**Q1 – Create a virtual machine named XYZ with 20gb of static storage, private network connection using type 2 hypervisor. Preserve the state of the machine.**

Ans - Sure, here are the steps to create a virtual machine named XYZ with 20GB of static storage, private network connection using type 2 hypervisor and preserving the state of the machine in VM workstation:

1. Launch the VMware Workstation application on your computer.

2. Click on the "Create a New Virtual Machine" option from the home screen.

3. In the New Virtual Machine Wizard, select "Typical" and click "Next".

4. Choose the "Installer disc image file" option and browse for the installation ISO file, or select "I will install the operating system later" option if you plan to install the operating system later.

5. Select the guest operating system and version that you want to install on the virtual machine and click "Next".

6. Name the virtual machine as "XYZ" and choose the location where you want to store the virtual machine files.

7. Set the disk capacity to 20GB and choose the "Allocate all disk space now" option to make it a static disk, and click "Next".

8. Configure the virtual machine's network settings by selecting the "Use network address translation (NAT)" option to create a private network connection for the virtual machine and click "Finish".

9. Before powering on the virtual machine, you can customize the virtual hardware settings as per your requirements.

10. Finally, power on the virtual machine and install the operating system or use the existing operating system installation.

To preserve the state of the virtual machine, you can either take a snapshot of the virtual machine or shut down the virtual machine properly without discarding changes made to the virtual machine. To take a snapshot, follow these steps:

1. Click on the "VM" menu from the VMware Workstation application and select "Snapshot".

2. Click on the "Take Snapshot" button and provide a name and description for the snapshot.

3. Click on "OK" to take the snapshot.

By taking a snapshot, you can preserve the state of the virtual machine at a particular point in time, and revert back to that state whenever required.

**Q2 - create a virtual machine named abc with 1 gb of Ram and dynamic storage of 20 gb using type 2 hypervisor. create an independent copy of the same. explain the steps in VM workstation**

Ans - Certainly, here are the steps to create a virtual machine named abc with 1GB of RAM and dynamic storage of 20GB using type 2 hypervisor, and creating an independent copy of the same in VMware Workstation:

1. Launch the VMware Workstation application on your computer.

2. Click on the "Create a New Virtual Machine" option from the home screen.

3. In the New Virtual Machine Wizard, select "Typical" and click "Next".

4. Choose the "Installer disc image file" option and browse for the installation ISO file, or select "I will install the operating system later" option if you plan to install the operating system later.

5. Select the guest operating system and version that you want to install on the virtual machine and click "Next".

6. Name the virtual machine as "abc" and choose the location where you want to store the virtual machine files.

7. Set the RAM to 1GB and choose the "Use a virtual hard disk" option to create a virtual hard disk, and click "Next".

8. Select the "Create a new virtual disk" option and set the disk capacity to 20GB and select the "Allocate space on demand" option to make it a dynamic disk, and click "Next".

9. Configure the virtual machine's network settings by selecting the appropriate option and click "Finish".

10. Before powering on the virtual machine, you can customize the virtual hardware settings as per your requirements.

11. Finally, power on the virtual machine and install the operating system or use the existing operating system installation.

To create an independent copy of the virtual machine, follow these steps:

1. Right-click on the virtual machine in the VMware Workstation Library and select "Manage" > "Clone".

2. In the Clone Virtual Machine Wizard, provide a name and location for the cloned virtual machine and click "Next".

3. Choose the appropriate cloning options, such as whether to create a linked or full clone and whether to make the clone independent of the original virtual machine.

4. Review the cloning settings and click "Finish" to start the cloning process.

5. Once the cloning process is completed, you can power on the cloned virtual machine and use it independently of the original virtual machine.

By creating an independent copy of the virtual machine, you can have a backup or a separate instance of the virtual machine to perform specific tasks or testing, without affecting the original virtual machine.

**Q3 – A user wants to deploy multiple copies of a vpn. Apply the suitable method for this in vsphere webclient environment.**

Ans – In the VMware Hands-on Lab environment, the most suitable method to deploy multiple copies of a VPN would be to use vSphere Templates.

Here are the steps to deploy multiple copies of a VPN using vSphere Templates in the VMware Hands-on Lab:

1. Launch the vSphere Web Client by clicking on the "vSphere Web Client" link on the desktop.

2. Log in using the credentials provided in the lab instructions.

3. Click on the "VMs and Templates" tab in the left-hand pane to view the virtual machines and templates in the inventory.

4. Right-click on the virtual machine that you want to use as the template for the VPN and select "Convert to Template".

5. In the "Convert to Template" wizard, specify the name and location of the template and click "OK".

6. Deploy multiple copies of the VPN by using the vSphere Template to create new virtual machines.

7. Right-click on the vSphere Template and select "Deploy Virtual Machine from this Template".

8. In the "Deploy from Template" wizard, specify the virtual machine name, location, and other details as per your requirements.

9. Choose the appropriate customization options, such as network settings and configuration details, and click "Finish".

10. Once the deployment is complete, you can power on the virtual machines and use them as VPN servers.

By using vSphere Templates, you can quickly deploy multiple copies of a VPN with the same specifications and configuration settings, which saves time and reduces the chance of errors. You can also customize each virtual machine as per your requirements, and maintain consistency across all the virtual machines.

**Q4 – Demonstrate an efficient method to search the currently running virtual machines of the same type of operating system in vshphere webclient.**

Ans - In vSphere Web Client, you can use the search function to efficiently search for currently running virtual machines of the same type of operating system. Here are the steps to do so:

1. Log in to the vSphere Web Client.

2. Click on the "VMs and Templates" tab in the left-hand pane to view the virtual machines in the inventory.

3. Click on the "Filter Results" button at the top of the inventory pane.

4. In the "Filter Virtual Machines" dialog box, click on the "Name" dropdown menu and select "Guest OS Family".

5. In the "Value" field, type the name of the operating system family that you want to search for, such as "Windows" or "Linux".

6. Click on the "Apply" button to apply the filter.

7. The inventory pane will now display only the virtual machines that are currently running and match the specified operating system family.

8. To further refine the search, you can click on the "Show Advanced Options" link in the "Filter Virtual Machines" dialog box and specify additional search criteria, such as the name of the virtual machine or the datastore where it is located.

9. Once you have found the virtual machines that you are looking for, you can perform various operations on them, such as powering them on or off, migrating them to a different host, or editing their configuration settings.

By using the search function in vSphere Web Client, you can quickly and efficiently search for currently running virtual machines of the same type of operating system, and perform various operations on them as needed.

**Q5 – Create a category 'Cloud' and then create a tag Virtualization under this category that will be associated with Vms, Host and Clusters only Allow multiple tags from the category to be applied to an object at any one time**

Ans - Here are the steps to create a category 'Cloud' and a tag 'Virtualization' in vSphere Web Client and configure it to apply to virtual machines, hosts, and clusters:

1. Log in to vSphere Web Client with an account that has administrative privileges.

2. Click on the "Menu" button in the top left corner and select "Tags & Custom Attributes".

3. Click on the "Categories" tab and then click the "Add" button to create a new category.

4. In the "Add Category" dialog box, enter the name of the category as "Cloud" and click "OK".

5. Next, click on the "Tags" tab and then click the "Add" button to create a new tag.

6. In the "Add Tag" dialog box, enter the name of the tag as "Virtualization", select the "Cloud" category, and check the boxes next to "Virtual Machines", "Hosts", and "Clusters" under the "Associable Object Types" section.

7. Click "OK" to create the new tag.

8. To apply the tag to an object, select the object (e.g. a virtual machine, host, or cluster) in the vSphere Web Client, and click on the "Tags" tab in the right-hand panel.

9. Click the "Assign Tags" button, select the "Virtualization" tag from the list, and click "OK".

10. You can also apply multiple tags from the same category to an object at the same time by selecting multiple tags in the "Assign Tags" dialog box and clicking "OK".

Once you have created the 'Cloud' category and 'Virtualization' tag and associated them with virtual machines, hosts, and clusters, you can easily search for and filter objects based on these tags in the vSphere Web Client.

**Q6 - Simulate the cloud environment by creating a datacenter with two hosts and run te cloudlets on it.**

Ans - To simulate a cloud environment using CloudSim with a datacenter consisting of two hosts and running cloudlets on it, you can follow these steps:

1. Install CloudSim on your local machine or server and set up the environment.

2. Create a new Java class in your CloudSim project and name it "DatacenterExample".

3. Import the necessary CloudSim packages at the top of your class file:

```

import org.cloudbus.cloudsim.Cloudlet;

import org.cloudbus.cloudsim.CloudletSchedulerSpaceShared;

import org.cloudbus.cloudsim.Datacenter;

import org.cloudbus.cloudsim.DatacenterBroker;

import org.cloudbus.cloudsim.DatacenterCharacteristics;

import org.cloudbus.cloudsim.Host;

import org.cloudbus.cloudsim.Pe;

import org.cloudbus.cloudsim.Storage;

import org.cloudbus.cloudsim.Vm;

import org.cloudbus.cloudsim.VmAllocationPolicySimple;

import org.cloudbus.cloudsim.VmSchedulerSpaceShared;

import org.cloudbus.cloudsim.core.CloudSim;

import org.cloudbus.cloudsim.provisioners.BwProvisionerSimple;

import org.cloudbus.cloudsim.provisioners.PeProvisionerSimple;

import org.cloudbus.cloudsim.provisioners.RamProvisionerSimple;

```

4. Define the main method of your class to initialize the CloudSim environment and create the hosts, datacenter, and broker. Here is some sample code for creating two hosts and a datacenter:

```

public static void main(String[] args) {

int numUser = 1; // number of cloud users

Calendar calendar = Calendar.getInstance();

boolean traceFlag = false; // trace events

CloudSim.init(numUser, calendar, traceFlag);

// Create two hosts

int numPe = 1; // number of CPUs per host

int mips = 1000; // total MIPS capacity of each host

int ram = 2048; // total RAM capacity of each host

int storage = 1000000; // total storage capacity of each host

int bw = 10000; // total bandwidth capacity of each host

List<Pe> peList = new ArrayList<Pe>();

peList.add(new Pe(0, new PeProvisionerSimple(mips)));

List<Host> hostList = new ArrayList<Host>();

hostList.add(new Host(0, new RamProvisionerSimple(ram), new BwProvisionerSimple(bw), storage, peList, new VmSchedulerSpaceShared(peList)));

hostList.add(new Host(1, new RamProvisionerSimple(ram), new BwProvisionerSimple(bw), storage, peList, new VmSchedulerSpaceShared(peList)));

// Create a datacenter

String name = "Datacenter";

DatacenterCharacteristics characteristics = new DatacenterCharacteristics("x86", "Linux", "Xen", new ArrayList<Host>(), 10.0, 3.0, 0.05, 0.001, 0.0);

VmAllocationPolicySimple policy = new VmAllocationPolicySimple(hostList);

Datacenter datacenter = null;

try {

datacenter = new Datacenter(name, characteristics, policy, new ArrayList<Storage>(), 0);

} catch (Exception e) {

e.printStackTrace();

}

// Create a datacenter broker

DatacenterBroker broker = null;

try {

broker = new DatacenterBroker("Broker");

} catch (Exception e) {

e.printStackTrace();

}

}

```

5. Create some virtual machines (VMs)

6. After creating the broker, create some virtual machines (VMs) and add them to the broker:

```

List<Vm> vmList = new ArrayList<Vm>();

int vmid = 0;

int mipsVM = 1000;

long sizeVM = 10000; // image size (MB)

int ramVM = 512; // vm memory (MB)

long bwVM = 1000;

Vm vm1 = new Vm(vmid++, broker.getId(), mipsVM, 1, ramVM, (int) bwVM, sizeVM, "Xen", new CloudletSchedulerSpaceShared());

Vm vm2 = new Vm(vmid++, broker.getId(), mipsVM, 1, ramVM, (int) bwVM, sizeVM, "Xen", new CloudletSchedulerSpaceShared());

vmList.add(vm1);

vmList.add(vm2);

broker.submitVmList(vmList);

```

7. Create some cloudlets to be executed on the VMs:

```

List<Cloudlet> cloudletList = new ArrayList<Cloudlet>();

int id = 0;

long length = 1000;

long fileSize = 300;

long outputSize = 300;

UtilizationModel utilizationModel = new UtilizationModelFull();

Cloudlet cloudlet1 = new Cloudlet(id++, length, 1, fileSize, outputSize, utilizationModel, utilizationModel, utilizationModel);

Cloudlet cloudlet2 = new Cloudlet(id++, length, 1, fileSize, outputSize, utilizationModel, utilizationModel, utilizationModel);

cloudlet1.setUserId(broker.getId());

cloudlet2.setUserId(broker.getId());

cloudletList.add(cloudlet1);

cloudletList.add(cloudlet2);

```

8. Submit the cloudlets to the broker:

```

broker.submitCloudletList(cloudletList);

```

9. Start the simulation and get the results:

```

CloudSim.startSimulation();

List<Cloudlet> resultList = broker.getCloudletReceivedList();

CloudSim.stopSimulation();

printCloudletList(resultList);

```

10. Run the program and check the output. If everything is set up correctly, you should see the cloudlets being executed on the VMs in the datacenter.

Note: This is a basic example of simulating a cloud environment with CloudSim. You can modify the code to create more hosts, VMs, and cloudlets, and add more features and configurations to the datacenter and the broker to make the simulation more realistic and complex.

**Q7 - Demonstrate the use of container technology by running any application on it**

Ans - Sure, here is an example of running a simple web application using container technology, specifically Docker:

1. First, create a Dockerfile that specifies the application and its dependencies. Here is an example of a Dockerfile for a simple Python Flask web application:

```

FROM python:3.8-slim-buster

WORKDIR /app

COPY requirements.txt .

RUN pip install -r requirements.txt

COPY . .

EXPOSE 5000

CMD [ "python", "app.py" ]

```

This Dockerfile specifies that the base image is Python 3.8, installs the dependencies specified in `requirements.txt`, copies the current directory to the `/app` directory in the container, exposes port 5000, and runs `app.py` when the container is started.

2. Build the Docker image using the Dockerfile:

```

docker build -t my-flask-app .

```

This command builds the Docker image and tags it with the name `my-flask-app`.

3. Run the Docker container:

```

docker run -p 5000:5000 my-flask-app

```

This command starts the container and maps port 5000 of the container to port 5000 of the host system.

4. Access the web application in a web browser by visiting `http://localhost:5000/`.

That's it! You now have a simple Flask web application running inside a Docker container. Container technology allows you to package applications and their dependencies into a portable and lightweight format, making it easier to deploy and manage applications across different environments.

**Q8 - Go Create a warning alert on Host if the Host CPU usage exceeds 50% for 10 minutes or critical if host memory usage is 90% for 5 min. Shutdown the host**

Creating a warning or critical alert and shutting down a host if it exceeds certain resource thresholds can help ensure the stability and reliability of your virtual environment. Here are the steps to create such alerts and automate the shutdown of a host on vSphere web client:

1. Open the vSphere web client and navigate to the "Hosts and Clusters" view.

2. Right-click on the host you want to create the alert for and select "Monitor" > "Create Alarm".

3. In the "Create Alarm" window, give the alarm a name, such as "High CPU Usage", and select the "Host System" object type.

4. Under "Triggers", click the "+" button to add a new trigger. Select "Host CPU Usage" as the metric and set the trigger condition to "Is Above" 50%. Set the "Repeat this trigger..." option to 10 minutes to ensure the host has been above the threshold for at least 10 minutes before triggering the alert.

5. Under "Actions", click the "+" button to add a new action. Select "Send a notification email" and enter the email address to receive the alert.

6. Repeat steps 3-5 to create a new alarm for high memory usage. Name the alarm "High Memory Usage" and set the trigger condition to "Is Above" 90% for 5 minutes.

7. To automate the shutdown of a host when a critical alarm is triggered, you can create a new "Host Power Policy" and set it to "Shutdown" when the "Host System" object triggers a "Critical" alarm. To do this, go to the "Configuration" tab of the host, select "Power Management", and click "Edit". Create a new power policy and set the "Action" to "Shutdown" and the "Trigger" to "Critical alarm".

Once these alerts and automated actions have been set up, you can ensure the stability and reliability of your virtual environment by proactively managing resource usage on your hosts.

**Q9 - Enable a method in Vsphere environment to provide continuous access of us at the times of host failure also Reserve 20% of cpu resource for failover.**

Ans - To provide continuous access to virtual machines even in the event of a host failure and reserve CPU resources for failover, you can use VMware High Availability (HA) feature in vSphere environment. Here are the steps to enable VMware HA:

1. Open the vSphere web client and navigate to the "Cluster" view.

2. Select the cluster you want to enable VMware HA for and click "Configure" > "vSphere Availability".

3. In the "vSphere HA" tab, click "Edit" and enable "Turn ON vSphere HA".

4. Under "Admission Control", select "Percentage of cluster resources" as the policy and set the "Failover Resources" to 20%. This will reserve 20% of the CPU and memory resources for failover use.

5. Under "VM Monitoring", select "VM and Application Monitoring" and set the "VM Monitoring Sensitivity" to "High".

6. Under "Heartbeat Datastores", select at least two datastores that are accessible from all hosts in the cluster.

7. Review the "Advanced Options" to customize any additional settings as needed.

8. Click "OK" to save the changes.

Once VMware HA is enabled, it will continuously monitor the health of virtual machines and host servers in the cluster. In the event of a host failure, VMware HA will automatically restart the virtual machines on another available host in the cluster, ensuring continuous access to the virtual machines. The reserved 20% of CPU resources will ensure that there is sufficient capacity to handle failover situations.

**Q10 - Enable a feature in vsphere environment to provide load balancing within the cluster Select the source automatically and destination manually -**

Ans - To provide load balancing within a cluster in vSphere environment and select the source automatically while allowing manual selection of the destination, you can use VMware Distributed Resource Scheduler (DRS) feature. Here are the steps to enable VMware DRS:

1. Open the vSphere web client and navigate to the "Cluster" view.

2. Select the cluster you want to enable VMware DRS for and click "Configure" > "vSphere DRS".

3. In the "vSphere DRS" tab, click "Edit" and enable "Turn ON vSphere DRS".

4. Under "Automation Level", select "Partially Automated" to allow automatic source selection and manual destination selection.

5. Under "DRS Migration Thresholds", set the "CPU and Memory Thresholds" to the desired levels. This will determine when DRS will initiate migrations of virtual machines between hosts to balance the load.

6. Under "VM Overrides", you can customize settings for individual virtual machines, such as preferred hosts and resource allocation.

7. Review the "Advanced Options" to customize any additional settings as needed.

8. Click "OK" to save the changes.

Once VMware DRS is enabled, it will continuously monitor the CPU and memory usage of hosts in the cluster and initiate migrations of virtual machines between hosts to balance the load. DRS will automatically select the source for migration based on the resource usage and requirements of the virtual machines. However, manual selection of the destination host will be allowed so that you can control where the virtual machines are migrated to.