Block Storage

1. Download and install the CLI of your cloud provider.
2. Use the CLI to create a public subnet in a VPC of your choice.
   1. Creating VPC

File : vpc-tags.json

A screen shot of a computer program

Description automatically generated

|  |
| --- |
| $vpcId = (aws ec2 `  create-vpc `  --profile aws\_admin\_user1 `  --no-paginate `  --no-cli-pager `  --cidr-block 10.0.0.0/16 `  --amazon-provided-ipv6-cidr-block `  --tag-specifications file://vpc-tags.json `  --output json | ConvertFrom-Json).Vpc.VpcId  $vpcId  aws ec2 describe-vpcs `  --profile aws\_admin\_user1 `  --no-paginate `  --no-cli-pager `  --vpc-ids $vpcId |

A screenshot of a computer

Description automatically generated

* 1. Creating Internet Gateway –

|  |
| --- |
| $internetGateway = aws ec2 `  create-internet-gateway `  --profile aws\_admin\_user1 `  --query 'InternetGateway.InternetGatewayId' `  --output text `  --no-paginate `  --no-cli-pager  $internetGateway  aws ec2 attach-internet-gateway `  --profile aws\_admin\_user1 `  --internet-gateway-id $internetGateway `  --vpc-id $vpcId `  --no-cli-pager  aws ec2 `  describe-internet-gateways `  --profile aws\_admin\_user1 `  --no-paginate `  --no-cli-pager `  --internet-gateway-ids $internetGateway |

A screen shot of a computer

Description automatically generated

* 1. Creating Public Subnet –

File – subnet-tags.json

A screen shot of a computer program

Description automatically generated

|  |
| --- |
| $subnetId = aws ec2 `  create-subnet `  --profile aws\_admin\_user1 `  --vpc-id $vpcId `  --cidr-block 10.0.1.0/24 `  --availability-zone us-east-1a `  --query 'Subnet.SubnetId' `  --output text `  --no-cli-pager `  --tag-specifications `  file://subnet-tags.json  $subnetId  aws ec2 `  modify-subnet-attribute `  --profile aws\_admin\_user1 `  --subnet-id $subnetId `  --map-public-ip-on-launch `  --no-cli-pager  aws ec2 `  describe-subnets `  --profile aws\_admin\_user1 `  --no-paginate `  --no-cli-pager `  --subnet-ids $subnetId |

A screenshot of a computer program

Description automatically generated

* 1. Creating Routing table-

File - routetable-tags.json

A screen shot of a computer program

Description automatically generated

|  |
| --- |
| $routeTableId = aws ec2 `  create-route-table `  --profile aws\_admin\_user1 `  --vpc-id $vpcId `  --query 'RouteTable.RouteTableId' `  --output text `  --no-cli-pager `  --tag-specifications file://routetable-tags.json  $routeTableId  aws ec2 `  describe-route-tables `  --profile aws\_admin\_user1 `  --route-table-ids $routeTableId `  --output table `  --no-cli-pager `  --no-paginate |

A screenshot of a computer screen

Description automatically generated

* 1. Create route in this route table and attach route table to the subnet-

|  |
| --- |
| aws ec2 create-route `  --profile aws\_admin\_user1 `  --route-table-id $routeTableId `  --destination-cidr-block 0.0.0.0/0 `  --gateway-id $internetGateway `  --no-cli-pager  aws ec2 associate-route-table `  --profile aws\_admin\_user1 `  --route-table-id $routeTableId `  --subnet-id $subnetId `  --no-cli-pager |

A screen shot of a computer

Description automatically generated

1. Use the CLI to create a Virtual Machine (Compute/EC2 instance) with a public IP address.

File : instance-tags.json

A screen shot of a computer program

Description automatically generated

|  |
| --- |
| $instanceId = aws ec2 `  run-instances `  --profile aws\_admin\_user1 `  --image-id ami-0bb84b8ffd87024d8 `  --count 1 `  --instance-type t2.micro `  --key-name pem-key-pair `  --subnet-id $subnetId `  --associate-public-ip-address `  --output text `  --query 'Instances[0].InstanceId' `  --no-cli-pager `  --no-paginate `  --tag-specifications file://instance-tags.json  $instanceId  aws ec2 `  describe-instances `  --profile aws\_admin\_user1 `  --instance-ids $instanceId `  --no-cli-pager `  --no-paginate `  --output table |

A screenshot of a computer

Description automatically generated

1. Check the storage types that are available for creating a block volume.

A screenshot of a computer

Description automatically generated

1. Use the CLI to create a 10 GB block storage volume of type Standard (Magnetic).

File - volume-tags.json

A screen shot of a computer

Description automatically generated

|  |
| --- |
| $volumeId = aws ec2 `  create-volume `  --profile aws\_admin\_user1 `  --availability-zone us-east-1a `  --size 10 `  --volume-type standard `  --query 'VolumeId' `  --tag-specifications file://volume-tags.json `  --output text `  --no-cli-pager `  --no-paginate  $volumeId  aws ec2 `  describe-volumes `  --profile aws\_admin\_user1 `  --volume-ids $volumeId `  --no-cli-pager `  --no-paginate `  --output table |

A screenshot of a computer program

Description automatically generated

1. Use the CLI to attach this storage volume to the compute instance that you just created:

|  |
| --- |
| aws ec2 attach-volume `  --profile aws\_admin\_user1 `  --volume-id $volumeId `  --instance-id $instanceId `  --device /dev/sdh `  --no-cli-pager |

A screen shot of a computer

Description automatically generated

1. Use the CLI to modify the volume to 25 GB or higher of type SSD.

|  |
| --- |
| aws ec2 modify-volume `  --profile aws\_admin\_user1 `  --volume-id $volumeId `  --size 25 `  --volume-type gp2 `  --no-cli-pager |

A screenshot of a computer program

Description automatically generated

1. Use the CLI to attempt to delete the volume.

|  |
| --- |
| aws ec2 delete-volume `  --profile aws\_admin\_user1 `  --volume-id $volumeId `  --no-paginate `  --no-cli-pager |

A screen shot of a computer

Description automatically generated

1. Use the CLI to delete the compute instance and any block storage volumes that was attached to it.

|  |
| --- |
| aws ec2 `  terminate-instances `  --profile aws\_admin\_user1 `  --instance-ids $instanceId `  --no-paginate `  --no-cli-pager  aws ec2 delete-volume `  --profile aws\_admin\_user1 `  --volume-id $volumeId `  --no-paginate `  --no-cli-pager  aws ec2 `  describe-volumes `  --profile aws\_admin\_user1 `  --volume-ids $volumeId `  --no-cli-pager `  --no-paginate `  --output table |

A screenshot of a computer program

Description automatically generated

A screenshot of a computer screen

Description automatically generated

1. Take screenshots of each step and paste them into a Microsoft Word document entitled Storage-Lab.doc.

Object Storage

1. Use the CLI to create an IAM policy that gives a principal (say an IAM Role) the permissions to
   1. create, list, view and delete object storage buckets
   2. put, get, list and delete objects in any bucket

|  |
| --- |
| $policyId = aws iam `  create-policy `  --profile aws\_admin\_user1 `  --policy-name my-policy `  --policy-document file://s3-policy.json `  --output text `  --query 'Policy.Arn' `  --no-cli-pager  $policyId  aws iam `  get-policy `  --profile aws\_admin\_user1 `  --policy-arn $policyId `  --output table `  --no-paginate `  --no-cli-pager |

A computer screen shot of a computer program

Description automatically generated

1. Use the CLI to create an object storage bucket

|  |
| --- |
| $bucketName = "my-lab6-object-storage-bucket"  aws s3api `  create-bucket `  --profile aws\_admin\_user1 `  --bucket $bucketName `  --no-paginate `  --output table `  --region us-east-1 `  --no-cli-pager |

A screenshot of a computer program

Description automatically generated

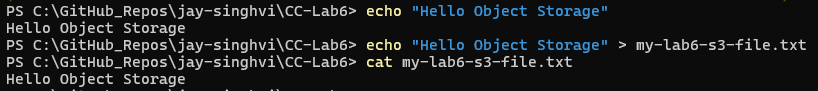
1. Use the CLI to list the buckets.

|  |
| --- |
| aws s3api `  list-buckets `  --profile aws\_admin\_user1 `  --no-paginate `  --output table `  --region us-east-1 `  --no-cli-pager |

A screenshot of a computer program

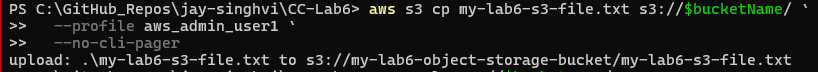
Description automatically generated

1. Create a text file containing the text "Hello Object Storage".



1. Use the CLI to copy the text file to the bucket.

|  |
| --- |
| aws s3 cp my-lab6-s3-file.txt s3://$bucketName/ `  --profile aws\_admin\_user1 `  --no-cli-pager |



1. Use the CLI to list the objects in the bucket.

|  |
| --- |
| aws s3 ls s3://$bucketName/ `  --profile aws\_admin\_user1 `  --recursive `  --human-readable `  --no-cli-pager |

A black screen with white text

Description automatically generated

1. Use the CLI to create a compute/EC2 instance.

|  |
| --- |
| $instanceId = aws ec2 `  run-instances `  --profile aws\_admin\_user1 `  --image-id ami-0bb84b8ffd87024d8 `  --count 1 `  --instance-type t2.micro `  --key-name pem-key-pair `  --subnet-id $subnetId `  --associate-public-ip-address `  --output text `  --query 'Instances[0].InstanceId' `  --no-cli-pager `  --no-paginate `  --tag-specifications file://instance-tags.json  $instanceId  aws ec2 `  describe-instances `  --profile aws\_admin\_user1 `  --instance-ids $instanceId `  --no-cli-pager `  --no-paginate `  --output table |

A screenshot of a computer

Description automatically generated

1. Write a simple Python Flask program that uses the Python SDK to read the text file from the bucket.

|  |
| --- |
| from flask import Flask, Response, jsonify  import boto3  from botocore.exceptions import (      NoCredentialsError,      PartialCredentialsError,      TokenRetrievalError,  )  app = Flask(\_\_name\_\_)  def create\_boto3\_session(profile\_name):      try:          session = boto3.Session(profile\_name=profile\_name)          return session      except TokenRetrievalError as e:          print(f"Token retrieval error: {e}")          print(              f"Please run `aws sso login --profile {profile\_name}` to renew your SSO token."          )          return None      except (NoCredentialsError, PartialCredentialsError) as e:          print(f"Error: {e}")          return None  def get\_file\_from\_s3(bucket\_name, file\_name, profile\_name):      session = create\_boto3\_session(profile\_name)      if not session:          return "AWS SSO token has expired or there is an issue with credentials."      s3\_client = session.client("s3", region\_name="us-east-1")      try:          obj = s3\_client.get\_object(Bucket=bucket\_name, Key=file\_name)          return obj["Body"].read().decode("utf-8")      except s3\_client.exceptions.NoSuchKey:          return "The specified key does not exist."      except Exception as e:          return f"Error retrieving file: {e}"  profile\_name = "aws\_admin\_user1"  bucket\_name = "my-lab6-object-storage-bucket"  file\_name = "my-lab6-s3-file.txt"  @app.route("/hello")  def hello():      return get\_file\_from\_s3(bucket\_name, file\_name, profile\_name)  if \_\_name\_\_ == "\_\_main\_\_":      app.run(debug=True) |

A screen shot of a computer

Description automatically generated

1. Return the text in the body of the HTTP response of a curl request to http://<IP>:5000/hello,



A close up of a logo

Description automatically generated

1. Use the CLI to delete the file, the bucket and the EC2 instance.

|  |
| --- |
| aws s3 rm s3://$bucketName/my-lab6-s3-file.txt `  --profile aws\_admin\_user1 `  --no-cli-pager  aws s3 rb s3://$bucketName `  --profile aws\_admin\_user1 `  --force `  --no-cli-pager  aws ec2 terminate-instances `  --profile aws\_admin\_user1 `  --instance-ids $instanceId `  --no-cli-pager |

A screenshot of a computer

Description automatically generated

1. Take screenshots of each step and paste them into Storage-Lab.doc.
2. Submit the document using the File Upload option in this assignment.