```
        - mountPath: "/mnt/azure"
          name: volume
  volumes:
    - name: volume
      persistentVolumeClaim:
        claimName: azure-managed-disk
```

We have briefly discussed the storage options available for AKS workloads. Next step is to create dynamic and static volumes for AKS. The following articles from the Microsoft documentation provide a holistic overview on how to do so:

- Create a static volume using Azure Disks (https://docs.microsoft.com/en-au/azure/aks/azure-disk-volume).
- Create a static volume using Azure Files (https://docs.microsoft.com/en-au/azure/aks/azure-files-volume).
- Create a dynamic volume using Azure Disks (https://docs.microsoft.com/en-au/azure/aks/azure-disks-dynamic-pv).
- Create a dynamic volume using Azure Files (https://docs.microsoft.com/en-au/azure/aks/azure-files-dynamic-pv).

Networking in Azure Kubernetes Service

Application components in a microservices approach must work together to process their desired tasks. This application communication can be achieved using a few components provided by Kubernetes. For an example, the applications can be either exposed internally or externally, can be load balanced for high availability, and have SSL.TLS termination for ingress traffic as well as for routing of multiple components. Furthermore, developers may need you to restrict the flow of network traffic into or between pods and nodes due to security concerns.

In this section, we will dive into core networking concepts of AKS and some of the examples of providing secure network connectivity to your pods and nodes.

Kubenet vs. Azure Container Networking Interface (CNI)

An AKS cluster uses one of the following two networking models:

Kubenet (Basic) Networking

This is the default configuration option for an AKS cluster. In kubenet, the AKS nodes obtain an IP address from the Azure VNet subnet. Pods receive an IP address from a logically different address space to the Azure VNet subnet of the nodes. For the pods to reach resources on the Azure VNet, network address translation (NAT) is then configured. The source IP address of the traffic is NAT'd to the node's primary IP address.

Nodes use the kubenet Kubernetes plugin. You can either allow Azure Fabric to create and configure the VNets for you or deploy your AKS cluster into an existing subnet of a predefined VNet. Even though you are deploying to a predefined VNet, only the nodes will receive a routable IP address; pods use NAT to communicate with other resources external to the AKS cluster.

Azure Container Networking Interface (CNI) - Adavanced Networking

Each pod gets an IP address from the subnet and can be accessed directly if you are using the Azure CNI model. But remember that these IP addresses must be unique across the VNet network space and must be planned in well advance. There is a configuration parameter for the maximum number of pods that each node supports. An equivalent number of IP addresses per node are then reserved for that node.

The following table lists the behavioral differences between kubenet and Azure CNI.

Table 7-1. *Behavioral Differences Between Kubenet and Azure CNI*

Capability	Kubenet	Azure CNI
Deploy cluster in existing or new virtual network	Supported – UDRs manually applied	Supported
Pod-pod connectivity	Supported	Supported
Pod-VM connectivity; VM in the same virtual network	Works when initiated by pod	Works both ways
Pod-VM connectivity; VM in peered virtual network	Works when initiated by pod	Works both ways
On-premises access using VPN or Express Route	Works when initiated by pod	Works both ways

(*continued*)

Table 7-1. (*continued*)

Capability	Kubenet	Azure CNI
Access to resources secured by service endpoints	Supported	Supported
Expose Kubernetes services using a load balancer service, App Gateway, or ingress controller	Supported	Supported
Default Azure DNS and Private Zones	Supported	Supported

Table 7-2 lists the advantages and disadvantages of kubenet and Azure CNI at a high level.

Table 7-2. *Advantages and Disadvantages of Kubenet vs. Azure CNI*

Model	Advantages	Disadvantages
Kubenet	• Conserves IP address space. • Uses Kubernetes internal or external load balancer to reach pods from outside of the cluster.	• You must manually manage and maintain user-defined routes (UDRs). • Maximum of 400 nodes per cluster.
Azure CNI	Pods get full virtual network connectivity and can be directly reached from outside of the cluster.	Requires more IP address space.

Regardless of the network model you have selected, support policies for AKS depict the network tuning capabilities such as service endpoints and UDRs that you can make in your AKS clusters:

- If you manually create the virtual network resources for an AKS cluster, you are supported when configuring your own UDRs or service endpoints.
- If the Azure platform automatically creates the virtual network resources for your AKS cluster, it is not supported to manually change those AKS-managed resources to configure your own UDRs or service endpoints.

Note For a complete record of support policies for AKS, visit the following URL: https://docs.microsoft.com/en-au/azure/aks/support-policies.

Network Security Groups and Network Policies

It is not recommended to manually configure network security group rules to filter pod traffic in an AKS cluster. The Azure platform will create and update the appropriate rules as part of the AKS managed service. In order to automatically apply traffic filter rules to pods, you can utilize **Network Policies**. On the one hand, it is a feature available in AKS that allows you to control the traffic between pods. You can decide whether to allow or deny traffic based on settings such as assigned labels, namespace, or traffic port. Network security groups on the other hand are for the AKS nodes, not pods.

Note For step-by-step instructions on securing pod traffic using Azure Network Policies in AKS, visit the following URL: https://docs.microsoft.com/en-au/azure/aks/use-network-policies.

Access and Identity in Azure Kubernetes Service

In Azure, there are multiple methods to authenticate and secure AKS clusters. Role-based access controls (RBACs) allow granting users or groups access to only the resources they need. By integrating AKS with Azure Active Directory, you are able to further enhance the security and permissions structure. This section provides a high-level overview of the access and identity options available to you when operating an AKS cluster.

Kubernetes Service Accounts

A service account is a primary user type in Kubernetes, and it exists in and is managed by the Kubernetes API. The service account credentials are stored as Kubernetes secrets, which allows them to be used by authorized pods to communicate with the API server. API requests provide an authentication token for a service account or a regular user account. Regular user accounts are leveraged to provide traditional access to