1. *Create an S3 bucket for yourself*
   - *Upload files directly to the root of S3 bucket e.g. s3://your-bucket/file.pdf*
   - *Upload files into a folder on your S3 bucket e.g. s3://your-bucket/your-folder/file.pdf*
   - *Sync the files from S3 bucket to your local folder e.g. ~/Downloads/S3/**
   - *Remove all the files in your S3 bucket*
   - *Delete your S3 bucket*

```
LC-MCAIR:simple-S3-cli-exercise laich$ aws s3 mb s3://lcchua-s3-29072024
make_bucket: lcchua-s3-29072024
LC-MCAIR:simple-S3-cli-exercise laich$ ls -l
total 32
-rw-r--r--  1 laich  staff  505 Jul 29 20:46 Dockerfile
-rw-r--r--  1 laich  staff   70 Jul 29 20:46 README.md
-rw-r--r--  1 laich  staff  163 Jul 29 20:46 app.py
-rw-r--r--  1 laich  staff    6 Jul 29 20:46 requirements.txt
drwxr-xr-x  4 laich  staff  128 Jul 29 21:50 samples
LC-MCAIR:simple-S3-cli-exercise laich$ aws s3 cp . s3://lcchua-s3-29072024 --recursive --exclude "samples/*"
upload: ./README.md to s3://lcchua-s3-29072024/README.md
upload: ./requirements.txt to s3://lcchua-s3-29072024/requirements.txt
upload: ./app.py to s3://lcchua-s3-29072024/app.py
upload: ./Dockerfile to s3://lcchua-s3-29072024/Dockerfile
LC-MCAIR:simple-S3-cli-exercise laich$ aws s3 ls s3://lcchua-s3-29072024
2024-07-29 21:59:54        505 Dockerfile
2024-07-29 21:59:51         70 README.md
2024-07-29 21:59:51        163 app.py
2024-07-29 21:59:51          6 requirements.txt
LC-MCAIR:simple-S3-cli-exercise laich$ aws s3 cp ./samples/ s3://lcchua-s3-29072024 --recursive --dryrun
(dryrun) upload: samples/happy.txt to s3://lcchua-s3-29072024/happy.txt
(dryrun) upload: samples/hello.txt to s3://lcchua-s3-29072024/hello.txt
LC-MCAIR:simple-S3-cli-exercise laich$ aws s3 ls s3://lcchua-s3-29072024
2024-07-29 21:59:54        505 Dockerfile
2024-07-29 21:59:51         70 README.md
2024-07-29 21:59:51        163 app.py
2024-07-29 21:59:51          6 requirements.txt
LC-MCAIR:simple-S3-cli-exercise laich$ aws s3 cp ./samples/ s3://lcchua-s3-29072024/samples/ --recursive
upload: samples/happy.txt to s3://lcchua-s3-29072024/samples/happy.txt
upload: samples/hello.txt to s3://lcchua-s3-29072024/samples/hello.txt
LC-MCAIR:simple-S3-cli-exercise laich$ aws s3 ls s3://lcchua-s3-29072024
                           PRE samples/
2024-07-29 21:59:54        505 Dockerfile
2024-07-29 21:59:51         70 README.md
2024-07-29 21:59:51        163 app.py
2024-07-29 21:59:51          6 requirements.txt
LC-MCAIR:simple-S3-cli-exercise laich$ ls -l
total 32
-rw-r--r--  1 laich  staff  505 Jul 29 20:46 Dockerfile
-rw-r--r--  1 laich  staff   70 Jul 29 20:46 README.md
-rw-r--r--  1 laich  staff  163 Jul 29 20:46 app.py
-rw-r--r--  1 laich  staff    6 Jul 29 20:46 requirements.txt
drwxr-xr-x  2 laich  staff   64 Jul 29 22:06 s3-downloads
drwxr-xr-x  4 laich  staff  128 Jul 29 21:50 samples
LC-MCAIR:simple-S3-cli-exercise laich$ aws s3 sync s3://lcchua-s3-29072024 ./s3-downloads
download: s3://lcchua-s3-29072024/samples/hello.txt to s3-downloads/samples/hello.txt
download: s3://lcchua-s3-29072024/app.py to s3-downloads/app.py
download: s3://lcchua-s3-29072024/samples/happy.txt to s3-downloads/samples/happy.txt
download: s3://lcchua-s3-29072024/README.md to s3-downloads/README.md
download: s3://lcchua-s3-29072024/requirements.txt to s3-downloads/requirements.txt
download: s3://lcchua-s3-29072024/Dockerfile to s3-downloads/Dockerfile
LC-MCAIR:simple-S3-cli-exercise laich$ ls -x
Dockerfile              README.md              app.py              requirements.txt        s3-downloads
samples
LC-MCAIR:simple-S3-cli-exercise laich$ ls -l s3-downloads
total 32
-rw-r--r--  1 laich  staff  505 Jul 29 21:59 Dockerfile
-rw-r--r--  1 laich  staff   70 Jul 29 21:59 README.md
-rw-r--r--  1 laich  staff  163 Jul 29 21:59 app.py
-rw-r--r--  1 laich  staff    6 Jul 29 21:59 requirements.txt
drwxr-xr-x  4 laich  staff  128 Jul 29 22:06 samples
LC-MCAIR:simple-S3-cli-exercise laich$
```

```
LC-MCAIR:simple-S3-cli-exercise laich$ aws s3 rm s3://lcchua-s3-29072024 --recursive
delete: s3://lcchua-s3-29072024/samples/hello.txt
delete: s3://lcchua-s3-29072024/app.py
delete: s3://lcchua-s3-29072024/requirements.txt
delete: s3://lcchua-s3-29072024/samples/happy.txt
delete: s3://lcchua-s3-29072024/Dockerfile
delete: s3://lcchua-s3-29072024/README.md
LC-MCAIR:simple-S3-cli-exercise laich$ aws s3 ls s3://lcchua-s3-29072024
LC-MCAIR:simple-S3-cli-exercise laich$ aws s3 rb s3://lcchua-s3-29072024
remove_bucket: lcchua-s3-29072024
LC-MCAIR:simple-S3-cli-exercise laich$ aws s3 ls s3://lcchua-s3-29072024

An error occurred (NoSuchBucket) when calling the ListObjectsV2 operation: The specified bucket does not exist
LC-MCAIR:simple-S3-cli-exercise laich$
```
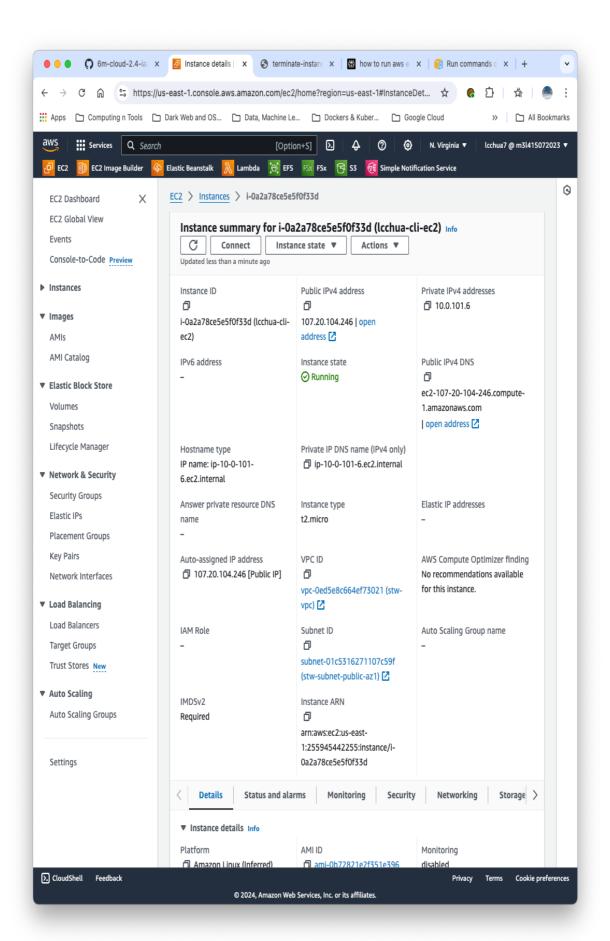
4. *Create an EC2 VM with the following:*
   - *Correct AMI ID,*
   - *Your own Subnet ID,*
   - *Your own Security Group ID,*
   - *Associate public IP enabled,*
   - *Your own key pair,*
   - *Your own name for the EC2*

```
aws ec2 run-instances \
        --image-id ami-0b72821e2f351e396 \
        --count 1 \
        --instance-type t2.micro \
        --key-name lcchua-useast1-20072024 \
        --security-group-ids sg-00c9d54ccab40e186 \
        --associate-public-ip-address \
        --subnet-id subnet-01c5316271107c59f \
        --tag-specifications 'ResourceType=instance,Tags=[{Key=Name,Value='"$TAG_Name"'}]'
LC-MCAIR:simple-ec2-cli-exercise laich$
LC-MCAIR:simple-ec2-cli-exercise laich$ ./create_ec2.sh
{
    "Groups": [],
    "Instances": [
        {
            "AmiLaunchIndex": 0,
            "ImageId": "ami-0b72821e2f351e396",
            "InstanceId": "i-0a2a78ce5e5f0f33d",
            "InstanceType": "t2.micro",
            "KeyName": "lcchua-useast1-20072024",
            "LaunchTime": "2024-07-31T03:44:22+00:00",
            "Monitoring": {
                "State": "disabled"
            },
            "Placement": {
                "AvailabilityZone": "us-east-1d",
                "GroupName": "",
                "Tenancy": "default"
            },
            "PrivateDnsName": "ip-10-0-101-6.ec2.internal",
            "PrivateIpAddress": "10.0.101.6",
            "ProductCodes": [],
            "PublicDnsName": "",
            "State": {
                "Code": 0,
                "Name": "pending"
            },
            "StateTransitionReason": "",
            "SubnetId": "subnet-01c5316271107c59f",
            "VpcId": "vpc-0ed5e8c664ef73021",
            "Architecture": "x86_64",
            "BlockDeviceMappings": [],
            "ClientToken": "270ed868-1929-4806-9f04-a18a373811b2",
            "EbsOptimized": false,
            "EnaSupport": true,
            "Hypervisor": "xen",
            "NetworkInterfaces": [
                {
                    "Attachment": {
                        "AttachTime": "2024-07-31T03:44:22+00:00",
                        "AttachmentId": "eni-attach-0dc00f67c5195b5b4",
                        "DeleteOnTermination": true,
                        "DeviceIndex": 0,
                        "Status": "attaching",
```

```
                "MacAddress": "0e:96:52:49:2e:5d",
                "NetworkInterfaceId": "eni-0a13bd284088aa8e7",
                "OwnerId": "255945442255",
                "PrivateDnsName": "ip-10-0-101-6.ec2.internal",
                "PrivateIpAddress": "10.0.101.6",
                "PrivateIpAddresses": [
                    {
                        "Primary": true,
                        "PrivateDnsName": "ip-10-0-101-6.ec2.internal",
                        "PrivateIpAddress": "10.0.101.6"
                    }
                ],
                "SourceDestCheck": true,
                "Status": "in-use",
                "SubnetId": "subnet-01c5316271107c59f",
                "VpcId": "vpc-0ed5e8c664ef73021",
                "InterfaceType": "interface"
            }
        ],
        "RootDeviceName": "/dev/xvda",
        "RootDeviceType": "ebs",
        "SecurityGroups": [
            {
                "GroupName": "lcchua-tf-sg-allow-ssh-http-https",
                "GroupId": "sg-00c9d54ccab40e186"
            }
        ],
        "SourceDestCheck": true,
        "StateReason": {
            "Code": "pending",
            "Message": "pending"
        },
        "Tags": [
            {
                "Key": "Name",
                "Value": "lcchua-cli-ec2"
            }
        ],
        "VirtualizationType": "hvm",
        "CpuOptions": {
            "CoreCount": 1,
            "ThreadsPerCore": 1
        },
        "CapacityReservationSpecification": {
            "CapacityReservationPreference": "open"
        },
        "MetadataOptions": {
            "State": "pending",
            "HttpTokens": "required",
            "HttpPutResponseHopLimit": 2,
            "HttpEndpoint": "enabled",
            "HttpProtocolIpv6": "disabled",
            "InstanceMetadataTags": "disabled"
```
:

```
        "RootDeviceName": "/dev/xvda",
        "RootDeviceType": "ebs",
        "SecurityGroups": [
            {
                "GroupName": "lcchua-tf-sg-allow-ssh-http-https",
                "GroupId": "sg-00c9d54ccab40e186"
            }
        ],
        "SourceDestCheck": true,
        "StateReason": {
            "Code": "pending",
            "Message": "pending"
        },
        "Tags": [
            {
                "Key": "Name",
                "Value": "lcchua-cli-ec2"
            }
        ],
        "VirtualizationType": "hvm",
        "CpuOptions": {
            "CoreCount": 1,
            "ThreadsPerCore": 1
        },
        "CapacityReservationSpecification": {
            "CapacityReservationPreference": "open"
        },
        "MetadataOptions": {
            "State": "pending",
            "HttpTokens": "required",
            "HttpPutResponseHopLimit": 2,
            "HttpEndpoint": "enabled",
            "HttpProtocolIpv6": "disabled",
            "InstanceMetadataTags": "disabled"
        },
        "EnclaveOptions": {
            "Enabled": false
        },
        "BootMode": "uefi-preferred",
        "PrivateDnsNameOptions": {
            "HostnameType": "ip-name",
            "EnableResourceNameDnsARecord": false,
            "EnableResourceNameDnsAAAARecord": false
        },
        "MaintenanceOptions": {
            "AutoRecovery": "default"
        },
        "CurrentInstanceBootMode": "legacy-bios"
      }
    ],
    "OwnerId": "255945442255",
    "ReservationId": "r-0f452389781b37186"
}
(END)
```
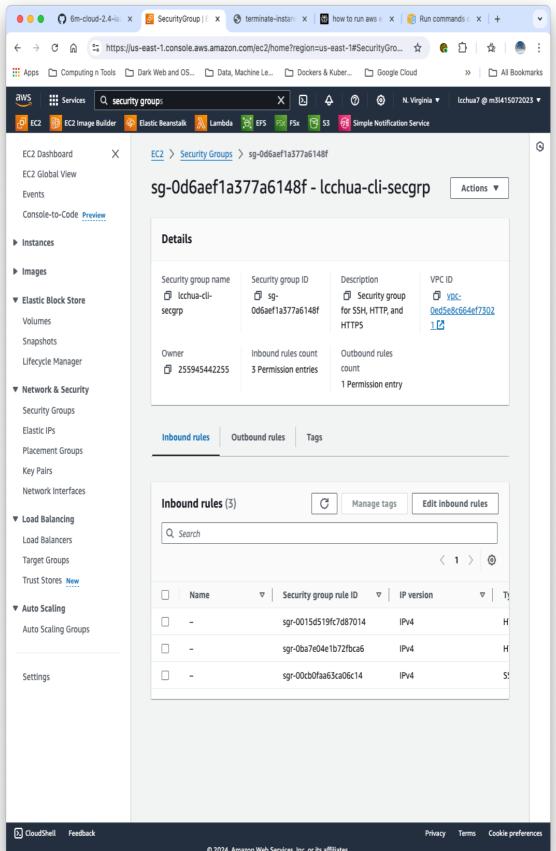
https://us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#InstanceDet...

Apps   Computing n Tools   Dark Web and OS...   Data, Machine Le...   Dockers & Kuber...   Google Cloud   »   All Bookmarks

aws   Services   Q Search   [Option+S]   N. Virginia ▼   lcchua7 @ m3l415072023 ▼

EC2   EC2 Image Builder   Elastic Beanstalk   Lambda   EFS   FSx   S3   Simple Notification Service

EC2 Dashboard	✕
EC2 Global View	
Events	
Console-to-Code	Preview

EC2 > Instances > i-0a2a78ce5f0f33d

Instance summary for i-0a2a78ce5f0f33d (lcchua-cli-ec2) Info

⟳   Connect   Instance state ▼   Actions ▼

Updated less than a minute ago

▶ Instances

▼ Images

   AMIs
   AMI Catalog

▼ Elastic Block Store

   Volumes
   Snapshots
   Lifecycle Manager

▼ Network & Security

   Security Groups
   Elastic IPs
   Placement Groups
   Key Pairs
   Network Interfaces

▼ Load Balancing

   Load Balancers
   Target Groups
   Trust Stores New

▼ Auto Scaling

   Auto Scaling Groups

   Settings

Instance ID	Public IPv4 address	Private IPv4 addresses
i-0a2a78ce5f0f33d (lcchua-cli-ec2)	107.20.104.246	open address ↗	10.0.101.6
IPv6 address	Instance state	Public IPv4 DNS
–	⊘ Running	ec2-107-20-104-246.compute-1.amazonaws.com | open address ↗
Hostname type	Private IP DNS name (IPv4 only)	
IP name: ip-10-0-101-6.ec2.internal	ip-10-0-101-6.ec2.internal	
Answer private resource DNS name	Instance type	Elastic IP addresses
–	t2.micro	–
Auto-assigned IP address	VPC ID	AWS Compute Optimizer finding
107.20.104.246 [Public IP]	vpc-0ed5e8c664ef73021 (stw-vpc) ↗	No recommendations available for this instance.
IAM Role	Subnet ID	Auto Scaling Group name
–	subnet-01c5316271107c59f (stw-subnet-public-az1) ↗	–
IMDSv2	Instance ARN	
Required	arn:aws:ec2:us-east-1:255945442255:instance/i-0a2a78ce5f0f33d	

◀ ▶   Details   Status and alarms   Monitoring   Security   Networking   Storage ▶

▼ Instance details Info

Platform	AMI ID	Monitoring
Amazon Linux (Inferred)	ami-0b72821e2f351e396	disabled

CloudShell   Feedback   Privacy   Terms   Cookie preferences

2. List down all available key pairs in your AWS account

```
LC-MCAIR:simple-ec2-cli-exercise laich$ aws ec2 describe-key-pairs --filters Name="key-name",Values="lcchua-useast1
-*"
{
    "KeyPairs": [
        {
            "KeyPairId": "key-02a8c35d12fef31d7",
            "KeyFingerprint": "f5:48:c9:e5:d6:25:1b:58:9b:3d:e4:e8:49:ab:7b:2a",
            "KeyName": "lcchua-useast1-30072024",
            "KeyType": "rsa",
            "Tags": [
                {
                    "Key": "Name",
                    "Value": "stw-key-pair"
                },
                {
                    "Key": "group",
                    "Value": "lcchua-STW"
                }
            ],
            "CreateTime": "2024-07-30T06:19:51.532000+00:00"
        },
        {
            "KeyPairId": "key-0396e45e6bdcf9a33",
            "KeyFingerprint": "p7xdwrxvwI3NEW9hEo3bgDeLVtBhIn5xoYU2CACM5Z8=",
            "KeyName": "lcchua-useast1-20072024",
            "KeyType": "ed25519",
            "Tags": [],
            "CreateTime": "2024-07-20T02:52:22.137000+00:00"
        }
    ]
}
LC-MCAIR:simple-ec2-cli-exercise laich$
```

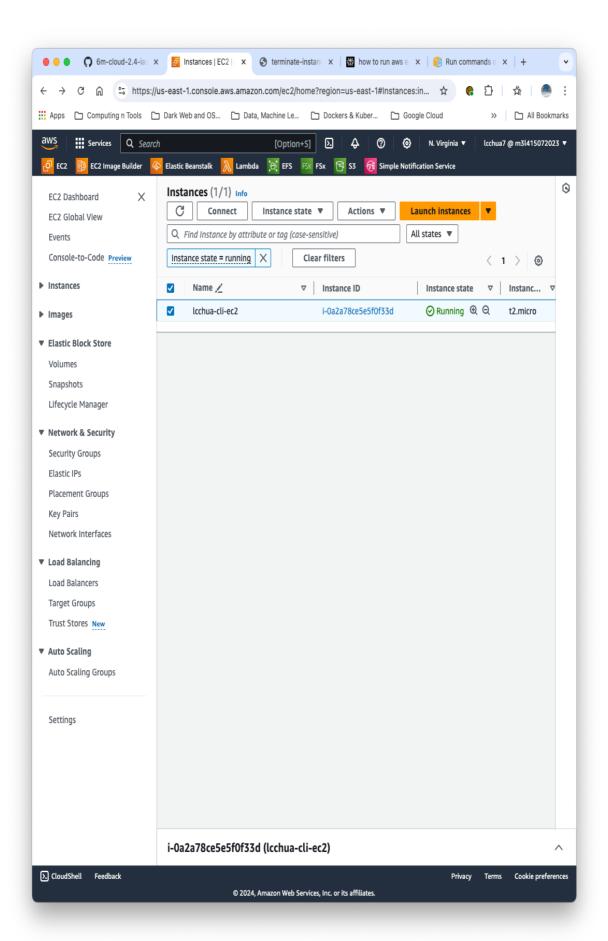3. *Create a security group that allows SSH, HTTP and HTTPS from anywhere*

```
aws ec2 authorize-security-group-ingress --group-id $GROUPID \
        --protocol tcp --port 22 --cidr 0.0.0.0/0
aws ec2 authorize-security-group-ingress --group-id $GROUPID \
        --protocol tcp --port 80 --cidr 0.0.0.0/0
aws ec2 authorize-security-group-ingress --group-id $GROUPID \
        --protocol tcp --port 443 --cidr 0.0.0.0/0
LC-MCAIR:simple-ec2-cli-exercise laich$ ./create_secgrp.sh
{
    "Return": true,
    "SecurityGroupRules": [
        {
            "SecurityGroupRuleId": "sgr-00cb0faa63ca06c14",
            "GroupId": "sg-0d6aef1a377a6148f",
            "GroupOwnerId": "255945442255",
            "IsEgress": false,
            "IpProtocol": "tcp",
            "FromPort": 22,
            "ToPort": 22,
            "CidrIpv4": "0.0.0.0/0"
        }
    ]
}
{
    "Return": true,
    "SecurityGroupRules": [
        {
            "SecurityGroupRuleId": "sgr-0ba7e04e1b72fbca6",
            "GroupId": "sg-0d6aef1a377a6148f",
            "GroupOwnerId": "255945442255",
            "IsEgress": false,
            "IpProtocol": "tcp",
            "FromPort": 80,
            "ToPort": 80,
            "CidrIpv4": "0.0.0.0/0"
        }
    ]
}
{
    "Return": true,
    "SecurityGroupRules": [
        {
            "SecurityGroupRuleId": "sgr-0015d519fc7d87014",
            "GroupId": "sg-0d6aef1a377a6148f",
            "GroupOwnerId": "255945442255",
            "IsEgress": false,
            "IpProtocol": "tcp",
            "FromPort": 443,
            "ToPort": 443,
            "CidrIpv4": "0.0.0.0/0"
        }
    ]
}
LC-MCAIR:simple-ec2-cli-exercise laich$
```

5. *Describe more information regarding your newly-created EC2 instance*

```
LC-MCAIR:simple-ec2-cli-exercise laich$ cat describe_ec2.sh
#!/bin/bash
aws ec2 describe-instances --instance-ids $1
//aws ec2 describe-instances --instance-ids $1 \
//      --query 'Reservations[*].Instances[*].{ID:InstanceId,Type:InstanceType,State:State.Name,PublicIP:PublicIpAd
dress,PrivateIP:PrivateIpAddress,LaunchTime:LaunchTime}' \
//      --output table
LC-MCAIR:simple-ec2-cli-exercise laich$ ./describe_ec2.sh i-0a2a78ce5e5f0f33d
{
    "Reservations": [
        {
            "Groups": [],
            "Instances": [
                {
                    "AmiLaunchIndex": 0,
                    "ImageId": "ami-0b72821e2f351e396",
                    "InstanceId": "i-0a2a78ce5e5f0f33d",
                    "InstanceType": "t2.micro",
                    "KeyName": "lcchua-useast1-20072024",
                    "LaunchTime": "2024-07-31T03:44:22+00:00",
                    "Monitoring": {
                        "State": "disabled"
                    },
                    "Placement": {
                        "AvailabilityZone": "us-east-1d",
                        "GroupName": "",
                        "Tenancy": "default"
                    },
                    "PrivateDnsName": "ip-10-0-101-6.ec2.internal",
                    "PrivateIpAddress": "10.0.101.6",
                    "ProductCodes": [],
                    "PublicDnsName": "ec2-107-20-104-246.compute-1.amazonaws.com",
                    "PublicIpAddress": "107.20.104.246",
                    "State": {
                        "Code": 16,
                        "Name": "running"
                    },
                    "StateTransitionReason": "",
                    "SubnetId": "subnet-01c5316271107c59f",
                    "VpcId": "vpc-0ed5e8c664ef73021",
                    "Architecture": "x86_64",
                    "BlockDeviceMappings": [
                        {
                            "DeviceName": "/dev/xvda",
                            "Ebs": {
                                "AttachTime": "2024-07-31T03:44:22+00:00",
                                "DeleteOnTermination": true,
                                "Status": "attached",
                                "VolumeId": "vol-087c8f4e87be3938e"
                            }
                        }
                    ],
                    "ClientToken": "270ed868-1929-4806-9f04-a18a373811b2",
                    "EbsOptimized": false,
```

```
"Reservations": [
    {
        "Groups": [],
        "Instances": [
            {
                "AmiLaunchIndex": 0,
                "ImageId": "ami-0b72821e2f351e396",
                "InstanceId": "i-0a2a78ce5e5f0f33d",
                "InstanceType": "t2.micro",
                "KeyName": "lcchua-useast1-20072024",
                "LaunchTime": "2024-07-31T03:44:22+00:00",
                "Monitoring": {
                    "State": "disabled"
                },
                "Placement": {
                    "AvailabilityZone": "us-east-1d",
                    "GroupName": "",
                    "Tenancy": "default"
                },
                "PrivateDnsName": "ip-10-0-101-6.ec2.internal",
                "PrivateIpAddress": "10.0.101.6",
                "ProductCodes": [],
                "PublicDnsName": "ec2-107-20-104-246.compute-1.amazonaws.com",
                "PublicIpAddress": "107.20.104.246",
                "State": {
                    "Code": 16,
                    "Name": "running"
                },
                "StateTransitionReason": "",
                "SubnetId": "subnet-01c5316271107c59f",
                "VpcId": "vpc-0ed5e8c664ef73021",
                "Architecture": "x86_64",
                "BlockDeviceMappings": [
                    {
                        "DeviceName": "/dev/xvda",
                        "Ebs": {
                            "AttachTime": "2024-07-31T03:44:22+00:00",
                            "DeleteOnTermination": true,
                            "Status": "attached",
                            "VolumeId": "vol-087c8f4e87be3938e"
                        }
                    }
                ],
                "ClientToken": "270ed868-1929-4806-9f04-a18a373811b2",
                "EbsOptimized": false,
                "EnaSupport": true,
                "Hypervisor": "xen",
                "NetworkInterfaces": [
                    {
                        "Association": {
                            "IpOwnerId": "amazon",
                            "PublicDnsName": "ec2-107-20-104-246.compute-1.amazonaws.com",
```

```
./describe_ec2.sh: line 3: //aws: No such file or directory
LC-MCAIR:simple-ec2-cli-exercise laich$
```

5. *Terminate your EC2 instance*

```
LC-MCAIR:simple-ec2-cli-exercise laich$ cat terminate_ec2.sh
#!/bin/bash
# Instance Id = $1
# Dry-run | No-dry-run = $2
aws ec2 terminate-instances --instance-ids $1 --$2
LC-MCAIR:simple-ec2-cli-exercise laich$ ./terminate_ec2.sh i-0a2a78ce5e5f0f33d no-dry-run
{
    "TerminatingInstances": [
        {
            "CurrentState": {
                "Code": 32,
                "Name": "shutting-down"
            },
            "InstanceId": "i-0a2a78ce5e5f0f33d",
            "PreviousState": {
                "Code": 16,
                "Name": "running"
            }
        }
    ]
}
LC-MCAIR:simple-ec2-cli-exercise laich$
```