Steps to create VPC

1. Creates a VPC.

   - Method: `EC2.createVpc()`
   - Parameters: `CidrBlock` ; Currently we are using `CidrBlock: "10.0.0.0/16"`
   - Return: *VpcId* such as ***vpc-a01106c2***

2. Creates Internet gateway.

   - Method: `EC2.createInternetGateway()`
   - Returns : *InternetGatewayId* such as ***igw-c0a643a9***

3. Attaches an Internet gateway to the VPC.
   Attaches an Internet gateway to a VPC, enabling connectivity between the Internet and the VPC.

   - Method: `EC2.attachInternetGateway()`
   - Parameters: `InternetGatewayId` and `VpcId`.

4. Creates a Subnet.
   create subnet of a size /24 subnet (a range of 256 private IP addresses) in the VPC.

   - Method: `EC2.createSubnet()`
   - Parameters: `CidrBlock` and `VpcId` . ; Currently we are using `CidrBlock: "10.0.1.0/24".`
   - Return: *SubnetId* such as ***subnet-9d4a7b6c***

Notes:

- When we create each subnet, we provide the VPC ID and the CIDR block we want for the subnet. After we create a subnet, we can’t change its CIDR block.

- If we create more than one subnet in a VPC, the subnets’ CIDR blocks must not overlap.The smallest subnet (and VPC) we can create uses a /28 netmask (16 IP addresses), and the largest uses a /16 netmask (65,536 IP addresses).

- The first four IP addresses and the last IP address in each subnet CIDR block are not available for us to use, and cannot be assigned to an instance. For example, in a subnet with CIDR block 10.0.0.0/24, the following five IP addresses are reserved:

    - 10.0.0.0: Network address.
    - 10.0.0.1: Reserved by AWS for the VPC router.
    - 10.0.0.2: Reserved by AWS. The IP address of the DNS server is always the base of the VPC network range plus two; however, we also reserve the base of each subnet range plus two. For more information, see Amazon DNS Server.
    - 10.0.0.3: Reserved by AWS for future use.
    - 10.0.0.255: Network broadcast address. We do not support broadcast in a VPC, therefore we reserve this address.

- Other optional parameter is `AvailabilityZone` which creates the Availability Zone for the subnet. that is the list enables us to select the Availability Zone in which to create the subnet. we can leave No Preference to let AWS choose an Availability Zone for we.

5. Creates a Route Table for the specified VPC

    - Method: `EC2.createRouteTable()`
    - Parameters: `VpcId`
    - Returns: *RouteTableId* such as ***rtb-22574640***

Notes:

- Every subnet that we create is automatically associated with the main route table for the VPC. After we create a route table, we can add routes and associate the table with a subnet.

6. Associates a Subnet with a Route Table.

    - Method: `EC2.associateRouteTable()`
    - Parameters: `RouteTableId` and `SubnetId`

Notes:

- The subnet and route table must be in the same VPC. This association causes traffic originating from the subnet to be routed according to the routes in the route table.
- A subnet can only be associated with one route table at a time,

but we can associate multiple subnets with the same route table.
- If we don’t explicitly associate a subnet with a particular route table, the subnet is implicitly associated with the main route table.

7. Create a Route inside Route table

    - Method: `createRoute()`
    - Parameters: `DestinationCidrBlock`, `RouteTableId`, `GatewayId` (If our target is InternetGateway)

Note: Route Tables for an Internet Gateway

- We must specify one of the following targets: Internet gateway or virtual private gateway, NAT instance, NAT gateway, VPC peering connection, or network interface.
- We can make a subnet a public subnet by adding a route to an Internet gateway. To do this, create and attach an Internet gateway to your VPC, and then add a route with a destination of 0.0.0.0/0 and a target of the Internet gateway ID (igw-xxxxxxxx). For more information, see Internet Gateways.

Assigning custome name to the created resources

for Assigning name to each resource type such as VPC, Route Table, Subnet or Internet Gateway.

- Method : `CreateTags()`

- Parameters : `Resources: ["rtb-78a54011"], Tags: [ { Key: "Name", Value: "Zymr-Test-RouteTable"} ]`

Notes:

- Maximum number of tags per resource—50
- Maximum key length—127 Unicode characters in UTF-8
- Maximum value length—255 Unicode characters in UTF-8
- Tag keys and values are case sensitive.
- Do not use the *aws:* prefix in tag names or values because it is reserved for AWS use. we can't edit or delete tag names or values with this prefix. Tags with this prefix do not count against your tags per resource limit.
- If our tagging schema will be used across multiple services and resources, remember that other services may have restrictions on allowed characters. Generally allowed characters are: letters, spaces, and numbers representable in UTF-8, plus the following special characters: + - = . _ : / @.

Below are the methods which can be used but need more clarifications and guidance to do so.

- Creates a VPC endpoint for a specified AWS service.

An endpoint enables us to create a private connection between our VPC and another AWS service in our account. We can specify an endpoint policy to attach to the endpoint that will control access to the service from our VPC. We can also specify the VPC route tables that use the endpoint.

- Adding Elastic IP Addresses

After we’ve launched an instance into the subnet, we must assign it an Elastic IP address if we want it to be reachable from the Internet

- Updating the Security Group Rules

VPC comes with a default security group. Each instance that we launch into a VPC is automatically associated with its default security group. The default settings for a default security group allow no inbound traffic from the Internet and allow all outbound traffic to the Internet. Therefore, to enable our instances to communicate with the Internet, create a new security group that allows public instances to access the Internet.

After we’ve created your subnet and configured your routing, We can launch an instance into the subnet. When we launch an EC2 instance into a VPC, we must specify the subnet in which to launch the instance. In this case, we’ll launch an instance into the public subnet of the VPC we created.

Steps to create spot EC2 instance

1. Request Spot Fleet Request When we create a Spot fleet

request, we must specify information about the Spot instances to launch, such as the instance type and the Spot price.

- Method : `RequestSpotFleet()`

- Parameters : This Json is generated when we request for spot instance from aws console and also generate KeyPair for SSH access

```
    "SpotFleetRequestConfig": {
        "IamFleetRole": "arn:aws:iam::9020187
21894:role/aws-ec2-spot-fleet-role",
        "AllocationStrategy": "lowestPrice",
        "TargetCapacity": 1,
        "SpotPrice": "0.067",
        "ValidFrom": "2016-10-10T05:22:21Z",
        "ValidUntil": "2017-10-10T05:22:21Z",
        "TerminateInstancesWithExpiration": t
rue,
        "LaunchSpecifications": [{
            "ImageId": "ami-2d39803a",
            "InstanceType": "m3.medium",
            "KeyName": "Zymr-KeyPair",
            "SpotPrice": "0.067",
            "IamInstanceProfile": {
                "Arn": "arn:aws:iam::90201872
1894:instance-profile/Admin"
```

```
            },
            "BlockDeviceMappings": [{
                "DeviceName": "/dev/sda1",
                "Ebs": {
                    "DeleteOnTermination": tr
ue,
                    "VolumeType": "gp2",
                    "VolumeSize": 8,
                    "SnapshotId": "snap-30671
52d"
                }
            }],
            "NetworkInterfaces": [{
                "DeviceIndex": 0,
                "SubnetId": "subnet-7785412c"
,
                "DeleteOnTermination": true,
                "AssociatePublicIpAddress": t
rue,
                "Groups": [
                    "sg-9f7a56e5"
                ]
            }]
        }],
        "Type": "request"
    }
};
```

The Spot fleet launches Spot instances when the Spot price is below our bid. The Spot instances run until either the bid price is no longer higher than the Spot price, or we terminate them ourself.

2. Add custom TCP rules this will open 8080 port with public IP and can be accessible from internet.

    - Method : `authorizeSecurityGroupIngress()`

    - Parameters :

```
            {
                GroupId: securityGroupID,
                IpPermissions: [{
                    FromPort: 8080,
                    IpProtocol: "tcp",
                    IpRanges: [{
                        CidrIp: '0.0.0.0/0'
                    }],
                    ToPort: 8080
                }]
            };
```