Assignment 2: Cloud Computing

Project Description

This project focuses on gaining hands-on experience with deploying reference software (RS), a React-based video game browser application. Alongside deployment, I will implement process monitoring using Linux commands, Python scripting, and crontab to generate and track system activity. Finally, using Amazon Web Services (AWS) I will create an Ubuntu virtual machine (EC2) in AWS. I will then configure Jupyter Notebook access over the internet using Elastic IP.

Prerequisites

- AWS Account
- Basic knowledge of Linux CLI
- Basic Python and shell scripting skills

Task 1: Acquire Reference Software (RS)

1. **Go to GitHub.**
2. **Search for “videogames browser app” or use this link:**

   https://github.com/Alais29/react-gameapp
3. **Confirm the repository is active and cloneable.**

Task 2: Create Build Script

1. **Create a script named `build.sh` to build the RS:**

```
#!/bin/bash
# Update the package list and install npm
sudo apt update && sudo apt install npm -y

# Clone this repository "react-gameapp" from GitHub
git clone https://github.com/Alais29/react-gameapp.git

# Change to the project directory
cd react-gameapp

# Install the dependencies from package.json
npm install

# Start the application
npm start
```

2. **Run the script:**

```
chmod +x build.sh
./build.sh
```

Task 3: Monitor System Process Count

Calculate the number of processes every minute by using a crontab

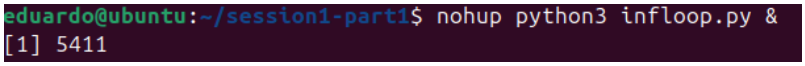
1. **Create Infinite Loop Python Script (`infloop.py`)**

```
#!/usr/bin/env python
import time
f=open("edu.txt","w")
```

```
while(True):
    f.write("edu")
    time.sleep(0.001)

f.close()
```

2. **Run the script in background:** `nohup python infloop.py &`



3. **Kill it with:**

```
ps aux | grep infloop.py
kill -9 <process_id>
```

4. **Count Processes - Bash Script (`count_processes.sh`)**
   - Below bash script was created to upload the csv file (list of data) of the number of processes.

```
#!/bin/bash
TIMESTAMP=$(date +"%Y-%m-%d %H:%M")
OUTPUT_FILE="/home/eduardo/session1-part1/process_count.csv"
PROCESS_COUNT=$(ps aux | wc -l)

echo "$TIMESTAMP, $PROCESS_COUNT" >> $OUTPUT_FILE
```

   - Make it executable:

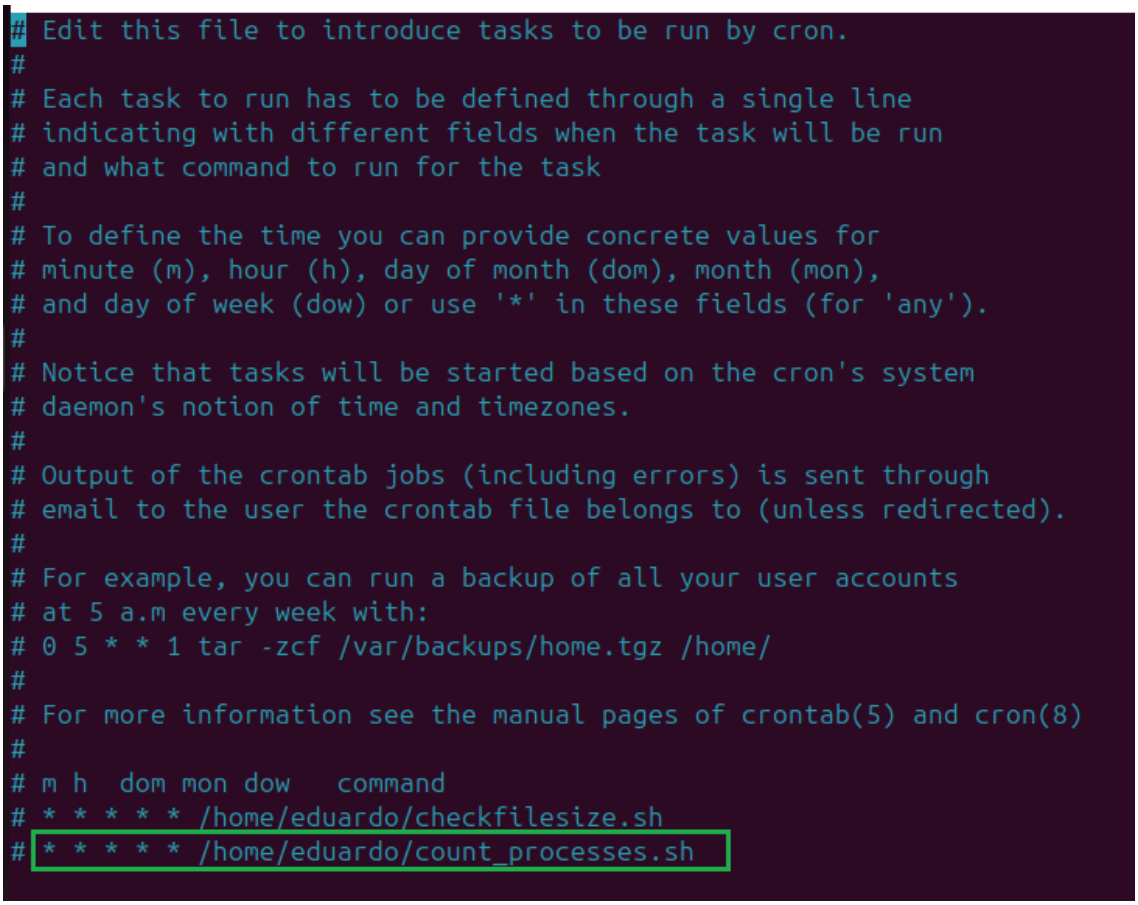
```
chmod +x count_processes.sh
```

3. **Set Up Cron Job that runs every minute:**
   - Edit cron:

```
crontab -e
```

- Add this line:

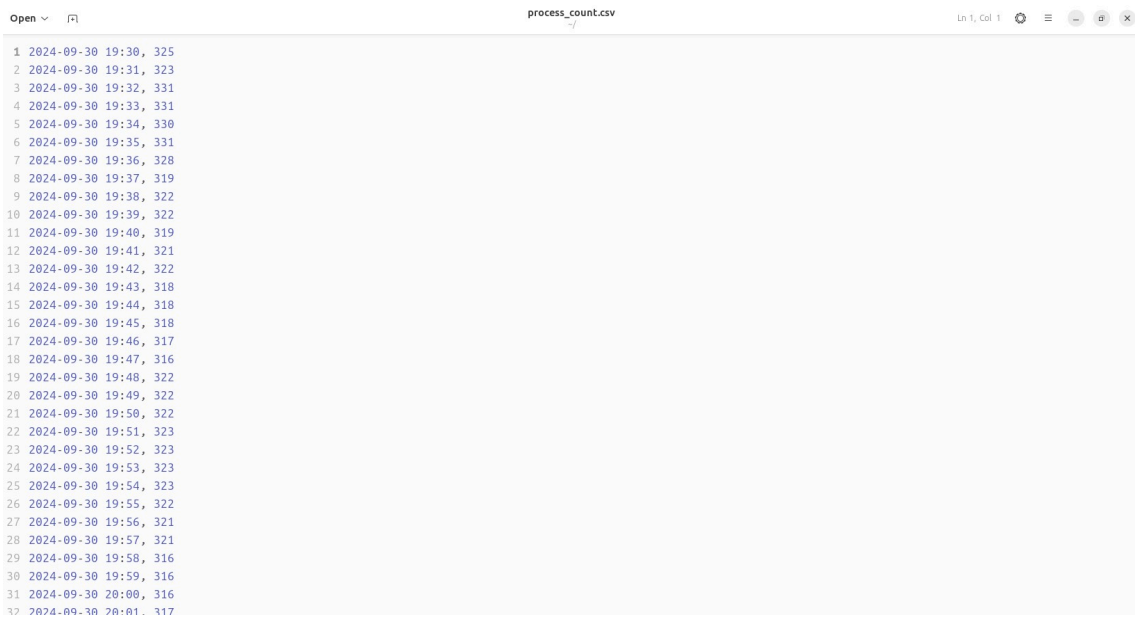
```
* * * * * /home/ubuntu/count_processes.sh
```



- View output:

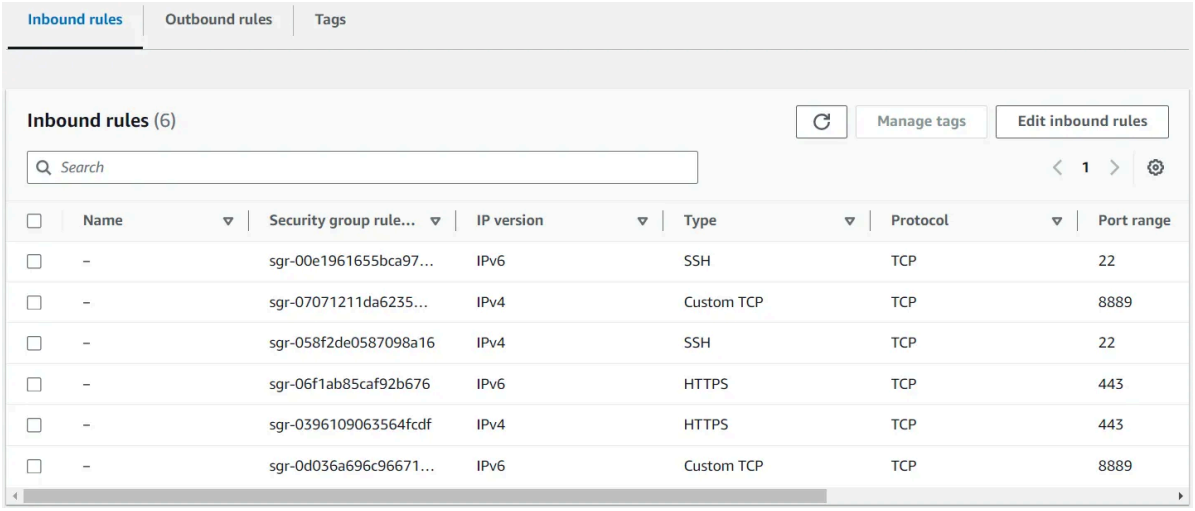
```
cat /home/ubuntu/process_count.csv
```

or open `process_count.csv`



Task 4: Set up AWS EC2 and Upload RS to it

1. **Create AWS EC2 instance**
    - Go to AWS Console → EC2 → Launch Instance
    - AMI: **Ubuntu Server 20.04 LTS**
    - Instance type: **t2.micro** (Free tier)
    - Create or select an existing **key pair**.
    - Configure **security group**:



- Allow SSH (port 22)
- Allow HTTP (port 80)
- Allow custom TCP for **port 8889** (Jupyter)

- Launch instance

---

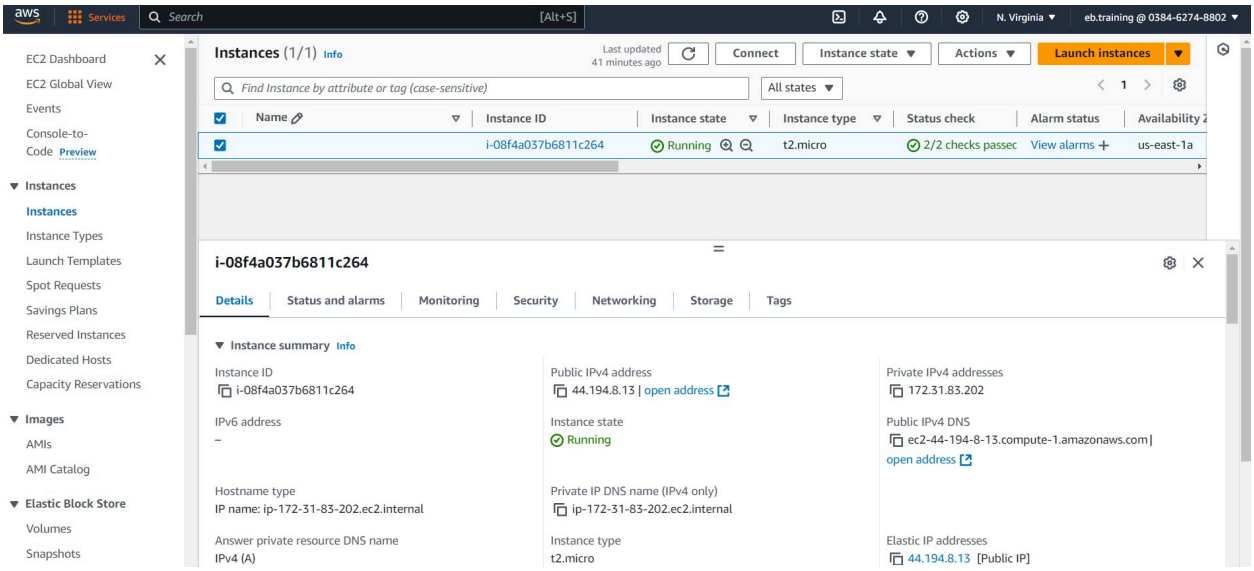
2. Connect to Instance (Replace values)

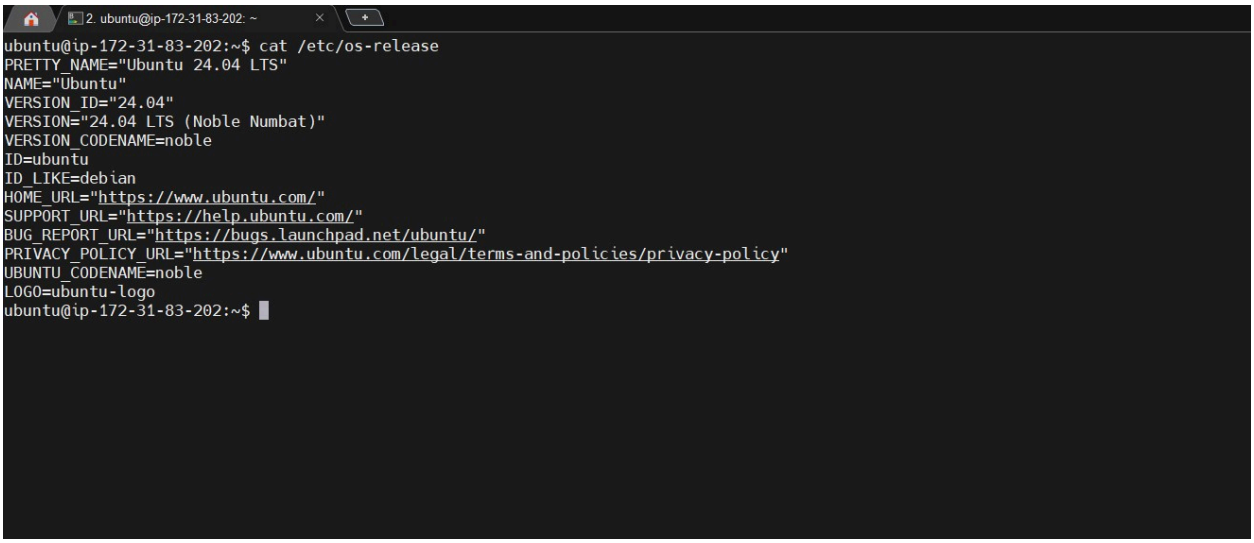
```
ssh -i ~/key_pairs/edu.pem ubuntu@<Public-IP>
```

Example:

SSH into the instance: example:

```
ssh -i key_pairs/edu.pem ubuntu@44.194.8.13
```





3. Upload RS to EC2

```
scp -i /path/to/your-key.pem /local/path/to/RS-file ec2-user@your-instance-ip:/remote/path/
```

Task 5: Install and Run Jupyter Notebook in AWS EC2 instance

Since we already have python within the server, we just need to install Jupyter in the EC2 instance

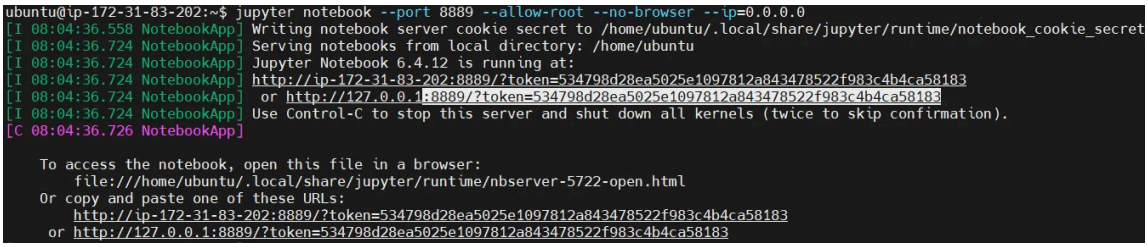
1. **Install Jupyter in AWS instance**

```
sudo apt update
sudo apt install jupyter-core -y
sudo apt install jupyter -y
```

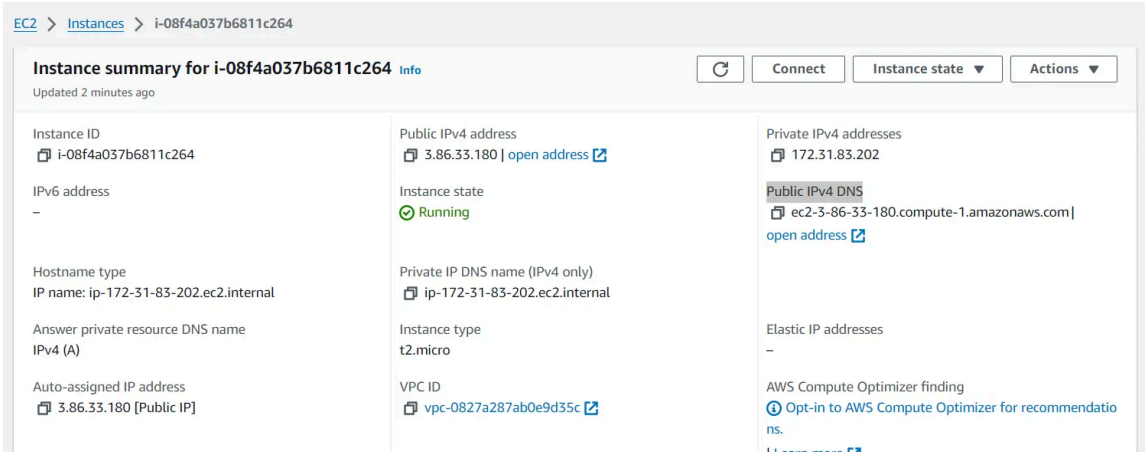
2. **Run Jupyter server:**

```
jupyter notebook --port 8889 --allow-root --no-browser --ip=0.0.0.0
```

- Then Obtain this part:

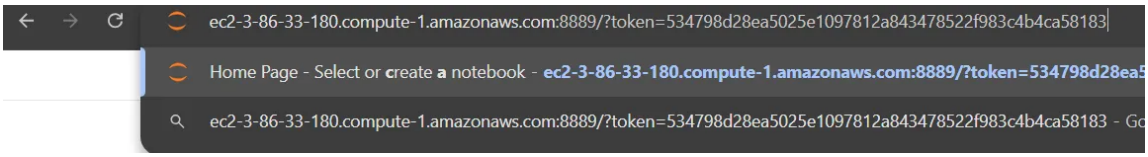


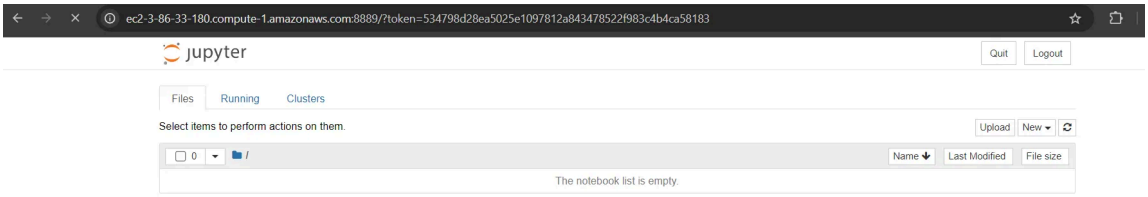
- Also this part:



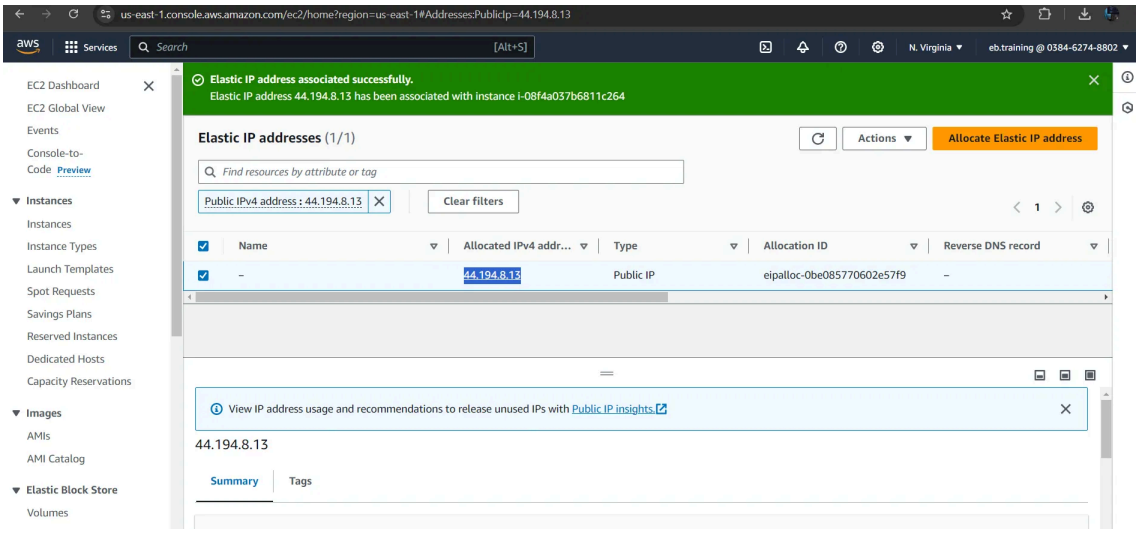
- then add both of this information into the web brower to confirm access:

  `<Public IPv4 DNS>:8889/?token=<token>`





- Create and Associate an Elastic IP
  - In AWS → EC2 → **Elastic IPs**
  - Allocate new address
  - Click **Associate Elastic IP**
    - Select your running instance
  - Obtain the Elastic IP



- Update URL: <Elastic IP>:8889/?token=<token>