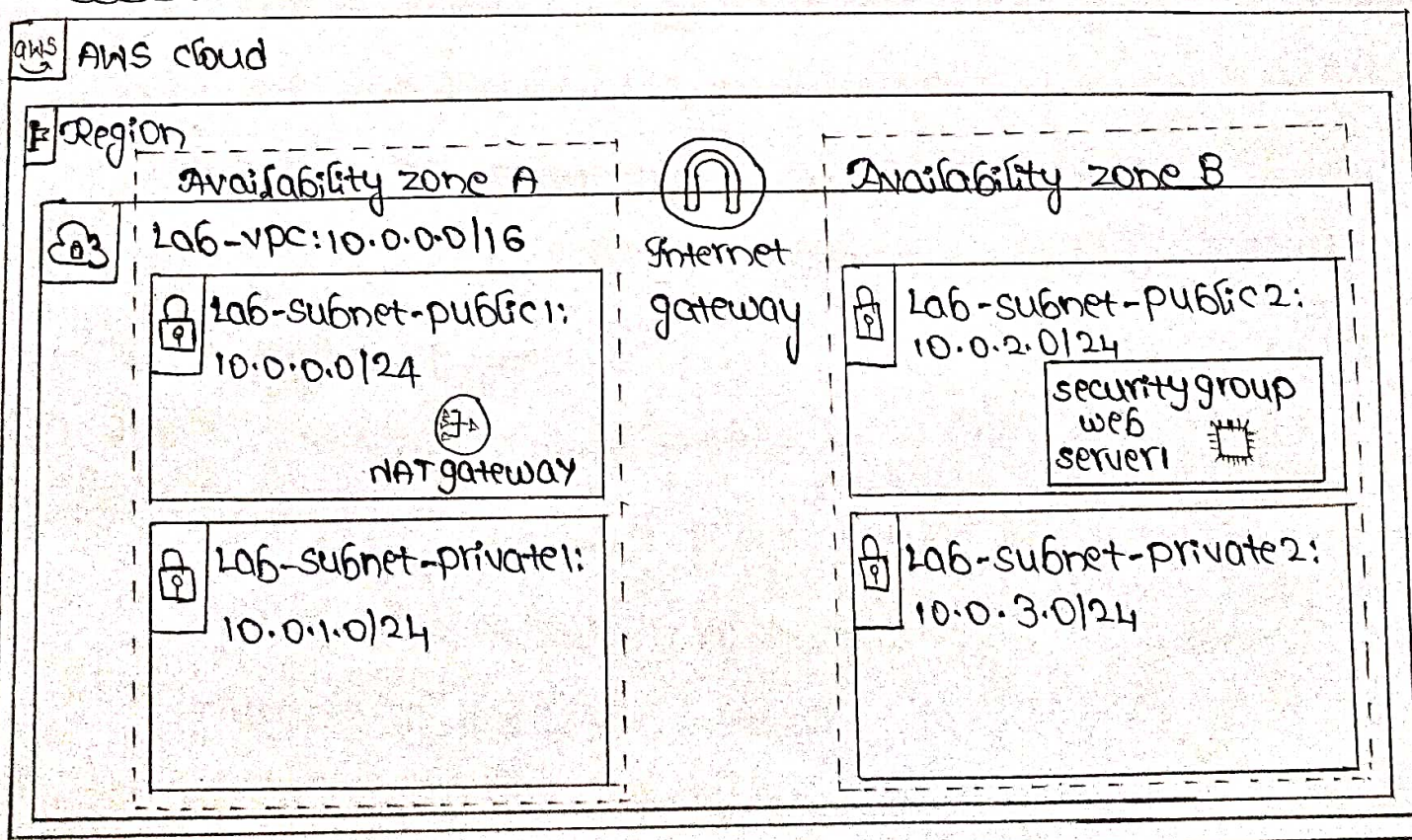
Aim :- To build your vpc and launch a web server.

Description :-

Amazon virtual private cloud (Amazon vpc) enables you to launch Amazon web services (AWS) resources into a virtual network that you defined. This virtual network closely resembles a traditional network that we would operate in our own data center, with the benefits of using the scalable infrastructure of AWS. we can create a vpc that spans multiple Availability zones. we should be able to do the create a vpc, create subnets, configure a security group and launch an EC2 instance into a vpc.

Architecture :-



Public Route Table

Destination	Target
10.0.0.0/16	Local
0.0.0.0/0	Internet gateway

Private Route Table.

Destination	Target
10.0.0.0/16	Local
0.0.0.0/0	NAT gateway

Steps followed to build vpc and launch a web server:.

* Choose start lab to launch the lab.
* Wait until we see the message "Lab status: ready".
* Choose AWS.

Task-1:. Create a vpc:.

* Choose services in navigation pane and choose vpc.
* Begin creating a vpc.
  - Verify that N.virginia (us-east-1) is the region.
  - Choose the vpc dashboard link.
  - Next, choose the create vpc.
* Configure the vpc details in vpc settings panel.
  - Choose vpc and more.
  - Under Name tag autogeneration.
  - Keep the IPV4 CIDR block set to 10.0.0.0/16
  - For number of availability zones, choose 1.
  - For number of public subnets, choose 1.
  - For number of private subnets, choose 1.

- Expand the customize subnets CIDR blocks section.
- Set NAT gateways to IN 1 AZ.
- set vpc endpoints to None.
- keep both DNS hostnames and DNS resolution enabled.

* In preview panel, confirm the settings we have configured:
  - vpc: lab-vpc
  - Subnets:
    - us-east-1a
    - public subnet name: lab-subnet-public1-us-east-1a
    - Private subnet name: lab-subnet-private1-us-east-1a
  - Route tables
    - lab-rtb-public
    - lab-rtb-private1-us-east-1a
  - Network connections
    - lab-igw
    - lab-nat-public1-us-east-1a

*choose create vpc.

*choose view vpc.

Task-2: create Additional subnets:

* In navigation pane, choose subnets.

* choose create subnet then configure:
  - vpc ID: lab-vpc
  - subnet name: lab-subnet-public2
  - Availability zone: select the second Availability zone.

- IPV4 CIDR block: 10.0.2.0/24

* choose create subnet, Here the second public subnet was created.
* choose create subnet
  - VPC ID: lab-vpc
  - Subnet name: lab-subnet-private2
  - Availability zone: select the second availability zone.
  - IPV4 CIDR block: 10.0.3.0/24
* choose the create subnet. Here the second private subnet was created.
* In navigation pane, choose Route tables.
* select ☑ the lab-rtb-private1-us-east-1a route table.
* choose the Routes tab.
* choose the subnet associations tab.
* In Explicit subnet associations panel, choose Edit subnet associations
* Leave lab-subnet-private1-us-east-1a selected, but also select ☑ lab-subnet-private2.
* choose save associations, Here lab-subnet-private-1 is created.
* select the ☑ lab-rtb-public route table.
* choose the Routes tab.
* choose the subnet associations tab
* choose the Edit subnet associations.
* Leave lab-subnet-public1-us-east-1a selected, but also select ☑ lab-subnet-public2.

* choose save associations. Here, our vpc now has public and private subnets configured in two availability zones.

Task-3: Create a vpc security Group:.

* In navigation pane, choose security groups.
* choose create security group and then configure:
  - security group name: web security group
  - Description: Enable HTTP access.
  - vpc: From drop down list, choose Lab-vpc.
* In the Inbound rules pane, choose Add rule
* configure the following settings:
  - Type ; HTTP
  - source ; Anywhere-IPV4
  - Description: permit web requests.
* choose create security group.

Task 4: Launch a web server Instance:.

* choose services, search for choose EC2 to open the EC2 console. From the launch instance menu choose Launch instance.
* Name the instance:
  - Give the name: web server 1.
* choose an AMI from which to create the instance:
  - keep the default Amazon Linux selected.
  - Also keep the default Amazon Linux 2023 AMI selected.
* choose an instance type:
  - In Instance type panel, keep the default t2.micro selected.

* select the key pair to associate with the instance:
  - From the key pair name menu, select vockey.
* Configure the network settings:.
  - Next to network settings, choose Edit, then configure:
    - Network: lab-vpc
    - Subnet: lab-subnet-public2
    - Auto-assign public IP: Enable.
  - Next, we will configure the instance to use the web security Group that we created earlier.
    - Under Firewall, choose ⊙ select existing security group.
    - For common security group, select ☑ web security group.
* In configure settings section, keep the default settings.
* Configure a script to run on the instance when it launches:
  - Expand the Advanced details panel.
  - Scroll to bottom of page, copy and paste the code shown below into the user data box.

```
#!/bin/bash
# Install Apache web server and PHP
dnf install -y httpd wget php mariadb105-server.
# Download lab files.
wget http://aws-tc-largeobjects.s3.us-west-2.amazonaws.com/
CUR/TF-100-ACCLFO-2/2-lab2-vpc/s3/lab-app.zip.
unzip lab-app.zip -d /var/www/html/
#Turn on web server.
chkconfig httpd on
service httpd start.
```

* In summary panel, choose Launch Instance
* choose View all instances
* wait until web server 1 shows 2/2 checks passed in the status check column.
* Select ☑ web server 1.
* copy the public IPV4 DNS value shown in Details tab.
* Open a new web browser tab, paste the public DNS value and press Enter.
* choose END Lab and choose yes to confirm that you want to end the lab.