Enable AKS-managed Microsoft Entra integration on an Azure Kubernetes Service cluster - Azure Kubernetes Service

By davidsmatlak

Archived: 2026-04-05 23:48:18 UTC

Enable AKS-managed Microsoft Entra integration for Kubernetes clusters with kubelogin

The AKS-managed Microsoft Entra integration simplifies the Microsoft Entra integration process. Previously, you were required to create a client and server app, and the Microsoft Entra tenant had to assign Directory Readers role permissions. Now, the Azure Kubernetes Service (AKS) resource provider manages the client and server apps for you.

Cluster administrators can configure Kubernetes role-based access control (Kubernetes RBAC) based on a user's identity or directory group membership. Microsoft Entra authentication is provided to AKS clusters with OpenID Connect. OpenID Connect is an identity layer built on top of the OAuth 2.0 protocol. For more information on OpenID Connect, see the OpenID Connect documentation.

Learn more about the Microsoft Entra integration flow in the Microsoft Entra documentation.

The following are constraints to integrate authentication on AKS:

- Integration can't be disabled after being added.
- Downgrades from an integrated cluster to the legacy Microsoft Entra ID clusters aren't supported.
- Clusters without Kubernetes RBAC support are unable to add the integration.

To install the AKS addon, verify you have the following items:

- You have Azure CLI version 2.29.0 or later installed and configured. To find the version, run the `az --version` command. If you need to install or upgrade, see Install Azure CLI.
- You need `kubectl` with a minimum version of 1.18.1 or `kubelogin`. With the Azure CLI and the Azure PowerShell module, these two commands are included and automatically managed. Meaning, they're upgraded by default and running `az aks install-cli` isn't required or recommended. If you're using an automated pipeline, you need to manage upgrades for the correct or latest version. The difference between the minor versions of Kubernetes and `kubectl` shouldn't be more than *one* version. Otherwise, authentication issues occur on the wrong version.
- If you're using helm, you need a minimum version of helm 3.3.
- This configuration requires you have a Microsoft Entra group for your cluster. This group is registered as an admin group on the cluster to grant admin permissions. If you don't have an existing Microsoft Entra group, you can create one using the `az ad group create` command.

Note

Microsoft Entra integrated clusters using a Kubernetes version newer than version 1.24 automatically use the `kubelogin` format. Beginning with Kubernetes version 1.24, the default format of the `clusterUser` credential for Microsoft Entra ID clusters is `exec`, which requires `kubelogin` binary in the execution `PATH`. There's no behavior change for non-Microsoft Entra clusters, or Microsoft Entra ID clusters running a version older than 1.24. Existing downloaded `kubeconfig` continues to work. An optional query parameter `format` is included when getting `clusterUser` credential to overwrite the default behavior change. You can explicitly specify format to `azure` if you need to maintain the old `kubeconfig` format.

1. Create an Azure resource group using the `az group create` command.

   ```
   az group create --name myResourceGroup --location centralus
   ```

2. Create an AKS cluster and enable administration access for your Microsoft Entra group using the `az aks create` command.

   ```
   az aks create \
    --resource-group myResourceGroup \
    --name myManagedCluster \
    --enable-aad \
    --aad-admin-group-object-ids <id> \
    --aad-tenant-id <id> \
    --generate-ssh-keys
   ```

   A successful creation of an AKS-managed Microsoft Entra ID cluster has the following section in the response body.

   ```
   "AADProfile": {
    "adminGroupObjectIds": [
    "aaaaaaaa-0000-1111-2222-bbbbbbbbbbbb"
    ],
    "clientAppId": null,
    "managed": true,
    "serverAppId": null,
    "serverAppSecret": null,
    "tenantId": "aaaabbbb-0000-cccc-1111-dddd2222eeee"
   }
   ```

Enable AKS-managed Microsoft Entra integration on your existing Kubernetes RBAC enabled cluster using the `az aks update` command. Make sure to set your admin group to keep access on your cluster.

```
az aks update \
 --resource-group MyResourceGroup \
```

```
  --name myManagedCluster \
  --enable-aad \
  --aad-admin-group-object-ids <id-1>,<id-2> \
  --aad-tenant-id <id>
```

A successful activation of an AKS-managed Microsoft Entra ID cluster has the following section in the response body:

```
 "AADProfile": {
  "adminGroupObjectIds": [
      "aaaaaaaa-0000-1111-2222-bbbbbbbbbbbb"
  ],
  "clientAppId": null,
  "managed": true,
  "serverAppId": null,
  "serverAppSecret": null,
  "tenantId": "aaaabbbb-0000-cccc-1111-dddd2222eeee"
}
```

If your cluster uses legacy Microsoft Entra integration, you can upgrade to AKS-managed Microsoft Entra integration through the `az aks update` command.

Warning

Free tier clusters might experience API server downtime during the upgrade. We recommend upgrading during your nonbusiness hours. After the upgrade, the `kubeconfig` content changes. You need to run `az aks get-credentials --resource-group <AKS resource group name> --name <AKS cluster name>` to merge the new credentials into the `kubeconfig` file.

```
az aks update \
  --resource-group myResourceGroup \
  --name myManagedCluster \
  --enable-aad \
  --aad-admin-group-object-ids <id> \
  --aad-tenant-id <id>
```

A successful migration of an AKS-managed Microsoft Entra ID cluster has the following section in the response body:

```
 "AADProfile": {
  "adminGroupObjectIds": [
      "aaaaaaaa-0000-1111-2222-bbbbbbbbbbbb"
  ],
  "clientAppId": null,
  "managed": true,
```

```
  "serverAppId": null,
  "serverAppSecret": null,
  "tenantId": "aaaabbbb-0000-cccc-1111-dddd2222eeee"
}
```

1. Get the user credentials to access your cluster using the `az aks get-credentials` command.

   ```
   az aks get-credentials --resource-group myResourceGroup --name myManagedCluster
   ```

2. Follow your sign in instructions.

3. Set `kubelogin` to use the Azure CLI.

   ```
   kubelogin convert-kubeconfig -l azurecli
   ```

4. View the nodes in the cluster with the `kubectl get nodes` command.

   ```
   kubectl get nodes
   ```

There are some non-interactive scenarios that don't support `kubectl`. In these cases, use `kubelogin` to connect to the cluster with a non-interactive service principal credential to perform continuous integration pipelines.

Note

Microsoft Entra integrated clusters using a Kubernetes version newer than version 1.24 automatically use the `kubelogin` format. Beginning with Kubernetes version 1.24, the default format of the `clusterUser` credential for Microsoft Entra ID clusters is `exec`, which requires `kubelogin` binary in the execution PATH. There's no behavior change for non-Microsoft Entra clusters, or Microsoft Entra ID clusters running a version older than 1.24. Existing downloaded `kubeconfig` continues to work. An optional query parameter `format` is included when getting `clusterUser` credential to overwrite the default behavior change. You can explicitly specify format to `azure` if you need to maintain the old `kubeconfig` format.

When getting the `clusterUser` credential, you can use the `format` query parameter to overwrite the default behavior. You can set the value to `azure` to use the original `kubeconfig` format:

```
az aks get-credentials --format azure
```

If your Microsoft Entra integrated cluster uses Kubernetes version 1.24 or lower, you need to manually convert the `kubeconfig` format.

```
export KUBECONFIG=/path/to/kubeconfig
kubelogin convert-kubeconfig
```

If you receive the message **error: The Azure auth plugin has been removed.**, you need to run the command `kubelogin convert-kubeconfig` to convert the `kubeconfig` format manually. For more information, see Azure Kubelogin Known Issues.

Important

The step described in this section suggests an alternative authentication method compared to the normal Microsoft Entra group authentication. Use this option only in an emergency.

If you lack administrative access to a valid Microsoft Entra group, you can follow this workaround. Sign in with an account that is a member of the Azure Kubernetes Service Cluster Admin role and grant your group or tenant admin credentials to access your cluster.

- Learn about Microsoft Entra integration with Kubernetes RBAC.
- Learn more about AKS and Kubernetes identity concepts.
- Learn how to use kubelogin for all supported Microsoft Entra authentication methods in AKS.
- Use Azure Resource Manager templates to create AKS-managed Microsoft Entra ID enabled clusters.
