Tutorial: Upgrade Kubernetes in Azure Kubernetes Service (AKS)

Article • 06/09/2022 • 6 minutes to read • 17 contributors

In this article

As part of the application and cluster lifecycle, you may wish to upgrade to the latest available version of Kubernetes and use new features. An Azure Kubernetes Service (AKS) cluster can be upgraded using the Azure CLI.

In this tutorial, part seven of seven, a Kubernetes cluster is upgraded. You learn how to:

- ✓ Identify current and available Kubernetes versions
- ✓ Upgrade the Kubernetes nodes
- ✓ Validate a successful upgrade

Before you begin

In previous tutorials, an application was packaged into a container image. This image was uploaded to Azure Container Registry, and you created an AKS cluster. The application was then deployed to the AKS cluster. If you have not done these steps, and would like to follow along, start with Tutorial 1 – Create container images.

Azure CLI | Azure PowerShell

This tutorial requires that you are running the Azure CLI version 2.0.53 or later. Run `az --version` to find the version. If you need to install or upgrade, see Install Azure CLI.

Get available cluster versions

Azure CLI | Azure PowerShell

Before you upgrade a cluster, use the az aks get-upgrades command to check which Kubernetes releases are available for upgrade:

Azure CLI Copy

```
az aks get-upgrades --resource-group myResourceGroup --name myAKSCluster
```

In the following example, the current version is *1.18.10*, and the available versions are shown under *upgrades*.

JSON Copy

```
{
  "agentPoolProfiles": null,
  "controlPlaneProfile": {
    "kubernetesVersion": "1.18.10",
    ...
    "upgrades": [
      {
        "isPreview": null,
        "kubernetesVersion": "1.19.1"
      },
      {
        "isPreview": null,
        "kubernetesVersion": "1.19.3"
      }
    ]
  },
  ...
}
```

Upgrade a cluster

To minimize disruption to running applications, AKS nodes are carefully cordoned and drained. In this process, the following steps are performed:

1. The Kubernetes scheduler prevents additional pods being scheduled on a node that is to be upgraded.
2. Running pods on the node are scheduled on other nodes in the cluster.

3. A node is created that runs the latest Kubernetes components.
4. When the new node is ready and joined to the cluster, the Kubernetes scheduler begins to run pods on it.
5. The old node is deleted, and the next node in the cluster begins the cordon and drain process.

> **Note**
>
> If no patch is specified, the cluster will automatically be upgraded to the specified minor version's latest GA patch. For example, setting `--kubernetes-version` to `1.21` will result in the cluster upgrading to `1.21.9`.
>
> When upgrading by alias minor version, only a higher minor version is supported. For example, upgrading from `1.20.x` to `1.20` will not trigger an upgrade to the latest GA `1.20` patch, but upgrading to `1.21` will trigger an upgrade to the latest GA `1.21` patch.

Azure CLI | Azure PowerShell

Use the az aks upgrade command to upgrade the AKS cluster.

Azure CLI | Copy

```
az aks upgrade \
    --resource-group myResourceGroup \
    --name myAKSCluster \
    --kubernetes-version KUBERNETES_VERSION
```

> **Note**
>
> You can only upgrade one minor version at a time. For example, you can upgrade from *1.14.x* to *1.15.x*, but cannot upgrade from *1.14.x* to *1.16.x* directly. To upgrade from *1.14.x* to *1.16.x*, first upgrade from *1.14.x* to *1.15.x*, then perform another upgrade from *1.15.x* to *1.16.x*.

The following condensed example output shows the result of upgrading to *1.19.1*. Notice the *kubernetesVersion* now reports *1.19.1*:

JSON | Copy

```
{
  "agentPoolProfiles": [
```

```
    {
      "count": 3,
      "maxPods": 110,
      "name": "nodepool1",
      "osType": "Linux",
      "storageProfile": "ManagedDisks",
      "vmSize": "Standard_DS1_v2",
    }
  ],
  "dnsPrefix": "myAKSClust-myResourceGroup-19da35",
  "enableRbac": false,
  "fqdn": "myaksclust-myresourcegroup-19da35-
bd54a4be.hcp.eastus.azmk8s.io",
  "id": "/subscriptions/<Subscription
ID>/resourcegroups/myResourceGroup/providers/Microsoft.ContainerService/
managedClusters/myAKSCluster",
  "kubernetesVersion": "1.19.1",
  "location": "eastus",
  "name": "myAKSCluster",
  "type": "Microsoft.ContainerService/ManagedClusters"
}
```


View the upgrade events

When you upgrade your cluster, the following Kubenetes events may occur on each node:

- Surge – Create surge node.
- Drain – Pods are being evicted from the node. Each pod has a 5 minute timeout to complete the eviction.
- Update – Update of a node has succeeded or failed.
- Delete – Deleted a surge node.

Use `kubectl get events` to show events in the default namespaces while running an upgrade. For example:

Azure CLI | Copy | Try It

```
kubectl get events
```

The following example output shows some of the above events listed during an upgrade.

Output | Copy

```
...
default 2m1s Normal Drain node/aks-nodepool1-96663640-vmss000001 Draining
node: [aks-nodepool1-96663640-vmss000001]
...
default 9m22s Normal Surge node/aks-nodepool1-96663640-vmss000002 Created a
surge node [aks-nodepool1-96663640-vmss000002 nodepool1] for agentpool
%!s(MISSING)
...
```

Validate an upgrade

Azure CLI | Azure PowerShell

Confirm that the upgrade was successful using the az aks show command as follows:

Azure CLI Copy

```
az aks show --resource-group myResourceGroup --name myAKSCluster --out-
put table
```

The following example output shows the AKS cluster runs *KubernetesVersion 1.19.1*:

Output Copy

```
Name          Location    ResourceGroup    KubernetesVersion
ProvisioningState    Fqdn
------------  ----------  ---------------  ------------------  --------
-----------  ----------------------------------------------------------
-----
myAKSCluster  eastus      myResourceGroup  1.19.1
Succeeded            myaksclust-myresourcegroup-19da35-
bd54a4be.hcp.eastus.azmk8s.io
```

Delete the cluster

Azure CLI | Azure PowerShell

As this tutorial is the last part of the series, you may want to delete the AKS cluster. As the Kubernetes nodes run on Azure virtual machines (VMs), they continue to incur charges even if you don't use the cluster. Use the az group delete command to remove the resource group, container service, and all related resources.

Azure CLI | Copy | Try It

```
az group delete --name myResourceGroup --yes --no-wait
```

> **Note**
>
> When you delete the cluster, the Azure Active Directory service principal used by the AKS cluster is not removed. For steps on how to remove the service principal, see **AKS service principal considerations and deletion**. If you used a managed identity, the identity is managed by the platform and does not require you to provision or rotate any secrets.

Next steps

In this tutorial, you upgraded Kubernetes in an AKS cluster. You learned how to:

- ✓ Identify current and available Kubernetes versions
- ✓ Upgrade the Kubernetes nodes
- ✓ Validate a successful upgrade

For more information on AKS, see AKS overview. For guidance on a creating full solutions with AKS, see AKS solution guidance.

Recommended content

Kubernetes on Azure tutorial - Update an application - Azure Kubernetes Service

In this Azure Kubernetes Service (AKS) tutorial, you learn how to update an existing application deployment to AKS with a new version of the application code.

Kubernetes on Azure tutorial - Deploy an application - Azure Kubernetes Service

In this Azure Kubernetes Service (AKS) tutorial, you deploy a multi-container application to your cluster using a custom image stored in Azure Container Registry.

Kubernetes on Azure tutorial - Create a container registry - Azure Kubernetes Service

In this Azure Kubernetes Service (AKS) tutorial, you create an Azure Container Registry instance and upload a sample application container image.

Kubernetes on Azure tutorial - Scale Application - Azure Kubernetes Service

In this Azure Kubernetes Service (AKS) tutorial, you learn how to scale nodes and pods in Kubernetes, and implement horizontal pod autoscaling.

Kubernetes on Azure tutorial - Prepare an application - Azure Kubernetes Service

In this Azure Kubernetes Service (AKS) tutorial, you learn how to prepare and build a multi-container app with Docker Compose that you can then deploy to AKS.

Kubernetes on Azure tutorial - Deploy a cluster - Azure Kubernetes Service

In this Azure Kubernetes Service (AKS) tutorial, you create an AKS cluster and use kubectl to connect to the Kubernetes master node.

Tutorial - Configure Azure Kubernetes Service (AKS) clusters in Azure using Ansible

Learn how to use Ansible to create and manage an Azure Kubernetes Service cluster in Azure

Automatically repairing Azure Kubernetes Service (AKS) nodes - Azure Kubernetes Service

Learn about node auto-repair functionality, and how AKS fixes broken worker nodes.

Show more