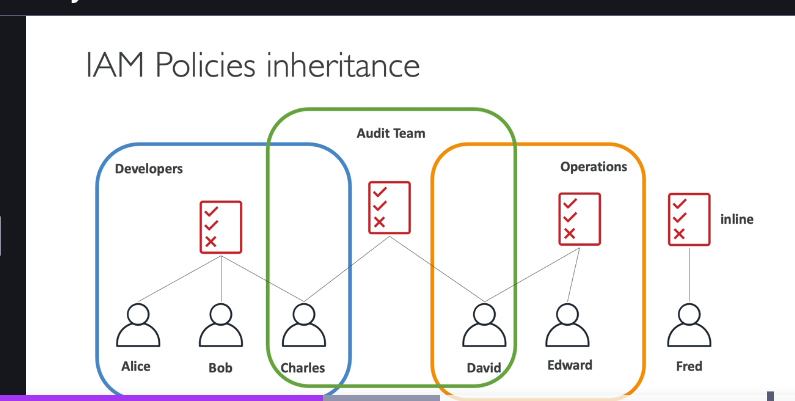
IAM

Policies:  


{

"Version": "2012-10-17",

"Statement": [

{

"Effect": "Allow",

"Action": [

"apigateway:\*"

],

"Resource": "arn:aws:apigateway:\*::/\*"

}

]

}

{

"Version": "2012-10-17",

"Statement": [

{

"Effect": "Deny",

"Action": [

"secretsmanager:GetSecretValue",

"dynamodb:PutItem",

"dynamodb:GetItem"

],

"Resource": "\*"

}

]

}

Firt we need to download aws cli and install on windows

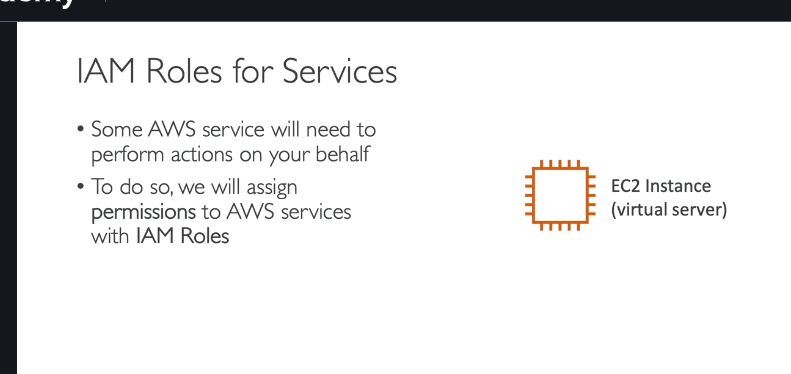
C:\Users\RAMU>aws configure

AWS Access Key ID [\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Q56A]: AKIAVK2WJ7L

AWS Secret Access Key [\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*MRvu]: iny8fhnV/MIif

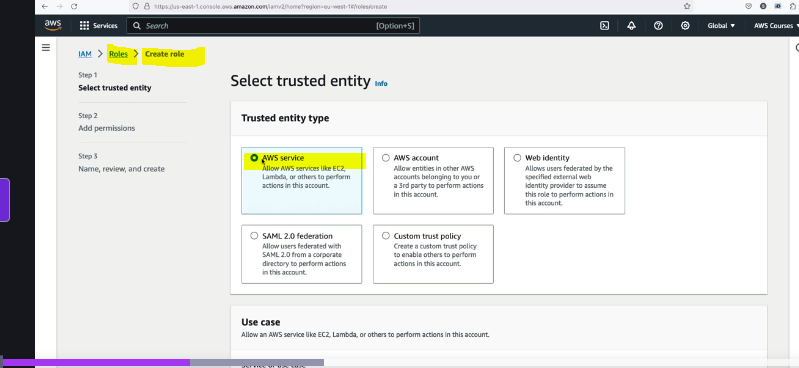
Default region name [us-east-1]:

Default output format [json]:

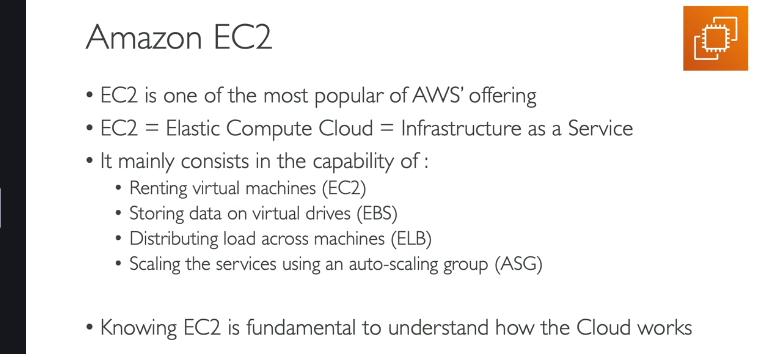


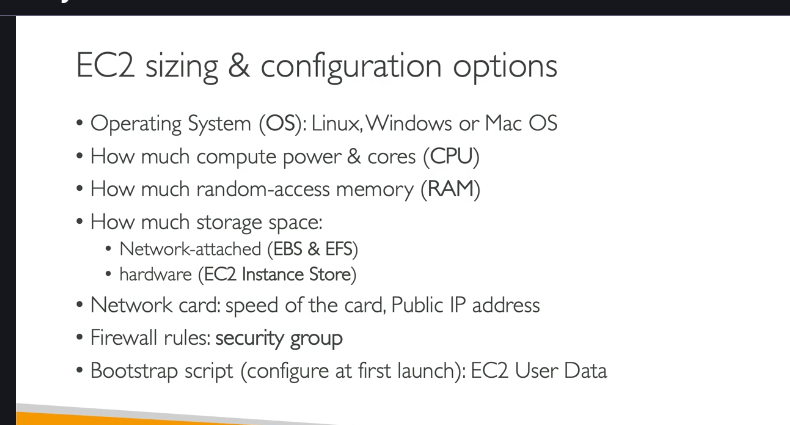
How to create a role for ec2 instance to access. After creating this role we attach it to ec2.

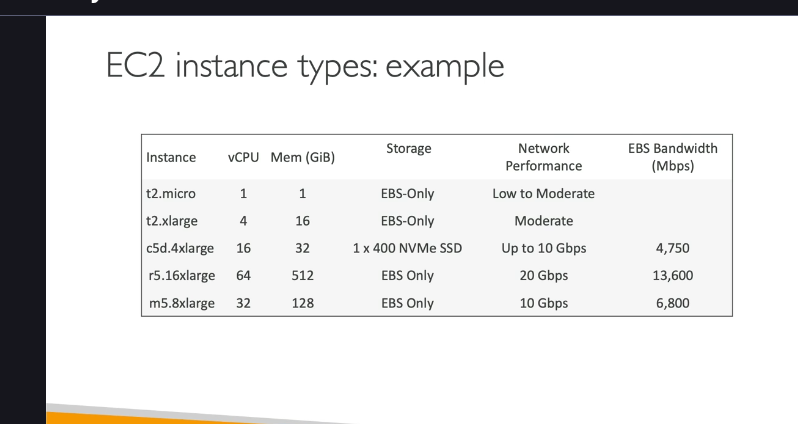
For a role we attach a policy.



**EC2: Elastic compute cloud : Infrastructure as service**







**While creating Ec2 we can provide user data, what is the use of this?**

while creating a ec2 instance we can configure user data what is the use waht data we can provide?

**ChatGPT said:**

Great question!

**✅ What is EC2 User Data?**

**User data** is a feature in AWS EC2 that allows you to **automatically run scripts** or pass configuration commands **when an EC2 instance is launched for the first time**.

It's like bootstrapping or auto-configuring your instance.

**🎯 Use Cases**

You can use **user data** to:

| **Purpose** | **Example** |
| --- | --- |
| 🛠 Install software | Apache, NGINX, Docker, Java, etc. |
| ⚙️ Configure settings | Set environment variables, create files |
| 🔐 Set up SSH keys | Add SSH public keys |
| 💻 Create users or cron jobs | Add a new user, schedule tasks |
| 📂 Mount volumes | Auto-mount EBS volumes |
| 🧪 Run startup logic | Start apps, send logs, etc. |

**📝 Types of Data You Can Provide**

You typically provide **shell scripts** or **cloud-init directives**.

**🟢 Example 1: Shell Script (Amazon Linux/Ubuntu)**

bash

CopyEdit

#!/bin/bash

yum update -y

yum install -y httpd

systemctl start httpd

systemctl enable httpd

echo "Welcome to EC2 instance!" > /var/www/html/index.html

**🟣 Example 2: Cloud-Init YAML (more structured)**

yaml

CopyEdit

#cloud-config

packages:

- httpd

runcmd:

- systemctl start httpd

- systemctl enable httpd

- echo "Deployed using cloud-init" > /var/www/html/index.html

**📌 Where Do You Enter User Data?**

You can provide user data:

* In the EC2 **Launch wizard** under “Advanced Details”
* Via **CLI** using --user-data file://my-script.sh
* Through **CloudFormation** using UserData section in your template

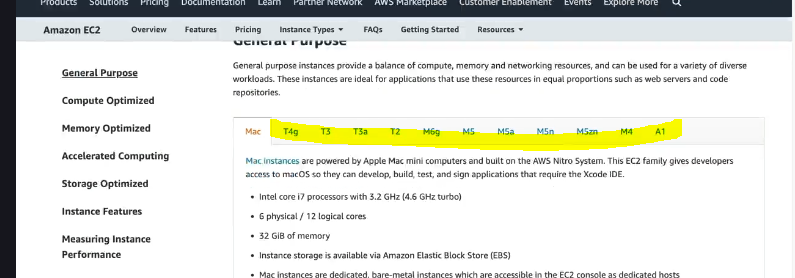
**🔄 One-Time Use Only**

* User data scripts run **only on first boot** by default.
* To make it re-run on every reboot, you must customize it (e.g., modify cloud-init settings or use cron).

**🧪 How to Test?**

1. Launch an EC2 instance and add the user data.
2. After boot, check the results:

sudo cat /var/log/cloud-init-output.log



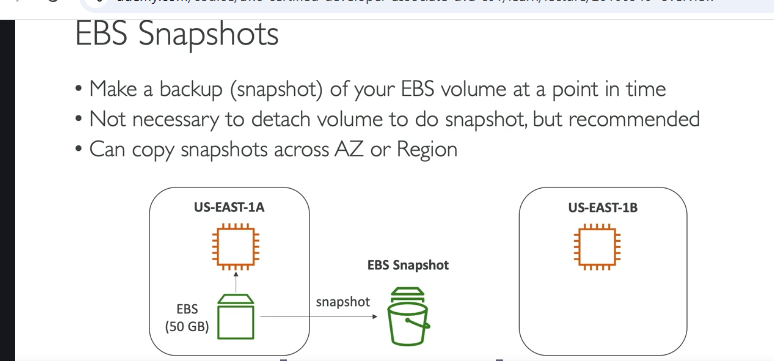
**Can we assign role ec2 ?**

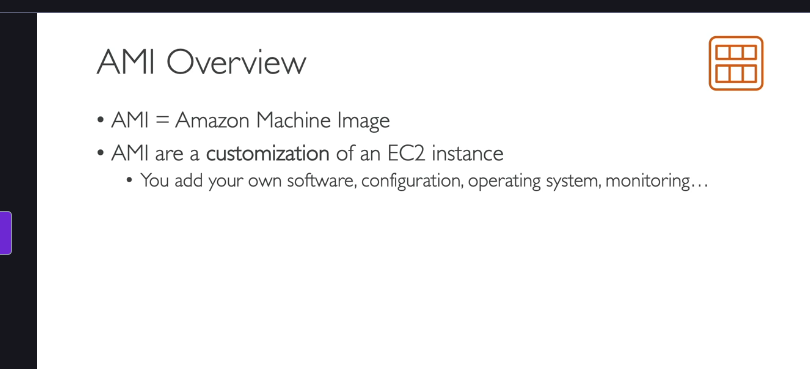
**Yes**

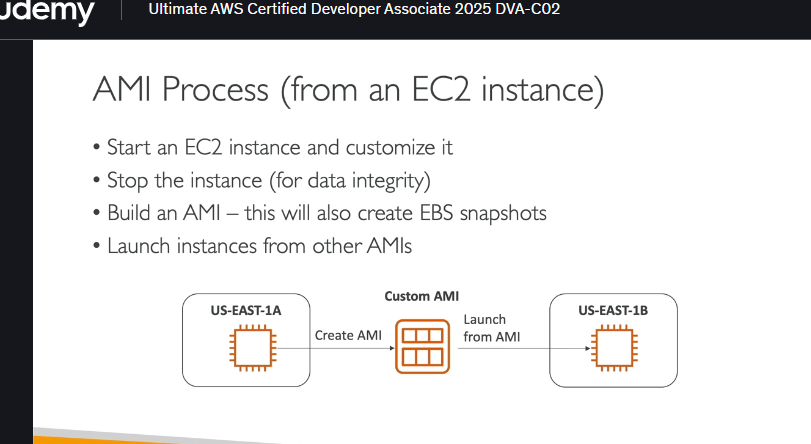
**…………………………………………………………………………………………………**

**EBS:**

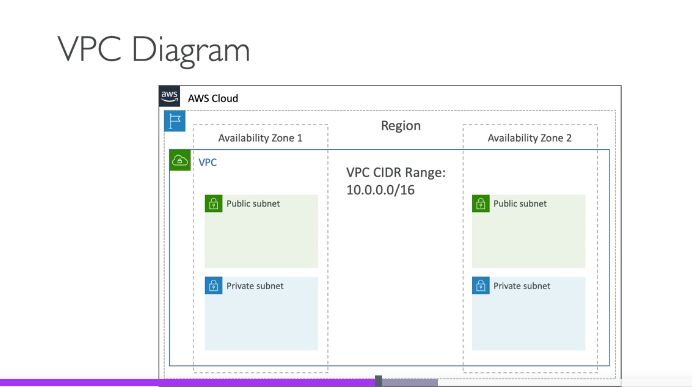
**EBS snapcahat:**

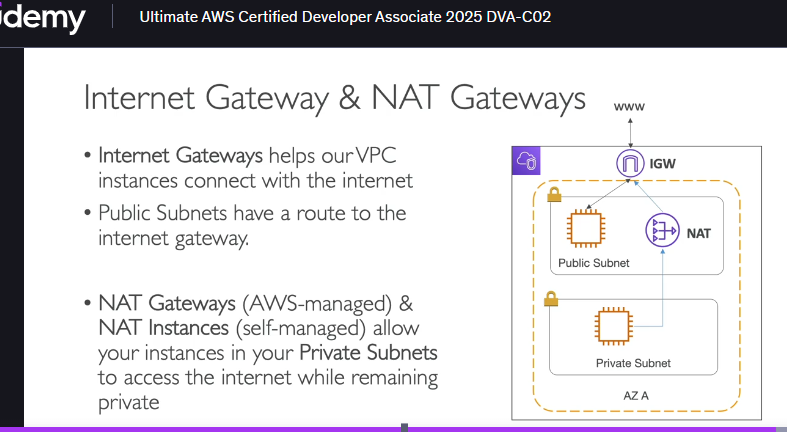


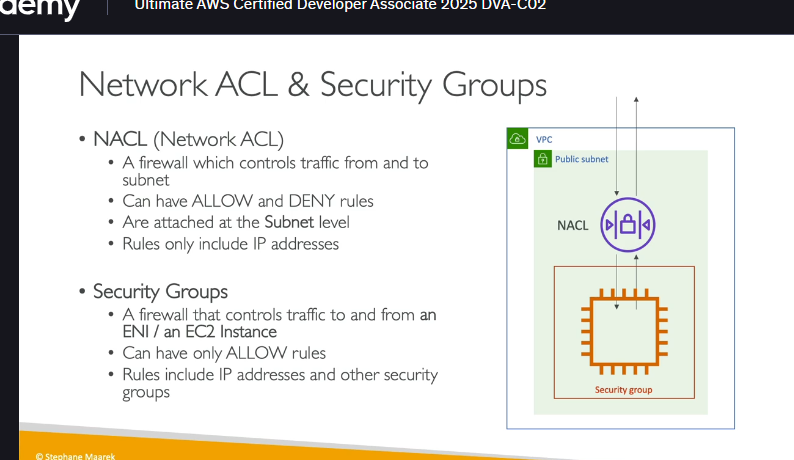


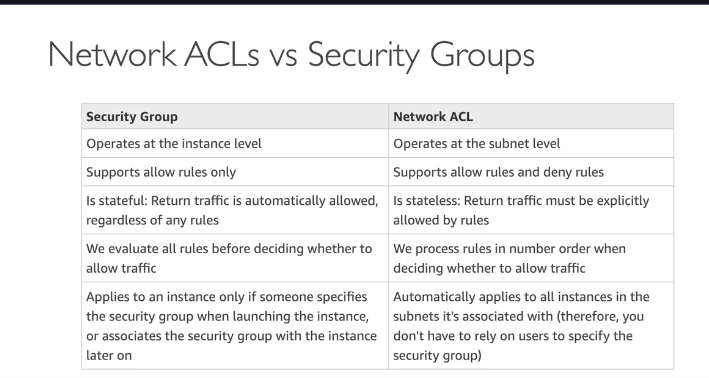


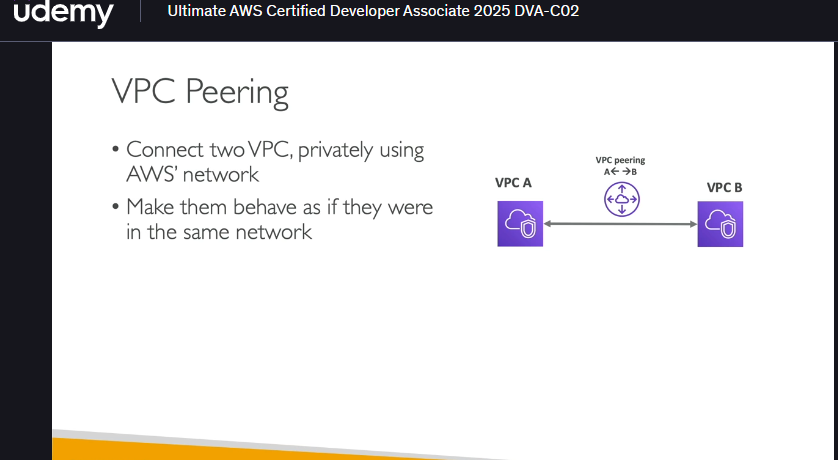
**VPC:**

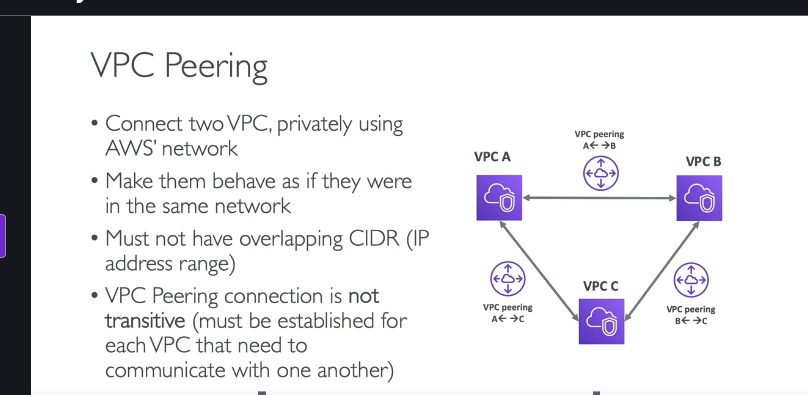


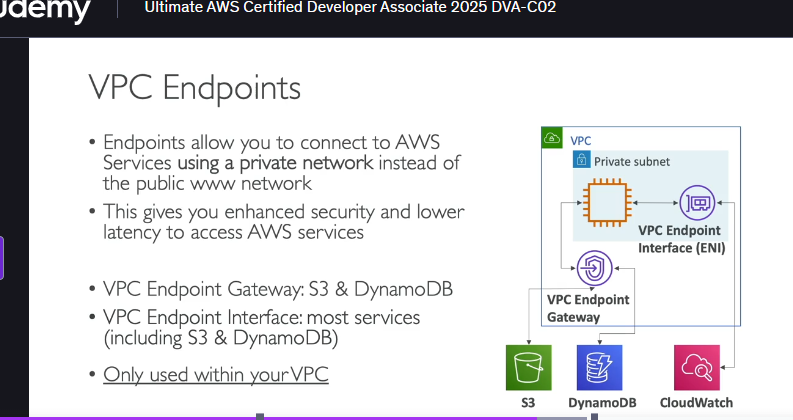


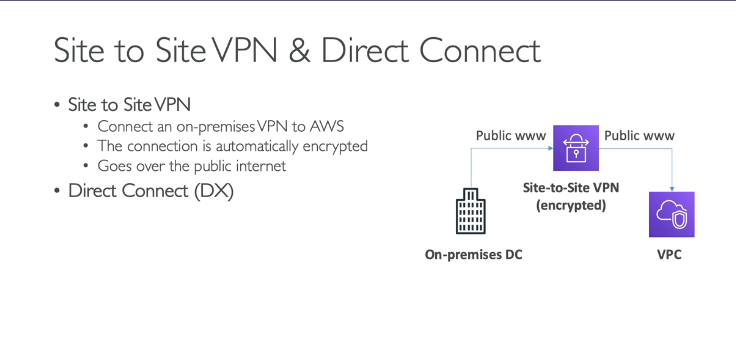


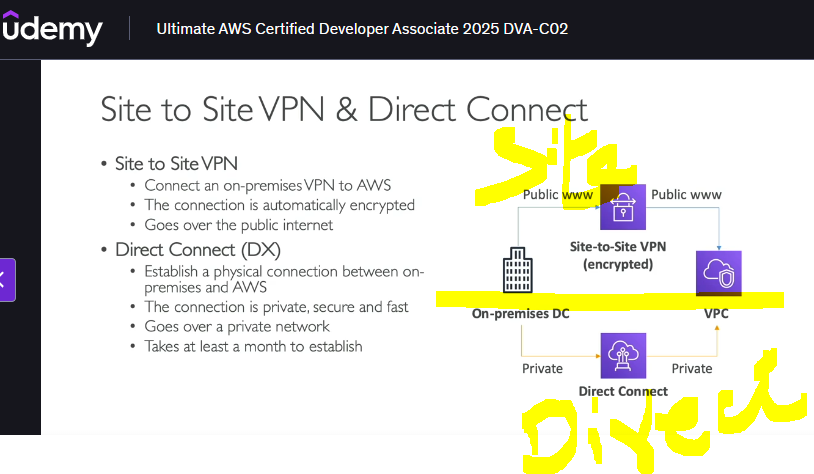


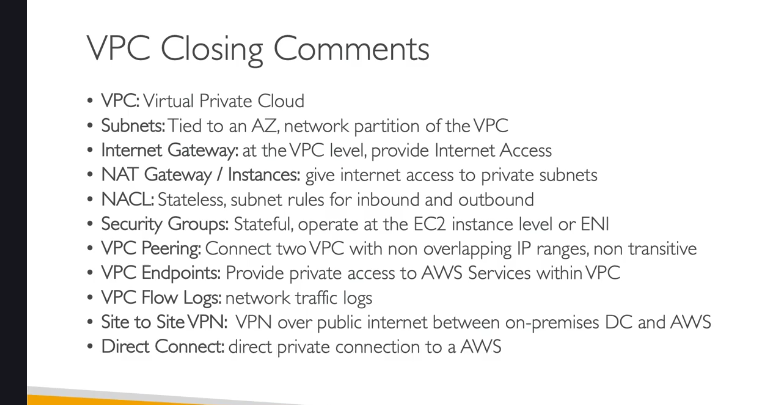












**S3:**

