**Problem Statement Overview**

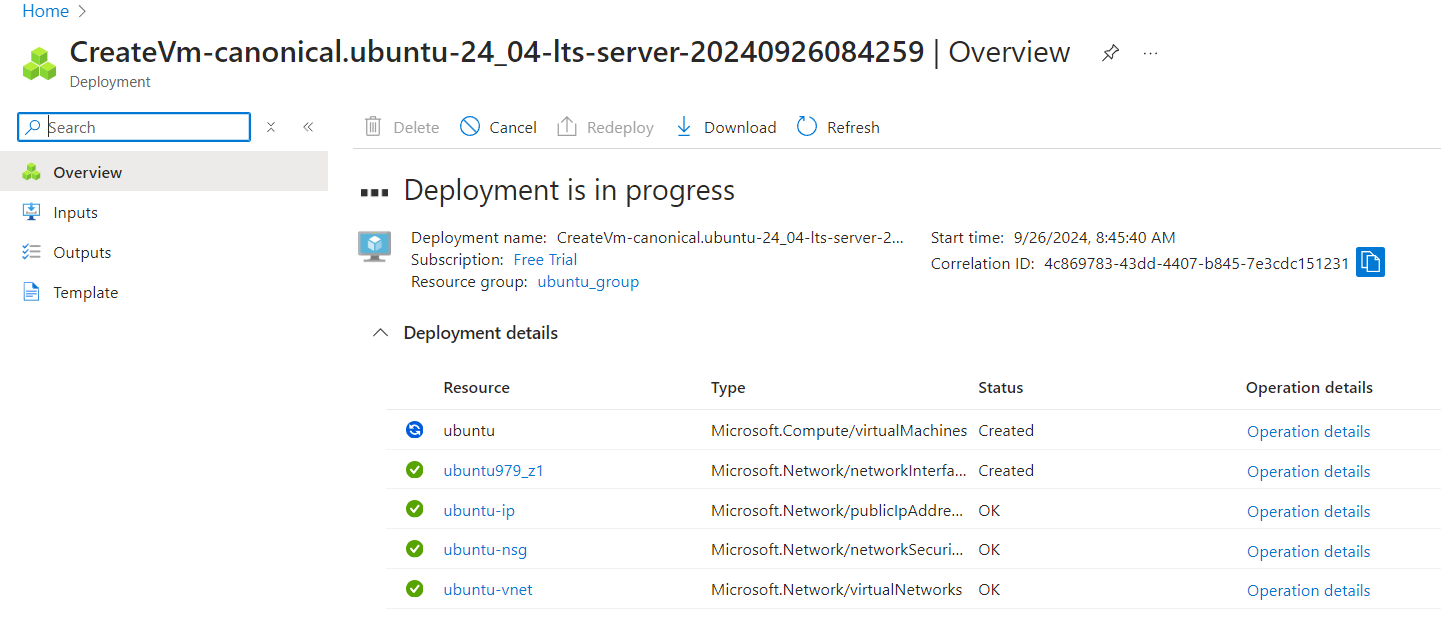
You need to:

* Scale VMs based on CPU utilization.
* Ensure fault tolerance by deploying in a way that minimizes the impact of potential failures.
* Use a custom VM image for consistency across your deployments.
* Deploy all VMs in the same availability zone to enhance reliability.

**Step-by-Step Implementation**

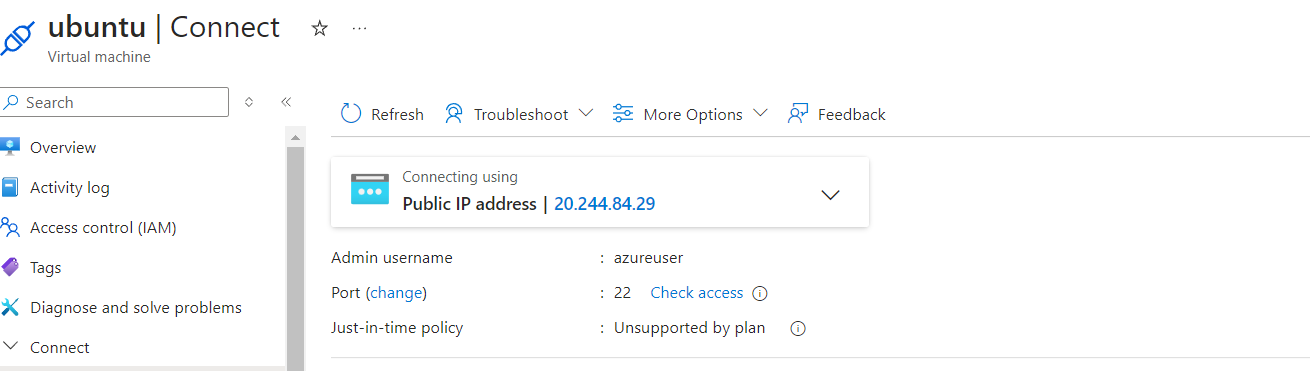
**Step 1: Create a Virtual Machine in Azure**

1. **Log in to the Azure Portal**:
   * Go to the [Azure Portal](https://portal.azure.com) and sign in with your account.
2. **Create a New Virtual Machine**:
   * In the Azure Portal, click on **"Create a resource"** from the left-hand menu.
3. **Configure the VM Basics**:
   * **Subscription**: Choose your subscription.
   * **Resource Group**: You can either create a new resource group or select an existing one.
   * **Virtual Machine Name**: Give your VM a name.
   * **Region**: Select **"West US"** from the dropdown menu.
   * **Availability Options**: Choose according to your needs (No infrastructure redundancy, Availability zone, etc.).
   * **Image**: Select **"Ubuntu"** from the list of available images (choose the specific version you need, e.g., Ubuntu 20.04 LTS).
   * **Size**: Choose the VM size based on your requirements.
4. **Configure Administrator Account**:
   * Choose the **Authentication type** (SSH public key or password). If using SSH, you'll need to generate an SSH key pair if you haven't already.
   * Enter the **Username** and **SSH public key** (if applicable).
5. **Configure Networking**:
   * Under the **Networking** tab, ensure a new virtual network and subnet are created or select an existing one.
   * Make sure to allow **Public IP** to connect to your VM.
6. **Open SSH Port**:
   * In the **Networking** section, add an inbound port rule to allow **SSH (port 22)** and **HTTPS (port 80)**
7. **Review + Create**:
   * Review your settings and click **"Create"** to provision the VM. This may take a few minutes.

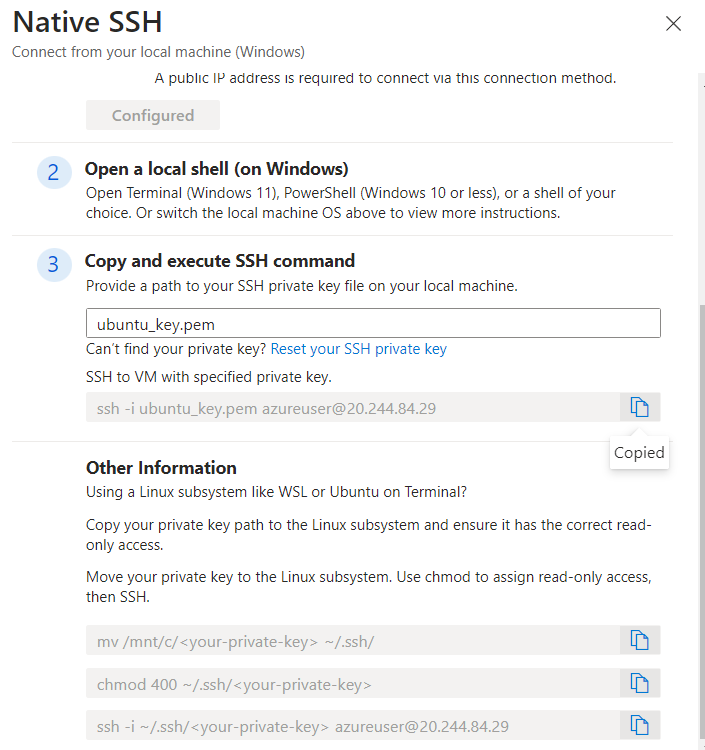


**Step 2: Connect to the Linux VM using Terminal**

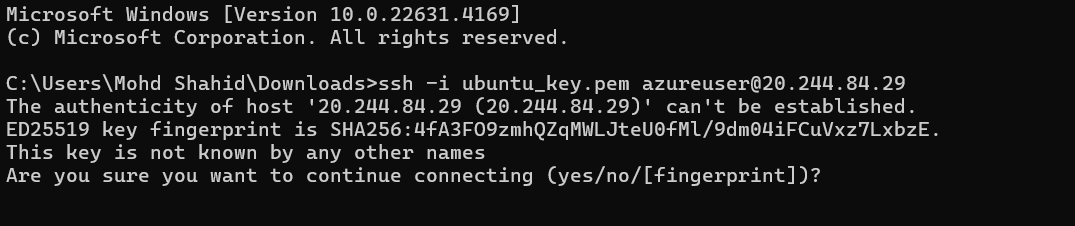
1. **Get the Public IP Address**:
   * Once the VM is created, go to the **"Overview"** page of your VM in the Azure Portal.
   * Note the **Public IP address** of the VM.



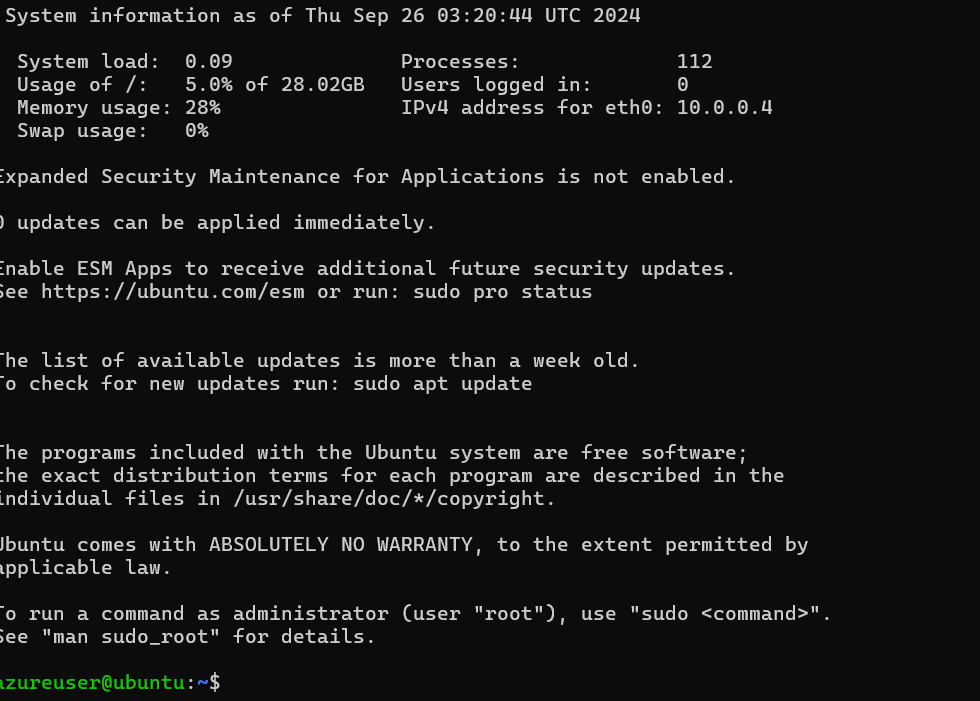
1. **Open Terminal**:
   * On your local machine, open a terminal (Linux, macOS, or Windows with WSL).
2. **Connect to the VM**:
   * Use the following command to connect via SSH:



1. **Accept the SSH Key**:
   * The first time you connect, you'll be asked to confirm the authenticity of the host. Type **"yes"** and hit **Enter**.



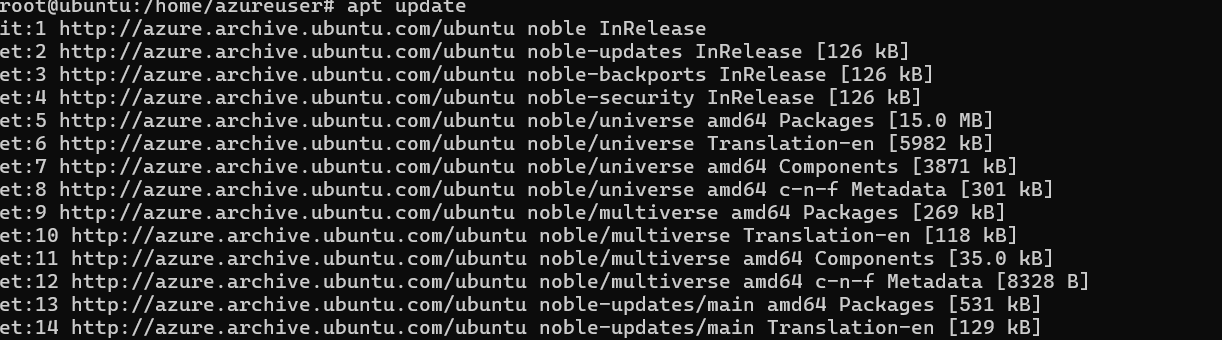
* + If you used a password for authentication, enter it when prompted.

.

**Step 2: Install Apache 2 Software**

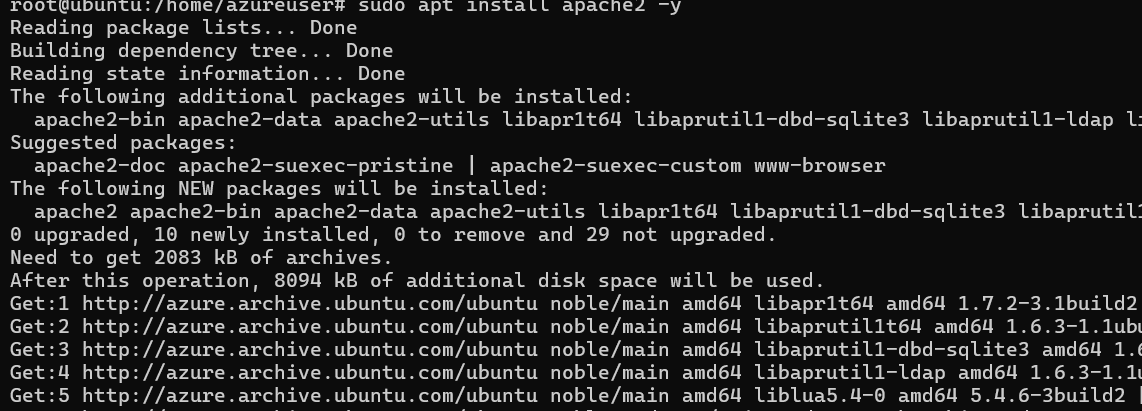
1. **Update the Package Index**:
   * Run the following command to update the package index:

**sudo apt update**

****

1. **Install Apache 2**:
   * Install the Apache 2 software with the following command:

**sudo apt install apache2 -y**

****

1. **Start Apache Service**:
   * Start the Apache service and enable it to start on boot:

**sudo systemctl start apache2**

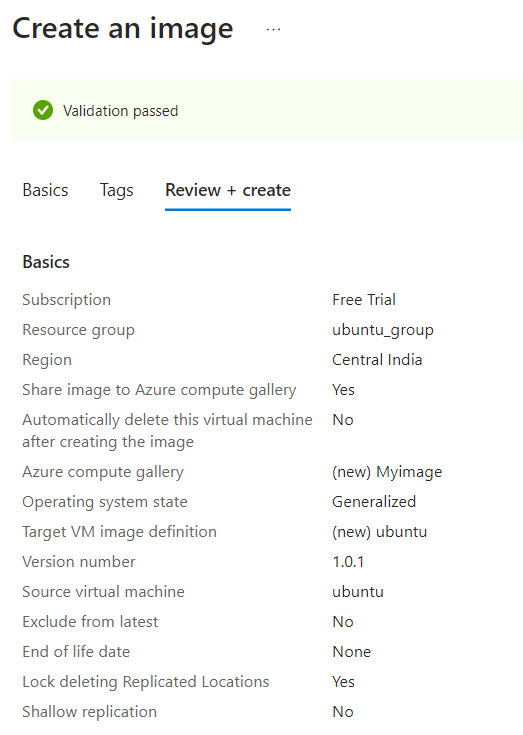
**sudo systemctl enable apache2**

1. **Verify Apache Installation**:



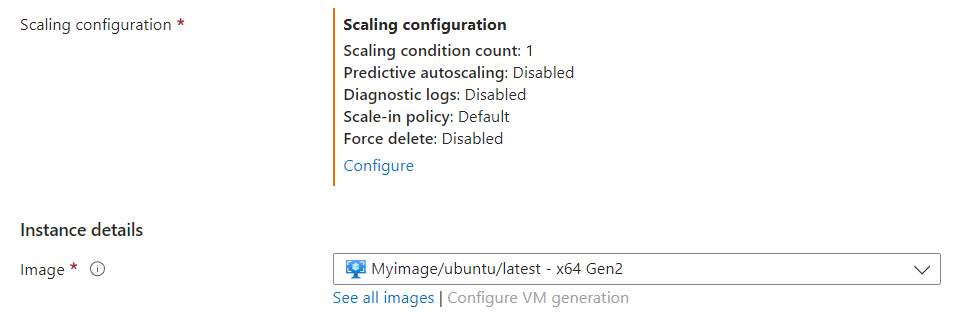
**Step 3: Create an Image Out of the VM**

* + Go to the Azure portal and select your VM from the list.
  + On the page for the VM, on the upper menu, select **Capture**.

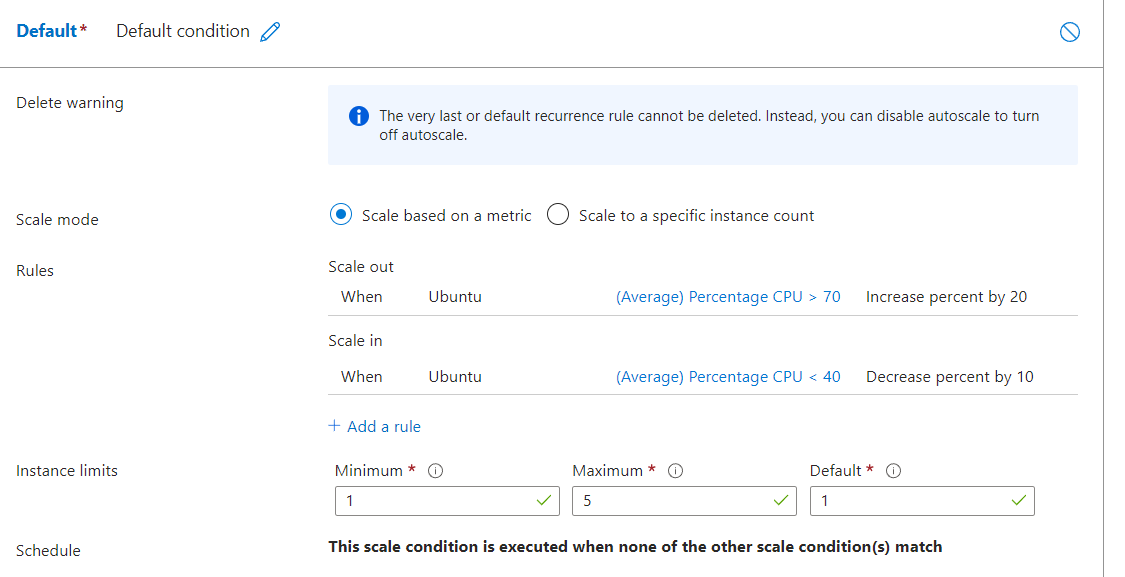


**Step 3: Create a Virtual Machine Scale Set**

1. **Create a Scale Set**:
   * In the Azure Portal, select **Create a resource** > **Compute** > **Virtual Machine Scale Set**.
   * Fill in the basics:
     + **Subscription**: Select your subscription.
     + **Resource Group**: Choose the resource group created earlier.
     + **Name**: Provide a name for your scale set.
     + **Region**: Choose the same region as your resource group.
     + **Image**: Select the custom image you created earlier.
     + **Instance count**: Set the initial count (e.g., 1).



1. **Configure Scaling Settings**:
   * **Autoscale**:
     + Enable autoscaling.
     + Set minimum instances to 1 and maximum to 5.
   * **Scale Out Rule**:
     + Condition: CPU utilization exceeds 70% over a 10-minute period.
     + Action: Increase the instance count by 20%.
   * **Scale In Rule**:
     + Condition: CPU utilization drops below 40% over a 10-minute period.
     + Action: Decrease the instance count by 10%.

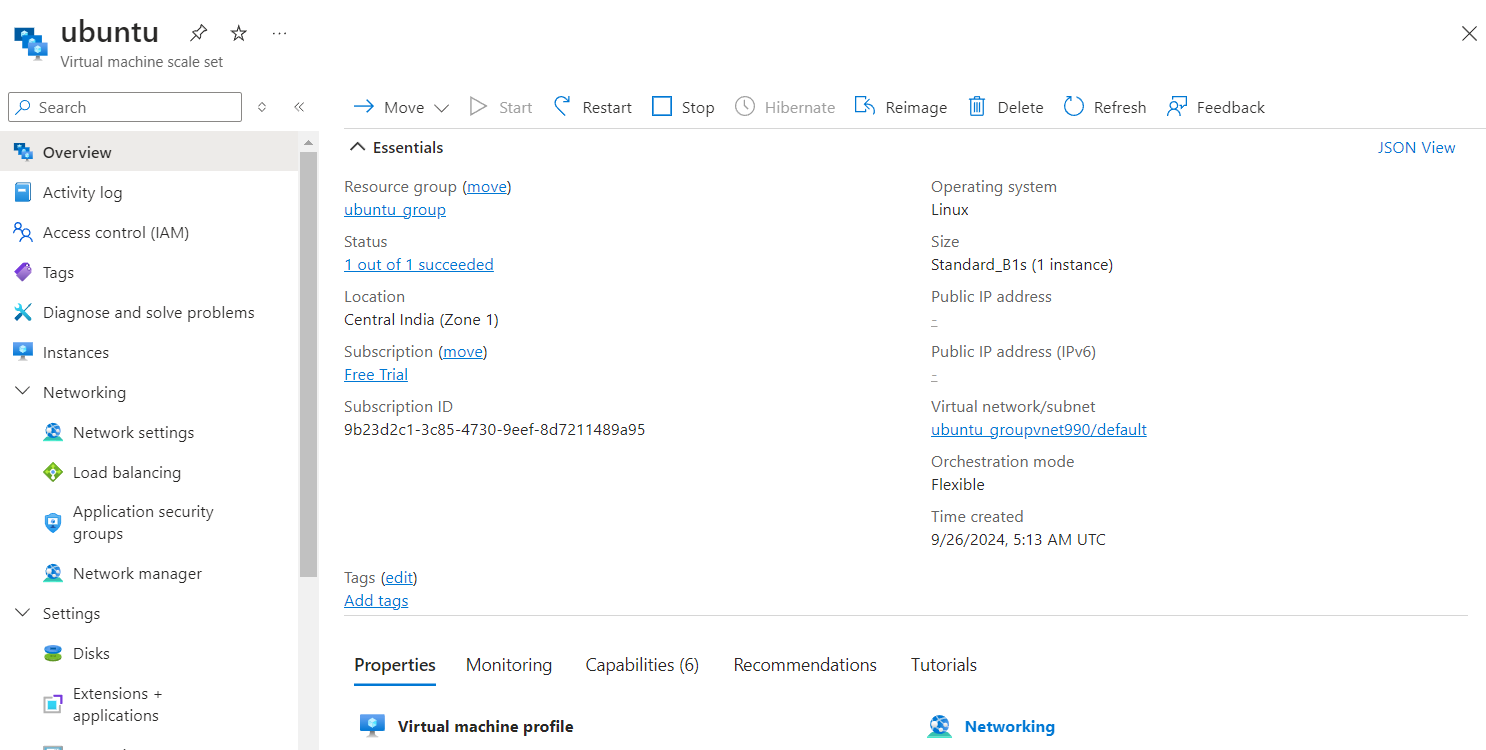


**Step 4: Networking Configuration**

1. **Availability Zones**:
   * During scale set creation, specify **Availability zones** to ensure that all VMs are deployed within the same zone.
   * This can be found in the **Availability options** section.

**Step 5: Verify Configuration**

1. **Review and Create**:
   * Go through all the configurations.
   * Click **Create** to deploy the scale set.



1. **Monitor and Test**:
   * After deployment, monitor the CPU utilization through the Azure Monitor or Insights to ensure scaling works as expected.
   * Simulate load testing to verify that the scale-out and scale-in rules are functioning correctly.

**Step 6: Ongoing Management**

1. **Regular Monitoring**:
   * Utilize Azure Monitor to set alerts based on scaling conditions.
   * Review the performance regularly to adjust scaling thresholds if necessary.
2. **Failure Management**:
   * Set up Azure Backup or Azure Site Recovery to ensure that your application is resilient to failures.
   * Ensure that other fault tolerance mechanisms are in place.