Amanda Chang
A01294905

ACIT 4640 - Lab 2

Part 1. AWS-CLI Installation and Configuration

Reference:
4640_Notes/Week2_aws_cli_user_setup.md · main · cit_4640 / 4640_Notes_W24 · GitLab

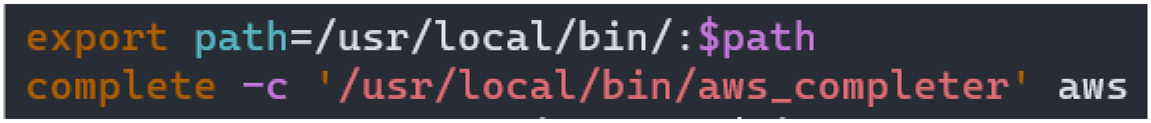
AWS-CLI Installation

```
> curl "https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip" -o
"awscliv2.zip"
> unzip awscliv2.zip
> sudo ./aws/install --bin-dir /usr/local/bin --install-dir
/usr/local/aws-cli --update
```

AWS-CLI Configuration

```
> sudo vi ~/.bashrc
```

Add the following to the end of the ~/.bashrc file:



```
export path=/usr/local/bin/:$path
complete -c '/usr/local/bin/aws_completer' aws
```

Create IAM User

1. Login AWS Console
2. **IAM** → **Users** → **Create user**
   - User name: `acit4640_admin`
   - Do not select "**Provide user access to the AWS Management Console**"
   - Select "**Add user to group**" → **Create group**
   - User group name: `acit4640`
   - Permission policies:
     - AdministratorAccess
     - AmazonVPCFullAccess
     - AmazonRDSFullAccess
     - AmazonRoute53FullAccess

Configure Security Credentials for the IAM User

1. **Access Management** → **Users** → `acit4640_admin`
2. **Security credentials** tab → In the Access keys section, click on **Create access key** → In the Use case section, select **Command Line Interface (CLI)** → **Create access key**

3. Download the access keys CSV file
4. Select **Done**

AWS-CLI Configuration

In WSL:

> `aws configure --profile acit4640_admin`

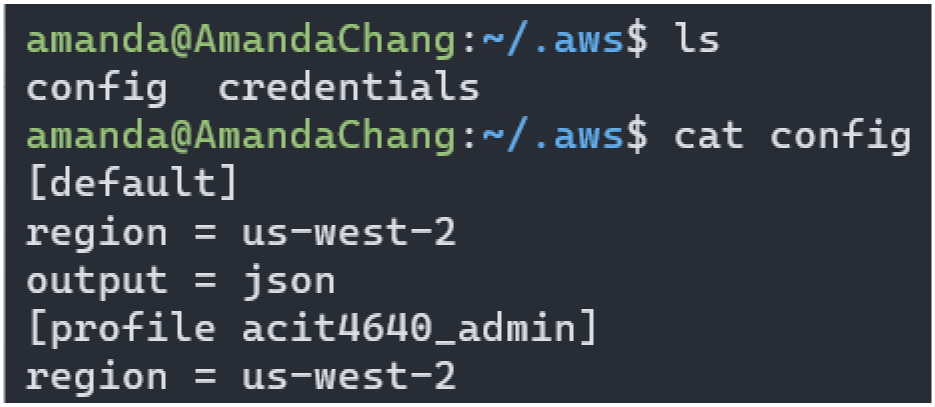
It will prompt you for the following:

- AWS Access Key ID (Found in Access Key CSV file)
- AWS Secret Access Key (Found in Access Key CSV file)
- Default Region Name: `us-west-2`
- Default Output Format (Press ENTER)

The above command will create two files:

> `cd ~/.aws`

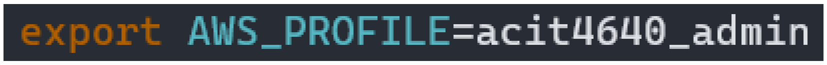
> `ls`



```
amanda@AmandaChang:~/.aws$ ls
config  credentials
amanda@AmandaChang:~/.aws$ cat config
[default]
region = us-west-2
output = json
[profile acit4640_admin]
region = us-west-2
```

> `sudo vi ~/.bashrc`

Add the following to the end of the ~/.bashrc file:



```
export AWS_PROFILE=acit4640_admin
```

RESTART WSL before continuing

Part 2. SSH Key Setup

Reference: [create-key-pair — AWS CLI 2.15.10 Command Reference (amazonaws.com)]

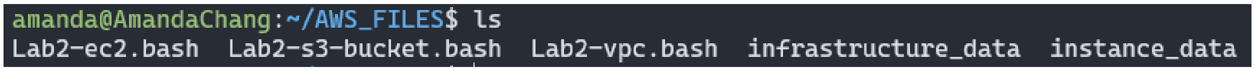
```
> aws ec2 create-key-pair --profile acit4640_admin --key-name MyKeyPair
--query 'KeyMaterial' --output text > MyKeyPair.pem
```

> `mv MyKeyPair.pem /home/amanda/.ssh`

Amanda Chang
A01294905

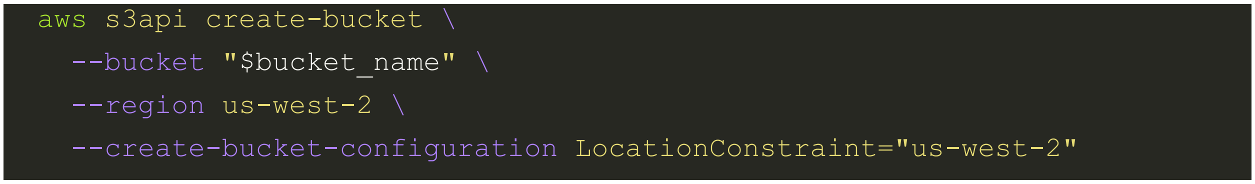
Part 3. AWS-CLI S3 Bucket Setup

I created a new folder called **AWS_FILES** that contains scripts for Lab 2.



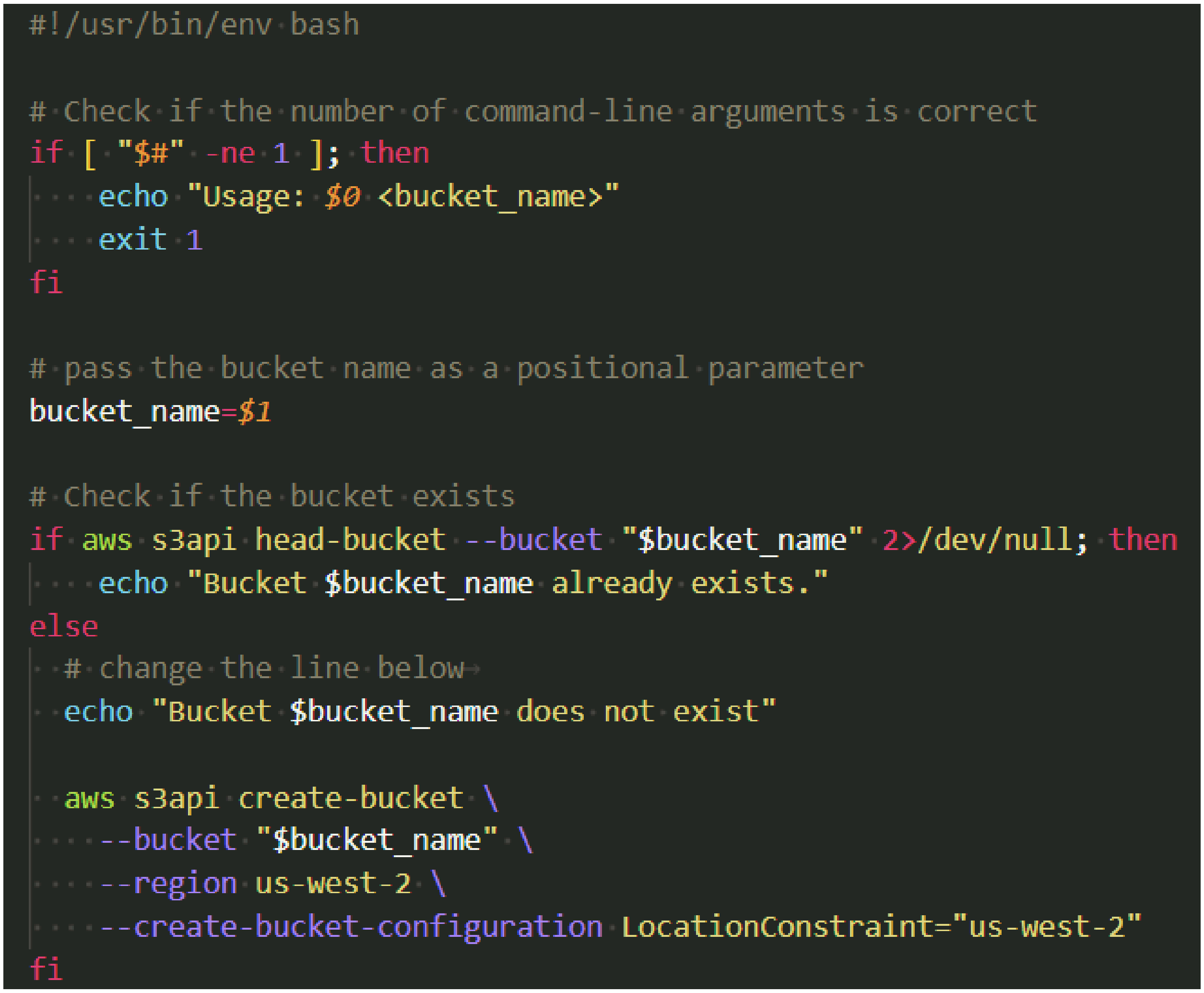
```
amanda@AmandaChang:~/AWS_FILES$ ls
Lab2-ec2.bash  Lab2-s3-bucket.bash  Lab2-vpc.bash  infrastructure_data  instance_data
```

The following command is added to the **Lab2-s3-bucket.bash** script:



```
aws s3api create-bucket \
  --bucket "$bucket_name" \
  --region us-west-2 \
  --create-bucket-configuration LocationConstraint="us-west-2"
```

Lab2-s3-bucket.bash script:

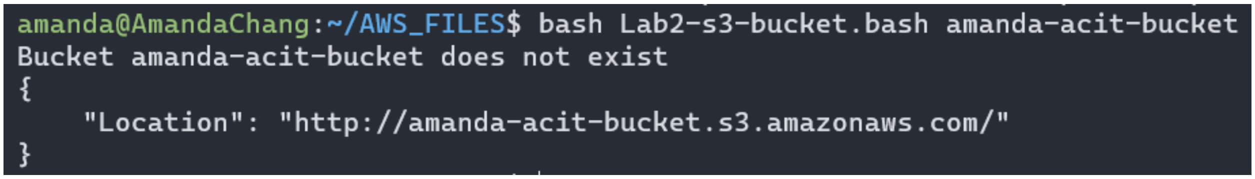


```
#!/usr/bin/env bash

# Check if the number of command-line arguments is correct
if [ "$#" -ne 1 ]; then
    echo "Usage: $0 <bucket_name>"
    exit 1
fi

# pass the bucket name as a positional parameter
bucket_name=$1

# Check if the bucket exists
if aws s3api head-bucket --bucket "$bucket_name" 2>/dev/null; then
    echo "Bucket $bucket_name already exists."
else
  # change the line below
  echo "Bucket $bucket_name does not exist"

  aws s3api create-bucket \
    --bucket "$bucket_name" \
    --region us-west-2 \
    --create-bucket-configuration LocationConstraint="us-west-2"
fi
```

Amanda Chang
A01294905

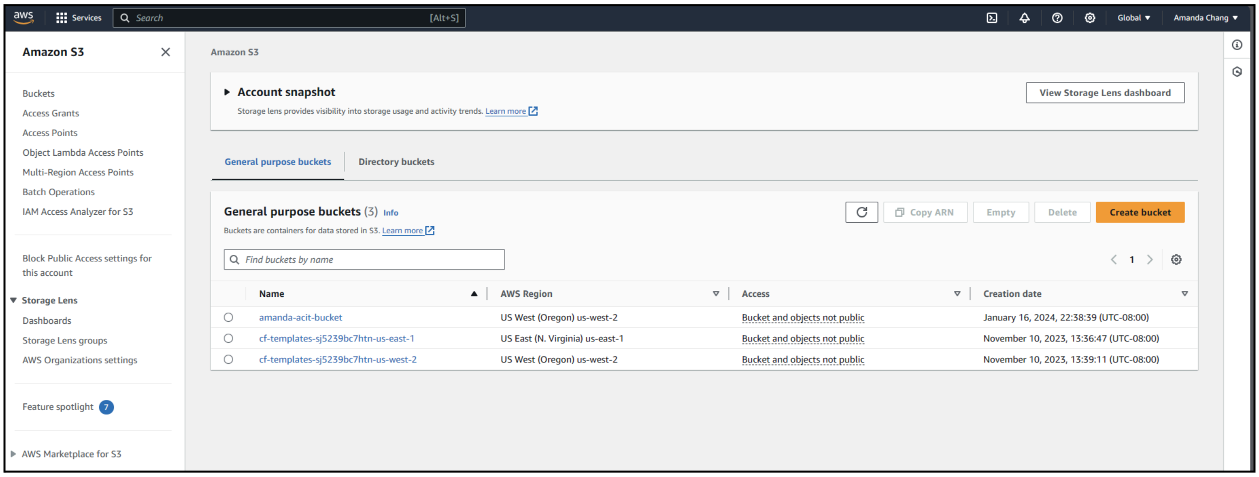
Execute the file by using the following command:

> bash Lab2-s3-bucket.bash amanda-acit-bucket



```
amanda@AmandaChang:~/AWS_FILES$ bash Lab2-s3-bucket.bash amanda-acit-bucket
Bucket amanda-acit-bucket does not exist
{
    "Location": "http://amanda-acit-bucket.s3.amazonaws.com/"
}
```

Go to S3 in AWS Console:



General purpose buckets (3) Info

Buckets are containers for data stored in S3. Learn more

	Name	AWS Region	Access	Creation date
○	amanda-acit-bucket	US West (Oregon) us-west-2	Bucket and objects not public	January 16, 2024, 22:38:39 (UTC-08:00)
○	cf-templates-sj5239bc7htn-us-east-1	US East (N. Virginia) us-east-1	Bucket and objects not public	November 10, 2023, 13:36:47 (UTC-08:00)
○	cf-templates-sj5239bc7htn-us-west-2	US West (Oregon) us-west-2	Bucket and objects not public	November 10, 2023, 13:39:11 (UTC-08:00)

Delete the bucket using the following command:

> aws s3api delete-bucket --bucket amanda-acit-bucket --region us-west-2

Amanda Chang
A01294905

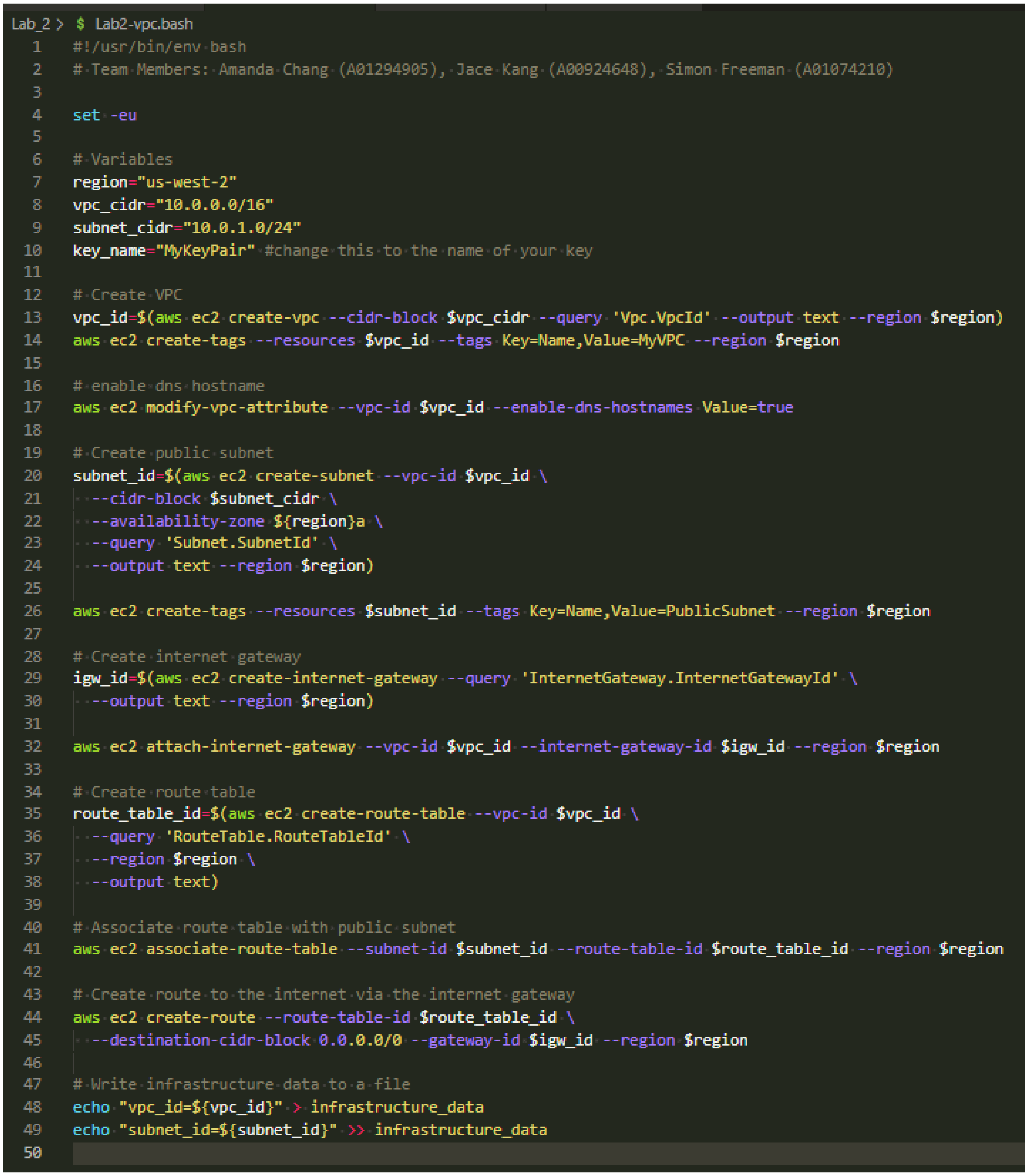
Part 4. AWS-CLI VPC and EC2 Creation

VPC Creation

The following command is modified in the **Lab2-vpc.bash** script:

```
key_name="MyKeyPair"
```

Lab2-vpc.bash script:

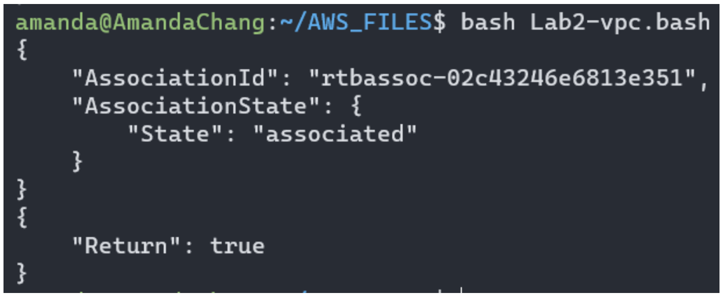


```bash
#!/usr/bin/env bash
# Team Members: Amanda Chang (A01294905), Jace Kang (A00924648), Simon Freeman (A01074210)

set -eu

# Variables
region="us-west-2"
vpc_cidr="10.0.0.0/16"
subnet_cidr="10.0.1.0/24"
key_name="MyKeyPair" #change this to the name of your key

# Create VPC
vpc_id=$(aws ec2 create-vpc --cidr-block $vpc_cidr --query 'Vpc.VpcId' --output text --region $region)
aws ec2 create-tags --resources $vpc_id --tags Key=Name,Value=MyVPC --region $region

# enable dns hostname
aws ec2 modify-vpc-attribute --vpc-id $vpc_id --enable-dns-hostnames Value=true

# Create public subnet
subnet_id=$(aws ec2 create-subnet --vpc-id $vpc_id \
  --cidr-block $subnet_cidr \
  --availability-zone ${region}a \
  --query 'Subnet.SubnetId' \
  --output text --region $region)

aws ec2 create-tags --resources $subnet_id --tags Key=Name,Value=PublicSubnet --region $region

# Create internet gateway
igw_id=$(aws ec2 create-internet-gateway --query 'InternetGateway.InternetGatewayId' \
  --output text --region $region)

aws ec2 attach-internet-gateway --vpc-id $vpc_id --internet-gateway-id $igw_id --region $region

# Create route table
route_table_id=$(aws ec2 create-route-table --vpc-id $vpc_id \
  --query 'RouteTable.RouteTableId' \
  --region $region \
  --output text)

# Associate route table with public subnet
aws ec2 associate-route-table --subnet-id $subnet_id --route-table-id $route_table_id --region $region

# Create route to the internet via the internet gateway
aws ec2 create-route --route-table-id $route_table_id \
  --destination-cidr-block 0.0.0.0/0 --gateway-id $igw_id --region $region

# Write infrastructure data to a file
echo "vpc_id=${vpc_id}" > infrastructure_data
echo "subnet_id=${subnet_id}" >> infrastructure_data
```

Amanda Chang
A01294905

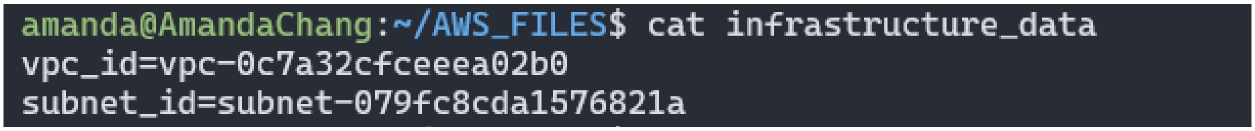
Execute the file by using the following command:

> bash Lab2-vpc.bash



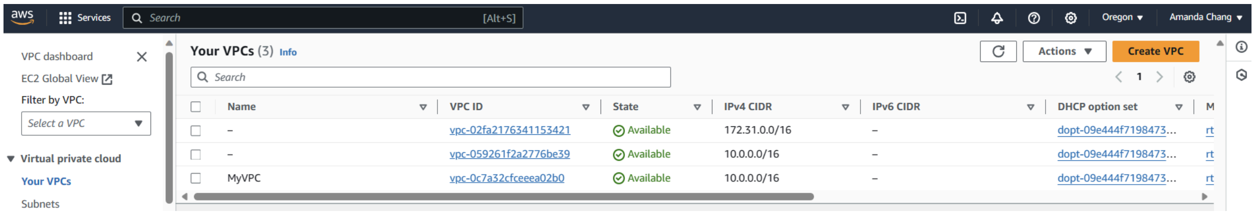
```
amanda@AmandaChang:~/AWS_FILES$ bash Lab2-vpc.bash
{
    "AssociationId": "rtbassoc-02c43246e6813e351",
    "AssociationState": {
        "State": "associated"
    }
}
{
    "Return": true
}
```

Infrastructure_data file:



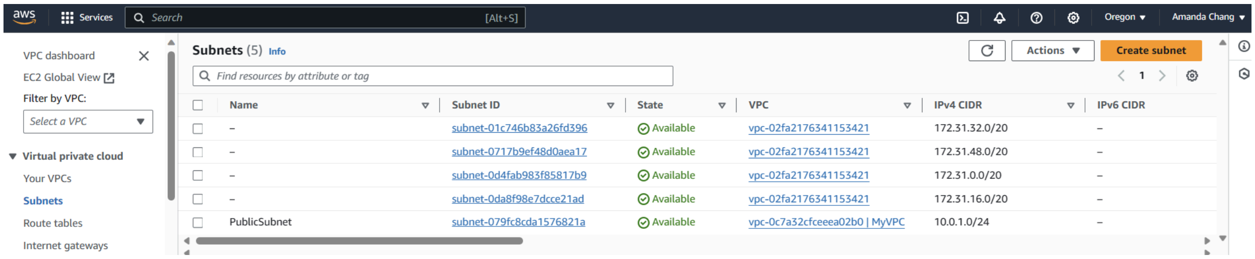
```
amanda@AmandaChang:~/AWS_FILES$ cat infrastructure_data
vpc_id=vpc-0c7a32cfceeea02b0
subnet_id=subnet-079fc8cda1576821a
```

Go to VPC in AWS Console:



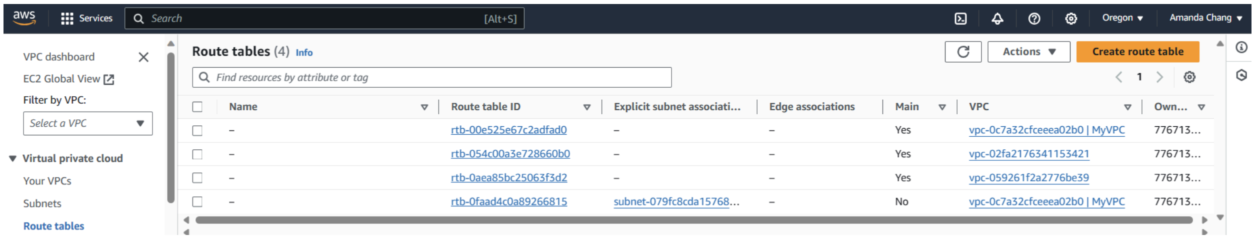
Your VPCs (3) Info

Name	VPC ID	State	IPv4 CIDR	IPv6 CIDR	DHCP option set	M
–	vpc-02fa2176341153421	✓ Available	172.31.0.0/16	–	dopt-09e444f7198473…	rt
–	vpc-059261f2a2776be39	✓ Available	10.0.0.0/16	–	dopt-09e444f7198473…	rt
MyVPC	vpc-0c7a32cfceeea02b0	✓ Available	10.0.0.0/16	–	dopt-09e444f7198473…	rt



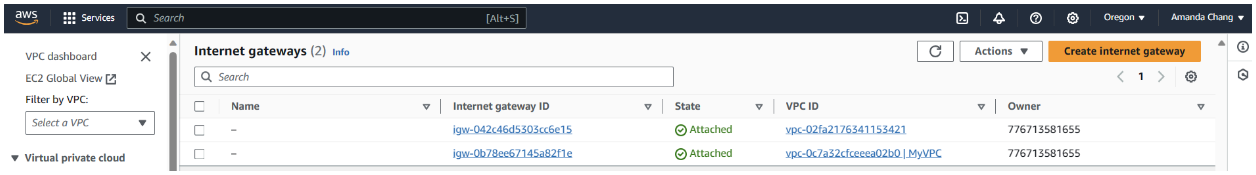
Subnets (5) Info

Name	Subnet ID	State	VPC	IPv4 CIDR	IPv6 CIDR
–	subnet-01c746b83a26fd396	✓ Available	vpc-02fa2176341153421	172.31.32.0/20	–
–	subnet-0717b9ef48d0aea17	✓ Available	vpc-02fa2176341153421	172.31.48.0/20	–
–	subnet-0d4fab983f85817b9	✓ Available	vpc-02fa2176341153421	172.31.0.0/20	–
–	subnet-0da8f98e7dcce21ad	✓ Available	vpc-02fa2176341153421	172.31.16.0/20	–
PublicSubnet	subnet-079fc8cda1576821a	✓ Available	vpc-0c7a32cfceeea02b0 | MyVPC	10.0.1.0/24	–



Route tables (4) Info

Name	Route table ID	Explicit subnet associati…	Edge associations	Main	VPC	Own…
–	rtb-00e525e67c2adfad0	–	–	Yes	vpc-0c7a32cfceeea02b0 | MyVPC	776713…
–	rtb-054c00a3e728660b0	–	–	Yes	vpc-02fa2176341153421	776713…
–	rtb-0aea85bc25063f3d2	–	–	Yes	vpc-059261f2a2776be39	776713…
–	rtb-0faad4c0a89266815	subnet-079fc8cda15768…	–	No	vpc-0c7a32cfceeea02b0 | MyVPC	776713…



Internet gateways (2) Info

Name	Internet gateway ID	State	VPC ID	Owner
–	igw-042c46d5303cc6e15	✓ Attached	vpc-02fa2176341153421	776713581655
–	igw-0b78ee67145a82f1e	✓ Attached	vpc-0c7a32cfceeea02b0 | MyVPC	776713581655

Amanda Chang
A01294905

EC2 Creation

Reference:
https://docs.aws.amazon.com/codedeploy/latest/userguide/instances-ec2-create.html

The following command is added in the **Lab2-ec2.bash** script:

```
source infrastructure_data
```

```
instance_id=$(aws ec2 run-instances \
  --image-id $ubuntu_ami \
  --key-name $key_name \
  --count 1 \
  --instance-type $instance_type \
  --subnet-id $subnet_id \
  --security-group-ids $security_group_id \
  --query 'Instances[0].InstanceId' \
  --output text \
  --associate-public-ip-address)
```

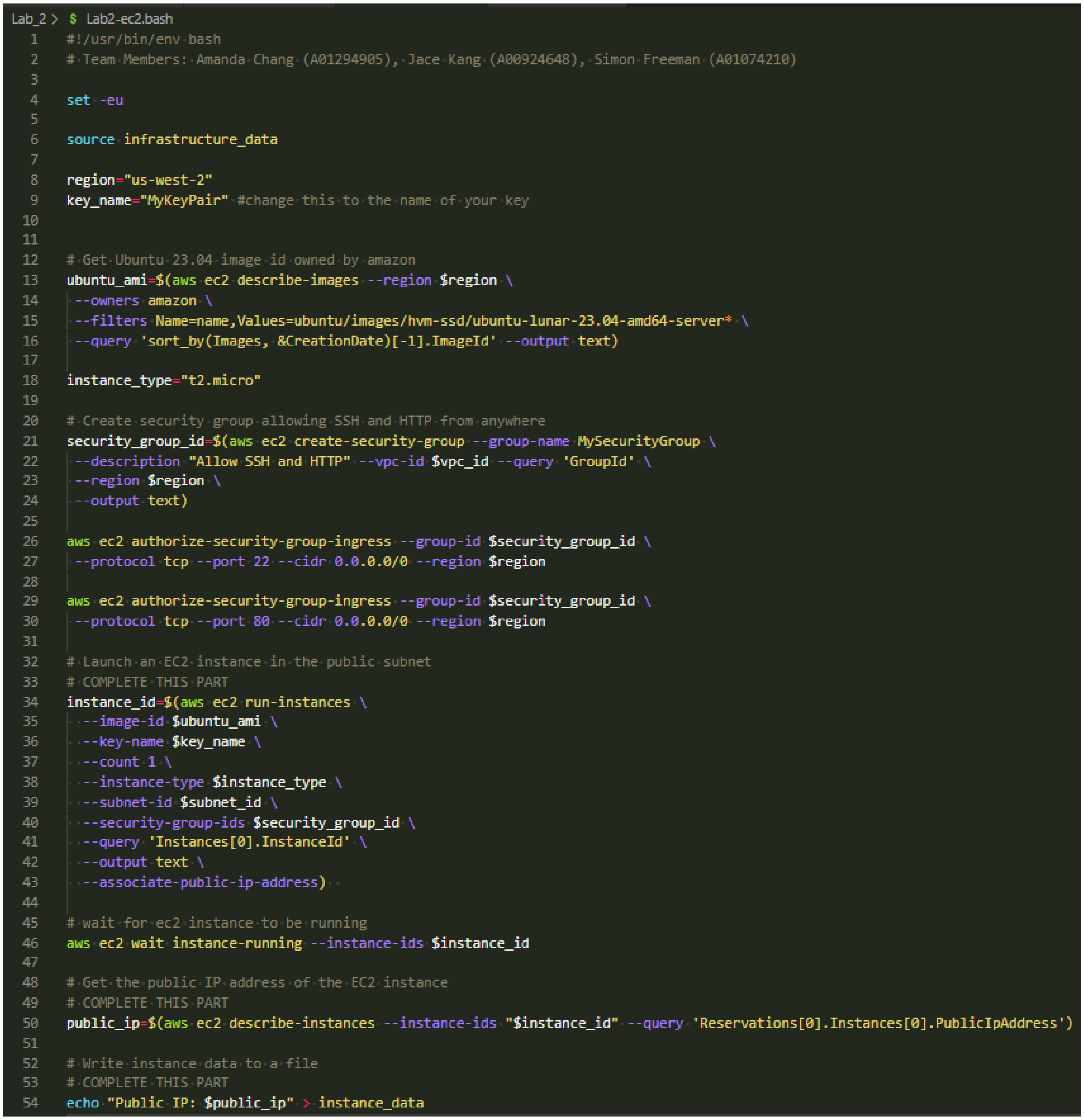
```
public_ip=$(aws ec2 describe-instances --instance-ids "$instance_id"
--query 'Reservations[0].Instances[0].PublicIpAddress')

echo "Public IP: $public_ip" > instance_data
```

The following command is modified in the **Lab2-ec2.bash** script:

```
key_name="MyKeyPair"
```

Amanda Chang
A01294905

Lab2-ec2.bash script:

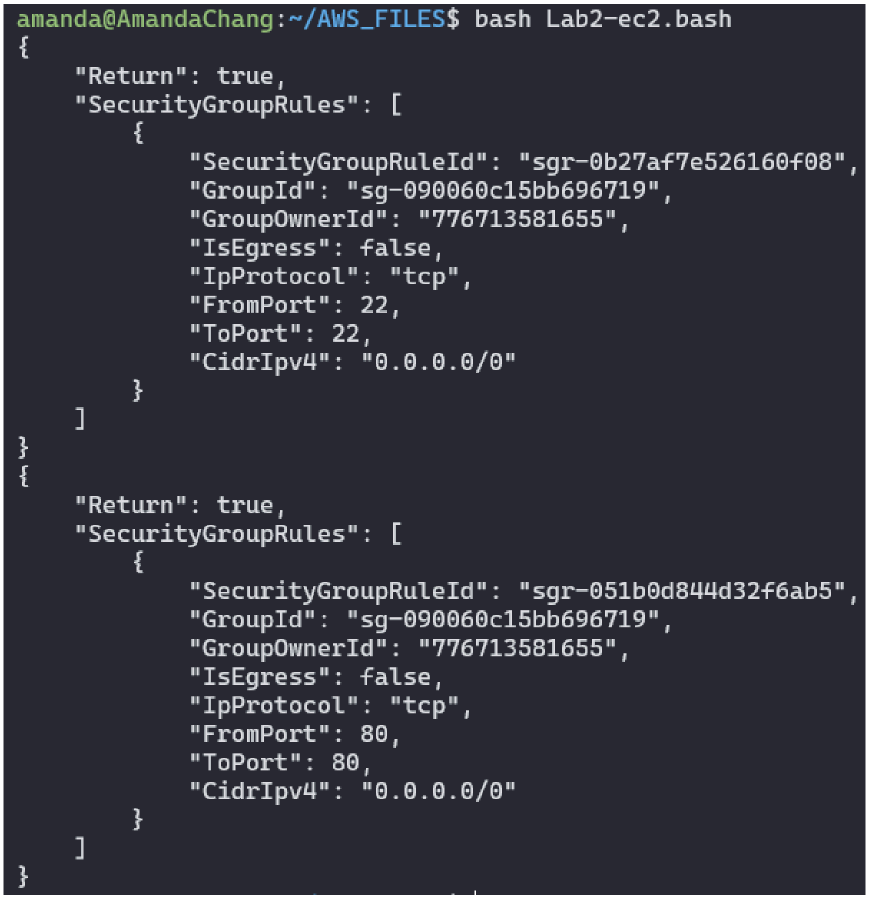


```bash
#!/usr/bin/env bash
# Team Members: Amanda Chang (A01294905), Jace Kang (A00924648), Simon Freeman (A01074210)

set -eu

source infrastructure_data

region="us-west-2"
key_name="MyKeyPair" #change this to the name of your key


# Get Ubuntu 23.04 image id owned by amazon
ubuntu_ami=$(aws ec2 describe-images --region $region \
 --owners amazon \
 --filters Name=name,Values=ubuntu/images/hvm-ssd/ubuntu-lunar-23.04-amd64-server* \
 --query 'sort_by(Images, &CreationDate)[-1].ImageId' --output text)

instance_type="t2.micro"

# Create security group allowing SSH and HTTP from anywhere
security_group_id=$(aws ec2 create-security-group --group-name MySecurityGroup \
 --description "Allow SSH and HTTP" --vpc-id $vpc_id --query 'GroupId' \
 --region $region \
 --output text)

aws ec2 authorize-security-group-ingress --group-id $security_group_id \
 --protocol tcp --port 22 --cidr 0.0.0.0/0 --region $region

aws ec2 authorize-security-group-ingress --group-id $security_group_id \
 --protocol tcp --port 80 --cidr 0.0.0.0/0 --region $region

# Launch an EC2 instance in the public subnet
# COMPLETE THIS PART
instance_id=$(aws ec2 run-instances \
  --image-id $ubuntu_ami \
  --key-name $key_name \
  --count 1 \
  --instance-type $instance_type \
  --subnet-id $subnet_id \
  --security-group-ids $security_group_id \
  --query 'Instances[0].InstanceId' \
  --output text \
  --associate-public-ip-address)

# wait for ec2 instance to be running
aws ec2 wait instance-running --instance-ids $instance_id

# Get the public IP address of the EC2 instance
# COMPLETE THIS PART
public_ip=$(aws ec2 describe-instances --instance-ids "$instance_id" --query 'Reservations[0].Instances[0].PublicIpAddress')

# Write instance data to a file
# COMPLETE THIS PART
echo "Public IP: $public_ip" > instance_data
```

Amanda Chang
A01294905

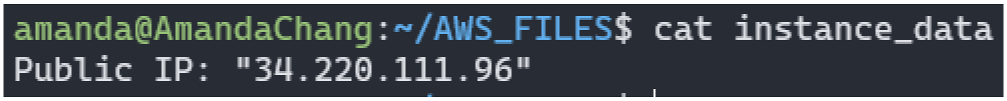
Execute the file by using the following command:

```
> bash Lab2-ec2.bash
```



```
amanda@AmandaChang:~/AWS_FILES$ bash Lab2-ec2.bash
{
    "Return": true,
    "SecurityGroupRules": [
        {
            "SecurityGroupRuleId": "sgr-0b27af7e526160f08",
            "GroupId": "sg-090060c15bb696719",
            "GroupOwnerId": "776713581655",
            "IsEgress": false,
            "IpProtocol": "tcp",
            "FromPort": 22,
            "ToPort": 22,
            "CidrIpv4": "0.0.0.0/0"
        }
    ]
}
{
    "Return": true,
    "SecurityGroupRules": [
        {
            "SecurityGroupRuleId": "sgr-051b0d844d32f6ab5",
            "GroupId": "sg-090060c15bb696719",
            "GroupOwnerId": "776713581655",
            "IsEgress": false,
            "IpProtocol": "tcp",
            "FromPort": 80,
            "ToPort": 80,
            "CidrIpv4": "0.0.0.0/0"
        }
    ]
}
```

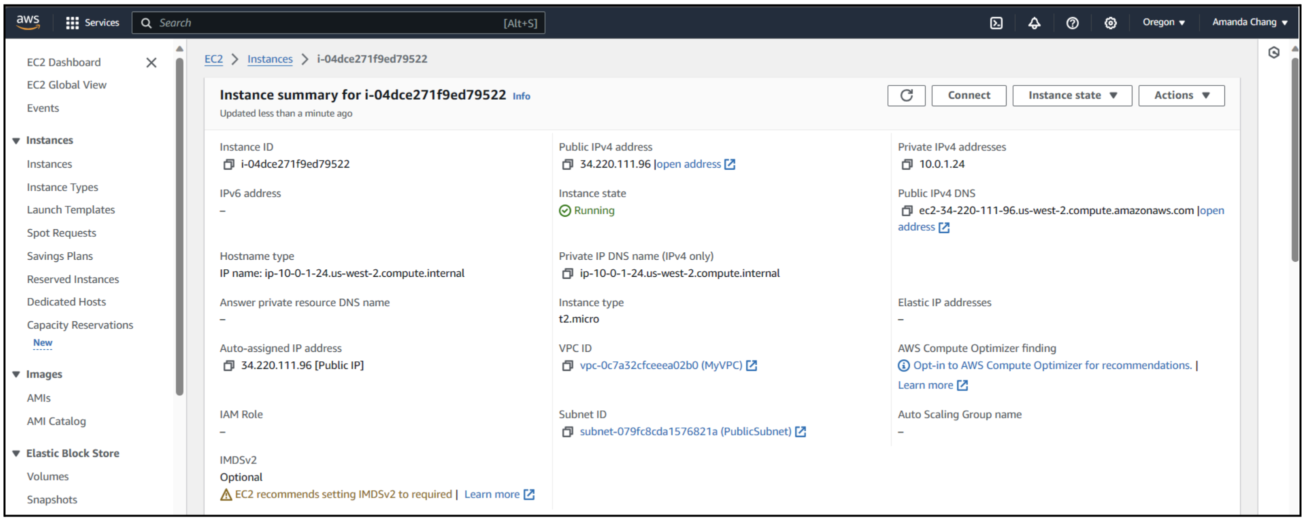
Output file that contains instance's public IP:



```
amanda@AmandaChang:~/AWS_FILES$ cat instance_data
Public IP: "34.220.111.96"
```

Amanda Chang
A01294905

Go to EC2 in AWS Console:



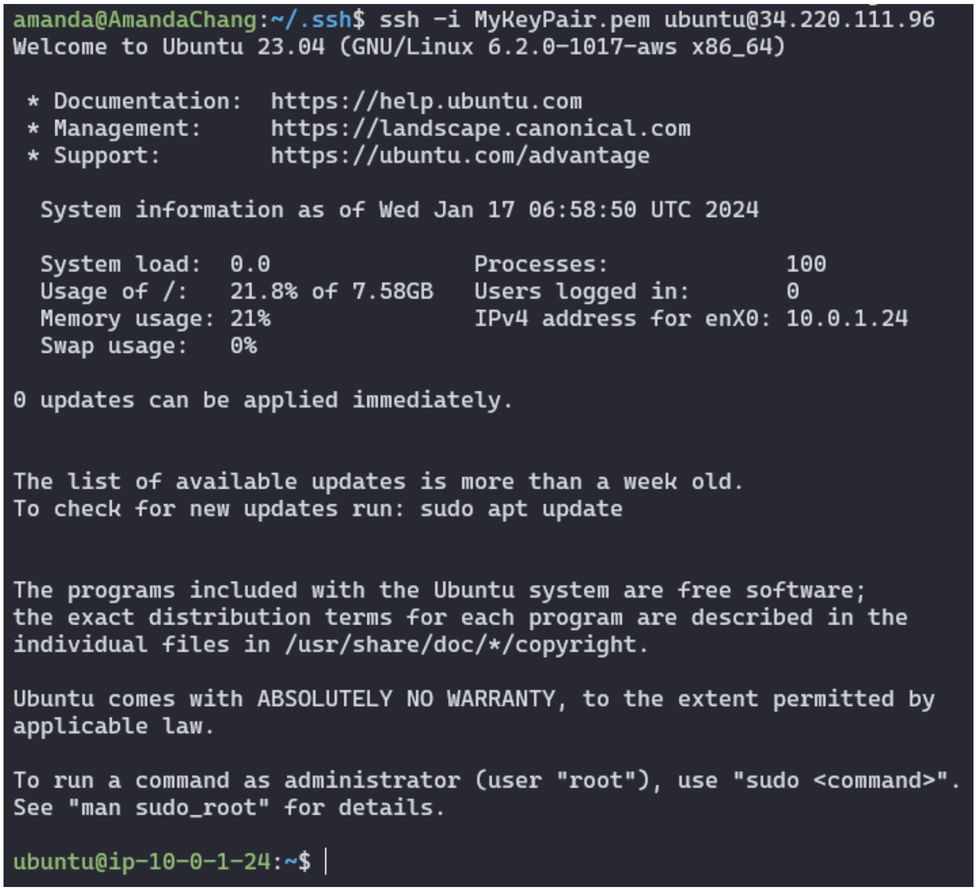
Instance summary for i-04dce271f9ed79522
Updated less than a minute ago

Instance ID: i-04dce271f9ed79522
Public IPv4 address: 34.220.111.96 |open address
Private IPv4 addresses: 10.0.1.24
IPv6 address: –
Instance state: Running
Public IPv4 DNS: ec2-34-220-111-96.us-west-2.compute.amazonaws.com |open address
Hostname type: IP name: ip-10-0-1-24.us-west-2.compute.internal
Private IP DNS name (IPv4 only): ip-10-0-1-24.us-west-2.compute.internal
Answer private resource DNS name: –
Instance type: t2.micro
Elastic IP addresses: –
Auto-assigned IP address: 34.220.111.96 [Public IP]
VPC ID: vpc-0c7a32cfceeea02b0 (MyVPC)
AWS Compute Optimizer finding: Opt-in to AWS Compute Optimizer for recommendations. | Learn more
IAM Role: –
Subnet ID: subnet-079fc8cda1576821a (PublicSubnet)
Auto Scaling Group name: –
IMDSv2: Optional
EC2 recommends setting IMDSv2 to required | Learn more

SSH into the VM using its Public IP:

> `cd /home/amanda/.ssh` ### Where the MyKeyPair.pem is located

> `ssh -i MyKeyPair.pem ubuntu@34.220.111.96`



```
amanda@AmandaChang:~/.ssh$ ssh -i MyKeyPair.pem ubuntu@34.220.111.96
Welcome to Ubuntu 23.04 (GNU/Linux 6.2.0-1017-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  System information as of Wed Jan 17 06:58:50 UTC 2024

  System load:  0.0               Processes:             100
  Usage of /:   21.8% of 7.58GB   Users logged in:       0
  Memory usage: 21%               IPv4 address for enX0: 10.0.1.24
  Swap usage:   0%

0 updates can be applied immediately.


The list of available updates is more than a week old.
To check for new updates run: sudo apt update


The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-10-0-1-24:~$
```