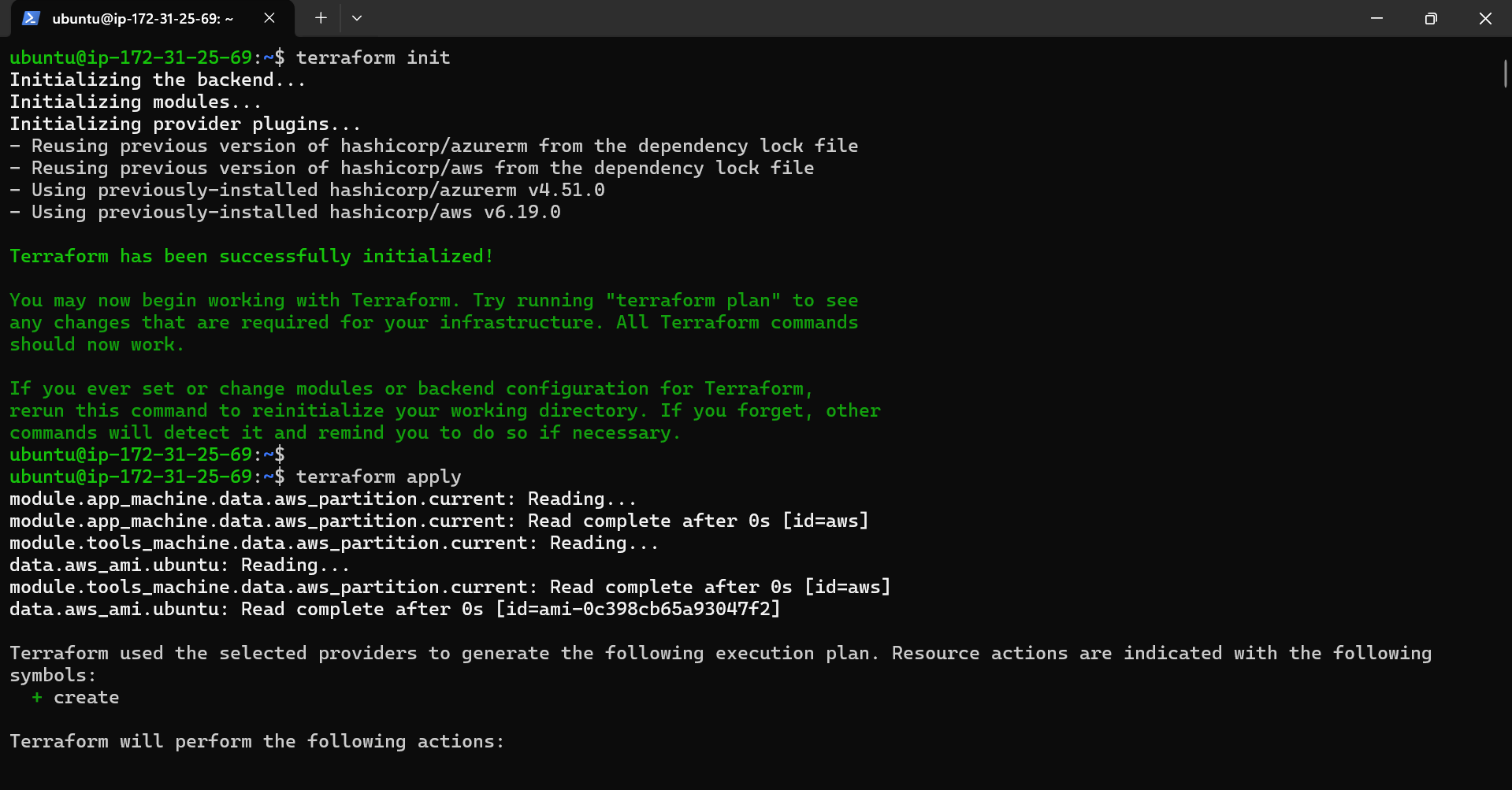
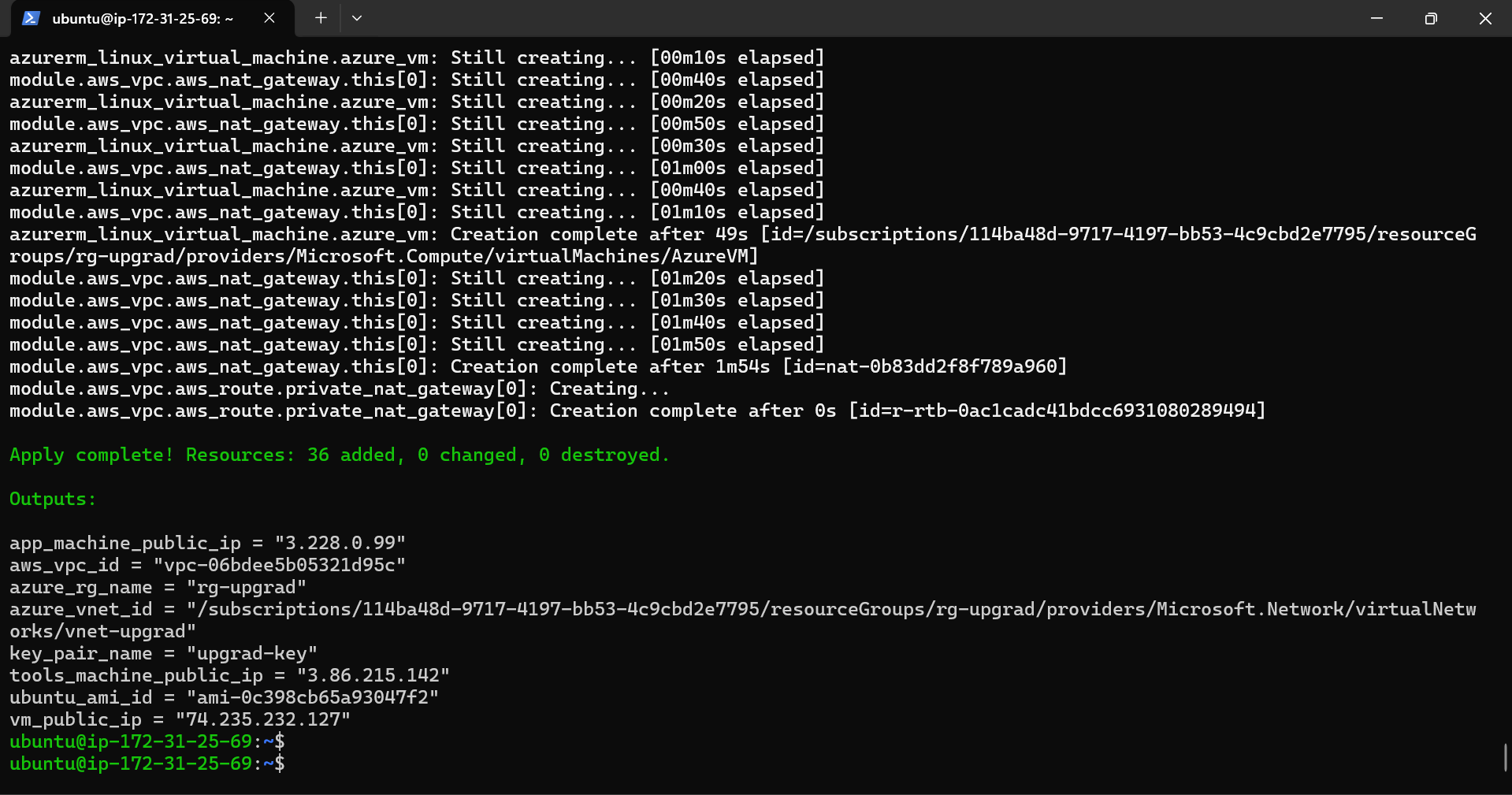
**Capstone Project**

**Task 1: Infrastructure Provisioning**

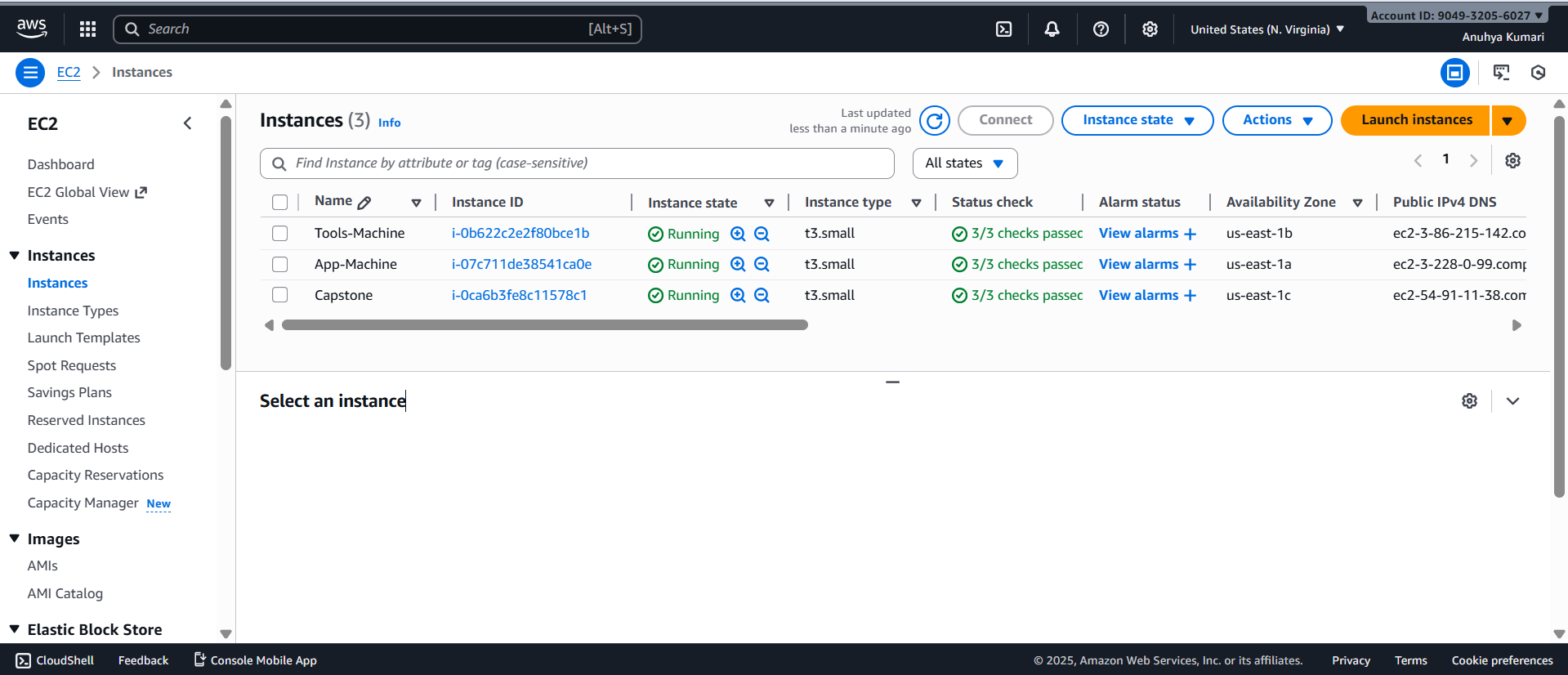
* After using terraform init command



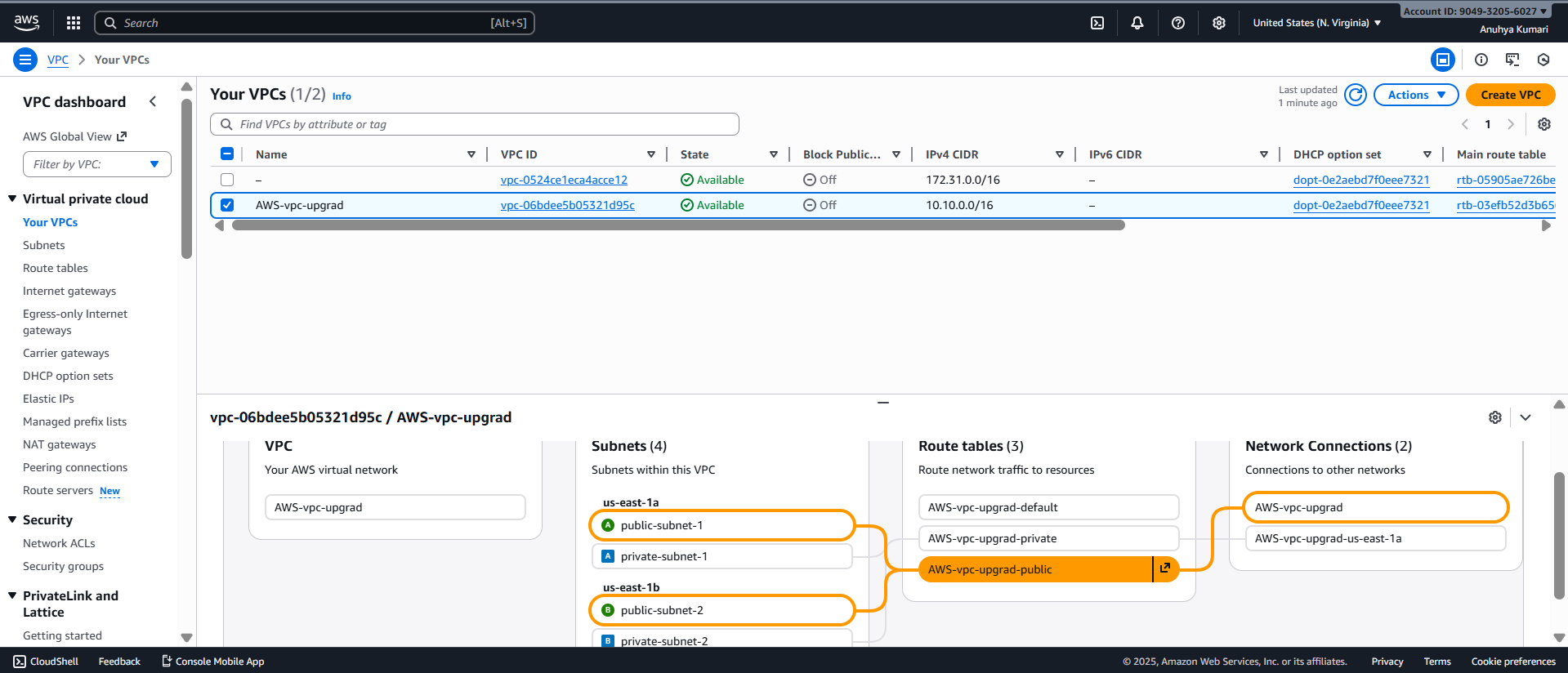
* After using Terraform apply command



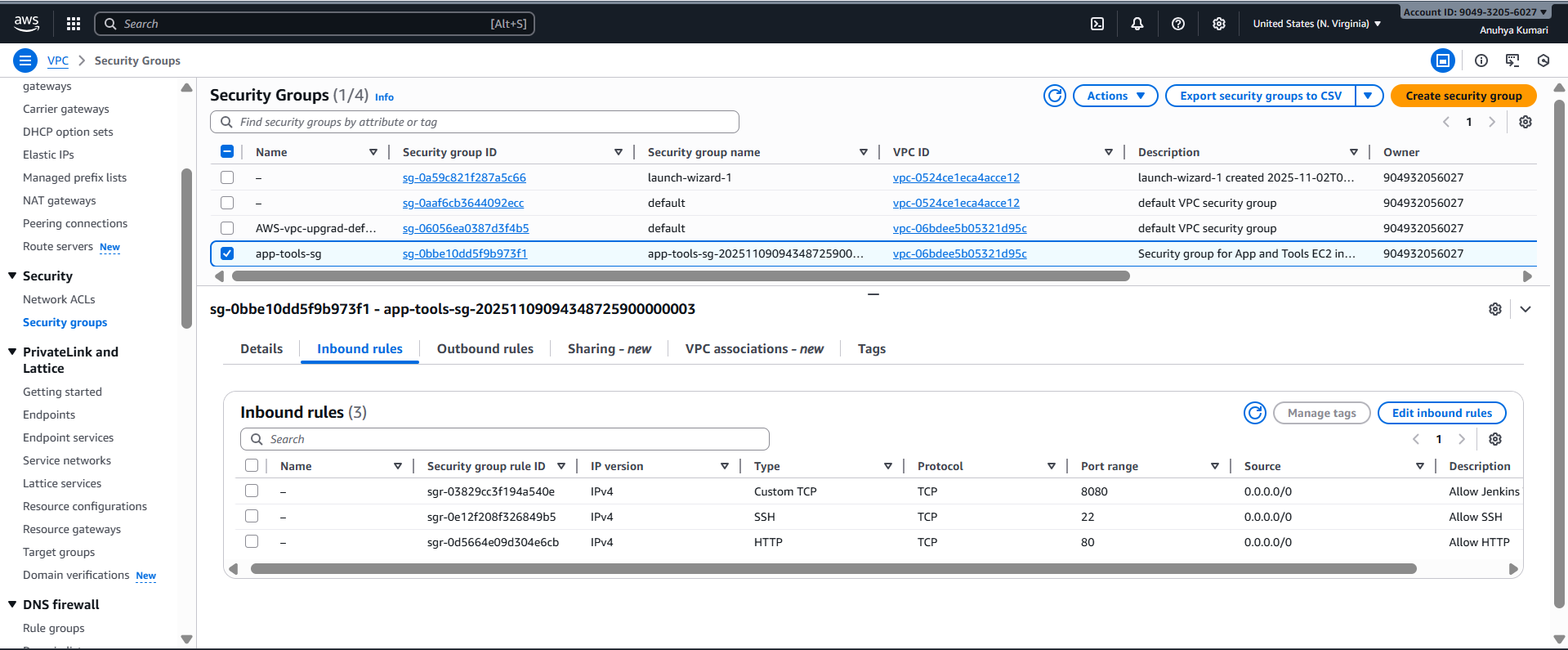
* Resources are created in AWS & Azure
  + 2 EC2 instances named App-machine and Tools-machine are created in AWS



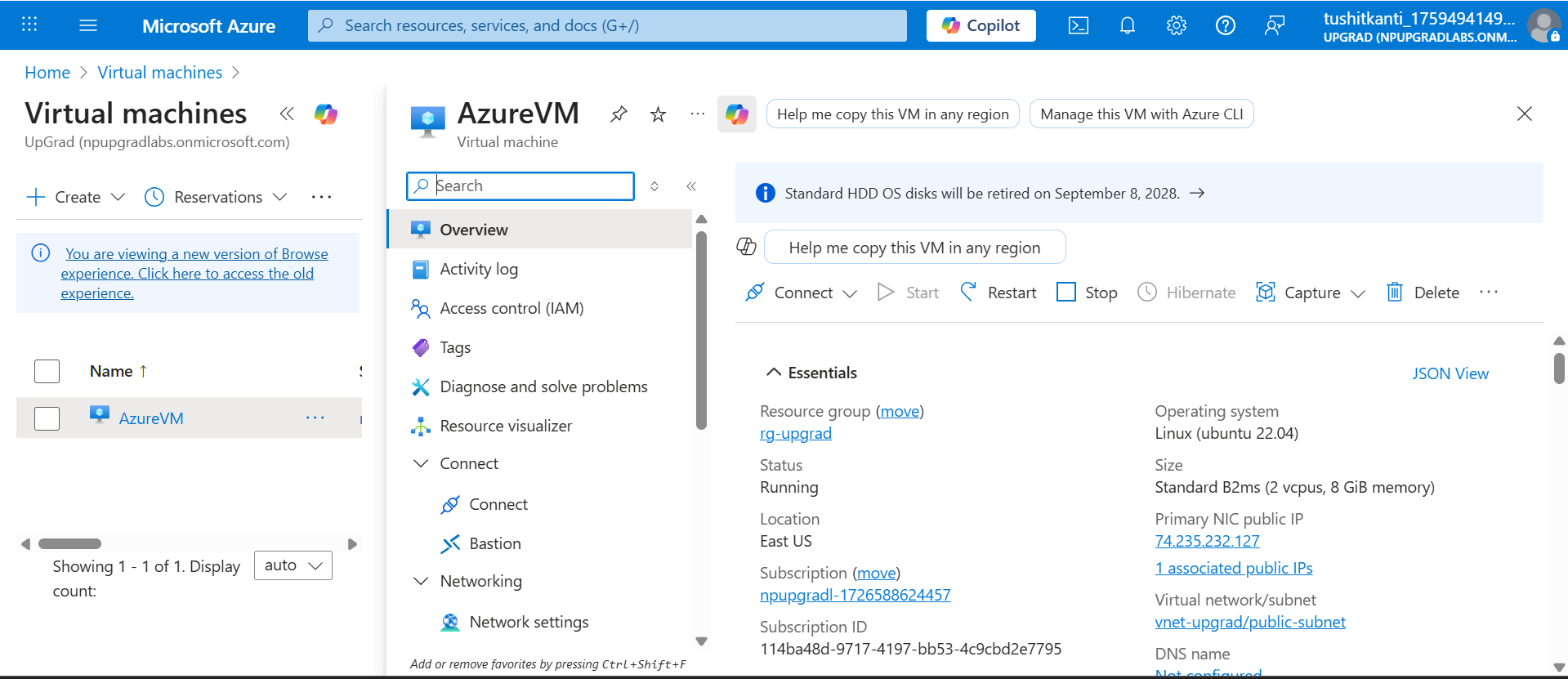
* + VPC named AWS-vpc-upgrad is created in AWS



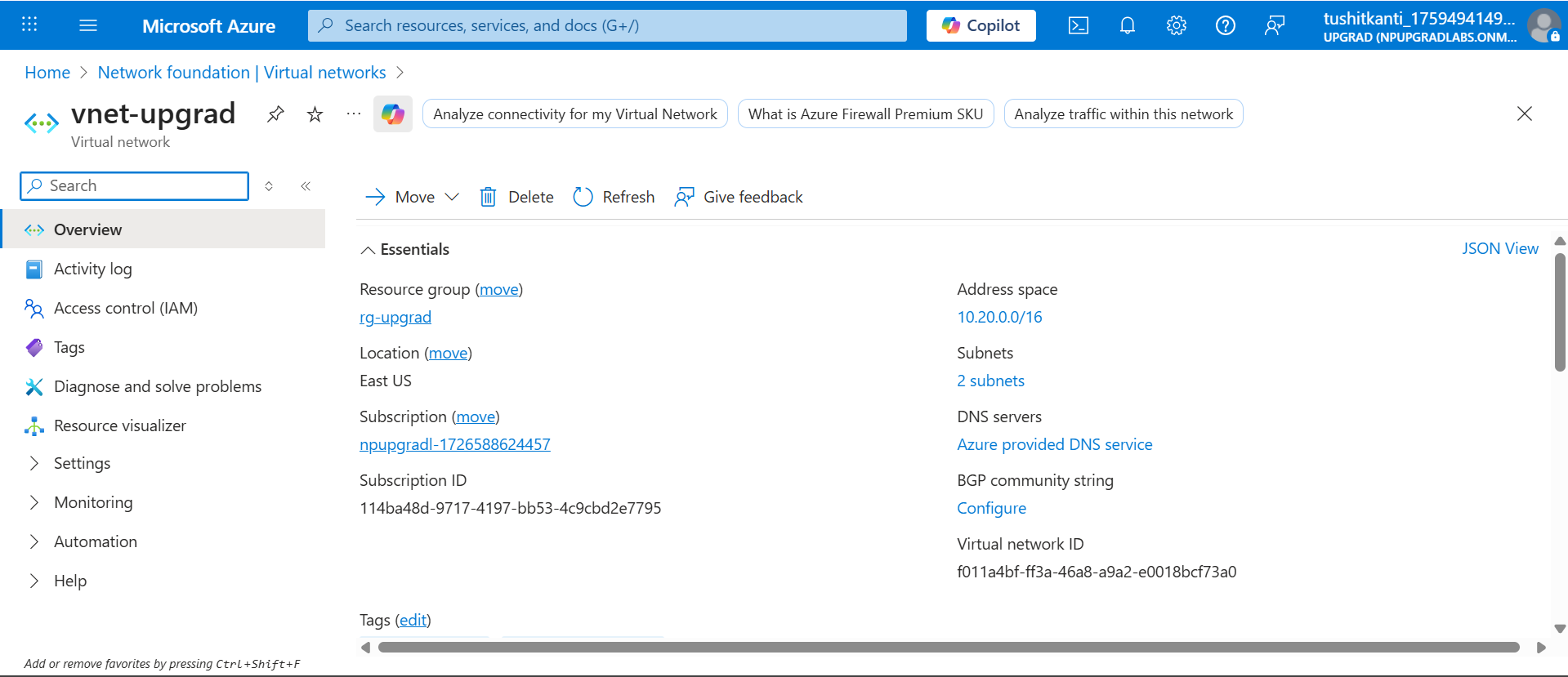
* + Created security group named app-tools-sg, and opened ports 22, 80, and 8080



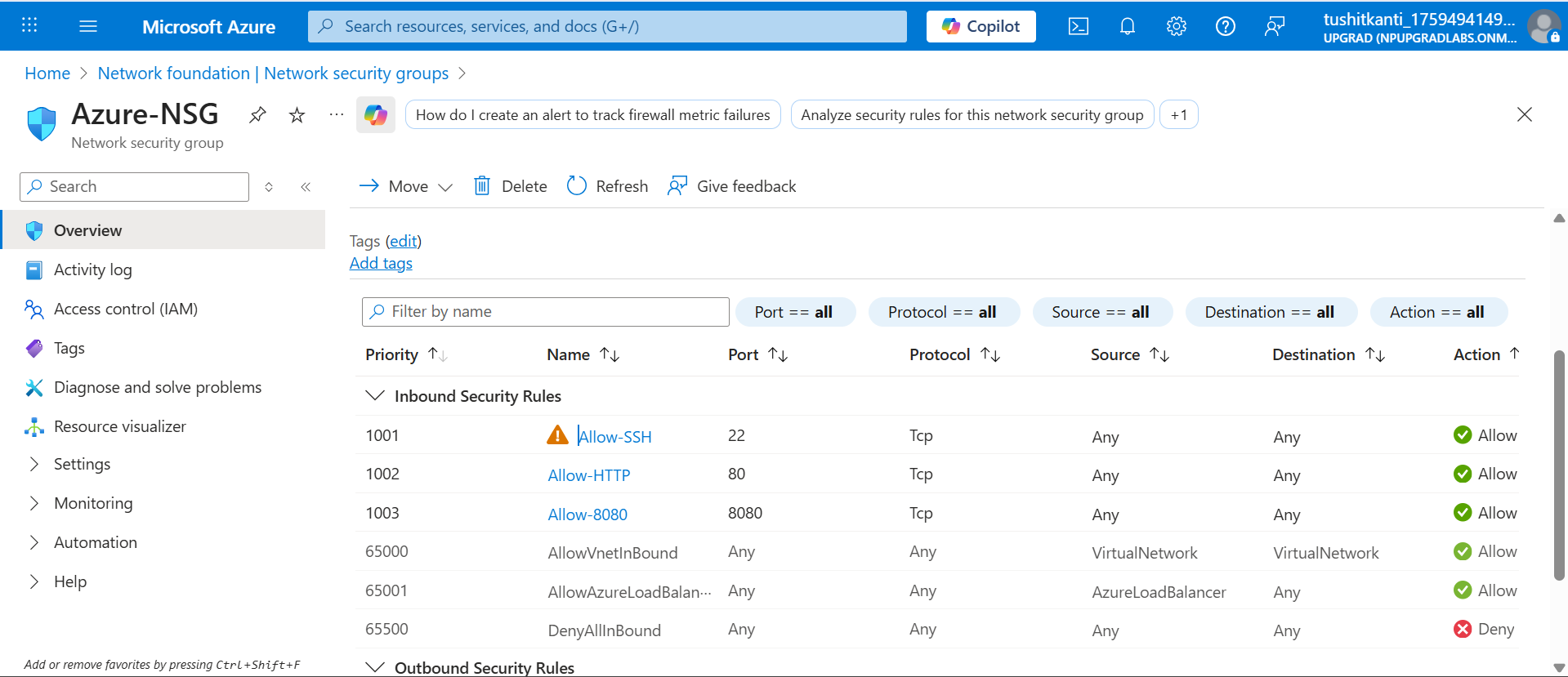
* + 1 Virtual Machine (VM) named AzureVM is created in Azure



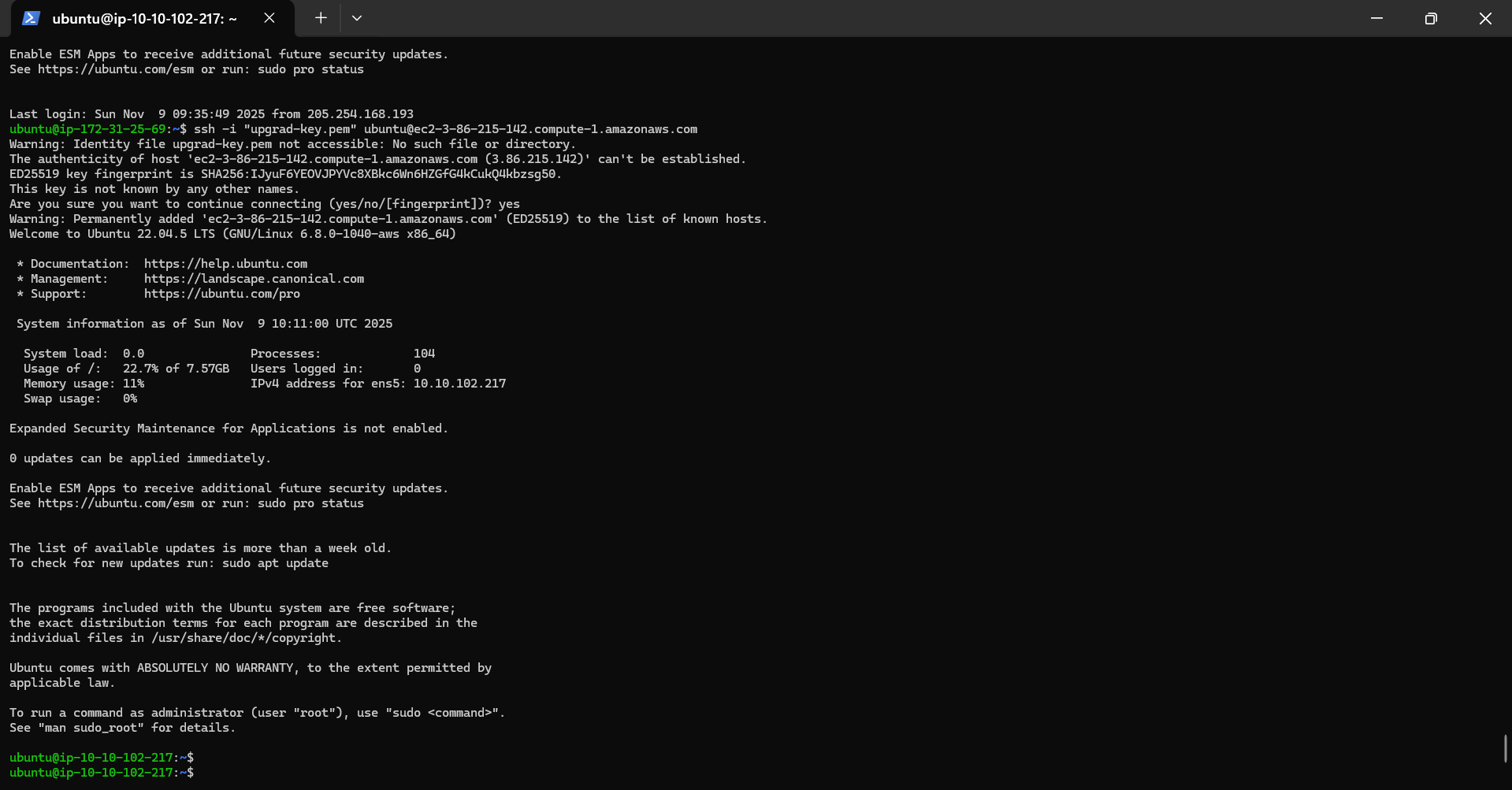
* + VNet named vnet-upgrad is created in Azure with 2 subnets (1 public and 1 private)



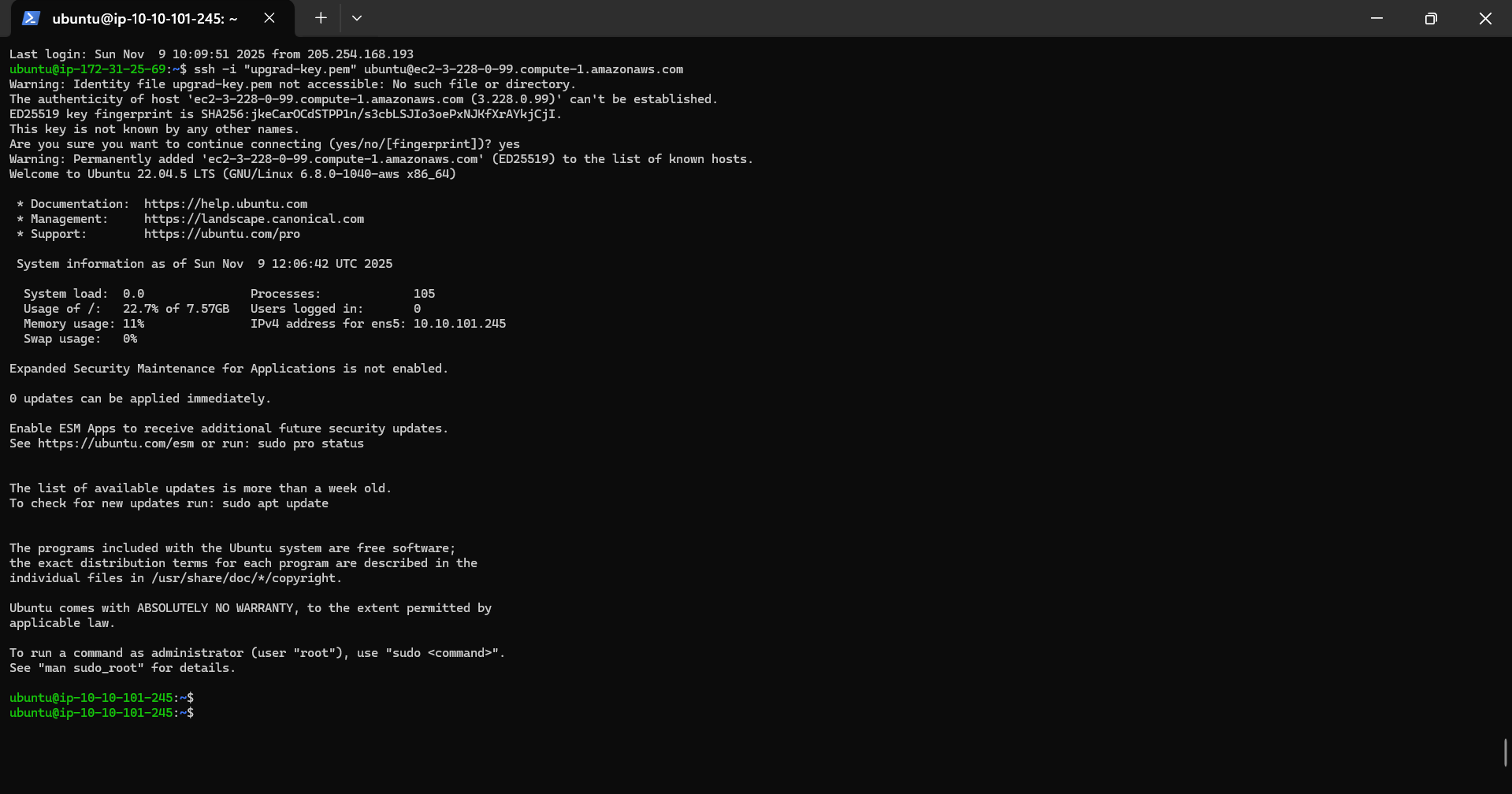
* + Created Network Security Group (NSG) named Azure-NSG, and opened ports 22, 80, and 8080



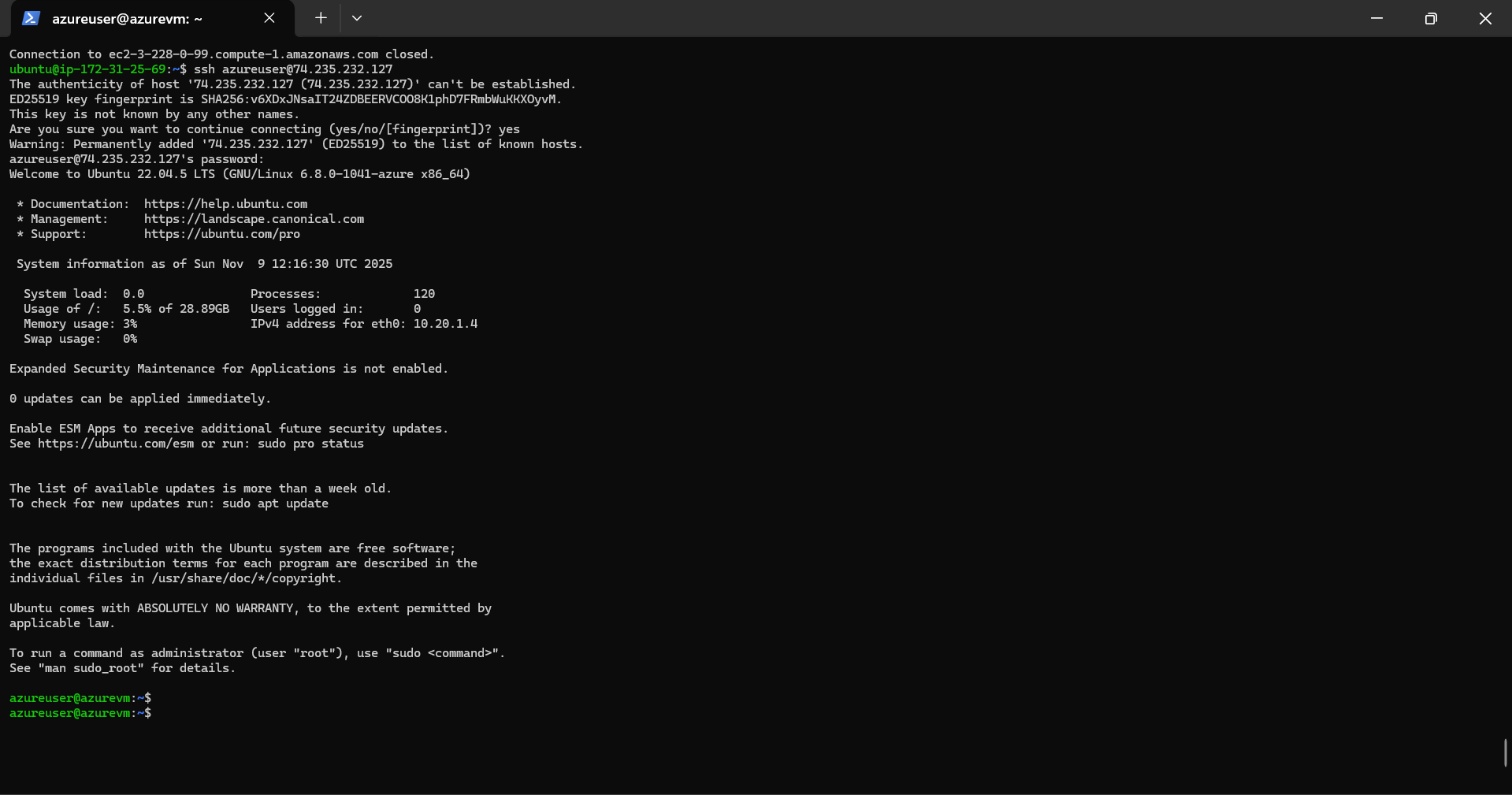
* SSH Connectivity is working for Tools-machine EC2 instance from my local machine.



* SSH Connectivity is working for App-machine EC2 instance from my local machine.

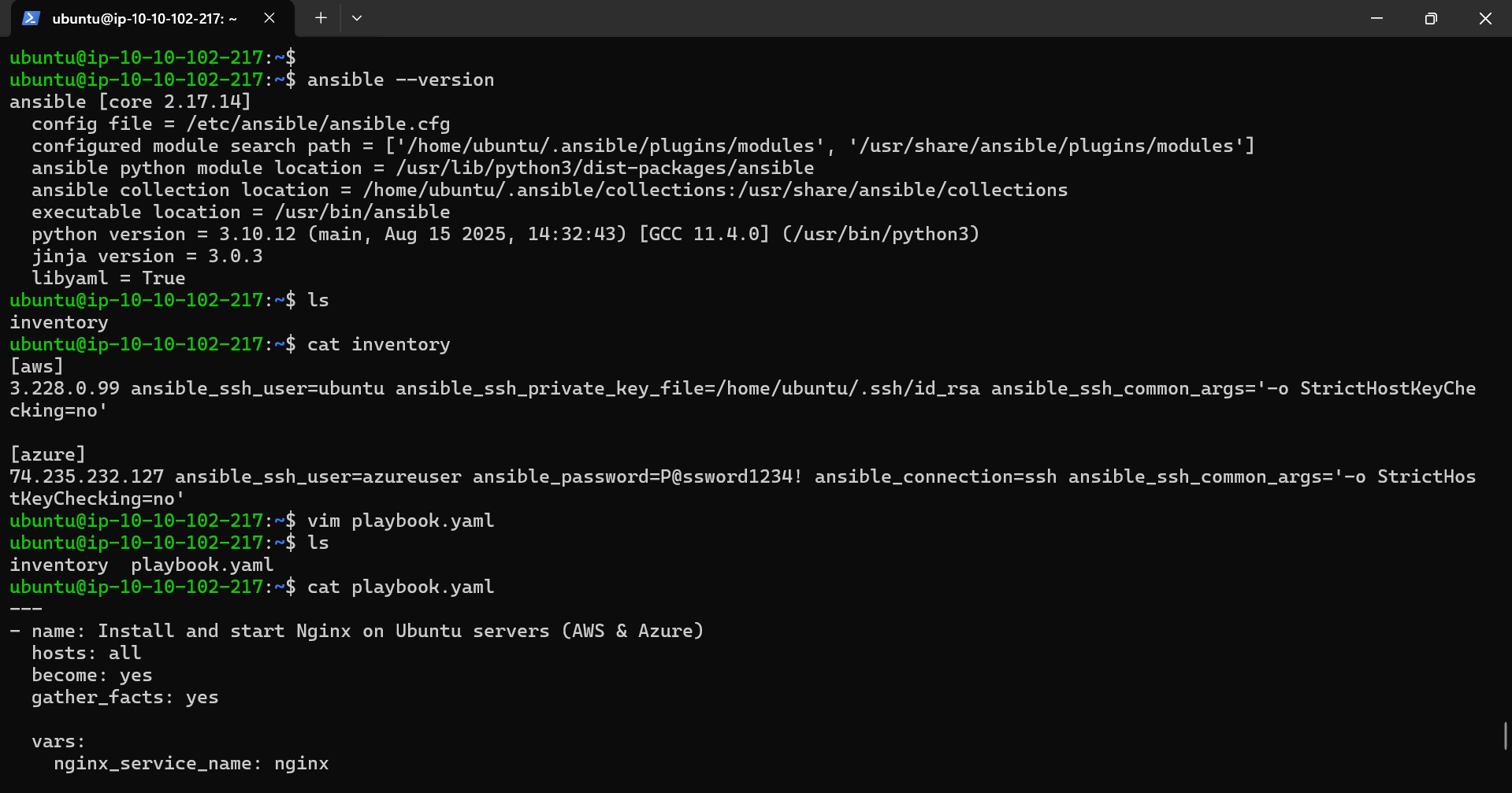


* SSH Connectivity is working for AzureVM instance from my local machine.

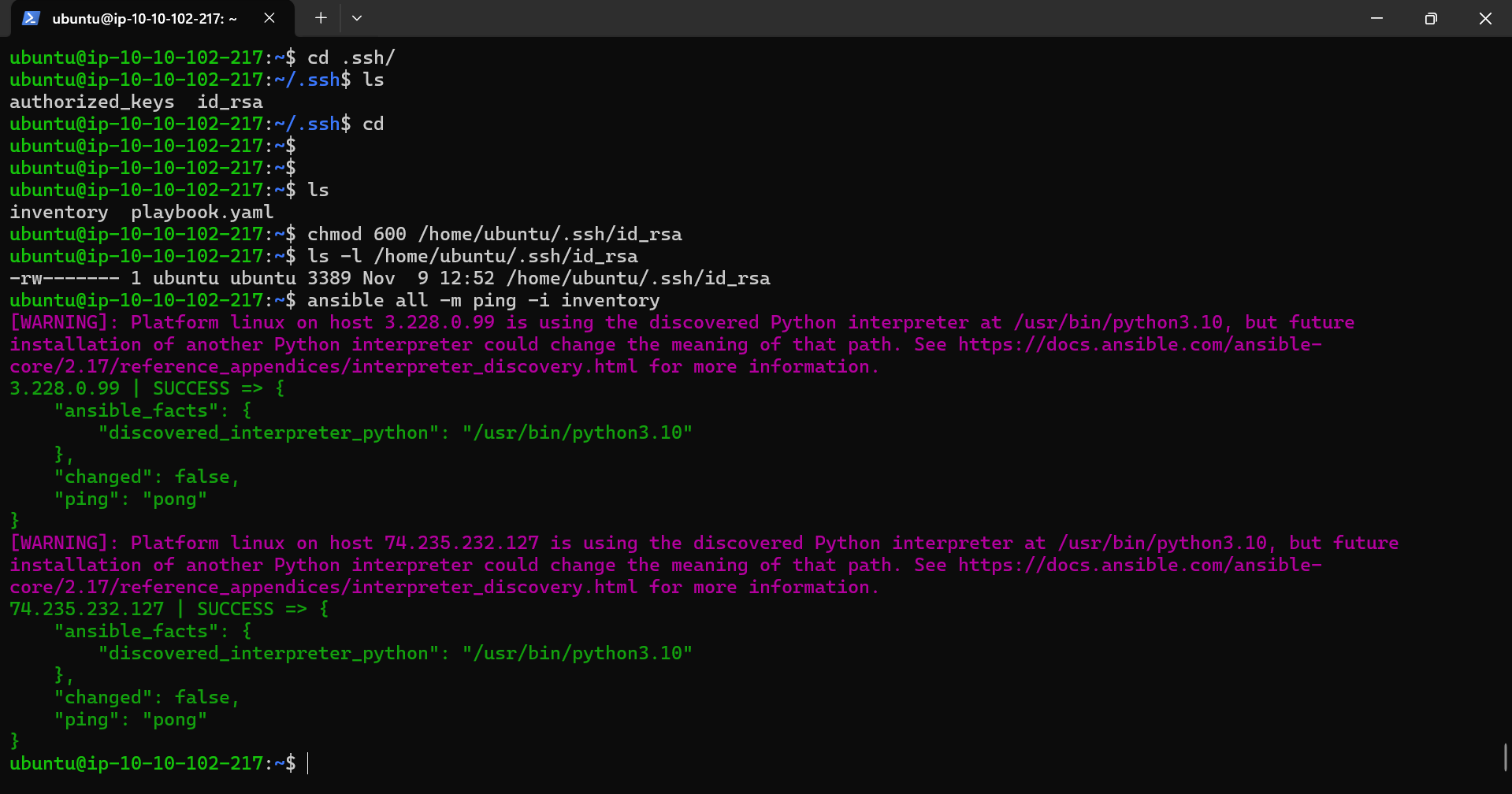


**Task 2: Configuration Management**

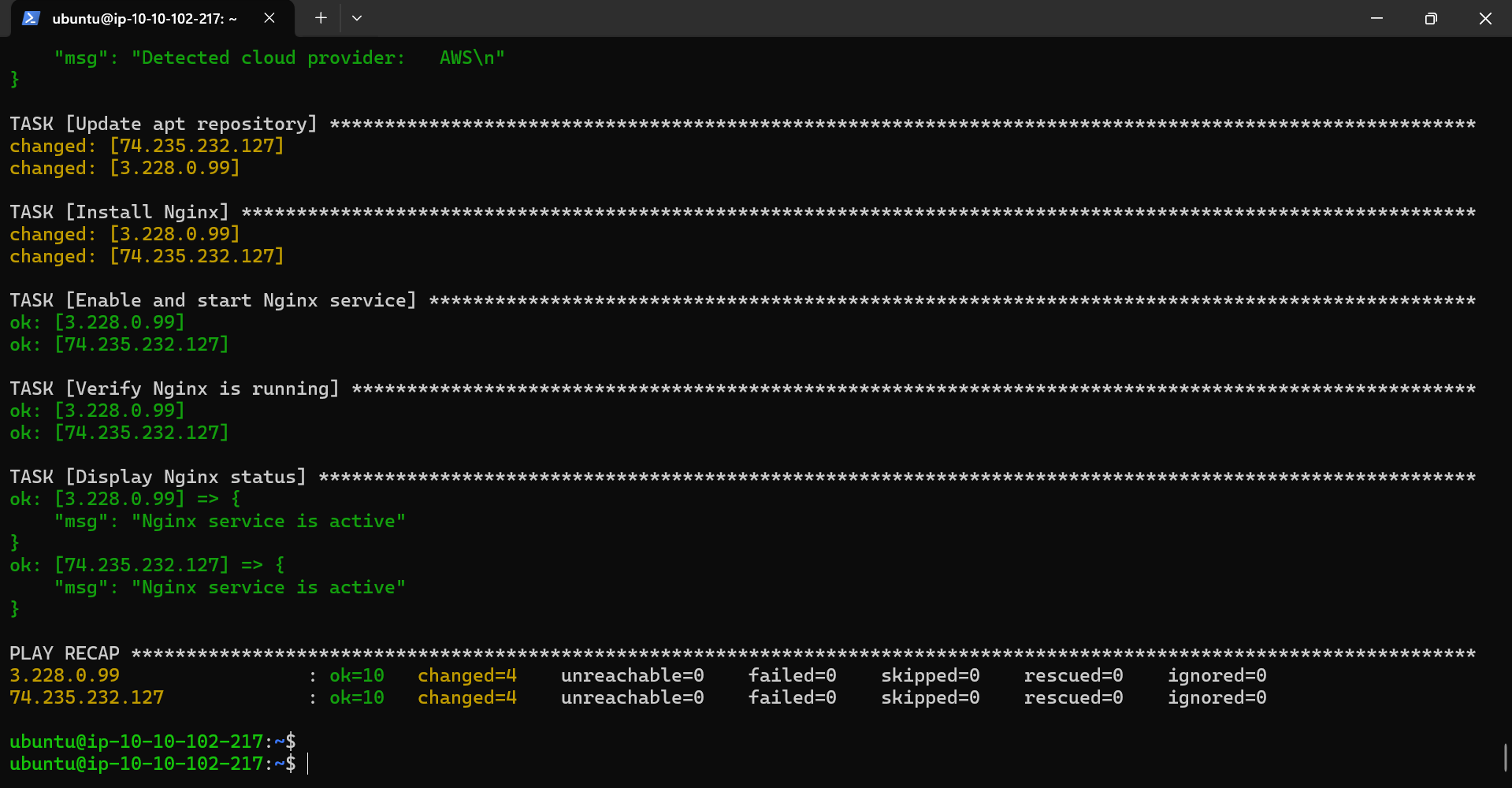
* Installed Ansible on the Tools-Machine (AWS). Also, created inventory and playbook.yaml to install and configure Nginx web server on both App-Machine (AWS) and AzureVM (Azure) machines.



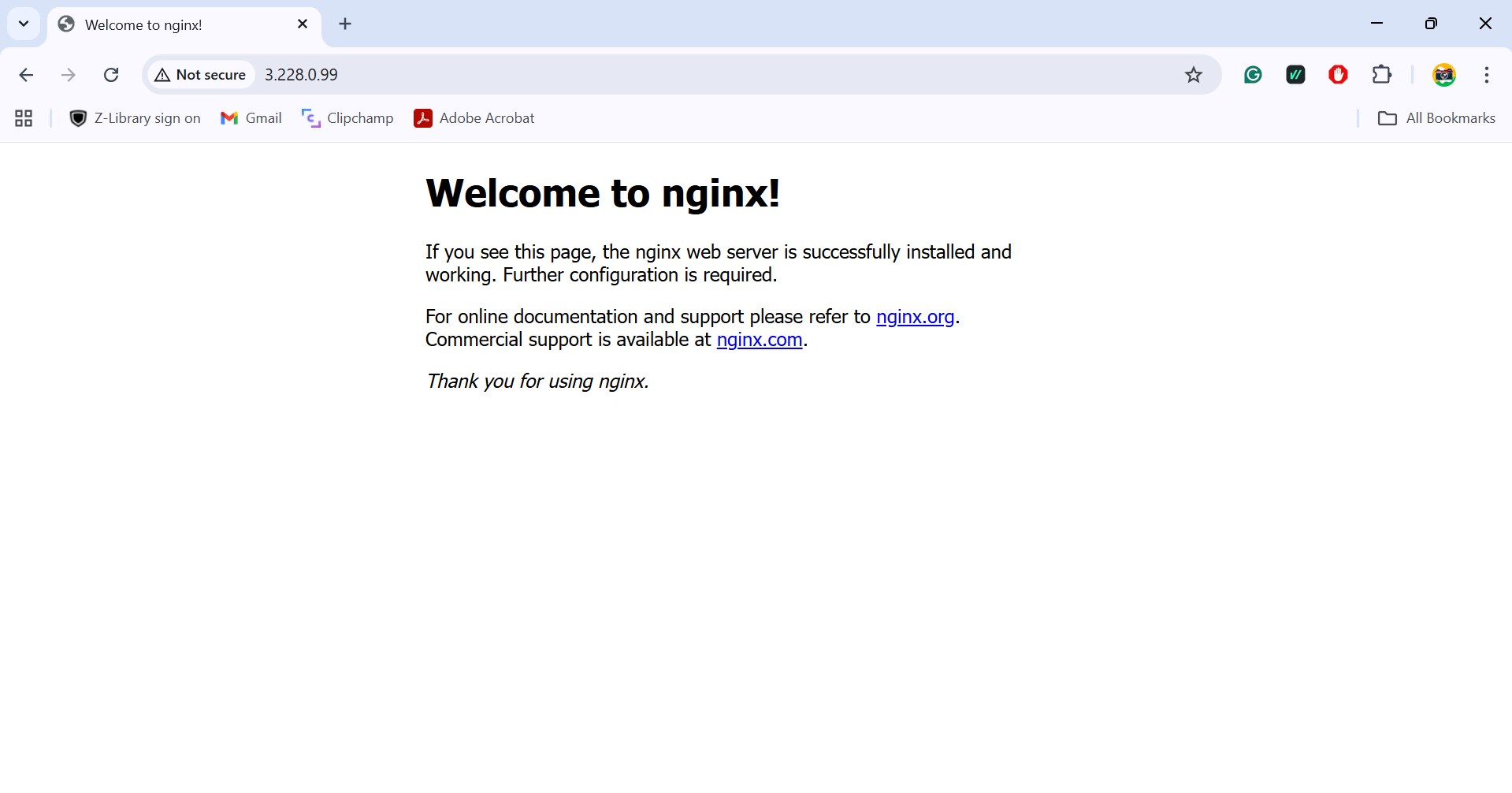
* Copied the private key (id\_rsa) that is created into the tools machine in the location .**ssh/**
* After using the command – **ansible all -m ping -i inventory**



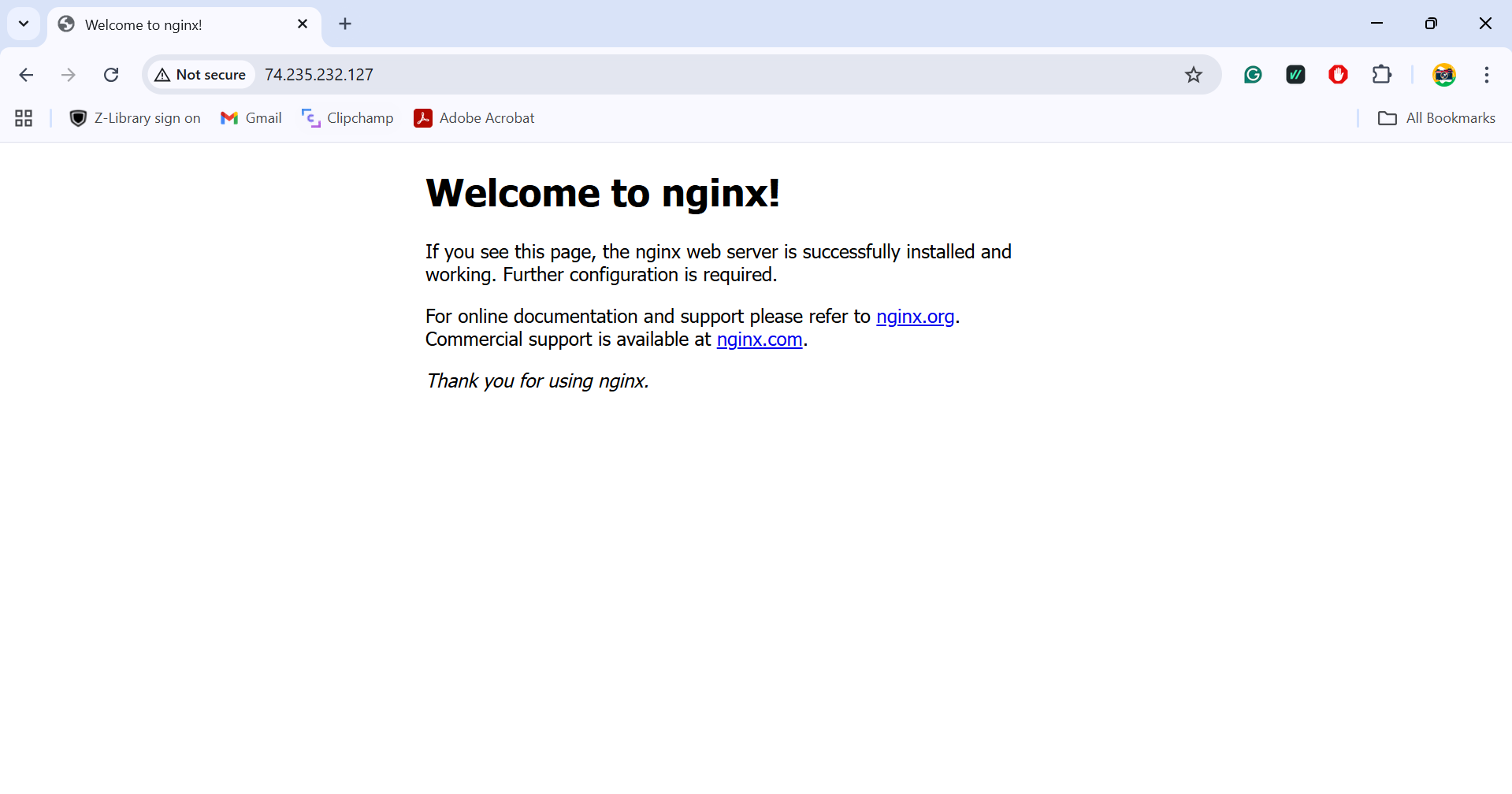
* After using the command - **ansible-playbook playbook.yaml -i inventory**



* Nginx is actively running on both App-Machine (AWS) and AzureVM (Azure) machines, showing the text **Welcome to Nginx**.
  + Nginx is active on App-machine, accessed the application via HTTP (IP:80) – *http://<ec2\_public\_ip>*

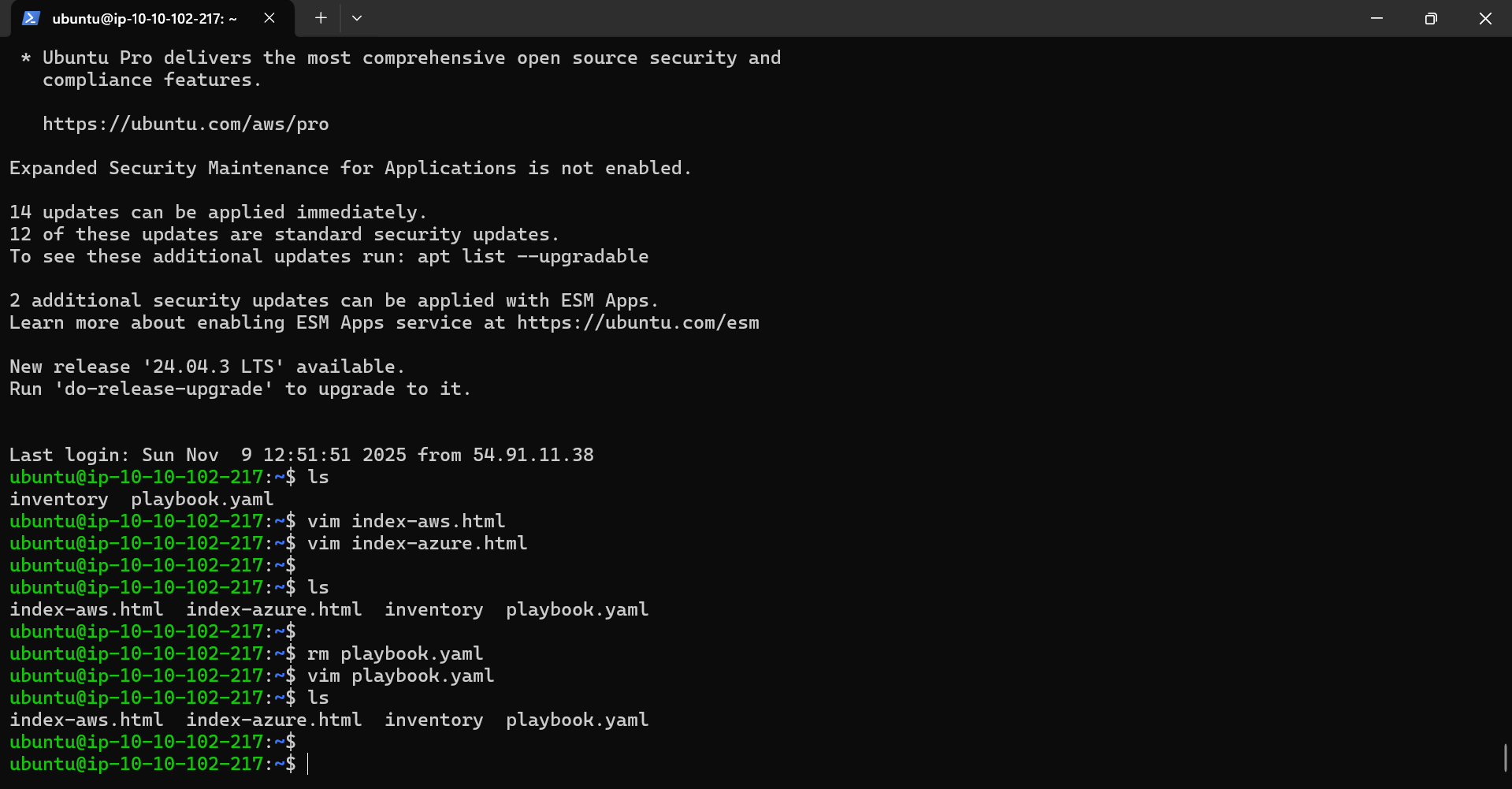


* + Nginx is active on AzureVM, accessed the application via HTTP (IP:80) – *http://<VM\_public\_ip>*

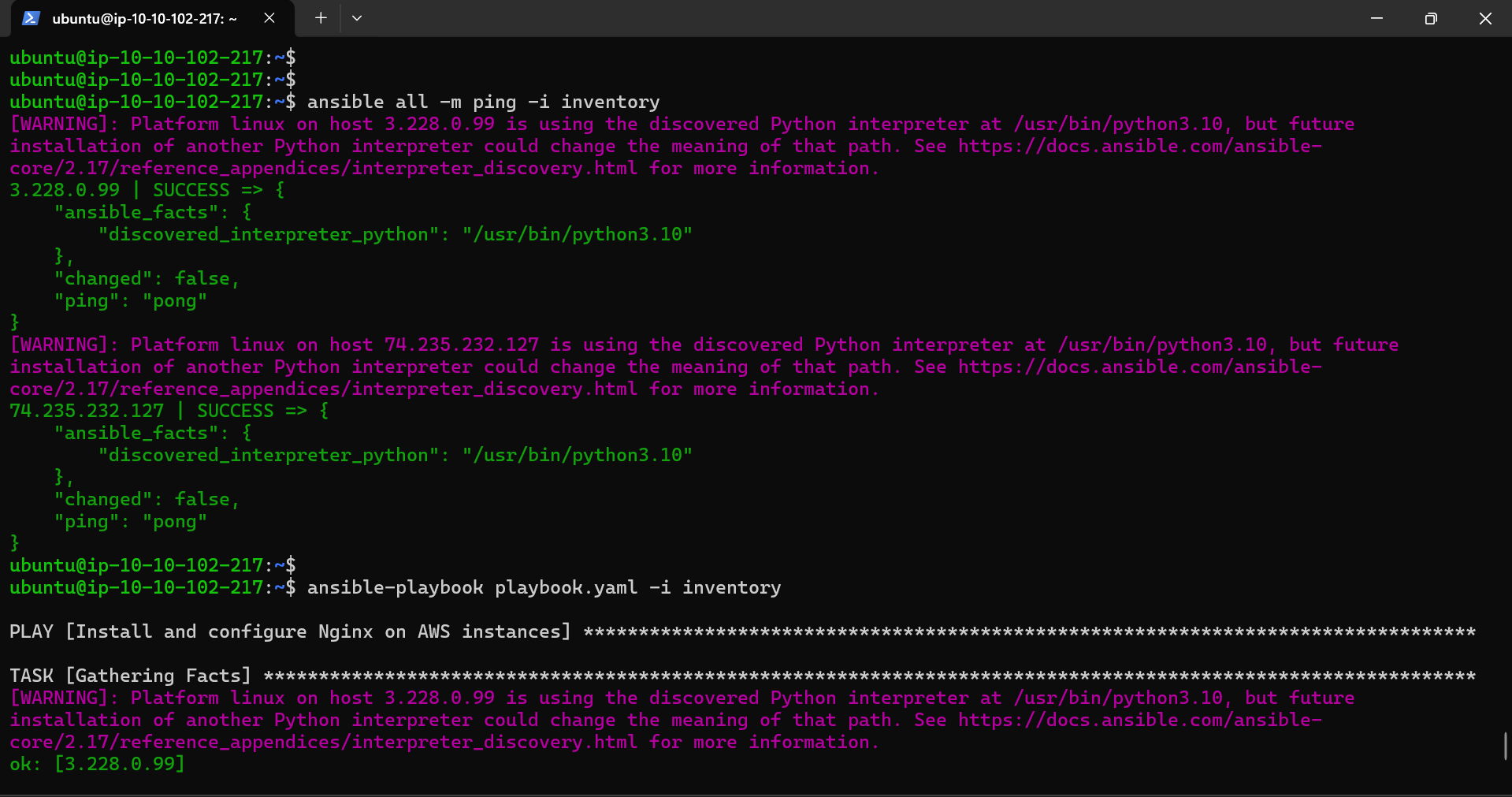


**Task 3: Application Deployment**

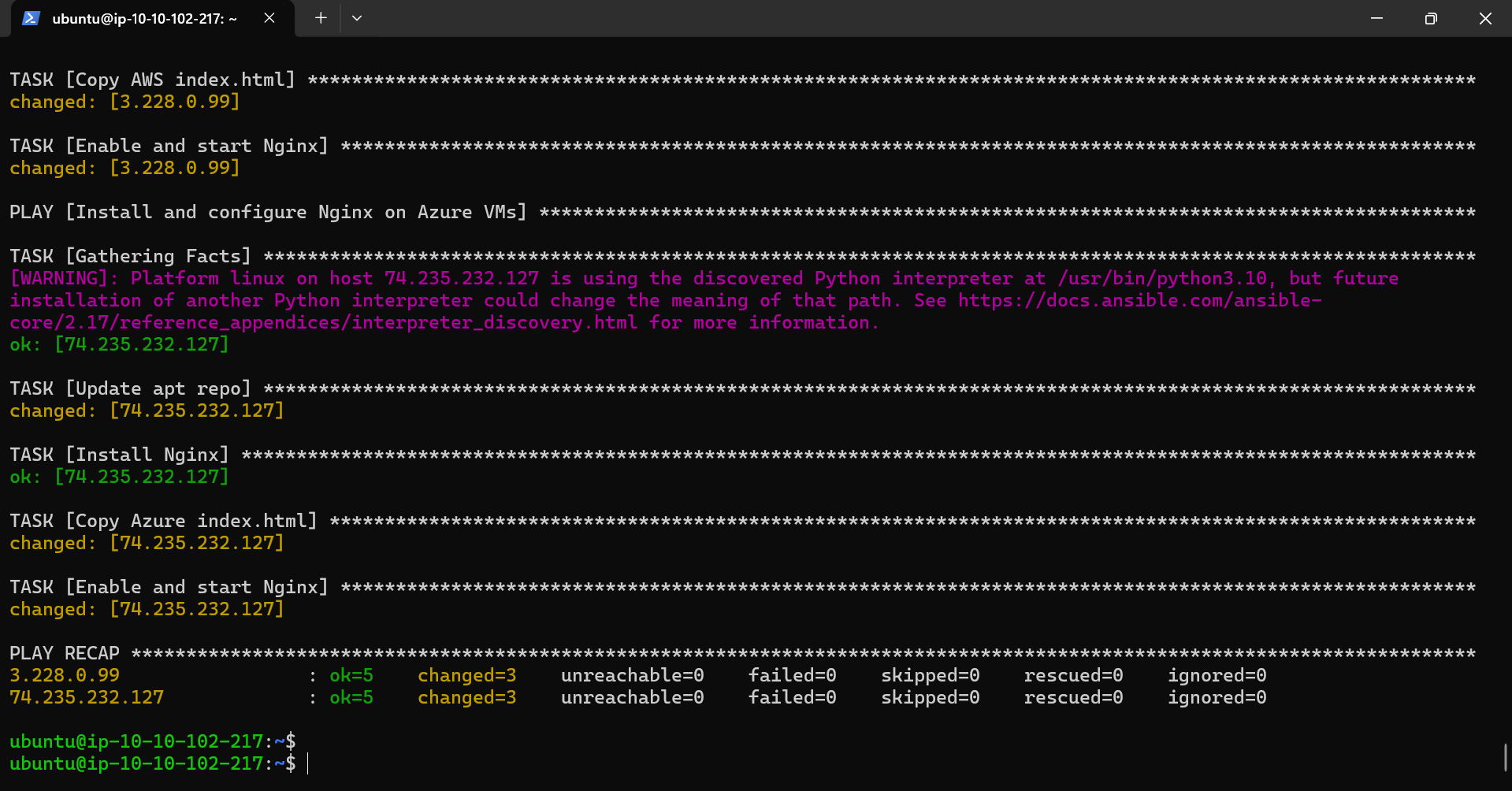
* Copy html file from **/var/www/html/index.nginx-debian.html.**
  + Created two customised index.html files, index-aws.html and index-azure.html as a copy of the default nginx welcome html page, and paste them in the folder where inventory and playbook.yaml are present.
  + Also, updated the playbook.yaml file to serve the new HTML page.
  + Modified the playbook such that index-aws.html and index-azure.html files will get copied to /var/www/html/ on the corresponding machine.



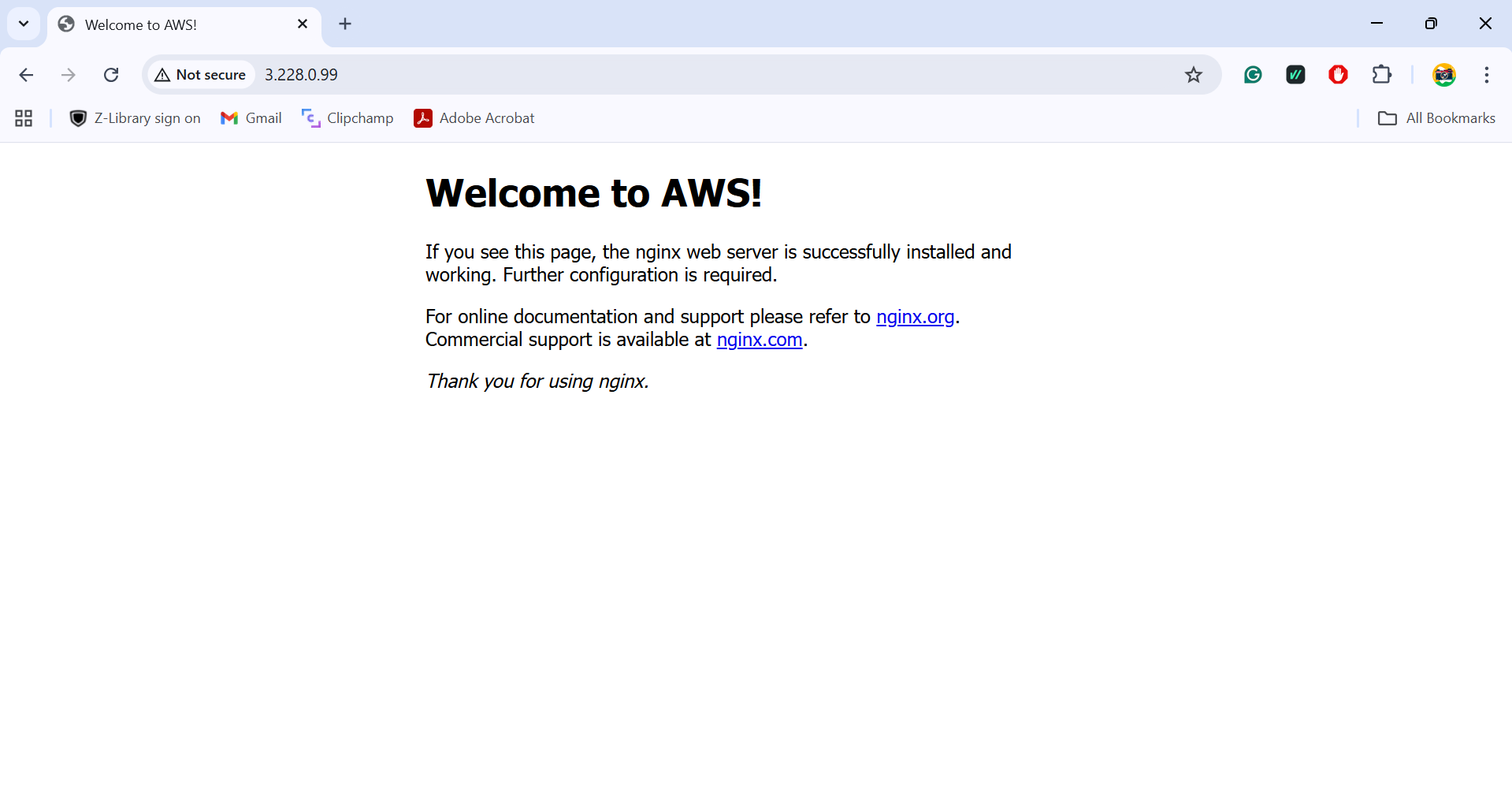
* Running the 2 Ansible commands on Tools Machine.
  + After using the command – **ansible all -m ping -i inventory**



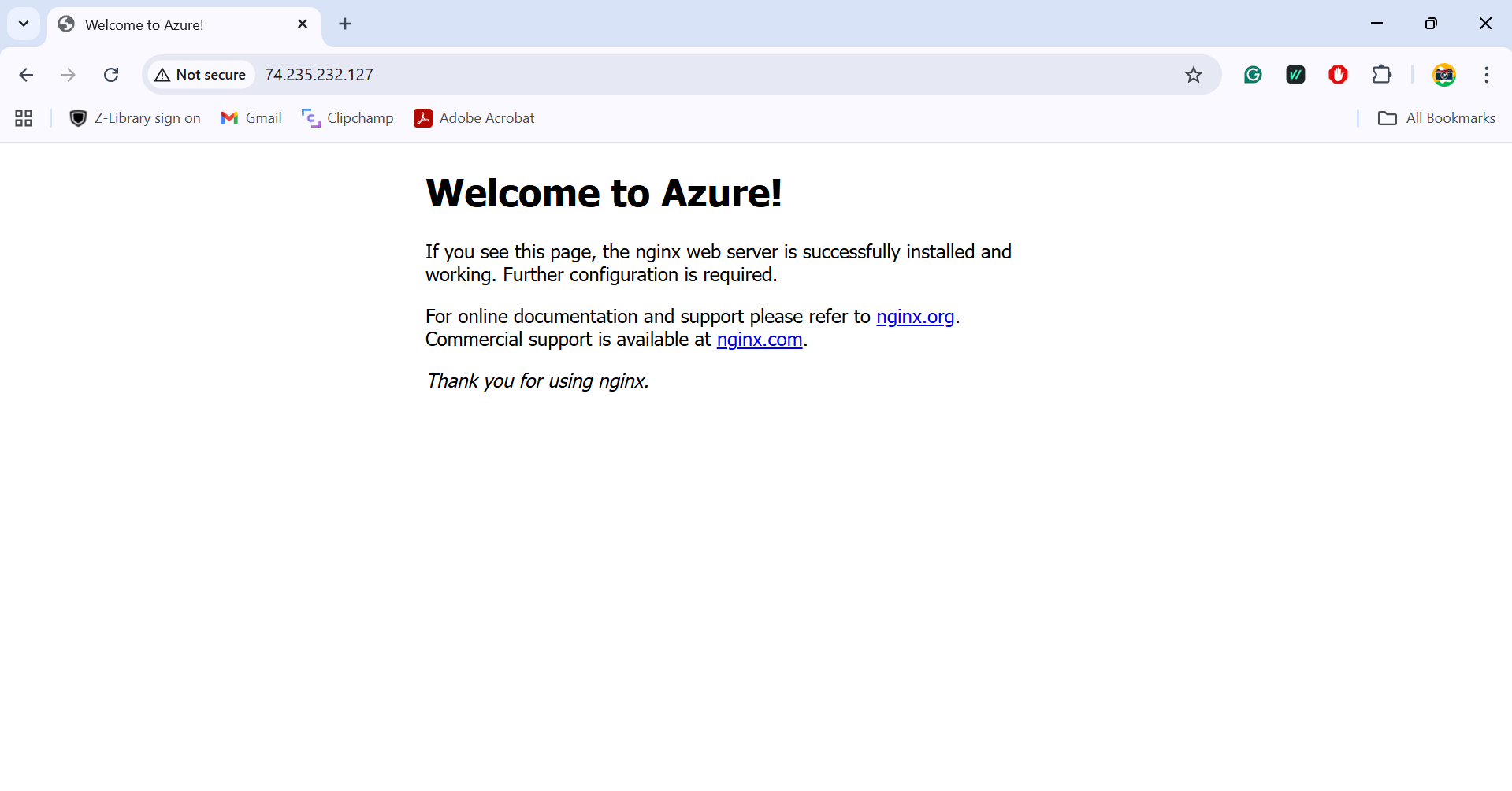
* + After using the command - **ansible-playbook playbook.yaml -i inventory**

****

* Nginx changes are applied and are actively running on both App-Machine (AWS) and AzureVM (Azure) machines, showing the text **Welcome to AWS! and Welcome to Azure!**
  + Nginx is active on App-machine, accessed the application via HTTP (IP:80) – *http://<ec2\_public\_ip>*

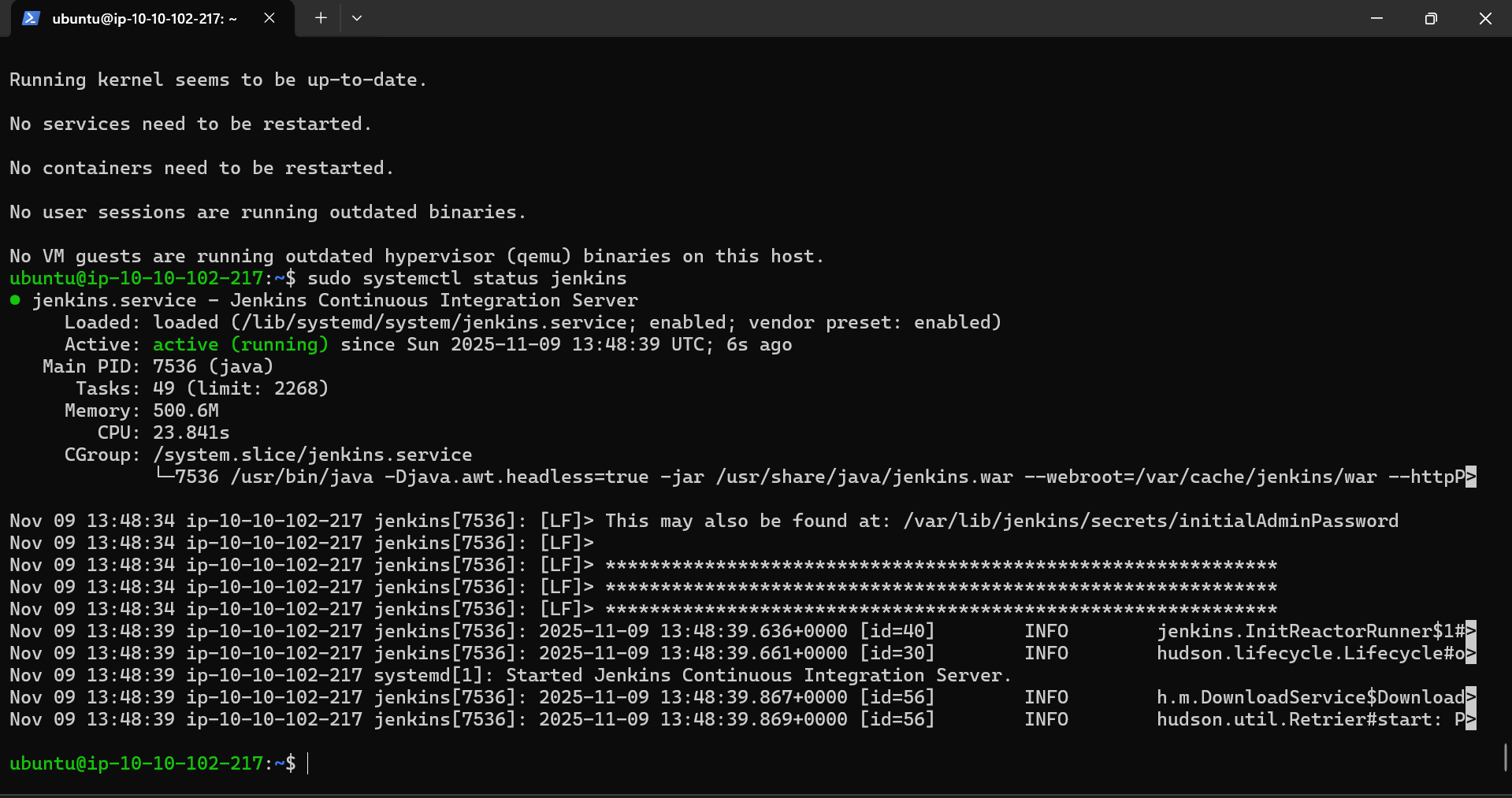


* + Nginx is active on AzureVM, accessed the application via HTTP (IP:80) – *http://<VM\_public\_ip>*

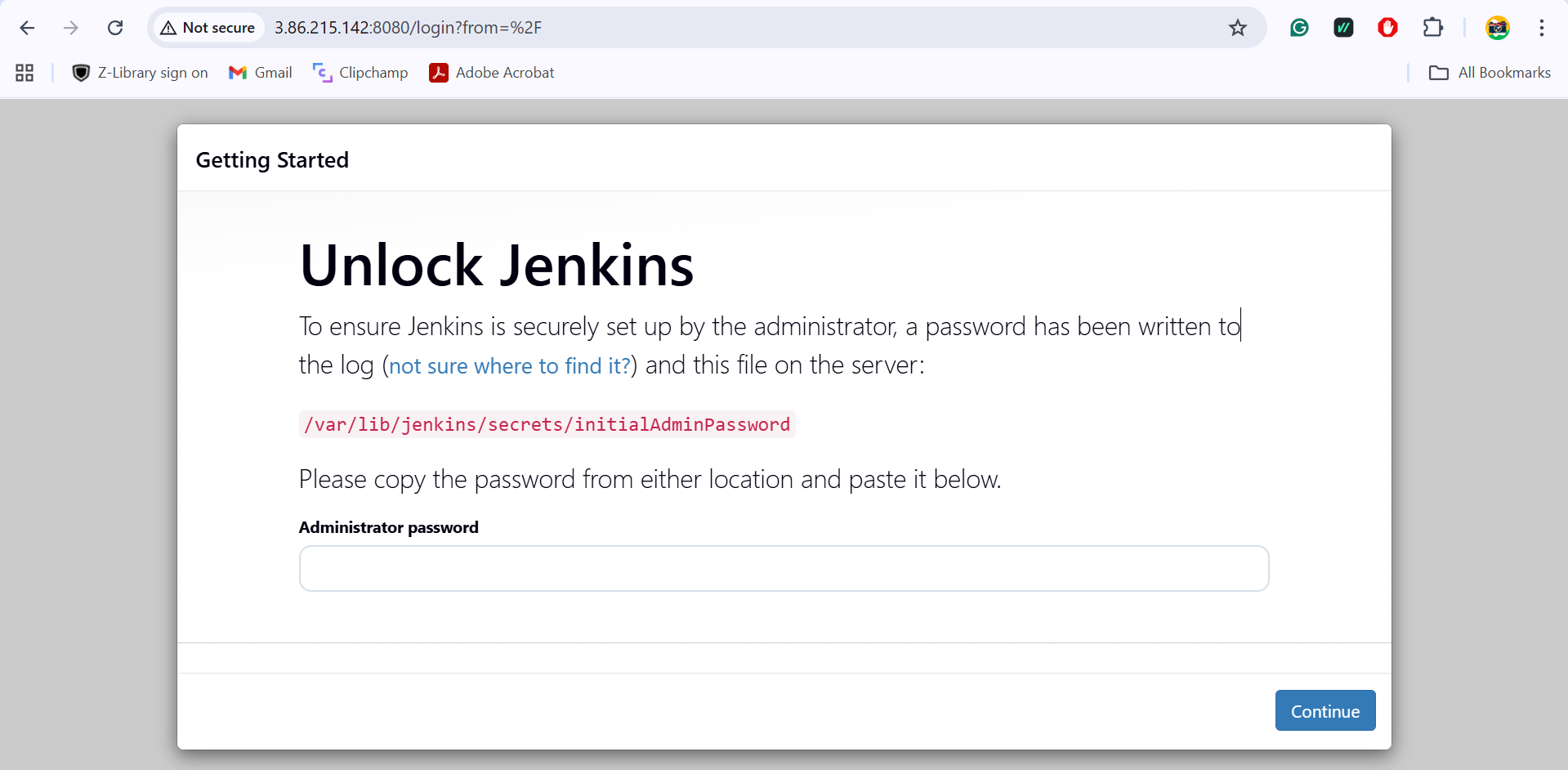


**Task 4: Jenkins Setup for Continuous Deployment**

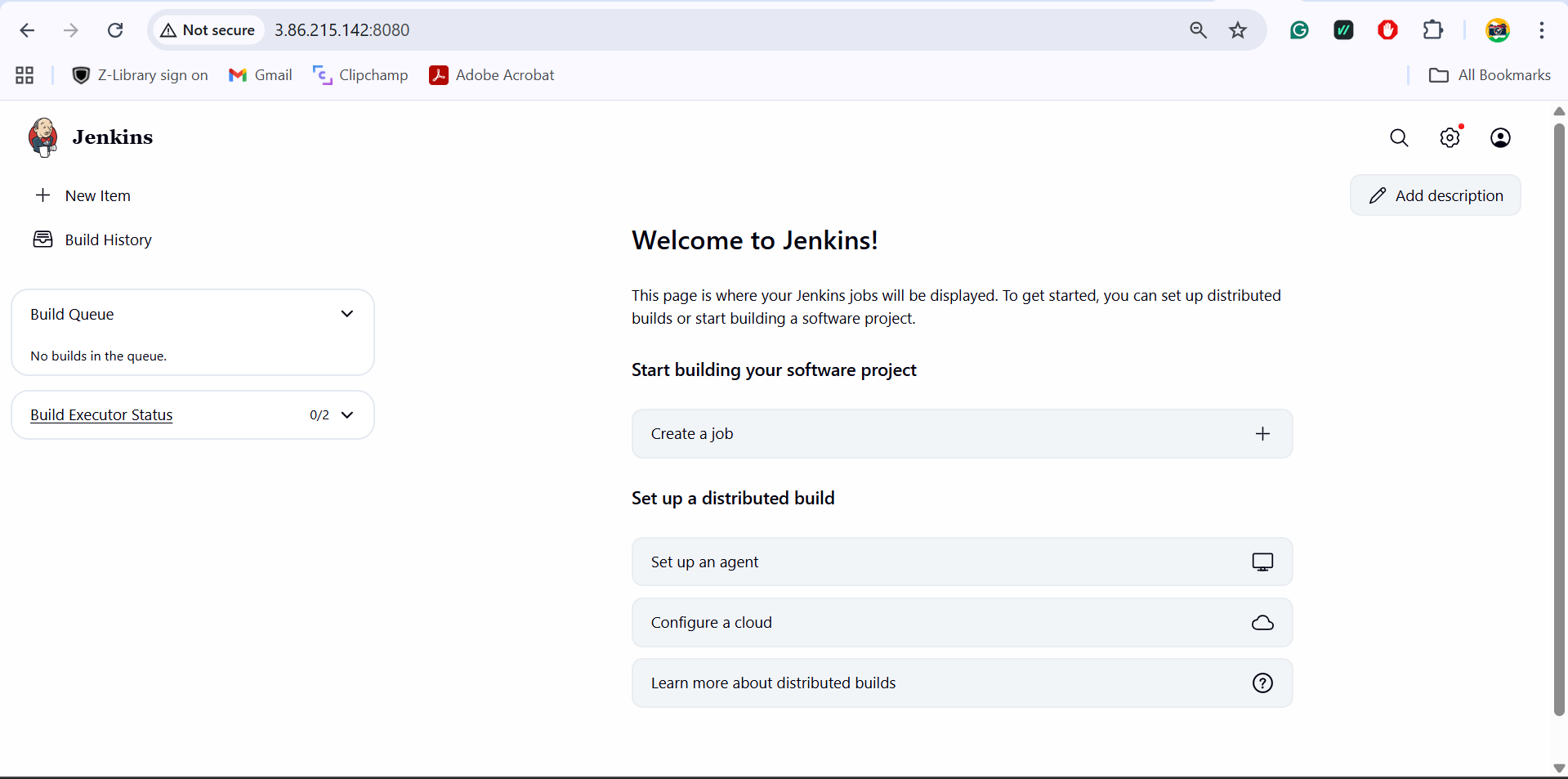
* Install Jenkins on the Tools Machine (AWS).



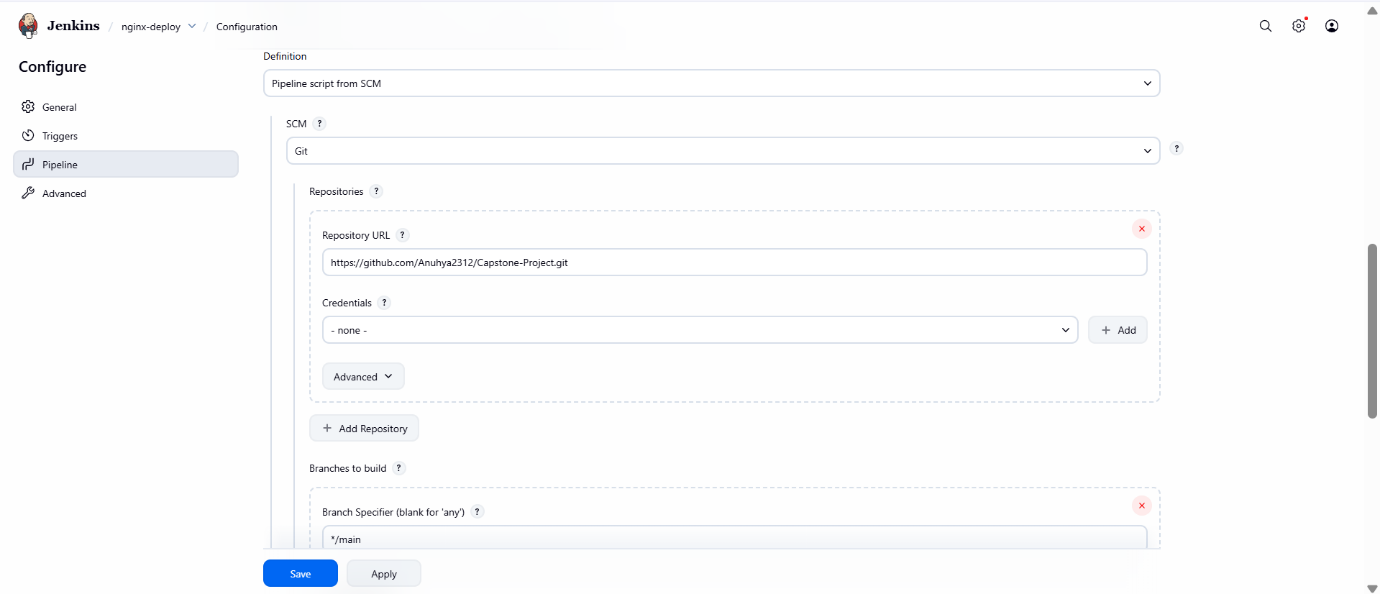
* Used *http://<tool\_ec2\_public\_ip>:8080* on browser for Jenkins page.



* After entering the Administration password and creating username, password – Jenkins page is displayed.



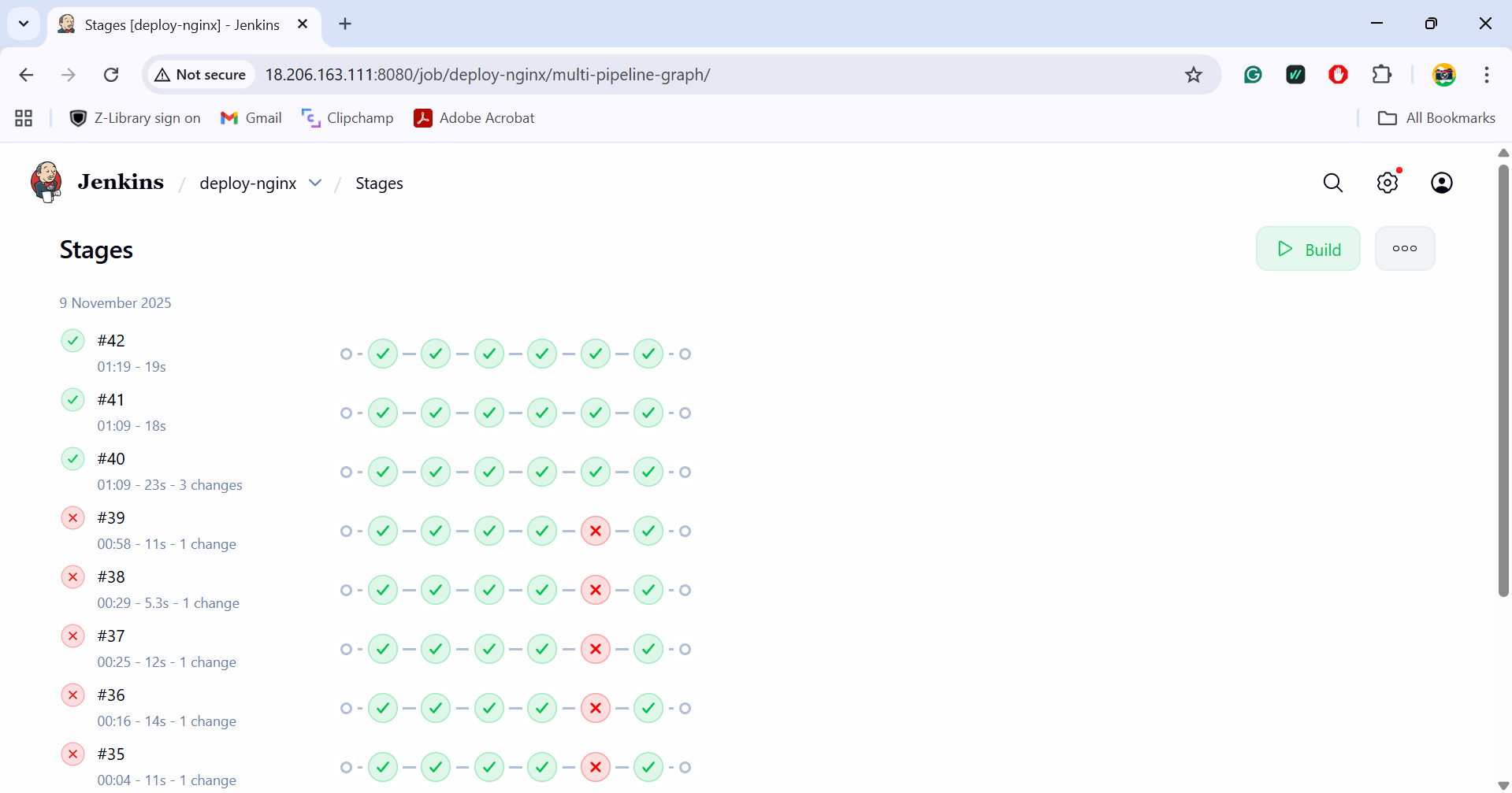
* Installed sshpass on Tools machine - **sudo apt install -y sshpass**
* Created Credentials for AWS private key and Azure username/password in Jenkins.
* Created Jenkins declarative pipeline to Pull index-aws.html and index-azure.html from GitHub.



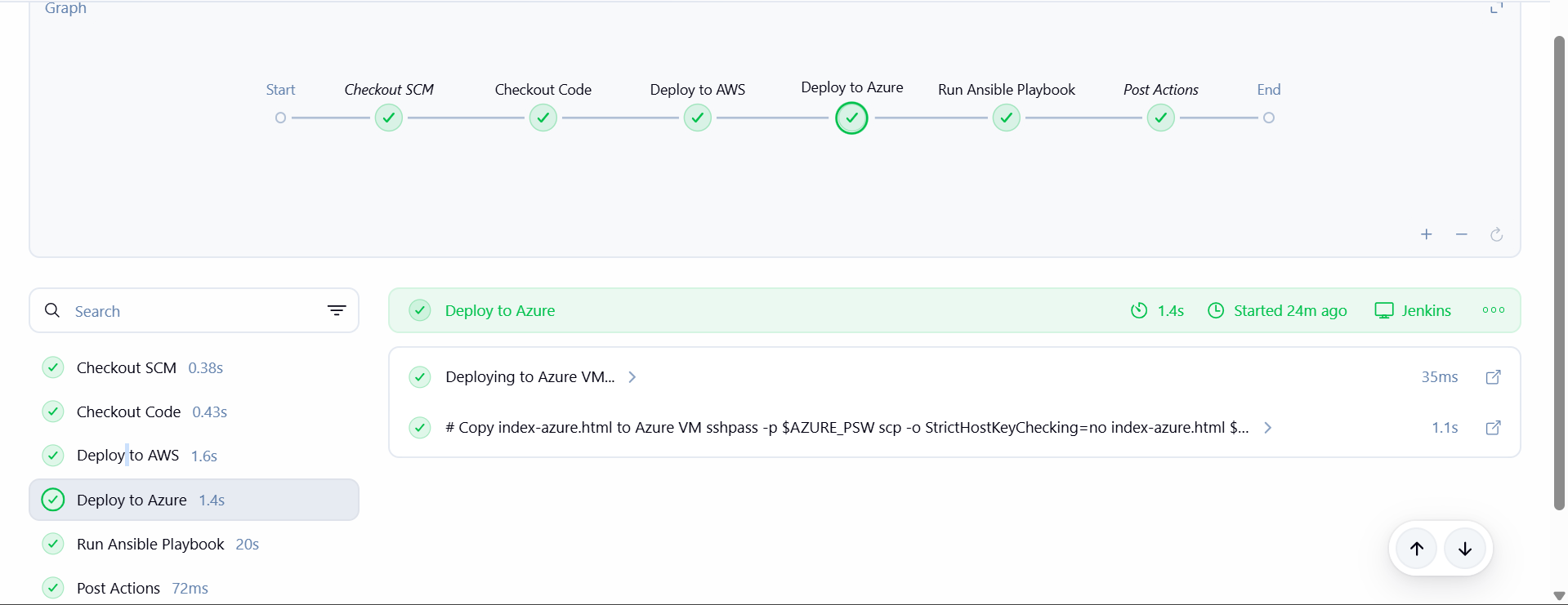
* Jenkins pipeline execution.



* Successful build and deployment.



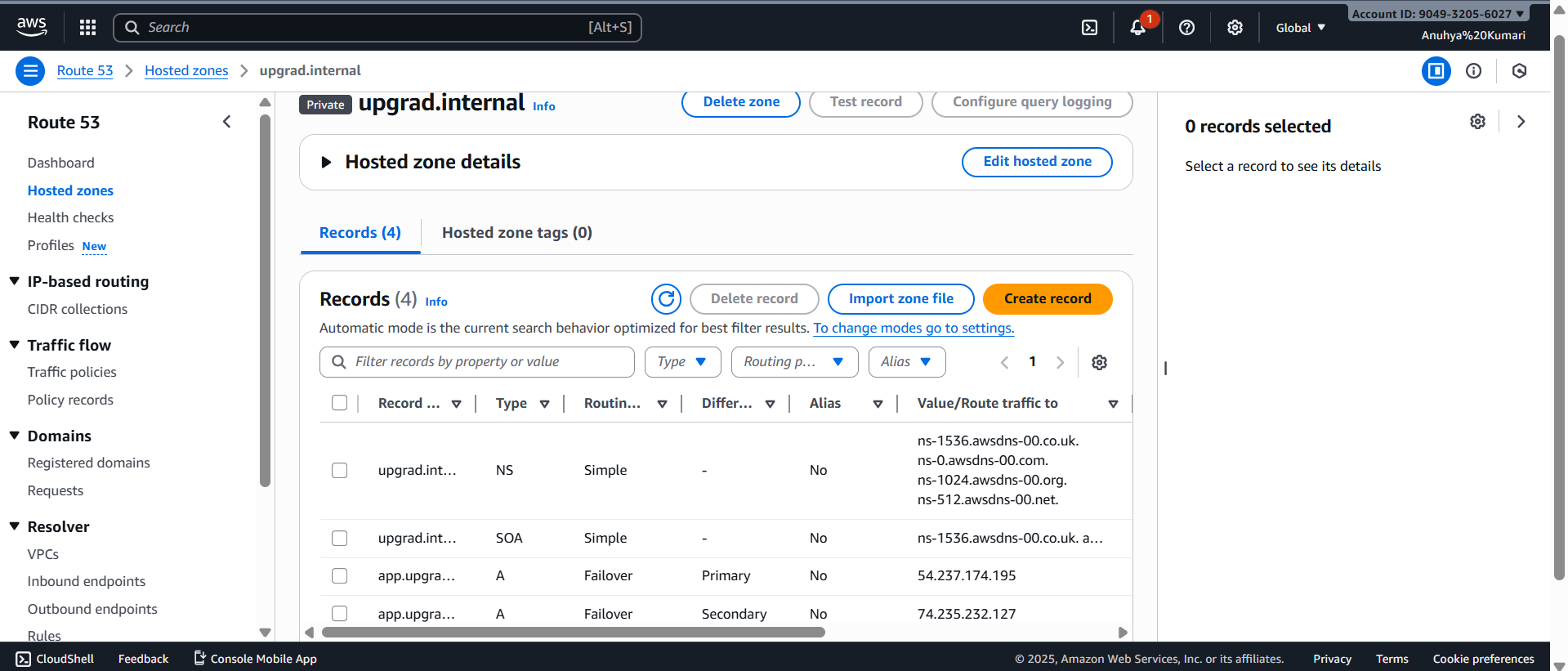
* Logs of Jenkins pulling the HTML files from GitHub and restarting Nginx.

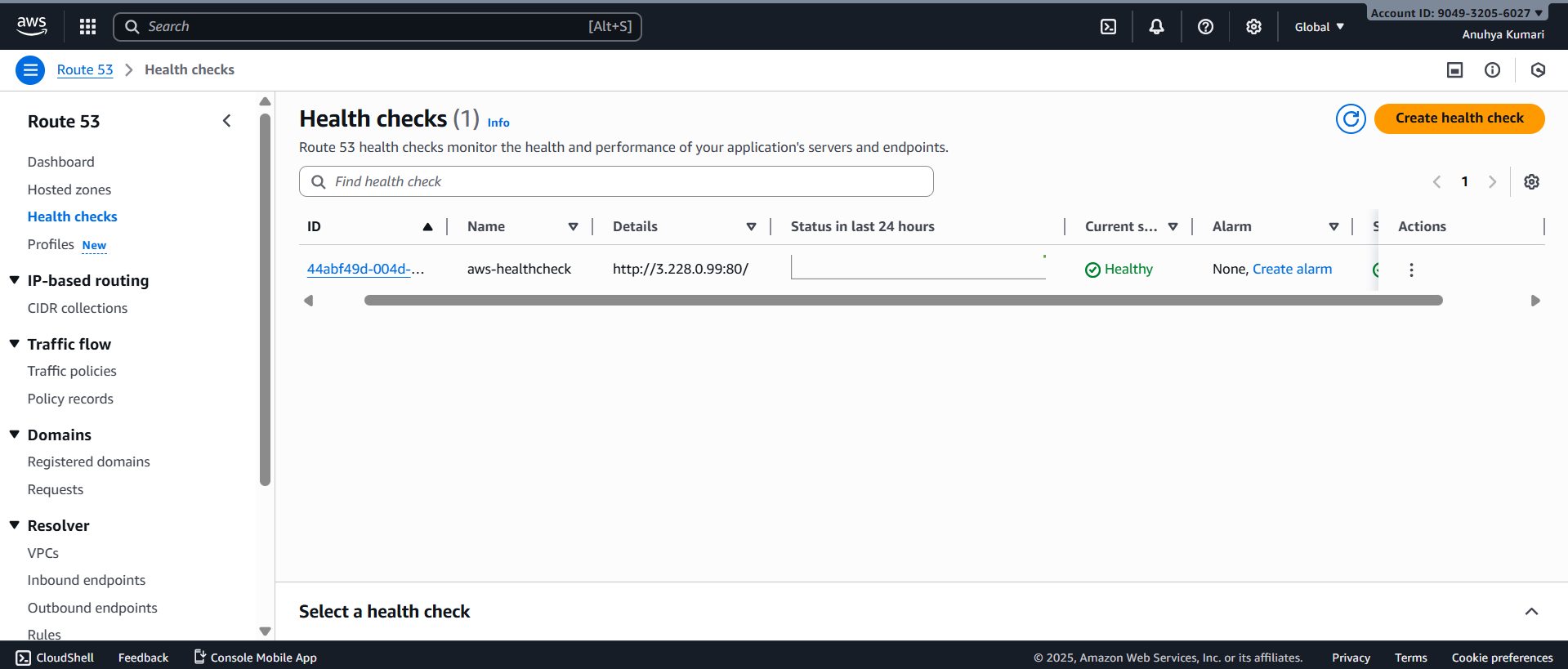




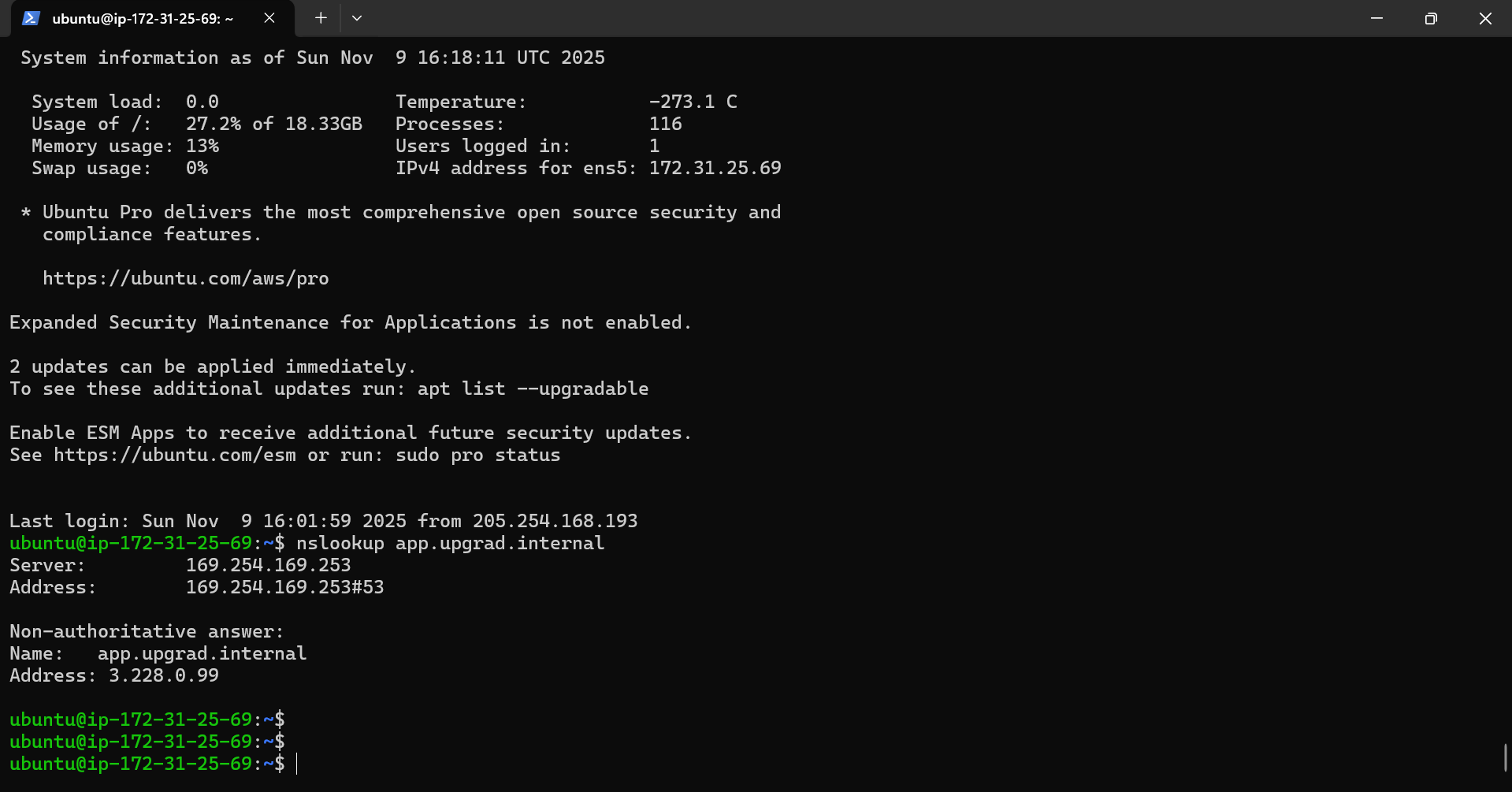
**Task 5: Traffic Management Using AWS Route 53**

* Hosted zone setup in AWS Route 53 and failover policy setup. (primary AWS, secondary Azure).

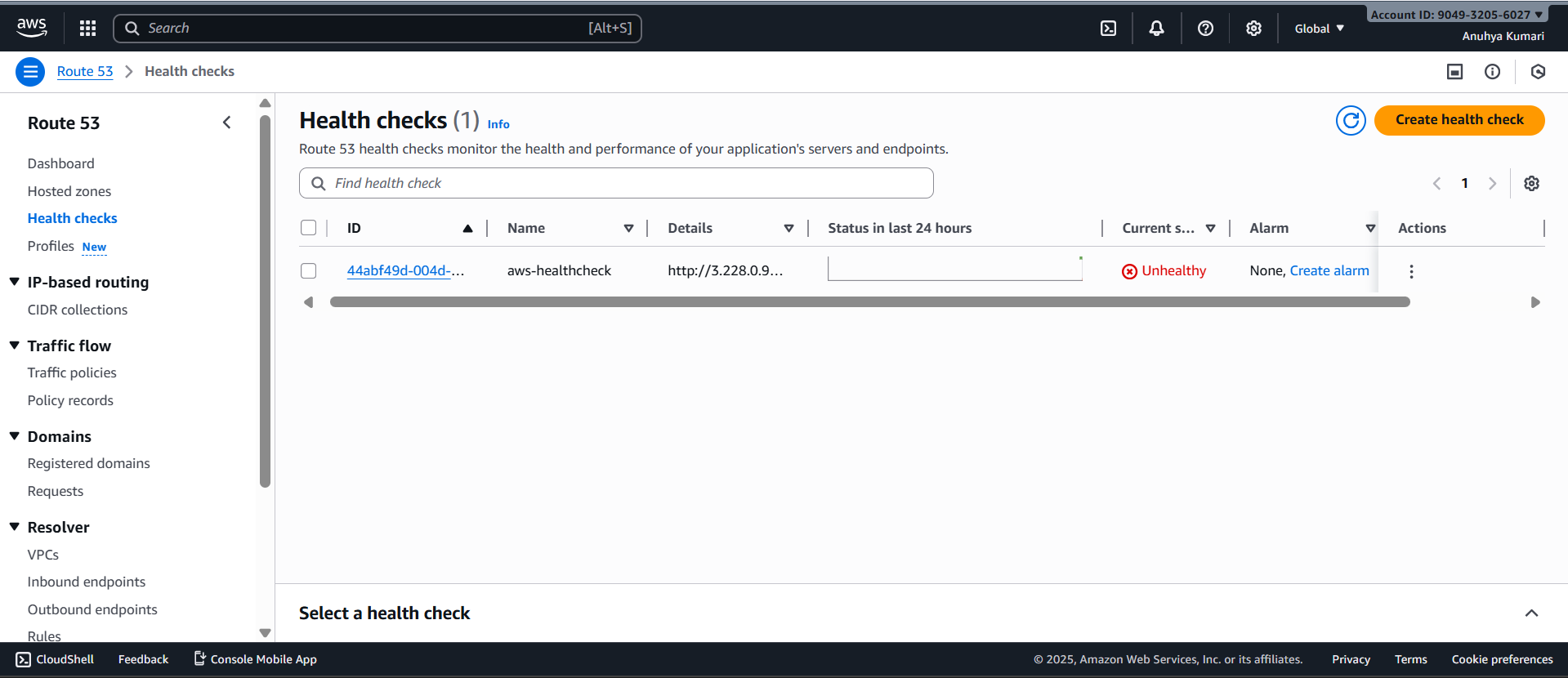




* Before stopping the App machine (AWS), it is displaying AWS public IP address.



* Stopped the App-machine (AWS) instance.



* Route 53 redirecting traffic to Azure.

