Use Case1

To connect Internet workloads using Vnet peering and assign a custom role for operating these workloads

**Description:**The ABC Enterprises Corporation is evaluating Azure as a deployment platform. To help the company with its evaluation, you need to create virtual networks in the region specified by ABC Enterprises Corporation. You have to create test virtual machines in two virtual networks, establish connectivity between the two networks via VNet peering, and ensure connectivity is established properly.

To test the platform, ABC Enterprises Corporation wants to onboard an employee on the company’s default Azure Active Directory and assign a Custom RBAC role, under which they will be able to read the network and storage along with the VM. Under this custom RBAC, the employee should also be given permission to start and restart the VM. You have to onboard the employee under the default Azure AD and create a custom RBAC for the role of computer operator for this employee.

As a security measure, you need to ensure that the onboarded user can only access the resources mentioned in the custom role and adhere to the principle of least privilege.

**Prerequisites:**None

**Expected Deliverables:**

* Identify the networks
* Workload deployed to these networks
* Establishing the connectivity between these networks
* Onboard a user
* Create and assign a custom role to the user.

Objective :- VNet peering:-

Virtual network peering **enables you to seamlessly connect two or more Virtual Networks in Azure**. The virtual networks appear as one for connectivity purposes. The traffic between virtual machines in peered virtual networks uses the Microsoft backbone infrastructure.

**Solution :- Configure Peering Between vnet01 and vnet02**

1. In the Azure Portal, open the portal menu.
2. Click on the all resources option.
3. Click on the app1-vnet resource.
4. Click Peerings and then Add.
5. Set the following values: Name: "app1-to-core1" Deployment model: Resource manager. ...
6. Click OK.
7. Firewall d disable

test VNet peering:-

Sign in to the Azure portal. In the web app, select networking, and then select VNet Integration. Check whether you can see the remote virtual network. Manually enter the remote virtual network address space (Sync Network and Add Routes).

**RBAC implementation**

1. Inventory your systems. Figure out what resources you have for which you need to control access, if you don't already have them listed. ...
2. Analyze your workforce and create roles. ...
3. Assign people to roles. ...
4. Never make one-off changes. ...
5. Audit.

**How to Implement RBAC in 8 Steps**

1. Audit your current status. The first thing you need to do is audit your current status. ...
2. Define roles. ...
3. Query-level implementation. ...
4. Interface-level implementation. ...
5. Component-level implementation. ...
6. Testing roles and implementation. ...
7. Role assignment. ...
8. Monitoring and auditing.

Use Case2

To create high available architecture by distributing incoming traffic among healthy service instances in cloud services or virtual machines in a load-balanced set with the help of a command-line interface

**Description:**The ABC Enterprises Corporation wants to deploy a web application in a highly available environment so that only the healthy instances will be serving the traffic so end users will not be facing any downtime. They have decided to work on an Azure public load balancer to implement the functionality.

The operations team at ABC decides to define the entire architecture using the load balancer and its backend pool, once that’s in place they intend to create the frontend IP and health probe along with virtual machines housing their application.

ABC Enterprises works extensively on delivering highly available web applications for their users in a secure way by avoiding directly exposing the virtual machines hosting the applications to the public internet. The communication from the application in the VM to the end-user must take place via the Load Balancer.

The expectation of the operation team is to create a reusable method that can be used for automation if in the future we need to deploy the same kind of infrastructure. So, rather than deploying resources in the Azure portal, they should leverage the command-line interface to deploy the resources so that in the future these commands can be used

As a security measure, you need to ensure that only the health instances of the virtual machine will be serving the traffic.

**Prerequisites:**None

**Expected Deliverables:**

* Identify Virtual machines and Networking
* Configure the load balancer
* Extend the load balancer with backend pool and frontend IP
* Define the Health probe
* Extend the security with the bastion Hosts

Configure Load Balancer

Setup new resource group

2. Setup two new windows VM

3. Setup IIS with sample web page

4. Create Azure load balancer

5. Create a backend pool

6. Create health probes

7. Create load balancer rule

8. Testing

**Objective: -** An Azure load balancer is **a Layer-4 (TCP, UDP) load balancer that provides high availability by distributing incoming traffic among healthy VMs**. A load balancer health probe monitors a given port on each VM and only distributes traffic to an operational VM.

**Solution:-**

In the search box type "**load balancer**" => **Resource Group>** **Name>** **Region>** **Type (**Public ( We are going to load balance internet traffic )> **SKU** : Basic > **Public IP Address> Public IP address name** > **Assignment**: Static >**Review + Create>** **Create**

**Create a backend pool**

The backend pool is a critical component of the load balancer. The backend pool **defines the group of resources that will serve traffic for a given load-balancing rule**. There are two ways of configuring a backend pool: Network Interface Card (NIC) IP address.

search box type "**load balancer**"> load balancer home page click on MyloadBalancer> **Backend pools>** Click on **Add>** **Name** > **Virtual Network>** **Associated to** : Virtual Machine>  click on**Add** button to create a Backend pool.

**Create health probes**

Go to **Myloadbalancer**load balancer properties page> Click on **Health Probes>** Click on**Add (**In the form provide a name for the probe. Then leave the protocol like **TCP.** We are running web service on**port 80** so leave the default value as it is.)

**Create load balancer rule**

Go to **Myloadbalancer** load balancer properties page > Click on **Load balancing rules>** Click on**Add**

In my setup, I am load balancing **TCP 80** traffic. So my rule configuration as following,

**Name>** **IP Version>** **Front End IP address >** **Protocol** : TCP> **Port**: 80> **Backend port**: 80> **Backend pool**: NamePool1> **Health probe**: Webservice> click on **OK** to create the rule>click on **OK** to create the rule

**Testing**

This completes the configuration. It is time for testing.

I went ahead and launch the web browser of my laptop and try to access public ip address of the load balancer.

As expected, now I can see the web site running from **VM01** back end server.

## **prerequisites**

Here are the required NSG rules that need to be taken care of for proper traffic flow from the internet to bastion subnet and bastion subnet to VMs subnet –

* **Inbound –**
  + RDP and SSH connections from the Azure Bastion subnet to the target VM subnet
  + TCP port **443** access from the internet to the Azure Bastion public IP
  + TCP access from Azure Gateway Manager on ports **443** or **4443**
* **Outbound –**
  + TCP access from the Azure platform on port **443** to support diagnostic logging.

## **How to Configure Azure Bastion Host**

To configure Azure Bastion Administrator requires **Contributor**access to create it. In this section, you can learn how to Configure Azure Bastion Host?

* **Login** to the[Azure portal](https://portal.azure.com/)
* Search**“Bastion”** in global search box.
* Under services, click **Bastions**.

Azure Portal – Under services, click **Bastions**. Step by Step Guide to Configure Azure Bastion Host

* We have to fill option which has Asterisk (\*) sign on below screenshot then fill **Tags** and click on “**Review+Create**”:

Create a Bastion – Provide all required information – Azure Bastion Host

**Note:** After we selected the required Virtual Network, it shows the subnet, then we have to choose or create the subnet named “**AzureBastionSubnet,**”

## **Connect to a VM using Bastion**

* Login to [Azure portal](https://portal.azure.com/) and navigate to Azure VM that you want to connect.
* On the **overview** page select **Connect.**Then select **Bastion**from the dropdown.

Azure Portal – Select **Bastion** from the dropdown – Step by Step Guide to Configure Azure Bastion Host

* Click on **“Use Bastion”** button.

Azure Bastion Host – Click on the **Use Bastion** button.

* On the **Connect using Azure Bastion** page, enter the username and password for your virtual machine and select the check box **“Open in new window”** this will open our VM in new tab .
* Click on **Connect**.

Use Case3

ABC Corporation is a globally distributed firm. They have their headquarters in the **Central India**with another branch office in the **South India**. Currently, they are working on a project and decided that the application tier of this project will reside in one of its branch regions. For security reasons, ABC Corporation management is adamant on keeping their data tier in the headquarter region

**Background of the problem statement:** as an organization, they are open to suggestions and are currently evaluating Azure as a deployment platform. To prepare for the deployment of IaaS **Standard\_B1ms**, ABC Corporation must deploy an IaaS v2 virtual network in the headquarters region for its database. But for the application, it should create another IaaS v2 virtual network in the branch region. In addition, because the communication between App and data should happen over a private channel, one needs to prepare their branch office virtual network for establishing connectivity to the headquarter’s IaaS v2 virtual network by creating a virtual network gateway and deploy a test IaaS **Standard\_B1ms**VM to the virtual networks for verifying the connection.

After the deployment team ensures the connectivity between both the networks, you can validate the same using Ping.

**Following requirements should be met:**

* Create virtual networks in the aforementioned region
* Create test virtual machines in both the virtual networks
* Establish the connectivity between both the networks via VNet peering
* Ensure connectivity is established properly