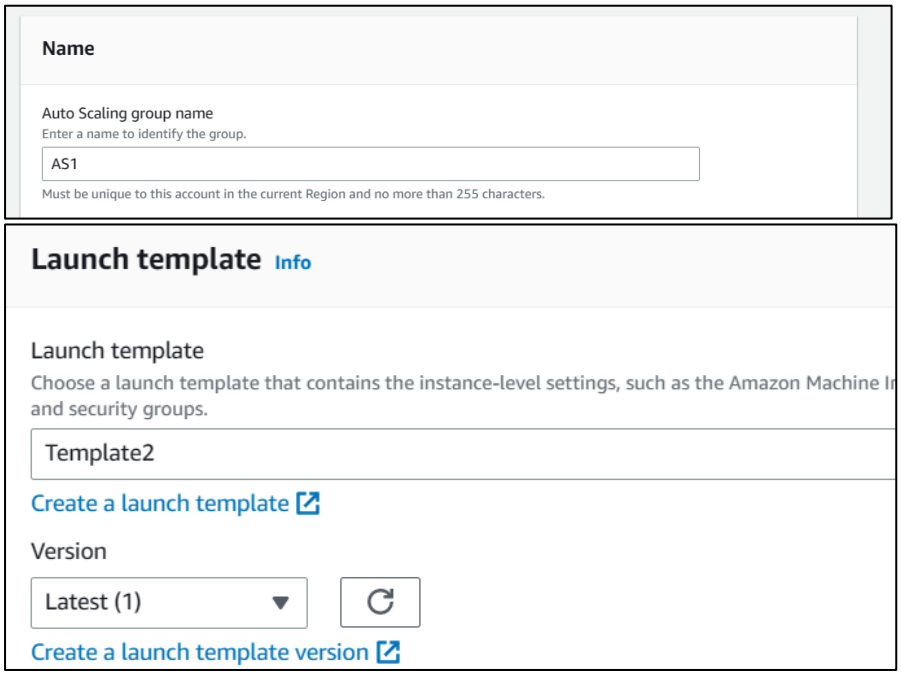


   d) Go to **Advanced details** and in user data write the required data/commands.
      - `#!/bin/bash`
      - `apt-get update`
      - `apt-get install -y nginx`
      - `systemctl start nginx`
      - `systemctl enable nginx`
      - `apt-get install -y git`
      - `curl -sL https://deb.nodesource.com/setup_18.x | sudo -E bash -`
      - `apt-get install -y nodejs`
      - `git clone https://github.com/Sayan-K-Dutta/Assign8-Repo2.git`
      - `cd Assign8-Repo2`
      - `npm install`
      - `node index.js`

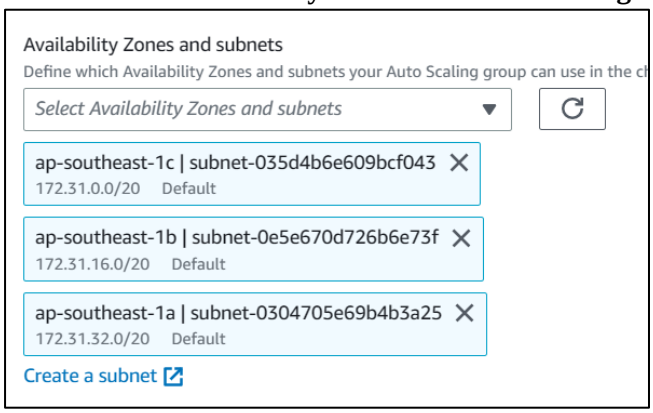
   (Before creating the Template make the required GitHub repo public).
   e) Create the template.



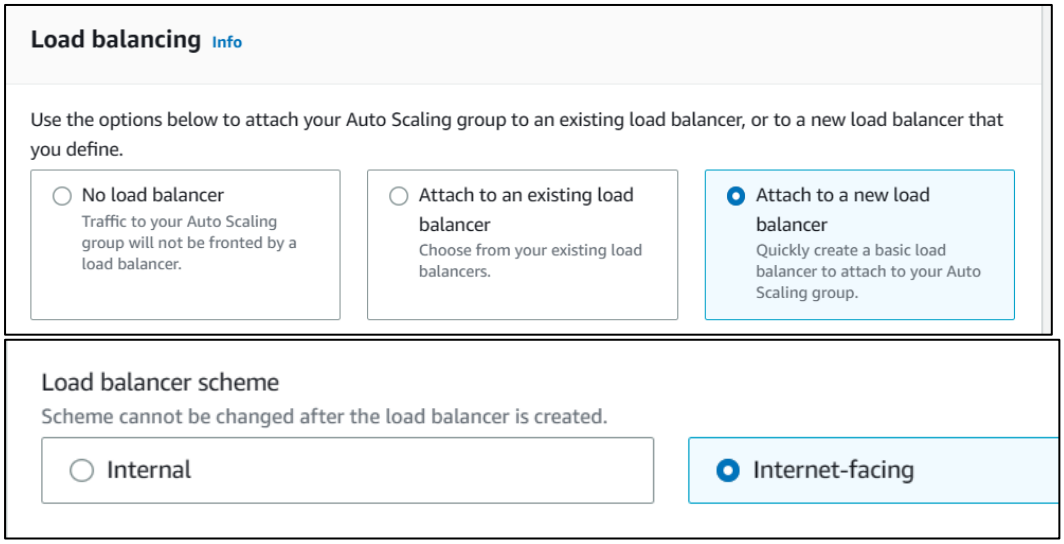
2. Creating Auto Scaling Group
   a) In the left panel, under **Auto Scaling**, go to **Auto Scaling Groups**. Click on **Create Auto Scaling group**.
   b) Give **Name** of the auto scaling group, select the launch template created and select version as **Latest(1).** Click on **Next**.



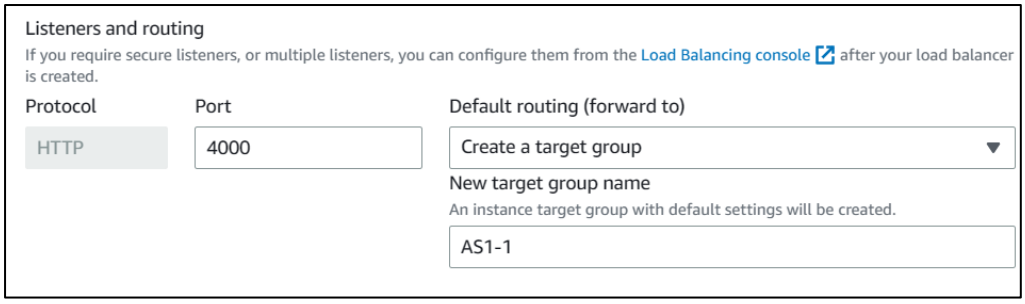
   c) Choose all the **Availability zones and subnets and go to next page.**



   d) Select **Attach to a new load balancer**. Choose Load balancer scheme as **Internet-facing**.

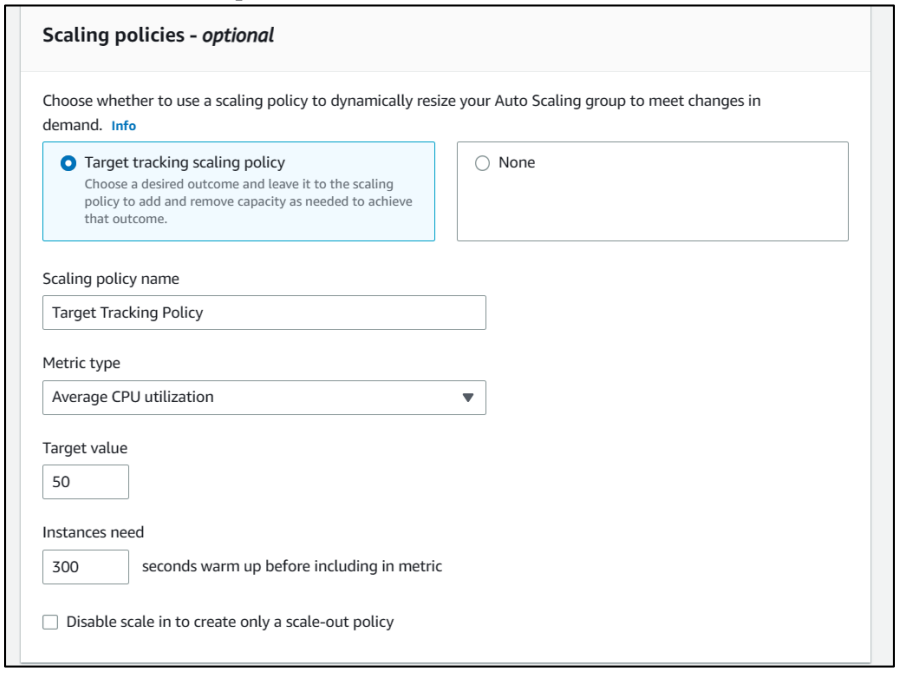


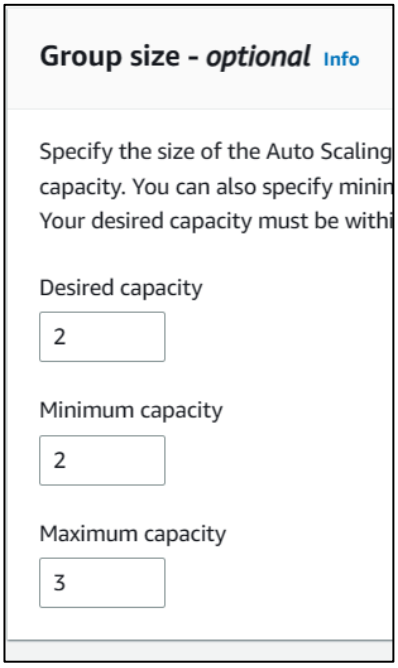
   e) In **Listeners and routing** give **Port** as **4000** and for **Default routing** select **Create a target group** and click the target group showing. Click on **Next**.



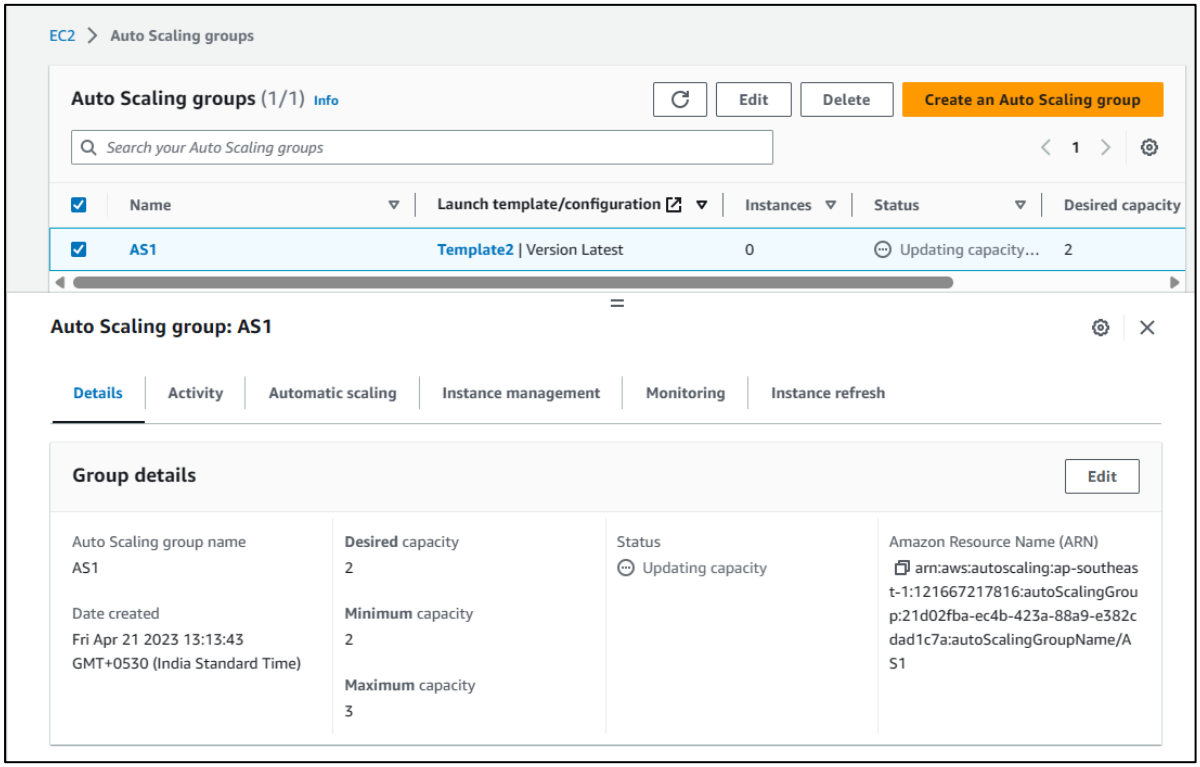
f) In Group size give the **Desired**, **Minimum** and **Maximum** capacity as **2**,**2** and **3** respectively.

g) In **Scaling policies** select **Target tracking scaling policy**. Select **Target value** as **50** and set Instances need **300** seconds warmup.

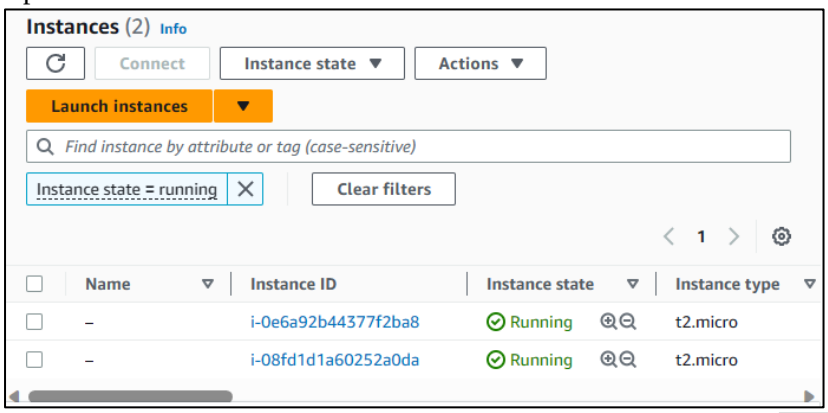




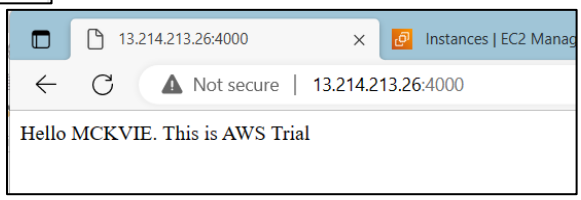
h) Go clicking on **Next** and create auto scaling group. Auto scaling group created.



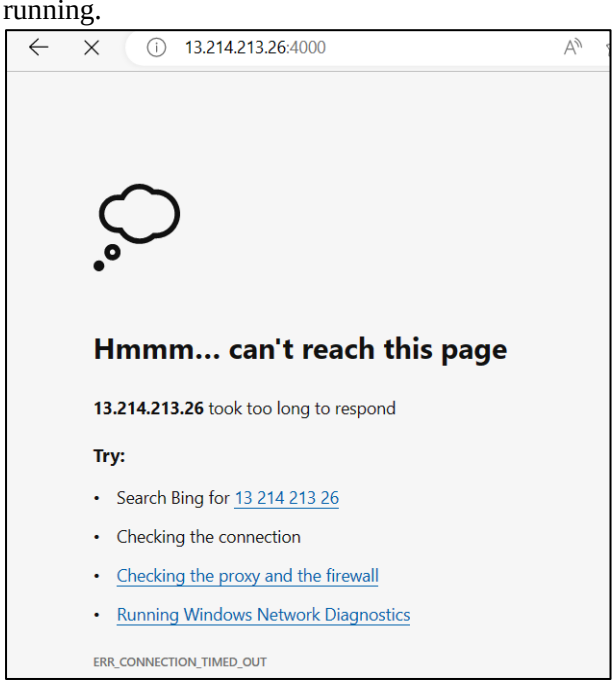
3. Open the Instances and we will see two instances created.



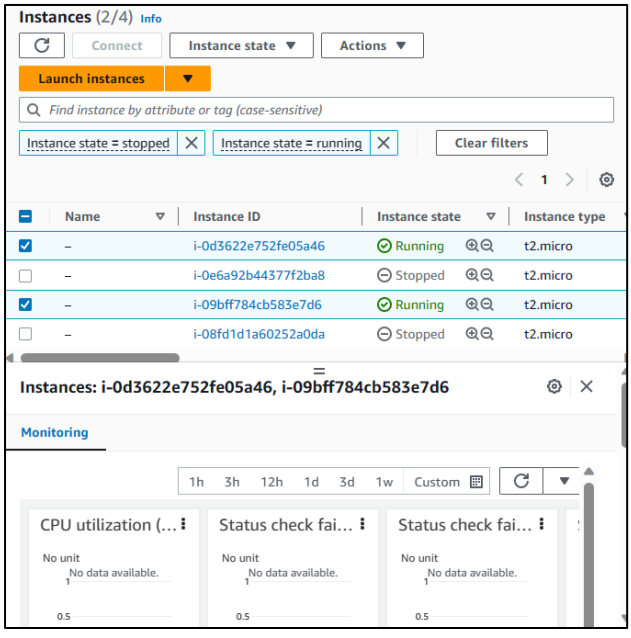
4. Copy the public IPv4 of an instance and open in web browser to see if it is working properly. Give port as 4000 to see the project running.

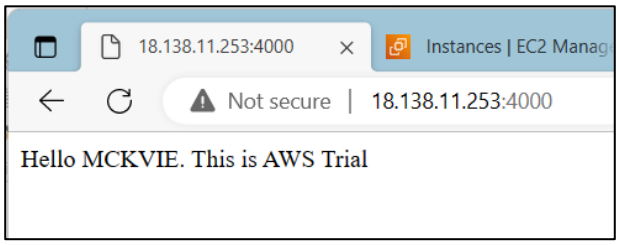


5. Now select both the instances and stop them (crashing the server) and refresh the website. We will no more see the project running.

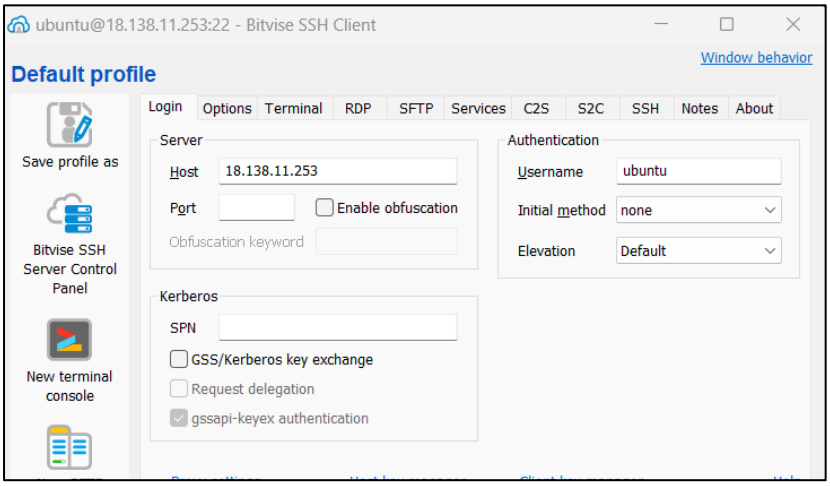


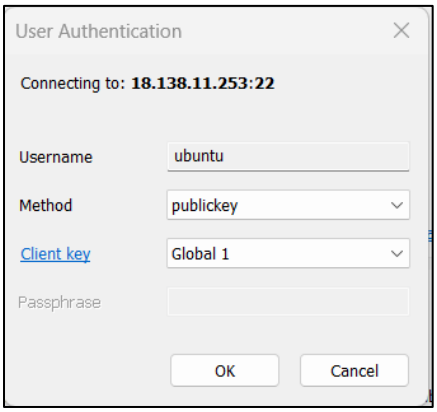
6. New instance will be created automatically. Open the project with the help of this instances' public IP address and port 4000. The project will be running.



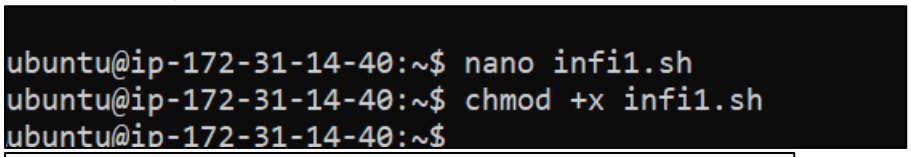


7. Open **Bitvise SSH Client** and connect a running instance.

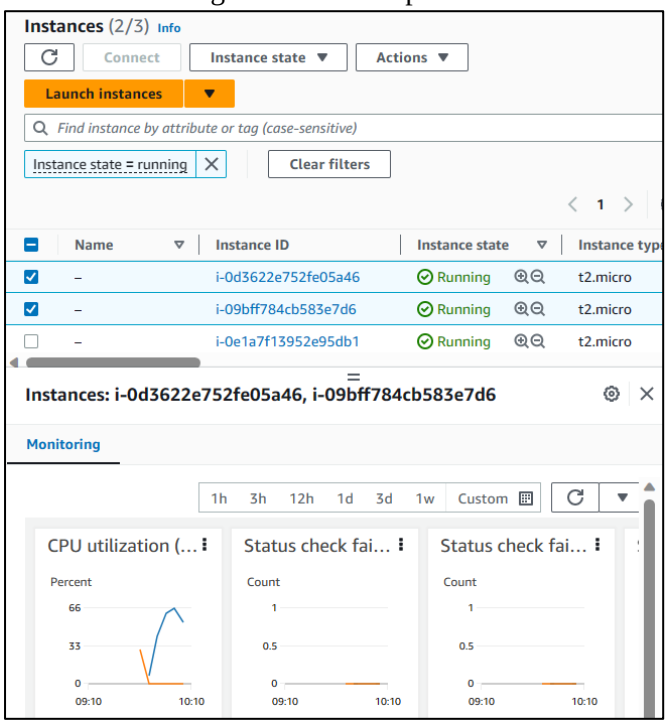




8. Login and open a new terminal console. Write an infinite loop code save and give necessary permission (`chmod +x infi1.sh`) and execute it.



9. The server running the infinite loop will be overloaded and new instance will be created.



10. Selecting the 3 running instances, go to **Monitoring** section and **enlarge** the **CPU Utilization** section. Go to **Custom** and select the **local time zone**. We can see new instance is created when one of the servers crossed 50% and the performance of these instances.