Cloud Access Software Deployment on Azure Stack

Jeff Dillabough

February 13, 2019

[1. Introduction](#_Toc261713)

[2. Prerequisite](#_Toc261714)

[3. Setup the Azure Stack Infrastructure](#_Toc261715)

[Create a Resource Group](#_Toc261716)

[Create a Virtual Network and configure Subnets](#_Toc261717)

[4. Connect to the corporate network](#_Toc261718)

[Create a Virtual Network Gateway](#_Toc261719)

[Create the VPN connection to the Corporate Network](#_Toc261720)

[5. Install the Teradici Cloud Access Connector](#_Toc261721)

[Create the Virtual Machine for the Cloud Access Connector](#_Toc261722)

[Configure DNS](#_Toc261723)

[Install Cloud Access Connector](#_Toc261724)

[6. Deploy a Workstation](#_Toc261725)

[7. Connect Clients](#_Toc261726)

[8. Troubleshooting](#_Toc261727)

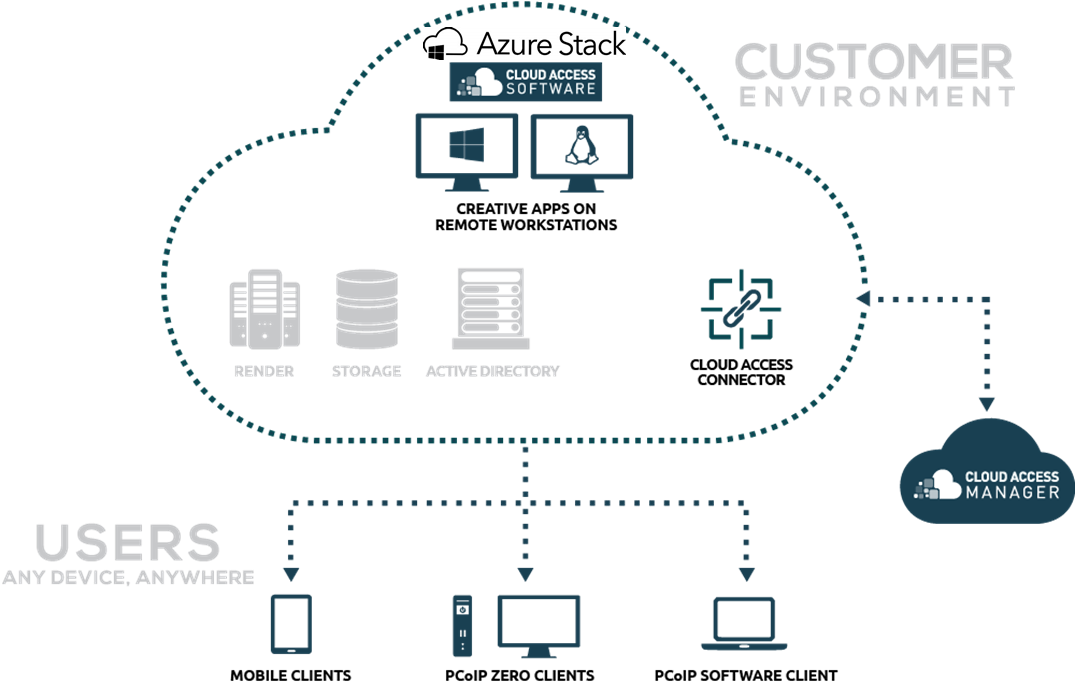
Version History

|  |  |  |
| --- | --- | --- |
| **Date** | **Description** | **Author** |
| 2019.1.28 | First Draft | Jeff DIllabough |
| 2019.1.31 | Minor content revisions, track changes on | John McVay |
| 2019.1.31 | Added Workstation and Clients section | John McVay |

# Introduction

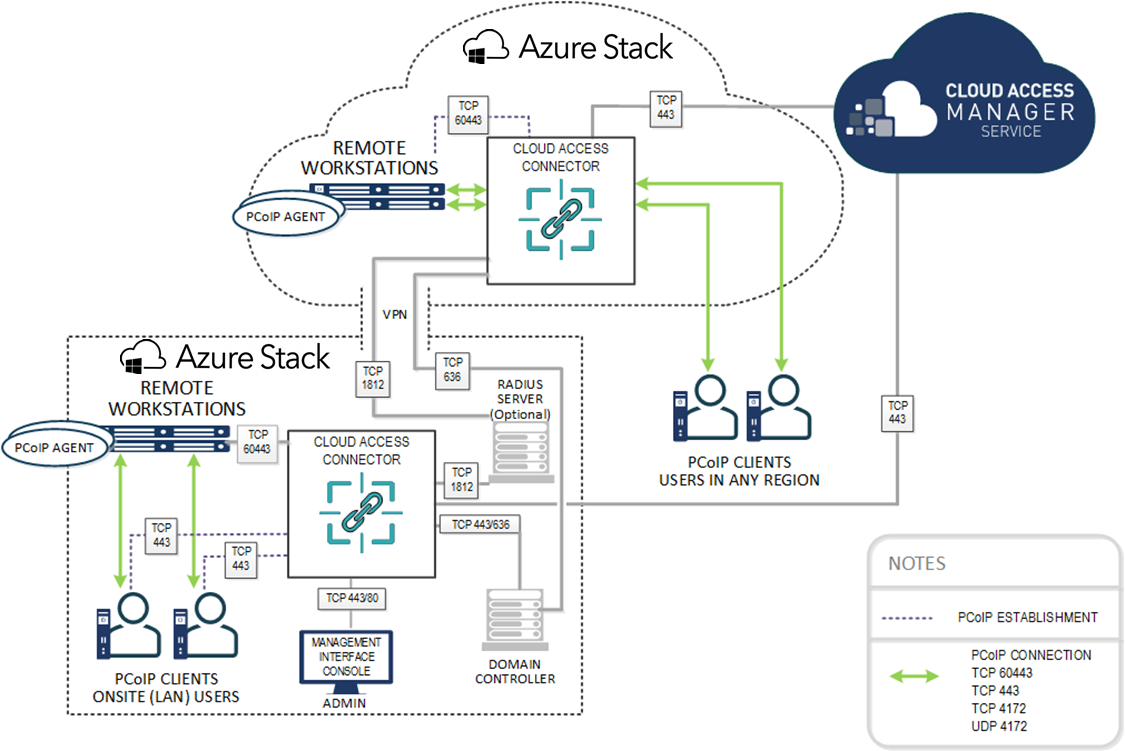
This document describes a Reference Architecture and procedure to set up the infrastructure to deploy Teradici Cloud Access Software on Microsoft Azure Stack. This Reference Architecture assumes that the Azure Stack deployment is connected to an existing corporate network and Windows domain (Domain Controller/Active Directory).

The following diagram highlights the Azure Stack and Teradici Cloud Access Software components and existing corporate network infrastructure components (shaded):



The Cloud Access Software Architecture allows for multiple Stacks or regions to be deployed separately but be managed as a single deployment. Azure Stack can be directly connected to the corporate network using an Azure Stack Local Network Gateway or deployed remotely from the corporate network and connected via an Azure Stack Virtual Network Gateway VPN. In addition, End Users connecting to their Remote Workstations can connect from within the corporate network or connect remotely over the Internet through either region’s publicly accessible Teradici Cloud Access Connector (CAC).

The following diagram shows an Azure Stack deployment consisting of two regions interconnected via a VPN using the Azure Stack Virtual Network Gateway. The bottom, on-premises region represents the directly connected corporate network and includes additional infrastructure such as Domain Controller and a Radius connection for Multi-factor authentication. End User connections to Remote Workstations in the directly connected Azure Stack are shown as being brokered entirely within the corporate network. In this case the Cloud Access Connector does not have an externally accessible public IP Address. The top region represents a remote site where End User connections are being brokered over an Internet accessible public interface. The Teradici Cloud Access for Azure Stack reference architecture is identical for either region and will be used as the example throughout this document.



This document will follow these steps:

1. Set up the Azure Stack resource group and virtual network infrastructure
2. Connect the Regions via VPN
3. Install a Cloud Access Connector
4. Deploy a Desktop/Workstation
5. Assign the Desktop/Workstation to a User
6. Broker an End User Session login to the Remote Desktop/Workstation

# Prerequisite

This document assumes an existing corporate network hosted on Azure Stack complete with Domain Controller and Cloud Access Connector as per bottom, on-premises region of the previous diagram. It should also be stated that Azure Stack Regions could equally be Azure Public Cloud Regions, as the Architecture is the same. Furthermore the example corporate network as shown in the bottom of the above diagram can be deployed on Azure Public Cloud via a simple Cloud Shell script, <https://www.teradici.com/web-help/pcoip_cloud_access_manager/current/>.

Parameters required from the existing corporate deployment include:

* Domain name (e.g. teradici.local)
* Domain administrator username and password (e.g. tera\_admin, tera\_admin\_pwd)
* Existing corporate network CIDR (e.g. 10.0.0.0/16)
* Domain Controller local IP Address (e.g. 10.0.0.4)

This document also assumes the person reading this reference architecture document has Azure Stack credentials with adequate permissions to create and install resources, knows how to connect a Windows or Linux machine to the Domain and has some familiarity with the Azure or Azure Stack Resource Manager portal.

# Setup the Azure Stack Infrastructure

## Create a Resource Group

Once logged into the Azure Stack Resource Manager portal, the first step is to create a resource group that will contain the deployed resources. This is easily done by navigating to the Resource Group tab and clicking Add. This will prompt you to name your resource group and click Create to complete the operation. For this reference architecture all resources will be deployed under this resource group.

## Create a Virtual Network and configure Subnets

A virtual network on Azure Stack will provide an address space that does not conflict with the corporate network and will set the basis for deployed virtual machines, VPN connectivity and gateways.

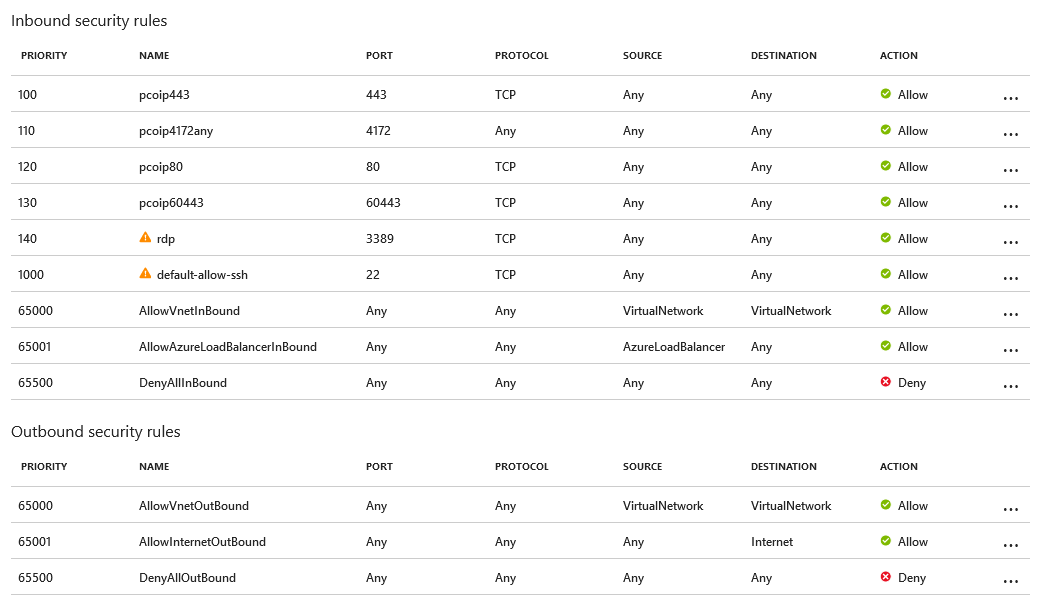
Navigate to Create a virtual network page. Assuming the corporate network is 10.0.0.0/16, create a network with an address space 10.1.0.0/16 and a name of your choosing. Select Use Existing Resource group and select the resource group created in step one. We also want to create 3 subnets within this address space:

* Subnet for the Cloud Access Connector(s), for example Subnet name “subnet-connector” with address range 10.1.0.128/26
* Subnet for the VPN gateway, example address range 10.1.0.64/26. The subnet name is fixed as “GatewaySubnet”
* Subnet for the deployed workstations, for example Subnet name “subnet-workstations” with address range 10.1.240.0/20

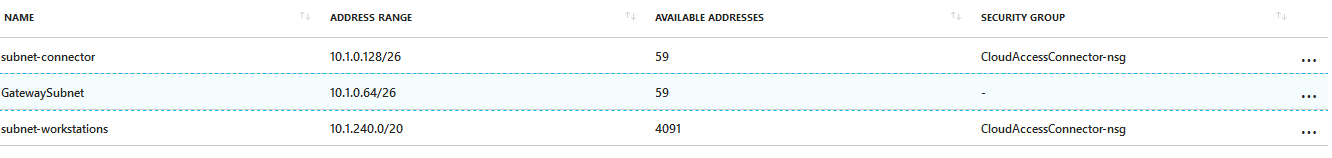
One of the connector or workstations subnets can be created when the vnet is created. The others can be added after the vnet is created. In the vnet tab the Gateway subnet is added by clicking “+ Gateway Subnet” and regular subnets by clicking the “+ Subnet”.

Security groups will help protect resources and are especially important on “subnet-connector” as the Cloud Access Connector deployed on this subnet will have a public IP address. Create a Network Security Group and apply it to the subnets, or better yet create the Network Security Group first and add to the subnet when creating them. The default security group will allow unlimited internal connectivity and allow unlimited connectivity out to the internet, but will block everything coming in from the internet. We need to allow internet access to the required PCoIP ports (TCP/UDP 4172, TCP 443, TCP 60443, TCP 80). We will also require SSH access to Linux virtual machines (TCP 22) and RDP access to windows virtual machines (TCP 3389). By establishing the VPN first we can access these through the VPN from the corporate network rather than from a public IP address and not open those ports in the security group.

Create the Network Security Group under the networking tab and once created add the PCoIP rules by navigating to the Network security group page and go to the Inbound security rules tab. Once here click “+ Add” to add new rules. The following diagram shows an example Network Security Group called “CloudAccessConnector-nsg” which also includes SSH and RDP ports (not recommended).



Once completed the subnet tab under the virtual network should look like this



# Connect to the corporate network

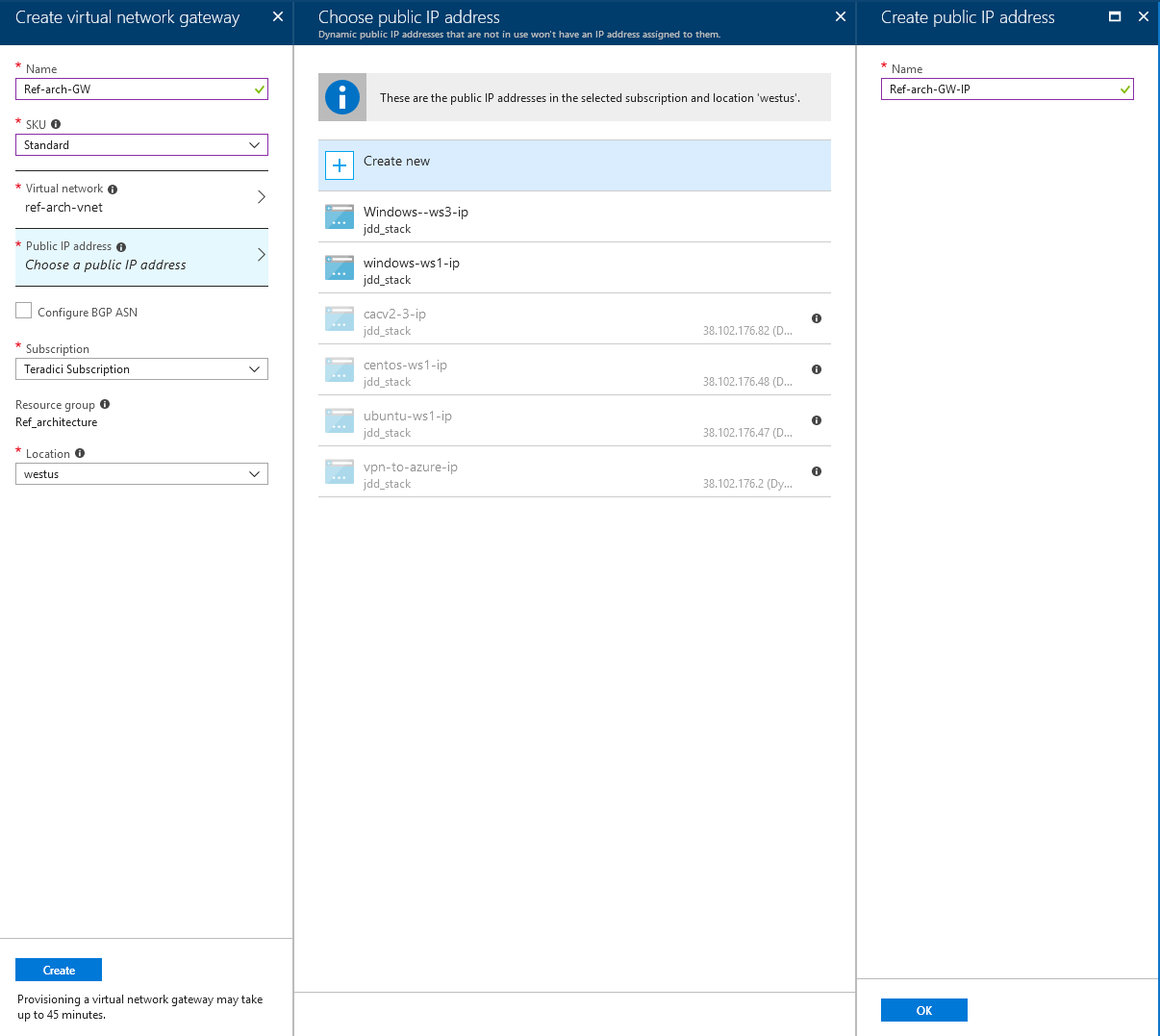
Teradici Cloud Access Manager (CAM) uses the Domain Controller to manage Users and virtual desktops. By connecting to the corporate network Domain Controller the CAM will have all the Users available to be assigned virtual desktops so they can be connected when End Users connect their PCoIP sessions.

## Create a Virtual Network Gateway

An easy and effective way to connect the new Azure Stack region to the corporate network is via the Azure Stack Virtual Network Gateway. This will be done on both the Azure Stack and the corporate network. There will be choices for type of route-based VPN: basic, standard and High-performance, more information can be found here https://docs.microsoft.com/en-us/azure/vpn-gateway/vpn-gateway-about-vpngateways. The Standard SKU should be sufficient but needs to be consideration for what traffic will be flowing across this VPN, for example is the file system only in one location resulting in a lot of file transfer.

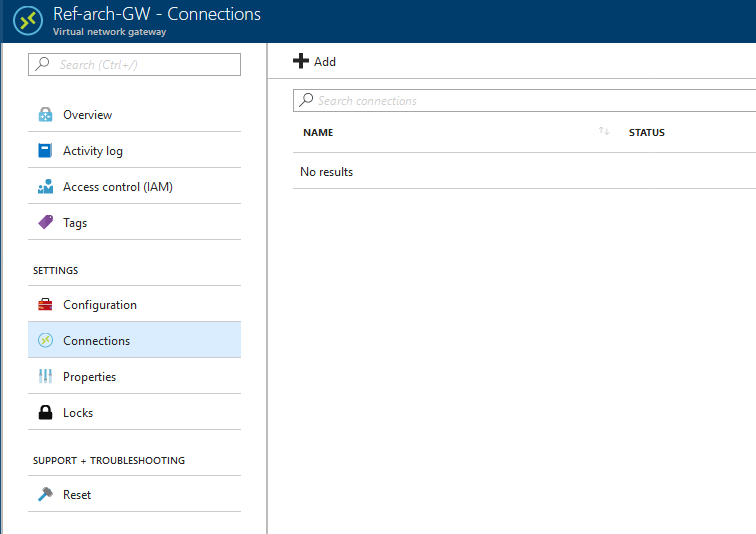
It is also possible for remote virtual desktops to be connected across this VPN. For example an End User works in the corporate headquarters and their machine is on the local corporate network. But when traveling it may be better to connect through the Azure Stack regional Cloud Access Connector and have the PCoIP Session connect to the virtual desktop through the VPN.

Assuming the Standard type of route-based VPN is sufficient, the gateway will need to be named, will need to select the Virtual Network created earlier, and will need to create a new Public IP address. The Create network gateway dialog will look like this and you will need the gateway on the corporate network side as well:

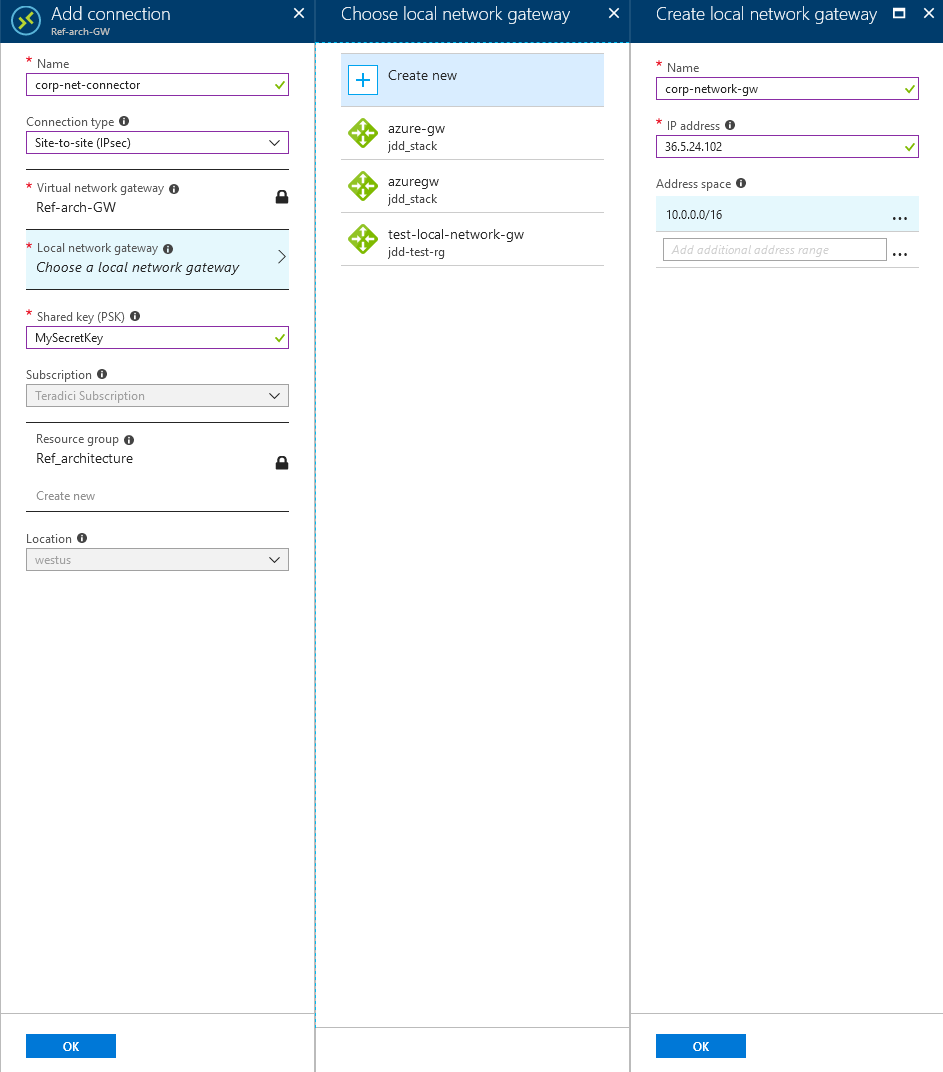


## Create the VPN connection to the Corporate Network

Once the network gateway is created, we need to add the connection to the corporate network. Navigate to the virtual network gateway just created, an easy way is through the resource group. Click on Connections and then “+Add”.

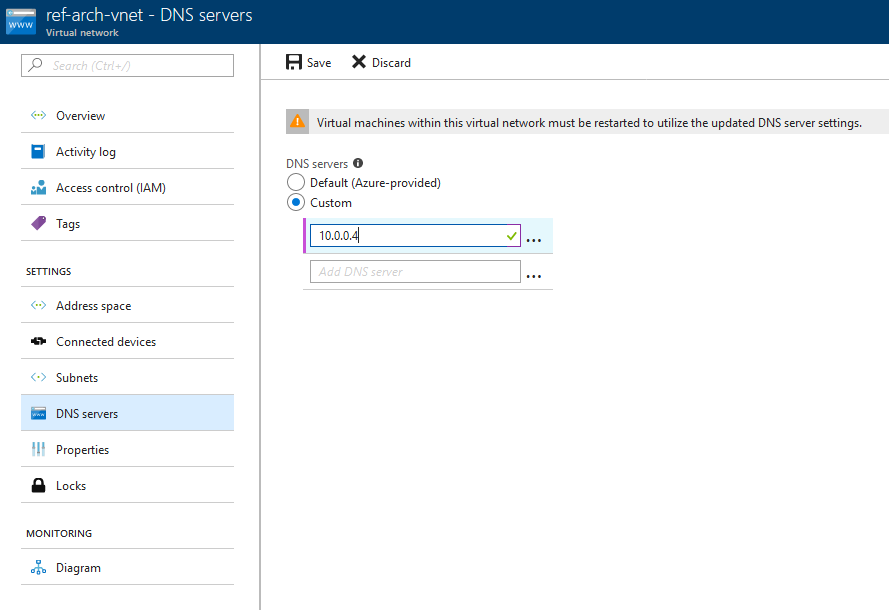


Click “+Add” will open a new dialog where the name, connection type and connection information about the other end of the VPN are entered. You will create a new “local network gateway” which is the gateway that you are connecting to on the other end, or in this case the corporate network. Note that we are selecting a Site-to-Site (IPsec) connection type and for the local network gateway we need its public IP address and the Address space which we identified earlier as 10.0.0.0/16 for the corporate network. We also need to enter a “Shared key” that will provide security to the connection.

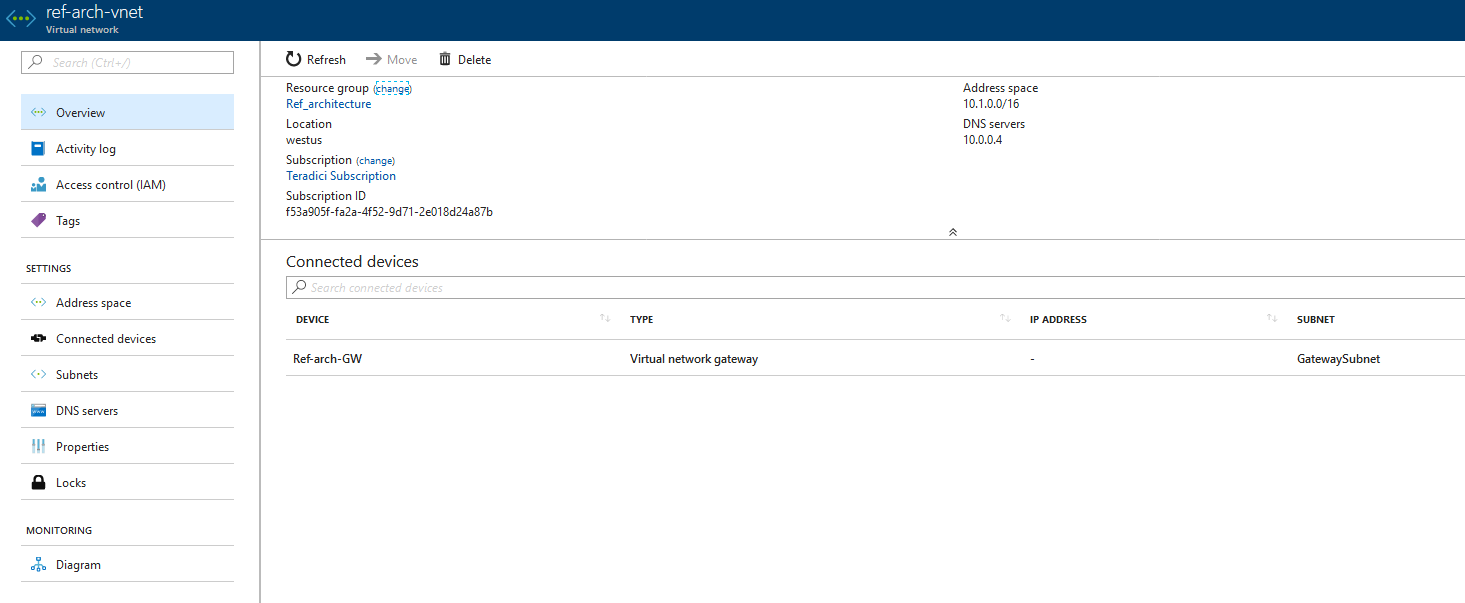


Once the connections are setup in each gateway the VPN will be established. This may take up to 45 minutes.

A final step once we have the VPN established is to set the DNS Server for the Azure Stack Deployment. This is done on the on the tab for the virtual network. In the Overview you will see that the DNS server is set to the “Azure provided DNS service”. We need this to point to the DNS in the corporate network which we said was 10.0.0.4. Clicking on the DNS servers button in the virtual network tab allows you to create a custom DNS server.



Once saved and the new DNS entry has been configured you should see this in the virtual network Overview. Notice top right DNS is now 10.0.0.4.



# Install the Teradici Cloud Access Connector

With the Azure Stack infrastructure complete we can now install the Teradici Cloud Access Connector (CAC). Instructions on CAC deployment can be found here:

<https://www.teradici.com/web-help/pcoip_cloud_access_manager/CACv2/>

## Create the Virtual Machine for the Cloud Access Connector

The CAC runs on an Ubuntu server 18.04 LTS with minimum 4GiB RAM, 12 GiB storage and 2 vCPUs. Creating this from the virtual machine tab will allow you to name the VM, select the size on the VM and provide settings. When creating the VM you will add a user name and password so you can log into the machine once built for further configuration. Add this new VM to the resource group. A machine size such as DS2\_v2 will be good for smaller deployments.

In the settings dialog you want to:

* None for availability set. If you are planning bigger deployments then this can be set.
* Set your storage preferences but defaults will work
* Make sure this VM is on the virtual network and subnet that was setup for the connector, “subnet-connector”
* This VM needs a (new) public IP address. A default IP address name will be provided but this can be changed. At the same time as changing or confirming the name this should be changed to a static IP address
* The network security group should already be set from the vnet but we want to make sure it is the one we setup earlier.

## Configure DNS

The CAC VM needs to use the deployment’s Domain Controller as the DNS to be able to resolve FQDN.

In the network security group there were entries for the SSH port 22. This is necessary to SSH into the Linux machine to install the Cloud Access Connector SW. With this firewall rule you can SSH into the CAC VM using the public IP address that was created as part of the VM create process. If this rule is left out of the network security group then you can still SSH into the machine from the corporate network but you will not be using the public IP address, instead you will use the private IP address that will be within the subnet address space. Both public and private IP addresses, along with the network security group firewall rules on the Virtual Machine Networking tab.

1. SSH into the CACv2 VM (using either the public or private IP address)
2. Change into root user

|  |
| --- |
| sudo su |

1. Edit /etc/systemd/resolved.conf, uncomment the DNS and Domains lines and use the Domain Controller’s internal IP address and the Domain handled by the Domain Controller:

|  |
| --- |
| [Resolve] DNS=10.0.0.4 Domains=syin.local |

1. Restart the resolved service

|  |
| --- |
| systemctl restart systemd-resolved.service |

1. You should now be able to ping the domain controller (remember to open Windows firewall to ping) and workstations from the CAC VM.

|  |
| --- |
| ping vm-dc.syin.local ping admin-rw.syin.local ping vm-dc ping admin-rw |

Note that your servers (domain controller or workstation) must be configured to respond to ping; if not, the name might be resolved to IP address but there will be no ICMP replies. For example, these instructions opens up the firewall for ICMP in Windows Server: <https://manage.accuwebhosting.com/knowledgebase/2609/How-to-Enable-PingorICMP-Echo-Request-in-Windows-Server.html>

## Install Cloud Access Connector

At this point, the CAC Software can be installed in the CAC VM. Follow the instructions [here](https://www.teradici.com/web-help/pcoip_cloud_access_manager/CACv2/installation/downloading_cac/) to download and install the Cloud Access Connector software.

# Deploy a Workstation

After the CAC software is installed in the CAC VM, you can begin deploying virtual workstations. There are multiple options for workstation deployment, including standard agents provided by Teradici to use with licenses acquired by you from the appropriate vendor. The PCoIP Standard Agent for Windows and Standard Agent for Linux enables Teradici customers to deliver virtual Windows and Linux desktops or custom applications to remote users as part of Teradici Cloud Access Software. The PCoIP Standard Agent is installed on a virtual machine in Azure Stack or Azure. End users connect to their remote desktops via a PCoIP client, either directly or via a connection broker. For standard agents, refer to [Standard Agent for Windows](https://docs.teradici.com/find/subscription/product/cloud-access-software/2.14/component/standard-agent-for-windows/2.14.0) and [Standard Agent for Linux](https://docs.teradici.com/find/subscription/product/cloud-access-software/2.14/component/standard-agent-for-linux/2.14.0).

# Connect Clients

Teradici PCoIP Software Clients are applications that establish PCoIP sessions with remote Windows or Linux desktops. Connections can be made to virtual machines via PCoIP agents, and to remote workstations via Remote Workstation Cards. Teradici supports Software and Mobile Clients, including [Software Client for Windows](https://docs.teradici.com/find/subscription/product/cloud-access-software/2.14/component/software-client-for-windows/3.7.0), [Software Client for MacOS](https://docs.teradici.com/find/subscription/product/cloud-access-software/2.14/component/software-client-for-macos/3.7.0), and [Software Client for Linux](https://docs.teradici.com/find/subscription/product/cloud-access-software/2.14/component/software-client-for-linux). Mobile Clients include [Mobile Client for Android Tablets](https://docs.teradici.com/find/subscription/product/cloud-access-software/2.14/component/mobile-client-for-android-tablets), [Mobile Client for iOS Tablets](https://docs.teradici.com/find/subscription/product/cloud-access-software/2.14/component/mobile-client-for-ios-tablets), and [Mobile Client for Chromebooks](https://docs.teradici.com/find/subscription/product/cloud-access-software/2.14/component/mobile-client-for-chromebooks).

# Troubleshooting

OpenVPN not routing traffic.

Some things to check on the OpenVPN VMs:

* IP forwarding turned on
* Iptables not blocking traffic
  + # iptables -L  
    Chain INPUT (policy ACCEPT)  
    target prot opt source destination   
    sshguard all -- anywhere anywhere   
    Chain FORWARD (policy ACCEPT)  
    target prot opt source destination   
    Chain OUTPUT (policy ACCEPT)  
    target prot opt source destination   
    Chain sshguard (1 references)  
    target prot opt source destination
  + # iptables -t nat -L  
    Chain PREROUTING (policy ACCEPT)  
    target prot opt source destination   
    Chain INPUT (policy ACCEPT)  
    target prot opt source destination   
    Chain OUTPUT (policy ACCEPT)  
    target prot opt source destination   
    Chain POSTROUTING (policy ACCEPT)  
    target prot opt source destination
* Turn on **IP forwarding** on Azure OpenVPN VM and turn on can-ip-forward on GCP OpenVPN VM (can only be done when VM was created)
* Network security group must have port 1194 open to internet (or at least open to the public IP of the other OpenVPN VM)
* Route Table points traffic to the remote VPC/vnet to the local OpenVPN VM

Not able to ping Domain Controller

* Need to log onto Windows 2016 and allow ping