Practical No. 1

Q1. Explain how to create a key pair for accessing your EC2 instance. What is the purpose of the key pair, and how does it enhance the security of your instance?

Steps to Create a Key Pair for EC2:

1. **Log in to AWS Console**: Go to AWS Management Console.
2. **Navigate to EC2 Dashboard**: Select **EC2** from the list of services.
3. **Create a New Key Pair**:
    - On the left-hand menu, click **Key Pairs** under **Network & Security**.
    - Click **Create Key Pair**.
    - Enter a name for your key pair.
    - Choose **RSA** or **ED25519** as the key pair type.
    - Click **Create Key Pair** and the `.pem` file will be downloaded (this file is your private key).
4. **Assign Key Pair to EC2 Instance**:
    - When launching a new EC2 instance, under **Key pair (login)**, select your newly created key pair from the dropdown or create one during the launch process.
5. **SSH into Your EC2 Instance**:
    - Use the downloaded key pair `.pem` file to SSH into your instance:

```
bash
Copy code
ssh -i /path_to_key_pair.pem ec2-user@your_instance_public_IP
```

Purpose and Security Enhancement:

- **Key Pair Purpose**: The key pair consists of a **public** and a **private** key. The public key is stored on the EC2 instance, while the private key remains with you. You use the private key to securely SSH into the instance.
- **Security Enhancement**: Password-based login can be vulnerable to brute-force attacks. Using key pairs eliminates the need for passwords, enhancing security by allowing only users with the correct private key to access the instance.

Q2.For security reasons, you want to limit SSH access to your EC2 instance to only your office IP address. How would you modify the inbound rules of your security group to achieve this? What steps would you follow to ensure that only your office IP address can connect via SSH?

Steps to Modify Inbound Rules for Security Group:

6. **Log in to AWS Console**: Go to AWS Management Console.
7. **Navigate to EC2 Dashboard**: Select **EC2** from the list of services.
8. **Access Security Groups**:
    - On the left-hand menu, click **Security Groups** under **Network & Security**.
    - Find and select the security group associated with your EC2 instance.
9. **Modify Inbound Rules**:
    - Click the **Inbound Rules** tab.
    - Click **Edit inbound rules**.
    - Remove any existing **SSH (port 22)** rule that allows access from anywhere (like `0.0.0.0/0`).
    - Add a new rule:
        - **Type**: SSH
        - **Protocol**: TCP
        - **Port Range**: 22
        - **Source**: Choose **My IP** (AWS will automatically detect your current IP), or manually enter your **office IP address** (e.g., `203.0.113.0/32`).
10. **Save Changes**: Click **Save rules** to apply the changes.

Ensuring Security:

- **Office IP Restriction**: By only allowing connections from your office IP address, you significantly reduce the attack surface. Only users from your specified IP can access the EC2 instance over SSH.
- **Double-check IP**: Ensure that your office IP is static or update the rule accordingly if it changes, otherwise you may get locked out.

Practical no. 2

Q1. Launch a Windows Server Amazon EC2 instance and connect using Windows Remote Desktop. [15M]

Steps to Launch a Windows Server EC2 Instance:

11. **Log in to AWS Console**: Go to AWS Management Console.
12. **Navigate to EC2 Dashboard**: Select **EC2** from the list of services.
13. **Launch Instance**:
    - Click **Launch Instance**.
    - Under **Amazon Machine Image (AMI)**, select a **Windows Server** version (e.g., **Windows Server 2019**).
    - Choose an appropriate instance type (e.g., **t2.micro** for free tier).
14. **Configure Instance Settings**:
    - Select the **VPC** and **subnet**.
    - Ensure **Auto-assign Public IP** is enabled.
15. **Create or Use Key Pair**:
    - Either select an existing key pair or create a new one.
    - Download the private key `.pem` file if creating a new pair.
16. **Launch the Instance**:
    - Review settings and click **Launch**.

Steps to Connect Using Remote Desktop:

17. **Obtain Instance's Public IP**:
    - Once the instance is running, navigate to the **Instances** page, and copy the **Public IPv4 address**.
18. **Retrieve Windows Administrator Password**:
    - On the **Instances** page, right-click your instance and choose **Get Windows Password**.
    - Upload the `.pem` key file you created earlier and click **Decrypt Password**. AWS will show the decrypted password.
19. **Connect Using Remote Desktop**:
    - Open **Remote Desktop Connection** (RDP) on your local machine.
    - Enter the **Public IP** of the EC2 instance.

- Use the decrypted **Administrator password** to log in.

Q2. How do you create an IAM policy that grants full access to EC2 instances but only allows starting and stopping instances?

Steps to Create the IAM Policy via the AWS Management Console:

20. **Log in to AWS Console**: Go to AWS Management Console.
21. **Navigate to IAM Dashboard**: Select **IAM** from the list of services.
22. **Create New Policy**:
    - In the left-hand menu, click **Policies**.
    - Click **Create Policy**.
23. **Select EC2 Service**:
    - In the **Visual editor**, choose **EC2** under **Service**.
24. **Set Allowed Actions**:
    - Under **Actions**, choose **Specific actions**.
    - In the search bar, type and select the following actions:
      - **StartInstances**
      - **StopInstances**
      - **DescribeInstances**
      - **DescribeInstanceStatus**
    - This limits the policy to only starting and stopping EC2 instances and allows users to describe the instances.
25. **Specify Resources**:
    - Under **Resources**, select **All resources** to apply these permissions to all EC2 instances.
26. **Review and Create**:
    - Click **Next: Tags**, then **Next: Review**.
    - Name the policy (e.g., `EC2StartStopPolicy`) and provide a description.
    - Click **Create policy**.

Attach Policy to User/Group:

27. **Assign Policy**:
    - Go to **Users** or **Groups** under IAM.
    - Select the user or group, then click **Add permissions**.
    - Search for the newly created policy and attach it.

This policy will now restrict the user to **only start and stop instances**, while allowing them to view instance information, without any need for writing JSON code.

Practical No. 3

Q1. Create a custom AMI from your configured EC2 instance. What steps would you take to delete an AMI? .

Steps to Create a Custom AMI:

28. **Log in to AWS Console**: Go to AWS Management Console.
29. **Navigate to EC2 Dashboard**: Select **EC2** from the list of services.
30. **Select EC2 Instance**:
    - Go to **Instances**, and find the EC2 instance you want to create an AMI from.
    - Right-click the instance and select **Image and templates** → **Create image**.
31. **Configure the AMI**:
    - Provide an **Image name** and **Description**.
    - You can optionally select or deselect **No Reboot** (if enabled, the instance will not reboot while creating the AMI).
    - Click **Create Image**.
32. **Monitor AMI Creation**:
    - Go to the **AMIs** section under **Images** in the left-hand menu to see the progress of the AMI creation.

Steps to Delete an AMI:

33. **Navigate to AMIs**:
    - Go to the **AMIs** section under **Images** in the EC2 dashboard.
34. **Select the AMI to Delete**:
    - Select the AMI you wish to delete.
    - Click **Actions** → **Deregister**.
    - Confirm the deregistration.
35. **Delete Associated Snapshots** (Optional):
    - AMIs use **EBS snapshots**. If you want to remove the associated snapshots:
        - Go to the **Snapshots** section under **Elastic Block Store**.
        - Select the snapshot related to the AMI and delete it.

Q2. Create a two S3 bucket in AWS and perform the following operation. Upload files to an S3 bucket Download a bucket item. Copy a bucket item to another bucket.

Steps to Create Two S3 Buckets:

36. **Log in to AWS Console**: Go to AWS Management Console.
37. **Navigate to S3 Dashboard**: Select **S3** from the list of services.
38. **Create the First Bucket**:
    - Click **Create bucket**.
    - Enter a unique bucket name (e.g., `bucket-1`).
    - Choose a region.
    - Configure other settings as needed (like versioning, public access settings).
    - Click **Create bucket**.
39. **Create the Second Bucket**:
    - Repeat the same process to create a second bucket (e.g., `bucket-2`).

Steps to Upload Files to an S3 Bucket:

40. **Go to Your First Bucket**:
    - In the **S3 Dashboard**, select **bucket-1**.
41. **Upload a File**:
    - Click **Upload** → **Add files**.
    - Select the file(s) from your local machine.
    - Click **Upload** to transfer the file to S3.

Steps to Download an S3 Bucket Item:

42. **Select the Item**:
    - Navigate to **bucket-1** in the **S3 Dashboard**.
    - Find the file you uploaded.
43. **Download the Item**:
    - Right-click the file or click the **Actions** button.
    - Select **Download** to download the file to your local machine.

Steps to Copy a Bucket Item to Another Bucket:

44. **Select the File to Copy**:
    - In **bucket-1**, check the box next to the file you want to copy.
45. **Copy the File**:
    - Click **Actions** → **Copy**.
    - A dialog box will appear where you can choose the destination.
46. **Choose Destination Bucket**:
    - In the destination field, select **bucket-2** (the second bucket you created).
    - Click **Copy**.

Your file will now be copied from **bucket-1** to **bucket-2**.

Practical 4

Q1. How do you create and manage AWS IAM users and groups?

Steps to Create AWS IAM Users:

47. **Log in to AWS Console**: Go to AWS Management Console.
48. **Navigate to IAM**: Select **IAM** from the list of services.
49. **Create a New User**:
    - On the left-hand menu, click **Users**, then click **Add users**.
    - Enter the username.
    - Select **AWS Management Console access** for users that need console access or **Programmatic access** for users using CLI/API.
    - Set a **password** or let AWS generate one.
50. **Assign Permissions**:
    - Choose how to assign permissions:
        - **Attach existing policies directly**: Select predefined policies.
        - **Add user to group**: Select existing groups with predefined permissions.
        - **Copy permissions from existing user**: Copy settings from another user.
51. **Review and Create**:
    - Click **Next: Tags** (optional).
    - Review the configuration and click **Create user**.

Steps to Create and Manage IAM Groups:

52. **Create a Group**:
    - In the IAM dashboard, click **Groups** on the left-hand menu.
    - Click **Create New Group**, name the group, and assign permissions by attaching predefined **policies**.
53. **Add Users to the Group**:
    - Select the group and click the **Add users to group** button.
    - Choose the users you want to assign to the group.

Managing Users and Groups:

- **Attach/Detach Policies**: You can attach or detach permissions from both users and groups anytime by editing their permission settings.
- **Remove Users from Group**: Go to the group, select the user, and remove them from the group.
- **Delete Users or Groups**: You can delete users or groups if they are no longer needed, but ensure that no dependent resources or permissions are impacted.

Q2. You want to ensure that your EC2 instance's data is backed up regularly. What methods would you use to back up data from your EC2 instance? Discuss options such as creating snapshots of EBS volumes and using AWS Backup.

Q2: Backing Up Data from EC2 Instances

1. Using Snapshots of EBS Volumes:

- **What are Snapshots?**: EBS (Elastic Block Store) snapshots are incremental backups of your EBS volumes. They store the changes made since the last snapshot and can be used to restore the volume.

Steps to Create EBS Snapshots:

54. **Log in to AWS Console**: Go to AWS Management Console.
55. **Navigate to EC2 Dashboard**: Select **EC2** from the list of services.
56. **Select EBS Volume**:
    - In the left-hand menu, click **Volumes** under **Elastic Block Store**.
    - Select the EBS volume attached to your EC2 instance.
57. **Create Snapshot**:
    - Select the volume and click **Actions > Create Snapshot**.
    - Provide a description and click **Create Snapshot**.
58. **Restore from Snapshot** (When needed):
    - Go to **Snapshots**, select the snapshot, and click **Actions > Create Volume** to create a new EBS volume from the snapshot.

2. Using AWS Backup:

- **What is AWS Backup?**: AWS Backup is a centralized service that allows you to automate and manage backups across multiple AWS services, including EC2, RDS, and EBS.

Steps to Use AWS Backup:

59. **Navigate to AWS Backup**:
    - In the AWS Console, search for and select **AWS Backup**.
60. **Create a Backup Plan**:
    - Go to **Backup Plans** and click **Create backup plan**.
    - Use a template or define your own plan.
    - Specify backup frequency (daily, weekly, etc.), retention periods, and the resources to back up (e.g., EC2 instances or EBS volumes).
61. **Assign Resources**:
    - Under **Backup Plans**, click **Assign Resources**.
    - Choose **EC2** as the resource type, then select the instance or volume you want to back up.
62. **Automated Backups**:
    - AWS Backup will now automatically take regular backups based on your plan.
    - You can restore backups via the **Backup vault** by selecting the relevant backup and restoring it to the same or a new EC2 instance.

3. Manual Data Backups:

- **S3 Backup**: You can manually back up important data by uploading files to an **S3 bucket**. Use scripts or AWS CLI commands to automate this.
- **Third-Party Solutions**: Some organizations also use third-party backup services integrated with AWS to create periodic backups of EC2 data.

In summary, **EBS snapshots** are ideal for backing up volume data, and **AWS Backup** allows automated and managed backups across multiple AWS services, providing a comprehensive backup solution.

Practical 5

Q1 How can you automate the process of adding or removing SSH public keys to EC2 instances using AWS Systems Manager

Steps:

63. **Ensure SSM Agent is Installed**:
    - Make sure the **SSM Agent** is installed and running on the EC2 instance.
64. **Create a Document**:
    - In the **Systems Manager Console**, create a new **Run Command Document**.
    - Define a script that appends or removes the public SSH key from the `authorized_keys` file.
    - Example for adding a public key:

```
bash
Copy code
echo "ssh-rsa AAAA..." >> ~/.ssh/authorized_keys
```

    - For removing a key:

```
bash
Copy code
sed -i '/ssh-rsa AAAA.../d' ~/.ssh/authorized_keys
```

65. **Execute Run Command**:
    - Go to **Run Command** in Systems Manager, select your document, and run it on the desired EC2 instances.
    - You can schedule this using **State Manager** for automation.

Q2 How can you assign a custom security group to an existing EC2 instance?

Steps:

66. **Go to EC2 Dashboard**:
    - Navigate to **EC2** > **Instances**.
67. **Select the Instance**:
    - Find the EC2 instance you want to modify and click on it.
68. **Modify Security Groups**:

- Under the **Actions** dropdown, select **Networking** > **Change Security Groups**.
- Select the **custom security group** you want to assign.
- Save the changes, and the new security group will be applied to the instance.

Practical 6

Q1. How can you create an IAM policy that allows only read access to S3 buckets?

Q1: Create an IAM Policy That Allows Only Read Access to S3 Buckets

Steps:

69. **Go to IAM Console**:
    - Navigate to the **IAM Dashboard** > **Policies** > **Create Policy**.
70. **Use Policy Generator**:
    - In the JSON editor, write the policy to allow read-only access:

```
json
Copy code
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:GetObject",
        "s3:ListBucket"
      ],
      "Resource": [
        "arn:aws:s3:::your-bucket-name",
        "arn:aws:s3:::your-bucket-name/*"
      ]
    }
  ]
}
```

71. **Save the Policy**:
    - Attach this policy to users or roles that require read-only access to the S3 bucket.

Q2 Create multiple key-pair and use same key-pair for multiple instances.

Steps:

72. **Create a Key Pair**:
    - In **EC2 Dashboard**, go to **Key Pairs** and create a new key pair.
    - Save the private key file (`.pem`).
73. **Launch Instances with the Same Key Pair**:
    - While launching multiple EC2 instances, select the **existing key pair** during the configuration step.
    - This allows you to SSH into all instances using the same key pair.
74. **Use the Same Key-Pair**:
    - When connecting to the instances via SSH, use the same private key for each:

```
bash
Copy code
ssh -i "your-key.pem" ec2-user@instance-public-ip
```

Practical no 7

Q1 2 How do you configure Network Access to an instance using Security groups? [15M]

Steps:

75. **Go to EC2 Dashboard**:
    - Navigate to **EC2** > **Security Groups**.
76. **Create or Edit a Security Group**:
    - Create a new security group or edit an existing one.
77. **Configure Inbound Rules**:
    - Add rules based on your network access needs. For example:
        - **SSH (port 22)**: To allow SSH access, set the source to your IP address or specific CIDR range.
        - **HTTP (port 80)**: To allow web traffic, allow inbound HTTP traffic from `0.0.0.0/0`.
78. **Save and Apply**:
    - Assign this security group to the EC2 instance by modifying its security group settings.

Q2 You have a web server EC2 instance that needs to be secured from unauthorized access while allowing necessary traffic. How would you configure security groups to secure your EC2 instance? Describe the process for creating and applying security group rules to allow only HTTP and SSH traffic while restricting all other access.

Steps:

79. **Go to Security Groups**:
    - In the **EC2 Dashboard**, navigate to **Security Groups**.
80. **Create a New Security Group**:
    - Define **Inbound Rules**:
        - **Allow HTTP** (port 80) from `0.0.0.0/0`.
        - **Allow SSH** (port 22) from your specific IP address.
    - **Outbound Rules**:
        - Allow all outbound traffic (default), or restrict based on your security requirements.

81. **Apply Security Group**:
    - Attach this security group to your web server EC2 instance, and it will only allow HTTP and SSH traffic while denying everything else.

Practical no. 8

Q1 If you have a custom AMI that you want to use to launch new EC2 instances. What are the steps to launch an EC2 instance using your custom AMI? Include details about choosing the AMI, selecting instance types, and configuring the instance settings.

Steps:

82. **Go to EC2 Dashboard**:
    - Navigate to **EC2** > **AMIs**.
83. **Select Your Custom AMI**:
    - Choose the custom AMI you want to use.
84. **Launch Instance**:
    - Click **Launch** from the AMI page.
    - Select the **instance type** (e.g., **t2.micro** for basic use).
85. **Configure Instance Settings**:
    - Choose network, subnet, and enable auto-assign public IP if needed.
    - Add storage as required (EBS).
86. **Select Security Group**:
    - Choose an existing security group or create a new one for HTTP and SSH access.
87. **Launch the Instance**:
    - Assign a key pair for SSH access and click **Launch**.

Q2 How do you configure AWS S3 versioning and lifecycle policies?

Steps:

88. **Enable S3 Versioning**:
    - Go to **S3** and select the bucket.
    - In the **Properties** tab, enable **Versioning**. This will keep multiple versions of objects.
89. **Configure Lifecycle Policies**:
    - In the **Management** tab, click **Lifecycle Rules** and create a new rule.
    - Define actions such as moving objects to **Glacier** after 30 days or **deleting** them after a certain period.
    - Save the policy to automatically manage object lifecycle based on your rules.

Practical no. 9

Q1. If your EC2 instance is not responding to requests and appears to be down. What steps would you take to troubleshoot and diagnose the issue? Include common checks and diagnostic tools you might use to resolve the problem. [15M]

Steps:

90. **Check Instance State**:
    - In the **EC2 Dashboard**, verify if the instance is running or stopped.
91. **Review CloudWatch Metrics**:
    - Check **CPU utilization**, **disk I/O**, and **network metrics** to see if the instance is overloaded.
92. **Check Security Group Rules**:
    - Ensure that the correct **inbound rules** (e.g., for SSH or HTTP) are in place.
93. **Use EC2 Serial Console (Linux)**:
    - If SSH access fails, use the **EC2 Serial Console** or **EC2 Instance Connect** to log in.
94. **Review System Logs**:
    - Check the **/var/log/messages** or **/var/log/syslog** to see if there are any critical errors.
95. **Reboot the Instance**:
    - If no issues are found, try **rebooting** the instance from the EC2 console.

Q2. How do you add an inbound rule to a custom security group to allow HTTP traffic on port 80?

Steps:

96. **Go to Security Groups**:
    - In the **EC2 Dashboard**, navigate to **Security Groups**.
97. **Edit Inbound Rules**:
    - Select the security group associated with your EC2 instance and click **Edit Inbound Rules**.
    - Add a rule:
        - **Type**: HTTP.
        - **Port Range**: 80.

    - **Source**: `0.0.0.0/0` (to allow traffic from anywhere).

98. **Save and Apply**:
    - Save the rule, and HTTP traffic on port 80 will be allowed.

Practical no 10

Q1 How do you add tags to an existing EC2 instance?

Steps:

99. **Log in to AWS Management Console**:
    - Navigate to AWS Console.
    - Go to the **EC2** Dashboard by searching for EC2 in the search bar.
100. **Select the Instance**:
    - In the **Instances** section, find the EC2 instance you want to add tags to.
    - Select the checkbox next to the instance name.
101. **Manage Tags**:
    - Click on the **Actions** button in the top-right menu.
    - Select **Instance Settings** > **Add/Edit Tags**.
102. **Add New Tags**:
    - Click **Add Tag**.
    - In the **Key** field, enter a descriptive label (e.g., `Environment`).
    - In the **Value** field, enter the corresponding value (e.g., `Production`).
103. **Save Tags**:
    - Once you've added all the necessary tags, click **Save** to apply the changes.

Why Add Tags?

- Tags help you organize and manage resources by assigning metadata (such as project, department, or environment) to instances for easier identification and tracking.

Q2 How do you configure AWS IAM policies for data access control?

Steps to Create a Data Access Control Policy:

104. **Log in to AWS Console**:
    - Navigate to the **IAM Dashboard**.
105. **Create a New Policy**:
    - In the left-hand menu, click **Policies** and then click **Create Policy**.
106. **Select Service**:
    - In the **Visual Editor**, select **S3** as the service.

107. **Define Permissions**:
    - Choose the specific S3 actions you want to allow. For example, select **GetObject** for read access, or **PutObject** for write access.
    - In the **Resources** section, specify the S3 buckets or objects to which this policy will apply (e.g., a specific bucket or all buckets).
108. **Review and Create Policy**:
    - After defining the actions and resources, click **Next: Tags** (optional).
    - Review the policy, give it a name (e.g., `S3DataAccessPolicy`), and click **Create Policy**.
109. **Attach the Policy to Users/Groups**:
    - Go to **Users** or **Groups** in the IAM dashboard.
    - Select the user or group that requires the policy.
    - Click **Add Permissions** and search for the newly created policy to attach it.

Importance:

- IAM policies help control which users or services can access specific data, ensuring that sensitive data is only accessible to authorized users.

Practical no. 11

Q1. How do you add an SSH public key to an existing EC2 instance using the AWS Management Console? [15M]

Steps:

110. **Log in to AWS Console**:
    - Go to the **EC2** dashboard.
111. **Use EC2 Instance Connect**:
    - Select the EC2 instance where you want to add the public SSH key.
    - Click **Connect** and choose **EC2 Instance Connect (browser-based SSH)**.
112. **Add the Public Key**:
    - Once connected via SSH, navigate to the `.ssh/authorized_keys` file:

```
bash
Copy code
sudo nano ~/.ssh/authorized_keys
```

    - Paste your public SSH key into the file.
113. **Save and Exit**:
    - Save the file and exit the text editor (`Ctrl + O` to save, `Ctrl + X` to exit).
114. **Verify Key Addition**:
    - Log out and test the connection using your new SSH public key.

Q2 How does an IAM user goes towards enabling Multi-Factor Authentication (MFA)?

Steps:

115. **Log in to AWS Console**:
    - Go to the **IAM** Dashboard.
116. **Select the User**:
    - Under **Users**, choose the IAM user for whom you want to enable MFA.
117. **Enable MFA**:
    - In the **Security Credentials** tab, under the **Multi-factor authentication (MFA)** section, click **Manage**.
    - Select the type of MFA device (e.g., **Virtual MFA Device**, **U2F Security Key**).
118. **Configure Virtual MFA Device**:

- If using a virtual MFA device, scan the QR code using an MFA app (e.g., Google Authenticator).
- Enter the two consecutive MFA codes displayed in the app to complete the setup.

119. **Save and Verify**:
- Save the changes, and now the user will be prompted for an MFA code when logging in.

Why Enable MFA?

- MFA adds an extra layer of security by requiring both a password and a one-time code from an authentication app, reducing the risk of unauthorized access.

Practical no. 12

Q1 How can you create an IAM policy that allows only read access to S3 buckets? [15M]

Steps:

120. Go to **IAM** > **Policies** > **Create Policy**.
121. Select **S3** service and choose **GetObject**, **ListBucket** actions for read access.
122. Define the resources (buckets or objects) the policy should apply to.
123. Attach the policy to a user or group.

Q2 How would you create a security group to ensure your EC2 instance is accessible only through necessary ports (e.g., HTTP and SSH)? What inbound and outbound rules would you configure, and why?

Steps:

124. Navigate to **Security Groups** in the EC2 dashboard.
125. **Create a New Security Group**:
    - Add **Inbound Rules**:
        - **HTTP**: Port 80, source as `0.0.0.0/0` or your IP.
        - **SSH**: Port 22, restrict access by specifying your IP address.
    - **Outbound Rules**: Allow all outbound traffic or restrict as needed.
126. **Assign Security Group to Instance**:
    - Go to **Instances** > Select your instance > **Actions** > **Networking** > **Change Security Groups**.

Practical no. 13

Q1 Your EC2 instance needs to access other AWS services, such as S3 buckets, securely. How would you set up an IAM role for your EC2 instance to grant it access to specific AWS services? Explain how you would attach the role to your instance and configure permissions. [15M]

Steps:

127. **Create an IAM Role**:
    - Go to **IAM** > **Roles** > **Create Role**.
    - Choose **EC2** as the trusted entity.
    - Attach the necessary permissions (e.g., S3 read access).
128. **Assign Role to EC2 Instance**:
    - Go to **Instances** > Select your instance > **Actions** > **Instance Settings** > **Attach/Replace IAM Role**.
    - Select the created role and attach it.

Q2. How can you access the system logs of an EC2 instance from the AWS Management Console?

Steps:

129. Navigate to **EC2 Dashboard** > **Instances**.
130. Select the instance, click **Actions** > **Monitor and troubleshoot** > **Get system log**.
131. View and troubleshoot issues using the log output.

Practical no. 14

Q1 Launch a new EC2 instance to run a basic web server application. Describe the steps you would take to launch an EC2 instance. Include details about choosing an AMI, selecting an instance type, configuring instance details, and assigning a security group.

Steps:

132. **Log into the AWS Management Console**:
- Navigate to the **EC2 Dashboard**.

133. **Launch Instance Wizard**:
- Click on **Launch Instance**.

134. **Choose an Amazon Machine Image (AMI)**:
- Select an **Amazon Linux 2** AMI or **Ubuntu** as it's ideal for running web servers.
- Ensure the AMI is free-tier eligible if applicable.

135. **Choose an Instance Type**:
- Select **t2.micro** (free-tier eligible) or another instance type depending on your application needs.
- The **t2.micro** instance is suitable for a basic web server.

136. **Configure Instance Details**:
- Set the number of instances (typically 1 for a single web server).
- Leave default **network settings** (VPC and subnet).
- Enable **Auto-assign Public IP** to allow public access.

137. **Add Storage**:
- Use the default **8 GB EBS volume** or modify it based on storage needs.
- For web servers, the default size is usually sufficient.

138. **Add Tags**:
- Add a **tag** to name the instance (e.g., `Key: Name`, `Value: WebServer`).

139. **Configure Security Group**:
- Create a new security group or choose an existing one.
- Ensure you allow the necessary traffic:
  - **HTTP (port 80)** for web traffic.
  - **SSH (port 22)** to access the instance via SSH.

140. **Review and Launch**:
- Review the configuration and click **Launch**.
- When prompted, create or choose an existing **key pair** for SSH access.

141. **Connect to the Instance**:
    - Once the instance is running, connect via SSH using the key pair:

```
bash
Copy code
ssh -i "your-key.pem" ec2-user@your-instance-public-ip
```

142. **Install Web Server (Apache or Nginx)**:
    - For **Apache**:

```
bash
Copy code
sudo yum update -y
sudo yum install httpd -y
sudo systemctl start httpd
sudo systemctl enable httpd
```

    - For **Nginx**:

```
bash
Copy code
sudo yum install nginx -y
sudo systemctl start nginx
sudo systemctl enable nginx
```

143. **Test the Web Server**:
    - Open a web browser and enter your instance's **public IP**. You should see the default web server page.

Q2. How can you launch an EC2 instance with a specific IAM role?

Steps:

144. **Create an IAM Role**:
    - In the **IAM Dashboard**, navigate to **Roles** and click **Create Role**.
    - Choose **EC2** as the trusted entity.
    - Attach a policy that defines the permissions required by the instance (e.g., **S3 Read Access**).
    - Complete the role creation and name it (e.g., `WebServerRole`).
145. **Launch a New EC2 Instance**:

- In the **EC2 Dashboard**, click **Launch Instance**.

146. **Choose an AMI and Instance Type**:
    - Select the AMI (e.g., **Amazon Linux 2**).
    - Choose the instance type (e.g., **t2.micro**).
147. **Configure Instance Details**:
    - Under **IAM Role**, choose the IAM role you created earlier (e.g., `WebServerRole`).
    - This will allow the instance to assume the role and access the specified AWS services.
148. **Complete the Launch Process**:
    - Add storage, configure the security group (allow HTTP/SSH), and review the settings.
    - Click **Launch** and assign a key pair for SSH access.
149. **Instance with Role**:
    - After the instance is launched, it will have the permissions assigned by the IAM role (e.g., access to an S3 bucket or other AWS services).

Practical no. 15

Q1. You want to protect your EC2 instances from common security vulnerabilities and attacks. What steps would you take to secure EC2 instances against common vulnerabilities such as unauthorized access, misconfigurations, and software vulnerabilities? Discuss strategies such as patch management, configuration hardening, and access control.

Steps to Secure EC2 Instances:

150. **Patch Management**:
    - Regularly apply OS and software updates.
    - Automate patching using **AWS Systems Manager Patch Manager**.
151. **Configuration Hardening**:
    - Disable unused ports and services.
    - Use security-enhanced configurations (e.g., **CIS benchmarks**).
    - Implement **least privilege access** via IAM roles and policies.
152. **Access Control**:
    - Enable **Multi-Factor Authentication (MFA)** for all IAM users.
    - Use **IAM roles** for access to AWS services instead of hardcoding credentials.
153. **Network Security**:
    - Create strict **Security Group** and **Network ACL** rules.
    - Use **AWS Shield** or **WAF** to protect against DDoS attacks.

Q2. How can you automate the process of adding or removing SSH public keys to EC2 instances using AWS Systems Manager?

Steps:

154. **Setup Systems Manager Agent**:
    - Ensure the **SSM Agent** is installed and running on the EC2 instance.
155. **Run Command for Automation**:

- In **AWS Systems Manager**, go to **Run Command**.
- Choose the **AWS-RunShellScript** document.
- In the script, add a command to modify the `authorized_keys` file:

```
bash
Copy code
echo "your-public-key" >> ~/.ssh/authorized_keys
```

- Run the command to add/remove keys across multiple instances.

Practical no. 16

Q1. Your EC2 instance needs to send outbound traffic to an external API but should not allow any other outbound traffic. How would you configure the outbound rules of your security group to permit only the necessary traffic? What considerations would you take into account for setting these rules? [15M]

Steps:

156. **Go to Security Groups**:
    - Navigate to **EC2** > **Security Groups**.
157. **Modify Outbound Rules**:
    - Select your security group and click on **Outbound Rules**.
    - **Remove** the default rule that allows all outbound traffic.
    - **Add a new rule** that permits outbound traffic only to the external API (e.g., by specifying the API's IP address and port).
158. **Save and Apply**:
    - Ensure that only the specified port for the external API is open (e.g., HTTP or HTTPS).
    - Restrict other outbound connections to prevent unwanted traffic.

Q2 How does AWS handle connections when an EC2 instance is stopped and started again?

Explanation:

- When an EC2 instance is **stopped**, its **public IP address** is released, and any network connections are terminated.
- Upon **restarting**, the instance is assigned a new public IP unless it's using an **Elastic IP**, which remains fixed.
- **Security Groups** and instance data stored in **EBS volumes** remain intact.

Practical no. 17

Q1 How do you restrict HTTP traffic to only a specific IP address or range in a Security Group?

Steps:

159. **Go to EC2 Dashboard**:
    - Navigate to **EC2** > **Security Groups**.
160. **Modify Inbound Rules**:
    - Select your security group and edit the **Inbound Rules**.
    - Add a rule allowing **HTTP (port 80)** but specify the **source IP** or **IP range** (e.g., `203.0.113.0/24`).
    - Remove any existing rule allowing traffic from `0.0.0.0/0` (anywhere).
161. **Save and Apply**:
    - Save the changes to restrict HTTP access only to the specified IP or range.

Q2. How do you attach a policy to an IAM user using the AWS Management Console?

Steps:

162. **Log in to AWS Console**:
    - Go to the **IAM** Dashboard.
163. **Select the User**:
    - Under **Users**, choose the user you want to modify.
164. **Attach Policy**:
    - In the **Permissions** tab, click **Add permissions**.
    - Choose to attach an existing policy or create a new one.
    - Select the appropriate policy (e.g., **S3 Full Access**) and attach it to the user.

Practical no. 18

Q1. How do you add an outbound rule to a Network ACL to block all traffic to the internet?

Steps:

165. **Go to VPC Dashboard**:
   - Navigate to **VPC** > **Network ACLs**.
166. **Select Your Network ACL**:
   - Choose the **Network ACL** associated with your subnet.
167. **Modify Outbound Rules**:
   - Add a rule with a **rule number** higher than any existing allow rules.
   - **Deny** all outbound traffic (`0.0.0.0/0` for both IPv4 and IPv6) by setting the **Action** to Deny.
168. **Save and Apply**:
   - This ensures that no outbound traffic is allowed to the internet.

Q2. How do you create an IAM policy that denies all access to a specific S3 bucket?

Steps:

169. **Go to IAM Policies**:
   - In the **IAM** Dashboard, go to **Policies** and click **Create Policy**.
170. **Define S3 Service Actions**:
   - In the policy editor, select **S3**.
   - Choose **Deny** for all actions (`s3:*`) on a specific bucket by specifying the bucket ARN (e.g., `arn:aws:s3:::your-bucket-name`).
171. **Attach Policy**:
   - Attach this policy to any users or groups that should be denied access to the bucket.

Practical no. 19

Q1. How do you implement a Network ACL that allows only HTTPS traffic on port 443 and denies all other traffic?

Steps:

172. **Navigate to VPC**:
    - Go to the **VPC Dashboard** and select **Network ACLs**.
173. **Modify Inbound/Outbound Rules**:
    - **Allow HTTPS**: Add a rule allowing traffic on **port 443** (HTTPS) for both inbound and outbound, specifying **source** as `0.0.0.0/0`.
    - **Deny All Other Traffic**: Add lower priority rules that **deny all other traffic** (`0.0.0.0/0` for both inbound and outbound).
174. **Save Changes**:
    - Ensure that the deny rules have a higher priority than the allow rule for HTTPS.

Q2. How do you grant read access to a specific S3 bucket for an IAM user?

Steps:

175. **Go to IAM**:
    - Navigate to the **IAM** Dashboard.
176. **Create/Attach Policy**:
    - Create a policy or attach an existing one that allows **s3** and **s3** for a specific S3 bucket.
    - Specify the bucket ARN in the resource section of the policy.
177. **Attach Policy to User**:
    - Attach the policy to the desired IAM user or group.

Practical no. 20

Q1. How do you allow only traffic from other EC2 instances in the same security group? [15M]

Steps:

178. **Go to EC2 Security Groups**:
    - Navigate to **EC2** > **Security Groups**.
179. **Modify Inbound Rules**:
    - Select your security group and edit the **Inbound Rules**.
    - Add a rule for the necessary ports (e.g., HTTP or SSH).
    - For the **Source**, select **Custom** and choose the **same security group ID**.
180. **Save and Apply**:
    - This allows only traffic from other instances using the same security group.

Q2. How can you set up an S3 bucket to be publicly accessible?

Steps:

181. **Go to S3**:
    - Navigate to **S3** and select the bucket you want to make public.
182. **Enable Public Access**:
    - In the **Permissions** tab, disable the **Block Public Access** settings.
183. **Bucket Policy**:
    - Create or edit a **bucket policy** to allow public access by specifying `"Effect": "Allow"`, `"Principal": "*"` for `s3:GetObject`.
184. **Save**:
    - Apply the policy and confirm the public accessibility of the bucket.