Assignment 3

Q1)Deploy Linux and Windows virtual machine and access them using SSH and RDP

My Youtube videos Link :-
https://youtu.be/XuV_G6Yy_2c?si=doyOD2gOGWVqZcfw,
https://youtu.be/1yysjxPYEK8?si=M0rXHPiK1bDiwzQx

Deploy a Linux VM on Azure

1. **Create a Resource Group:**
   - Go to the Azure portal.
   - Select "Resource groups" from the left menu and click "Add".
   - Fill in the necessary details and create the resource group.
2. **Create a Linux VM:**
   - Go to "Virtual machines" and click "Add".
   - Choose your subscription and resource group.
   - Fill in the VM details:
     - **Image:** Select a Linux distribution (e.g., Ubuntu Server).
     - **Size:** Choose an appropriate size based on your needs.
     - **Authentication:** Choose SSH public key for authentication.
     - **Inbound ports:** Allow SSH (port 22).
   - **Generate SSH Keys:**
     - On your local machine, generate SSH keys if you haven't already:

       ```
       bash
       Copy code
       ssh-keygen -t rsa -b 2048
       ```

     - Copy the public key content and paste it into the SSH public key field in the Azure portal.
3. **Review and Create:**
   - Review the settings and click "Create".

Access the Linux VM:

- Open a terminal on your local machine and use the SSH command:

  ```
  bash
  Copy code
  ssh username@public-ip-address
  ```

Replace `username` with the username you specified during VM creation and `public-ip-address` with the public IP address of your VM (found in the Azure portal under your VM's overview).

Deploy a Windows VM on Azure

1. **Create a Resource Group:**
   - Go to the Azure portal.
   - Select "Resource groups" from the left menu and click "Add".
   - Fill in the necessary details and create the resource group.
2. **Create a Windows VM:**
   - Go to "Virtual machines" and click "Add".
   - Choose your subscription and resource group.
   - Fill in the VM details:
     - **Image:** Select a Windows Server version (e.g., Windows Server 2019 Datacenter).
     - **Size:** Choose an appropriate size based on your needs.
     - **Administrator account:** Set a username and password for RDP access.
     - **Inbound ports:** Allow RDP (port 3389).
3. **Review and Create:**
   - Review the settings and click "Create".

Access the Windows VM:

1. **Retrieve the Public IP Address:**
   - Find the public IP address of your Windows VM in the Azure portal under your VM's overview.
2. **Use Remote Desktop Connection (RDP):**
   - On your local machine, open the Remote Desktop Connection application.
   - Enter the public IP address of your Windows VM.
   - Click "Connect" and enter the username and password you specified during VM creation.

With these steps, you should be able to deploy and access both Linux and Windows VMs on Azure using SSH and RDP, respectively.

Q2)Create an App Service Plan

To create an App Service Plan in Azure, follow these steps:

1. **Navigate to the Azure Portal:**
   - Open a web browser and go to the Azure portal.
2. **Create a Resource Group (if needed):**
   - In the Azure portal, select "Resource groups" from the left-hand menu.
   - Click "Add".

- Enter the necessary details:
    - **Subscription:** Select your Azure subscription.
    - **Resource group:** Enter a unique name for the resource group.
    - **Region:** Choose the region where you want to deploy your resources.
- Click "Review + Create" and then "Create".

3. **Create an App Service Plan:**
    - In the Azure portal, select "Create a resource" from the left-hand menu.
    - In the "Search the Marketplace" box, type "App Service Plan" and select it from the dropdown list.
    - Click "Create".
4. **Configure the App Service Plan:**
    - **Basics:**
        - **Subscription:** Select your Azure subscription.
        - **Resource group:** Select the resource group you created or an existing one.
        - **Name:** Enter a name for your App Service Plan.
        - **Operating System:** Choose either "Windows" or "Linux".
        - **Region:** Select the region where you want to deploy your App Service Plan.
    - **Sku and Size:**
        - Click on "Change size" to select the pricing tier that fits your needs. You can choose between different tiers (e.g., Free, Shared, Basic, Standard, Premium).
        - Click "Apply" after selecting the size.
5. **Review and Create:**
    - Review your configuration settings.
    - Click "Review + create".
    - After validation passes, click "Create" to deploy the App Service Plan.

Once the App Service Plan is created, you can deploy web apps, mobile app backends, or API apps into this plan. This service plan defines the region, capacity, and pricing tier for your web apps.

Q3) Provision a Web App in the existing App Service Plan and deploy a simple welcome page on it .

To provision a Web App in an existing App Service Plan and deploy a simple welcome page on it, follow these steps:

Provision a Web App

1. **Navigate to the Azure Portal:**
    - Open a web browser and go to the Azure portal.
2. **Create a Web App:**
    - In the Azure portal, select "Create a resource" from the left-hand menu.

   - In the "Search the Marketplace" box, type "Web App" and select it from the dropdown list.
   - Click "Create".
3. **Configure the Web App:**
   - **Basics:**
     - **Subscription:** Select your Azure subscription.
     - **Resource Group:** Select the resource group that contains your App Service Plan.
     - **Name:** Enter a unique name for your Web App. This will be the URL of your Web App (e.g., `yourappname.azurewebsites.net`).
     - **Publish:** Select "Code".
     - **Runtime Stack:** Choose the runtime stack you want to use (e.g., .NET, Node.js, PHP, Python).
     - **Operating System:** Select the operating system that matches your App Service Plan (Windows or Linux).
     - **Region:** Ensure it matches the region of your App Service Plan.
   - **App Service Plan:**
     - Click "App Service Plan/Location".
     - Select "Existing App Service Plan".
     - Choose the App Service Plan you created previously.
4. **Review and Create:**
   - Review your configuration settings.
   - Click "Review + create".
   - After validation passes, click "Create" to provision the Web App.

Deploy a Simple Welcome Page

1. **Prepare the Welcome Page:**
   - Create a simple HTML file named `index.html` with the following content:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
    <title>Welcome</title>
</head>
<body>
    <h1>Welcome to Azure Web App!</h1>
    <p>This is a simple welcome page deployed on an Azure Web
App.</p>
</body>
</html>
```

2. **Deploy the Welcome Page:**
   - **Using FTP or FTPS:**
     - In the Azure portal, go to your Web App.
     - Under the "Deployment" section, select "Deployment Center".
     - Choose the "FTP" option and note the FTP/FTPS credentials provided.
     - Use an FTP client (e.g., FileZilla) to connect to your Web App using the FTP/FTPS credentials.

        - Upload the `index.html` file to the `/site/wwwroot` directory.
    - **Using the Azure CLI:**
        - Install the [Azure CLI]() if you haven't already.
        - Open a terminal and log in to Azure:

          ```
          bash
          Copy code
          az login
          ```

        - Navigate to the directory containing your `index.html` file.
        - Deploy the file using the Azure CLI:

          ```
          bash
          Copy code
          az webapp up --name yourappname --resource-group
          yourresourcegroup --plan yourserviceplan
          ```

3. **Verify Deployment:**
    - Open a web browser and navigate to `https://yourappname.azurewebsites.net`.
    - You should see the simple welcome page you deployed.

These steps will help you provision a Web App in an existing App Service Plan and deploy a simple welcome page on it.

Q4) Create ACR and pull image from ACR and Create a container from it

To create an Azure Container Registry (ACR), push a Docker image to it, and then create a container from that image in an Azure Container Instance (ACI), follow these steps:

Step 1: Create an Azure Container Registry (ACR)

1. **Navigate to the Azure Portal:**
    - Open a web browser and go to the [Azure portal]().
2. **Create ACR:**
    - Select "Create a resource" from the left-hand menu.
    - Search for "Container Registry" and select it.
    - Click "Create".
3. **Configure the Container Registry:**
    - **Basics:**
        - **Subscription:** Select your Azure subscription.
        - **Resource Group:** Select an existing resource group or create a new one.
        - **Registry Name:** Enter a unique name for your container registry.
        - **Location:** Choose a region.
        - **SKU:** Choose a SKU (e.g., Basic, Standard, Premium).

4. **Review and Create:**
    - Review your configuration settings.
    - Click "Review + create".
    - After validation passes, click "Create".

Step 2: Push a Docker Image to ACR

1. **Login to ACR:**
    - Open a terminal on your local machine.
    - Log in to the Azure CLI:

      ```
      bash
      Copy code
      az login
      ```

    - Log in to your container registry:

      ```
      bash
      Copy code
      az acr login --name yourRegistryName
      ```

2. **Build and Tag your Docker Image:**
    - Navigate to the directory containing your Dockerfile.
    - Build the Docker image:

      ```
      bash
      Copy code
      docker build -t yourRegistryName.azurecr.io/yourImageName:tag .
      ```

3. **Push the Docker Image to ACR:**
    - Push the image to your container registry:

      ```
      bash
      Copy code
      docker push yourRegistryName.azurecr.io/yourImageName:tag
      ```

Step 3: Create a Container Instance from the Image in ACR

1. **Create a Container Instance:**
    - In the Azure portal, select "Create a resource".
    - Search for "Container Instances" and select it.
    - Click "Create".
2. **Configure the Container Instance:**
    - **Basics:**
        - **Subscription:** Select your Azure subscription.
        - **Resource Group:** Select the resource group where your ACR is located.
        - **Container name:** Enter a name for your container.
        - **Region:** Choose the same region as your ACR.
    - **Image Source:**
        - **Image Source:** Select "Azure Container Registry".
        - **Registry:** Select your ACR.

        - **Image:** Enter `yourRegistryName.azurecr.io/yourImageName:tag`.
    - **Instance Details:**
        - Configure the instance details as needed (e.g., size, DNS name label).
3. **Registry Authentication:**
    - Under the "Registry" section, enable "Admin user" in the ACR if you haven't already:
        - Go to your ACR in the Azure portal.
        - Under "Settings", select "Access keys".
        - Enable "Admin user".
        - Use the provided username and password for the registry authentication in your container instance setup.
4. **Review and Create:**
    - Review your configuration settings.
    - Click "Review + create".
    - After validation passes, click "Create".

Step 4: Verify the Container Instance

1. **Check the Status:**
    - In the Azure portal, go to "Container instances".
    - Select your container instance to view its status and logs.
2. **Access the Container:**
    - If you configured a DNS name label, you can access your container via the URL `http://yourdnsname.region.azurecontainer.io`.

By following these steps, you will have created an Azure Container Registry, pushed a Docker image to it, and created a container instance from that image.

Q5) Create Container Instance and deploy a simple docker application on it.

To create an Azure Container Instance (ACI) and deploy a simple Docker application on it, follow these steps:

Step 1: Create an Azure Container Registry (ACR) (if you don't have one)

1. **Navigate to the Azure Portal:**
    - Open a web browser and go to the Azure portal.
2. **Create ACR:**
    - Select "Create a resource" from the left-hand menu.
    - Search for "Container Registry" and select it.
    - Click "Create".
3. **Configure the Container Registry:**
    - **Basics:**

        - **Subscription:** Select your Azure subscription.
        - **Resource Group:** Select an existing resource group or create a new one.
        - **Registry Name:** Enter a unique name for your container registry.
        - **Location:** Choose a region.
        - **SKU:** Choose a SKU (e.g., Basic, Standard, Premium).

4. **Review and Create:**
    - Review your configuration settings.
    - Click "Review + create".
    - After validation passes, click "Create".

Step 2: Push a Simple Docker Application to ACR

1. **Prepare a Simple Docker Application:**
    - Create a simple Dockerfile for a web application (e.g., using Nginx):

```
Dockerfile
Copy code
FROM nginx:alpine
COPY ./index.html /usr/share/nginx/html/index.html
```

    - Create an `index.html` file in the same directory with the following content:

```
html
Copy code
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
    <title>Welcome</title>
</head>
<body>
    <h1>Welcome to Azure Container Instance!</h1>
    <p>This is a simple web application running in a Docker
container.</p>
</body>
</html>
```

2. **Login to ACR:**
    - Open a terminal on your local machine.
    - Log in to the Azure CLI:

```
bash
Copy code
az login
```

    - Log in to your container registry:

```
bash
Copy code
az acr login --name yourRegistryName
```

3. **Build and Tag your Docker Image:**
   - Navigate to the directory containing your Dockerfile and `index.html`.
   - Build the Docker image:

     ```
     bash
     Copy code
     docker build -t yourRegistryName.azurecr.io/yourImageName:tag .
     ```

4. **Push the Docker Image to ACR:**
   - Push the image to your container registry:

     ```
     bash
     Copy code
     docker push yourRegistryName.azurecr.io/yourImageName:tag
     ```

Step 3: Create an Azure Container Instance

1. **Create a Container Instance:**
   - In the Azure portal, select "Create a resource".
   - Search for "Container Instances" and select it.
   - Click "Create".
2. **Configure the Container Instance:**
   - **Basics:**
     - **Subscription:** Select your Azure subscription.
     - **Resource Group:** Select the resource group where your ACR is located.
     - **Container name:** Enter a name for your container.
     - **Region:** Choose the same region as your ACR.
   - **Image Source:**
     - **Image Source:** Select "Azure Container Registry".
     - **Registry:** Select your ACR.
     - **Image:** Enter `yourRegistryName.azurecr.io/yourImageName:tag`.
   - **Instance Details:**
     - Configure the instance details as needed (e.g., size, DNS name label).
3. **Registry Authentication:**
   - Under the "Registry" section, enable "Admin user" in the ACR if you haven't already:
     - Go to your ACR in the Azure portal.
     - Under "Settings", select "Access keys".
     - Enable "Admin user".
     - Use the provided username and password for the registry authentication in your container instance setup.
4. **Review and Create:**
   - Review your configuration settings.
   - Click "Review + create".
   - After validation passes, click "Create".

Step 4: Verify the Container Instance

1. **Check the Status:**
   - In the Azure portal, go to "Container instances".
   - Select your container instance to view its status and logs.

2. **Access the Container:**
    - If you configured a DNS name label, you can access your container via the URL `http://yourdnsname.region.azurecontainer.io`.
    - If not, you can find the IP address of the container instance in its overview and access it via `http://public-ip-address`.

By following these steps, you will have created an Azure Container Registry, pushed a Docker image of a simple web application to it, and created a container instance from that image.

Q6) Create Container Groups and test functionality

To create a container group in Azure and test its functionality, you can follow these steps. A container group can include multiple containers that share the same network and storage resources. For this example, we'll create a container group with a single container running a simple web application.

Step 1: Create a Container Registry (if not already created)

If you don't have an Azure Container Registry (ACR), follow the steps below. Otherwise, skip to Step 2.

1. **Navigate to the Azure Portal:**
    - Open the Azure portal.
2. **Create ACR:**
    - Select "Create a resource" from the left-hand menu.
    - Search for "Container Registry" and select it.
    - Click "Create".
3. **Configure the Container Registry:**
    - **Basics:**
        - **Subscription:** Select your Azure subscription.
        - **Resource Group:** Select an existing resource group or create a new one.
        - **Registry Name:** Enter a unique name for your container registry.
        - **Location:** Choose a region.
        - **SKU:** Choose a SKU (e.g., Basic, Standard, Premium).
4. **Review and Create:**
    - Review your configuration settings.
    - Click "Review + create".
    - After validation passes, click "Create".

Step 2: Push a Simple Docker Application to ACR

1. **Prepare a Simple Docker Application:**
    - Create a simple Dockerfile for a web application (e.g., using Nginx):

```
Dockerfile
Copy code
FROM nginx:alpine
COPY ./index.html /usr/share/nginx/html/index.html
```

- Create an `index.html` file in the same directory with the following content:

```
html
Copy code
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Welcome</title>
</head>
<body>
    <h1>Welcome to Azure Container Instance!</h1>
    <p>This is a simple web application running in a Docker container.</p>
</body>
</html>
```

2. **Login to ACR:**
   - Open a terminal on your local machine.
   - Log in to the Azure CLI:

```
bash
Copy code
az login
```

   - Log in to your container registry:

```
bash
Copy code
az acr login --name yourRegistryName
```

3. **Build and Tag your Docker Image:**
   - Navigate to the directory containing your Dockerfile and `index.html`.
   - Build the Docker image:

```
bash
Copy code
docker build -t yourRegistryName.azurecr.io/yourImageName:tag .
```

4. **Push the Docker Image to ACR:**
   - Push the image to your container registry:

```
bash
Copy code
docker push yourRegistryName.azurecr.io/yourImageName:tag
```

Step 3: Create a Container Group

1. **Create a Container Group:**
   - In the Azure portal, select "Create a resource".
   - Search for "Container Instances" and select it.
   - Click "Create".
2. **Configure the Container Group:**
   - **Basics:**
     - **Subscription:** Select your Azure subscription.
     - **Resource Group:** Select the resource group where your ACR is located.
     - **Container group name:** Enter a name for your container group.
     - **Region:** Choose the same region as your ACR.
   - **Container Image:**
     - Click "Add Container".
     - **Name:** Enter a name for your container.
     - **Image source:** Select "Azure Container Registry".
     - **Registry:** Select your ACR.
     - **Image:** Enter `yourRegistryName.azurecr.io/yourImageName:tag`.
     - **Ports:** Add port 80 (if you're using a web application).
   - **Instance Size:**
     - Configure the instance size as needed (CPU cores and memory).
3. **Registry Authentication:**
   - Under the "Registry" section, enable "Admin user" in the ACR if you haven't already:
     - Go to your ACR in the Azure portal.
     - Under "Settings", select "Access keys".
     - Enable "Admin user".
     - Use the provided username and password for the registry authentication in your container instance setup.
4. **Review and Create:**
   - Review your configuration settings.
   - Click "Review + create".
   - After validation passes, click "Create".

Step 4: Verify the Container Group

1. **Check the Status:**
   - In the Azure portal, go to "Container instances".
   - Select your container group to view its status and logs.
2. **Access the Container:**
   - Find the IP address of the container group in its overview.
   - Open a web browser and navigate to `http://public-ip-address`.

You should see the simple web application displaying the welcome page.

By following these steps, you will have created an Azure Container Registry, pushed a Docker image to it, created a container group with a container running that image, and verified its functionality.