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.
    a. Creating VPC

File : vpc-tags.json

vpc-tags.json

```
[
    {
        "ResourceType": "vpc",
        "Tags": [
            {
                "Key": "Name",
                "Value": "my_lab6_vpc"
            },
            {
                "Key": "Environment",
                "Value": "Preprod"
            }
        ]
    }
]
```

```
$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
```

```
PS C:\GitHub_Repos\jay-singhvi\CC-Lab6> $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
PS C:\GitHub_Repos\jay-singhvi\CC-Lab6>
PS C:\GitHub_Repos\jay-singhvi\CC-Lab6> $vpcId
vpc-0838ee0065cbe58b0
PS C:\GitHub_Repos\jay-singhvi\CC-Lab6>
PS C:\GitHub_Repos\jay-singhvi\CC-Lab6> aws ec2 describe-vpcs `
>>   --profile aws_admin_user1 `
>>   --no-paginate `
>>   --no-cli-pager `
>>   --vpc-ids $vpcId
-------------------------------------------------------------
|                        DescribeVpcs                       |
+-----------------------------------------------------------+
||                           Vpcs                          ||
|+-----------------------+---------------------------------+|
||  CidrBlock            |  10.0.0.0/16                    ||
||  DhcpOptionsId        |  dopt-094328b8da4ccebf3         ||
||  InstanceTenancy      |  default                        ||
||  IsDefault            |  False                          ||
||  OwnerId              |  381491908351                   ||
||  State                |  available                      ||
||  VpcId                |  vpc-0838ee0065cbe58b0          ||
|+-----------------------+---------------------------------+|
|||                CidrBlockAssociationSet                ||
||+-----------------+-----------------------------------+  ||
|||  AssociationId  |  vpc-cidr-assoc-0e9d81db0412f8b0d |  ||
|||  CidrBlock      |  10.0.0.0/16                      |  ||
||+-----------------+-----------------------------------+  ||
||||                   CidrBlockState                    |  ||
|||+--------------------+--------------------------------+  ||
||||  State             |  associated                    |  ||
|||+--------------------+--------------------------------+  ||
|||               Ipv6CidrBlockAssociationSet             ||
||+--------------------+-----------------------------------+||
|||  AssociationId     |  vpc-cidr-assoc-05a7ec8677772d9e9 |||
|||  Ipv6CidrBlock     |  2600:1f18:5244:eb00::/56         |||
|||  Ipv6Pool          |  Amazon                           |||
|||  NetworkBorderGroup|  us-east-1                        |||
||+--------------------+-----------------------------------+||
||||                  Ipv6CidrBlockState                   |||
|||+--------------------+--------------------------------+  ||
||||  State             |  associated                    |  ||
|||+--------------------+--------------------------------+  ||
|||                          Tags                           ||
||+---------------------------+-----------------------------+||
|||            Key            |            Value            |||
||+---------------------------+-----------------------------+||
|||  Environment              |  Preprod                    |||
|||  Name                     |  my_lab6_vpc                |||
||+---------------------------+-----------------------------+||
```

b. 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
```

```
PS C:\GitHub_Repos\jay-singhvi\CC-Lab6> $internetGateway = aws ec2 `
>>   create-internet-gateway `
>>   --profile aws_admin_user1 `
>>   --query 'InternetGateway.InternetGatewayId' `
>>   --output text `
>>   --no-paginate `
>>   --no-cli-pager
PS C:\GitHub_Repos\jay-singhvi\CC-Lab6>
PS C:\GitHub_Repos\jay-singhvi\CC-Lab6> $internetGateway
igw-09dceb8a92a7ff5f1
PS C:\GitHub_Repos\jay-singhvi\CC-Lab6>
PS C:\GitHub_Repos\jay-singhvi\CC-Lab6> aws ec2 attach-internet-gateway `
>>   --profile aws_admin_user1 `
>>   --internet-gateway-id $internetGateway `
>>   --vpc-id $vpcId `
>>   --no-cli-pager
PS C:\GitHub_Repos\jay-singhvi\CC-Lab6>
PS C:\GitHub_Repos\jay-singhvi\CC-Lab6> aws ec2 `
>>   describe-internet-gateways `
>>   --profile aws_admin_user1 `
>>   --no-paginate `
>>   --no-cli-pager `
>>   --internet-gateway-ids $internetGateway
-------------------------------------------
|         DescribeInternetGateways          |
+-------------------------------------------+
||            InternetGateways             ||
|+----------------------+----------------+|
||    InternetGatewayId   |    OwnerId     ||
|+----------------------+----------------+|
||  igw-09dceb8a92a7ff5f1 |  381491908351  ||
|+----------------------+----------------+|
|||              Attachments              |||
||+------------+--------------------------+||
|||    State   |          VpcId           |||
||+------------+--------------------------+||
|||  available |  vpc-0838ee0065cbe58b0   |||
||+------------+--------------------------+||
PS C:\GitHub_Repos\jay-singhvi\CC-Lab6>
```

c. Creating Public Subnet –

File – subnet-tags.json


```
$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
```

```
PS C:\GitHub_Repos\jay-singhvi\CC-Lab6> $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
PS C:\GitHub_Repos\jay-singhvi\CC-Lab6>
PS C:\GitHub_Repos\jay-singhvi\CC-Lab6> $subnetId
subnet-02c1305ba18d0ebb2
PS C:\GitHub_Repos\jay-singhvi\CC-Lab6>
PS C:\GitHub_Repos\jay-singhvi\CC-Lab6> aws ec2 `
>>   modify-subnet-attribute `
>>   --profile aws_admin_user1 `
>>   --subnet-id $subnetId `
>>   --map-public-ip-on-launch `
>>   --no-cli-pager
PS C:\GitHub_Repos\jay-singhvi\CC-Lab6>
PS C:\GitHub_Repos\jay-singhvi\CC-Lab6> aws ec2 `
>>   describe-subnets `
>>   --profile aws_admin_user1 `
>>   --no-paginate `
>>   --no-cli-pager `
>>   --subnet-ids $subnetId
```

```
-----------------------------------------------------------------------------------------------------------
|                                            DescribeSubnets                                              |
+---------------------------------------------------------------------------------------------------------+
||                                              Subnets                                                  ||
|+-----------------------------+-------------------------------------------------------------------------+|
||  AssignIpv6AddressOnCreation |  False                                                                  ||
||  AvailabilityZone           |  us-east-1a                                                             ||
||  AvailabilityZoneId         |  use1-az4                                                               ||
||  AvailableIpAddressCount    |  251                                                                    ||
||  CidrBlock                  |  10.0.1.0/24                                                            ||
||  DefaultForAz               |  False                                                                  ||
||  EnableDns64                |  False                                                                  ||
||  Ipv6Native                 |  False                                                                  ||
||  MapCustomerOwnedIpOnLaunch |  False                                                                  ||
||  MapPublicIpOnLaunch        |  True                                                                   ||
||  OwnerId                    |  381491908351                                                           ||
||  State                      |  available                                                              ||
||  SubnetArn                  |  arn:aws:ec2:us-east-1:381491908351:subnet/subnet-02c1305ba18d0ebb2     ||
||  SubnetId                   |  subnet-02c1305ba18d0ebb2                                               ||
||  VpcId                      |  vpc-0838ee0065cbe58b0                                                  ||
|+-----------------------------+-------------------------------------------------------------------------+|
|||                              PrivateDnsNameOptionsOnLaunch                                            |||
||+-----------------------------------------------------------------+-----------------------+|||
|||  EnableResourceNameDnsAAAARecord                                |  False                |||
|||  EnableResourceNameDnsARecord                                   |  False                |||
|||  HostnameType                                                   |  ip-name              |||
||+-----------------------------------------------------------------+-----------------------+|||
|||                                          Tags                                           |||
||+-----------------------------------------+-----------------------------------------------+|||
|||                  Key                    |                  Value                        |||
||+-----------------------------------------+-----------------------------------------------+|||
|||  Name                                   |  my_lab6_subnet                               |||
|||  Environment                            |  Preprod                                      |||
||+-----------------------------------------+-----------------------------------------------+|||
```

d. Creating Routing table-

File - routetable-tags.json

routetable-tags.json > ...

```
[
    {
        "ResourceType": "route-table",
        "Tags": [
            {
                "Key": "Name",
                "Value": "my_lab6_route_table"
            },
            {
                "Key": "Environment",
                "Value": "Preprod"
            }
        ]
    }
]
```

```
$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
```

```
PS C:\GitHub_Repos\jay-singhvi\CC-Lab6> $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
PS C:\GitHub_Repos\jay-singhvi\CC-Lab6> $routeTableId
rtb-0990a95033e5e3c66
PS C:\GitHub_Repos\jay-singhvi\CC-Lab6>
PS C:\GitHub_Repos\jay-singhvi\CC-Lab6> aws ec2 `
>>   describe-route-tables `
>>   --profile aws_admin_user1 `
>>   --route-table-ids $routeTableId `
>>   --output table `
>>   --no-cli-pager `
>>   --no-paginate
```

```
|                                  DescribeRouteTables                                  |
+---------------------------------------------------------------------------------------+
||                                     RouteTables                                     ||
|+-----------------------------------+-----------------------------------------------+|
||  OwnerId                          |  381491908351                                 ||
||  RouteTableId                     |  rtb-0990a95033e5e3c66                        ||
||  VpcId                            |  vpc-0838ee0065cbe58b0                        ||
|+-----------------------------------+-----------------------------------------------+|
||                                       Routes                                        ||
|+----------------------+-----------------------------+------------+-----------------+---------+|
|| DestinationCidrBlock | DestinationIpv6CidrBlock    | GatewayId  |     Origin      |  State  ||
|+----------------------+-----------------------------+------------+-----------------+---------+|
||  10.0.0.0/16         |                             |  local     |  CreateRouteTable |  active ||
||                      |  2600:1f18:5244:eb00::/56   |  local     |  CreateRouteTable |  active ||
|+----------------------+-----------------------------+------------+-----------------+---------+|
||                                        Tags                                         ||
|+-----------------------------------+-----------------------------------------------+|
||               Key                 |                    Value                      ||
|+-----------------------------------+-----------------------------------------------+|
||  Environment                      |  Preprod                                      ||
||  Name                             |  my_lab6_route_table                          ||
|+-----------------------------------+-----------------------------------------------+|
```

e. 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
```

```
PS C:\GitHub_Repos\jay-singhvi\CC-Lab6> 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
--------------------
|    CreateRoute   |
+---------+--------+
|  Return |  True  |
+---------+--------+
PS C:\GitHub_Repos\jay-singhvi\CC-Lab6>
PS C:\GitHub_Repos\jay-singhvi\CC-Lab6> aws ec2 associate-route-table `
>>     --profile aws_admin_user1 `
>>     --route-table-id $routeTableId `
>>     --subnet-id $subnetId `
>>     --no-cli-pager
-------------------------------------------------
|              AssociateRouteTable              |
+---------------+-------------------------------+
|  AssociationId |  rtbassoc-0c6d490bef009a08e  |
+---------------+-------------------------------+
||              AssociationState                ||
|+---------------+-----------------------------+|
||  State        |  associated                 ||
|+---------------+-----------------------------+|
PS C:\GitHub_Repos\jay-singhvi\CC-Lab6>
```

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

File : instance-tags.json

instance-tags.json

```
[
    {
        "ResourceType": "instance",
        "Tags": [
            {
                "Key": "Name",
                "Value": "my_lab6_ec2"
            },
            {
                "Key": "Environment",
                "Value": "Preprod"
            }
        ]
    }
]
```

```
$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
```

```
PS C:\GitHub_Repos\jay-singhvi\CC-Lab6> $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
PS C:\GitHub_Repos\jay-singhvi\CC-Lab6>
PS C:\GitHub_Repos\jay-singhvi\CC-Lab6> $instanceId
i-0fa2a6e09d5c0f0ad
PS C:\GitHub_Repos\jay-singhvi\CC-Lab6>
PS C:\GitHub_Repos\jay-singhvi\CC-Lab6> aws ec2 `
>>   describe-instances `
>>   --profile aws_admin_user1 `
>>   --instance-ids $instanceId `
>>   --no-cli-pager `
>>   --no-paginate `
>>   --output table
```

```
-------------------------------------------------------------------------
|                           DescribeInstances                           |
+-----------------------------------------------------------------------+
||                            Reservations                             ||
|+----------------------------+----------------------------------------+|
||  OwnerId                   |  381491908351                          ||
||  ReservationId             |  r-086b3ec5eb719bc0e                   ||
|+----------------------------+----------------------------------------+|
|||                             Instances                             |||
||+---------------------------+---------------------------------------+||
|||  AmiLaunchIndex           |  0                                    |||
|||  Architecture             |  x86_64                               |||
|||  BootMode                 |  uefi-preferred                       |||
|||  ClientToken              |  381aff6e-44bd-4e14-993b-7e8db757ffe8 |||
|||  CurrentInstanceBootMode  |  legacy-bios                          |||
|||  EbsOptimized             |  False                                |||
|||  EnaSupport               |  True                                 |||
|||  Hypervisor               |  xen                                  |||
|||  ImageId                  |  ami-0bb84b8ffd87024d8                |||
|||  InstanceId               |  i-0fa2a6e09d5c0f0ad                  |||
|||  InstanceType             |  t2.micro                             |||
|||  KeyName                  |  pem-key-pair                         |||
|||  LaunchTime               |  2024-05-17T23:05:40+00:00            |||
|||  PlatformDetails          |  Linux/UNIX                           |||
|||  PrivateDnsName           |  ip-10-0-1-207.ec2.internal           |||
|||  PrivateIpAddress         |  10.0.1.207                           |||
|||  PublicDnsName            |                                       |||
|||  PublicIpAddress          |  54.197.93.27                         |||
|||  RootDeviceName           |  /dev/xvda                            |||
|||  RootDeviceType           |  ebs                                  |||
|||  SourceDestCheck          |  True                                 |||
|||  StateTransitionReason    |                                       |||
|||  SubnetId                 |  subnet-02c1305ba18d0ebb2             |||
|||  UsageOperation           |  RunInstances                         |||
|||  UsageOperationUpdateTime |  2024-05-17T23:05:40+00:00            |||
|||  VirtualizationType       |  hvm                                  |||
|||  VpcId                    |  vpc-0838ee0065cbe58b0                |||
```

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

```
"--volume-type" (string)

   The volume type. This parameter can be one of the following values:

   * General Purpose SSD: "gp2" | "gp3"

   * Provisioned IOPS SSD: "io1" | "io2"

   * Throughput Optimized HDD: "st1"

   * Cold HDD: "sc1"

   * Magnetic: "standard"

   Warning: Throughput Optimized HDD ("st1" ) and Cold HDD ("sc1" ) volumes
     can't be used as boot volumes.

   For more information, see Amazon EBS volume types in the *Amazon
   EBS User Guide* .

   Default: "gp2"

   Possible values:

   * "standard"

   * "io1"

   * "io2"

   * "gp2"

   * "sc1"

   * "st1"

   * "gp3"
```

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

File - volume-tags.json

volume-tags.json > ...

```
[
    {
        "ResourceType": "volume",
        "Tags": [
            {
                "Key": "Name",
                "Value": "my_lab6_volume"
            },
            {
                "Key": "Environment",
                "Value": "Preprod"
            }
        ]
    }
]
```

```
$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
```

```
PS C:\GitHub_Repos\jay-singhvi\CC-Lab6> $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
PS C:\GitHub_Repos\jay-singhvi\CC-Lab6>
PS C:\GitHub_Repos\jay-singhvi\CC-Lab6> $volumeId
vol-02c324b420706e3e0
PS C:\GitHub_Repos\jay-singhvi\CC-Lab6>
PS C:\GitHub_Repos\jay-singhvi\CC-Lab6> aws ec2 `
>>   describe-volumes `
>>   --profile aws_admin_user1 `
>>   --volume-ids $volumeId `
>>   --no-cli-pager `
>>   --no-paginate `
>>   --output table
-------------------------------------------------------------
|                       DescribeVolumes                     |
+-----------------------------------------------------------+
||                          Volumes                        ||
|+--------------------+--------------------------------------+|
||  AvailabilityZone  |  us-east-1a                          ||
||  CreateTime        |  2024-05-17T23:39:07.874000+00:00    ||
||  Encrypted         |  False                               ||
||  MultiAttachEnabled|  False                               ||
||  Size              |  10                                  ||
||  SnapshotId        |                                      ||
||  State             |  available                           ||
||  VolumeId          |  vol-02c324b420706e3e0                ||
||  VolumeType        |  standard                            ||
|+--------------------+--------------------------------------+|
|||                         Tags                            |||
||+----------------------+------------------------------------+||
|||          Key         |              Value                 |||
||+----------------------+------------------------------------+||
|||  Name                |  my_lab6_volume                    |||
|||  Environment         |  Preprod                           |||
||+----------------------+------------------------------------+||
```

6. 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
```

```
PS C:\GitHub_Repos\jay-singhvi\CC-Lab6> aws ec2 attach-volume `
>>   --profile aws_admin_user1 `
>>   --volume-id $volumeId `
>>   --instance-id $instanceId `
>>   --device /dev/sdh `
>>   --no-cli-pager
-------------------------------------------------
|                  AttachVolume                 |
+-------------+---------------------------------+
|  AttachTime |  2024-05-17T23:40:03.883000+00:00  |
|  Device     |  /dev/sdh                       |
|  InstanceId |  i-0fa2a6e09d5c0f0ad            |
|  State      |  attaching                      |
|  VolumeId   |  vol-02c324b420706e3e0          |
+-------------+---------------------------------+
```

7. 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
```

```
PS C:\GitHub_Repos\jay-singhvi\CC-Lab6> aws ec2 modify-volume `
>>   --profile aws_admin_user1 `
>>   --volume-id $volumeId `
>>   --size 25 `
>>   --volume-type gp2 `
>>   --no-cli-pager
-------------------------------------------------------------
|                       ModifyVolume                        |
+-----------------------------------------------------------+
||                   VolumeModification                    ||
|+-----------------------------+-------------------------+|
||  ModificationState          |  modifying              ||
||  OriginalMultiAttachEnabled |  False                  ||
||  OriginalSize               |  10                     ||
||  OriginalVolumeType         |  standard               ||
||  Progress                   |  0                      ||
||  StartTime                  |  2024-05-17T23:47:45+00:00  ||
||  TargetIops                 |  100                    ||
||  TargetMultiAttachEnabled   |  False                  ||
||  TargetSize                 |  25                     ||
||  TargetVolumeType           |  gp2                    ||
||  VolumeId                   |  vol-02c324b420706e3e0  ||
|+-----------------------------+-------------------------+|
```

8. 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
```

```
PS C:\GitHub_Repos\jay-singhvi\CC-Lab6> aws ec2 delete-volume `
>>   --profile aws_admin_user1 `
>>   --volume-id $volumeId `
>>   --no-paginate `
>>   --no-cli-pager

An error occurred (VolumeInUse) when calling the DeleteVolume operation: Volume vol-02c324b420706e3e0
is currently attached to {i-0fa2a6e09d5c0f0ad}
```

9. 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
```

```
PS C:\GitHub_Repos\jay-singhvi\CC-Lab6> aws ec2 terminate-instances `
>>   --profile aws_admin_user1 `
>>   --instance-ids $instanceId `
>>   --no-paginate `
>>   --no-cli-pager
-----------------------------
|     TerminateInstances      |
+-----------------------------+
||   TerminatingInstances    ||
|+-------------------------+|
||        InstanceId         ||
|+-------------------------+|
||  i-0fa2a6e09d5c0f0ad      ||
|+-------------------------+|
|||      CurrentState       |||
||+-------+-----------------+||
||| Code  |      Name       |||
||+-------+-----------------+||
|||  32   |  shutting-down  |||
||+-------+-----------------+||
|||      PreviousState      |||
||+---------+---------------+||
|||  Code   |     Name      |||
||+---------+---------------+||
|||  16     |  running      |||
||+---------+---------------+||
PS C:\GitHub_Repos\jay-singhvi\CC-Lab6> aws ec2 `
>>   terminate-instances `
>>   --profile aws_admin_user1 `
>>   --instance-ids $instanceId `
>>   --no-paginate `
>>   --no-cli-pager
---------------------------
|    TerminateInstances    |
+--------------------------+
||  TerminatingInstances  ||
|+----------------------+|
||       InstanceId       ||
|+----------------------+|
||  i-0fa2a6e09d5c0f0ad   ||
|+----------------------+|
|||     CurrentState     |||
||+-------+-------------+||
||| Code  |    Name     |||
||+-------+-------------+||
|||  48   |  terminated |||
||+-------+-------------+||
|||     PreviousState    |||
||+-------+-------------+||
||| Code  |    Name     |||
||+-------+-------------+||
|||  48   |  terminated |||
||+-------+-------------+||
```

```
PS C:\GitHub_Repos\jay-singhvi\CC-Lab6> aws ec2 delete-volume `
>>   --profile aws_admin_user1 `
>>   --volume-id $volumeId `
>>   --no-paginate `
>>   --no-cli-pager
PS C:\GitHub_Repos\jay-singhvi\CC-Lab6> aws ec2 `
>>   describe-volumes `
>>   --profile aws_admin_user1 `
>>   --volume-ids $volumeId `
>>   --no-cli-pager `
>>   --no-paginate `
>>   --output table

An error occurred (InvalidVolume.NotFound) when calling the DescribeVolumes operation: The volume 'vol
-02c324b420706e3e0' does not exist.
```

10. 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
    a) create, list, view and delete object storage buckets
    b) 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
```

```
PS C:\GitHub_Repos\jay-singhvi\CC-Lab6> $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
PS C:\GitHub_Repos\jay-singhvi\CC-Lab6>
PS C:\GitHub_Repos\jay-singhvi\CC-Lab6> $policyId
arn:aws:iam::381491908351:policy/my-policy
PS C:\GitHub_Repos\jay-singhvi\CC-Lab6>
PS C:\GitHub_Repos\jay-singhvi\CC-Lab6> aws iam `
>>   get-policy `
>>   --profile aws_admin_user1 `
>>   --policy-arn $policyId `
>>   --output table `
>>   --no-paginate `
>>   --no-cli-pager
```

```
|                                  GetPolicy                                  |
+-----------------------------------------------------------------------------+
||                                   Policy                                  ||
|+-------------------------------+---------------------------------------------+|
||  Arn                          |  arn:aws:iam::381491908351:policy/my-policy  ||
||  AttachmentCount              |  0                                           ||
||  CreateDate                   |  2024-05-18T01:40:23+00:00                   ||
||  DefaultVersionId             |  v1                                          ||
||  IsAttachable                 |  True                                        ||
||  Path                         |  /                                           ||
||  PermissionsBoundaryUsageCount|  0                                           ||
||  PolicyId                     |  ANPAVRUVQR37ULSKB5M74                       ||
||  PolicyName                   |  my-policy                                   ||
||  UpdateDate                   |  2024-05-18T01:40:23+00:00                   ||
|+-------------------------------+---------------------------------------------+|
```

2. 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
```

```
PS C:\GitHub_Repos\jay-singhvi\CC-Lab6> $bucketName = "my-lab6-object-storage-bucket"
PS C:\GitHub_Repos\jay-singhvi\CC-Lab6> aws s3api `
>>   create-bucket `
>>   --profile aws_admin_user1 `
>>   --bucket $bucketName `
>>   --no-paginate `
>>   --output table `
>>   --region us-east-1 `
>>   --no-cli-pager
```

```
|              CreateBucket               |
+----------+------------------------------+
|  Location|  /my-lab6-object-storage-bucket|
+----------+------------------------------+
```

3. 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
```

```
PS C:\GitHub_Repos\jay-singhvi\CC-Lab6> aws s3api `
>>   list-buckets `
>>   --profile aws_admin_user1 `
>>   --no-paginate `
>>   --output table `
>>   --region us-east-1 `
>>   --no-cli-pager
-------------------------------------------------------------------------------------------
|                                        ListBuckets                                      |
+-----------------------------------------------------------------------------------------+
||                                        Buckets                                        ||
|+--------------------------+--------------------------------------------------------------+|
||  CreationDate            |  2024-05-18T03:41:54+00:00                                   ||
||  Name                    |  my-lab6-object-storage-bucket                               ||
|+--------------------------+--------------------------------------------------------------+|
||                                         Owner                                         ||
|+-------------+---------------------------------------------------------------------------+|
||  DisplayName|  jay.singhvi                                                              ||
||  ID         |  65e9570cf9881c8d66c9447e80d100bb0fd3c4876f8acb82f64676b07e4e3375         ||
|+-------------+---------------------------------------------------------------------------+|
```

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

```
PS C:\GitHub_Repos\jay-singhvi\CC-Lab6> echo "Hello Object Storage"
Hello Object Storage
PS C:\GitHub_Repos\jay-singhvi\CC-Lab6> echo "Hello Object Storage" > my-lab6-s3-file.txt
PS C:\GitHub_Repos\jay-singhvi\CC-Lab6> cat my-lab6-s3-file.txt
Hello Object Storage
```

5. 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
```

```
PS C:\GitHub_Repos\jay-singhvi\CC-Lab6> aws s3 cp my-lab6-s3-file.txt s3://$bucketName/ `
>>   --profile aws_admin_user1 `
>>   --no-cli-pager
upload: .\my-lab6-s3-file.txt to s3://my-lab6-object-storage-bucket/my-lab6-s3-file.txt
```

6. 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
```

```
PS C:\GitHub_Repos\jay-singhvi\CC-Lab6> aws s3 ls s3://$bucketName/ `
>>   --profile aws_admin_user1 `
>>   --recursive `
>>   --human-readable `
>>   --no-cli-pager
2024-05-17 20:57:13   22 Bytes my-lab6-s3-file.txt
```

7. 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
```

```
PS C:\GitHub_Repos\jay-singhvi\CC-Lab6> $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
PS C:\GitHub_Repos\jay-singhvi\CC-Lab6>
PS C:\GitHub_Repos\jay-singhvi\CC-Lab6> $instanceId
i-005bf96fba9ce80cd
PS C:\GitHub_Repos\jay-singhvi\CC-Lab6>
PS C:\GitHub_Repos\jay-singhvi\CC-Lab6> aws ec2 `
>>   describe-instances `
>>   --profile aws_admin_user1 `
>>   --instance-ids $instanceId `
>>   --no-cli-pager `
>>   --no-paginate `
>>   --output table
```

```
-------------------------------------------------------------------
|                         DescribeInstances                       |
+-----------------------------------------------------------------+
||                          Reservations                         ||
|+---------------------------+-------------------------------------+|
||  OwnerId                  |  381491908351                       ||
||  ReservationId            |  r-00fd78c5eab76c9a6                ||
|+---------------------------+-------------------------------------+|
|||                          Instances                            |||
||+--------------------------+-------------------------------------+||
|||  AmiLaunchIndex          |  0                                  |||
|||  Architecture            |  x86_64                             |||
|||  BootMode                |  uefi-preferred                     |||
|||  ClientToken             |  1c17f894-6256-4b4b-b253-692c2c9e095d|||
|||  CurrentInstanceBootMode |  legacy-bios                        |||
|||  EbsOptimized            |  False                              |||
|||  EnaSupport              |  True                               |||
|||  Hypervisor              |  xen                                |||
|||  ImageId                 |  ami-0bb84b8ffd87024d8              |||
|||  InstanceId              |  i-005bf96fba9ce80cd                |||
|||  InstanceType            |  t2.micro                           |||
|||  KeyName                 |  pem-key-pair                       |||
|||  LaunchTime              |  2024-05-18T04:01:09+00:00          |||
|||  PlatformDetails         |  Linux/UNIX                         |||
|||  PrivateDnsName          |  ip-10-0-1-98.ec2.internal          |||
|||  PrivateIpAddress        |  10.0.1.98                          |||
|||  PublicDnsName           |                                     |||
|||  PublicIpAddress         |  54.90.244.192                      |||
|||  RootDeviceName          |  /dev/xvda                          |||
|||  RootDeviceType          |  ebs                                |||
|||  SourceDestCheck         |  True                               |||
|||  StateTransitionReason   |                                     |||
|||  SubnetId                |  subnet-02c1305ba18d0ebb2           |||
|||  UsageOperation          |  RunInstances                       |||
|||  UsageOperationUpdateTime|  2024-05-18T04:01:08+00:00          |||
|||  VirtualizationType      |  hvm                                |||
|||  VpcId                   |  vpc-0838ee0065cbe58b0              |||
||+--------------------------+-------------------------------------+||
```

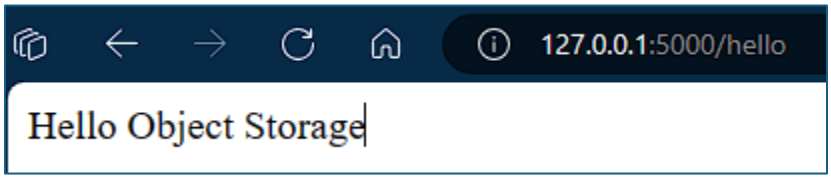
8. Write a simple Python Flask program that uses the Python SDK to read the text file from the bucket.

```
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)
```

```
PS C:\GitHub_Repos\jay-singhvi\CC-Lab6> python .\my-lab6-s3-python.py
 * Serving Flask app 'my-lab6-s3-python'
 * Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI
 server instead.
 * Running on http://127.0.0.1:5000
Press CTRL+C to quit
 * Restarting with watchdog (windowsapi)
 * Debugger is active!
 * Debugger PIN: 235-714-247
127.0.0.1 - - [17/May/2024 22:11:25] "GET /hello HTTP/1.1" 200 -
127.0.0.1 - - [17/May/2024 22:11:34] "GET /hello HTTP/1.1" 200 -
```

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

```
PS C:\Users\41222> curl http://127.0.0.1:5000/hello
Hello Object Storage
```



10. 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
```

```
PS C:\GitHub_Repos\jay-singhvi\CC-Lab6> aws s3 rm s3://$bucketName/my-lab6-s3-file.txt `
>>   --profile aws_admin_user1 `
>>   --no-cli-pager
delete: s3://my-lab6-object-storage-bucket/my-lab6-s3-file.txt
PS C:\GitHub_Repos\jay-singhvi\CC-Lab6> aws s3 rb s3://$bucketName `
>>   --profile aws_admin_user1 `
>>   --force `
>>   --no-cli-pager
remove_bucket: my-lab6-object-storage-bucket
PS C:\GitHub_Repos\jay-singhvi\CC-Lab6>
PS C:\GitHub_Repos\jay-singhvi\CC-Lab6> aws ec2 terminate-instances `
>>   --profile aws_admin_user1 `
>>   --instance-ids $instanceId `
>>   --no-cli-pager
-----------------------------
|     TerminateInstances      |
+-----------------------------+
||   TerminatingInstances    ||
|+---------------------------+|
||        InstanceId         ||
|+---------------------------+|
||  i-005bf96fba9ce80cd      ||
|+---------------------------+|
|||      CurrentState       |||
||+-------+-----------------+||
||| Code  |      Name       |||
||+-------+-----------------+||
|||  32   |  shutting-down  |||
||+-------+-----------------+||
|||      PreviousState      |||
||+---------+---------------+||
|||  Code   |     Name      |||
||+---------+---------------+||
|||  16     |  running      |||
||+---------+---------------+||
```

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