Deploy a Kubernetes cluster to a custom virtual network on Azure Stack Hub

mattbriggs

# Deploy a Kubernetes cluster to a custom virtual network on Azure Stack Hub

You can deploy a Kubernetes cluster using the Azure Kubernetes Service (AKS) engine on a custom virtual network. This article looks at finding the information you need in your virtual network. You can find steps for calculating the IP addresses used by your cluster, setting the vales in the API Model, and setting the route table and network security group.

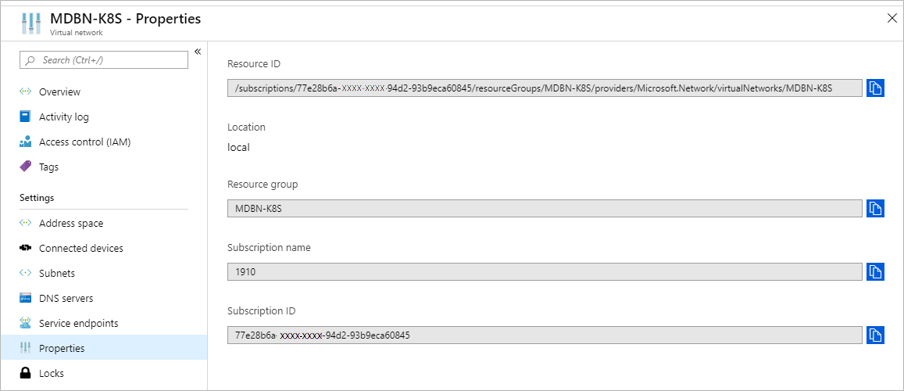
For a discussion of kubenet networking in Azure, see [Use kubenet networking with your own IP address ranges in Azure Kubernetes Service (AKS)](https://docs.microsoft.com/azure/aks/configure-kubenet).

## Create custom virtual network

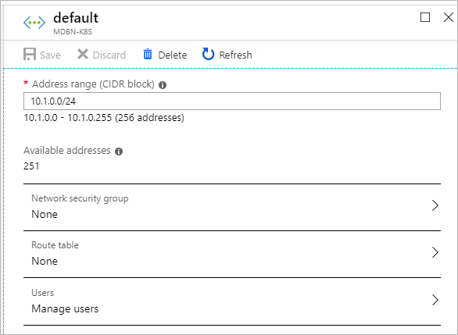
You must have a custom virtual network in your Azure Stack instance. For more information, see [Quickstart: Create a virtual network using the Azure portal](https://docs.microsoft.com/azure/virtual-network/quick-create-portal).

You will need to the get the Resource path ID of the subnet and the IP address range in your subnet. You will use the Resource path ID and range in your API model when you deploy your cluster.

1. Open the Azure Stack Hub user portal in your Azure Stack Hub instance.
2. Select **All resources**.
3. Enter the name of your virtual network in the search box.
4. Select **Properties** in the **Virtual networks** blade. Copy the **Resource ID**, and then add /subnets/<nameofyoursubnect>. You will use this value as your value for the vnetSubnetId key in the API model for your cluster. The Resource path ID for the subnet uses the following format:/subscriptions/SUB\_ID/resourceGroups/RG\_NAME/providers/Microsoft.Network/virtualNetworks/VNET\_NAME/subnets/SUBNET\_NAME

* 
* virtual network id

1. Select **Subnets** in the **Virtual networks** blade. Select the subnet name, for example default.
2. In the subnet blade, make a note of the address range and the virtual network CIDR Block, for example: 10.1.0.0 - 10.1.0.255 (256 addresses) and 10.1.0.0/24.



virtual network CIDR block

## Get the IP address block

The AKS engine supports deploying into an existing virtual network. When deploying into an existing subnet, your cluster will consume a block of IP addresses. You will need to set your cluster to avoid assignment collision with other resources using IP addresses and the resources needed by your cluster.

You will need to set two values. You will need to know the number of IP addresses you will need to reserve for your cluster, and first Consecutive Static IP within the subnet IP space.

Your Kubernetes cluster will take up a block of IP addresses. The following elements of the cluster consume IP addresses: - Each pod requires an IP address in the subnet. - Each master requires an IP address. - Each node requires an IP address - If more than one master: - One for the API server load balancer. - One for each master up to five masters. - 10 following the last master for headroom reservation.

The subnet requires the following allocations of the existing IP addresses: - The first IP address to use for static IP allocation. - The first four and the last IP address is reserved and can't be used in any Azure subnet - Leave a buffer of 16 IP addresses. - The value of the first IP should be toward the end of the address space to avoid IP conflicts.

ipAddressCount has a default count of 31. This is one for the node and 30 for the pods. If you are using a subnet with 256 addresses, for example 10.1.0.0/24, you will need to set your first consecutive static IP address at 207. The following table shows the addresses and considerations:

|  |  |  |
| --- | --- | --- |
| Range for /24 subnet | Number | Note |
| 10.1.0.0 - 10.1.03 | 4 | Reserved in Azure subnet. |
| 10.1.0.207-10.1.0.238 | 31 | Default IP address count for an AKS engine defined cluster. |
| 10.1.0.239 - 10.1.0.255 | 16 | 16 IP address buffer. |
| 10.1.0.256 | 1 | Reserved in Azure subnet. |

For larger subnets, for example /16 with more than 60 thousand addresses, you may not find it to be practical to set your static IP assignments to the end of the network space. Set your cluster static IP address range away from the first 24 addresses in your IP space so that the cluster can be resilient when claiming addresses.

## Update the API model

Update the API model used to deploy the cluster from your AKS Client machine to your custom virtual network.

In the array **masterProfile** set the following values:

|  |  |  |
| --- | --- | --- |
| Field | Example | Description |
| vnetSubnetId | /subscriptions/77e28b6a-582f-42b0-94d2-93b9eca60845/resourceGroups/MDBN-K8S/providers/Microsoft.Network/virtualNetworks/MDBN-K8S/subnets/default | Specify the Azure Resource Manager path ID the subnet. |
| firstConsecutiveStaticIP | 10.1.0.207 | Assign to the firstConsecutiveStaticIP configuration property an IP address that is near the *end* of the available IP address space in the desired subnet. |
| vnetCidr | 10.1.0.0/24 | The subnet's address range in CIDR notation (for example, 192.168.1.0/24). It must be contained by the address space of the virtual network. The address range of a subnet, which is in use can't be edited. |

In the array **agentPoolProfiles** set the following values:

|  |  |  |
| --- | --- | --- |
| Field | Example | Description |
| vnetSubnetId | /subscriptions/77e28b6a-582f-42b0-94d2-93b9eca60845/resourceGroups/MDBN-K8S/providers/Microsoft.Network/virtualNetworks/MDBN-K8S/subnets/default | Specify the Azure Resource Manager path ID the subnet. |

For example:

"masterProfile": {  
 ...  
 "vnetSubnetId": "/subscriptions/77e28b6a-582f-42b0-94d2-93b9eca60845/resourceGroups/MDBN-K8S/providers/Microsoft.Network/virtualNetworks/MDBN-K8S/subnets/default",  
 "firstConsecutiveStaticIP": "10.1.0.207",   
 "vnetCidr": "10.1.0.0/24",  
 ...  
},  
...  
"agentPoolProfiles": [  
 {  
 ...  
 "vnetSubnetId": "/subscriptions/77e28b6a-582f-42b0-94d2-93b9eca60845/resourceGroups/MDBN-K8S/providers/Microsoft.Network/virtualNetworks/MDBN-K8S/subnets/default",  
 ...  
 },

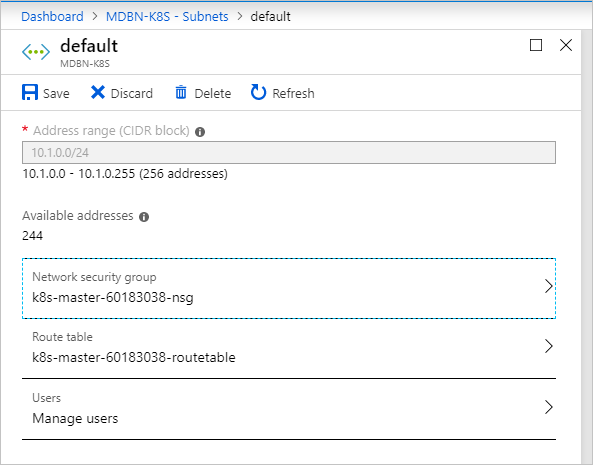
## Deploy your cluster

After adding the values to your API model, you can deploy your cluster from your client machine using the deploy command with aks-engine. For instructions, see [Deploy a Kubernetes cluster](azure-stack-kubernetes-aks-engine-deploy-cluster.md#deploy-a-kubernetes-cluster).

## Set the route table and network security group

Set both the route table and the network security group (NSG) in the subnet blade in the Azure Stack user portal. If you're not using Azure CNI (for example, networkPlugin: kubenet in the kubernetesConfig API model configuration object. After you have successfully deployed a cluster to your custom virtual network, get the ID of the Route Table resource from **Network** blade in your cluster's resource group.

1. Open the Azure Stack Hub user portal in your Azure Stack Hub instance.
2. Select **All resources**.
3. Enter the name of your virtual network in the search box.
4. Select **Subnets** and then select the name of the subnet that contains your cluster.

* 
* route table and network security group

1. Select **Route table** and then select the route table for your cluster.
2. Select Network security group and then select the NSG for your cluster.

[!Note]  
Custom virtual network for Kubernetes Windows cluster has a [known issue](https://github.com/Azure/aks-engine/issues/371).

## Next steps

* Read about the [The AKS engine on Azure Stack Hub](azure-stack-kubernetes-aks-engine-overview.md)
* Read about [Azure Monitor for containers overview](https://docs.microsoft.com/azure/azure-monitor/insights/container-insights-overview)