**Demo Script for OCI Workshop**

**1. Create Compartment, VCN, and Subnet**

**Explanation:**  
Compartments help organize resources within OCI, while a Virtual Cloud Network (VCN) provides a secure, isolated network environment for your resources. Subnets allow further segmentation of the VCN.

**Script:** “Let’s start by organizing our resources. I’ll log into the OCI Console. First, I’ll navigate to **Identity & Security** and select **Compartments**. Here, I’ll create a new compartment. [Create a compartment with a name and description.]

Now, I’ll go to **Networking** and select **Virtual Cloud Networks** to create a new VCN. I’ll enter a CIDR block, such as 10.0.0.0/16, and click **Create**.

Next, I’ll create a subnet under this VCN. I’ll click on the **Subnets** tab and provide a CIDR block, like 10.0.1.0/24, and designate it as a public or private subnet. Let’s create it!”

**2. Create Linux and Windows VMs**

**Explanation:**  
Launching virtual machines allows you to run applications in the cloud. Different operating systems can be deployed based on your needs.

**Script:** “Now, we’ll launch our virtual machines. I’ll navigate to **Compute** and click on **Instances**.

First, I’ll create a Linux VM. [Choose a Linux image, configure the settings, and select the subnet we created.] Now, I’ll repeat the process for a Windows VM. [Select a Windows image and configure the VM settings accordingly.]

Both VMs are now being launched.”

**3. Connect to VMs via SSH and RDP**

**Explanation:**  
SSH provides command-line access to Linux VMs, while RDP enables graphical access for managing Windows VMs.

**Script:** “Let’s connect to our Linux VM using SSH. I’ll open my SSH client, input the public IP address, and use my SSH key to connect. [Demonstrate the SSH connection.]

Now, for the Windows VM, I’ll use Remote Desktop Connection. I’ll enter the public IP and log in with the administrator credentials. [Demonstrate the RDP connection.]”

**4. Create Console Connections to Each VM**

**Explanation:**  
Console connections allow direct access to the VM’s terminal for management, useful if SSH or RDP is unavailable.

**Script:** “I’ll show you how to create console connections for both VMs. I’ll go to the instance details for the Linux VM and click on **Console Connection**. [Create the connection.]

I’ll do the same for the Windows VM. This will allow us to access the terminal directly if SSH or RDP is unavailable.”

**5. Create Custom Images and Launch VMs from Them**

**Explanation:**  
Custom images are snapshots of VMs that facilitate rapid and consistent deployment of new instances.

custom image is a copy of an instance's boot disk that can be used to launch other instances. Custom images include the software, configuration, and customizations installed when the image was created.

**Script:** “Next, I’ll create a custom image from our Linux VM. I’ll select the VM, click on **Create Image**, and provide a name.

Once the image is created, I’ll launch a new VM using this custom image by selecting it under **Create Instance**. This is great for consistent deployments!”

**6. Launch and Connect to an Oracle Linux Instance Using Local Console and Cloud Shell**

**Explanation:**  
Local console access provides direct terminal interaction, while Cloud Shell is a browser-based CLI for managing resources.

**Script:** “I’ll now demonstrate connecting to our Oracle Linux instance using the local console. I’ll create a console connection as shown earlier and connect.

Next, I’ll open Cloud Shell from the OCI Console. In Cloud Shell, I’ll SSH into the instance using its private IP address. [Demonstrate the connection.]”

**7. Launch and Connect to a Windows Instance Using Local Console**

**Explanation:**  
Using local console access for Windows provides management capabilities similar to RDP but without requiring configuration.

**Script:** “Now, let’s access the Windows instance using the local console. I’ll create a console connection and connect. This gives me direct management access to the VM without RDP.”

**8. Clone and Troubleshoot Unresponsive VMs**

**Explanation:**  
Cloning a VM creates a backup, while troubleshooting involves identifying issues like high resource usage or network problems.

**Script:** “If a VM becomes unresponsive, we can clone it for backup. I’ll select our Linux VM, click **Create Clone**, and provide a name.

For troubleshooting, I’ll check the instance status and logs to see if there’s an issue. If necessary, I can force stop and restart the VM. [Demonstrate stopping and starting the VM.]”

**9. Oracle Cloud Agents Status and Behavior**

**Explanation:**  
Oracle Cloud Agents run on VMs to provide monitoring and management functions. Their status is crucial for maintaining resource health.

**Script:** “Now let’s check the Oracle Cloud Agent status. I’ll navigate to the VM instance details and look for the agent’s status. This agent provides vital monitoring and management functions for our VMs.”

**10. Force Stop and Start VMs**

**Explanation:**  
Force stopping or starting a VM is useful for resetting unresponsive instances to restore normal operations.

**Script:** “I’ll demonstrate how to force stop a VM. [Select the VM, go to the instance menu, and choose Force Stop.] Now, I’ll start it again to show how to reset it if it becomes unresponsive.”

**11. Create VNIC Interfaces for VMs**

**Explanation:**  
Virtual Network Interface Cards (VNICs) enable VMs to connect to different networks and subnets, enhancing network management.

**Script:** “Let’s create Virtual Network Interface Cards (VNICs) for our VMs. I’ll go to the VCN, click **Create VNIC**, and attach it to the desired VM. This allows for better network management and traffic routing.”