**Creating AG: Using single subnet/VNN Listener**

**DNN vs VNN methods**

There are 2 methods of Listener creation in Azure. DNN and VNN. In VNN the listener will have single static IP address and it will need a load balancer, However if we create listener using DNN, it will not need DNN.

Choice of methods/benefits and limitations:

* VNN method is preferable for legacy application that do not have or accept “multisubnet=true” feature in the connection string. DNN works with “Multisubnet=true” connection string.
* VNN method will need load balancer, DNN doesn’t need load balancer.
* VNN is slower (upto 10 seconds for failover or connection), DNN is faster.
* DNN doesn’t work with Distributed AG. Not sure if DNN supports AG + Replication together.

For reading: [Feature interoperability with availability groups and DNN listener - SQL Server on Azure VMs | Microsoft Learn](https://learn.microsoft.com/en-us/azure/azure-sql/virtual-machines/windows/availability-group-dnn-interoperability?view=azuresql&source=recommendations)

Important Note:

In both methods, we create the AG role in the cluster. The only difference is in the configuration of the listener. In VNN, we can create the listener from the SSMS and we will need a load balancer and a powershell script to configure probe parameter. In DNN, we will run only the powershell script (from Microsoft) that will create a listener and the listener will not have a single IP address. In DNN the IP address is not visible from SSMS and in Cluster manager.

In both cases, the listener has to be added to the AG as a **dependency**.

**The Below steps is for VNN configuration**

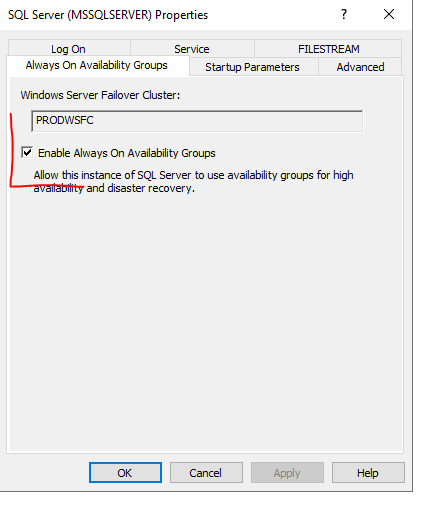
At this step the cluster is created

You have SQL server installed and SSMS.

To create AG: [Tutorial: Configure a SQL Server Always On availability group - SQL Server on Azure VMs | Microsoft Learn](https://learn.microsoft.com/en-us/azure/azure-sql/virtual-machines/windows/availability-group-manually-configure-tutorial-single-subnet?view=azuresql)

**Step 1: enable alwayson feature.**

Go to the SQL server configuration manager property and add alwayson feature



**Step 2: create database and add availability group**

Below is a testAG database created and backup is taken. Backup to a shared folder is needed since the database will be restored to the secondary replica.

A screenshot of a computer

Description automatically generated

1. **Give name to Availability group and click next**

A screenshot of a computer

Description automatically generated

1. **Choose the database that has a backup taken (requirement fulfilled)**

**\_ restore** the database to secondary server **with NORECOVERY.** This will let you **synchronize** later by “**Join**”ing.

A screenshot of a computer

Description automatically generated

1. **Add replica and check automatic failover and synchronous commit mode**

If you encounter a problem:

* Check Alwayson feature is turn on in the secondary server
* Check TCP is enabled in the secondary server

A screenshot of a computer

Description automatically generated

**Listener configuration (Optional)**

**Optionally** you can create a listener here and give the ip address of the frontend ip of Load Balancer created in the previous steps.

You can create a listener at the later stage and skip this step without creating the listener. If you create a listener (a.k.a **Client Access Point**) from the cluster manager , you will need to add dependency under the Availablity group. The Listener name has to be added as a dependent resource in the AG. **But if you create the listener from the SSMS as below, it will be registered as an independent resource automatically.**

A screenshot of a computer

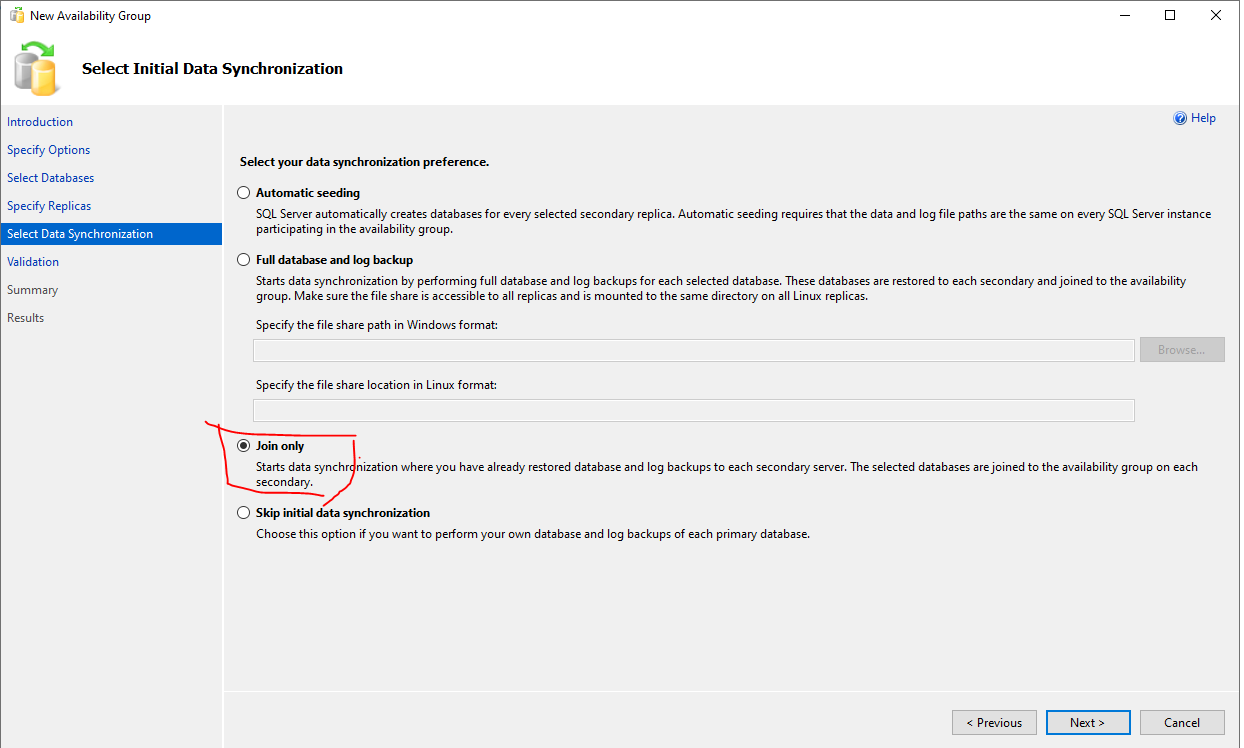
Description automatically generated

A screen shot of a computer

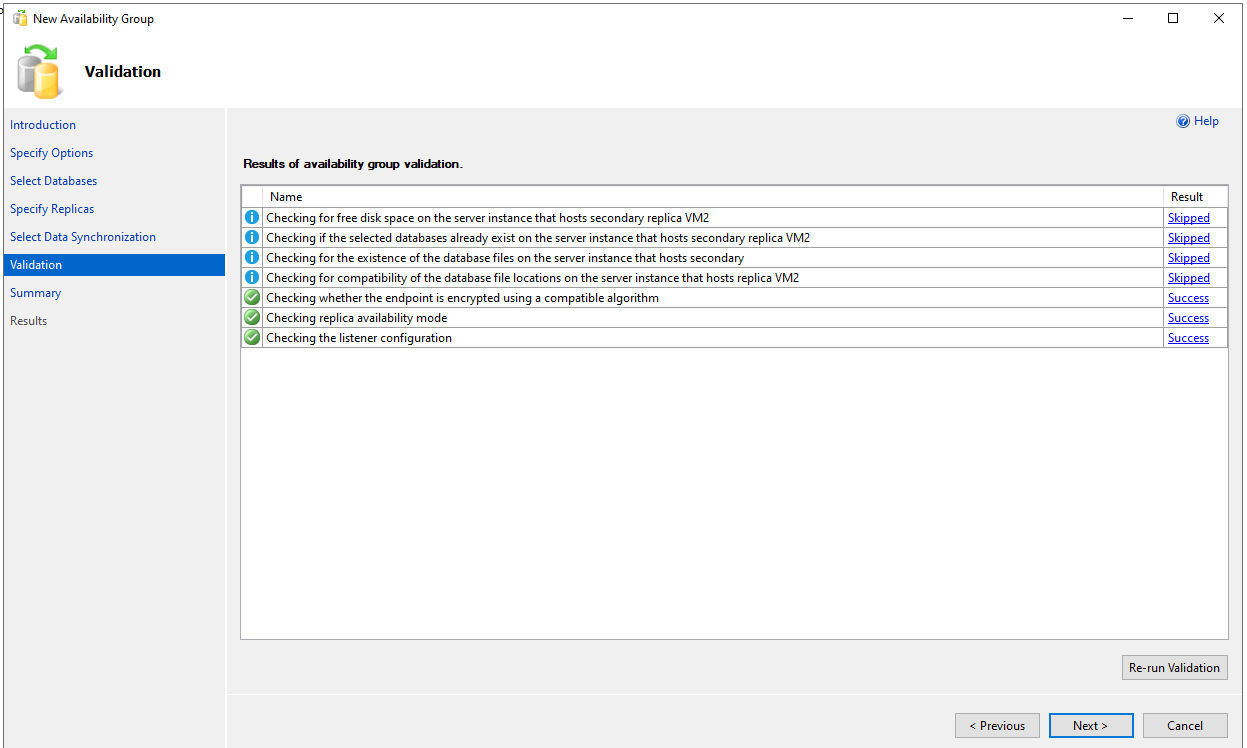
Description automatically generated

1. **Add the database /synchronize**

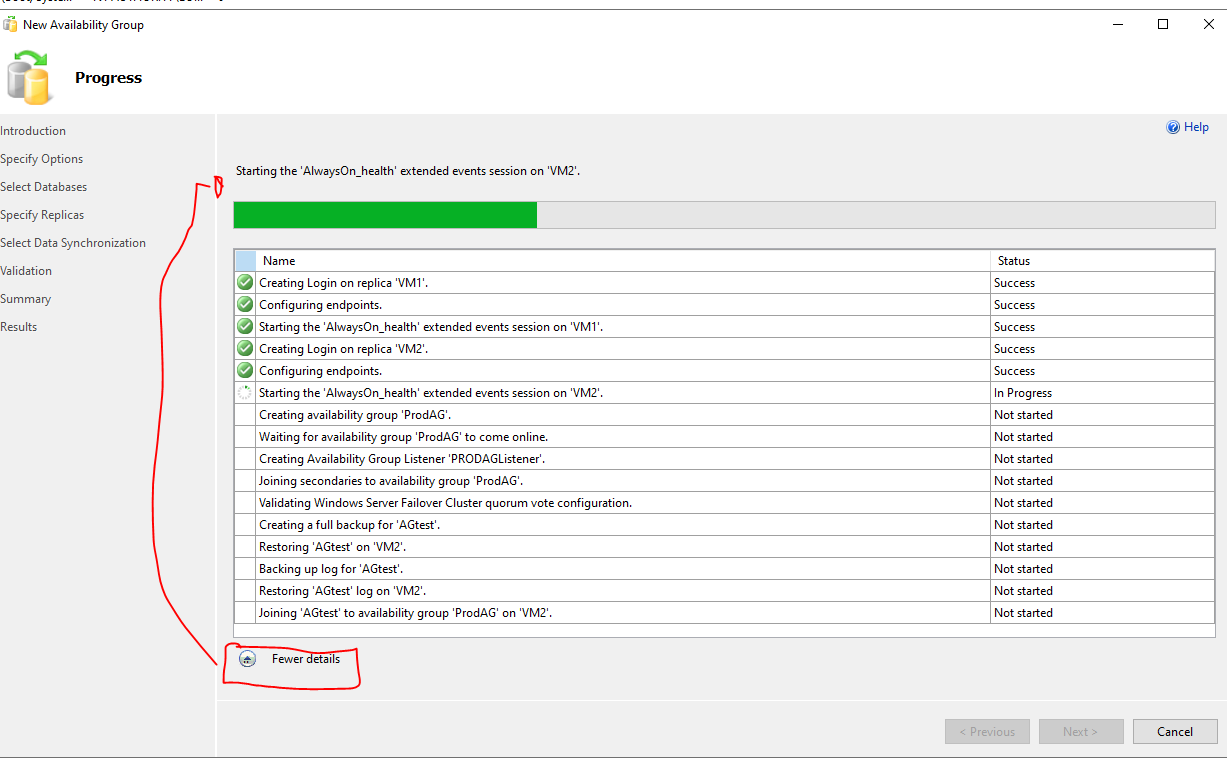
Since I restore the backup with NORECOVERY, I will join here



1. **VALIDATE**

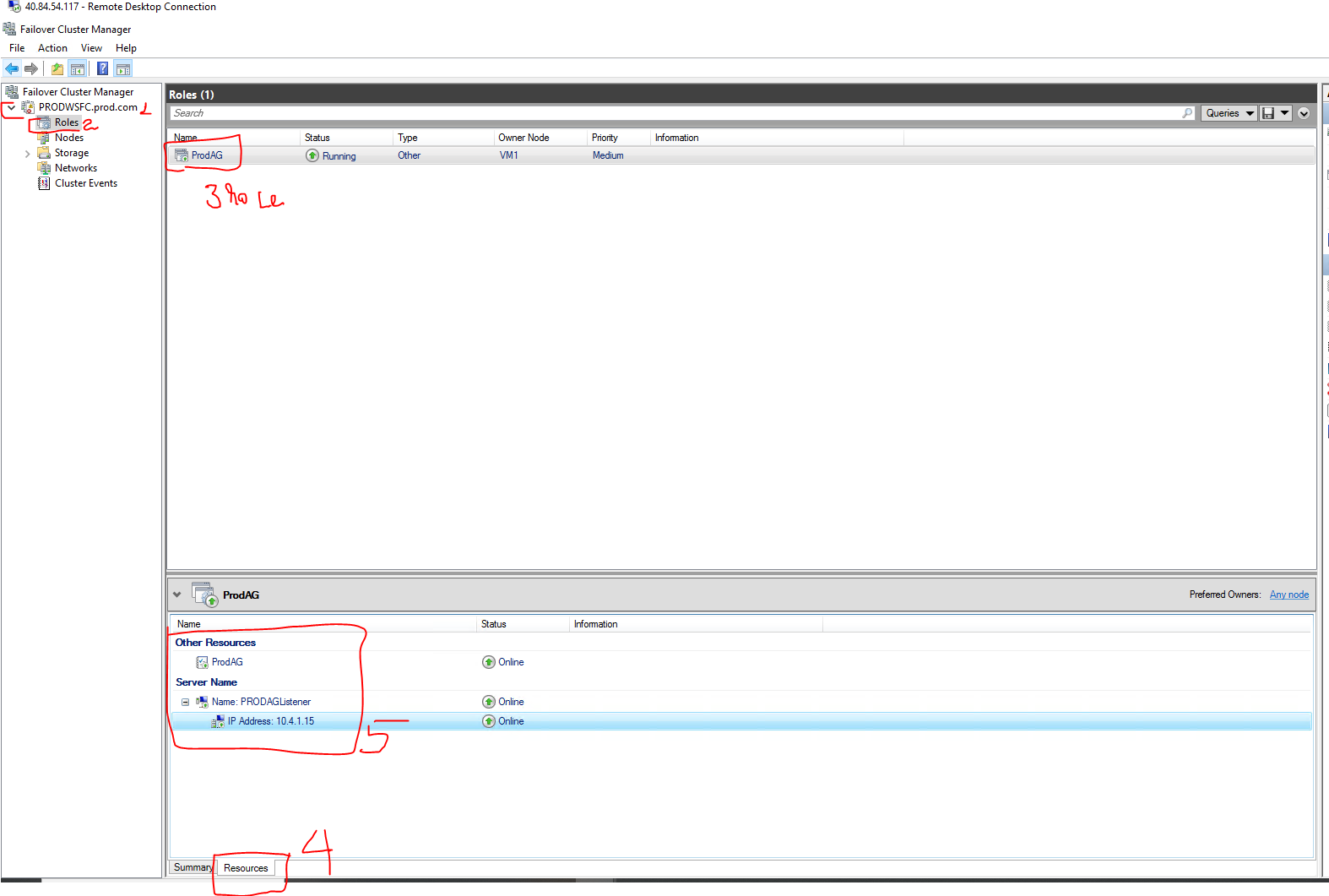


1. **EXPAND AND CHECK DETAILS**



**STEP 3 : CLUSTER VALIDATION – VALIDATE the AG and Listener**

1. Validate a new ROLE is added (created above) and since we added the listener from ssms check it appears with the IP address we provided

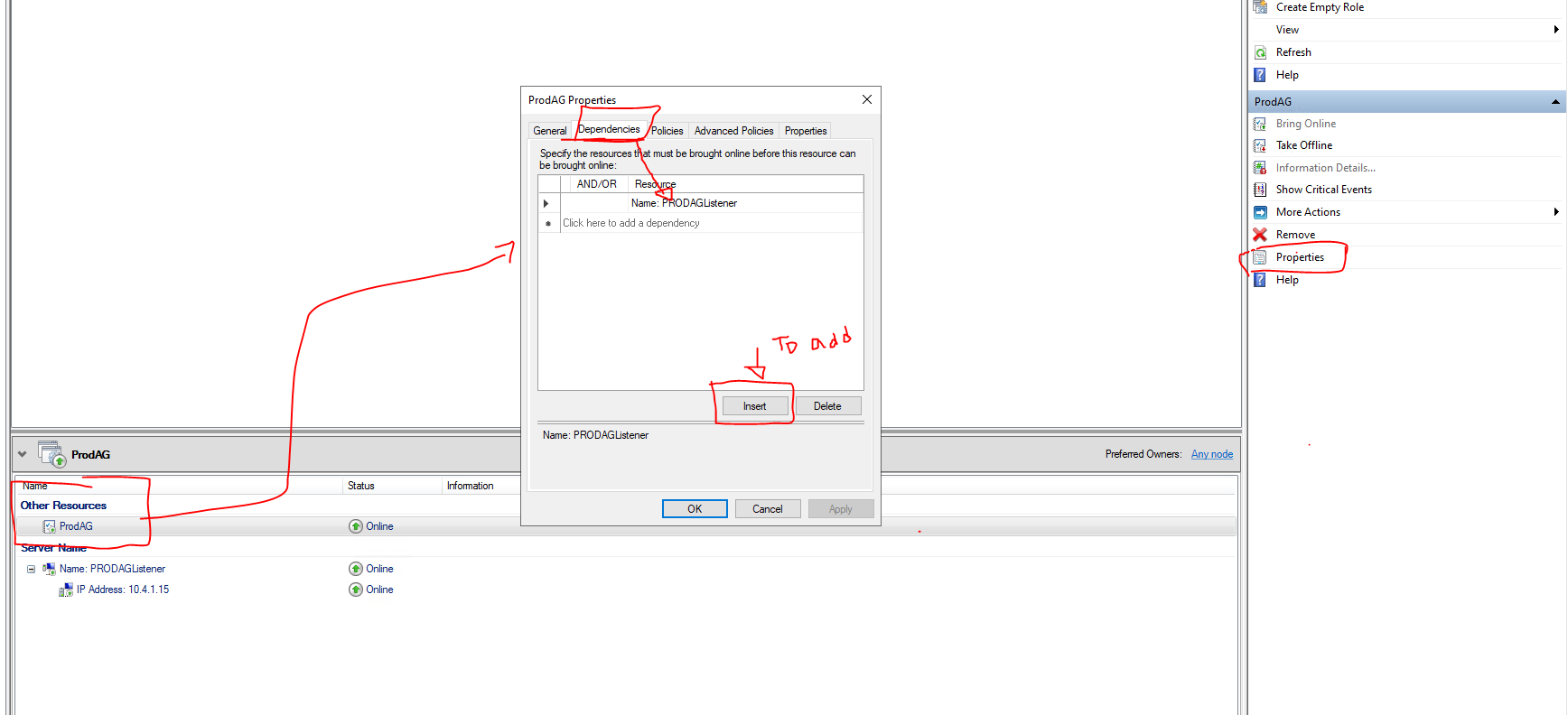


1. **Check if the listener is added as a dependent resource in the AG role**

**Note: since we haven’t run the PowerShell script that confirms a VNN listener, the current listener works from the primary node only. It cannot be used for connection from secondary or other server.**

Right click on the AG – properties – dependencies

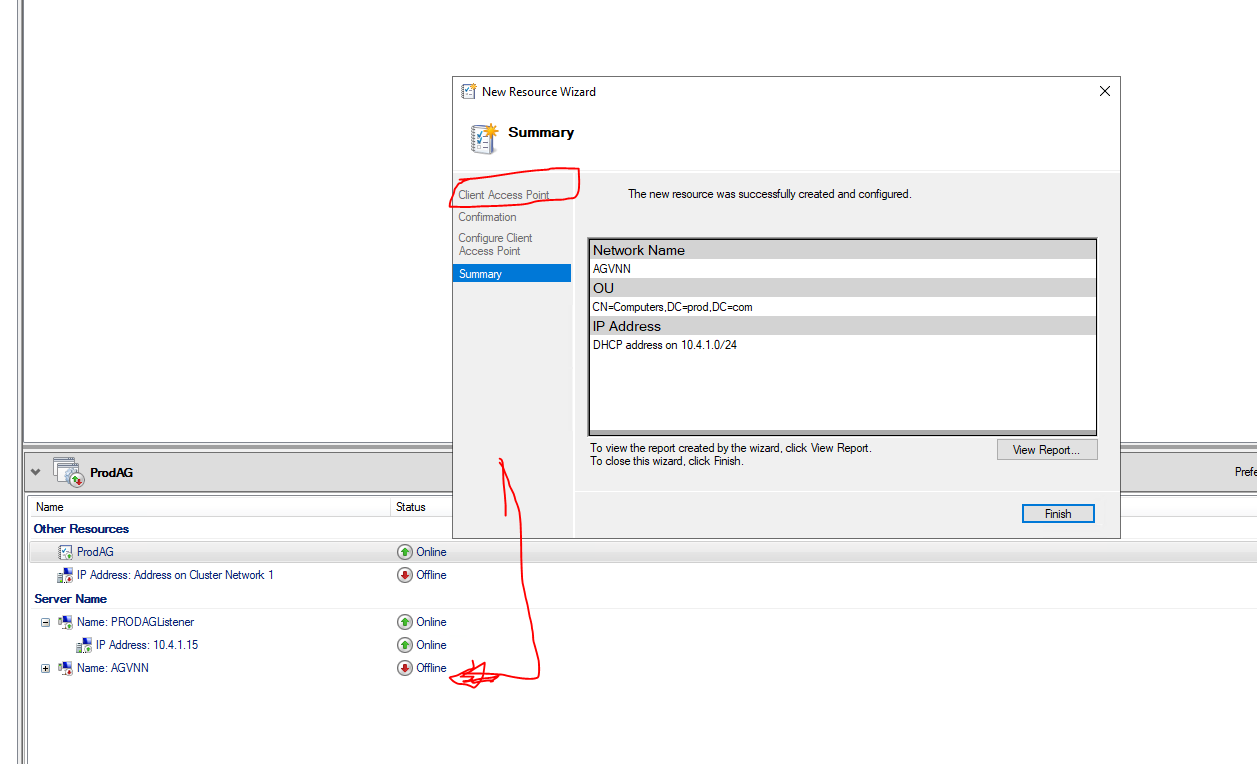
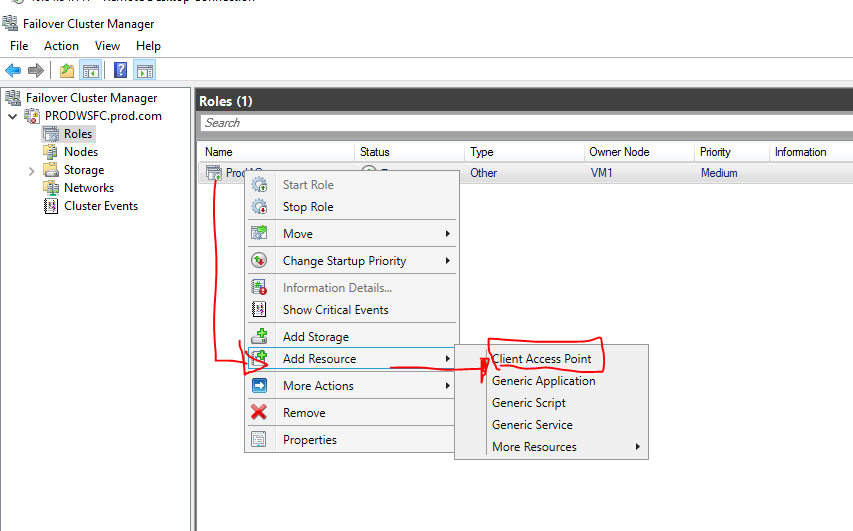
Note: since we created the Listener from SSMS, it is added automatically as a dependency. However, if it is created from the cluster under AG as a new resource (CLIENT ACCESS POINT), then we have to click “insert” and add the LISTENER (CCLIENT ACCESS POINT) NAME.



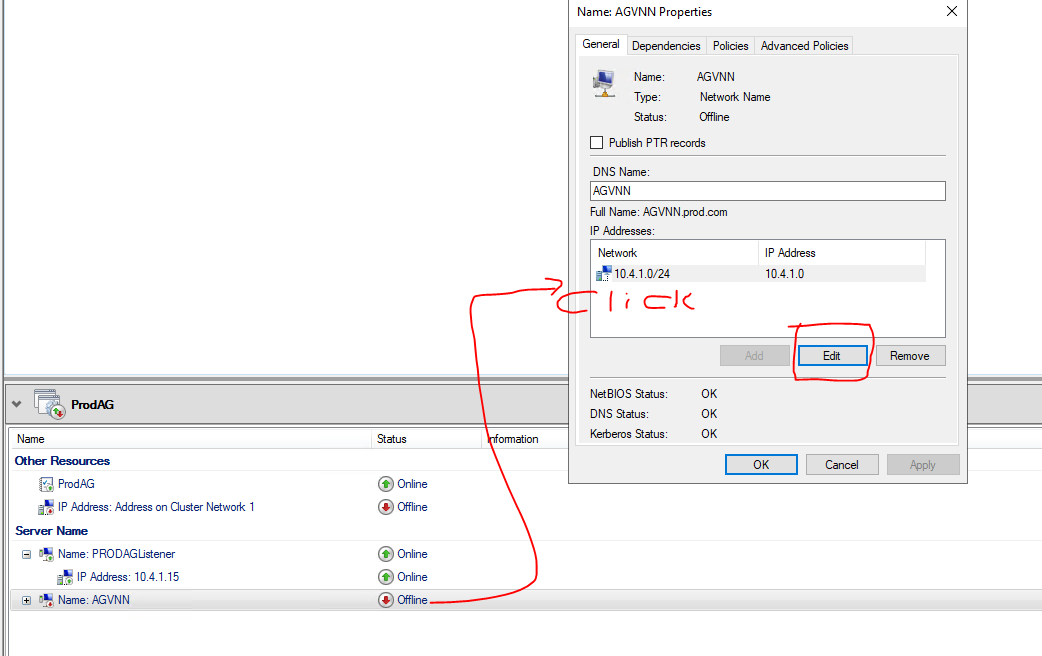
If you want to add the listener as a client access point right click on the AG role and click “Add resource” then “Client Access Point”

**Create Client Access Point (CAP) : skip this and go to STEP 4 if listener is already created in the previous step.**

CAP or the new listener (e.g. below AGVNN) will have automatically assigned an IP address which will be the IP address of the primary node. Since the new CAP will be created with the IP of the primary node, there will be an IP conflict, hence the CAP will not be online until a new IP (the IP of the Load Balancer’s front IP) is supplied. To do that right click on the listener (AGVNN) and



To change the IP address and provide a static IP address: click on the ip address and click “EDIT” and put a static IP.



A screenshot of a computer

Description automatically generated

**Step 4: use PowerShell to create the VNN validation**

As noted above, the listener works only on the primary. To resolve this and use a powershell script provided by Microsoft to create the Load Balancer

Resource: [Configure an Azure load balancer for an AG VNN listener - SQL Server on Azure VMs | Microsoft Learn](https://learn.microsoft.com/en-us/azure/azure-sql/virtual-machines/windows/availability-group-vnn-azure-load-balancer-configure?view=azuresql&tabs=ilb)

**PowerShell script: run** on the primary node (to get the details on what to fill, look at the pictures below)

Make sure there are no improper spaces e.g.

" ProdAG\_192.168.2.15" this ***works*** but

" ProdAG\_192.168.2.15" this **fails** because of the ***space*** at the beginning

------------------------

$ClusterNetworkName = "Cluster Network 1"

$IPResourceName = "ProdAG\_10.4.1.15"

$ILBIP = "10.4.1.15"

[int]$ProbePort = 59999

Import-Module FailoverClusters

Get-ClusterResource $IPResourceName | Set-ClusterParameter -Multiple @{"Address"="$ILBIP";"ProbePort"=$ProbePort;"SubnetMask"="255.255.255.255";"Network"="$ClusterNetworkName";"EnableDhcp"=0}

-----------------

To get parameter values:

$ClusterNetworkName – right click on the “***networks***” under cluster manager. It is mostly “***Cluster Network 1***” unless new NIC is added??A screenshot of a computer

Description automatically generated

$IPResourceName: under the ***listener*** properties – copy the ***name*** ENTIRELY e.g. PRODAG\_10.4.1.15 or whatevername+ip

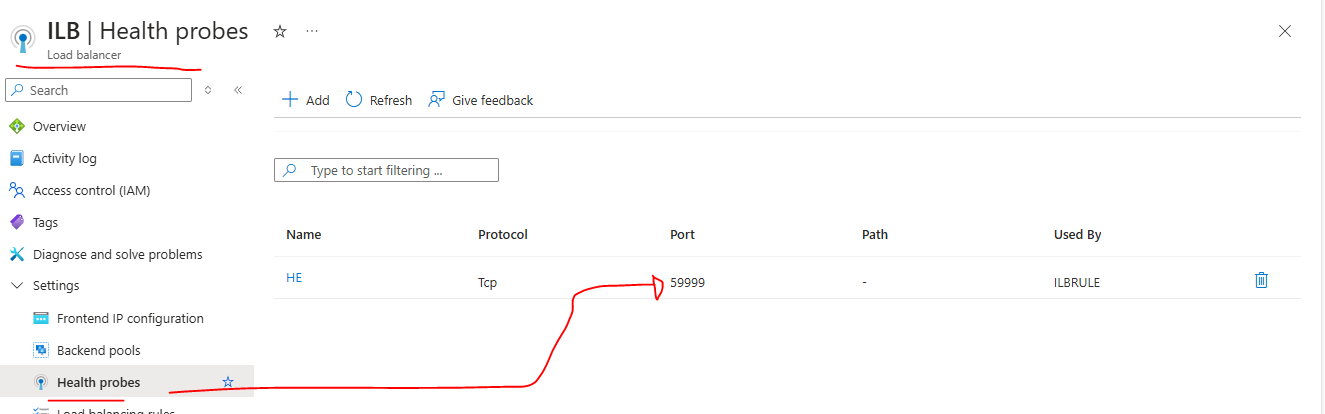
A screenshot of a computer

Description automatically generated

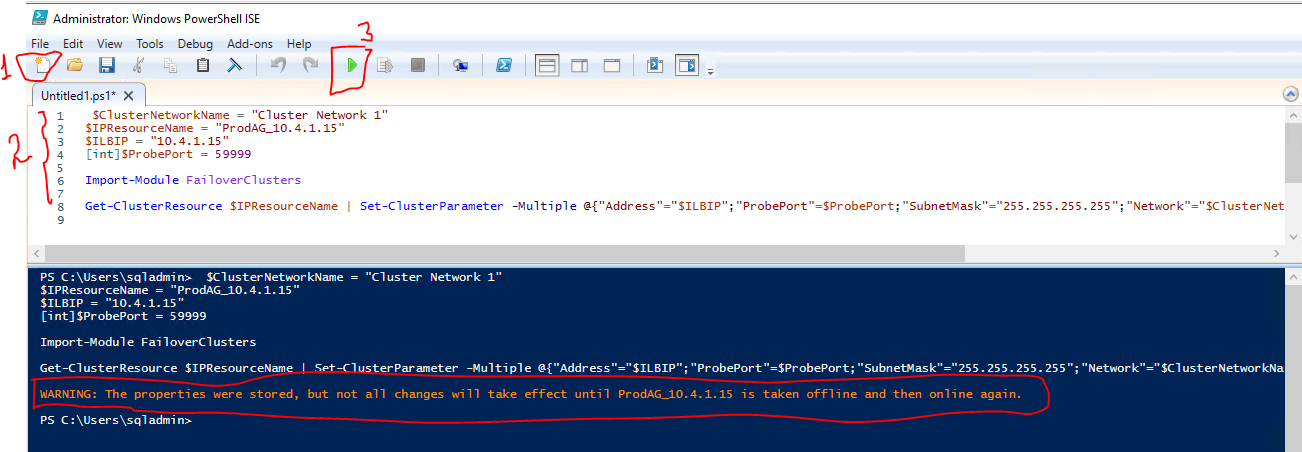
$ILBIP = "10.4.1.15"

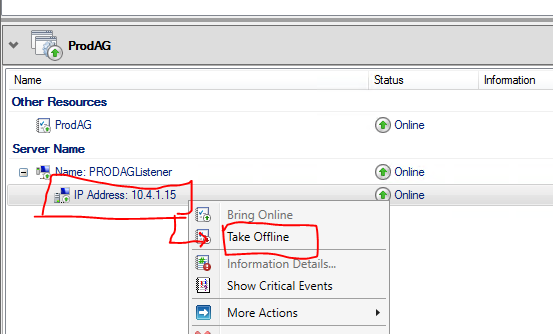
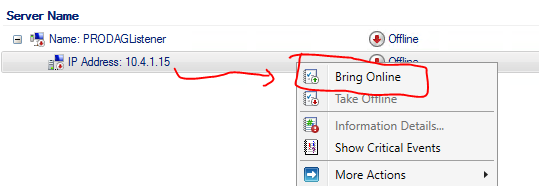
$ILBIP : the listener or load balancer’s frontip e.g. "10.4.1.15"

$ProbePort: the ***health probe TCP port*** we configured under ***load balancer*** in the Azure portal i.e. 59999



**Warning: after you run the PowerShell command, take the listener (PRODAGLISTENER) “offline” then online and make sure everything is online.**



Bring other resources to ONLINE if they are showing OFFLINE.

COMPLETED. TRY to connect using the listener from the nodes and a different node. Run select @@servername to check what the primary is after failover.

Do failover for test

A screenshot of a computer

Description automatically generated