­­

Nextgen cloud

Insight Enterprises

**Contents**

[Purpose 2](#_Toc102038158)

[Provisioning Project Using Cloud hosted pool 3](#_Toc102038159)

[Provisioning Project Using Self-hosted VMSS pool 6](#_Toc102038160)

[Provisioning Project Using Self-hosted AKS pool 8](#_Toc102038161)

[Adding new Service Connection in the ADO 10](#_Toc102038162)

[Using Python Script 10](#_Toc102038163)

[Directly in the ADO 11](#_Toc102038164)

[Conclusion 13](#_Toc102038165)

# Purpose

This document provides instructions on how to provision a project in the ADO. This document also provides instructions on how to add a new service connection in the ADO. There are 3 ways to provision the project in the ADO based on type of agents to be used:

* Using Cloud hosted pool (**Note:** Only for Microsoft 365 customers)
* Using Self-hosted VMSS pool
* Using Self-hosted AKS pool

Also, the project can be added in existing ADO, or it can be added in the new ADO, which is created through scripts. All the codes for the service catalogs and its pipelines are maintained in **Azure IaC, which is the** centralized project.

**Prerequisites before provisioning the project**

* Name of the client organization

Diagram

Description automatically generated

* Email id associated with the client organization

A picture containing text

Description automatically generated

* Personal Access Token with Full Access of the client organization

Graphical user interface, text, application

Description automatically generated

* Contact the “Insight Team” for the Personal Access Token of the Central Repository.

# Provisioning Project Using Cloud hosted pool

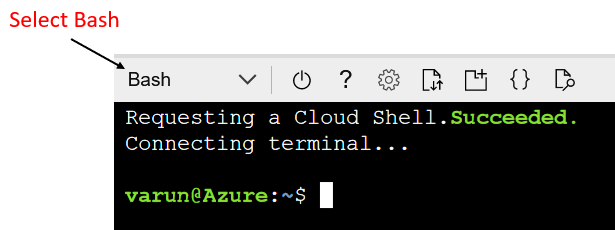
Perform the following steps to provision the project using the Cloud hosted pool in new ADO:

1. Open the Cloud Shell on the Azure Portal.

Graphical user interface, text, application

Description automatically generated

1. Click the dropdown menu and select **Bash** as the cloud shell environment.



**Note:** Script cannot be run in **PowerShell** environment.

1. Git clone the repository on the cloud shell. Repository Link: [**https://nsitmspdevops@dev.azure.com/nsitmspdevops/Insight%20CMP%20Development/\_git/ado\_migration**](https://nsitmspdevops@dev.azure.com/nsitmspdevops/Insight%20CMP%20Development/_git/ado_migration)

While cloning, enter the password provided by the Insight Team.

1. Open the **ado\_migration** folder and then open the **ado\_construction** folder.
2. Open the **ado\_cloudhosted** folder.
3. Use the **python3 create.py** command to run the **create.py** file.
4. Enter the name of the azure subscription.
5. Enter the number associated to the region to provision the resources.

If we want to create a new ADO for the project, follow the below steps:

1. Enter **1** to create a new ADO.
2. Enter the name of the organization to construct ADO.
3. Enter the number associated to the region to provision the ADO.
4. Enter the PAT token of the central repository.

If we want to use the existing ADO for the project, follow the below steps:

1. Enter **2** to use an existing ADO.
2. Enter the name of the organization of the ADO.
3. Enter the email id associated with this organization.
4. Enter the PAT token of this organization.
5. Enter the PAT token of the central repository.

The following image represents the running of the python script.

Graphical user interface, text

Description automatically generated

This script helps provision a project by migrating all the service catalog terraform code from central repository to the new ADO. It also helps in installing all the terraform extensions in the ADO as well as creating a service connection.

**Note:** This script will not create any agents pool because default agent pool will be required for triggering the pipelines. Also, default agent pool is enabled only for Microsoft subscription customers.

After the successful execution of the script, a new project will be created in the ADO.

As previously stated, default agents can only be used by Microsoft Subscription Customers. The following image represents the error shown in the pipeline for non-subscription customers.

Graphical user interface, text, application, email, Teams

Description automatically generated

**Perform the following steps to enable the Microsoft subscription in the ADO to use the default agents:**

1. Navigate to the ADO. Click on **Organization Settings** and click **Billing.**

Graphical user interface, application

Description automatically generated

1. Click **Set up billing** to setup the Azure Subscription for the billing.

After setting up the billing for the subscription, we will be able to use the default agents for running the pipeline. The following image represents the billing page after the full setup.

Graphical user interface, application, email

Description automatically generated

**Perform the following steps to delete the project in the ADO:**

1. Open the Cloud Shell on the Azure Portal.
2. Open the **ado\_migration** folder and then open the **ado\_construction** folder.
3. Open the **ado\_cloudhosted** folder.
4. Use the **python3 destroy.py** command to run the **destroy.py** file.

This script will destroy the project in the ADO as well as destroy the service connection that was created while provisioning the project.

# Provisioning Project Using Self-hosted VMSS pool

**Perform the following steps to provision the project using the Self-hosted VMSS pool:**

1. Open the Cloud Shell on the Azure Portal.

Graphical user interface, text, application

Description automatically generated

1. Click the dropdown menu and select **Bash** as the cloud shell environment.

Graphical user interface, text

Description automatically generated

**Note:** Script cannot be run in **PowerShell** environment.

1. Git clone the repository on the cloud shell. Repository Link: [**https://nsitmspdevops@dev.azure.com/nsitmspdevops/Insight%20CMP%20Development/\_git/ado\_migration**](https://nsitmspdevops@dev.azure.com/nsitmspdevops/Insight%20CMP%20Development/_git/ado_migration)

While cloning, enter the password provided by the Insight Team.

1. Open the **ado\_migration** folder and then open the **ado\_construction** folder.
2. Open the **ado\_selfhosted\_vmss** folder.
3. Use the **python3 create.py** command to run the **create.py** file.
4. Enter the name of the azure subscription.
5. Enter the number associated to the region to provision the resources.

If we want to create a new ADO for the project, follow the below steps:

1. Enter **1** to create a new ADO.
2. Enter the name of the organization to construct ADO.
3. Enter the number associated to the region to provision the ADO.
4. Enter the PAT token of the central repository.

If we want to use the existing ADO for the project, follow the below steps:

1. Enter **2** to use an existing ADO.
2. Enter the name of the organization of the ADO.
3. Enter the email id associated with this organization.
4. Enter the PAT token of this organization.
5. Enter the PAT token of the central repository.

The following image represents the running of the python script.

Graphical user interface, text

Description automatically generated

This script helps provision a project by migrating all the service catalog terraform code from central repository to the new ADO. It also helps in installing all the terraform extensions in the ADO, creating a service connection as well as creating the VMSS agents in the ADO.

After the successful execution of the script, a new project will be created in the ADO.

**Note:** If the script is unable to add virtual machine scale set as agent pool, follow the below steps to add the agent pool manually in the ADO.

1. Open the project in the ADO. Navigate to the **Project Settings**.
2. Click **Agent pools** to add the VMSS pool agents.
3. Click the **Add pool** button.
4. In Add agent pool form, select **Azure Virtual Machine Scale Set** in the pool type field.
5. Select the azure subscription from the subscription field.
6. Select **vmssagentspool** in the Virtual Machine Scale Set field.
7. Enter the name of the agent pool as **vmsspool**.
8. Configure the settings per requirements for the **Maximum number of virtual machines in the scale set**, **Number of agents to keep on standby**, and **Delay in minutes before deleting excess idle agents**.
9. Select the checkbox to grant access permissions to all the pipelines.
10. Click **Create** to create an agent pool in the ADO.

Graphical user interface, application

Description automatically generated

Graphical user interface, text, application

Description automatically generated

This will add an agent pool in the ADO, which will be required for triggering the pipelines.

**Perform the following steps to delete the project in the ADO:**

1. Open the Cloud Shell on the Azure Portal.
2. Open the **ado\_migration** folder and then open the **ado\_construction** folder.
3. Open the **ado\_selfhosted\_vmss** folder.
4. Use the **python3 destroy.py** command to run the **destroy.py** file.

This script will destroy the project in the ADO as well as destroy the service connection and agent pools that were created while provisioning the project.

# Provisioning Project Using Self-hosted AKS pool

**Perform the following steps to provision the project using the Self-hosted AKS pool:**

1. Open the Cloud Shell on the Azure Portal.

Graphical user interface, text, application

Description automatically generated

1. Click the dropdown menu and select **Bash** as the cloud shell environment.

Graphical user interface, text

Description automatically generated

**Note:** Script cannot be run in **PowerShell** environment.

1. Git clone the repository on the cloud shell. Repository Link: [**https://nsitmspdevops@dev.azure.com/nsitmspdevops/Insight%20CMP%20Development/\_git/ado\_migration**](https://nsitmspdevops@dev.azure.com/nsitmspdevops/Insight%20CMP%20Development/_git/ado_migration)

While cloning, enter the password provided by the Insight Team.

1. Open the **ado\_migration** folder and then open the **ado\_construction** folder.
2. Open the **ado\_selfhosted\_aks** folder.
3. Use the **python3 create.py** command to run the **create.py** file.
4. Enter the name of the azure subscription.
5. Enter the number associated to the region to provision the resources.

If we want to create a new ADO for the project, follow the below steps:

1. Enter **1** to create a new ADO.
2. Enter the name of the organization to construct ADO.
3. Enter the number associated to the region to provision the ADO.
4. Enter the PAT token of the central repository.

If we want to use the existing ADO for the project, follow the below steps:

1. Enter **2** to use an existing ADO.
2. Enter the name of the organization of the ADO.
3. Enter the email id associated with this organization.
4. Enter the PAT token of this organization.
5. Enter the PAT token of the central repository.

The following image represents the running of the python script.

Graphical user interface, text

Description automatically generated

This script helps provision a project by migrating all the service catalog terraform code from central repository to the new ADO. It also helps in installing all the terraform extensions in the ADO, creating a service connection as well as creating the AKS cluster agents in the ADO.

After the successful execution of the script, a new project will be created in the ADO.

**Perform the following steps to delete the project in the ADO:**

1. Open the Cloud Shell on the Azure Portal.
2. Open the **ado\_migration** folder and then open the **ado\_construction** folder.
3. Open the **ado\_selfhosted\_aks** folder.
4. Use the **python3 destroy.py** command to run the **destroy.py** file.

This script will destroy the project in the ADO as well as destroy the service connection and agent pools that were created while provisioning the project.

# Adding new Service Connection in the ADO

A user can add a new service connection in the ADO either by using a python script or directly adding in the ADO. This section provides instructions on both the ways.

## Using Python Script

Perform the following steps to add a new service connection in the ADO using the python script:

1. Open the Cloud Shell on Azure Portal.

Graphical user interface, text, application

Description automatically generated

1. Click the dropdown menu and select **Bash** as the cloud shell environment.

Graphical user interface, text

Description automatically generated

**Note:** Script cannot be run in **PowerShell** environment.

1. Git clone the repository on the cloud shell. Repository Link: [**https://nsitmspdevops@dev.azure.com/nsitmspdevops/Insight%20CMP%20Development/\_git/ado\_migration**](https://nsitmspdevops@dev.azure.com/nsitmspdevops/Insight%20CMP%20Development/_git/ado_migration)

While cloning, enter the password provided by the Insight Team.

1. Open the **ado\_migration** folder and then open the **serviceconnection** folder.

In this folder there are three sub folders, and these are:

* + - * **add\_one\_service\_connection\_ado :** contains python script for adding only one service connection in the ADO.
      * **add\_two\_service\_connection\_ado :** contains python script for adding two service connections in the ADO at once.
      * **add\_three\_service\_connection\_ado :** contains python script for adding three service connections in the ADO at once.

1. Open any folder as per the requirement and run **create.py** using **python3** **create.py** command.
2. Enter the number associated to the region to provision the resources.
3. Enter the organization name of the ADO.
4. Enter the email id associated with this organization.
5. Enter the PAT token of this organization.
6. Enter the name of the azure subscriptions.

After the successful execution of the script, new service connections will be created in the ADO under Service connections setting present in the Project Settings.

The following image represents the running of the python script.

Text

Description automatically generated

## Directly in the ADO

Perform the following steps to add a new service connection in the ADO, without using the python script:

1. Open the Cloud Shell on Azure Portal.

Graphical user interface, text, application

Description automatically generated

1. Click the dropdown menu and select **Bash** as the cloud shell environment.

Graphical user interface, text

Description automatically generated

**Note:** Script cannot be run in **PowerShell** environment.

1. Paste the following command in the cloud shell.

**“az ad sp create-for-rbac --role contributor --scope /subscriptions/<Enter Subscription ID>”**

Text

Description automatically generated

1. Copy the **appId** (Service Principal ID), **password** (Service Principal Key) and **tenant** (Tenant ID)
2. Open the Project in the ADO. Navigate to **Project Settings**.
3. Under Pipelines, click **Service Connections.**
4. Click **New service connection.**
5. Select **Azure Resource Manager** and click **Next**.
6. Select **Service principal (manual)** as Authentication method and click **Next**.
7. Enter the **Subscription Id** and the **Subscription Name.**
8. Paste the **service principal id**, **service principal key** and the **tenant id** copied from step 3 into their respective fields.
9. Click **Verify**.
10. Set the **name** of the service connection as the name of the **subscription**.
11. Select the checkbox to grant permission access to all the pipelines.
12. Click **Verify and save** to create a new service connection for the respective subscription in the ADO.

Graphical user interface, application

Description automatically generated

Graphical user interface

Description automatically generated

# Conclusion

In this document, we have discussed three different ways of provisioning the project in the ADO based on the types of agents that are required. In this ADO, we have installed all the terraform extensions as well as established the service connection and the agent pools that are required to run the pipelines. We also discussed two different ways of adding a new service connection in the ADO.