Infoblox

Deployment Guide

Deploy vNIOS in Azure Using ARM Templates

Table of Contents

Introduction

Infoblox vNIOS for Azure is a virtual appliance designed for deployment as a Virtual Machine (VM) in Microsoft Azure. Infoblox vNIOS for Azure enables you to deploy robust, manageable, and cost effective Infoblox appliances in the Microsoft Cloud.

Infoblox NIOS is the underlying software running on Infoblox appliances and provides core network services and a framework for integrating all the components of the modular Infoblox solution. It provides integrated, secure, and easy-to-manage DNS (Domain Name System), IPAM (IP address management) and other services.

Infoblox vNIOS for Azure appliances can be joined to an existing on-premise or hybrid/multi cloud grid, or the entire grid can run in Azure. The vNIOS appliance can be configured as a primary DNS server for your Azure networks. You can also use Infoblox Cloud Network Automation with vNIOS for Azure to improve visibility of cloud resources and increase the flexibility of your cloud environment.

Using Azure Resource Manager (ARM) templates allows you to deploy vNIOS in Azure in configurations not supported via deployment from the Azure Portal. For example, deploying multiple vNIOS VMs in one Resource Group and into Availability Sets. ARM templates also allow you to automate deployments, creating Infrastructure as Code.

Prerequisites

The following are prerequisites for deploying and managing an Infoblox vNIOS for Azure appliance:

- Valid subscription for Microsoft Azure.
- Permissions to create Resource Groups, Virtual Networks, Virtual Machines, and App Registrations in your Azure subscription.
- Additionally, to deploy vNIOS using ARM templates, you may need administrator permissions on your local workstation in order to install required software.
- Understanding of basic networking concepts and tools, including public and private IP addressing, DNS, Secure Shell (SSH), and command line/terminal applications.

Workflow

The following outline lays out the basic steps to deploy and configure Infoblox vNIOS using ARM templates, in a new Azure subscription:

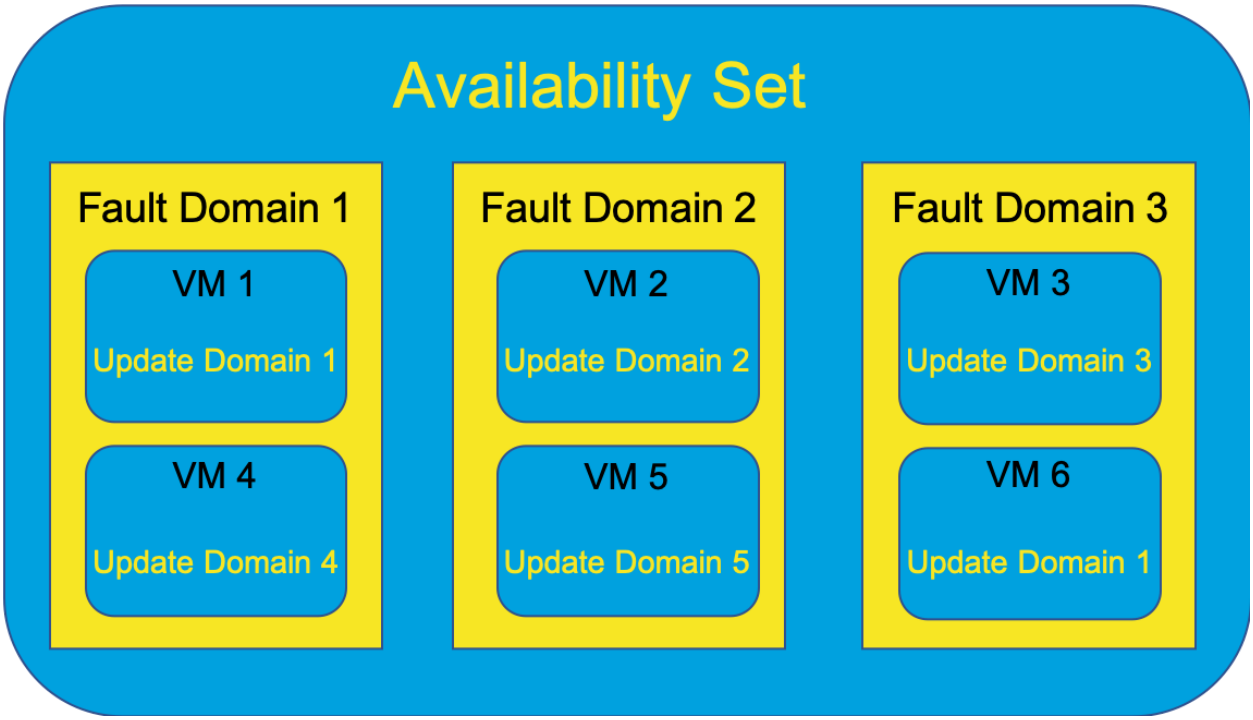
1. Deploy an Azure Virtual Network (VNET)
2. Deploy vNIOS using the Azure Portal / Create Base ARM Template
3. Customize Template and Parameter Files
4. Install the Azure Command Line Interface (CLI)
5. Deploy vNIOS using ARM Template and Azure CLI
6. Configure the vNIOS Appliances
7. Configure Azure VNET DNS
8. Perform vDiscovery for Azure

High Availability in Azure: Availability Sets

The Infoblox High Availability (HA) feature provides redundancy and fault tolerance in an easy to manage and implement configuration, ideally suited for local/on-premise networks (both physical and virtual). However, this feature is not supported in vNIOS for Azure deployments.

To implement highly available DNS and IPAM services with Infoblox servers in Azure, the appliances can be deployed as part of an Availability Set.

Azure datacenters classify groups of hardware under two domains, update domains and fault domains. Update domains are groupings of hardware that may be patched or rebooted at the same time during maintenance periods. Fault domains describe hardware sharing a common power source and network switch. When VMs are deployed as part of an availability set, they are separated across 5 update domains and up to 3 fault domains by default.



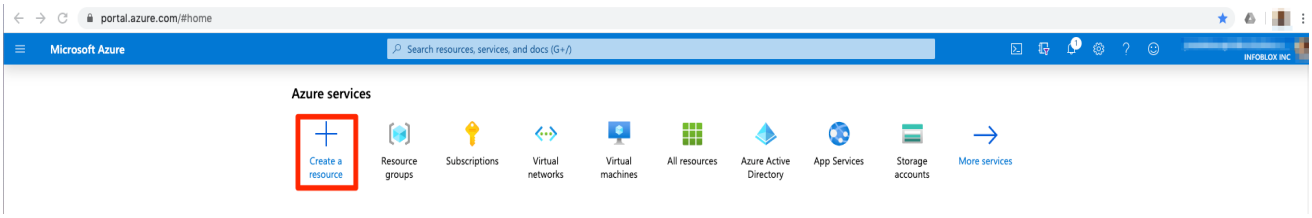
When you deploy at least two vNIOS for Azure VMs in an availability set, you ensure that at least one VM will remain available during planned or unplanned maintenance events, meeting the 99.5% Azure SLA.

Deploy Azure Virtual Network

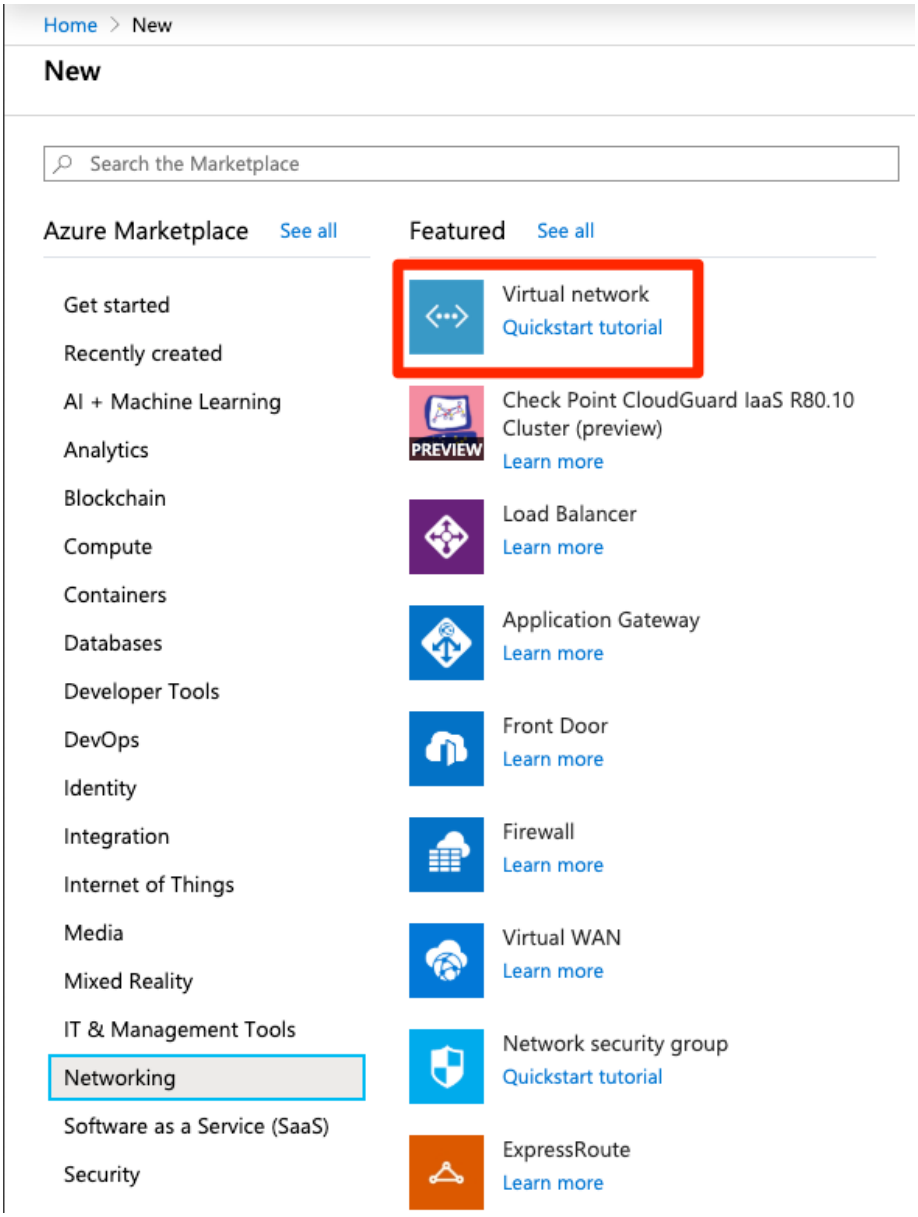
Infoblox vNIOS for Azure appliances need to connect to a VNET with at least 2 separate subnets. It is possible to create a new VNET when deploying vNIOS for Azure appliances, however creating VNETs first allows for greater control and customization. This also allows the ARM templates you will create to work in an Azure environment with existing VNETs.

To deploy a VNET in Azure:

1. Login to the Azure Portal at https://portal.azure.com. Click on **Create a resource**.

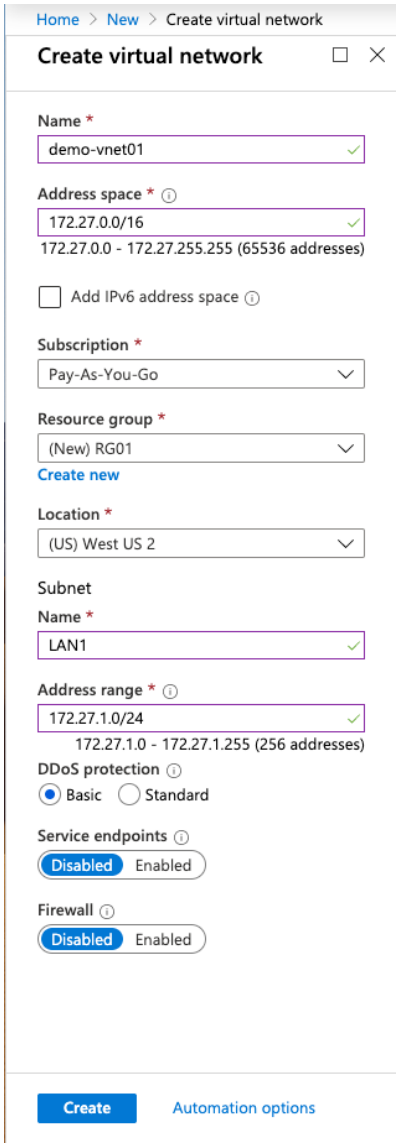


2. In the Azure Marketplace, select **Networking** and click on **Virtual network**.

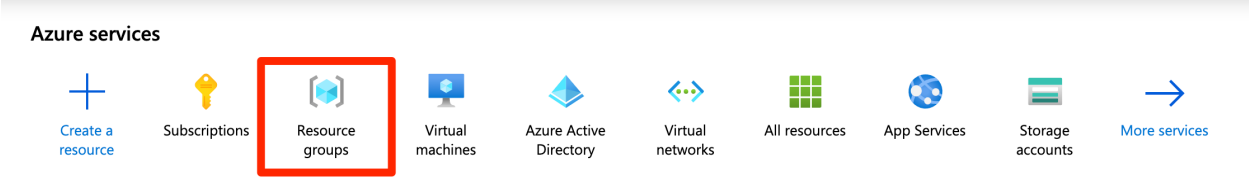


3. Give the network a name.
4. Assign an IP Address space, for example 172.27.0.0/16.
5. Under Resource Group, click **Create New**. Give the resource group a name and click ok.

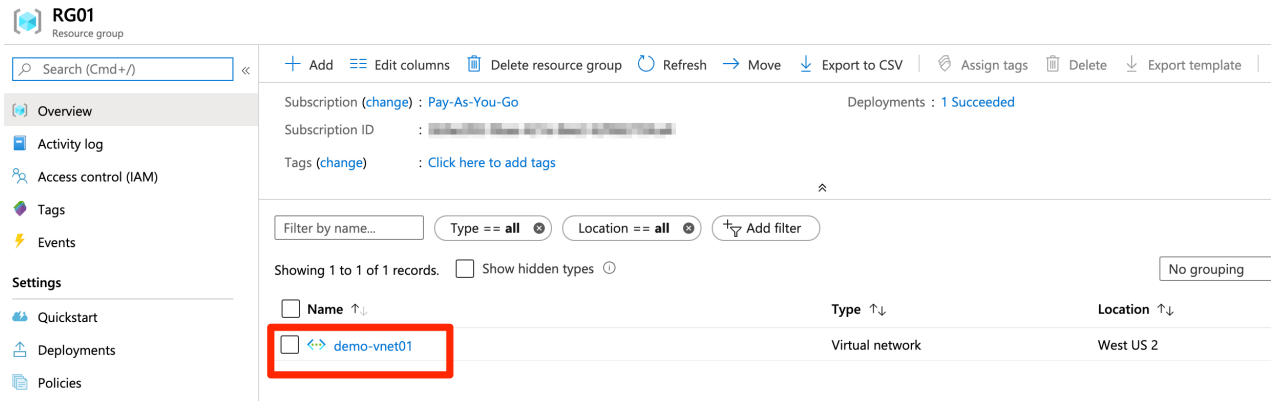
6. Under Subnet, give the subnet a descriptive name and assign an IP Address range that falls inside the VNET, for example 172.27.1.0/24.
7. Click **Create**.



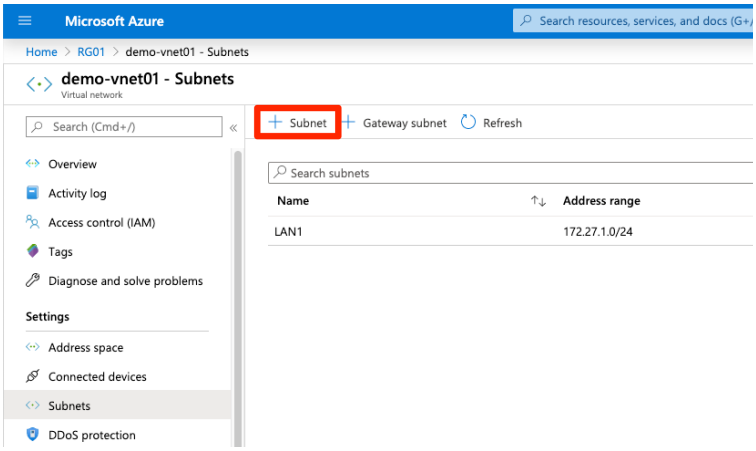
8. From the dashboard, click on **Resource groups**



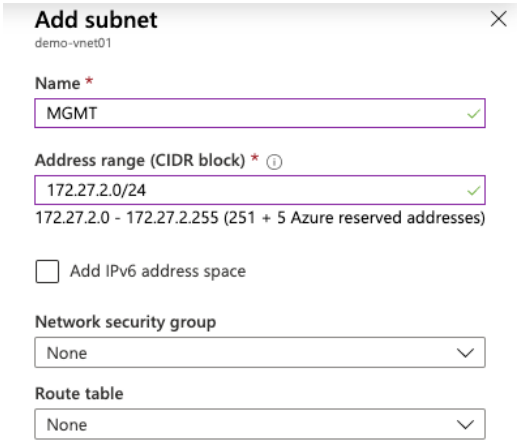
9. Select your resource group from the list.
10. Select your VNET from the resource group overview page.



11. Click on **Subnets**. Click on **+ Subnet** to create a new subnet.



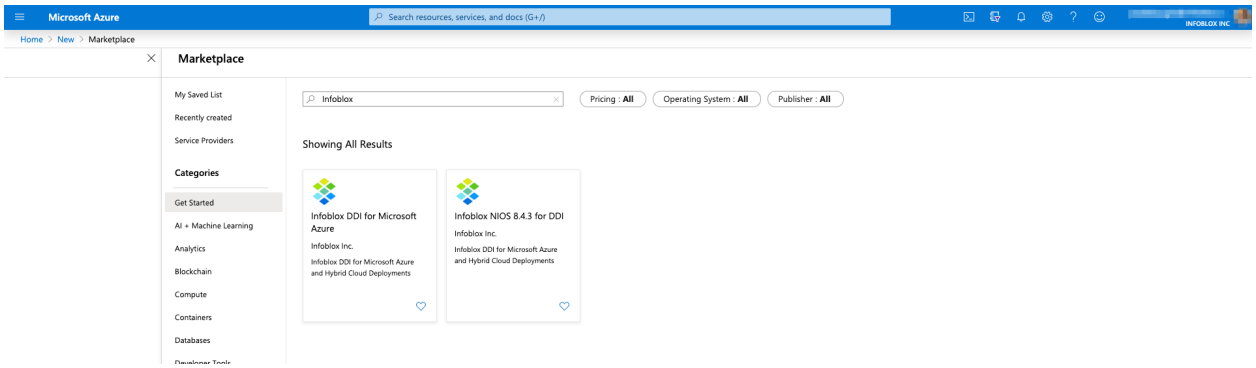
12. Give the new subnet a descriptive name.
13. Assign an IP Address range that falls inside the VNET and does not conflict with the other subnet.
14. Click **OK**.



Deploy vNIOS Using Azure Portal

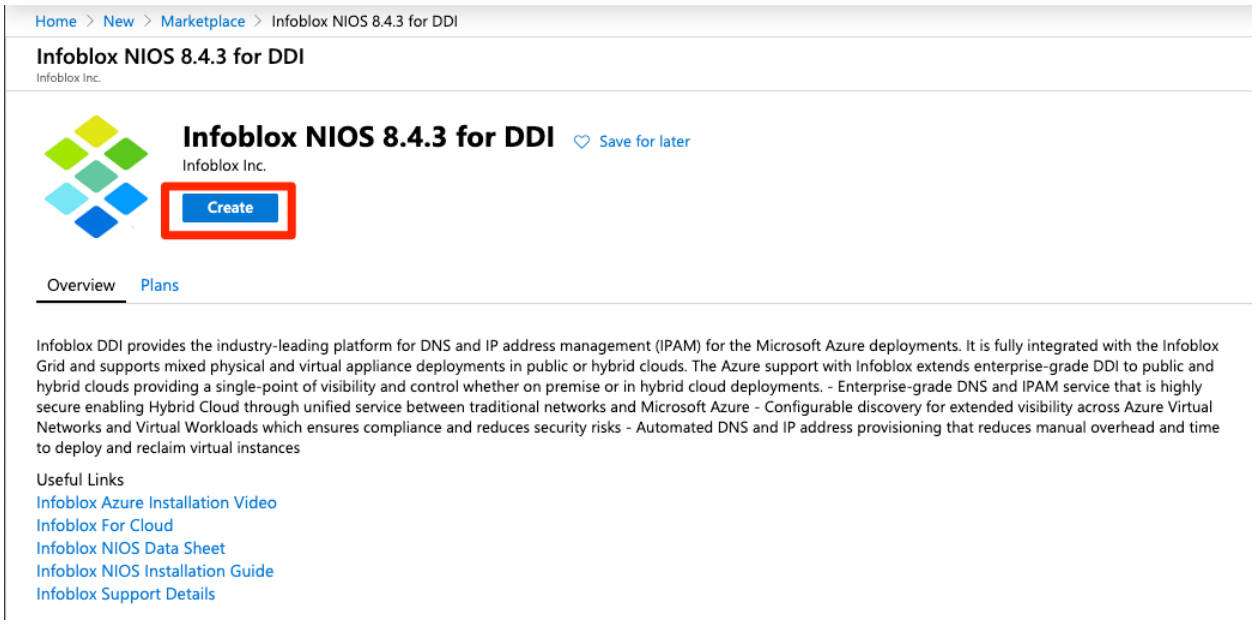
The intent of this guide is to deploy vNIOS in Azure using ARM templates to allow for customization not available from portal deployment. However, the simplest and most reliable way to create ARM templates is to start in the Azure Portal. The following instructions can be used to deploy vNIOS using the Azure Portal or to create a base ARM template that can be modified for further customization.

1. In the Azure Portal, click on **Create a resource**.
2. In the Azure Marketplace search box, type **Infoblox** and press **Enter**.
3. Select the latest Infoblox vNIOS for Azure offering.



Note: Offerings can change often as new vNIOS versions are released. Versions currently available may vary from those displayed here.

4. Review the Overview page and click **Create**.



5. On the Basics tab, select a vNIOS model from the dropdown. Both Trinzic (TE) and Cloud Platform (CP) models are available. This guide will use the TE models.
6. Name the NIOS VM. Enter a password for the admin user.

Note: The password must be between 6 and 72 characters long, and contain characters from at least 3 of the following groups: uppercase letters, lowercase letters, numbers, and special characters. Additionally, Azure does not allow some specific passwords. The list can be found here: https://docs.microsoft.com/en-us/azure/virtual-machines/windows/faq#what-are-the-password-requirements-when-creating-a-vm.

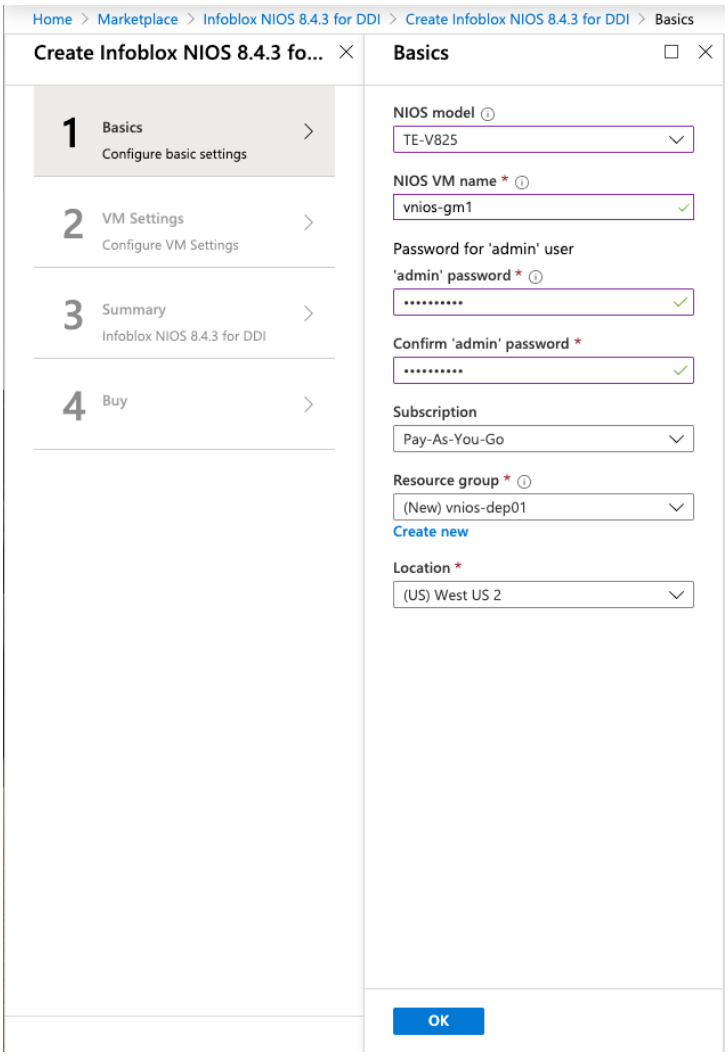
7. Select the desired Subscription from the dropdown if you have more than one.
8. Under Resource group, click **Create new**. Name the resource group and click **OK**.

Note: When setting up vNIOS deployment through the Azure Portal, a new or empty resource group is required.

9. Select a Location from the dropdown.

Note: This location must be the same as the location of the VNET which will be used.

10. Click **OK** to advance to the next tab.

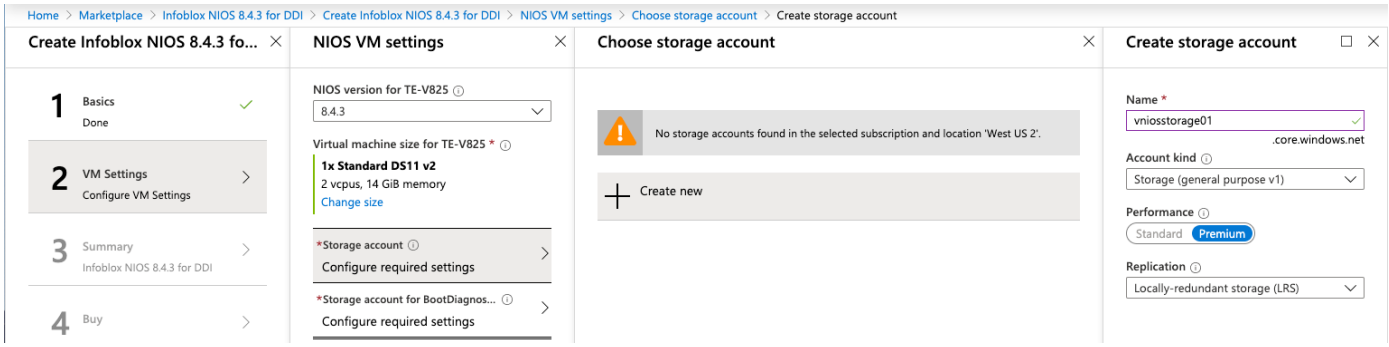


11. On the VM Settings tab, click on **Storage account**. Click **Create new**.

12. Enter a name for the storage account.

Note: Azure requires that the storage account name must be globally unique.

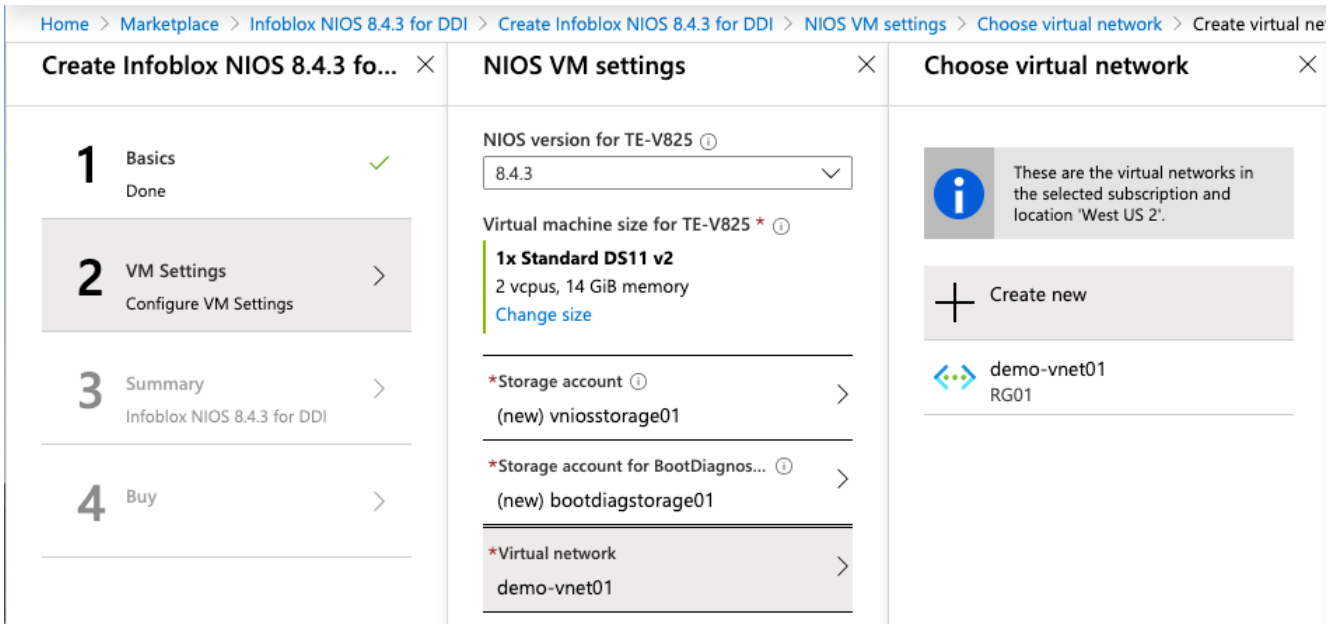
13. Ensure Premium is selected under Performance. Click **OK**.



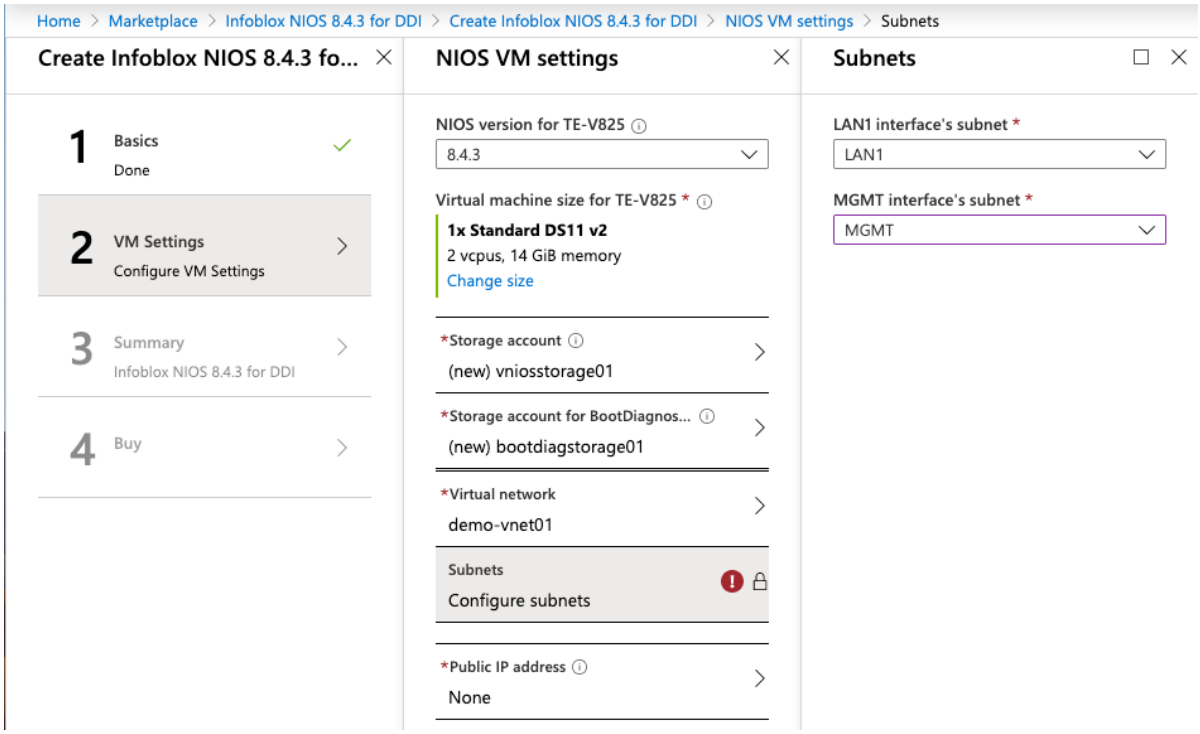
14. Click **Storage account for BootDiagnostics**. Click **Create new**.
15. Enter a name for the storage account.

Note: Azure requires that the storage account name must be globally unique.

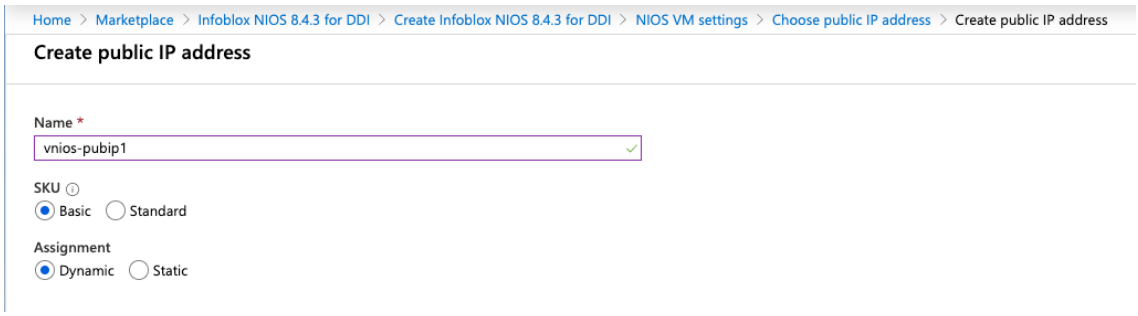
16. Ensure Standard is selected under Performance. Click **OK**.
17. Click **Virtual network**. Select your VNET from the list.



18. Click **Subnets**. Under LAN1, select the first subnet from the dropdown. Under MGMT, select the second subnet from the dropdown.



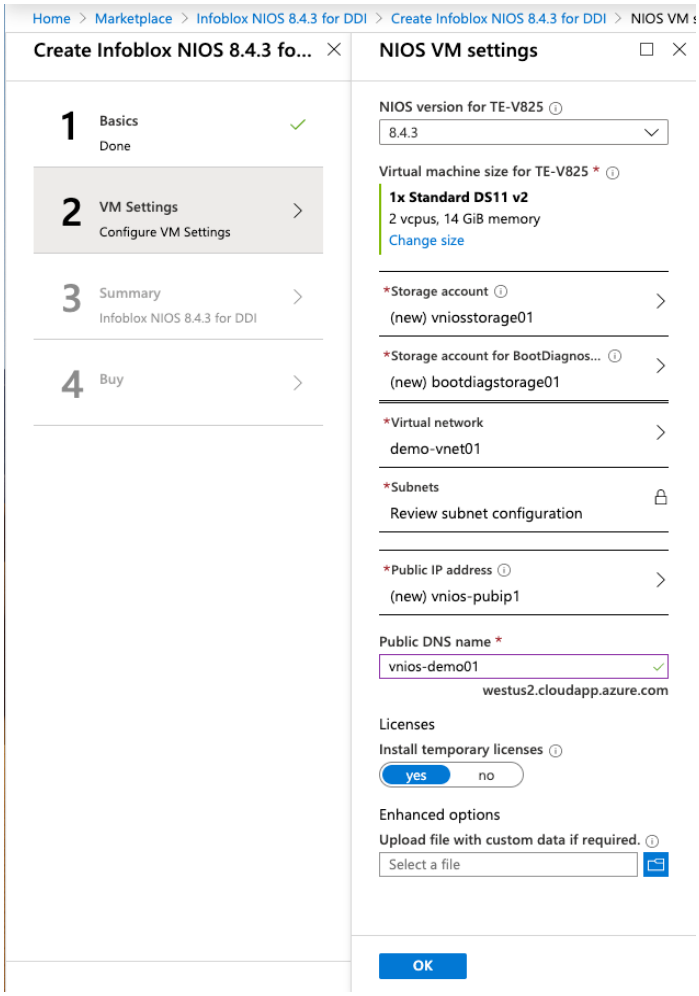
19. Click **Public IP address**. Click **Create new**.
20. Enter a name for the public IP and click **OK**.



21. Enter a Public DNS name for the vNIOS VM.

Note: The public DNS name must be globally unique.

22. Under Licenses, select **yes** to install temporary licenses.
23. Click **OK** to advance to the next tab.

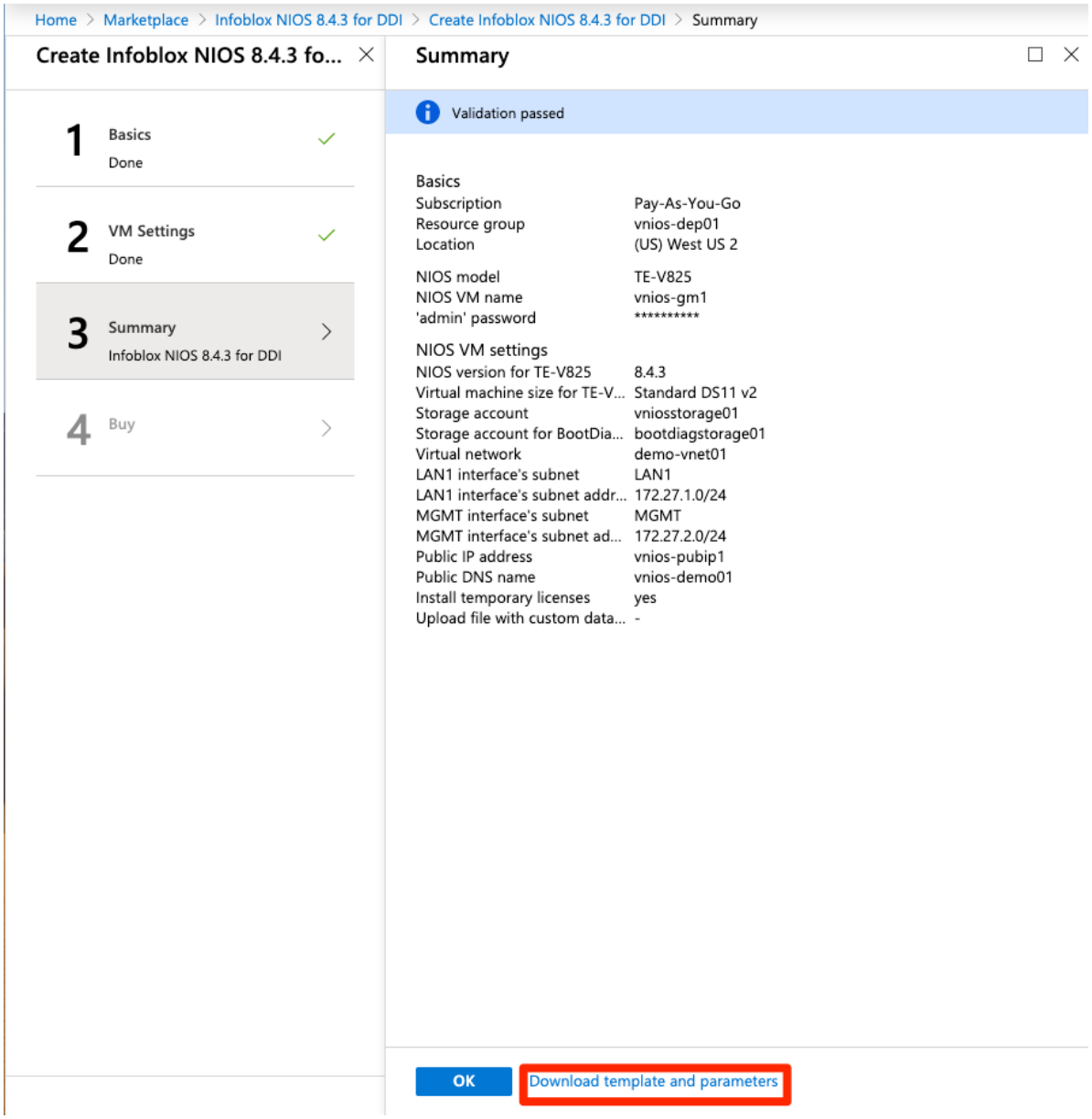


24. On the Summary tab, wait for validation to complete.
25. Review the settings for the vNIOS VM.

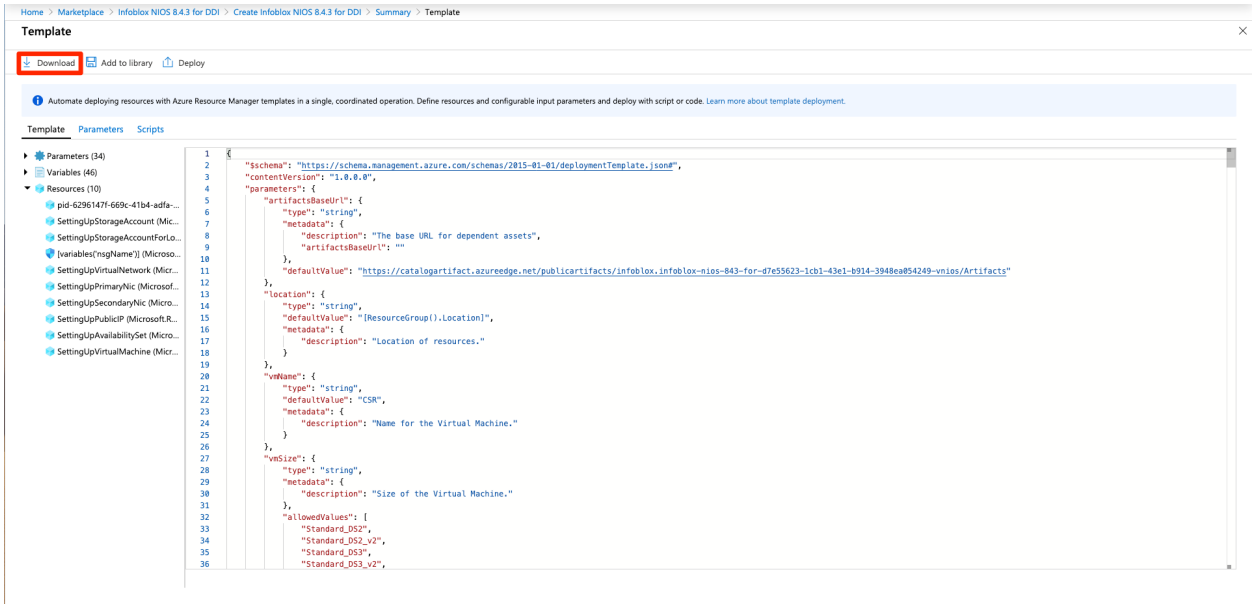
Note: To create the VM from the Azure Portal, click OK and then click Create on the Buy tab. To continue with ARM template deployment, instead proceed to the next section.

Create Base ARM Template

1. On the Summary tab, click **Download template and parameters**.



2. On the Template page, click **Download.**



3. Extract the contents of template.zip. To extract using the command line unzip program, enter the following command, inserting the path to the directory you want to use:

   unzip template.zip -d path/to/directory

Modify Template and Parameter Files

After downloading and extracting the base ARM templates, they can be customized in many ways to allow for varying deployment scenarios. In this guide, we will create and customize two parameter files to deploy two vNIOS appliances in an Azure Availability Set.

Open ARM Template Files

ARM Templates can be opened in any standard text editor as well as many Integrated Development Environments (IDE). Microsoft Visual Studio Code has extensions created specifically for working with ARM templates and will be used for this guide. VS Code is a free tool and can be installed from this site: https://code.visualstudio.com/.

Add Extension to VS Code

The Azure Resource Manager Tools extension for VS Code provides syntax, validation, and other tools for working with ARM templates.

1. To install the extension, open VS Code.
2. On the Activity Bar, click the **Extensions** Icon.
3. Type **ARM** in the Extensions search box.
4. Select **Azure Resource Manager (ARM) Tools** from the list.
5. Click **Install**.

ARM Template Structure

There are two basic files used with ARM templates, template files and parameter files. Both files are written in Javascript Object Notation (JSON).

Template files can contain a number of required and optional elements:

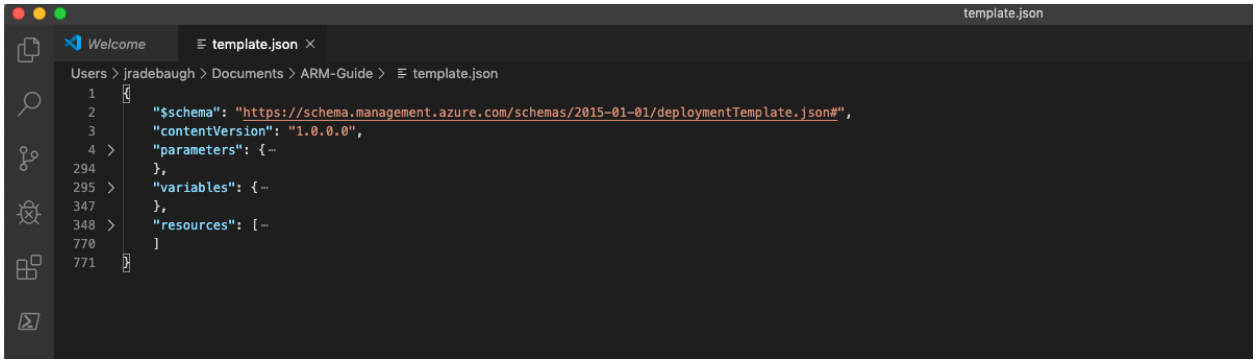
- Schema - Required - This describes the location of the JSON schema file which describes the version of the template language.
- Content Version - Required - This describes the version of the template. This can be any value the template author designates.
- Parameters - Optional - This section contains references to values that will be provided during deployment. This section often contains default values and lists of acceptable values for certain parameters.

```
42          "niosModel": {
43              "type": "string",
44              "defaultValue": "cp-v1405",
45              "metadata": {
46                  "description": "niosModel."
47              },
48              "allowedValues": [
49                  "IB-V825",
50                  "IB-V1425",
51                  "IB-V2225",
52                  "cp-v805",
53                  "cp-v1405",
54                  "cp-v2205"
55              ]
56          },
57          "niosVersion": {
58              "type": "string",
59              "defaultValue": "latest",
60              "metadata": {
61                  "description": "niosVersion"
62              },
63              "allowedValues": [
64                  "latest",
65                  "843.383835.0"
66              ]
67          },
```

This screenshot shows a sample of the parameters listed in the template referenced throughout this guide. The **niosModel** and **niosVersion** parameters each have default values and a list of accepted values.

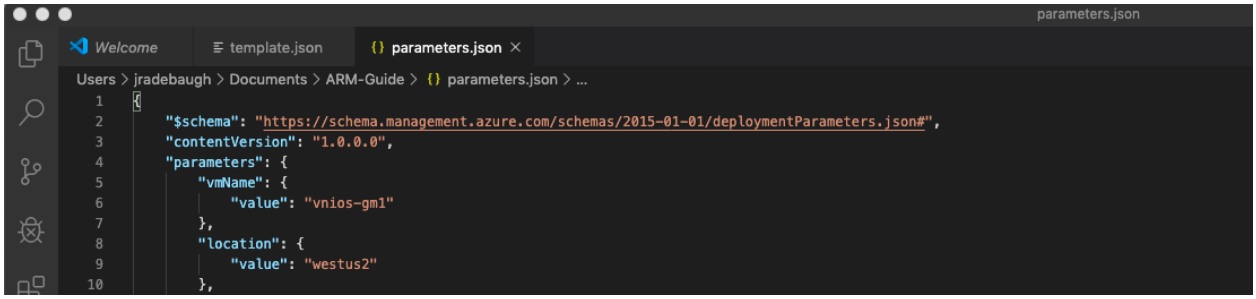
Note: These allowable values can vary and change with different NIOS releases. Refer to the latest offering in the Azure Marketplace for current values.

- Variables - Optional - This section contains values constructed for use in other elements of the template.
- Functions - Optional - This section is used to define complex expressions for use in other elements of the template.
- Resources - Optional - This section defines the resources that will be created or updated.
- Outputs - Optional - This section specifies values that will be returned when deployment is complete.



This screenshot shows the template referenced throughout this guide, with sections collapsed. This template uses the Schema, Content Version, Parameters, Variables, and Resources elements.

The second file type, the parameter file, contains three elements, Schema, Content Version, and Parameters. This file is used to pass parameter values into the template during deployment. The bulk of this file consists of parameter names and values which correspond to the parameters listed in the template file.



This screenshot shows the first few lines of a parameter file used in this guide. The first name/value pair is vmName/vnios-gm1.

Additional general information on Azure Resource Manager template structure and usage can be found on Microsoft's site: https://docs.microsoft.com/en-us/azure/azure-resource-manager/templates/template-syntax

Infoblox Parameters

ARM templates allow a great deal of customization used to deploy in specific and unique environments. This customization is specified through the parameters passed into the template during deployment and used to set values for resource attributes. Infoblox vNIOS appliances require some specific parameters for their configuration, such as the NIOS model and NIOS version. The following table provides explanations for each of these parameters and allowable values where required.

Parameter	Description	Allowed values
baseUrl	Base URL for dependent assets	
location	Location of resources.	
vmName	Name for the Virtual Machine.	
vmSize	Size of the Virtual Machine. (corresponds to vNIOS model virtual hardware requirements)	"Standard_DS2","Standard_DS2_v2", "Standard_DS3", "Standard_DS3_v2","Standard_DS11_v2", "Standard_DS12_v2", "Standard_DS13_v2"
niosModel	vNIOS appliance model.	"IB-V825","IB-V1425", "IB-V2225","cp-v805","cp-v1405", "cp-v2205"
niosVersion	Version of NIOS software to use.	"latest", "843.383835.0"
adminPassword	Password for the command line and web interfaces.	
virtualNetworkName	VNET name	
virtualNetworkExistingRGName	Resource Group containing existing network	
virtualNetworkAddressPrefix	Virtual Network Address prefix, using CIDR block notation	
vnetNewOrExisting	Identifies whether to use new or existing Virtual Network	
subnet1Name	Subnet 1 Name	
subnet1Prefix	Subnet 1 Prefix, using CIDR block notation	
subnet1StartAddress	Subnet 1 Starting IP Address	*Must be an unused IP address in the subnet*
subnet2Name	Subnet 2 Name	
subnet2Prefix	Subnet 2 Prefix, using CIDR block notation	
subnet2StartAddress	Subnet 2 Starting IP Address	*Must be an unused IP address in the subnet*
newStorageAccountName	Unique Name for Storage Account where the Virtual Machine's disks will be placed.	
storageAccountType	The type of storage account created.	“Premium_LRS”

storageAccountNewOrExisting	Identifies whether to use new or existing Storage Account	"new", "existing"
storageAccountExistingRG	Resource Group containing existing storage account	
newStorageAccountForLogsName	Unique Name for Storage Account where the Virtual Machine's boot diagnostics will be placed.	
storageAccountForLogsType	The type of storage account created for boot diagnostics.	“Standard_LRS”
storageAccountForLogsNewOrExisting	Identifies whether to use new or existing Storage Account for boot diagnostics	"new", "existing"
storageAccountForLogsExistingRG	Resource Group containing existing storage account for boot diagnostics	
publicIPAddressName	Name of the Public IP Address	
publicIPDnsName	Unique DNS Prefix for the Public IP used to access the Virtual Machine.	
publicIPNewOrExistingOrNone	Indicates whether the Public IP is new or existing	"new","existing", "none"
publicIPExistingRGName	Resource Group containing existing public IP	
availabilitySetNewOrExistingOrNone	Indicates whether the availability Set is new, none or existing	"new", "existing", "none"
availabilitySetName	Availability set name	
tempLicenseOption	Temporary license options.	"none","TE","CP","TE-SoT","CP-SoT"

Note: Allowed values and even which parameters are available may change with different NIOS releases. Generate new templates from the Azure Portal to find the latest parameters and values.

Configure Parameters for First vNIOS Appliance

In order to deploy the vNIOS appliances in an availability set, you need to change some of the default values in the parameter file.

1. Open the parameters.json file in your text editor.
2. Locate the **adminPassword** parameter. Change the value from null to a password.

Note: The password must be between 6 and 72 characters long, and contain characters from at least 3 of the following groups: uppercase letters, lowercase letters, numbers, and special characters. Additionally, Azure does not allow some specific passwords. The list can be found here: https://docs.microsoft.com/en-us/azure/virtual-machines/windows/faq#what-are-the-password-requirements-when-creating-a-vm.

3. Locate the **availabilitySetNewOrExistingOrNone** parameter. Change the value for this parameter from none to **new**.
4. Add a new parameter, **availabilitySetName** underneath **availabilitySetNewOrExistingOrNone**. Give the new parameter a value to name the availability set. Example:

   "availabilitySetName": {
   "value": "NIOS-AVset"
   },

5. Save the parameters.json file.

Create Parameters File for Second vNIOS Appliance

To deploy a second vNIOS appliance in the availability set, we will create a second parameter file and modify values from the original.

1. Create a copy of the parameters.json file with a new name, for example **parameters2.json**. Open this new file in your text editor.

2. Locate the **vmName** parameter. Change the value to a new name for this second vNIOS appliance.

3. Locate the parameter **subnet1StartAddress**. Change the value to an available IP address in this subnet, different from the fist vNIOS IP. For example, if the original value is 172.27.1.4, change the IP to **172.27.1.5**.
4. Locate the parameter **subnet2StartAddress**. Change the value to an available IP address in this subnet, different from the fist vNIOS IP. For example, if the original value is 172.27.2.4, change the IP to **172.27.2.5**.

5. Locate the **storageAccountNewOrExisting** parameter. Change the value from new to **existing**.
6. Locate the **storageAccountForLogsNewOrExisting** parameter. Change the value from new to **existing**.

7. Locate the **publicIPAddressName** parameter. Change the value to a new name for this second vNIOS appliance public IP.
8. Locate the **publicIPDnsName** parameter. Change the value to a new name for this second vNIOS appliance public DNS name.
9. Locate the **availabilitySetNewOrExistingOrNone** parameter. Change the value to **existing**.

10. Save the parameters2.json file.

Azure CLI

There are multiple methods of deploying ARM templates, using the Azure Portal, command line, or 3rd party tools. This guide uses the Azure Command Line Interface (CLI) to deploy the templates.

Install Azure CLI

Azure CLI is available for Windows, macOS, and most Linux distributions. To install the CLI on your computer, visit https://docs.microsoft.com/en-us/cli/azure/install-azure-cli?view=azure-cli-latest and select the instructions for your operating system.

Azure CLI Basics

- To use the Azure CLI after installation, open a command line or terminal program. Azure CLI commands are organized into groups that represent various services available in Azure. To see a list of the available groups, use the command:

  az --help

- The **--help** argument is globally available in the CLI and can be used to dig deeper into subgroups, commands, and arguments available. For example, to find the subgroups and commands available for the "**group"** group, use the command:

  az group --help

- For more information on the Azure CLI, visit: https://docs.microsoft.com/en-us/cli/azure/?view=azure-cli-latest.

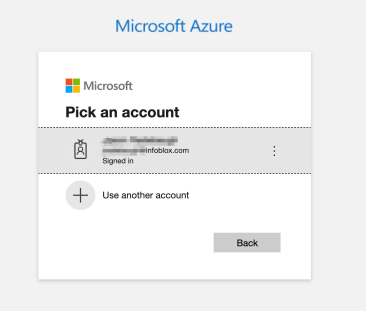
Deploy vNIOS in Availability Set Using ARM Template

To deploy your availability set using the ARM templates you created:

1. Open a command prompt or terminal application.
2. Change directories to the location of the ARM templates created earlier.
3. Login to Azure using the following command:

   az login

4. A login window will open in your default browser. Follow the prompts to sign in with your Azure credentials.



5. After logging in, return to the command line or terminal application.
6. To find your available subscriptions, run the following command:

   az account list

7. Set the subscription you want to deploy to by substituting your subscription ID into:

   az account set --subscription <your-subscription-ID>

8. Verify the correct subscription is set using:

   az account show

9. Next, create a resource group to deploy the vNIOS appliances into:

   az group create --location <location> --name <resource-group-name>

The location and name should match those specified in the parameters.json file created earlier. Example:

Validate ARM Templates

The Azure CLI provides tools to validate ARM templates prior to deployment. This tool only validates the syntactical correctness of templates and does not guarantee a successful deployment.

To validate your template, run the following command, substituting in the name of your resource group:

az group deployment validate --resource-group <resource-group-name> --template-file template.json --parameters @parameters.json

There may be a large amount of output from this command. Look at the first line of output for **"error": null**. This indicates that the file was successfully validated. If you don't see this, look through the error messages to troubleshoot your template files.

You can repeat this process for the second parameters file using:

az group deployment validate --resource-group <resource-group-name> --template-file template.json --parameters @parameters2.json

Deploy ARM Template Using Azure CLI

To deploy the Availability Set along with the first vNIOS for Azure VM, run the following command, inserting the name of your resource group:

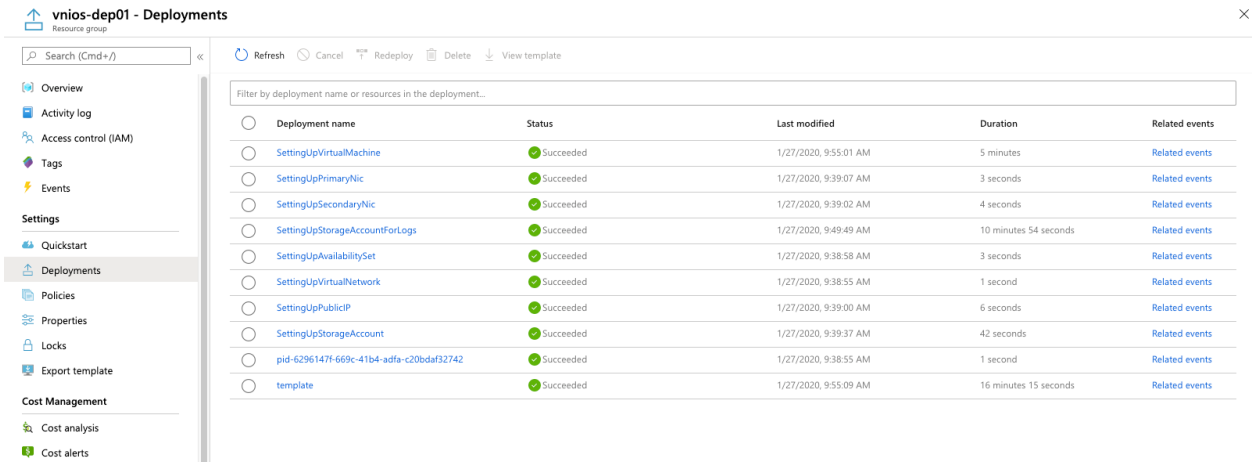
az group deployment create --resource-group <resource-group-name> --template-file template.json --parameters @parameters.json

The deployment may take 10 to 20 minutes to complete.

Monitoring Deployment

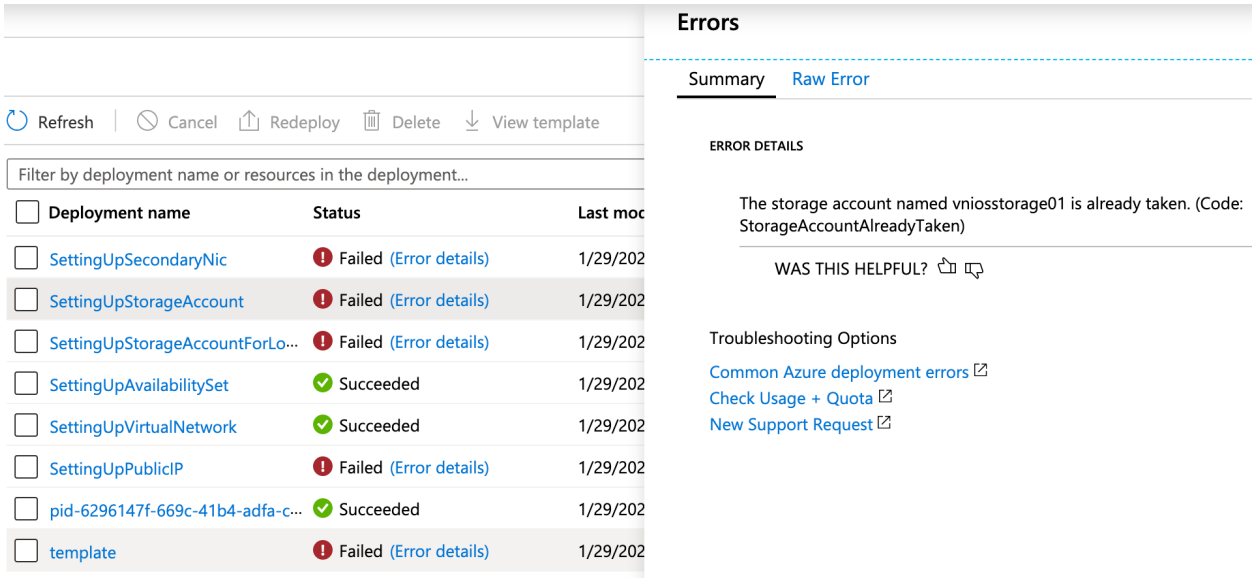
To monitor the progress of your deployment:

1. Open the Azure Portal in your browser.
2. Click on **Resource groups**.
3. Select the resource group you created from the list.
4. In the Resource group blade, under Settings, click on **Deployments**.



5. Watch for the status of all resources to show Succeeded.

If any of the resources show a status of Failed, Click on **Error details** to explore the cause.



If the deployment fails, it is best to delete the entire resource group, make needed corrections to your template files, and start the deployment again by recreating the resource group.

Deploy Second vNIOS Appliance

To deploy the second vNIOS for Azure VM into the availability set, ensure the first deployment has successfully completed.

Return to your command prompt or terminal application.

Run the following command, inserting the name of your resource group:

az group deployment create --resource-group <resource-group-name> --template-file template.json --parameters @parameters2.json

The deployment may take 10 to 20 minutes to complete. This deployment can be monitored using the same method as the first.

Configure Azure vNIOS Appliances

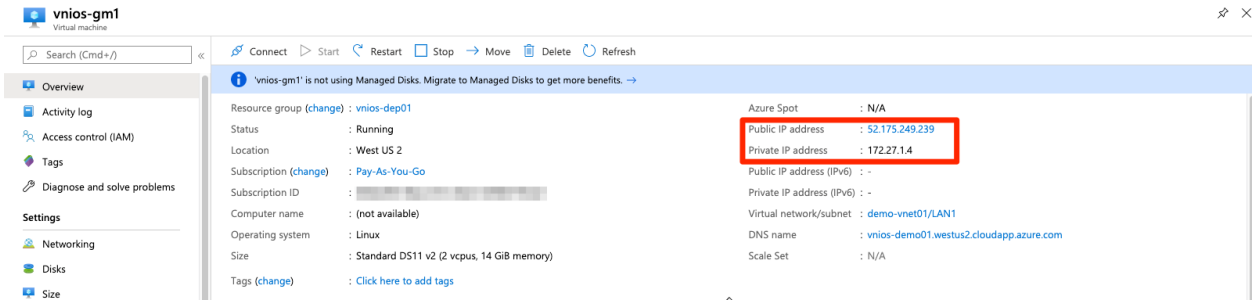
The vNIOS for Azure appliances can be added as members to an existing on-premise or multi-cloud grid, or configured as a new grid running entirely in Azure. To add your vNIOS appliances to an existing grid or for other use cases not covered by this guide, refer to the NIOS Administrator guide or other documents available on the Infoblox Support site (https://support.infoblox.com/).

Find Public and Private IP Addresses for vNIOS VMs

Before configuring your vNIOS for Azure appliances, you need to find the public and private IP addresses for these VMs. The public IP addresses will be used to connect to the VMs. The private IP addresses will be used in the configuration process.

1. In the Azure Portal, click on **Virtual machines**.
2. Find your first vNIOS VM and click on its name.

3. On the Overview page of the VM blade, locate the Public and Private IP addresses and make a note of both.

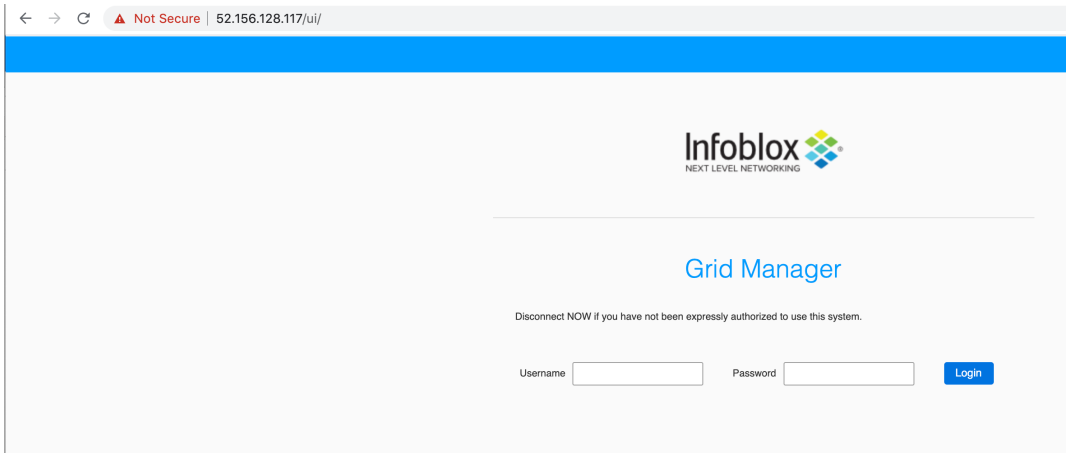


4. On the Virtual machines blade, click on the name of your second vNIOS VM.
5. Note the Public and Private IP addresses for this VM.

Configure Grid Master and Add Member

To connect to your vNIOS VM:

1. Open a web browser.
2. Navigate to the the public IP address of your VM, prefixing the address with **https://**

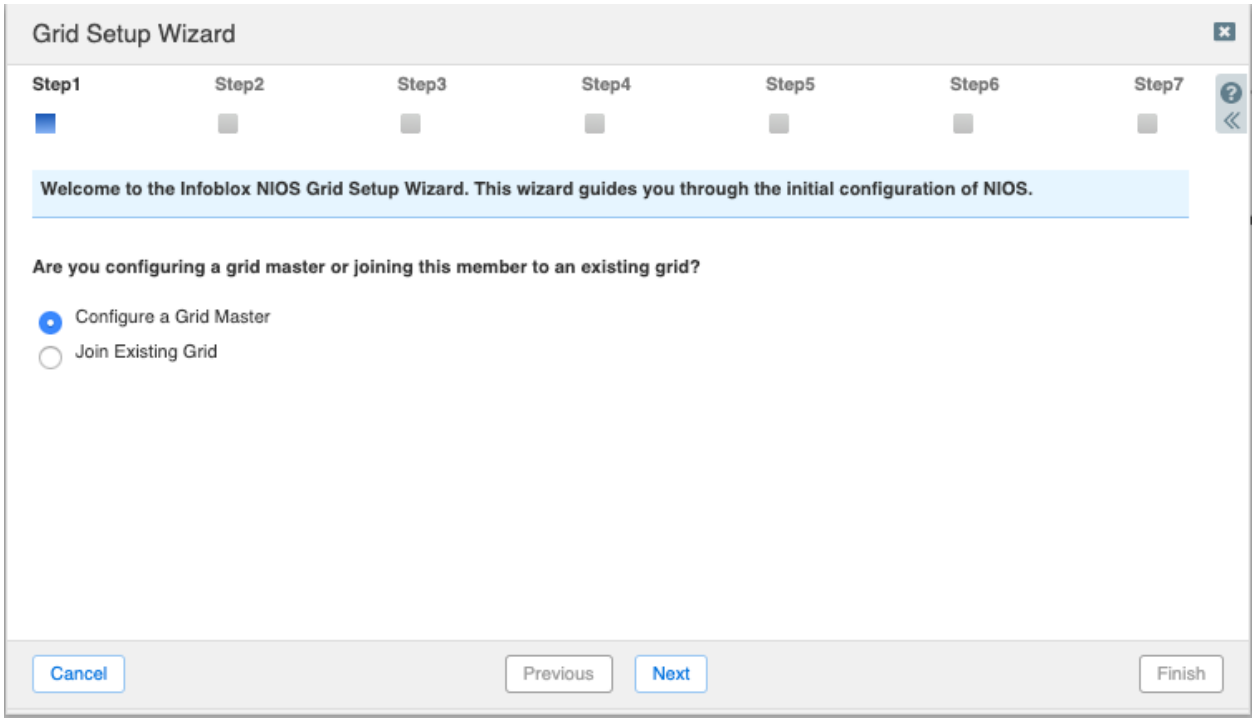


Note: NIOS uses a self-signed certificate. Warnings about the connection being insecure are to be expected and might require that you add an exception before being able to connect.

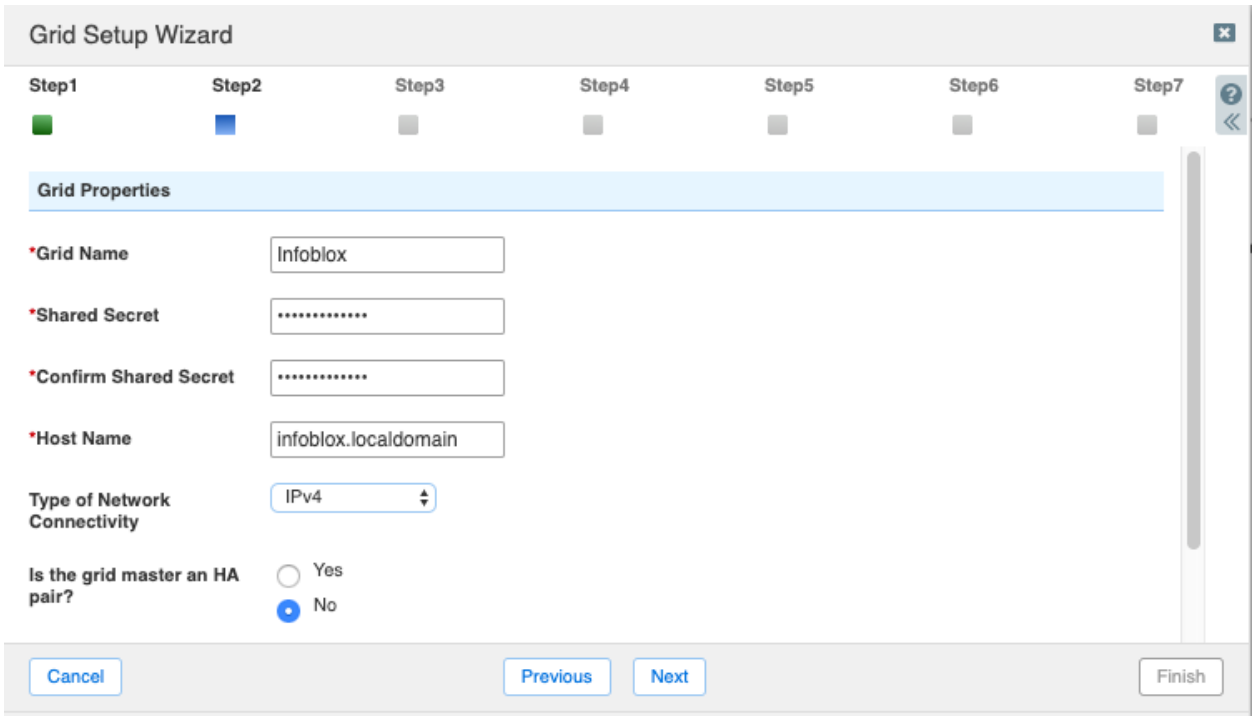
3. Login with the username **admin** and the password you created during deployment.
4. Read and accept the Infoblox End-User License Agreement.

Configure vNIOS VM as a Grid Master

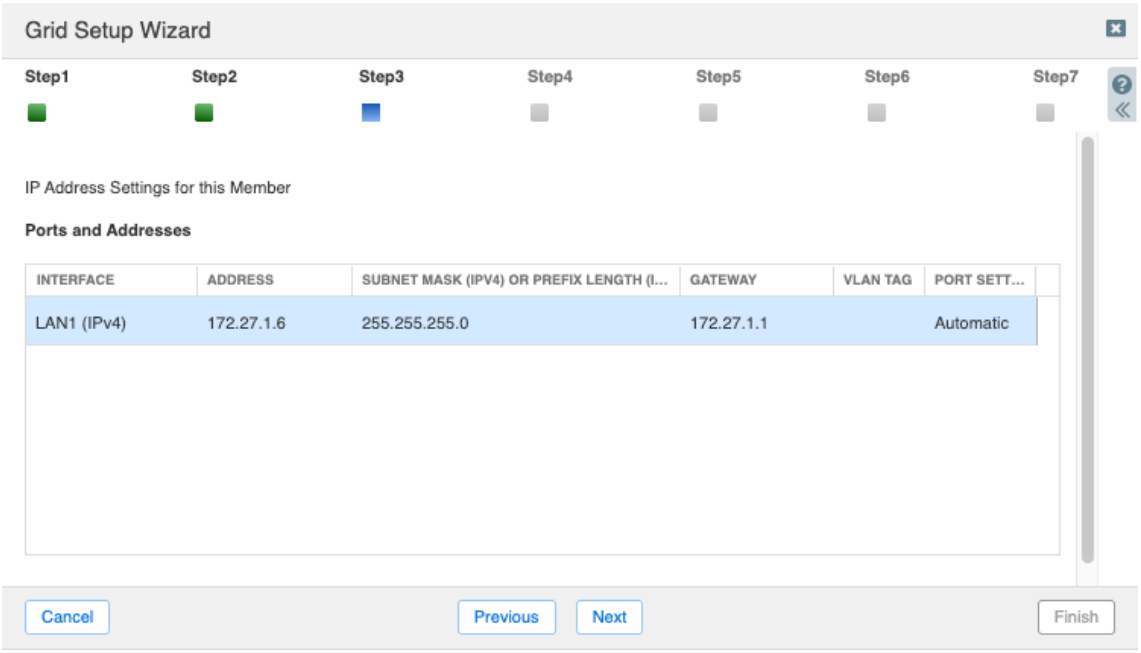
1. When you login to the Grid Manager for the first time, the Grid Setup Wizard will open.
2. Select **Configure a Grid Master** and click **Next**.



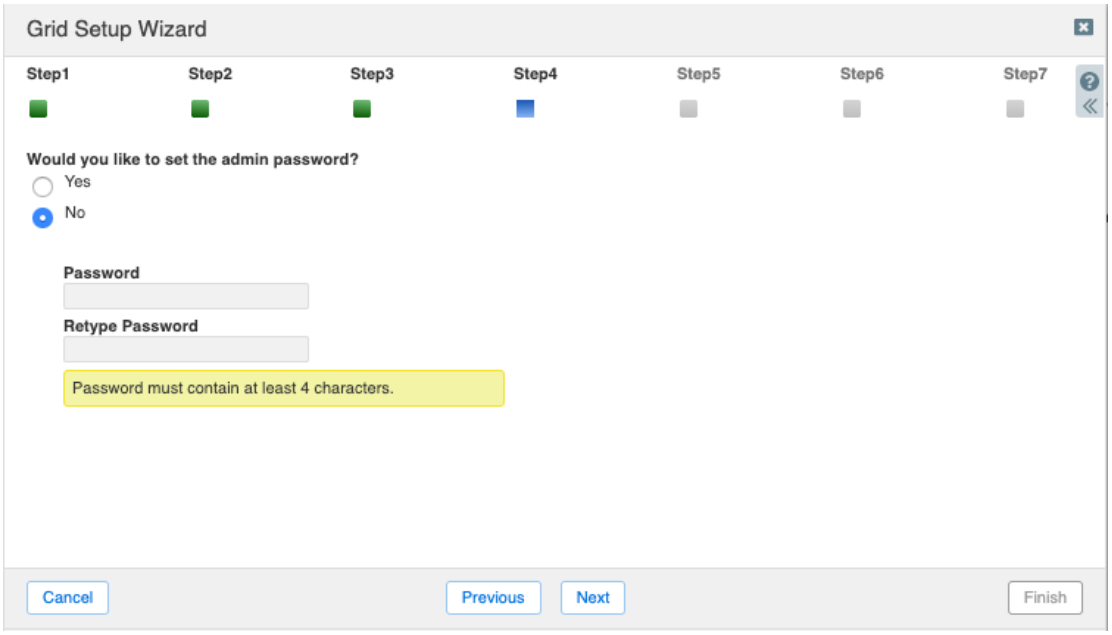
3. On Step 2, enter a name for your grid, a Shared Secret used to join new members to the grid, and a Host Name for you grid master. Or, leave the default values and click Next.



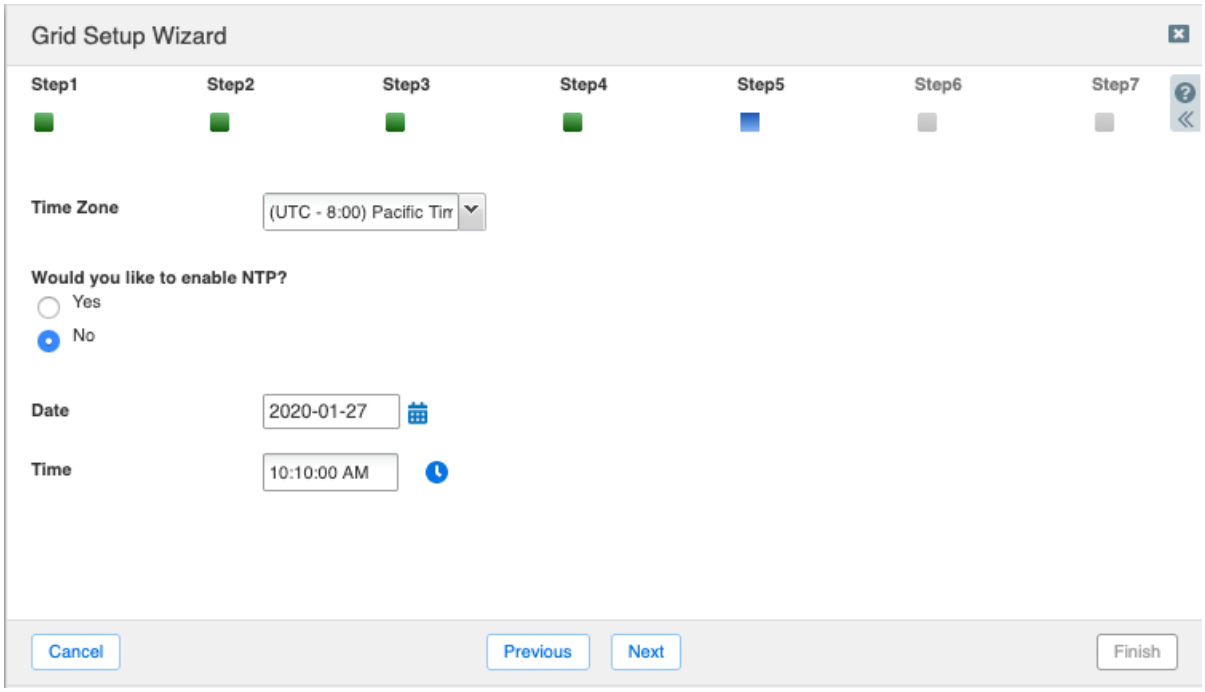
4. On Step 3, verify the IP address information for your grid master. This should be the internal IP address of vNIOS appliance. Do not make any changes to this page unless necessary. Click **Next** to continue to Step 4.



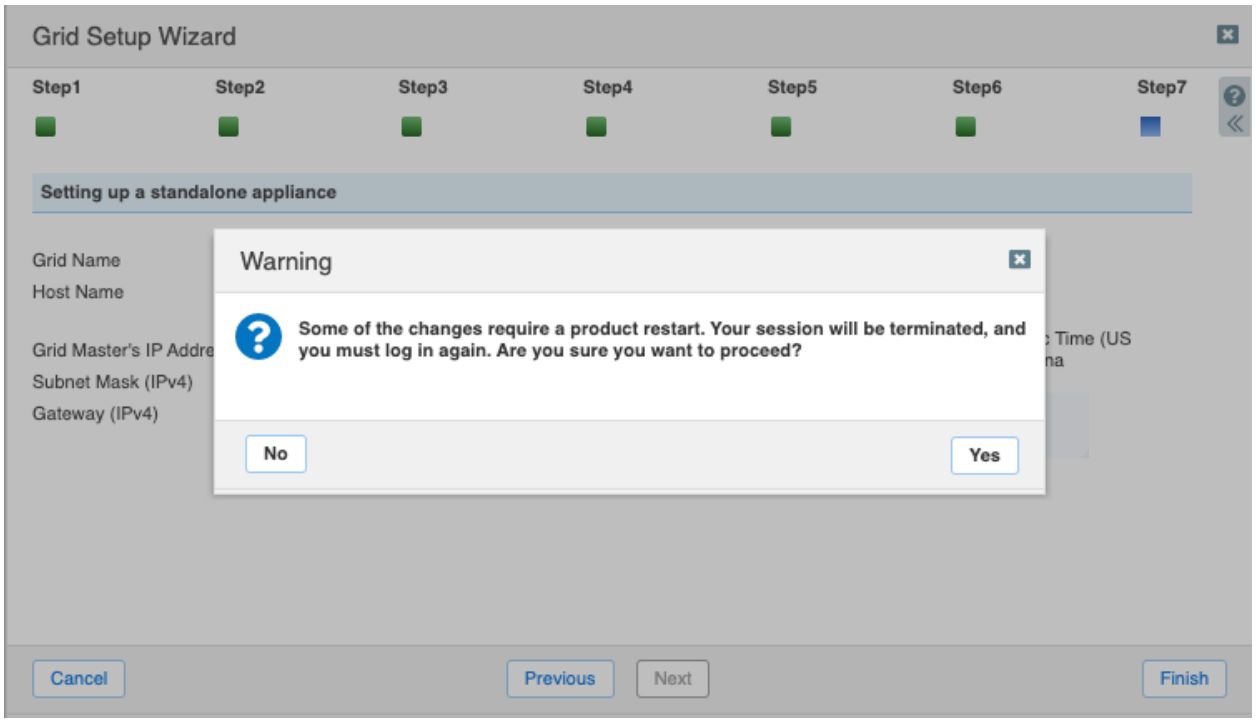
5. On Step 4, you can optionally change the administrator password. Click **Next** to continue.



6. On Step 5, select your Time Zone from the dropdown.
7. Enter the current time and click **Next**.



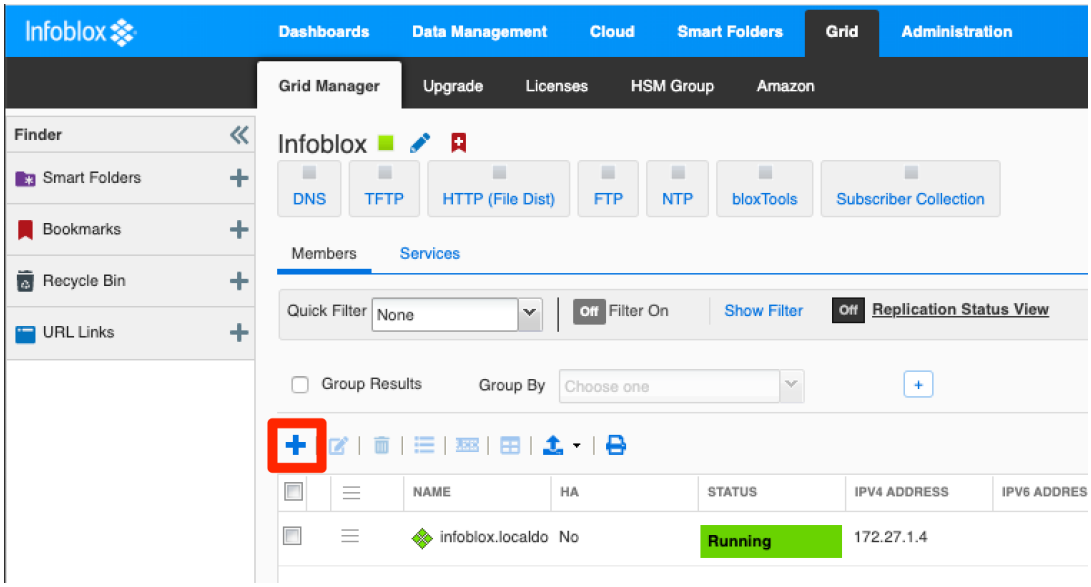
8. Leave the default values for Step 6 and click **Next**.
9. On Step 7, click **Finish**.
10. Click **Yes** in the warning window to restart your vNIOS appliance.



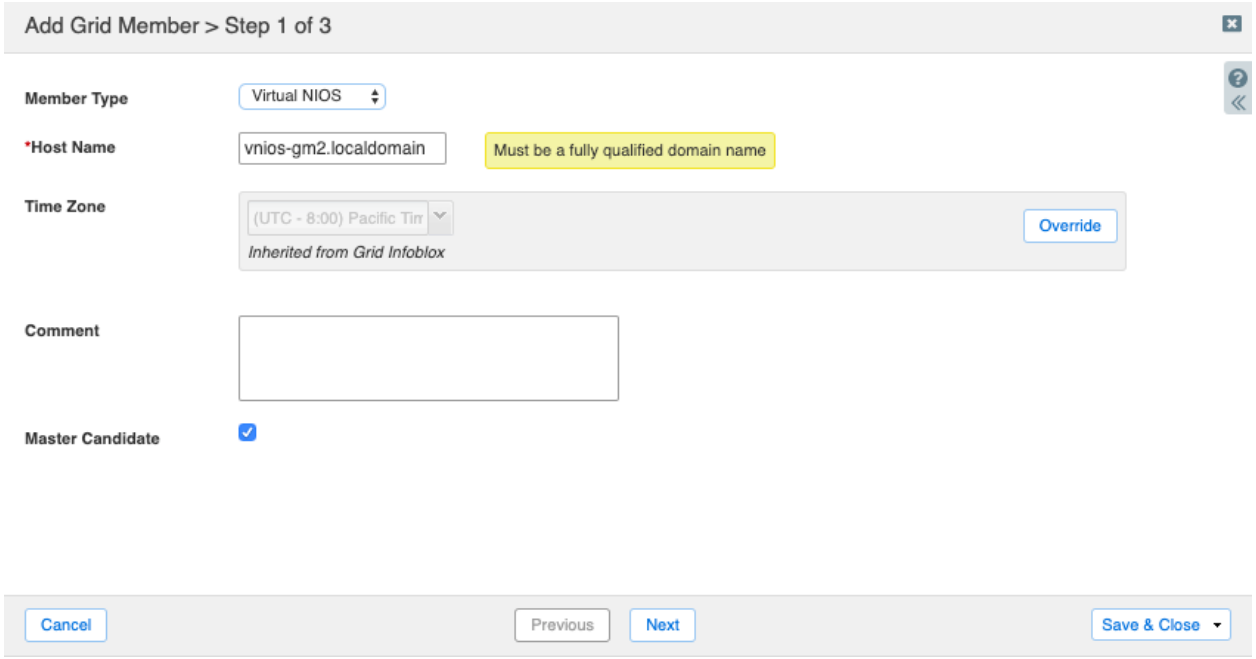
Add Second vNIOS VM as a Grid Member

1. After the vNIOS appliance reboots, log back in to Grid Manager.

2. From the **Grid** tab, navigate to **Grid Manager** → **Members**.

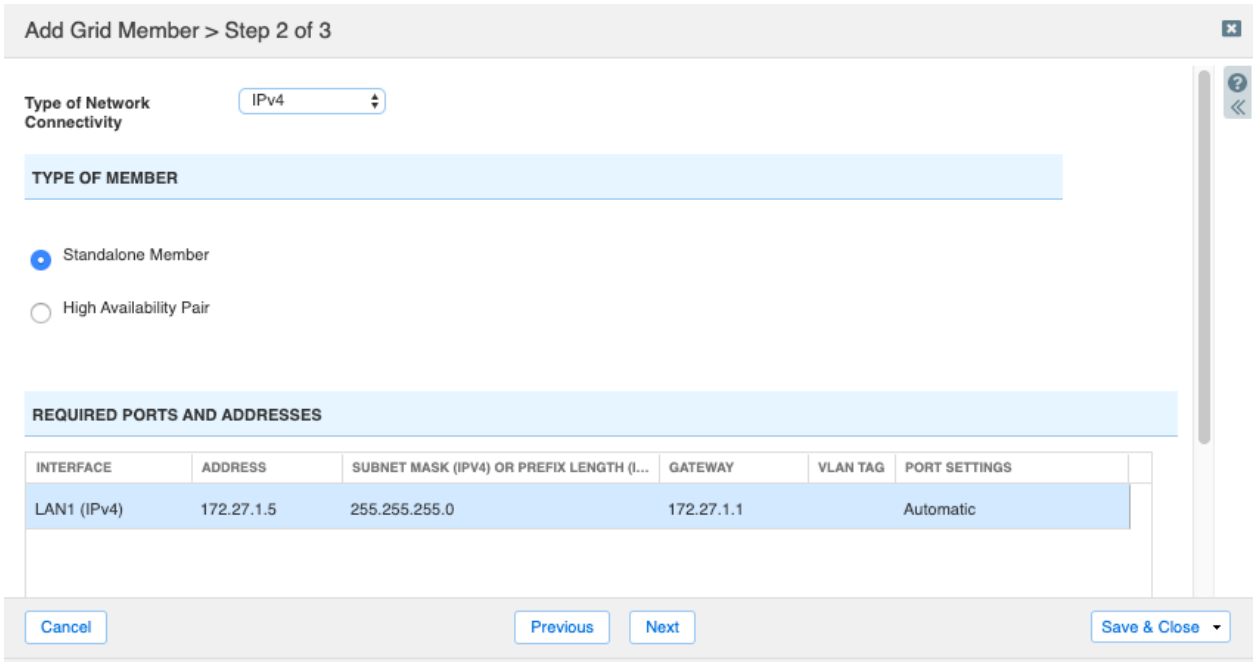


3. Click the + to add a new grid member.
4. In the Add Grid Member wizard, select **Virtual NIOS** from the Member Type dropdown.
5. Enter a Host Name for the new member (this must be a fully qualified domain name but does not have to be resolvable, example: **vnios-gm2.localdomain**).
6. Optional: Select the checkbox next to Master Candidate and click **Next**.



7. On Step 2, enter the private IP address of your second vNIOS appliance.

8. Enter the Subnet Mask and Gateway Address (the gateway will be your subnet prefix followed by .1 as a default in Azure, example: 172.27.1.1).

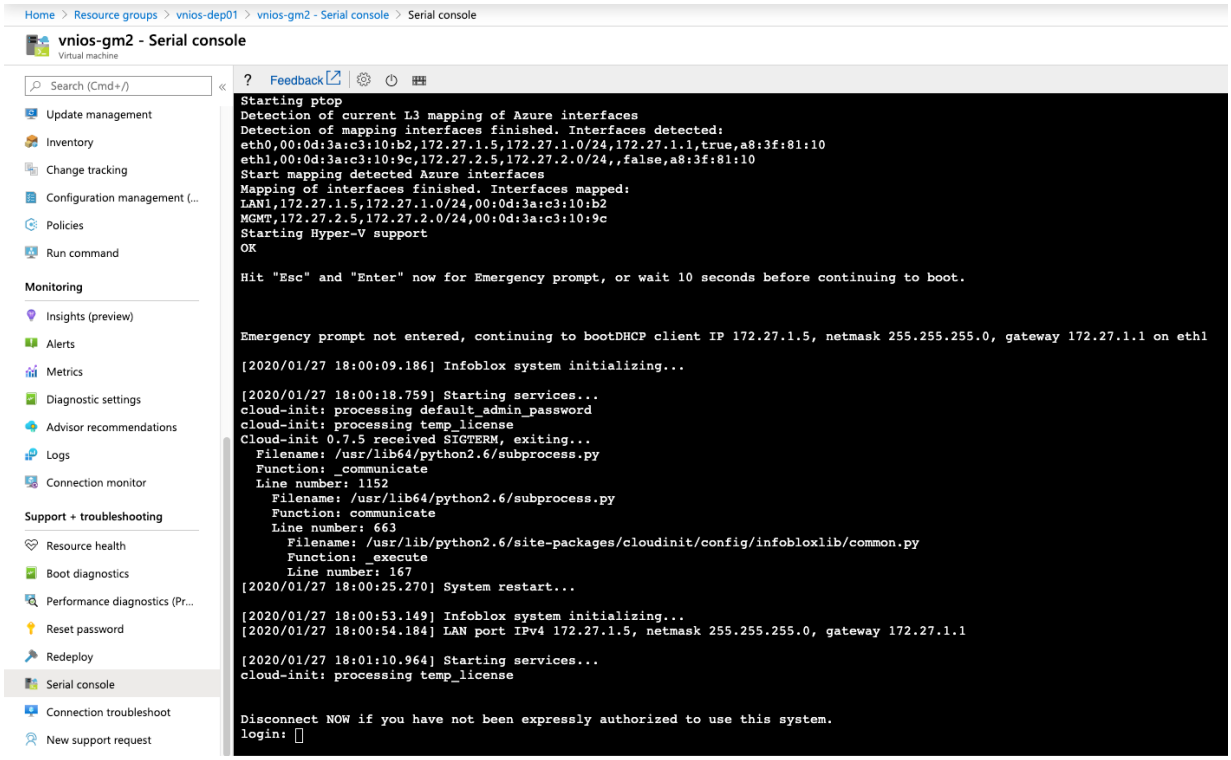


9. Click **Save & Close**.

Join Second vNIOS VM to the Grid

Azure provides a virtual serial console for VMs, which can be used to connect to the command line interface of your vNIOS appliance. You could also connect via SSH using the public IP of your appliance.

1. In the Azure Portal, navigate to your second vNIOS VM.
2. On the VM blade, scroll down and select **Serial console** under Support + troubleshooting.



3. Login to the console with username **admin** and the password you created during deployment.
4. To join the appliance to the grid, use the command:

   set membership

5. When prompted, enter the private IP address of your Grid Master, press **Enter**.
6. Enter the Grid Name (default is **Infoblox**), press **Enter**.
7. Enter the Grid Shared Secret (default is **test**), press **Enter**.
8. Enter **y** to confirm and press **Enter**.

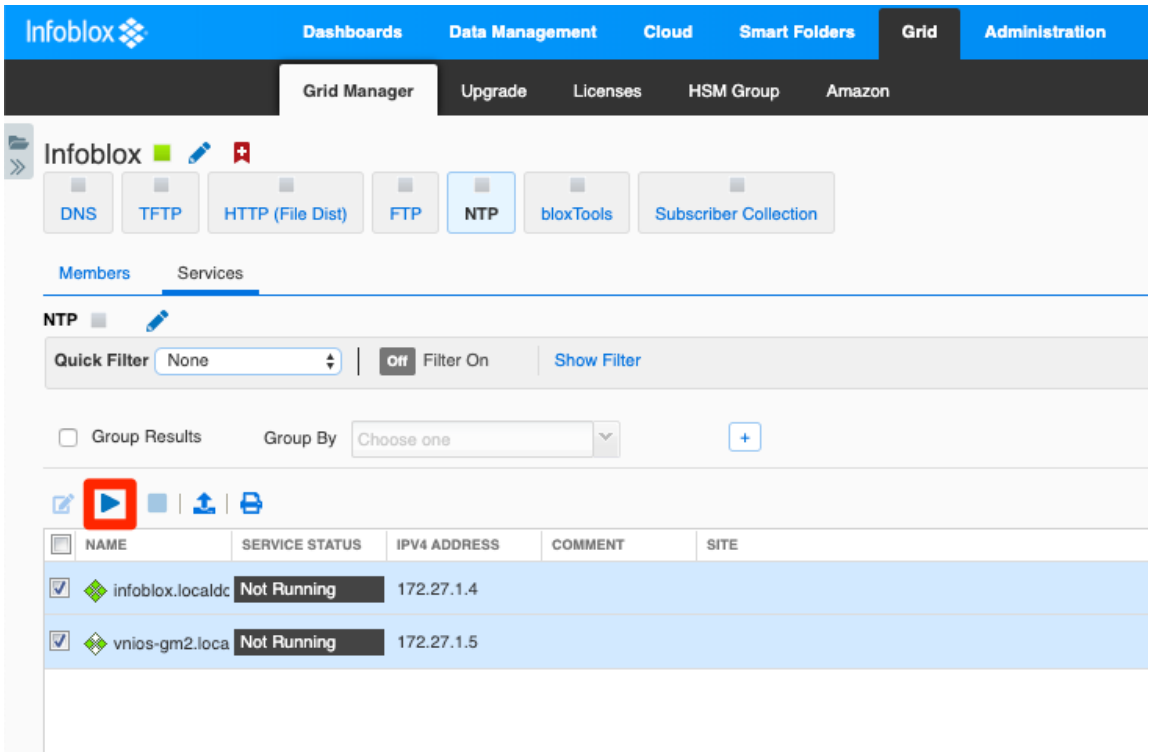
```
Disconnect NOW if you have not been expressly authorized to use this system.
login: admin
Local password:

              Infoblox NIOS Release 8.4.3-383835 (64bit)
     Copyright (c) 1999-2019 Infoblox Inc. All Rights Reserved.

                  type 'help' for more information


Infoblox > set membership
Join status: No previous attempt to join a grid.
 Enter New Grid Master VIP: 172.27.1.4
Enter Grid Name [Default Infoblox]: Infoblox
Enter Grid Shared Secret: test
  Join grid as member with attributes:
     Grid Master VIP:     172.27.1.4
     Grid Name:           Infoblox
     Grid Shared Secret:  test

WARNING: Joining a grid will replace all the data on this node!
        Is this correct? (y or n): y
```

Configure NTP and DNS

In order to use your new vNIOS appliances for DNS and discovery of resources in Azure, you will need to enable some basic services, Network Time Protocol (NTP), and Domain Name System (DNS).

Start the NTP Service

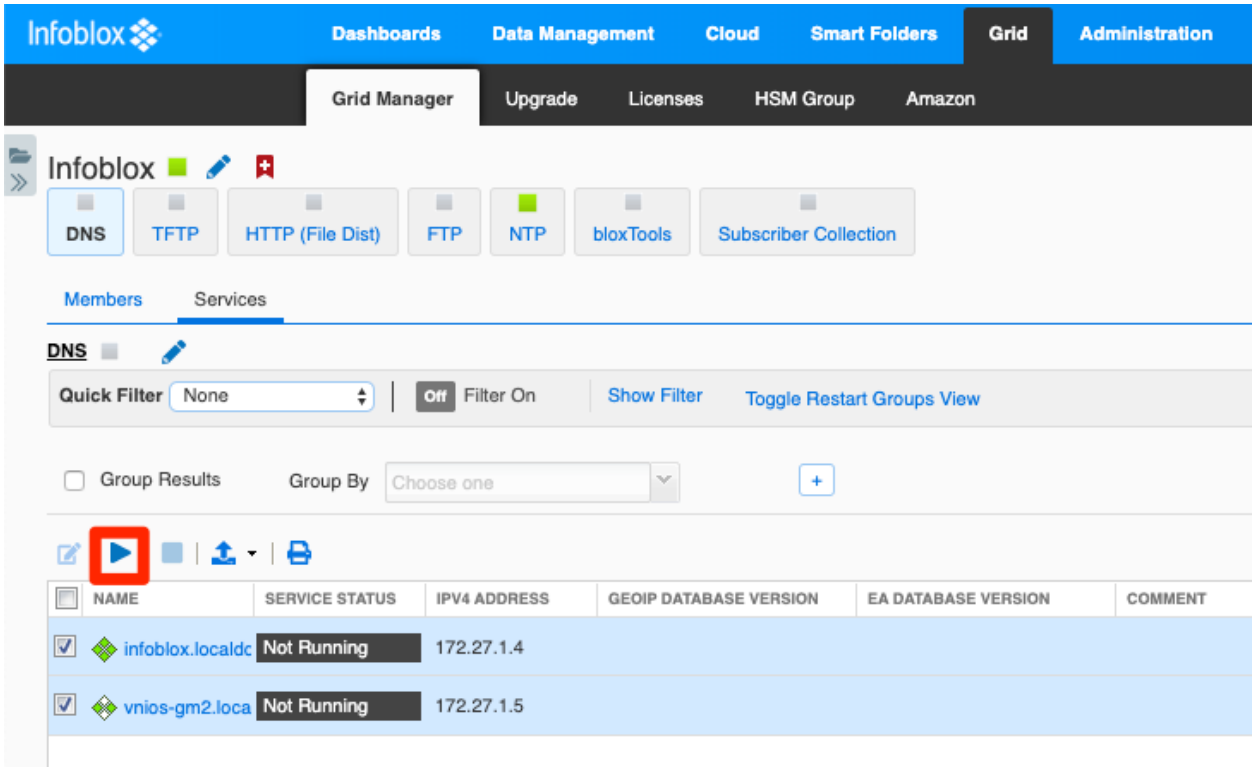
1. Log back in to Grid Manager
2. From the Grid tab, navigate to **Grid Manager** → **Members**.
3. Verify that your second vNIOS member shows a Status of **Running**.
4. Click on **NTP** in the Services bar.



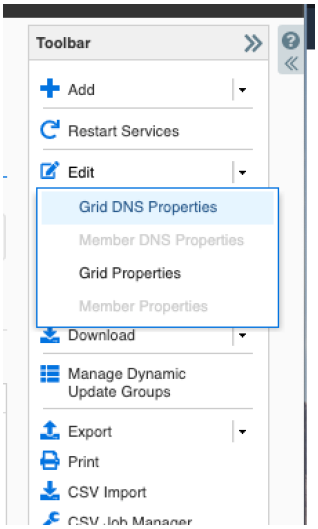
5. Select the checkboxes next to both grid members.
6. Click the ▶ to start the NTP service.
7. Click **Yes** in the warning window.

Start and Configure the DNS Service

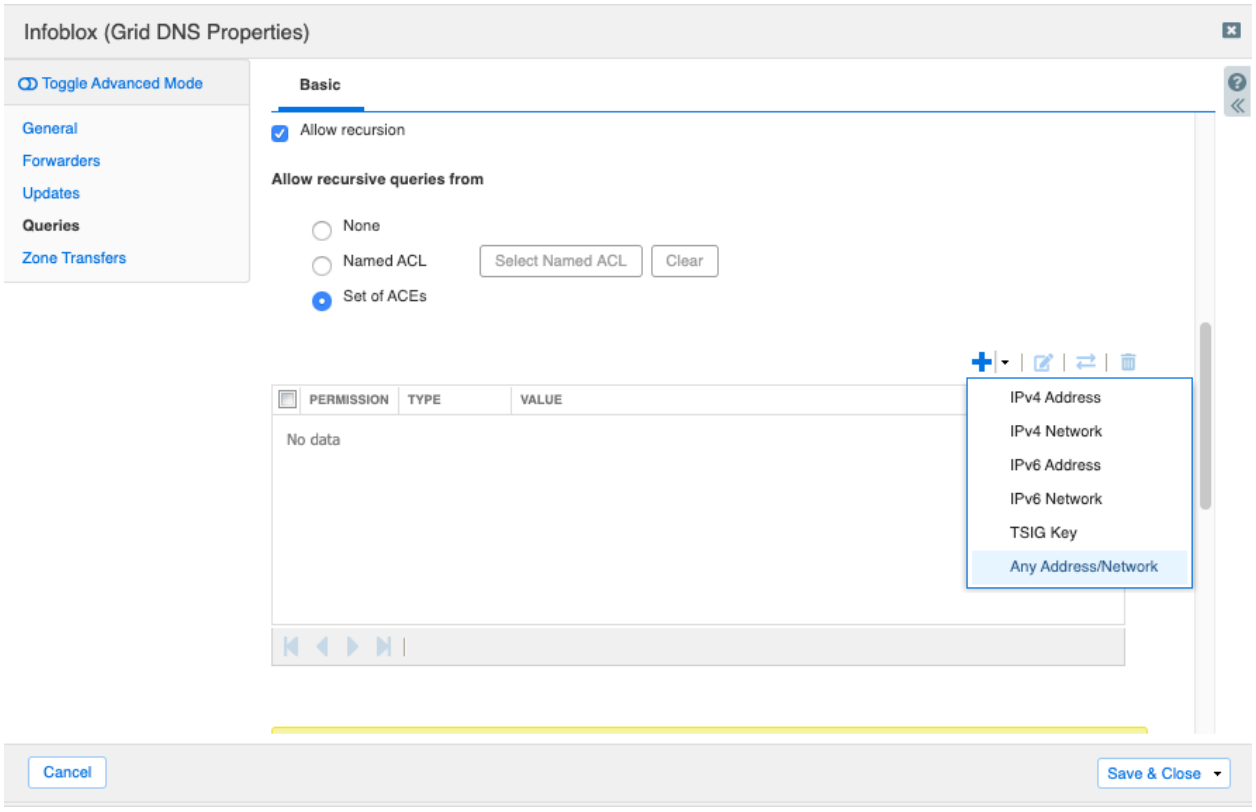
1. Click on **DNS** in the Services bar.



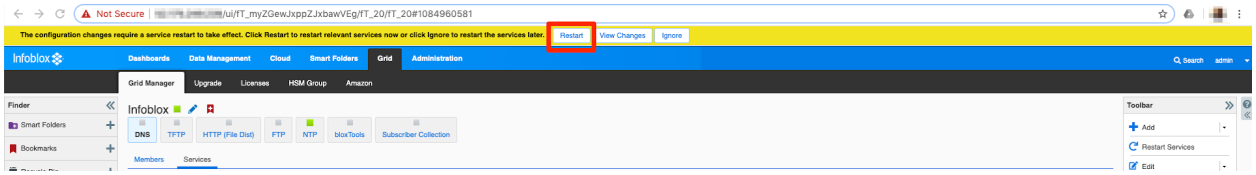
2. Select the checkboxes next to both grid members.
3. Click the  to start the DNS service.
4. Click **Yes** in the warning window.
5. From the **Toolbar**, click **Edit**, then **Grid DNS Properties** from the dropdown.



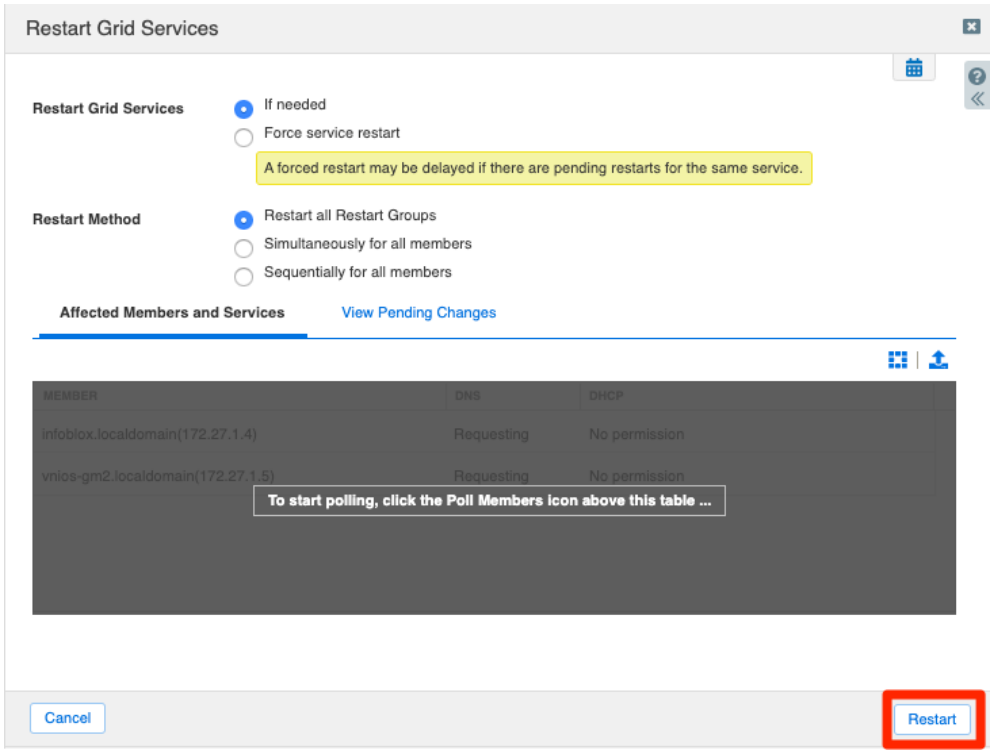
6. Select the **Queries** tab.
7. Select the checkbox for **Allow recursion**.
8. **Select Set of ACEs**.
9. Open the add ACE dropdown by clicking the + ▾ .
10. Select **Any Address/Network** from the dropdown.



11. Click **Save & Close**.
12. Click **Yes** in the warning window.
13. Click **Restart** in the warning bar when prompted.



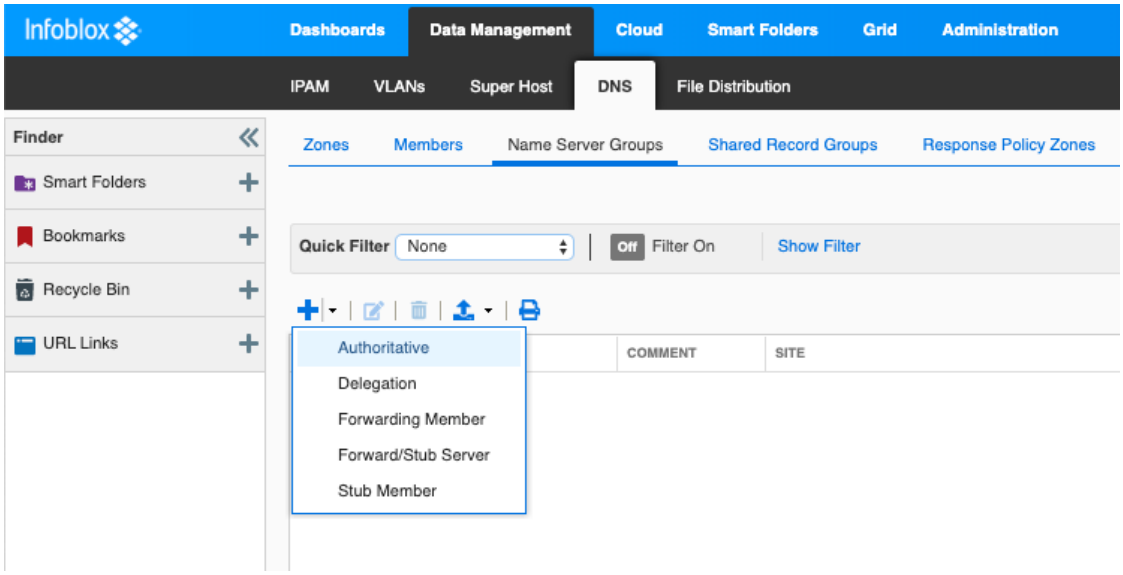
14. Click **Restart** in the Restart Grid Services window.



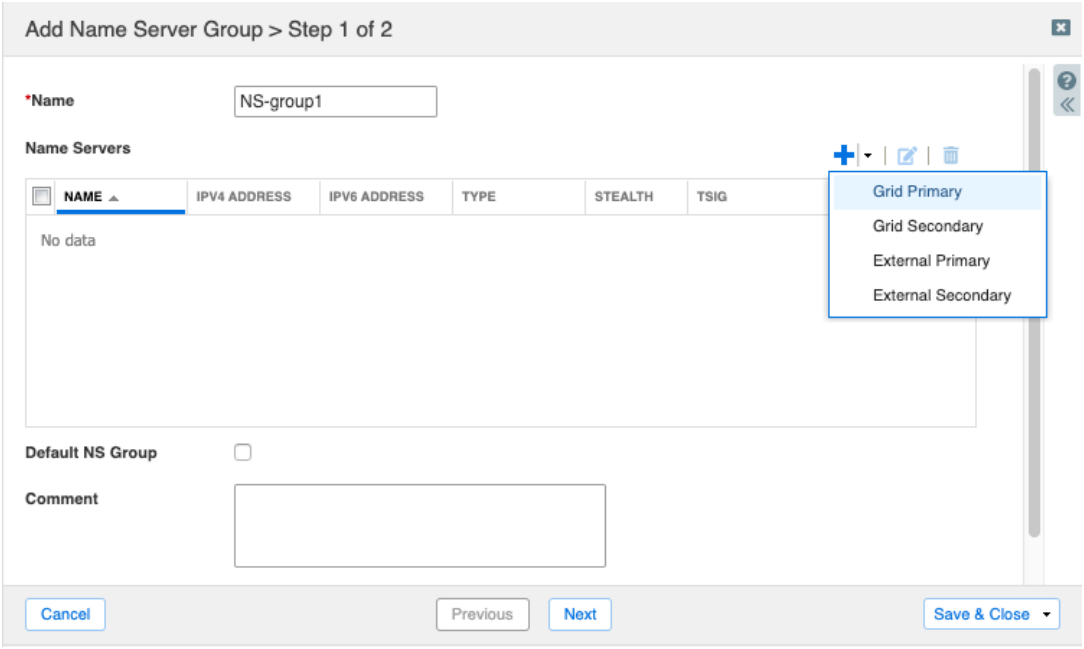
Create a DNS Name Server Group

You can create name server groups using Infoblox NIOS, which streamlines management of DNS zones and records.

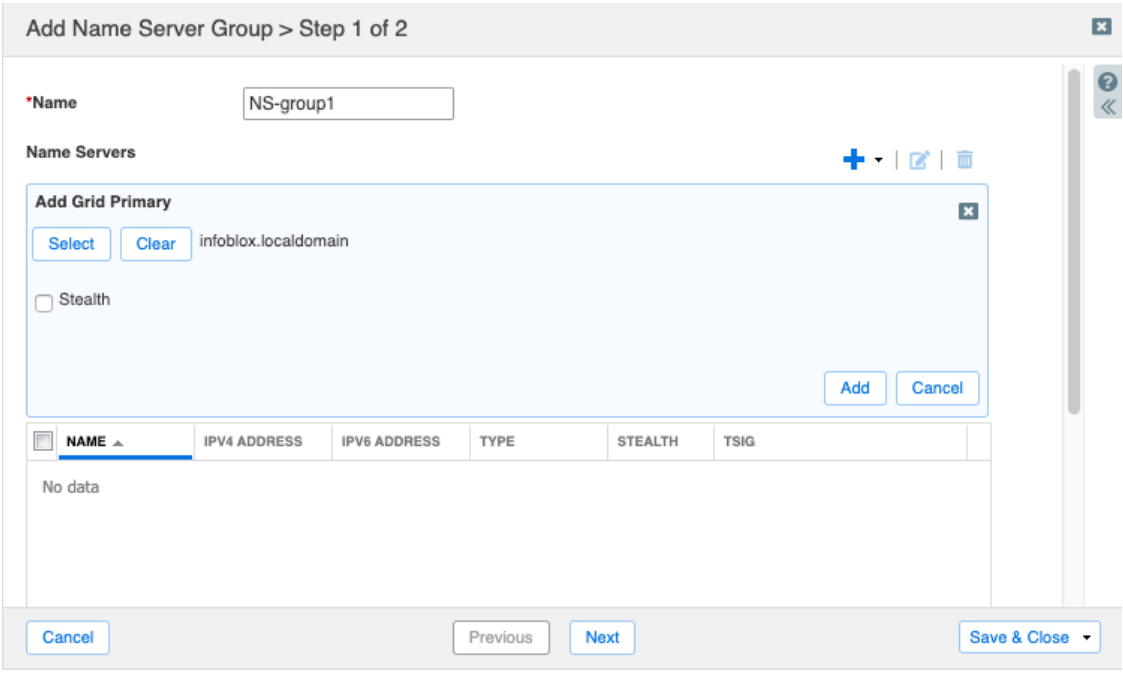
1. From the **Data Management** tab, navigate to **DNS → Name Server Groups**.
2. Open the add Name Server Group dropdown by clicking the + .
3. Select **Authoritative** from the dropdown.



4. In the Add Name Server Group wizard, name the group.
5. Open the add name server dropdown by clicking the + ▾ .
6. Select **Grid Primary** from the dropdown.



7. Click **Select** and click on your Grid Master in the list.
8. Click **Add**.



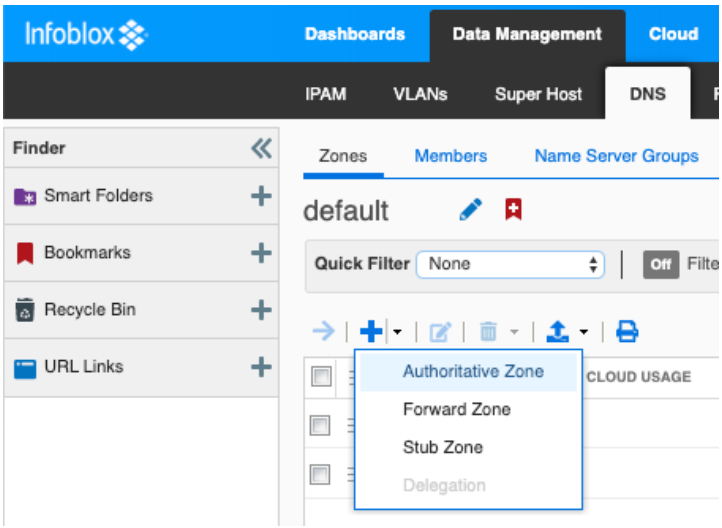
9. Repeat the above steps to add your second grid member to the group.

10. Click **Save & Close**.
11. Click **Yes** in the warning window.

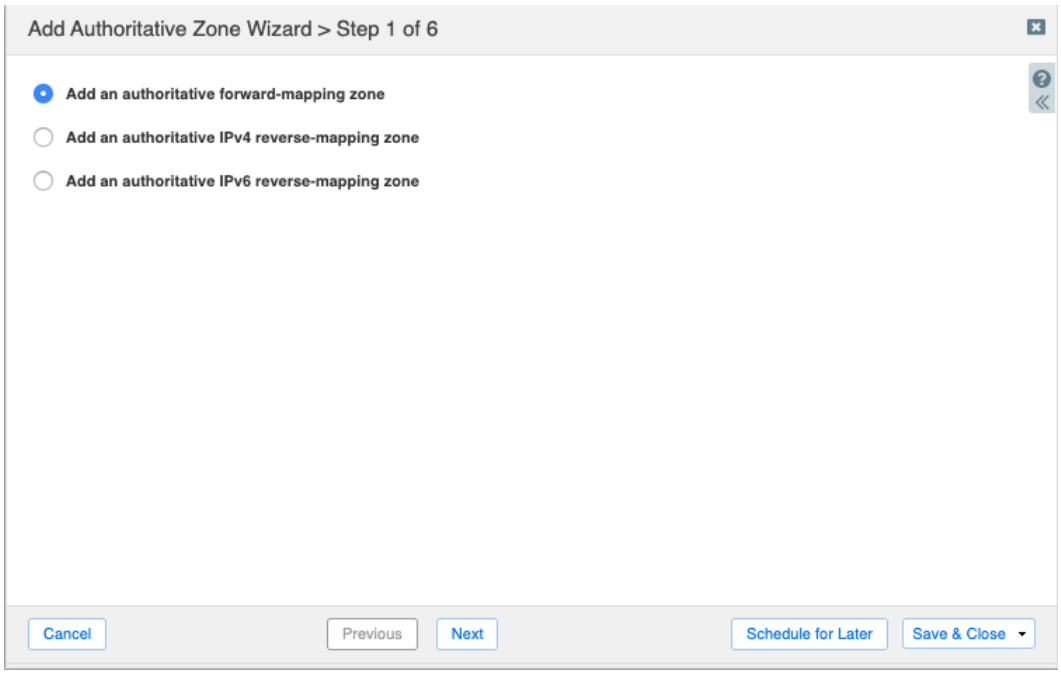
Create a DNS Zone

To enable automatic creation of DNS records when using vDiscovery for Azure, the Infoblox grid must be authoritative for at least one DNS Zone. To create a DNS zone in Grid Manager:

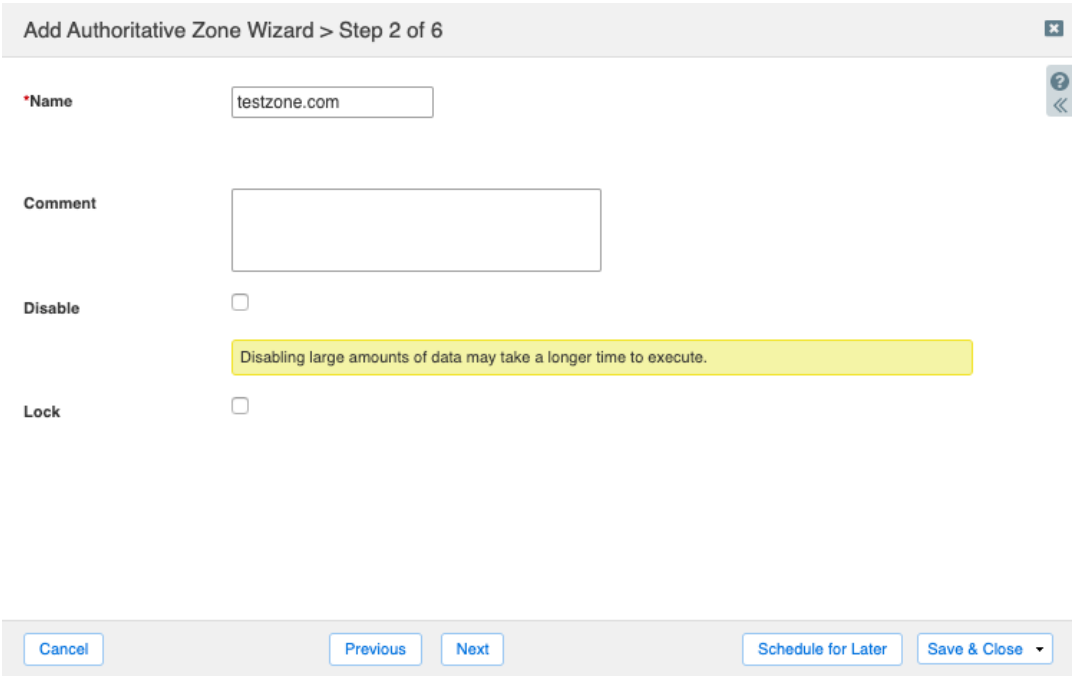
1. From the Data Management tab, navigate to **DNS → Zones**.
2. Open the add zone dropdown by clicking the  .
3. Select **Authoritative Zone** from the dropdown.



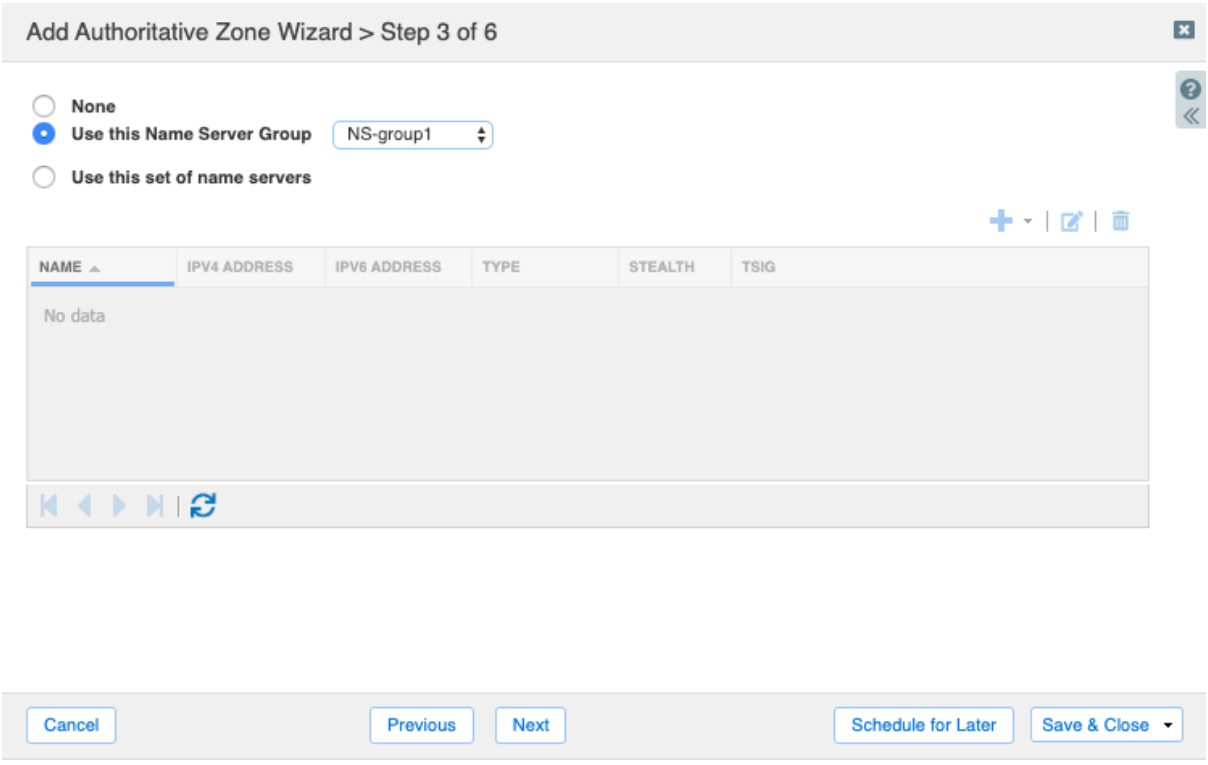
4. In the Add Authoritative Zone Wizard, select **Add an authoritative forward-mapping zone**



5. Click **Next**.
6. Name your zone and click **Next**.



7. On Step 3, select **Use this Name Server Group**.
8. Select your Name Server Group from the dropdown.

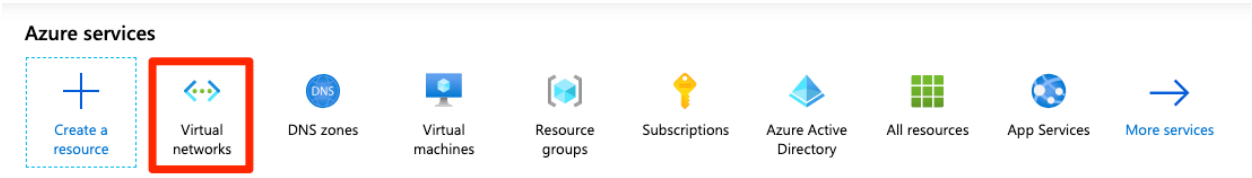


9. Click **Save & Close**.
10. Click **Yes** in the warning window.
11. Click **Restart** in the warning bar when prompted.
12. Click **Restart** in the Restart Grid Services window.

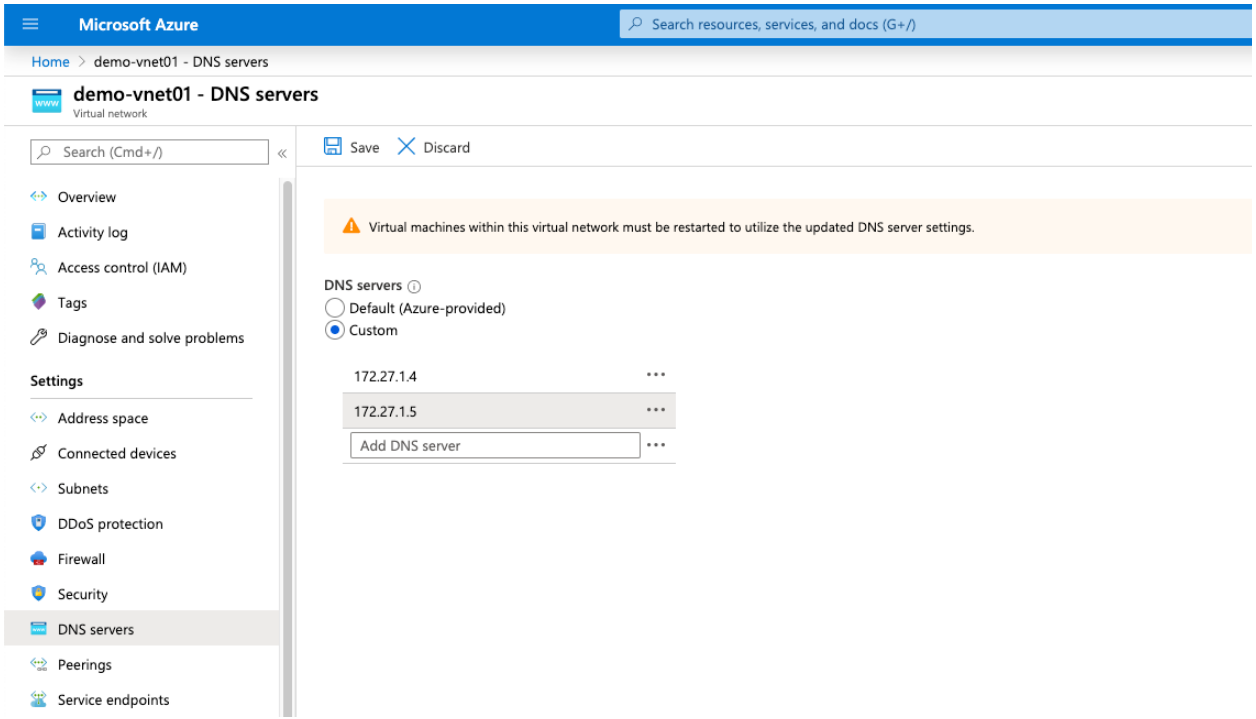
Configure vNIOS as Primary DNS Server for Azure VNETs

Once your vNIOS for Azure appliance has been deployed, you can update Azure VNET settings to allow VMs to use the Infoblox device for DNS resolution.

1. In the Azure Portal, click on **Virtual network**



2. Select your VNET from the list.
3. In the VNET blade, click on **DNS servers** under settings.



4. Select **Custom**.
5. Enter the internal IP address of your vNIOS VM.
6. Add your second vNIOS VM internal IP if desired.
7. Click **Save**.

Infoblox vDiscovery for Azure

The Infoblox vDiscovery feature is very useful for detecting and obtaining information about Tenants, VNETs, Subnets, and Virtual Machines (VM's) operating in your public cloud environments. This can include Microsoft Azure, Amazon Web Services (AWS), and Google Cloud Platform (GCP).

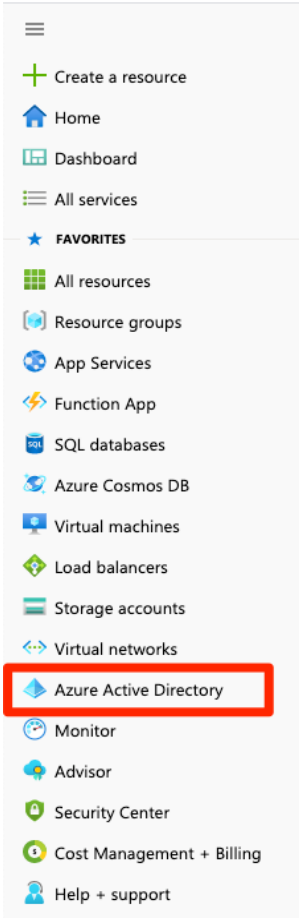
Many organizations operate hybrid and multi-cloud environments that may contain many subscriptions and accounts. These environments tend to be very dynamic, with things such as VMs being created and terminated on a frequent basis. This makes it difficult to keep track of everything. With Infoblox vDiscovery, tasks can be configured to run automatically allowing your Infoblox vNIOS appliance to keep track of all cloud environments, storing this data in IPAM. Using vDiscovery in conjunction with the Cloud Network Automation (CNA) feature, you will gain enhanced visibility into your cloud environments, all within a 'single pane of glass'.

Enable vDiscovery in Azure

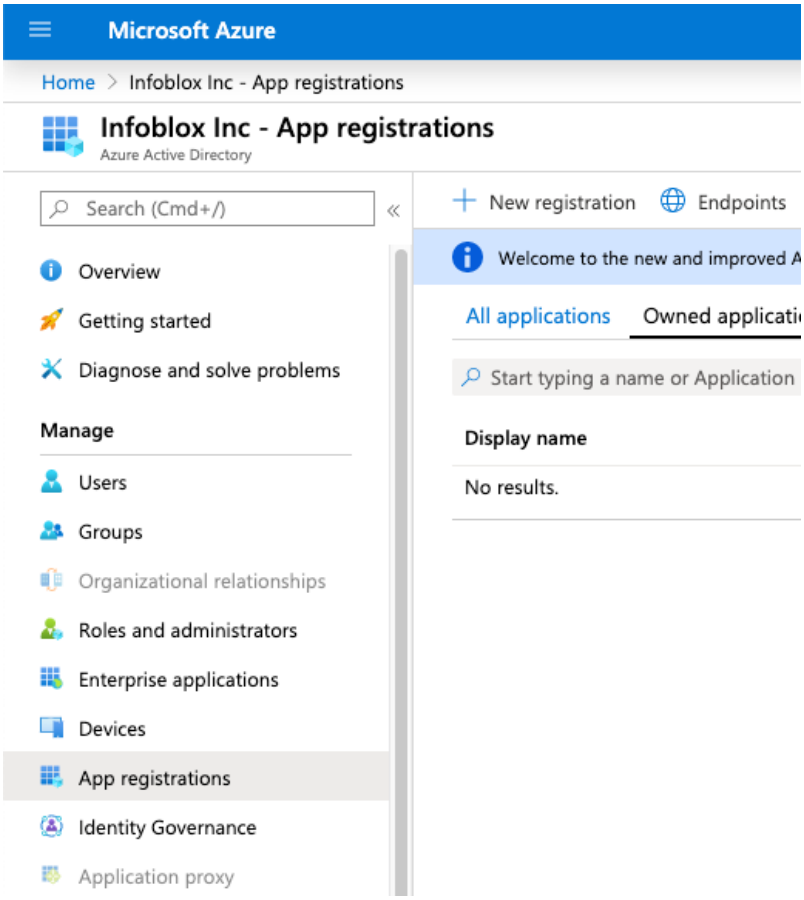
In order to use vDiscovery in Azure, you must integrate the discovery application with Azure Active Directory (AAD) for secure sign in and authorization.

Create an App Registration in Azure Active Directory

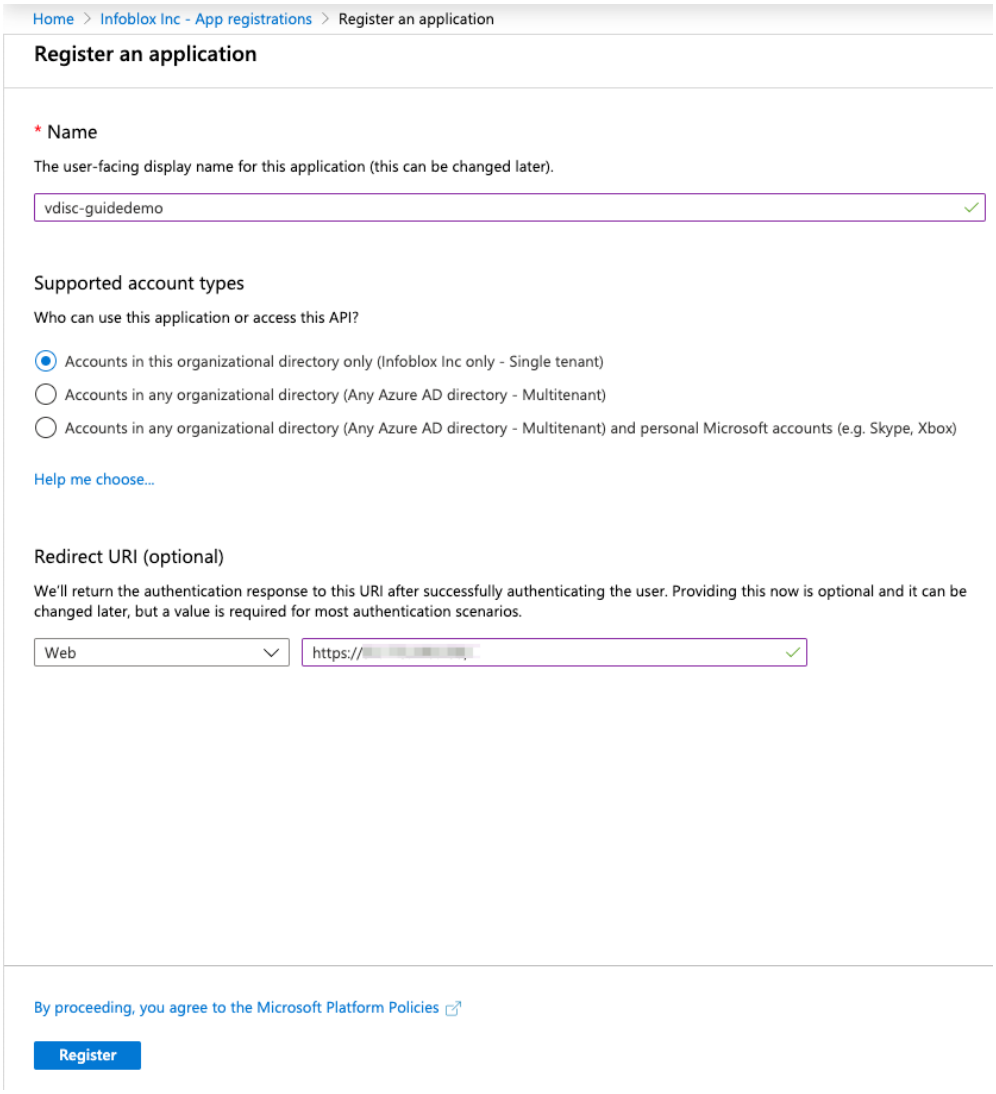
1. From the Azure Portal, click the  menu.
2. Select **Azure Active Directory**.



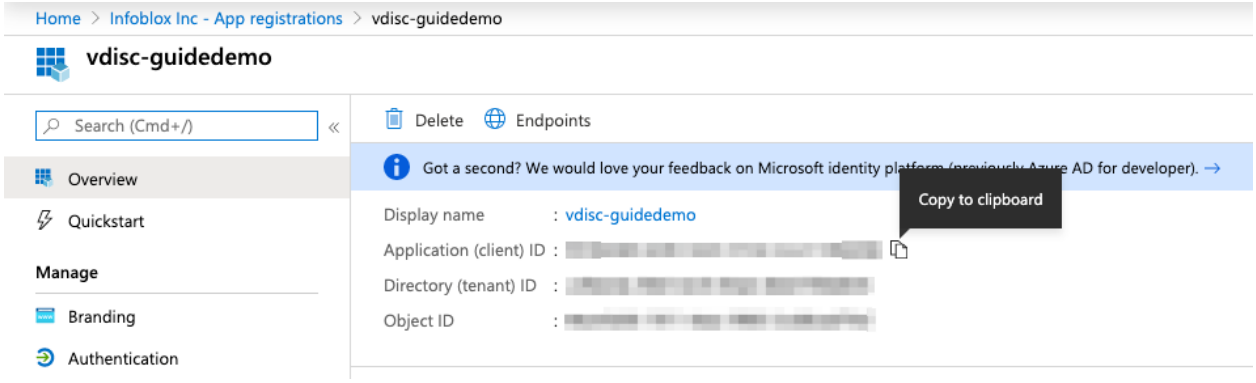
3. Click on **App registrations**.
4. Click **New registration**.



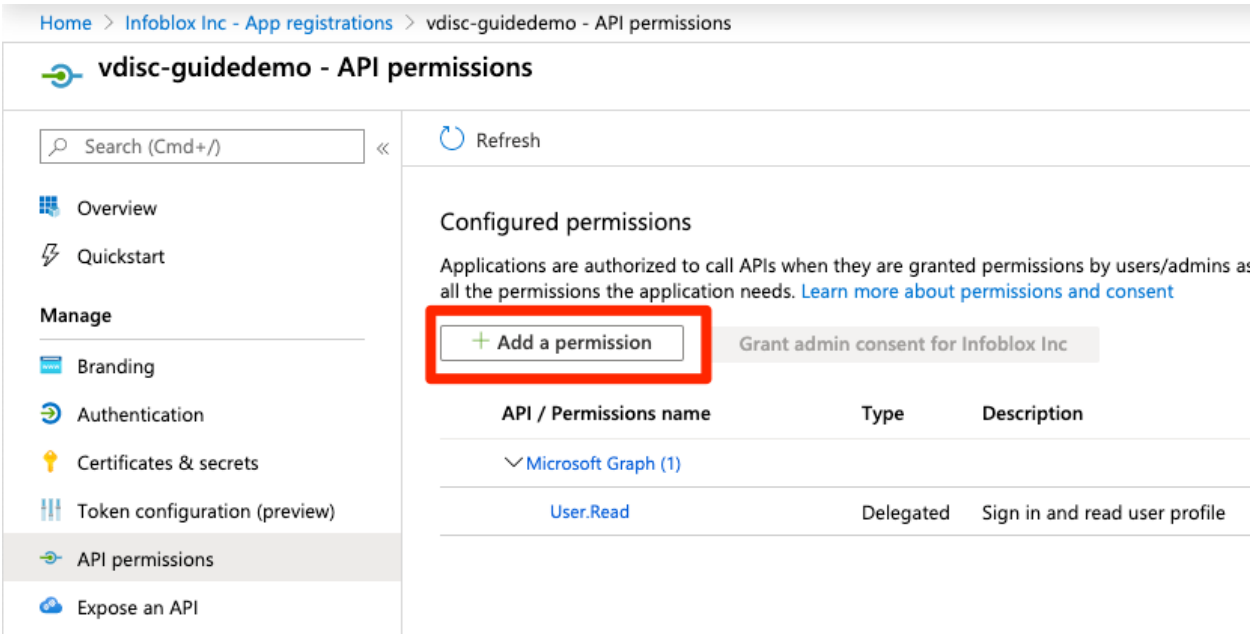
5. Type a name for your App.
6. Ensure **Accounts in this organizational directory only** is selected under **Supported account types**.
7. Enter a URI under **Redirect URI** (not currently used for vDiscovery, can be URL of your Grid Manager).



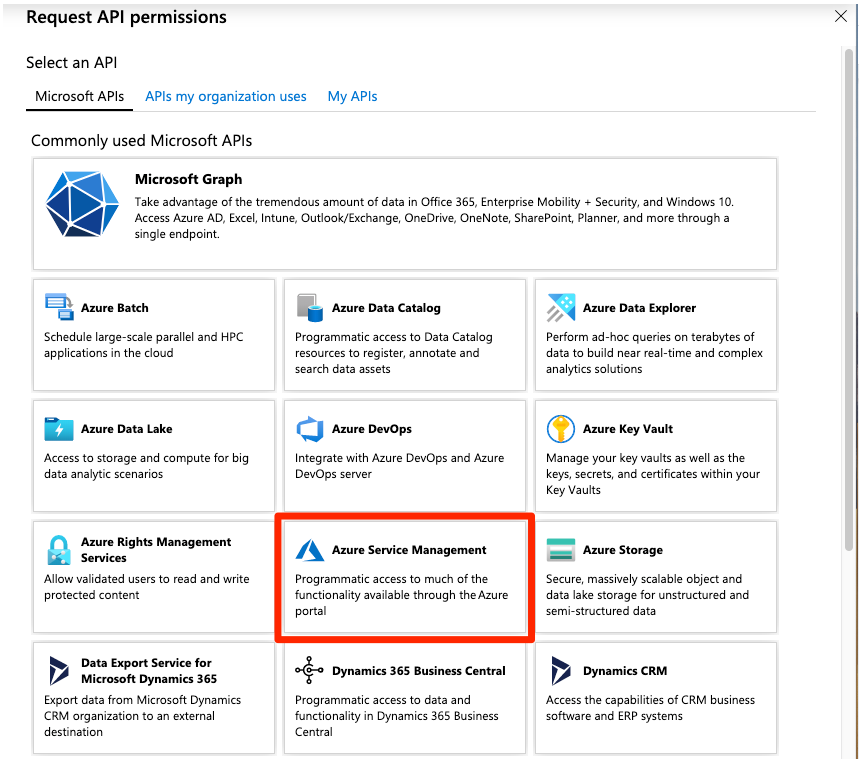
8. Click **Register**.
9. On the App's overview page, hover over **Application (client) ID**.
10. Click to copy the value to the clipboard. Save this ID.



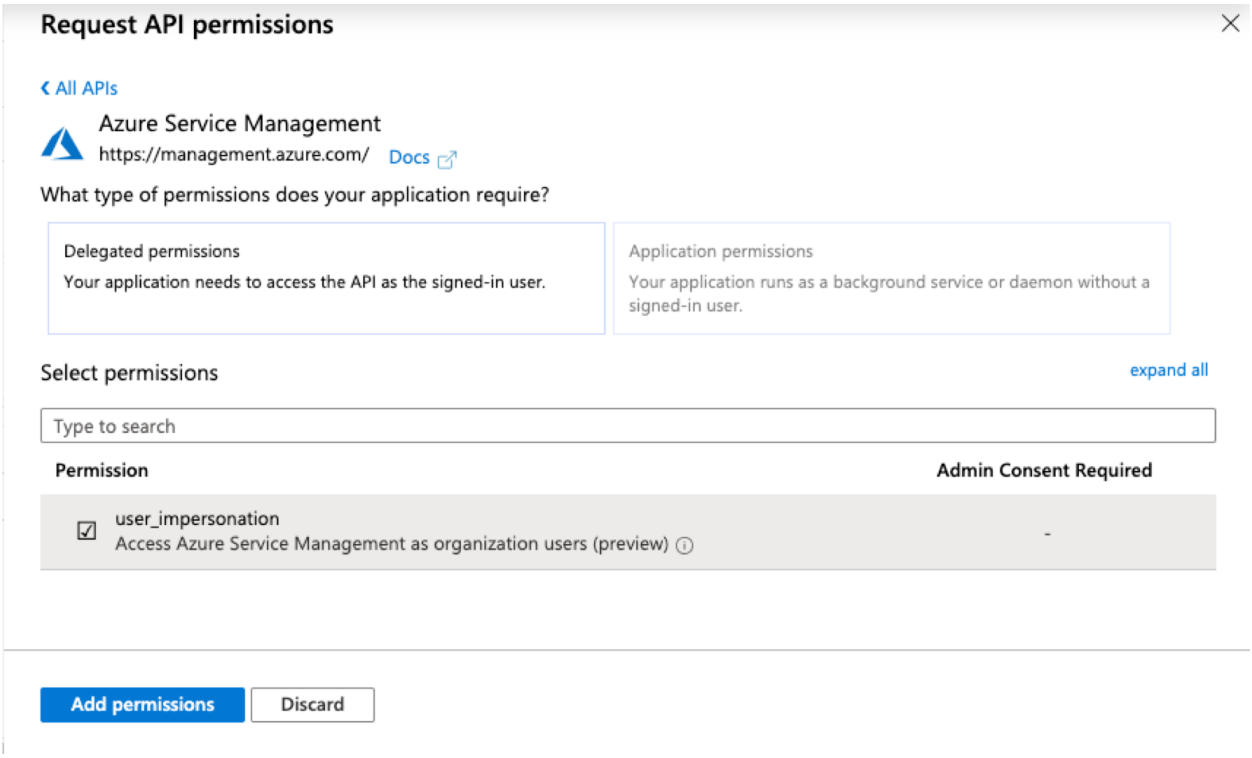
11. Click on **API permissions**.
12. Click on **Add a permission**.



13. Select the **Azure Service Management** API.



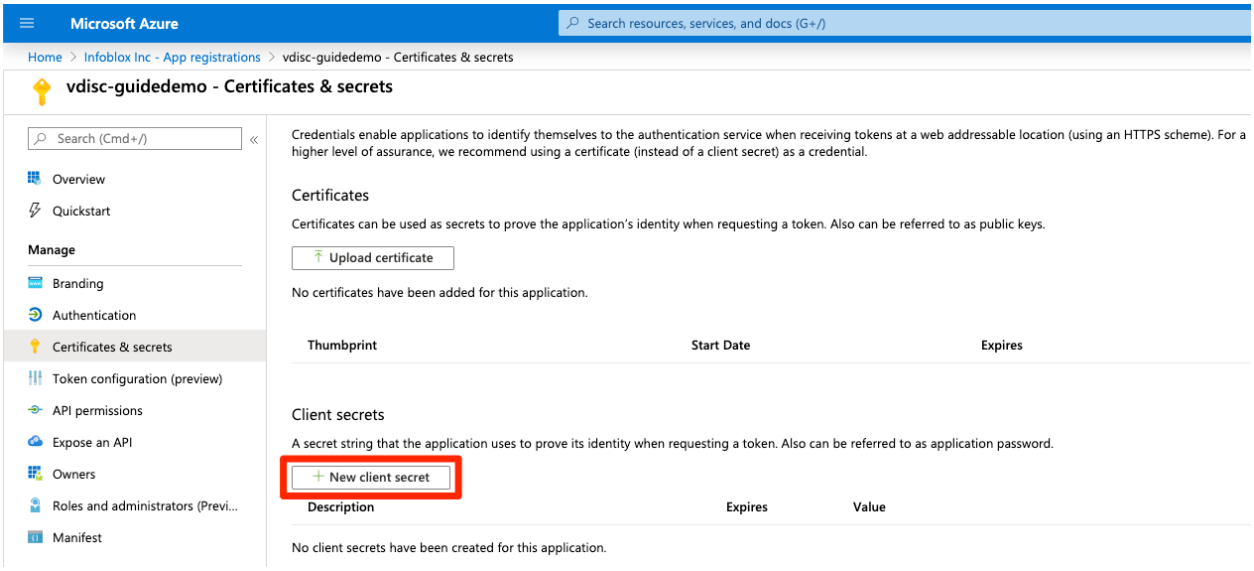
14. Select the checkbox for user_impersonation.



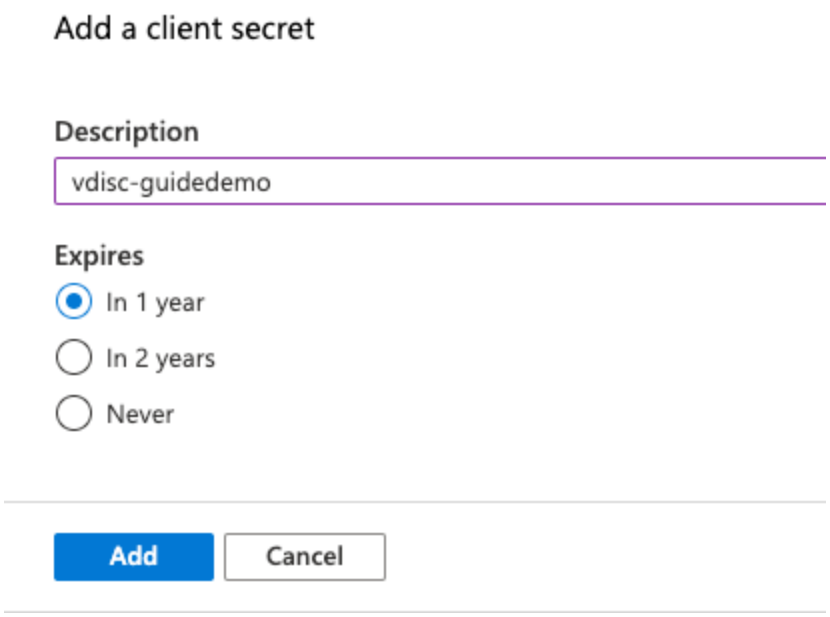
15. Click **Add permissions**.

Client Secret and Endpoint

1. Click on **Certificates & secrets**.
2. Click **New client secret**.

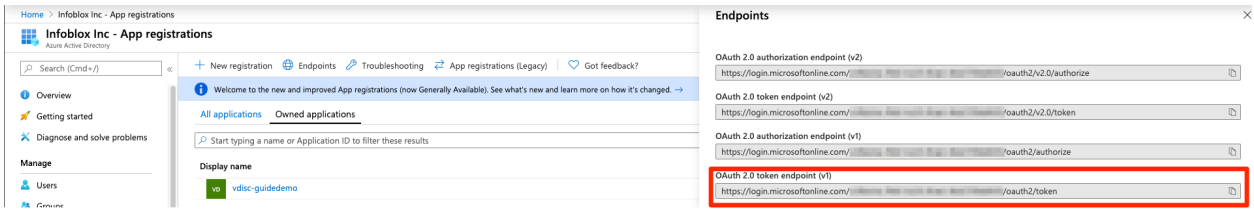


3. Type a description and click **Add**.



4. Hover over the key Value and click to copy the value to the clipboard. Save this Client Secret.

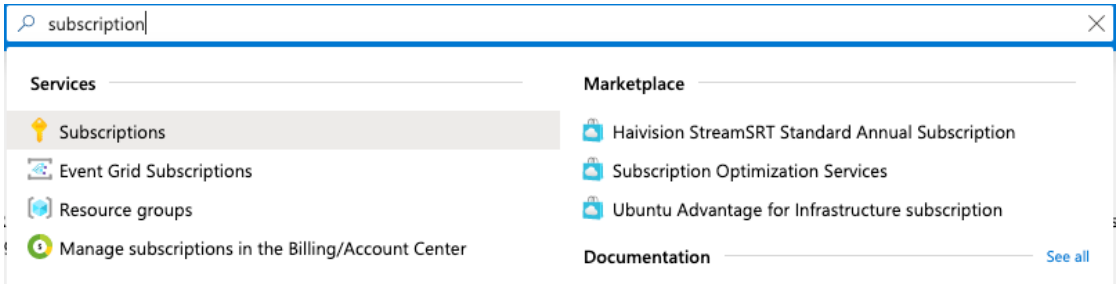
5. Navigate back to **Azure Active Directory → App registrations**.
6. Click on **Endpoints.**
7. Hover over the **OAuth 2.0 token endpoint (v1)** and click to copy the value to the clipboard. Save this Endpoint.



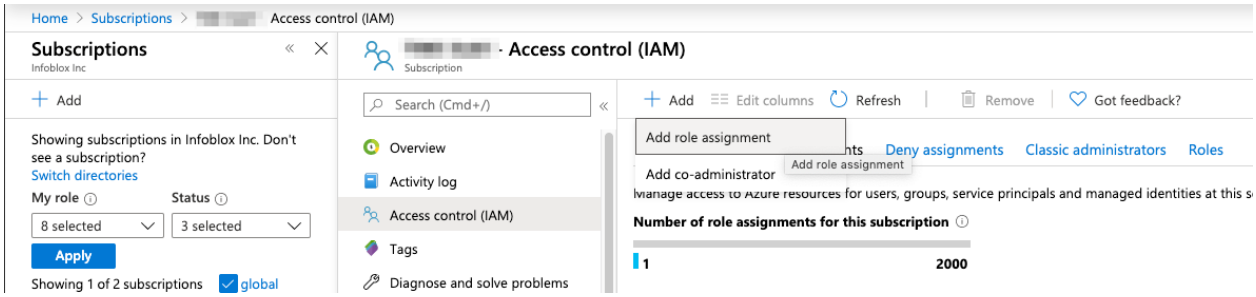
Set Permissions in Azure Subscriptions

For each Azure subscription where vDiscovery will be conducted, the new App needs to be added as a Reader. Alternatively, Reader permissions can be assigned at the Resource Group level for more granular control of what is included for vDiscovery.

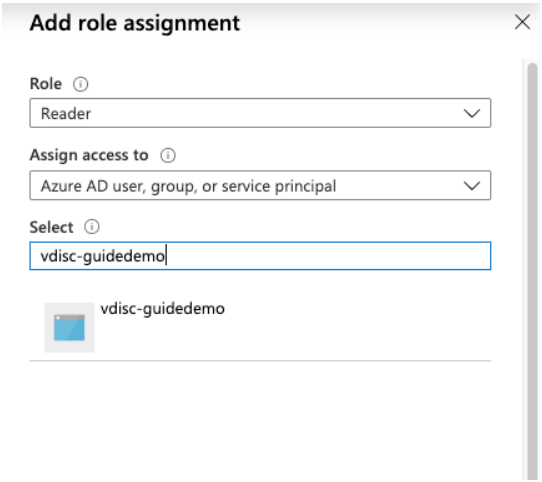
1. From the Azure Portal, type **subscription** in the the search box.
2. Click on **Subscriptions**.



3. Select your subscription from the list.
4. From the Subscription blade, select **Access control (IAM)**.
5. Click on **Add,** select **Add role assignment** from the dropdown.



6. Select **Reader** from the Role dropdown.
7. Type the name of your App in the Select box.



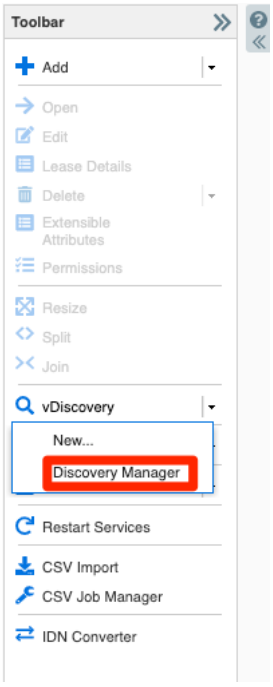
8. Click on your App.
9. Click **Save**.

Configure vDiscovery in Grid Manager

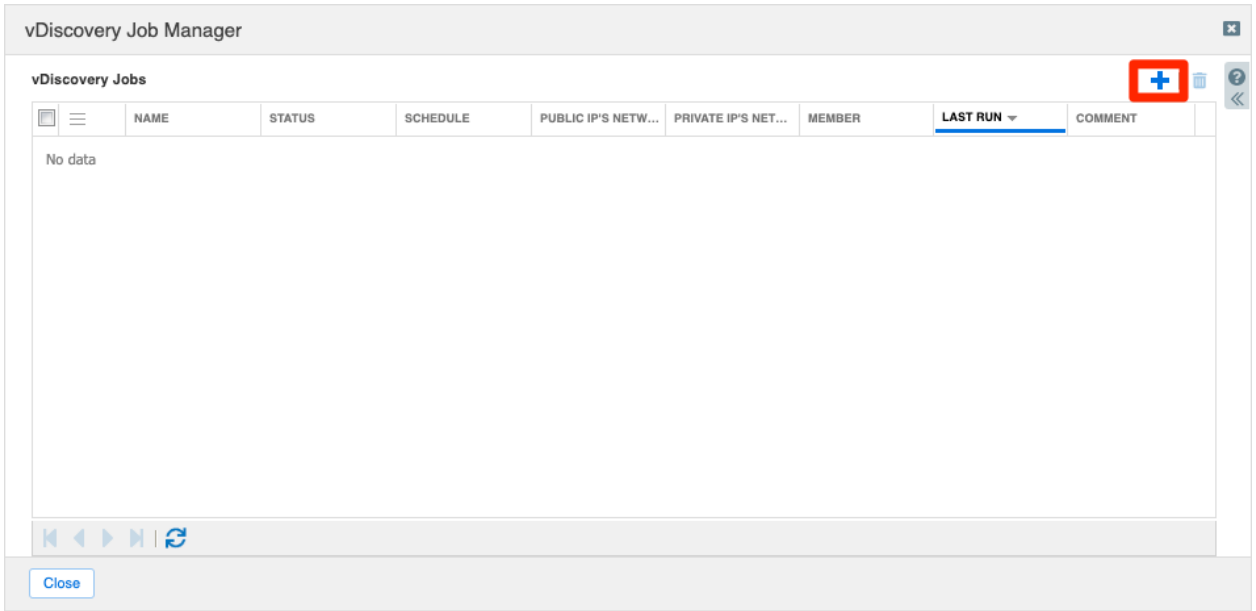
To conduct vDiscovery in Azure, you must configure a discovery job, using the Client ID, Client Secret, and Endpoint identified in Azure.

1. Log back into the Grid Manager.

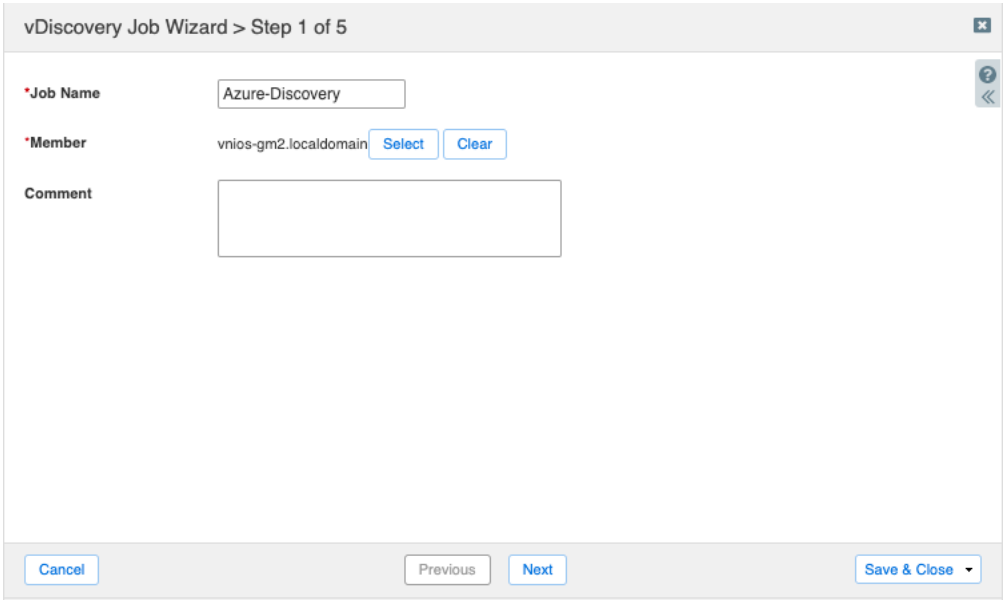
2. Navigate to the **Data Management** → **IPAM** tab.
3. In the **Toolbar**, Open the **vDiscovery** dropdown.
4. Select **Discovery Manager**.



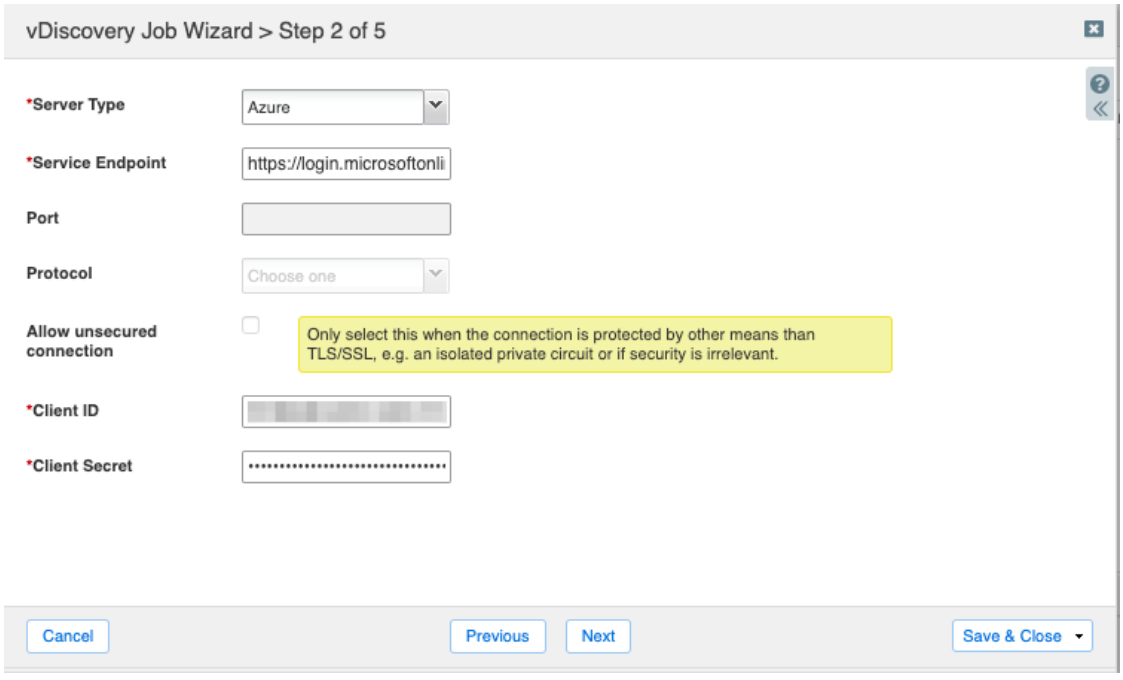
5. In the vDiscovery Job Manager window, click (Add) to add a new job.



6. In the vDiscovery Job Wizard, enter a name for the job.
7. Next to Member, click **Select**.
8. Select a grid member to use for this vDiscovery job.

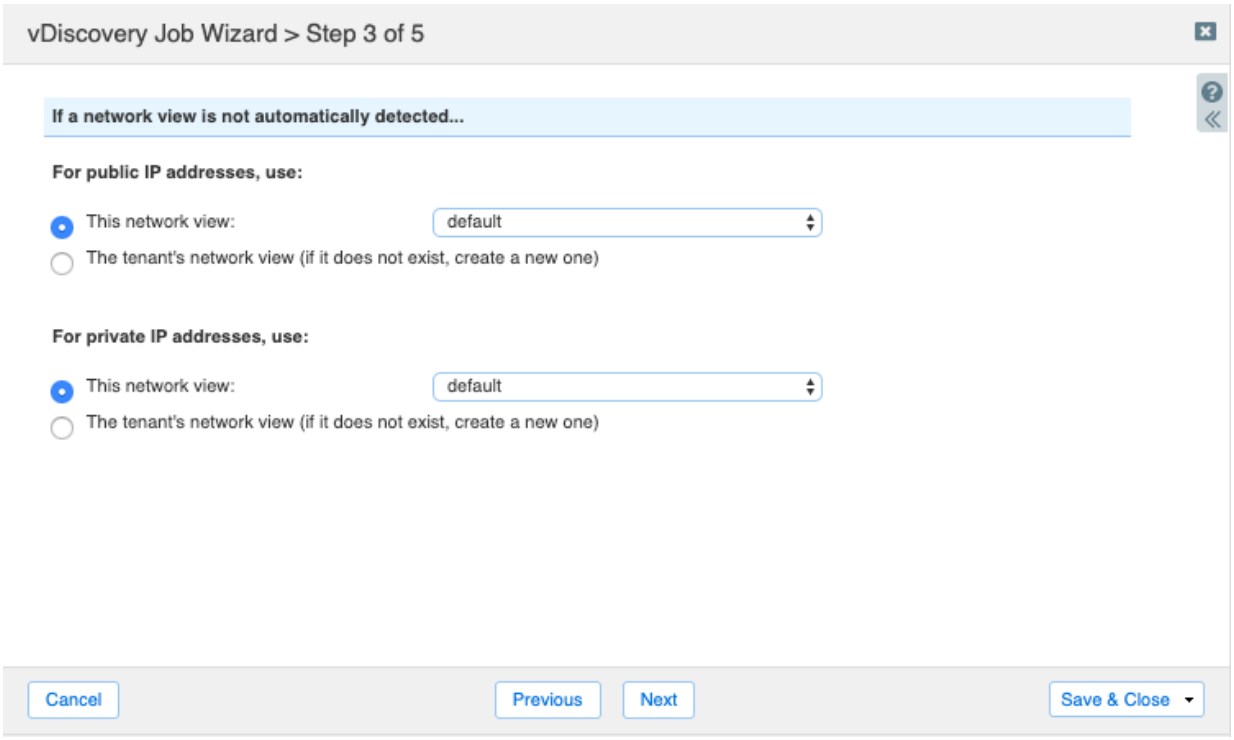


9. Click **Next**.
10. On Step 2, select **Azure** for **Server Type**.
11. For **Service Endpoint**, enter the OAuth 2.0 token endpoint (v1) that you saved earlier.
12. Enter the **Client ID** and **Client Secret** from your App registration.



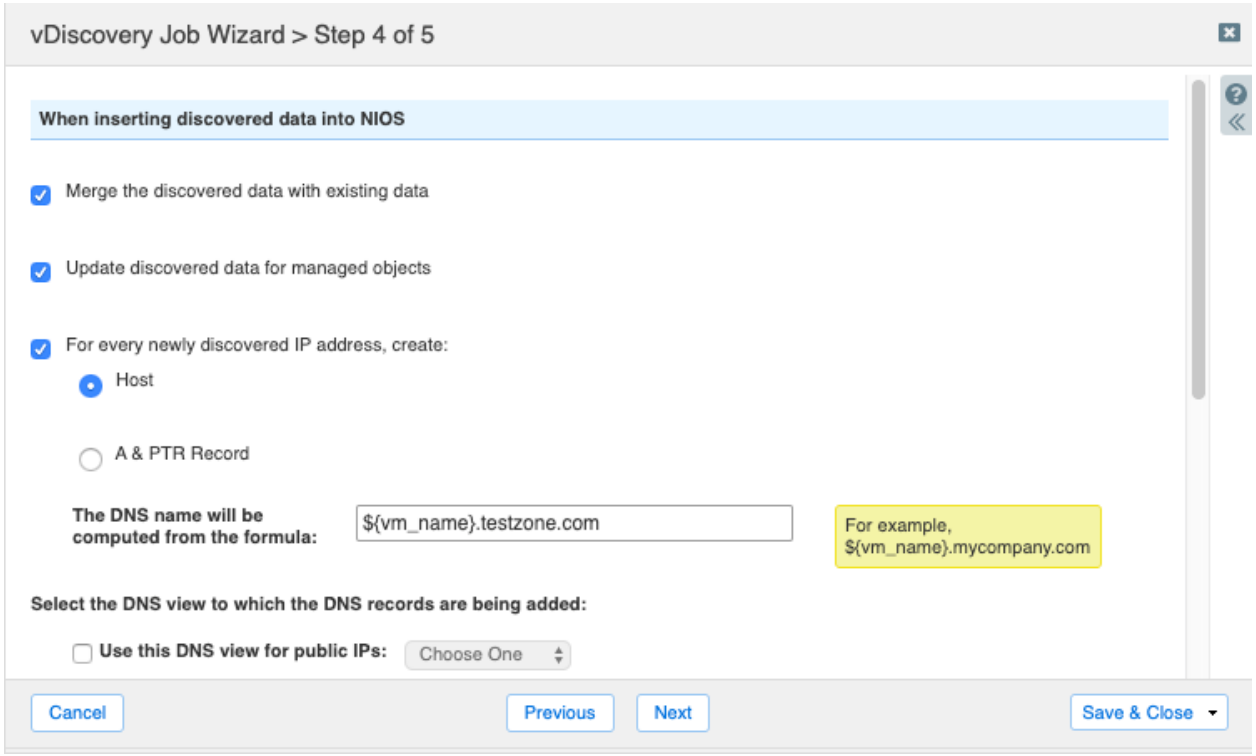
13. Click **Next**.

14. Review the configuration for Network Views on Step 3.



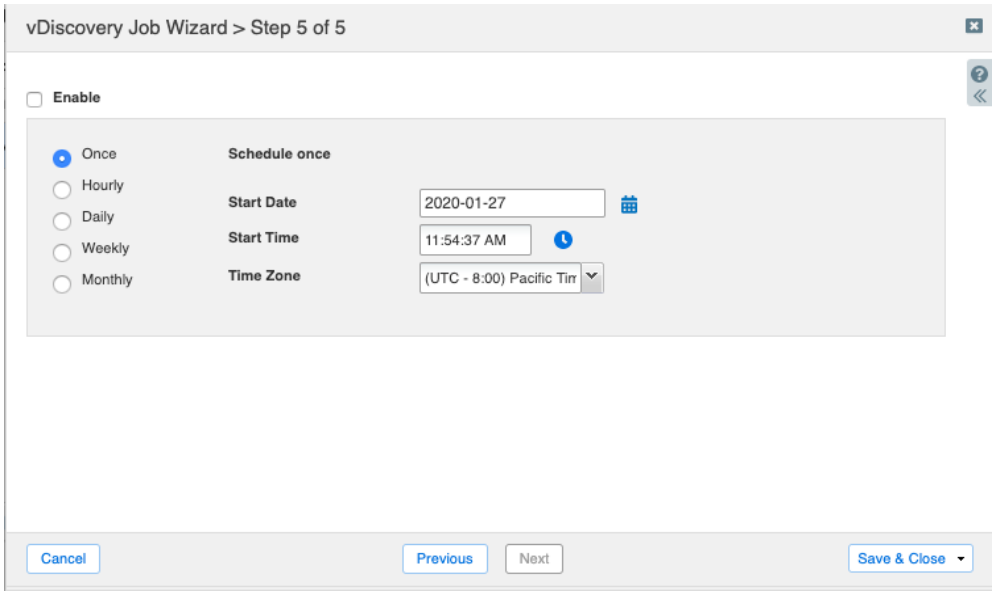
Note: The most common cause for vDiscovery to fail to import any data is a "Sync Error" due to overlapping/conflicting address space. To account for any address space conflicts that are encountered during the vDiscovery process or with your existing IPAM data, you may need to select the option to use ***The tenant's network view (if it does not exist, create a new one)****.*

15. Click **Next**.
16. Optional: For automatic creation of DNS records, on step 4 select the checkbox **For every newly discovered IP address, create:**.
    a. Select the desired DNS record object type. If in doubt, stick with the default (Host) option.
    b. The name for DNS records that are created is controlled with a macro, with the most commonly used macro being ${vm_name). In the text box, type the desired macro, followed by the zone that you want to use. Example: **${vm_name}.testzone.com**.



Note: If a different format is desired for the DNS record name, a full list of available macros can be found in the Help panel. To view this, click on  *(Help) at the top-right hand corner of the window and scroll down to the section titled "**The DNS name will be computed from the formula**".*

17. Click **Next**.
18. Optional: Configure a schedule to automatically run the vDiscovery task.



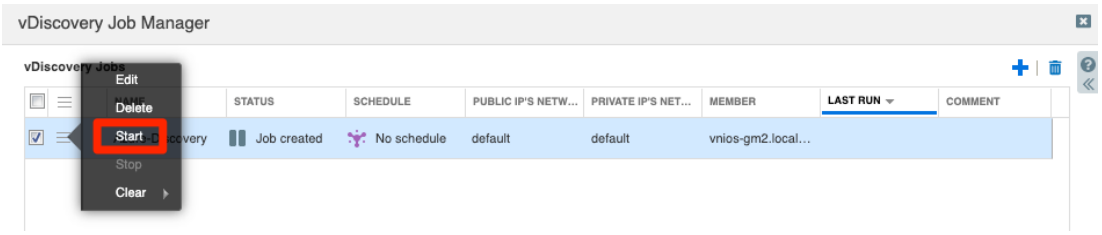
Note: The scheduler enables you to run the vDiscovery task as frequently as once an hour. If this must be run more frequently, this can be accomplished using the API. Refer to the Infoblox REST API guide for examples and guidelines on this process.

19. Click **Save & Close**.

Run vDiscovery

To run your vDiscovery job, from the vDiscovery Job Manager window click the (Action Menu) for your vDiscovery job.

Select **Start**.



Click **Yes** in the warning window.

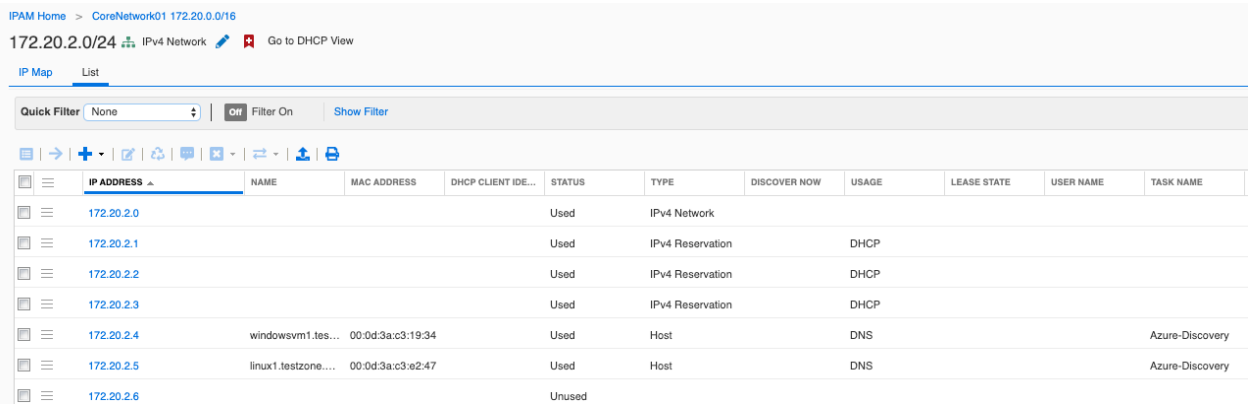
vDiscovery Data

Data collected by vDiscovery can be tracked through Data Management (IPAM, DHCP and DNS) and if the CNA license is installed, additional details will be found under the Cloud tab. Objects created by vDiscovery will automatically include metadata in their properties or extensible attributes (EA's), a useful addition that enables you to easily identify, locate and report on your resources deployed in the cloud.

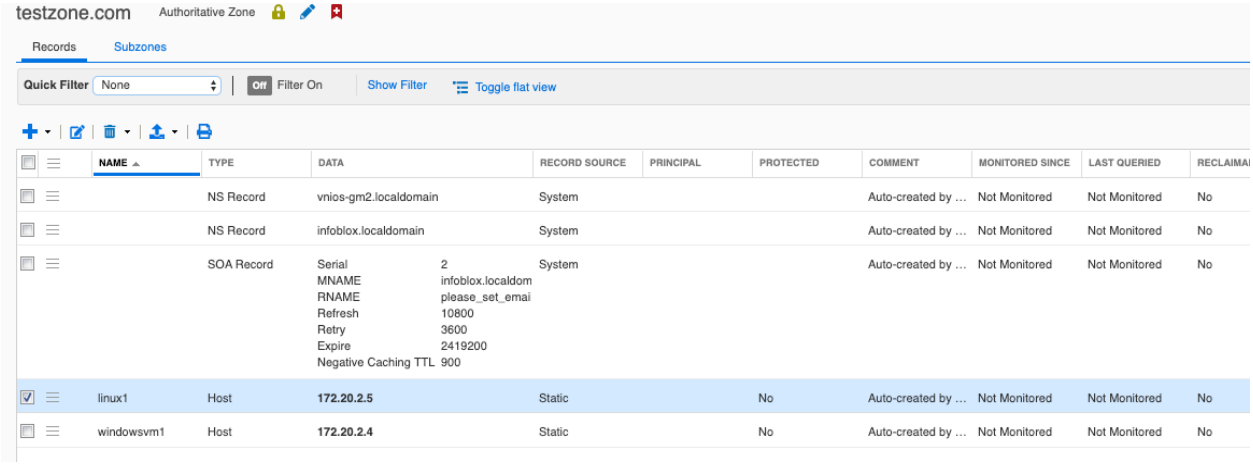
Data Management

From the Data Management tab, you can access IPAM and DNS data discovered from your Azure environment.

- **IPAM:** IPAM, or IP Address Management, provides an easy view of all data from an IP address perspective. If you are looking for an object based on its IP address, this can be one of the easiest ways to drill down and see everything there is for that IP, including all objects that are associated with it.



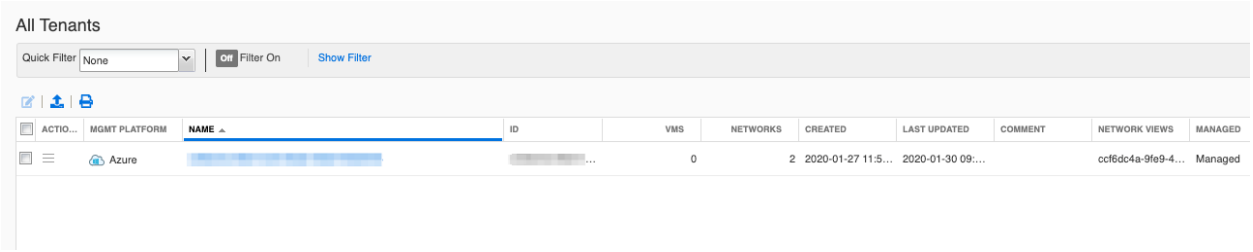
- **DNS:** If you enabled the automatic creation of DNS records, the records can be viewed by drilling down into the DNS zone you specified.



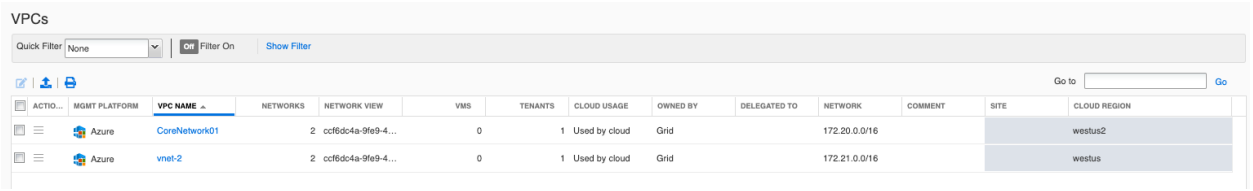
Cloud Network Automation

When the CNA license is installed, you will find the Cloud tab in your Grid Manager GUI. The Cloud tab includes four additional tabs that each provide different perspectives for viewing your cloud data, making it easy to see what is running in your cloud environments.

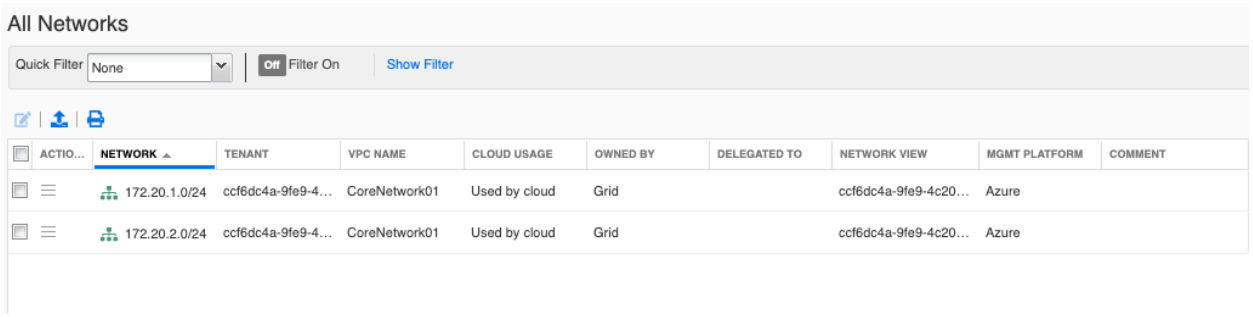
- **Tenants:** For Azure vDiscovery, entries on this tab correspond to AAD tenants. You can drill down to review all subnets and VMs that have been discovered under that tenant.



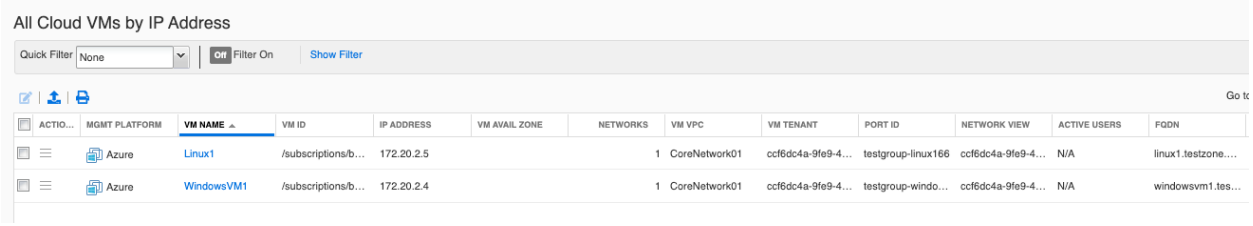
- **VPCs:** This tab displays any discovered Azure VNETs. You can drill down to review all subnets and VMs that have been discovered under an individual VNET.



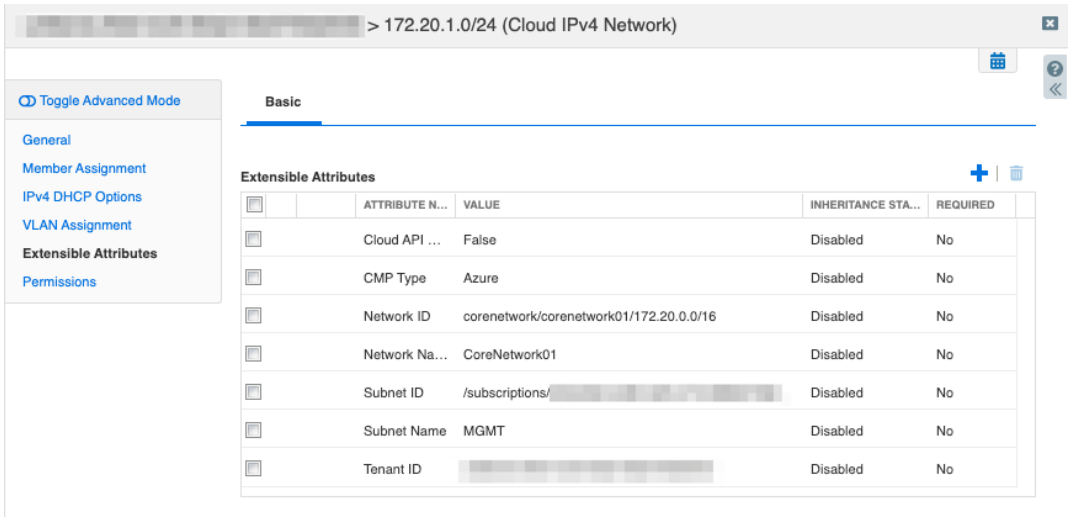
- **Networks:** This tab displays all subnets that have been discovered in your Azure VNETs. Easily jump to IPAM or other perspectives to view additional details for a subnet. Searches, Smart Folders and reports can also leverage the metadata stored as EAs for each subnet.



- **VMs:** This tab shows all VMs that have been discovered and are displayed per IP address. Metadata is stored in the properties for each VM, and you can readily jump to other perspectives to view and manage additional resources, including any DNS records that may have been created for the VM.



Metadata collected for each type of object discovered varies and is stored as Extensible Attributes in the Infoblox grid. The following is an example of EAs for a Subnet.



Infoblox is the leader in modern, cloud-first networking and security services. Through extensive integrations, its solutions empower organizations to realize the full advantages of cloud networking today, while maximizing their existing infrastructure investments. Infoblox has over 12,000 customers, including 70 percent of the Fortune 500.

Corporate Headquarters | 2390 Mission College Boulevard, Ste. 501 | Santa Clara, CA | 95054
+1.408.986.4000 | info@infoblox.com | www.infoblox.com