

## Set -B

**Task 1:** Set up AWS Organizations for managing multiple AWS accounts.

### Login to AWS console go to Organization

The screenshot shows the AWS Console Home page. In the top right corner, there is a user menu with the account ID "3311-0291-3523" and the name "Aftab Shakir Mulla". Below the user menu, there is a sidebar titled "Recently visited" which includes links to AWS Organizations, EC2, IAM, S3, Key Management Service, Amazon Redshift, and CloudFormation. The "AWS Organizations" link is highlighted with a blue border. To the right of the sidebar, there is a section titled "Applications (0)" with a sub-section for "us-east-1 (Current Region)". At the bottom of this section, there is a "Create application" button. Further down, there is a "Cost and usage" section with a "Go to myApplications" button.

The screenshot shows the "AWS accounts | AWS Organizations" page. On the left, there is a sidebar titled "AWS Organizations" with options for "AWS accounts", "Invitations", "Services", "Policies", "Settings", and "Get started". The "AWS accounts" option is expanded, showing "Organization ID" and "o-e01ph9vwbk". The main content area is titled "AWS accounts" and contains a section for "Organization". It shows an organizational unit named "Root" with a member account "Aftab Shakir Mulla" listed. The account information includes "management account", "Joined 2023/12/19", and the email "331102913523 | aftab.smulla@gmail.com". There is also a "Hierarchy" and "List" button for viewing the account structure.

## **Create A New Organization Unit Name Test**

**Select root => Action => create new**

The screenshot shows the AWS Organizations console. On the left, there's a sidebar with 'AWS accounts' selected. The main area displays 'AWS accounts' with a search bar and a button to 'Add an AWS account'. Below this is the 'Organization' section, which explains that organizational units (OUs) enable grouping accounts. It shows a tree structure with 'Root' expanded, revealing 'Aftab Shakir Mulla' as a 'management account' joined on 2023/12/19. A context menu is open over the 'Root' node, listing actions: 'Create new', 'Rename', 'Delete', 'AWS account', 'Move', 'Remove from organization', and 'Export account list'. Another context menu is open over the 'Aftab Shakir Mulla' account, showing options like 'Edit account' and 'Delete account'. The bottom of the screen shows the Windows taskbar with various pinned icons.

The screenshot shows the 'Create organizational unit in Root' dialog. The 'Details' tab is selected, containing a form to enter the 'Organizational unit name' (set to 'Test-OU'). Below the input field is a note: 'An OU name can be up to 128 characters.' The 'Tags' tab is also present, stating 'No tags are associated with the resource.' and providing a link to 'Add tag'. At the bottom right of the dialog are 'Cancel' and 'Create organizational unit' buttons. The bottom of the screen shows the Windows taskbar with various pinned icons.



## **Test-OU Created**

The screenshot shows the AWS Organizations console with the URL <https://us-east-1.console.aws.amazon.com/organizations/v2/home/accounts>. The left sidebar shows 'AWS Organizations' and 'AWS accounts' with sub-options like 'Invitations', 'Services', 'Policies', 'Settings', and 'Get started'. The main area displays the 'AWS accounts' page with the heading 'AWS accounts'. It states: 'The accounts listed below are members of your organization. The organization's management account is responsible for paying the bills for all accounts in the organization. You can use the tools provided by AWS Organizations to centrally manage these accounts.' Below this, there is a search bar, an 'Actions' dropdown, and two buttons: 'Hierarchy' (highlighted in blue) and 'List'. The 'Organizational structure' section shows a tree view starting from 'Root' (which has 'Test-OU' as a child). Under 'Test-OU', there is one account: 'Aftab Shakir Mulla' (management account), with the details 'Joined 2023/12/19' and the email '331102913523 | aftab.smulla@gmail.com'. The status of the account is shown as 'This resource is empty'.

## **ADD AN TEST NAME AWS ACCOUNT IN TEST-OU**

The screenshot shows the 'Add an AWS account' wizard with the URL <https://us-east-1.console.aws.amazon.com/organizations/v2/home/accounts/add/create>. The left sidebar is identical to the previous screenshot. The main area shows the 'Add an AWS account' page with the heading 'Add an AWS account'. It explains that you can add an AWS account to your organization either by creating an account or by inviting one or more existing AWS accounts to join your organization. Two options are available: 'Create an AWS account' (selected) and 'Invite an existing AWS account'. The 'Create an AWS account' section contains fields for 'AWS account name' (set to 'Test'), 'Email address of the account's owner' (set to 'test.aws@google.com'), and 'IAM role name' (set to 'OrganizationAccountAccessRole'). Below these fields is a 'Tags' section with the note: 'Tags are key-value pairs that you can add to AWS resources to help identify, organize, and secure your AWS resources.'

The screenshot shows the AWS Organizations console in a web browser. A green banner at the top indicates a request to create one or more AWS accounts has been submitted. Below this, a blue banner says "AWS is creating 1 account." On the left sidebar, under "AWS accounts", there are links for Invitations, Services, Policies, Settings (which is highlighted in blue), and Get started. The Organization ID is listed as o-e01ph9vwbk. The main content area is titled "AWS accounts" and shows the organizational structure. It lists a "Root" node which contains a "Test-OU" node. Under "Test-OU", there is an account named "Aftab Shakir Mulla" (management account) with the email 331102913523 | aftab.smulla@gmail.com. This account was joined on 2023/12/19. At the bottom of the page, there are standard browser navigation and search bars, along with system status icons like CloudShell, Feedback, and a weather icon for 20°C Haze.

## **SUCCESSFULLY Set up AWS Organizations (TEST-OU) for managing multiple AWS account(TEST).**

This screenshot is from the same AWS Organizations console as the previous one, but it shows a different state. A green banner at the top indicates that the AWS account 'Test' was successfully moved to the organizational unit 'Test-OU'. The rest of the interface is identical to the first screenshot, showing the same organizational structure, account details, and system status icons.

**Task 2:** What is the purpose of use of AWS-CLI? What are the key requirements to access AWS account using AWS-CLI.

Create a working custom VPC. Launch one server in each subnet & connect private server with bastion host using CLI.

**ANS:-**

The AWS Command Line Interface (AWS-CLI) is a set of command-line tools provided by Amazon Web Services (AWS) to interact with AWS services and manage resources from the command line. It allows users to automate various tasks, such as creating and configuring AWS resources, managing security settings, and performing other administrative tasks.

**Key purposes and use cases of AWS-CLI include:**

1. **Resource Management:** Create, configure, and manage AWS resources such as EC2 instances, S3 buckets, RDS databases, and more.
2. **Automation:** Script and automate AWS tasks to improve efficiency and repeatability.
3. **Security and Access Control:** Set up and manage AWS Identity and Access Management (IAM) policies, roles, and permissions.
4. **Data Transfer:** Upload and download files to and from AWS S3, as well as transfer data between AWS services.
5. **Monitoring and Logging:** Access and analyze AWS CloudWatch metrics, logs, and events.

**Key requirements to access AWS accounts using AWS-CLI:**

1. **AWS Account:** You need an AWS account to use AWS-CLI.
2. **AWS-CLI Installation:** Install the AWS-CLI on your local machine. You can download and install it from the official AWS website or use package managers like pip.
3. **Access Key ID and Secret Access Key:** AWS-CLI requires security credentials in the form of an Access Key ID and Secret Access Key to authenticate and authorize your requests. You can obtain these credentials from the AWS Management Console by creating an IAM user with the necessary permissions.
4. **IAM Role or User Permissions:** Ensure that the IAM user or role associated with the provided credentials has the necessary permissions to perform the actions you want through AWS-CLI. Permissions are managed through IAM policies.
5. **Configuration:** After obtaining your security credentials, configure the AWS-CLI on your machine using the `aws configure` command. This involves providing your Access Key ID, Secret Access Key, region, and preferred output format.

Once these requirements are met, you can use AWS-CLI commands to interact with AWS services from your command line or scripts.

**SCREENSHOT:=>**

## Step 1: Create a VPC

Your VPCs (1) Info

Name	VPC ID	State	IPv4 CIDR	IPv6 CIDR
default	vpc-05c95515091ca2b1c	Available	172.31.0.0/16	-

Select a VPC above

```
PS C:\Users\Aftab> aws configure
AWS Access Key ID [*****]: AKIAU2F2FQPZRNQVONN7
AWS Secret Access Key [*****]: Fj77rccrZIY614YhHIPiMt+7K0azZuqLU1l+koEEV
Default region name [us-east-1]:
Default output format [json]:
PS C:\Users\Aftab> aws ec2 create-vpc --cidr-block 10.0.0.0/16
{
    "Vpc": {
        "CidrBlock": "10.0.0.0/16",
        "DhcpOptionsId": "dopt-040d92248c2b7a008",
        "State": "pending",
        "VpcId": "vpc-0d4b7c8fa5556ab17",
        "OwnerId": "331102913523",
        "InstanceTenancy": "default",
        "Ipv6CidrBlockAssociationSet": [],
        "CidrBlockAssociationSet": [
            {
                "AssociationId": "vpc-cidr-assoc-0b0587d326563d7c2",
                "CidrBlock": "10.0.0.0/16",
                "CidrBlockState": {
                    "State": "associated"
                }
            }
        ],
        "IsDefault": false
    }
}
PS C:\Users\Aftab>
```

The screenshot shows the AWS VPC console interface. On the left, there's a navigation sidebar with options like 'VPC dashboard', 'EC2 Global View', 'Filter by VPC', 'Virtual private cloud' (with 'Your VPCs' selected), 'Subnets', 'Route tables', 'Internet gateways', 'Egress-only internet gateways', 'Carrier gateways', 'DHCP option sets', 'Elastic IPs', 'Managed prefix lists', 'Endpoints', 'Endpoint services', 'NAT gateways', 'Peering connections', and 'Security'. At the top, there are tabs for 'Console Home' and 'vpcs | VPC Console'. The main area displays a table titled 'Your VPCs (2) info' with columns: Name, VPC ID, State, IPv4 CIDR, and IPv6 CIDR. Two entries are listed: 'default' (VPC ID: [vpc-03c95515091ca2b1c](#), State: Available, IPv4 CIDR: 172.31.0.0/16) and another unnamed VPC (VPC ID: [vpc-0d4b7c8fa5556ab17](#), State: Available, IPv4 CIDR: 10.0.0.0/16). Below the table, a message says 'Select a VPC above'.

## Step 2: Create Subnets

```
PS C:\Users\Aftab> # Subnet 1 (Public)
PS C:\Users\Aftab> aws ec2 create-subnet --vpc-id vpc-0d4b7c8fa5556ab17 --cidr-block 10.0.1.0/24 --availability-zone us-east-1a
{
    "Subnet": {
        "AvailabilityZone": "us-east-1a",
        "AvailabilityZoneId": "use1-az6",
        "AvailableIpAddressCount": 251,
        "CidrBlock": "10.0.1.0/24",
        "DefaultForAz": false,
        "MapPublicIpOnLaunch": false,
        "State": "available",
        "SubnetId": "subnet-05eb08c29345a8267",
        "VpcId": "vpc-0d4b7c8fa5556ab17",
        "OwnerId": "331102913523",
        "AssignIpv6AddressOnCreation": false,
        "Ipv6CidrBlockAssociationSet": [],
        "SubnetArn": "arn:aws:ec2:us-east-1:331102913523:subnet/subnet-05eb08c29345a8267",
        "EnableDns64": false,
        "Ipv6Native": false,
        "PrivateDnsNameOptionsOnLaunch": {
            "HostnameType": "ip-name",
            "EnableResourceNameDnsARecord": false,
            "EnableResourceNameDnsAAAARecord": false
        }
    }
}

PS C:\Users\Aftab>
PS C:\Users\Aftab> # Subnet 2 (Private)
PS C:\Users\Aftab> aws ec2 create-subnet --vpc-id vpc-0d4b7c8fa5556ab17 --cidr-block 10.0.2.0/24 --availability-zone us-east-1a
{
    "Subnet": {
        "AvailabilityZone": "us-east-1a",
        "AvailabilityZoneId": "use1-az6",
        "AvailableIpAddressCount": 251,
        "CidrBlock": "10.0.2.0/24",
        "DefaultForAz": false,
        "MapPublicIpOnLaunch": false,
        "State": "available",
    }
}
```

```

Windows PowerShell
PS C:\Users\Aftab>
PS C:\Users\Aftab> # Subnet 2 (Private)
PS C:\Users\Aftab> aws ec2 create-subnet --vpc-id vpc-0d4b7c8fa5556ab17 --cidr-block 10.0.2.0/24 --availability-zone us-east-1a
{
    "Subnet": {
        "AvailabilityZone": "us-east-1a",
        "AvailabilityZoneId": "use1-az6",
        "AvailableIpAddressCount": 251,
        "CidrBlock": "10.0.2.0/24",
        "DefaultForAz": false,
        "MapPublicIpOnLaunch": false,
        "State": "available",
        "SubnetId": "subnet-05c874f2084d30b8b",
        "VpcId": "vpc-0d4b7c8fa5556ab17",
        "OwnerId": "331102913523",
        "AssignIpv6AddressOnCreation": false,
        "IPv6CidrBlockAssociationSet": [],
        "SubnetArn": "arn:aws:ec2:us-east-1:331102913523:subnet/subnet-05c874f2084d30b8b",
        "EnableDns64": false,
        "Ipv6Native": false,
        "PrivateDnsNameOptionsOnLaunch": {
            "HostnameType": "ip-name",
            "EnableResourceNameDnsARecord": false,
            "EnableResourceNameDnsAAAARecord": false
        }
    }
}

PS C:\Users\Aftab>

```



## NUMBER OF EC2 INSTANCE

AWS Services Key Management Service Instances (2) Info

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4
linux	i-0d879f0f33b338b37	Stopped	t2.micro	-	<a href="#">View alarms</a>	us-east-1a	-
git	i-05cca0cd6fc63355d	Stopped	t2.micro	-	<a href="#">View alarms</a>	us-east-1c	-

Select an instance

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

## Step 5: Create a Route Table for Public Subnet

## Step 6: Add a Route to the Route Table

```
ec2-user@ip-10-0-2-133:~ % Windows PowerShell % + - x
"EnableDns64": false,
"IPv6Native": false,
"PrivateDnsNameOptionsOnLaunch": {
    "HostnameType": "ip-name",
    "EnableResourceNameDnsARecord": false,
    "EnableResourceNameDnsAAAARecord": false
}
}

PS C:\Users\Aftab> aws ec2 create-internet-gateway
{
    "InternetGateway": {
        "Attachments": [],
        "InternetGatewayId": "igw-06054c91a298bb0e3",
        "OwnerId": "331102913523",
        "Tags": []
    }
}

PS C:\Users\Aftab> aws ec2 attach-internet-gateway --vpc-id vpc-0d4b7c8fa5556ab17 --internet-gateway-id igw-06054c91a298bb0e3
PS C:\Users\Aftab> aws ec2 create-route-table --vpc-id vpc-0d4b7c8fa5556ab17
{
    "RouteTable": {
        "Associations": [],
        "PropagatingVgws": [],
        "RouteTableId": "rtb-0c52a4ca535c83d07",
        "Routes": [
            {
                "DestinationCidrBlock": "10.0.0.0/16",
                "GatewayId": "local",
                "Origin": "CreateRouteTable",
                "State": "active"
            }
        ],
        "Tags": [],
        "VpcId": "vpc-0d4b7c8fa5556ab17",
        "OwnerId": "331102913523"
    }
}

19°C Partly cloudy 21:37 ENG IN 14-01-2024
```

```
ec2-user@ip-10-0-2-133:~ % Windows PowerShell % + - x
PS C:\Users\Aftab> aws ec2 attach-internet-gateway --vpc-id vpc-0d4b7c8fa5556ab17 --internet-gateway-id igw-06054c91a298bb0e3
PS C:\Users\Aftab> aws ec2 create-route-table --vpc-id vpc-0d4b7c8fa5556ab17
{
    "RouteTable": {
        "Associations": [],
        "PropagatingVgws": [],
        "RouteTableId": "rtb-0c52a4ca535c83d07",
        "Routes": [
            {
                "DestinationCidrBlock": "10.0.0.0/16",
                "GatewayId": "local",
                "Origin": "CreateRouteTable",
                "State": "active"
            }
        ],
        "Tags": [],
        "VpcId": "vpc-0d4b7c8fa5556ab17",
        "OwnerId": "331102913523"
    }
}

PS C:\Users\Aftab> aws ec2 create-route --route-table-id rtb-0c52a4ca535c83d07 --destination-cidr-block 0.0.0.0/0 --gateway-id igw-06054c91a298bb0e3
{
    "Return": true
}

PS C:\Users\Aftab> aws ec2 associate-route-table --subnet-id subnet-05eb08c29345a8267 --route-table-id rtb-0c52a4ca535c83d07
{
    "AssociationId": "rtbassoc-09d7ebbc9b817f875",
    "AssociationState": {
        "State": "associated"
    }
}

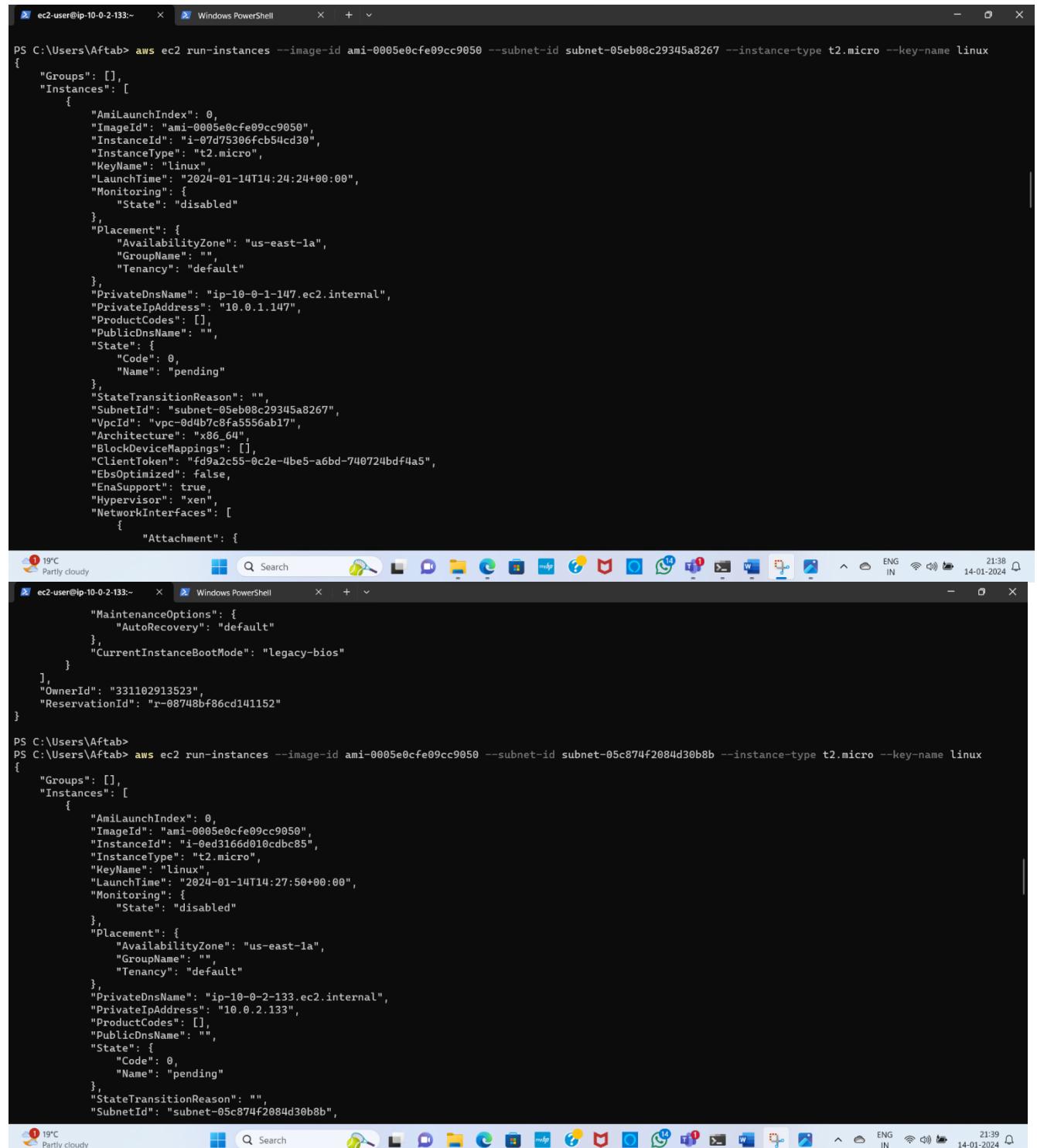
PS C:\Users\Aftab> aws ec2 run-instances --image-id ami-0005e0cf09cc9050 --subnet-id subnet-05eb08c29345a8267 --instance-type t2.micro --key-name linux
{
    "Groups": [],
    "Instances": [
        {
            "ImageId": "ami-0005e0cf09cc9050",
            "InstanceType": "t2.micro",
            "KeyName": "linux",
            "MinCount": 1,
            "MaxCount": 1
        }
    ]
}

19°C Partly cloudy 21:38 ENG IN 14-01-2024
```

## Step 7: Associate the Route Table with the Public Subnet

## Step 8: Launch an EC2 Instance in the Public Subnet (Bastion Host)

## Step 9: Launch an EC2 Instance in the Private Subnet



The image shows two side-by-side Windows PowerShell windows. Both windows have the title bar "Windows PowerShell". The top window's command is:

```
PS C:\Users\Aftab> aws ec2 run-instances --image-id ami-0005e0cf09cc9050 --subnet-id subnet-05eb08c29345a8267 --instance-type t2.micro --key-name linux
```

The bottom window's command is:

```
PS C:\Users\Aftab> aws ec2 run-instances --image-id ami-0005e0cf09cc9050 --subnet-id subnet-05c874f2084d30b8b --instance-type t2.micro --key-name linux
```

In both windows, the command is identical except for the subnet ID. The output of the command is displayed below the prompt, showing the details of the launched EC2 instance, such as its ID, type, state, and network configuration.

```

ec2-user@ip-10-0-2-133:~ % Windows PowerShell
DeviceIndex": 0,
"Status": "attaching",
"NetworkCardIndex": 0
},
"Description": "",
"Groups": [
{
"GroupName": "default",
"GroupId": "sg-0552c847592f70374"
}
],
"Ipv6Addresses": [],
"MacAddress": "0e:5a:df:bb:55:f5",
"NetworkInterfaceId": "eni-0f03307e2e61a7b22",
"OwnerId": "331102913523",
"PrivateIpAddress": "10.0.2.133",
"PrivateIpAddresses": [
{
"Primary": true,
"PrivateIpAddress": "10.0.2.133"
},
{
"SourceDestCheck": true,
"Status": "in-use",
"SubnetId": "subnet-05c874f2084d30b8b",
"VpcId": "vpc-0d4b7c8fa5556ab17",
"InterfaceType": "interface"
},
{
"RootDeviceName": "/dev/xvda",
"RootDeviceType": "ebs",
"SecurityGroups": [
{
"GroupName": "default",
"GroupId": "sg-0552c847592f70374"
}
],
"SourceDestCheck": true,
"StateReason": {
"Code": "pending",
}
]

```

## NUMBER OF SERVER

**Instances (1/5) Info**

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Put
i-07d75306fc54cd30	i-07d75306fc54cd30	Terminated	t2.micro	-	View alarms +	us-east-1a	-
Bastion	i-05833b3b3cd1eca37	Running	t2.micro	Initializing	View alarms +	us-east-1a	-
linux	i-0d879f0f33b338b37	Stopped	t2.micro	-	View alarms +	us-east-1a	-
git	i-05cca0cd6fc63355d	Stopped	t2.micro	-	View alarms +	us-east-1c	-
private	i-0ed3166d010cdcb85	Running	t2.micro	2/2 checks passed	View alarms +	us-east-1a	-

**Instance: i-05833b3b3cd1eca37 (Bastion)**

**Details** | Status and alarms New | Monitoring | Security | Networking | Storage | Tags

**Instance summary**

Instance ID i-05833b3b3cd1eca37 (Bastion)	Public IPv4 address 52.90.239.160 [open address]	Private IPv4 addresses 10.0.1.206
IPv6 address -	Instance state Running	Public IPv4 DNS -
Hostname type IP name: ip-10-0-1-206.ec2.internal	Private IP DNS name (IPv4 only) ip-10-0-1-206.ec2.internal	

## Step 10: Connect to the Bastion Host

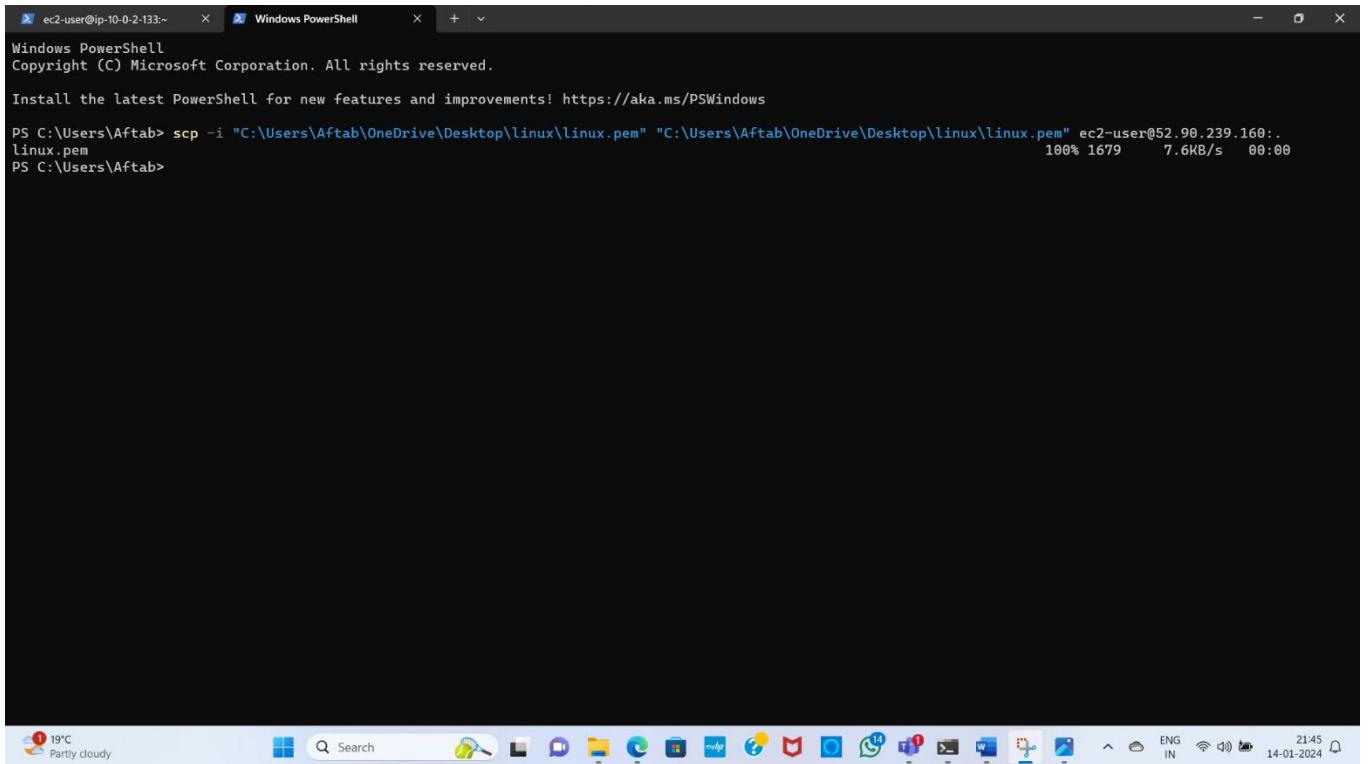
```
PS C:\Users\Aftab> aws ec2 modify-subnet-attribute --subnet-id subnet-05eb08c29345a8267 --map-public-ip-on-launch
PS C:\Users\Aftab> ssh -i YOUR_KEY_PAIR_FILE.pem ec2-user@
     ssh -i linux.pem ec2-user@52.90.239.160
Warning: Identity file linux.pem not accessible: No such file or directory.
The authenticity of host '52.90.239.160 (52.90.239.160)' can't be established.
ED25519 key fingerprint is SHA256:905JLBduX+xcZt5HvTdiMrDA8SY6aiZ31EM1Mvhj8.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '52.90.239.160' (ED25519) to the list of known hosts.
ec2-user@52.90.239.160: Permission denied (publickey,gssapi-keyex,gssapi-with-mic).
PS C:\Users\Aftab> ssh -i linux.pem ec2-user@52.90.239.160
Warning: Identity file linux.pem not accessible: No such file or directory.
ec2-user@52.90.239.160: Permission denied (publickey,gssapi-keyex,gssapi-with-mic).
PS C:\Users\Aftab>
     ssh -i C:\Users\Aftab\OneDrive\Desktop\linux.pem ec2-user@52.90.239.160
Warning: Identity file C:\Users\Aftab\OneDrive\Desktop\linux.pem not accessible: No such file or directory.
ec2-user@52.90.239.160: Permission denied (publickey,gssapi-keyex,gssapi-with-mic).
PS C:\Users\Aftab> ssh -i C:\Users\Aftab\OneDrive\Desktop\linux.pem ec2-user@52.90.239.160
Warning: Identity file C:\Users\Aftab\OneDrive\Desktop\linux.pem not accessible: No such file or directory.
ec2-user@52.90.239.160: Permission denied (publickey,gssapi-keyex,gssapi-with-mic).
PS C:\Users\Aftab> ssh -i C:\Users\Aftab\OneDrive\Desktop\linux.pem ec2-user@52.90.239.160
     scp -i C:\Users\Aftab\OneDrive\Desktop\linux.pem ec2-user@52.90.239.160
usage: scp [-346ABCpqrv] [-c cipher] [-F ssh_config] [-i identity_file]
          [-J destination] [-l limit] [-o ssh_option] [-P port]
          [-S program] source ... target
PS C:\Users\Aftab> ssh -i C:\Users\Aftab\OneDrive\Desktop\linux.pem ec2-user@52.90.239.160
Warning: Identity file C:\Users\Aftab\OneDrive\Desktop\linux.pem not accessible: No such file or directory.
ec2-user@52.90.239.160: Permission denied (publickey,gssapi-keyex,gssapi-with-mic).
PS C:\Users\Aftab>
     > ssh -i "C:\Users\Aftab\OneDrive\Desktop\linux\linux.pem" ec2-user@52.90.239.160
#_#_#_#
Amazon Linux 2023
\###\#
\##|_
#/ __ https://aws.amazon.com/linux/amazon-linux-2023
V~'^'>
/ /
/ /
/m/
```



```
PS C:\Users\Aftab> ssh -i C:\Users\Aftab\OneDrive\Desktop\linux.pem ec2-user@52.90.239.160
Warning: Identity file C:\Users\Aftab\OneDrive\Desktop\linux.pem not accessible: No such file or directory.
ec2-user@52.90.239.160: Permission denied (publickey,gssapi-keyex,gssapi-with-mic).
PS C:\Users\Aftab>
     > ssh -i "C:\Users\Aftab\OneDrive\Desktop\linux\linux.pem" ec2-user@52.90.239.160
#_#_#_#
Amazon Linux 2023
\###\#
\##|_
#/ __ https://aws.amazon.com/linux/amazon-linux-2023
V~'^'>
/ /
/ /
/m/
[ec2-user@ip-10-0-1-206 ~]$ ssh -i linux.pem ec2-user@10.0.2.133
Warning: Identity file linux.pem not accessible: No such file or directory.
The authenticity of host '10.0.2.133 (10.0.2.133)' can't be established.
ED25519 key fingerprint is SHA256:AsqAK6Ah/sclvDWBXFJ8ZVEPY1qe9pvE67yf0pGoCRY.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.0.2.133' (ED25519) to the list of known hosts.
ec2-user@10.0.2.133: Permission denied (publickey,gssapi-keyex,gssapi-with-mic).
[ec2-user@ip-10-0-1-206 ~]$ ^C
[ec2-user@ip-10-0-1-206 ~]$ ls
linux.pem
[ec2-user@ip-10-0-1-206 ~]$ ssh -i linux.pem ec2-user@10.0.2.133
0000000000000000000000000000000000000000000000000000000000000000
@   WARNING: UNPROTECTED PRIVATE KEY FILE!    @
000000000000000000000000000000000000000000000000000000000000000
Permissions 0664 for 'linux.pem' are too open.
It is required that your private key files are NOT accessible by others.
This private key will be ignored.
Load key "linux.pem": bad permissions
ec2-user@10.0.2.133: Permission denied (publickey,gssapi-keyex,gssapi-with-mic).
[ec2-user@ip-10-0-1-206 ~]$ sudo chmod 400 linux.pem
[ec2-user@ip-10-0-1-206 ~]$ ssh -i linux.pem ec2-user@10.0.2.133
#_#_#_#
Amazon Linux 2023
\###\#
```



## **.pem file from laptop to server**

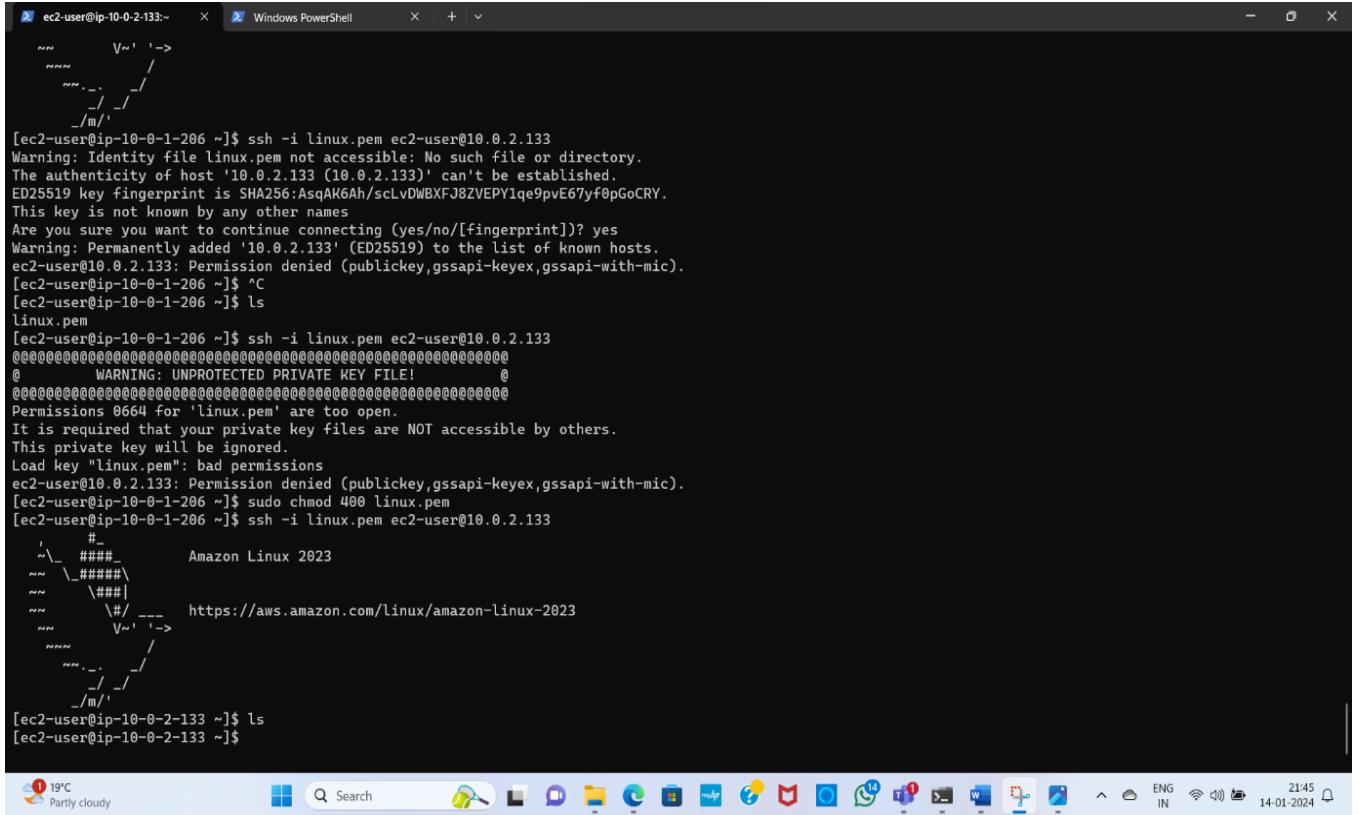


```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\Aftab> scp -i "C:\Users\Aftab\OneDrive\Desktop\linux\linux.pem" "C:\Users\Aftab\OneDrive\Desktop\linux\linux.pem" ec2-user@52.90.239.160:.
 100% 1679    7.6KB/s   00:00
PS C:\Users\Aftab>
```

## **Step 11: From the Bastion Host, Connect to the Private Server**



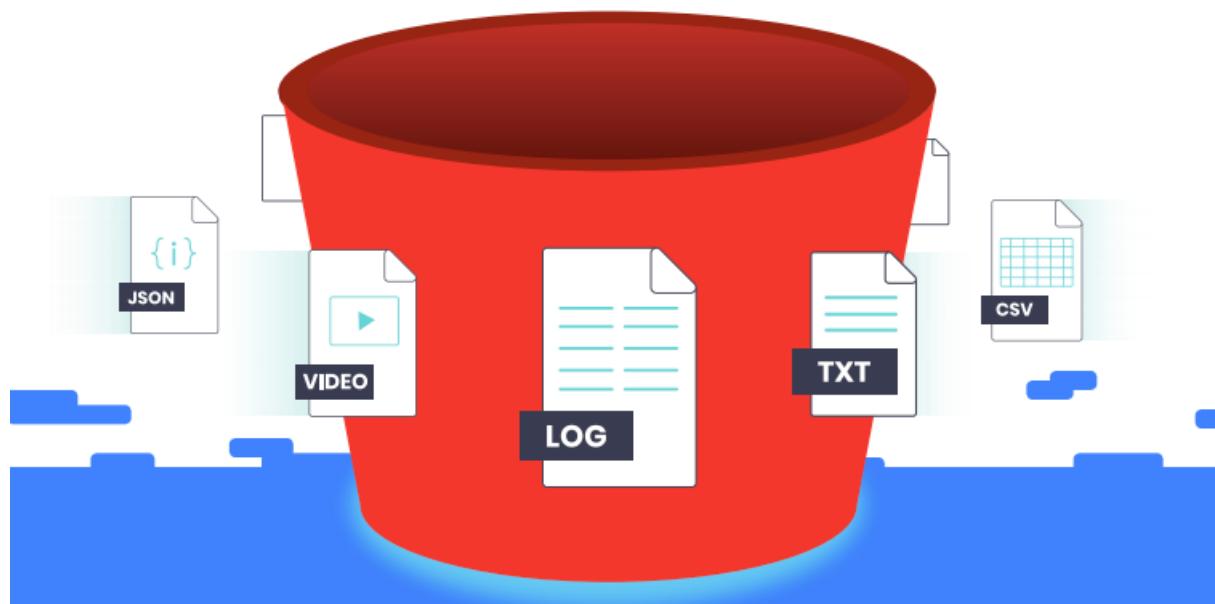
```
[ec2-user@ip-10-0-1-206 ~]$ ssh -i linux.pem ec2-user@10.0.2.133
Warning: Identity file linux.pem not accessible: No such file or directory.
The authenticity of host '10.0.2.133' (10.0.2.133) can't be established.
ED25519 key fingerprint is SHA256:AsqAK6Ah/scLvDWBXFJ8ZVEPV1qe9pvE67yf0pGoCRY.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.0.2.133' (ED25519) to the list of known hosts.
[ec2-user@10.0.2.133: Permission denied (publickey,gssapi-keyex,gssapi-with-mic).]
[ec2-user@ip-10-0-1-206 ~]$ ^C
[ec2-user@ip-10-0-1-206 ~]$ ls
linux.pem
[ec2-user@ip-10-0-1-206 ~]$ ssh -i linux.pem ec2-user@10.0.2.133
@@@@@@@@@@@ 10.0.2.133: Permission denied (publickey,gssapi-keyex,gssapi-with-mic).
@          WARNING: UNPROTECTED PRIVATE KEY FILE!
@@@@@@@@@@@ 10.0.2.133: Permission denied (publickey,gssapi-keyex,gssapi-with-mic).
Permissions 0664 for 'linux.pem' are too open.
It is required that your private key files are NOT accessible by others.
This private key will be ignored.
Load key "linux.pem": bad permissions
[ec2-user@10.0.2.133: Permission denied (publickey,gssapi-keyex,gssapi-with-mic).]
[ec2-user@ip-10-0-1-206 ~]$ sudo chmod 400 linux.pem
[ec2-user@ip-10-0-1-206 ~]$ ssh -i linux.pem ec2-user@10.0.2.133
#
#          Amazon Linux 2023
#\_ #####\
~\###|  https://aws.amazon.com/linux/amazon-linux-2023
~\#/__>
~\_\_/
~\_\_/_/
~\_\_/_/
[ec2-user@ip-10-0-2-133 ~]$ ls
[ec2-user@ip-10-0-2-133 ~]$
```

**Task 3:** Write a short note on S3. What is difference between Static website and Dynamic website?

Create a static website using S3 with minimum 3 pages having images and videos.

**Ans:-**

## The Emergence of the AWS S3 Data Lake



Amazon Simple Storage Service (S3) is a highly scalable, durable, and secure object storage service offered by Amazon Web Services (AWS). It is designed to store and retrieve any amount of data from anywhere on the web. Here's a short note on S3:

### **Amazon S3: Simple Storage Service**

Amazon S3 is a versatile and widely used cloud storage service that provides businesses and developers with a reliable and scalable solution for storing and retrieving data. **Key features and characteristics of Amazon S3 include:**

#### **1. Scalability:**

- S3 is built to scale, accommodating data ranging from a few bytes to petabytes. Its architecture allows for easy expansion as your storage needs grow.

#### **2. Durability and Reliability:**

- S3 ensures high durability by replicating data across multiple geographically dispersed data centers. This replication strategy helps protect against data loss and provides a reliable storage solution.

#### **3. Object-Based Storage:**

- S3 is an object storage service, meaning it stores data as objects rather than as blocks like traditional file storage systems. Each object consists of data, a unique key, and metadata.

#### **4. Universal Accessibility:**

- S3 provides a simple and intuitive web interface, as well as a comprehensive set of APIs, allowing users to access their stored data from anywhere on the internet. This makes it a suitable solution for various applications and workflows.

#### **5. Data Lifecycle Management:**

- S3 allows you to define lifecycle policies to automatically transition objects between storage classes or delete them when they are no longer needed. This feature helps optimize storage costs and efficiently manage data over its lifecycle.

#### **6. Security and Access Control:**

- S3 offers robust security features, including access control lists (ACLs) and bucket policies to regulate access at both the bucket and object levels. Additionally, AWS Identity and Access Management (IAM) can be used to define fine-grained access controls.

#### **7. Versioning:**

- S3 supports versioning, enabling users to preserve, retrieve, and restore every version of every object stored in a bucket. This feature is valuable for data recovery and compliance purposes.

#### **8. Data Transfer Acceleration:**

- Amazon S3 Transfer Acceleration enables faster uploads to and downloads from S3 by utilizing the CloudFront global content delivery network.

#### **9. Integration with Other AWS Services:**

- S3 seamlessly integrates with various AWS services, such as AWS Lambda, Amazon CloudFront, and AWS Glacier, allowing users to build comprehensive and scalable solutions.

#### **10. Cost-Effective Storage:**

- S3 provides multiple storage classes with different performance and cost characteristics, allowing users to choose the appropriate class based on their data access patterns and budget constraints.



<b>Static website</b>	<b>Dynamic website</b>
A static website contains information that does not change. It remains the same, or static, for every viewer of the site.	A dynamic website contains information that changes, depending on the viewer, the time of the day, the time zone, the viewer's native language, and other factors.
It requires HTML, CSS and JavaScript.	It requires HTML, CSS and JavaScript as well as PHP or ASP.Net or JSP and database MySQL, SQL,Oracle.
Easy to develop and a bit experienced people can develop it.	Websites not easy to develop because require qualify developers to develop it.
It never uses database connectivity.	It deals with database and generate the contents dynamically using database queries.
Websites are highly secure than dynamic sites because it behaves as a half duplex approach so only one way communication is possible i.e. server to client.	Websites are less secure because it behaves as full duplex approach so both side communications is possible so user can change the server data.

The primary difference between static and dynamic websites lies in how the content is generated and delivered to the user:

## 1. **Static Website:**

- **Content:** A static website consists of fixed, unchanging content. The content is manually created and hardcoded into HTML files.
- **Page Generation:** All pages are pre-built and exist as static files on the server. When a user requests a page, the server directly serves the pre-existing HTML, CSS, and other files.
- **Interactivity:** Static websites are typically less interactive, as the content remains the same for all users and does not change based on user input or other factors.
- **Examples:** Brochure websites, personal blogs, and informational sites with content that rarely changes.

## 2. **Dynamic Website:**

- **Content:** Dynamic websites generate content on the fly, often by retrieving data from a database or other sources. Content can vary based on user interactions, preferences, or real-time data.
- **Page Generation:** Web pages are created dynamically in response to user requests. Server-side scripting languages (such as PHP, Python, or Node.js) are often used to generate HTML on the server before sending it to the user's browser.
- **Interactivity:** Dynamic websites can be highly interactive, allowing users to submit forms, log in, access personalized content, and experience a more engaging and customized browsing experience.
- **Examples:** E-commerce sites, social media platforms, forums, and content management systems (CMS) like WordPress, where content is regularly updated and tailored to user interactions.

In summary, the key distinction is that static websites serve fixed, pre-built content to all users, while dynamic websites generate content dynamically, allowing for interactivity, personalization, and real-time updates. The choice between static and dynamic depends on the specific requirements and goals of a website. Static sites are often simpler and faster, while dynamic sites provide more flexibility and user engagement but may require additional server-side processing. Many modern websites use a combination of static and dynamic elements to achieve a balance between performance and interactivity.

## Screen Shot :=>

### ➤ Create A New Bucket

The screenshot shows the AWS Console Home page. On the left, there's a sidebar with 'Recently visited' services: EC2, CloudFormation, S3, CloudFront, AWS Organizations, Elastic Beanstalk, IAM, CloudTrail, Billing and Cost Management, Key Management Service, and Amazon Redshift. Below this is a 'View all services' link. On the right, there's a 'Applications (0)' section with a 'Create application' button and a note that says 'No applications'. It also includes a 'Find applications' search bar and a 'Go to myApplications' link. At the bottom of the page is a Windows taskbar with various pinned icons.

The screenshot shows the Amazon S3 service page. On the left, there's a sidebar with options like Buckets, Access Grants, Access Points, Object Lambda Access Points, Multi-Region Access Points, Batch Operations, IAM Access Analyzer for S3, and Block Public Access settings for this account. There's also a 'Storage Lens' section with Dashboards, Storage Lens groups, and AWS Organizations settings. The main area shows an 'Account snapshot' with a 'View Storage Lens dashboard' button. Below it is a 'General purpose buckets' section with a 'Create bucket' button. A table lists 10 general purpose buckets, each with a name, AWS Region, access level, and creation date. The table includes columns for Name, AWS Region, Access, and Creation date. The buckets listed are: 'aftab-s3-12072001' (US East (N. Virginia) us-east-1, Objects can be public, November 22, 2023, 09:34:10 (UTC+05:30)), 'asm-12072001' (US East (N. Virginia) us-east-1, Objects can be public, November 25, 2023, 08:36:02 (UTC+05:30)), and 'demobucket2-12072001' (US East (N. Virginia) us-east-1, Bucket and objects not public, November 29, 2023, 09:02:30 (UTC+05:30)). At the bottom is a Windows taskbar with various pinned icons.

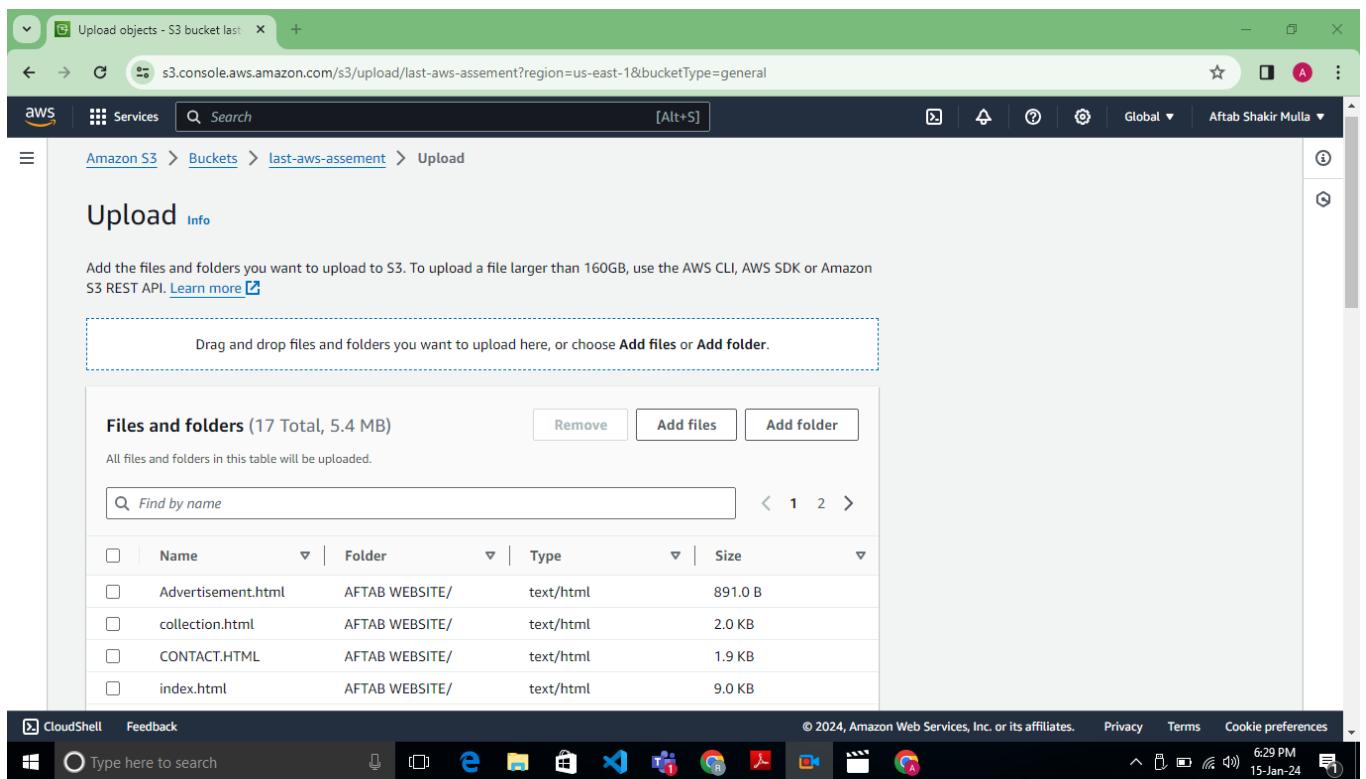
➤ **Name of New bucket(unique name,small letter,Public bucket,ACL enable)**

The screenshot shows the AWS S3 buckets page. A green banner at the top indicates that a bucket named "last-aws-assement" has been successfully created. Below the banner, there's an "Account snapshot" section with a link to "View Storage Lens dashboard". The main area displays two "General purpose buckets": "aftab-s3-12072001" and "asm-12072001". Both buckets are located in the "US East (N. Virginia) us-east-1" region. The first bucket has "Objects can be public" access and was created on November 22, 2023, at 09:34:10 UTC+05:30. The second bucket also has "Objects can be public" access and was created on November 25, 2023, at 08:36:02 UTC+05:30. A "Create bucket" button is visible at the top right of the table.

➤ **Html,CSS,image,video File created in notepad saved Webdevelopment folder in Pc:-**

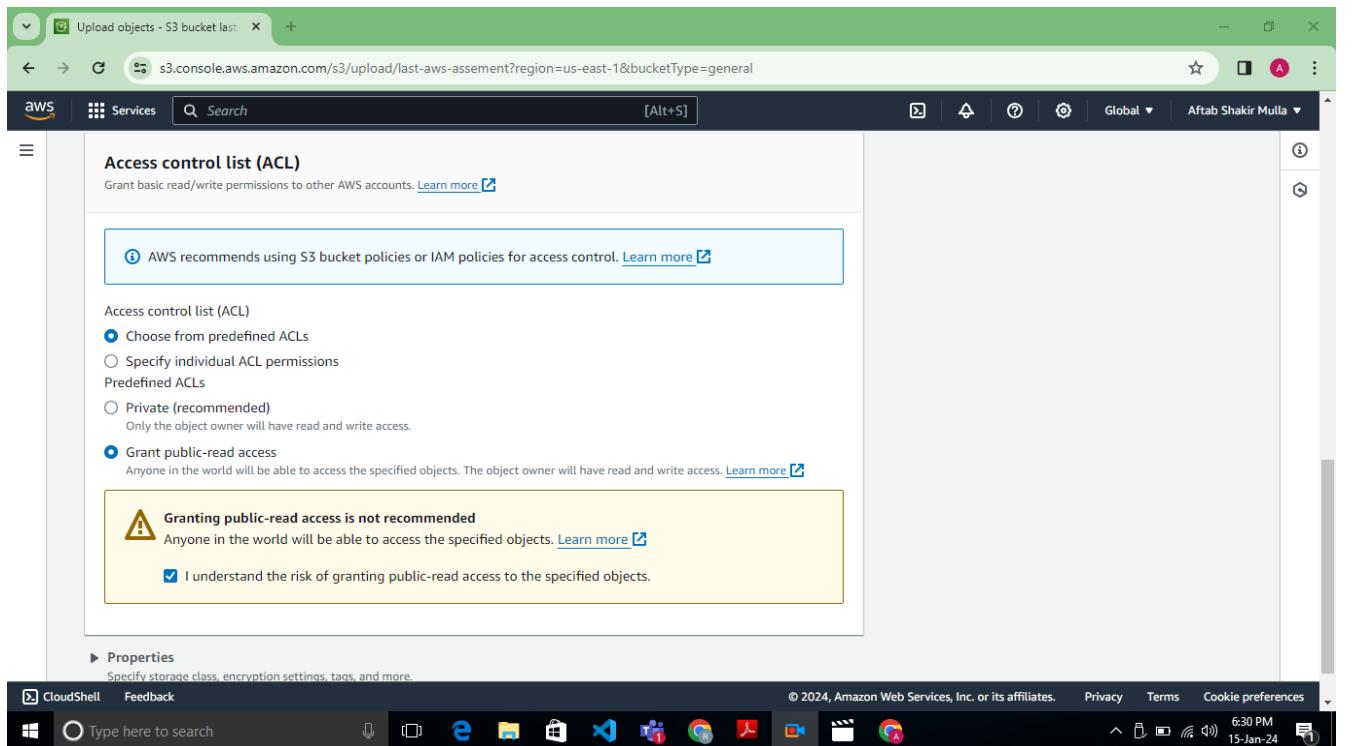
The screenshot shows a Windows File Explorer window with the title bar "AFTAB WEBSITE". The address bar shows the path: This PC > Desktop > WEB DEVELOPMENT > AFTAB WEBSITE. The left sidebar lists drives and folders like OneDrive, This PC, Desktop, Documents, Downloads, Music, Pictures, Videos, and system volumes (Windows 10 (C:), New Volume (D:), etc.). The main pane displays a list of files and folders in the "AFTAB WEBSITE" folder. The files include ".vscode", "Advertisement", "collection", "CONTACT", "index", "men", "nikecollection", "nikecollection2", "nike-shoe1", "nike-shoe2", "nike-womenshoe1", "nike-womenshoe2", "pixels\_videos\_2788 (1080p)", "script", "#style", "#style1", and "women". The "pixels\_videos\_2788 (1080p)" file is a large MP4 file (5,489 KB). The "script" file is a JavaScript source file (1 KB). The "#style" and "#style1" files are CSS source files (3 KB and 1 KB respectively). The "Advertisement", "collection", "CONTACT", "index", "men", "nikecollection", "nikecollection2", "nike-shoe1", "nike-shoe2", "nike-womenshoe1", "nike-womenshoe2", and "pixels\_videos\_2788 (1080p)" files are Chrome HTML documents (1 KB, 3 KB, 2 KB, 10 KB, 2 KB, 11 KB, 8 KB, 4 KB, 3 KB, 4 KB, 4 KB, and 5,489 KB respectively).

## ➤ Upload Web Development Folder:-



The screenshot shows the AWS S3 'Upload' interface. At the top, the URL is s3.console.aws.amazon.com/s3/upload/last-aws-assement?region=us-east-1&bucketType=general. The main area is titled 'Upload' with an 'Info' link. A large box with a dashed border encourages users to 'Drag and drop files and folders you want to upload here, or choose Add files or Add folder.' Below this, a table lists 'Files and folders (17 Total, 5.4 MB)' with columns for Name, Folder, Type, and Size. The table includes buttons for 'Remove', 'Add files', and 'Add folder'. A search bar and pagination controls (1, 2) are also present. The bottom of the screen shows the Windows taskbar with various pinned icons.

## ➤ Public Permission to object:-



The screenshot shows the 'Access control list (ACL)' page for an object. The title is 'Access control list (ACL)' with a note: 'Grant basic read/write permissions to other AWS accounts.' Below this, a box contains the recommendation: 'AWS recommends using S3 bucket policies or IAM policies for access control.' Under 'Access control list (ACL)', there are two options: 'Choose from predefined ACLs' (selected) and 'Specify individual ACL permissions'. Under 'Predefined ACLs', 'Private (recommended)' is selected, with the note: 'Only the object owner will have read and write access.' The 'Grant public-read access' option is also shown, with the note: 'Anyone in the world will be able to access the specified objects. The object owner will have read and write access.' A warning box states: 'Granting public-read access is not recommended' and 'Anyone in the world will be able to access the specified objects.' A checkbox at the bottom accepts this risk. The bottom of the screen shows the Windows taskbar with various pinned icons.

## ➤ Upload successfully:-

The screenshot shows the AWS S3 console interface. At the top, a green header bar indicates "Upload succeeded" with a link to "View details below." Below this, a section titled "Upload: status" displays summary information. It shows a table with two rows: "Destination" (s3://last-aws-assement) with "Succeeded" (17 files, 5.4 MB (100.00%)) and "Failed" (0 files, 0 B (0%)). Below the table, tabs for "Files and folders" and "Configuration" are visible, with "Files and folders" being the active tab. A sub-section titled "Files and folders (17 Total, 5.4 MB)" lists the uploaded files. The bottom of the screen shows a Windows taskbar with various pinned icons.

## ➤ Copy URL:-

The screenshot shows the AWS S3 console displaying the details of a specific object, "index.html". The top navigation bar shows the path: Amazon S3 > Buckets > last-aws-assemement > AFTAB WEBSITE/ > index.html. On the right, there are buttons for "Copy S3 URI", "Download", "Open", and "Object actions". Below these buttons, the "Properties" tab is selected. The "Object overview" section contains detailed information about the object, including its owner (aftab.smullaa), AWS Region (US East (N. Virginia) us-east-1), Last modified (January 15, 2024, 18:31:20 (UTC+05:30)), Size (9.0 KB), and Type (html). To the right of this information, there are links for "S3 URI" (s3://last-aws-assemement/AFTAB WEBSITE/index.html), "Amazon Resource Name (ARN)" (arn:aws:s3:::last-aws-assemement/AFTAB WEBSITE/index.html), "Entity tag (Etag)" (ff7aaaf94758c3e1ea9d94f040911def), and "Object URL" (<https://last-aws-assemement.s3.amazonaws.com/AFTAB+WEBSITE/index.html>). The bottom of the screen shows a Windows taskbar with various pinned icons.

Screenshot of the AWS S3 console showing the properties of an object named 'AFTAB WEBSITE/index.html'. The object was last modified on January 15, 2024, at 18:31:20 (UTC+05:30). It has a size of 9.0 KB and is of type html. The key is 'AFTAB WEBSITE/index.html'. A tooltip indicates that the 'Object URL Copied'.

Owner: aftab.smulaa  
AWS Region: US East (N. Virginia) us-east-1  
Last modified: January 15, 2024, 18:31:20 (UTC+05:30)  
Size: 9.0 KB  
Type: html  
Key: AFTAB WEBSITE/index.html

Object management overview: The following bucket properties and object management configurations impact the behavior of this object.

➤ OutPut:-

✓ Nike webpage (Index.html):-

Screenshot of a Nike website (last-aws-assesments3.amazonaws.com/AFTAB+WEBSITE/index.html) showing a limited edition collection. The page features a banner for the 'NIKE LIMITED EDITION' and sections for men's and women's basketball shoes.

Explore The Limited Edition

## NIKE LIMITED EDITION

Move, Shop, Customise & Celebrate With Us.

Inspiring the world's athletes, Nike delivers innovative products, experiences and services.

SHOP NOW!

Men's Basketball Shoes: LeBron Soldier 12(Team)

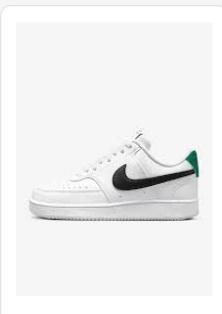
Women's Shoes: Nike Cortez

Men's Shoes: Kyrie 3

Women's Basketball Shoes: LeBron XX

✓ Nike Men web pag(men.html) :-

The screenshot shows a web browser window with three tabs at the top: "Upload objects - S3 bucket last...", "AFTAB WEBSITE/index.html - O...", and "Nike Men". The "Nike Men" tab is active. The page title is "Nike Men". Below the title, there is a section titled "Featured Products" containing two items:

-  Nike Air Max  
\$129.99
-  Nike React Element  
\$99.99

At the bottom of the page, there is a dark footer bar with the text "© 2024 Nike Men". Below the footer, the Windows taskbar is visible, showing various pinned icons and the date/time "15-Jan-24 6:36 PM".

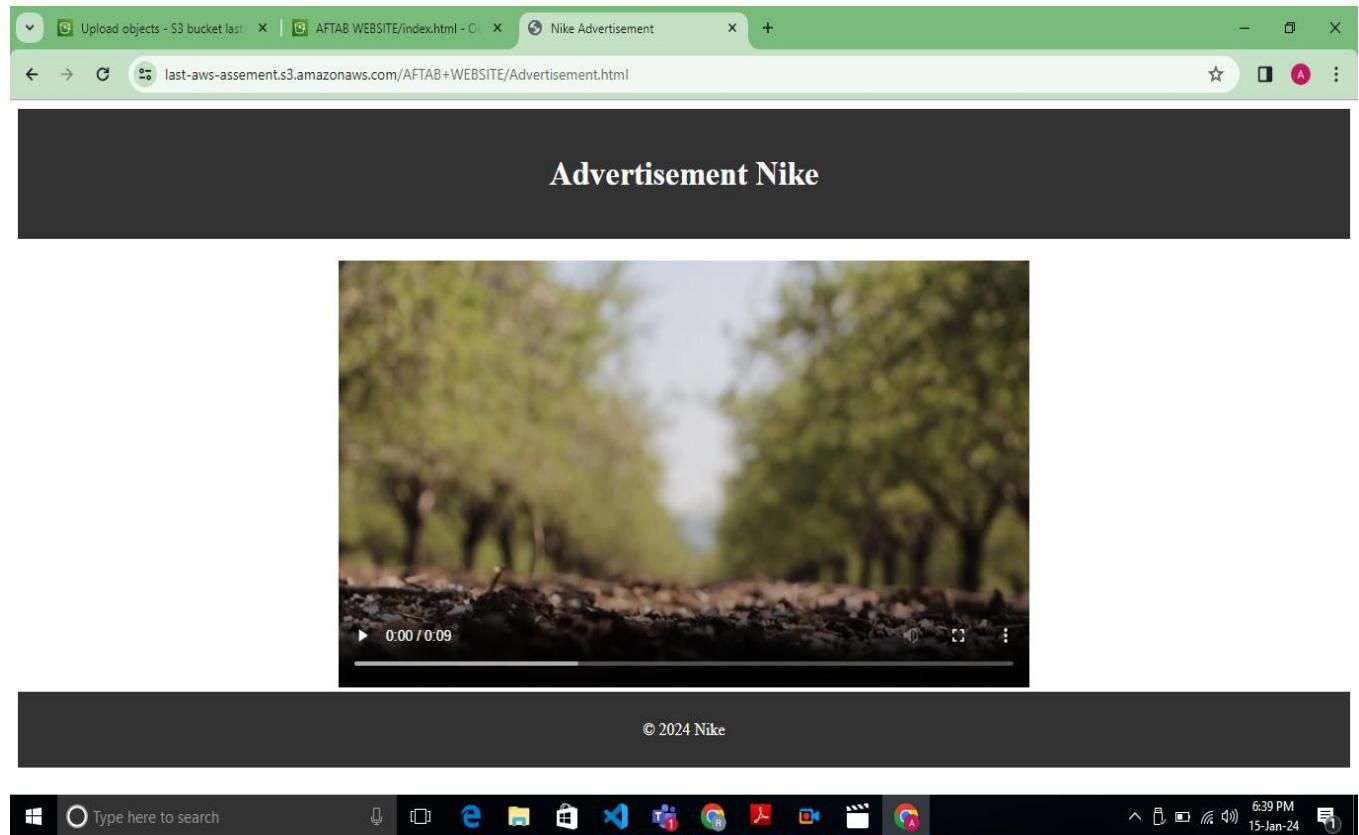
✓ Nike Women webpag(women.html) :-

The screenshot shows a web browser window with three tabs at the top: "Upload objects - S3 bucket last...", "AFTAB WEBSITE/index.html - O...", and "Nike Men". The "Nike Men" tab is active. The page title is "Nike Women". Below the title, there is a section titled "Featured Products" containing two items:

-  Nike Air Max  
\$129.99
-  Nike React Element  
\$99.99

At the bottom of the page, there is a dark footer bar with the text "© 2024 Nike Women". Below the footer, the Windows taskbar is visible, showing various pinned icons and the date/time "15-Jan-24 6:37 PM".

## **✓ Nike Adverstiment Page with video(Advertisement.html) :-**



### **Task 4: What is Instance Refresh?**

Demonstrate the use of Instance Refresh.

Or

Company is using an application running on Apache with auto scaling server and now they want to use Nginx web server with auto scaling. Now Company want to use newer version of application without downtime.

Maximum capacity of server- 4

Minimum capacity of server- 2

### **ANS:**

Instance Refresh is a concept commonly associated with cloud computing services, particularly with Infrastructure as a Service (IaaS) providers like Amazon Web Services (AWS).

In the context of AWS, Instance Refresh refers to a feature provided by AWS Auto Scaling Groups. An Auto Scaling Group helps you ensure that a specified number of instances (Amazon EC2 instances, in the case of AWS) are running at all times. These instances can automatically increase or decrease based on defined conditions.

Instance Refresh is a mechanism within Auto Scaling Groups that allows you to replace instances in a way that ensures minimal downtime and maintains the desired capacity of your group. The primary

use case for Instance Refresh is to update instances with a new Amazon Machine Image (AMI), which might contain security patches, application updates, or other changes.

Here's a general outline of how Instance Refresh works:

**1. Create a New Launch Configuration/Template:** You define a new launch configuration or template with the desired changes (such as a new AMI).

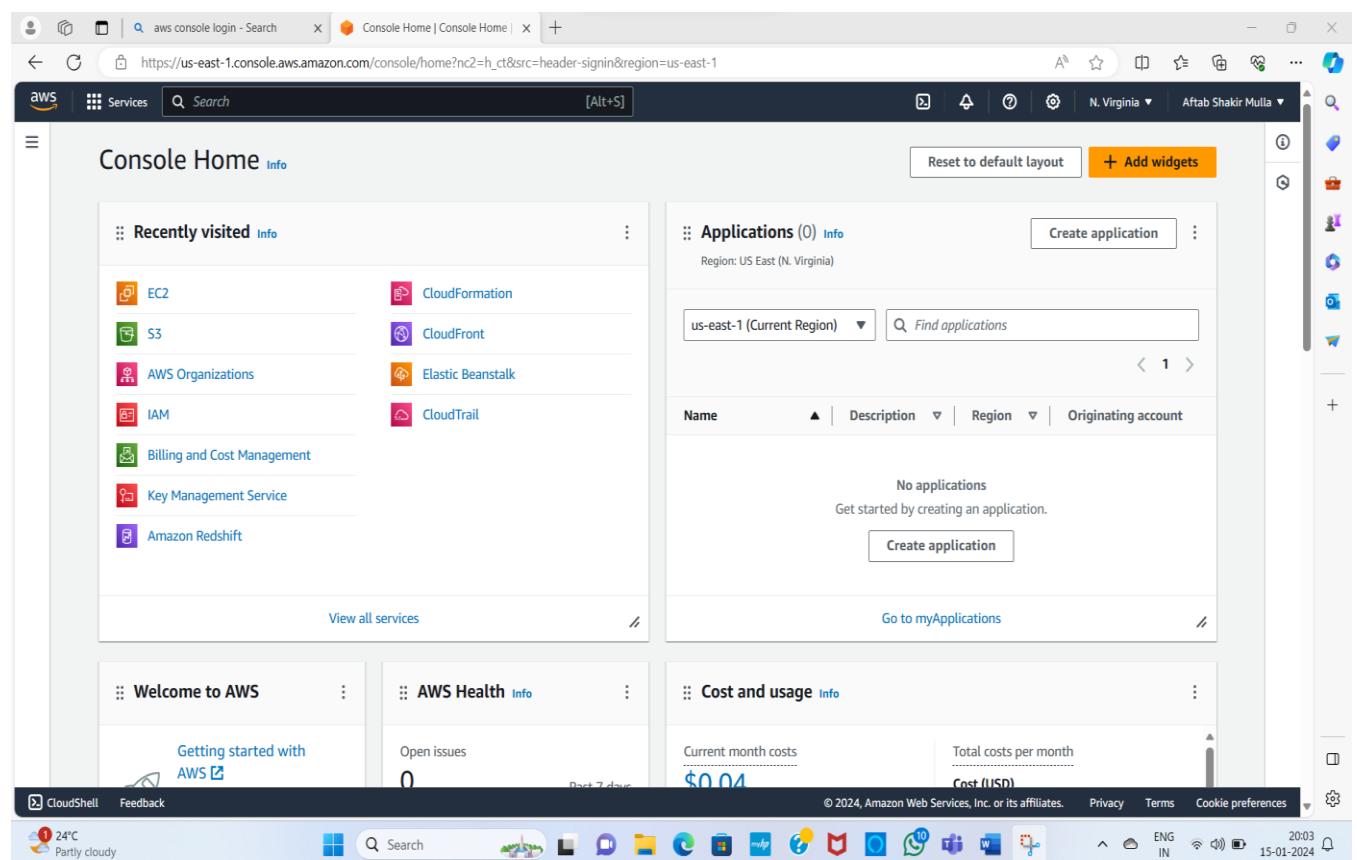
**2. Start Instance Refresh:** You initiate the Instance Refresh process within your Auto Scaling Group. This process automatically replaces instances in the group with instances launched from the new configuration or template.

**3. Terminate Old Instances:** The old instances are gradually terminated and replaced with new instances. Auto Scaling ensures that the desired capacity is maintained during this process.

**4. Monitoring and Rollback:** The process is monitored, and if any issues arise (e.g., failed launches or health checks), Auto Scaling can automatically rollback to the previous configuration to maintain the group's capacity.

This allows you to perform updates to your instances in a controlled and automated manner, reducing the potential impact on your application's availability. Instance Refresh helps in maintaining the health and security of your infrastructure while minimizing disruptions. Keep in mind that the specific details of Instance Refresh may vary between cloud providers and their respective services.

## Screen Shot :=>



## Company Apache With Auto Scaling Server

The screenshot shows the AWS EC2 Auto Scaling Groups page. At the top, there are four tabs: 'Launch an instance', 'Auto Scaling groups', 'Create launch template', and 'Create Auto Scaling group'. Below the tabs, a search bar says 'Search your Auto Scaling groups'. A table lists one Auto Scaling group:

Name	Launch template/configuration	Instances	Status	Desired capacity	Min	Max	A...
assement	apache   Version Default	2	-	2	2	4	us...

Below the table, the 'Auto Scaling group: assement' details are shown. The 'Details' tab is selected. The 'Group details' section includes:

Auto Scaling group name assement	Desired capacity 2	Desired capacity type Units (number of instances)	Amazon Resource Name (ARN) arn:aws:autoscaling:us-east-1:331102913523:autoScalingGroup:e34ee135-f131-49fd-h87
-------------------------------------	-----------------------	--	--

The 'Launch configurations' tab is also visible at the top of the page.

The screenshot shows the AWS EC2 Auto Scaling Groups page, identical to the first one but with updated capacity settings. The 'Launch configurations' tab is now selected at the top.

The 'Group details' section now includes:

Auto Scaling group name assement	Desired capacity 2	Desired capacity type Units (number of instances)	Amazon Resource Name (ARN) arn:aws:autoscaling:us-east-1:331102913523:autoScalingGroup:e34ee135-f131-49fd-b877-1cc81687ef78:autoScalingGroupName/assement
Date created Mon Jan 15 2024 20:16:48 GMT+0530 (India Standard Time)	Minimum capacity 2	Status -	
	Maximum capacity 4		

The 'Launch template' tab is also visible at the bottom of the page.

The screenshot shows the AWS CloudShell interface with the AWS Management Console open. The browser tab bar includes 'aws console login - Search', 'Launch an instance | EC2 | us-east-1', 'Auto Scaling groups | EC2 | us-east-1', and 'Create launch template | EC2 | us-east-1'. The main content area displays the 'Auto Scaling groups' page with one group named 'assement'. The 'Launch template' details for this group are shown, including the AMI ID (ami-0b9e65074c0709eb8), Instance type (t2.micro), Owner (arn:aws:iam::331102913523:root), and Create time (Mon Jan 15 2024 20:13:28 GMT+0530 (India Standard Time)). The CloudShell toolbar at the bottom shows various icons for file operations and help.

## Launch A New Instance With Nginx Webserver

The screenshot shows the AWS CloudShell interface with the AWS Management Console open. The browser tab bar includes 'aws console login - Search', 'EC2 | us-east-1', 'Auto Scaling groups | EC2 | us-east-1', and 'Create launch template | EC2 | us-east-1'. The main content area displays the 'Launch an instance' page. In the 'Name and tags' section, the name is set to 'Nginx server'. Under 'Application and OS Images (Amazon Machine Image)', the software image (AMI) is selected as 'Amazon Linux 2023.3.2...'. The instance type is 't2.micro', and the security group is 'linux demo'. A storage volume of 1 volume(s) - 8 GiB is selected. A summary box indicates a free tier usage of 750 hours of t2.micro or t3.micro instances per month. The 'Launch instance' button is prominently displayed at the bottom right.

## Install a Nginx webserver and Enable

The screenshot shows a terminal session in AWS CloudShell. The user is installing Nginx using the command `sudo dnf install nginx -y`. The output shows the package details and dependencies being resolved. The transaction summary indicates the successful installation of the Nginx server.

```
[ec2-user@ip-172-31-90-0 ~]$ sudo dnf install nginx -y
Last metadata expiration check: 0:00:51 ago on Mon Jan 15 14:57:17 2024.
Dependencies resolved.

Transaction Summary

  Package          Architecture Version      Repository  Size
Installing:
  nginx           x86_64      1:1.24.0-1.amzn2023.0.2  amazonlinux 32 k
Installing dependencies:
  generic-logos-httd noarch      18.0.0-12.amzn2023.0.3  amazonlinux 19 k
  gperf-tools-libs x86_64      2.9.1-1.amzn2023.0.3  amazonlinux 308 k
  libunwind        x86_64      1.4.0-5.amzn2023.0.2  amazonlinux 66 k
  nginx-core       x86_64      1:1.24.0-1.amzn2023.0.2  amazonlinux 586 k
  nginx-filesystem noarch      1:1.24.0-1.amzn2023.0.2  amazonlinux 9.1 k
  nginx-mimetypes noarch      2.1.49-3.amzn2023.0.3  amazonlinux 21 k

Transaction Summary

  i-000672cb579c90b36 (Nginx server)
  PublicIPs: 44.201.85.58  PrivateIPs: 172.31.90.0
```

## Create a AMI Of Nginx Server

The screenshot shows the AWS EC2 Instances page. A context menu is open over the running instance named "Nginx server". The "Actions" menu is expanded, showing options like "Connect", "View details", "Manage instance state", "Instance settings", "Networking", "Security", "Image and templates", and "Monitor and troubleshoot". The "Image and templates" option is highlighted. A sub-menu under "Image and templates" shows "Create image", "Create template from instance", and "Launch more like this".

The screenshot shows the 'Create image' page in the AWS EC2 console. The instance ID is [i-000672cb579c90b36](#) (Nginx server). The image name is set to 'Nginx'. There is an optional image description field. The 'No reboot' checkbox is checked. Under 'Instance volumes', an EBS volume is selected with a size of 8 GiB, a General Purpose SSD volume type, and a throughput of 3000 MiB/s. The 'Delete on termination' and 'Encrypted' checkboxes are checked. The CloudShell and Feedback buttons are at the bottom.

**AMI CREATED SUCCESSFULLY**

The screenshot shows the 'Amazon Machine Images (AMIs)' page in the AWS EC2 console. The newly created AMI is listed with the following details:

Name	AMI name	AMI ID	Source	Owner
NGINX	Nginx	ami-08d74fccd7a536840	331102913523/Nginx	331102913523

The AMI ID is **ami-08d74fccd7a536840 (NGINX)**. The Details tab shows the following configuration:

AMI ID	Image type	Platform details	Root device type
ami-08d74fccd7a536840 (NGINX)	machine	Linux/UNIX	EBS
AMI name	Owner account ID	Architecture	Usage operation
Nginx	331102913523	x86_64	RunInstances
Root device name	Status	Source	Virtualization type
/dev/xvda	Pending	331102913523/Nginx	hvm
Boot mode	State reason	Creation date	Kernel ID
uefi-preferred	-	Mon Jan 15 2024 20:31:21 GMT+0530	-

## Create A Launch Template (Give a Nginx AMI,AZ,VPC,SG,Key pair) :

The screenshot shows the 'Create launch template' wizard on the AWS EC2 console. The current step is 'Launch template name and description'. The 'Launch template name - required' field contains 'Nginx'. The 'Template version description' field contains 'A prod webserver for MyApp'. Under 'Auto Scaling guidance', there is a checked checkbox 'Provide guidance to help me set up a template that I can use with EC2 Auto Scaling'. A tooltip for the free tier is visible, stating: 'Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 30 GiB of EBS storage, 2 million IOs, 1 GB of snapshots, and 100 GB of bandwidth to the internet.' At the bottom right of the wizard is a 'Create launch template' button.

The screenshot shows the 'Create launch template' wizard on the AWS EC2 console. The current step is 'Application and OS Images (Amazon Machine Image) - required'. It shows a search bar and two filter tabs: 'Owned by me' (selected) and 'Shared with me'. Below the search bar is a table for the selected AMI, 'Nginx'. The table includes columns for AMI ID, Creation Date, Virtualization, ENA enabled, and Root device type. The AMI ID is 'ami-08d74fccd7a536840', Creation Date is '2024-01-15T15:01:21.000Z', Virtualization is 'hvm', ENA enabled is 'true', and Root device type is 'ebs'. The 'Description' and 'Architecture' sections are also present. A tooltip for the free tier is visible, stating: 'Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 30 GiB of EBS storage, 2 million IOs, 1 GB of snapshots, and 100 GB of bandwidth to the internet.' At the bottom right of the wizard is a 'Create launch template' button.

The screenshot shows the AWS EC2 'Create launch template' wizard. On the left, the 'Instance type' section is selected, showing a t2.micro instance (Free tier eligible) with details like Family: t2, 1 vCPU, 1 GiB Memory, Current generation: true, and various On-Demand pricing options. Below it, the 'Key pair (login)' section shows a selected key pair named 'linux'. The 'Network settings' section includes a subnet selection dropdown set to 'Don't include in launch template'. On the right, the 'Summary' pane lists the chosen configurations: Software Image (AMI) as Nginx ami-08d74fccc7a536840, Virtual server type (instance type) as t2.micro, Firewall (security group) as 'linux demo', and Storage (volumes) as 1 volume(s) - 8 GiB. A tooltip for the Free tier indicates it covers 750 hours of usage per month. At the bottom right is a 'Create launch template' button.

## Successfully created a nginx launch template :

The screenshot shows the AWS EC2 'Launch templates' page. A green success message at the top states 'Successfully created Nginx(lt-0b27ca2b1e2dab635).'. Below it, the 'Next Steps' section provides links to 'Launch an instance', 'Create an Auto Scaling group from your template', 'Create Auto Scaling group', and 'Create Spot Fleet'. The 'Launch an instance' link is described as launching an On-Demand instance from the template. The 'Create an Auto Scaling group from your template' link is described as using Auto Scaling to maintain application availability. The 'Create Auto Scaling group' link is described as helping to scale capacity up or down automatically. The 'Create Spot Fleet' link is described as using Spot Instances for cost savings. The bottom of the screen shows the standard AWS navigation bar with CloudShell, Feedback, and various icons.

## Start instance refresh

The screenshot shows the 'Start instance refresh' page in the AWS EC2 console. On the left, a sidebar lists various services like Images, Elastic Block Store, Network & Security, Load Balancing, and Auto Scaling. The main content area is titled 'Start instance refresh' and includes sections for 'Availability settings - new' and 'Instance replacement method'. Under 'Instance replacement method', three options are shown: 'Prioritize availability' (radio button), 'Launch before terminating' (radio button selected), and 'Control costs' (radio button). The 'Control costs' section is highlighted with a blue border. Below these, there's a 'Set healthy percentage' section with a slider from 90% to 100% and a note about scaling limits.

## Give Nginx Launch Template:

The screenshot shows the 'Give Nginx Launch Template' page in the AWS EC2 console. The sidebar is identical to the previous screenshot. The main content area is titled 'Desired configuration - optional' and includes tabs for 'Use console interface' (selected) and 'Use code editor'. A checkbox for 'Update launch template' is checked. Below it, a 'Launch template' section shows a dropdown set to 'Nginx' and a 'Create a launch template' button. A 'Version' dropdown is set to '1'. The main table displays configuration details: Description (empty), Launch template (Nginx), Instance type (t2.micro); AMI ID (ami-08d74fcfd7a536840), Security groups (-), Request Spot Instances (No); Key pair name (linux), Security group IDs (sg-0cb2c27e289e31c98).

aws console login - EC2 | us-east-1 Instances | EC2 | us- Auto Scaling group | EC2 | us-east-1 Auto Scaling group | Create launch temp | +

https://us-east-1.console.aws.amazon.com/ec2/v2/home?region=us-east-1#AutoScalingGroupDetails:id=assement;view=instanceRefresh

aws Services Search [Alt+S]

New Images AMIs AMI Catalog Elastic Block Store Volumes Snapshots Lifecycle Manager Network & Security Security Groups Elastic IPs Placement Groups Key Pairs Network Interfaces Load Balancing Load Balancers Target Groups Trust Stores New Auto Scaling Auto Scaling Groups CloudShell Feedback

23°C Partly cloudy

Instance refresh started successfully

EC2 > Auto Scaling groups > assement

## assement

Details Activity Automatic scaling Instance management Monitoring Instance refresh

Active instance refresh Info

Instance refresh ID f7caf13a-bd44-4745-a391-ffa2689d6560	Minimum healthy percentage 90%	Skip matching Enabled	Desired configuration View
Status 2 instances left to update	Instance warmup 0% 300 seconds	Scale-in protected instances Ignore	Auto rollback Disabled
Start time 2024 January 15, 08:42:13 PM +05:30	Checkpoints -	Standby instances Ignore	CloudWatch alarm -

Actions Start instance refresh

Instance refresh history (1) Info

Filter instance refresh history
---------------------------------

© 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

CloudShell Feedback

2042 ENG IN 15-01-2024

This screenshot shows the AWS EC2 Auto Scaling Groups page for the 'assement' group. A green banner at the top indicates that an instance refresh has started successfully. The 'Instance refresh' tab is selected. The 'Active instance refresh' section displays details such as the ID, minimum healthy percentage (90%), and status (2 instances left to update). The 'Instance refresh history' section shows one entry: a successful refresh completed at 100%.

aws console login - EC2 | us-east-1 Instances | EC2 | us- Auto Scaling group | EC2 | us-east-1 Auto Scaling group | Create launch temp | +

https://us-east-1.console.aws.amazon.com/ec2/v2/home?region=us-east-1#AutoScalingGroupDetails:id=assement;view=instanceRefresh

aws Services Search [Alt+S]

EC2 Dashboard EC2 Global View Events Console-to-Code Preview Instances Instances Instance Types Launch Templates Spot Requests Savings Plans Reserved Instances Dedicated Hosts Capacity Reservations New Images AMIs AMI Catalog Elastic Block Store Volumes Snapshots Lifecycle Manager CloudShell Feedback

15°C Partly cloudy

assement

Details Activity Automatic scaling Instance management Monitoring Instance refresh

Active instance refresh Info

No active instance refresh

Start an instance refresh to perform rolling updates on the Auto Scaling group's instances. Only one instance refresh can be active at a time.

Start instance refresh

Instance refresh history (1) Info

Filter instance refresh history
---------------------------------

Instance refresh ID	Instance refresh status	Rollback status	Status reason	Percentage completed
f7caf13a-bd44-4745-a391-ffa2689d6560	Successful	-	-	100%

© 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

CloudShell Feedback

ENG IN 15-01-2024

This screenshot shows the same AWS EC2 Auto Scaling Groups page as the previous one, but with a different state. The green banner at the top is no longer present, indicating that the instance refresh has completed. The 'Active instance refresh' section now says 'No active instance refresh'. The 'Instance refresh history' section shows the same successful entry as before, but it is now listed under the 'Completed' section.

**Task 5:** Create a architectural diagram of uses of cloud front.

