Task 1: In your opinion is it a good practice to create multiple users in AWS?

Justify your answer and give examples.

I think it is a good practice to create multiple users in AWS.
An advantage of having multiple users is that you can assign permissions individually to each user. You might assign administrative permissions to a few users, who then can administer your AWS resources and can even create and manage other IAM users. In most cases, however, you want to limit a user's permissions to just the tasks (AWS actions or operations) and resources that are needed for the job.
Imagine a user named Diego. When you create the IAM user Diego, you can create a password for that user. You also attach permissions to the IAM user that let him launch a specific Amazon EC2 instance and read (GET) information from a table in an Amazon RDS database.

Task 2: What could be a use case for an IAM *role*?

User's job function change
At some point, one of the developers, Zhang, changes job functions and becomes a manager. We assume that he no longer needs access to the documents in the share/developers directory. John, as an admin, moves Zhang to the Managers user group and out of the Developers user group. With just that simple reassignment, Zhang automatically gets all permissions granted to the Managers user group but can no longer access data in the share/developers directory.

Task 3: What is the difference between an IAM *role* and an IAM *policy*?

IAM Roles manage who has access to your AWS resources, whereas IAM policies control their permissions.
A Role with no Policy attached to it won't have to access any AWS resources.
A Policy that is not attached to an IAM role is effectively unused. Permissions listed in an IAM policy are only enforced when that policy is attached to an IAM identity.
Therefore, we should IAM roles and policies together to manage the security of our AWS resources.

Task 4: What is needed in order for the EC2 instance to be able to access the newly created DynamoDB table? Please consider following the best practices.

We need add an IAM role to the EC2 instance, and this IAM role should have the access to

the DynamoDB table.

Task 5: Report the steps you took in order to EC2 instance access the DynamoDB table.

1. Create a new IAM role, select the permissions policies “AmazonDynamoDBFullAccess”
2. Attach this new IAM role to the EC2 instance

Task 6: If you go to EC2 > Instances and click on the Instance you have created, then you will notice that there is a plethora of information about your newly created machine. There is an IPv4 Public IP created for your EC2 instance. If you right click and Stop the machine and then Start it again, you will realize that the IP assigned to that machine is changed. Why is that? What would you do in order to give your machine an IP Address that persists through reboots.

Why:

A public IP address is assigned to your instance from Amazon's pool of public IPv4 addresses, and is not associated with your AWS account. When a public IP address is disassociated from your instance, it is released back into the public IPv4 address pool, and you cannot reuse it.

You cannot manually associate or disassociate a public IP (IPv4) address from your instance. Instead, in certain cases, we release the public IP address from your instance, or assign it a new one:

- We release your instance's public IP address when it is stopped, hibernated, or terminated. Your stopped or hibernated instance receives a new public IP address when it is started.
- We release your instance's public IP address when you associate an Elastic IP address with it. When you disassociate the Elastic IP address from your instance, it receives a new public IP address.
- If the public IP address of your instance in a VPC has been released, it will not receive a new one if there is more than one network interface attached to your instance.
- If your instance's public IP address is released while it has a secondary private IP address that is associated with an Elastic IP address, the instance does not receive a new public IP address.

If you require a persistent public IP address that can be associated to and from instances as you require, use an Elastic IP address instead.

Assign an Elastic IP address to your instance.
Steps:

Allocate an Elastic IP address:

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose Network & Security, Elastic IPs.
3. Choose Allocate Elastic IP address.
4. For Public IPv4 address pool, choose Amazon's pool of IPv4 addresses.
5. Choose Allocate.

Associate an Elastic IP address with an instance:

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose Elastic IPs.
3. Select the Elastic IP address to associate and choose Actions, Associate Elastic IP address.
4. For Resource type, choose Instance.
5. For instance, choose the instance with which to associate the Elastic IP address. You can also enter text to search for a specific instance.
6. Choose Associate.

Task 7: If you try to setup a Web server listening to the port 8081 inside your instance you will soon realize that it is not accessible from the outside world. What is the AWS component responsible for allowing traffic to be sent to port 8081? What steps would you take in order to make it accessible from the outside world?

AWS component: security group
Add a new inbound rule to the current security group of the instance.
The rule is:

IP version	Type	Protocol	Port range	Source
IPv4	Custom TCP	TCP	8081	0.0.0.0/0

Task 8: What is the range of the IPs in the VPC you just created?

10.0.0.0/16: 10.0.0.0-10.0.255.254

Task 9: What is the difference between a VPC and a Virtual Private Network (VPN)?

A Virtual Private Cloud (VPC) allows you to virtually create a private and isolated network in the cloud. Just as a virtual private network (VPN) provides secure data transfer over the public Internet, a VPC provides secure data transfer between a private enterprise and a public cloud provider.

Task 10: What are the IP ranges of the two subnets you created?

10.0.1.0/24: 10.0.1.0-10.0.1.254
10.0.2.0/24: 10.0.2.0-10.0.2.254

Task 11: Why would someone create a public and a private subnet. What are the uses of each of them? Provide an example.

The configuration for this scenario includes a virtual private cloud (VPC) with a public subnet and a private subnet. We recommend this scenario if you want to run a public-facing web application, while maintaining back-end servers that aren't publicly accessible. A common example is a multi-tier website, with the web servers in a public subnet and the database servers in a private subnet. You can set up security and routing so that the web servers can communicate with the database servers.

Task 12: If we launch two instances, one in the public subnet and one in the private subnet, the one in the private subnet will not have internet access. How is it possible to connect to the instance in the private subnet through SSH?

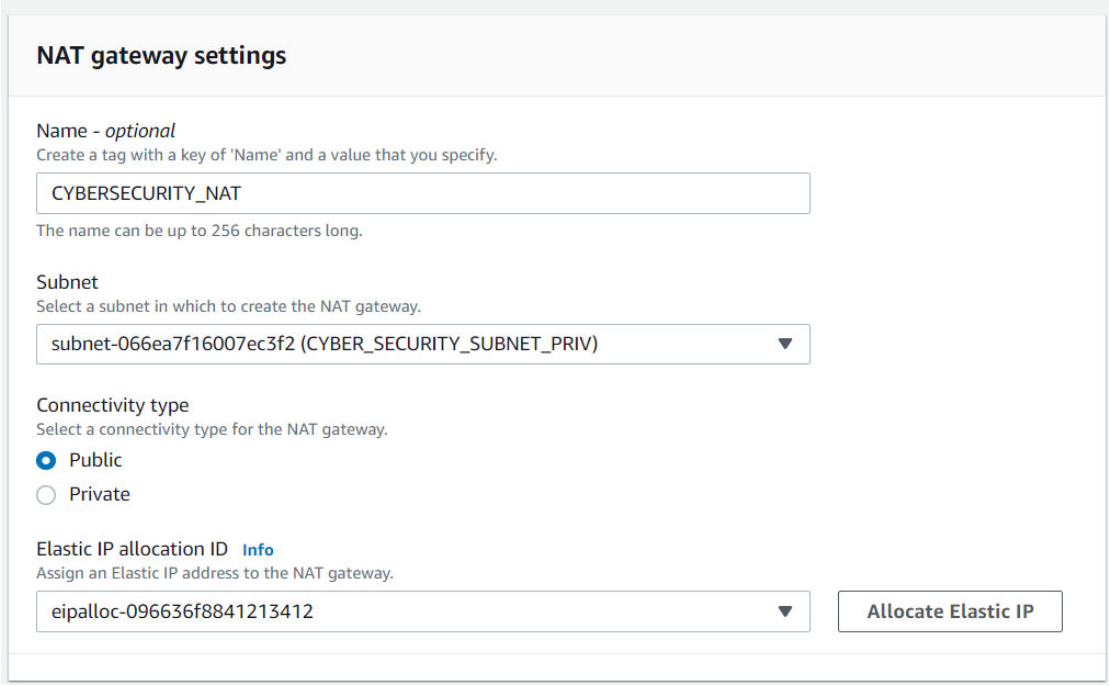
You can SSH into EC2 instances in a private subnet using SSH agent forwarding. This method allows you to securely connect to Linux instances in private Amazon VPC subnets via a bastion host (aka jump host) that is located in a public subnet.

Task 13: We can give internet access to the private subnet by creating a NAT Gateway. What is the difference between the NAT Gateway and the Internet Gateway?

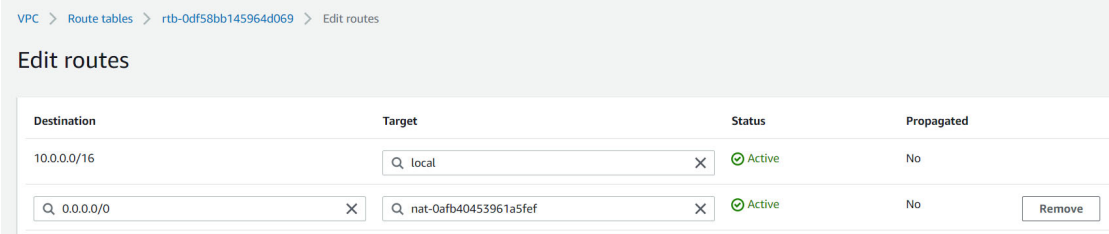
The Internet Gateway allows both inbound and outbound access to the internet whereas the NAT Gateway only allows outbound access. Thus, the Internet Gateway allows instances with public IPs to access the internet whereas the NAT Gateway allows instances with private IPs to access internet.
You only need one Internet Gateway per VPC whereas you need one NAT Gateway per Availability Zone (AZ).
There is no additional cost to use the Internet Gateway whereas the NAT Gateway incurs charges based on the creation and usage.

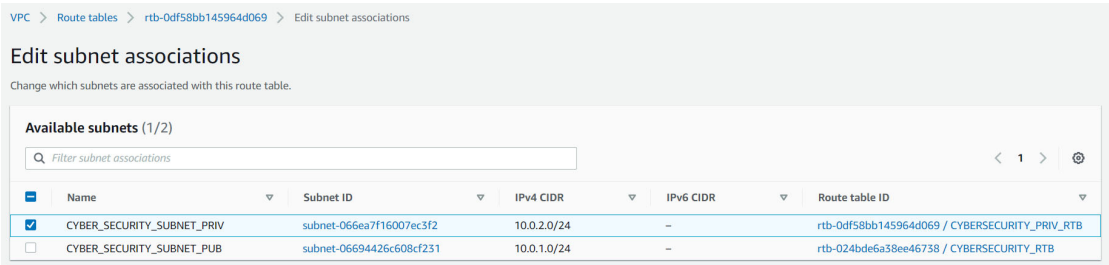
Task 14: What are the steps needed to be taken in order to create a NAT Gateway into the public subnet to provide the private subnet with internet access? You can try it by launching two instances and experimenting with the NAT Gateway.

Steps:
1. Create a public VPC subnet to host your NAT gateway.
2. Create and attach an internet gateway to your VPC.
3. Create a custom route table for your public subnet with a route to the internet gateway.
4. Verify that the network access control list (ACL) for your public VPC subnet allows inbound traffic from the private VPC subnet.
5. Create a public NAT gateway then create and associate your new or existing Elastic IP address.



6. Update the route table of your private VPC subnet to point internet traffic to your NAT gateway.





Task 15: In VPC under Security there is another module called Network ACL.

What is the difference between Network ACL and Security Groups?

Security Group	Network ACL
It enhances a security film to EC2 examples that controller together incoming and outbound circulation at the occurrence equal.	NACL correspondingly complements an extra layer of security connected with subnets that controller together inbound and outbound circulation at the subnet equal.

It provisions individual allow instructions, and through avoidance, all the rubrics remain refuted. You cannot reject the law for founding a joining.	It supports together permit and reject instructions, and through default, altogether the instructions remain refuted. You essentially complement the regulation which you can moreover permit or reject.
It remains functional to an example individual when you stipulate a security group although initiation an occurrence.	Network ACL consumes practical mechanically to altogether the occurrences which are connected with an illustration.
It remains the primary layer of protection.	It remains the second layer of protection.
The Security groups are tied to an instance.	Network ACLs are tied to the subnet.
Any changes applied to an incoming rule will be automatically applied to the outgoing rule in security groups.	In network ACL any changes applied to an incoming rule will not be applied to the outgoing rule.
All the rules are evaluated in security groups before allowing a traffic.	NACLs do the same in the number order which is from top to bottom.

Task 16: Report the steps required to create a Network ACL. How would you integrate it into the public subnet you previously created?

Steps to create a Network ACL:
To create an ACL from the AWS Console, select 'VPC > Network ACLs > Create Network ACL'. Enter a name for your ACL and select the VPC in which you want it to reside. Then select 'Yes, Create'.

Integrate it into the public subnet:
Select the Subnet Associations tab | Edit subnet associations.
Select CYBER_SECURITY_SUBNET_PUB and click on Edit.