MERN Stack Deployment on AWS

Overview

This repository automates the infrastructure deployment and configuration of a MERN (MongoDB, Express, React, Node.js) application on AWS using:

✅ Terraform for infrastructure provisioning (EC2, Security Groups, VPC).

✅ Ansible for configuring servers (Node.js, MongoDB, PM2, application deployment).

📌 Project Architecture

This deployment consists of:

Web Server (Public EC2) → Hosts the frontend & backend app.

Database Server (Private EC2) → Runs MongoDB in a private subnet.

Security Groups → Restrict access between servers.

Terraform → Creates infrastructure (VPC, Subnets, EC2).

Ansible → Configures and deploys the MERN app.

1️⃣ Infrastructure Deployment with Terraform

🔹 Prerequisites

Ensure you have installed:

Terraform (>=1.0)

AWS CLI (configured with credentials)

Ansible (>=2.9)

🔹 Clone the Repository

git clone https://github.com/Manjyyot/AnsibleTM.git

cd AnsibleTM

🔹 Configure Terraform Variables

Edit the terraform.tfvars file (or provide values when prompted):

vpc\_id = "vpc-XXXXXXXXXXXXXX"

allowed\_ssh\_ip = "YOUR\_PUBLIC\_IP/32"

web\_subnet\_cidr = "10.0.1.0/24"

db\_subnet\_cidr = "10.0.2.0/24"

key\_pair\_name = "newManjyyot"

🔹 Deploy Infrastructure

terraform init

terraform apply -auto-approve

This will create:

✅ VPC, Subnets (Public for Web, Private for DB)

✅ Security Groups (Allows only necessary traffic)

✅ EC2 Instances (Web & DB)

🔹 Get Deployed Resources

After successful deployment, Terraform outputs:

terraform output

Example Output:

db\_server\_private\_ip = "10.0.2.171"

web\_server\_public\_ip = "34.238.127.133"

2️⃣ Server Configuration & Application Deployment with Ansible

🔹 Update the Ansible Inventory

Terraform automatically generates the inventory.ini file:

[web]

web\_server ansible\_host=34.238.127.133 ansible\_user=ubuntu ansible\_ssh\_private\_key\_file=/home/ubuntu/newManjyyot.pem

[db]

db\_server ansible\_host=10.0.2.171 ansible\_user=ubuntu ansible\_ssh\_private\_key\_file=/home/ubuntu/newManjyyot.pem ansible\_ssh\_common\_args='-o ProxyCommand="ssh -W %h:%p -i /home/ubuntu/newManjyyot.pem ubuntu@34.238.127.133"'

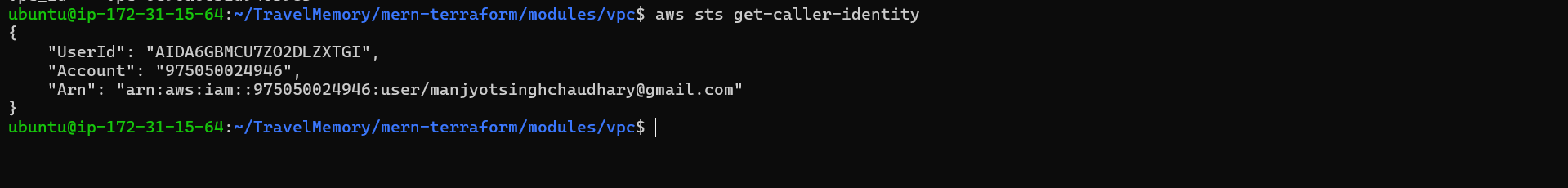
[all:vars]

ansible\_python\_interpreter=/usr/bin/python3

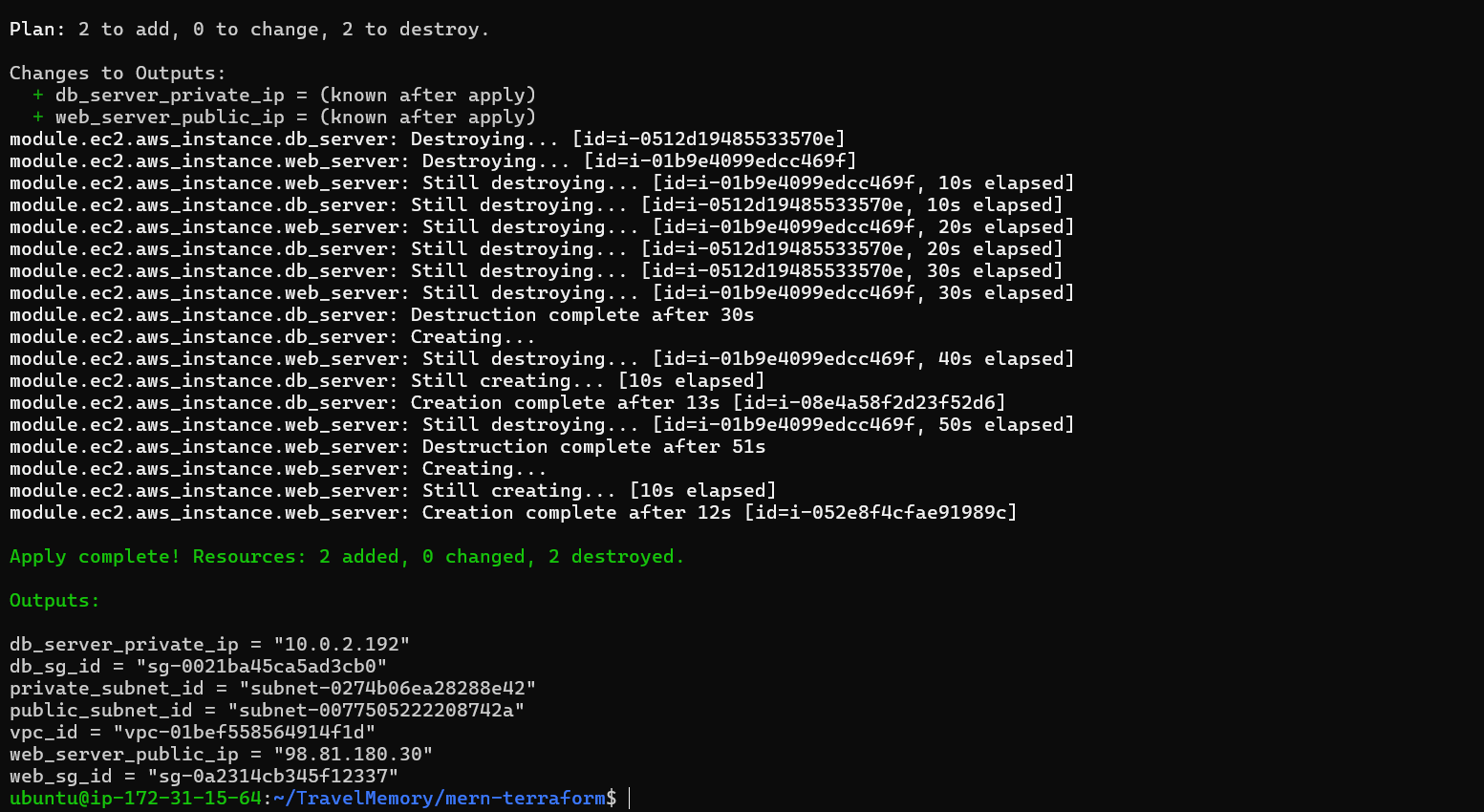
🔹 Deploy MongoDB on the Database Server

ansible-playbook -i inventory.ini ansible/playbook.yml

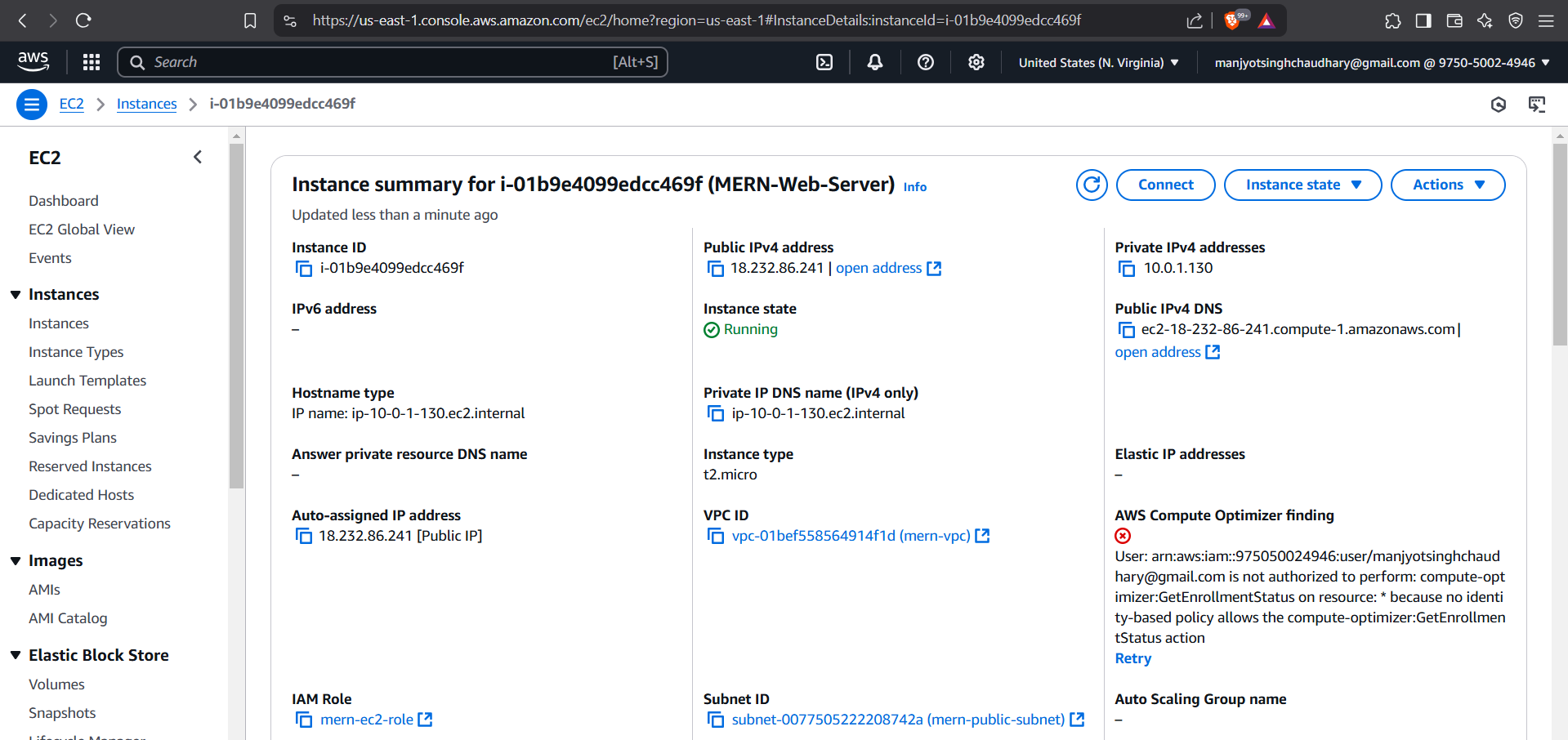
AWS configured properly



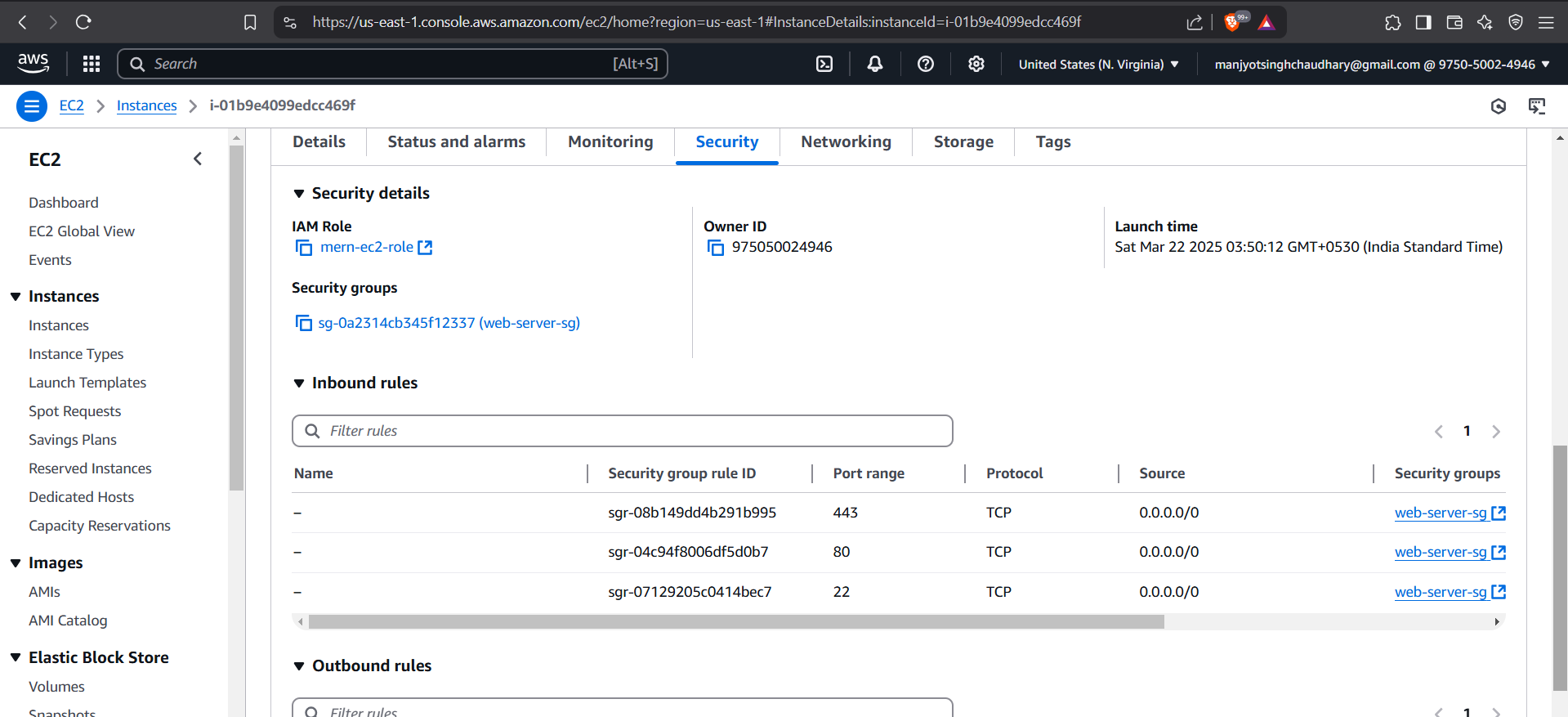
Terraform Output



EC2 created after terraform run

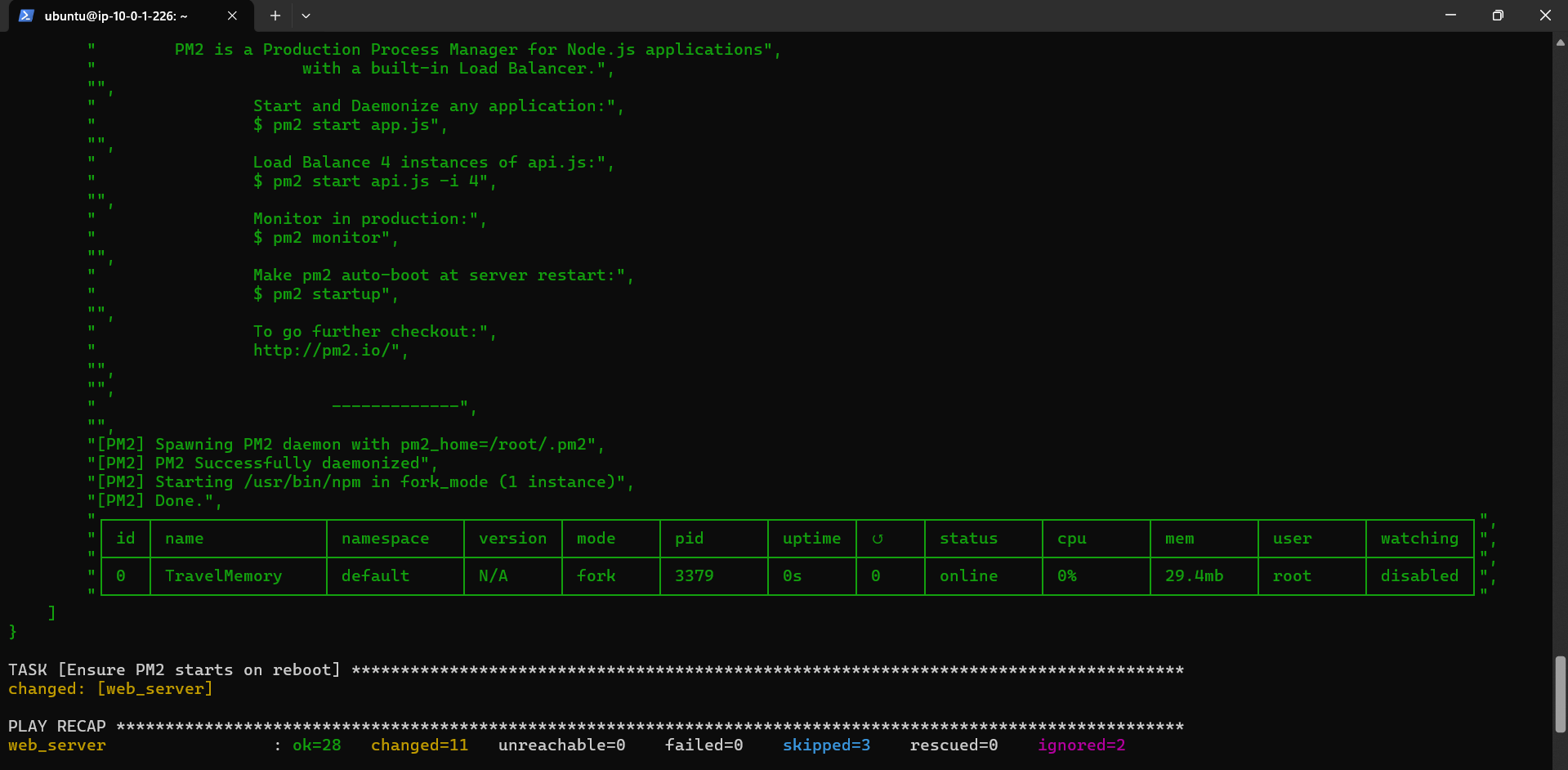


Security groups associated with the created EC2



Web Server Ansible:

Original EC2



Output after running ansible-playbook

