Goal

Access VNet resources from Pods in your cluster. We will access the following

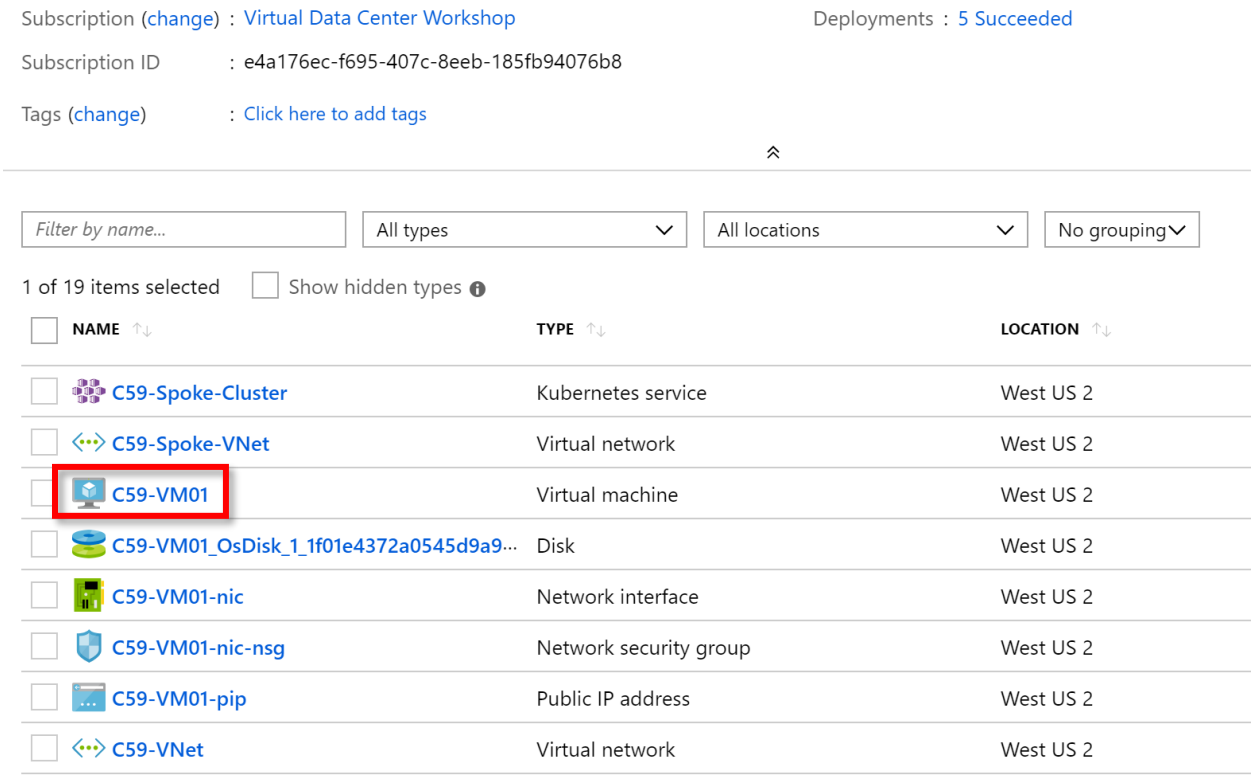
1. A website running on a VM in the hub VNet that is peered to the cluster (Spoke) VNet.
2. Retrieve a blob from a storage account tied to your cluster subnet and protected by Service Endpoints

Note: Run the lab from the same directory where you found this instructions file

Steps

Access a Website in your VNet

The website is running in a VM in the hub VNet. It is called "C*-VM01" where * represents your company no. See sample for Company59 below

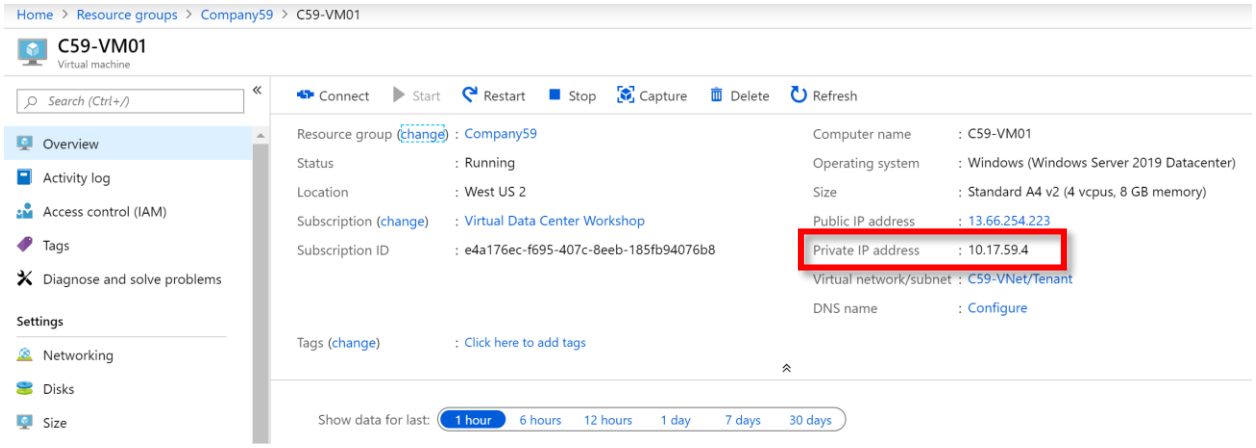


1. Create a Pod in your cluster to access the website from
   Create an Alpine Pod and get shell access.

   ```
   kubectl run samplepod --rm -it --image=alpine --generator=run-pod/v1
   ```

2. Get the IP Address of the VM

In the portal, click on the VM to go to the VM blade. You will find the Private IP on the blade. Make a note of it
Below is a sample for Company59



3. Access the website from this pod
   wget downloads the http file of the website. Specify the IP address of the VM here.
   `wget -qO- http://<VM_PrivateIP>`

4. Exit the Pod
   This is done by typing "exit" at the shell prompt

Accessing Storage Resource Protected by Service Endpoint

We will start by first creating a storage accounting and uploading a blob to it.

1. Setup a storage account and upload a blob
   Note: In all these commands replace '*' with the company no.

   //Create a storage account
   ```
   az storage account create -n c*aksstorage -g Company* --access-tier Hot --kind StorageV2 --sku Standard_RAGRS
   ```

   //Create a storage container
   ```
   az storage container create -n c*akscontainer --public-access blob --account-name c*aksstorage
   ```

   //Upload a blob to this container
   ```
   az storage blob upload -c c*akscontainer --account-name c*aksstorage -f ./blob.txt -n aksblob
   ```

   //Generate a URL for the blob
   ```
   az storage blob url -c c*akscontainer -n aksblob --account-name c*aksstorage
   ```

   Make a note of this URL

https://c59aksstorage.blob.core.windows.net/c59akscontainer/aksblob

2. Access the blob URL from your browser
   Observe that you are able to access the blob and view its contents

3. Setup ServiceEndpoint for your Storage Account
   Note: In all these commands replace '*' with the company no

   // Configure ServiceEndpoint for Storage on your AKS cluster subnet
   ```
   az network vnet subnet update --service-endpoints Microsoft.Storage
   -n Cluster -g Company* --vnet-name C*-Spoke-Vnet
   ```

   // Tie your Storage to your AKS cluster subnet
   ```
   az storage account update --resource-group Company* --name
   c*aksstorage --default-action Deny

   az storage account network-rule add -n c*aksstorage -g Company* --
   subnet Cluster --vnet-name C*-Spoke-Vnet
   ```

4. Access the blob URL
   Try accessing the URL from your browser and observe that you are denied access

   Now access the same URL from your Pod using the following commands

   //Create an Alpine Pod and get shell access.
   ```
   kubectl run samplepod --rm -it --image=alpine --generator=run-pod/v1
   ```

   //Access the blob URL from the shell
   ```
   wget -qO- <URL>
   ```

5. Exit the Pod
   This is done by typing "exit" at the shell prompt