Cranium Deployment Guide

Azure ML and Azure OpenAI

September 2023

Overview

This document walks through the step-by-step process of setting up and deploying a Cranium Integration to your organization's Azure Machine Learning and Azure OpenAI instances. Support is available for any questions or issues, either by emailing support@cranium.ai, or submitting a ticket on support.cranium.ai.

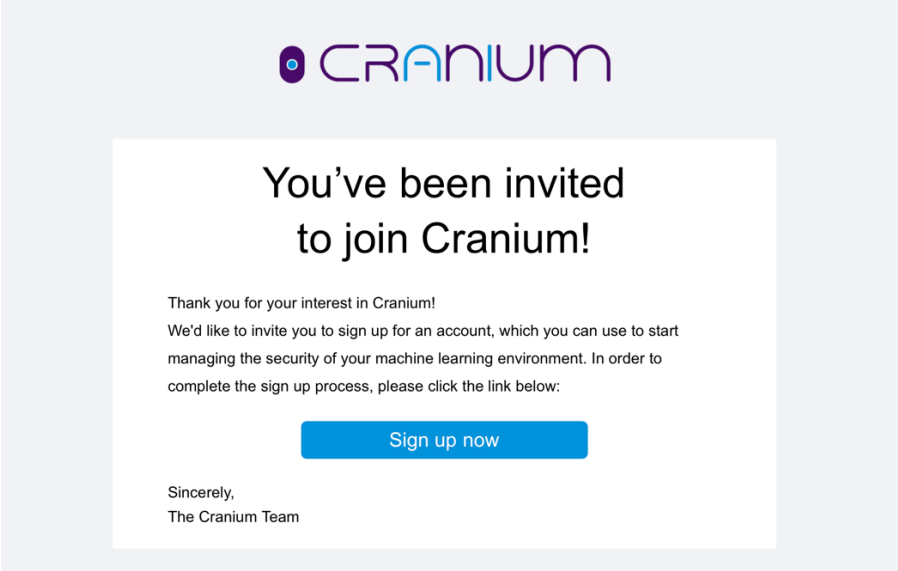
Included in this guide:

- Pre-Requisites
- Deployment Steps
- Getting Started with Cranium
- Architecture Diagram
- Asset Information Received from Connector
- Analysis Information Received from Connector

Pre-Requisites

Prior to integration, please review the following items and contact support if any items are missing.

1. Confirm that an email was sent to you or someone on your team from noreply@cranium.net with the subject "Invitation to join Cranium".

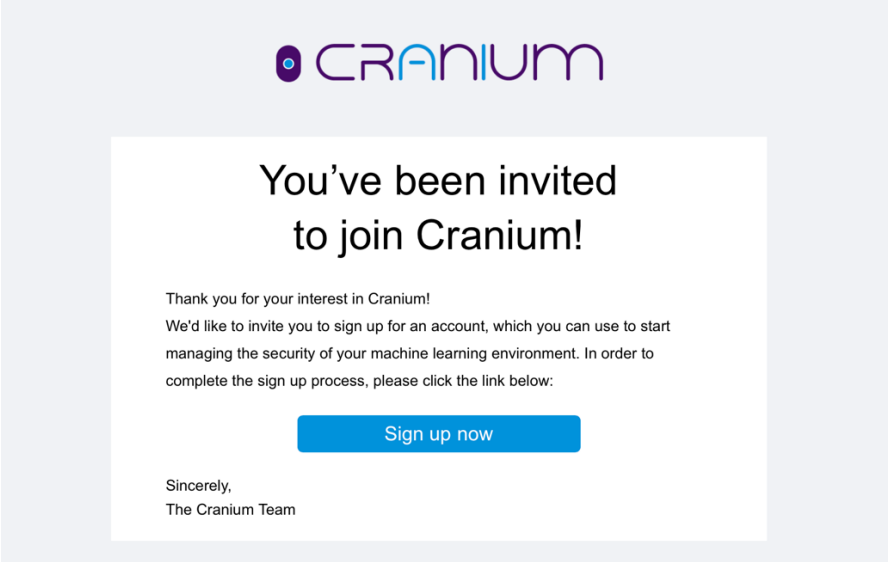


2. Azure admin privileges are needed to grant the proper API app permissions in Step 2.5 of the Cranium integration deployment.
    a. If an Azure admin is not available at the time of the scheduled integration, Step 2 can be completed prior to the rest of the integration.
    b. Note that the secret value created for the new App registration will need to be available at the time of deployment for Step 4.
3. Owner privileges are needed in Azure for the specific resource group, Azure OpenAI serivce, subscription, or management group that will be connected to the Cranium Connector.

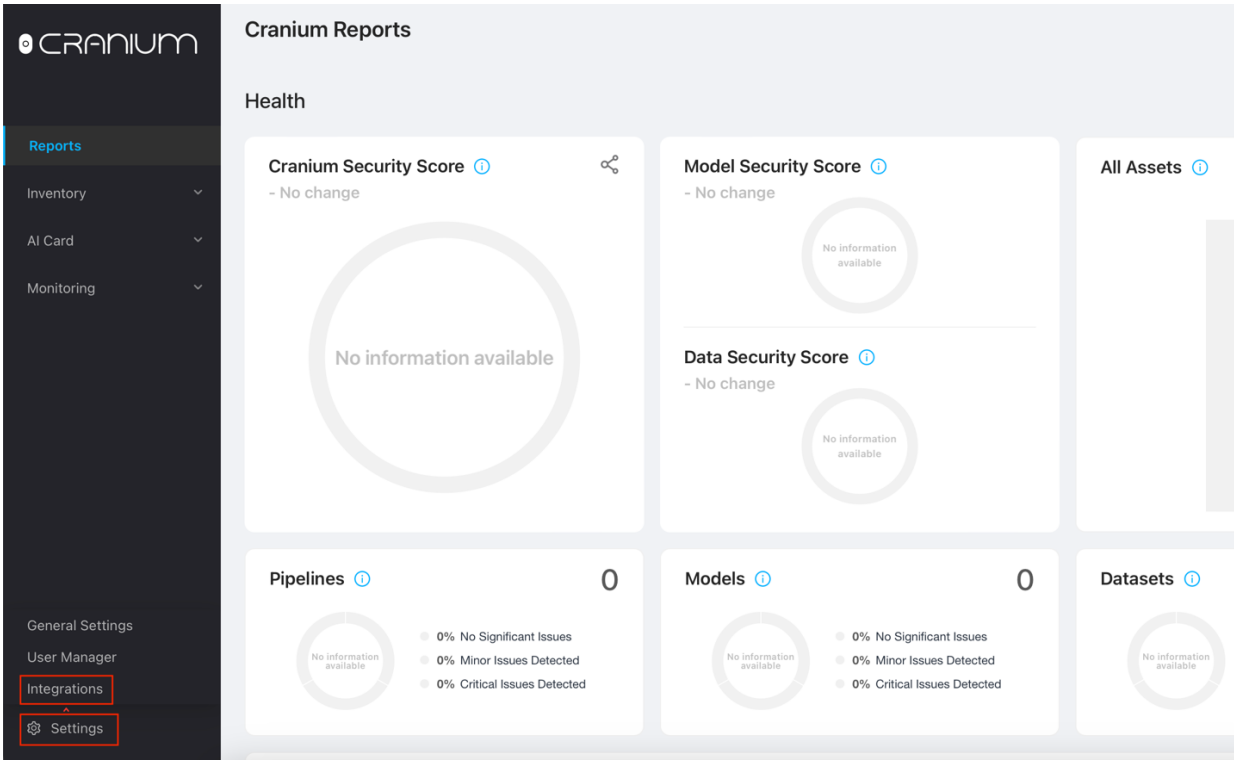
Deployment Steps

Step 1: Create an Azure ML Integration in Cranium and save the integration token

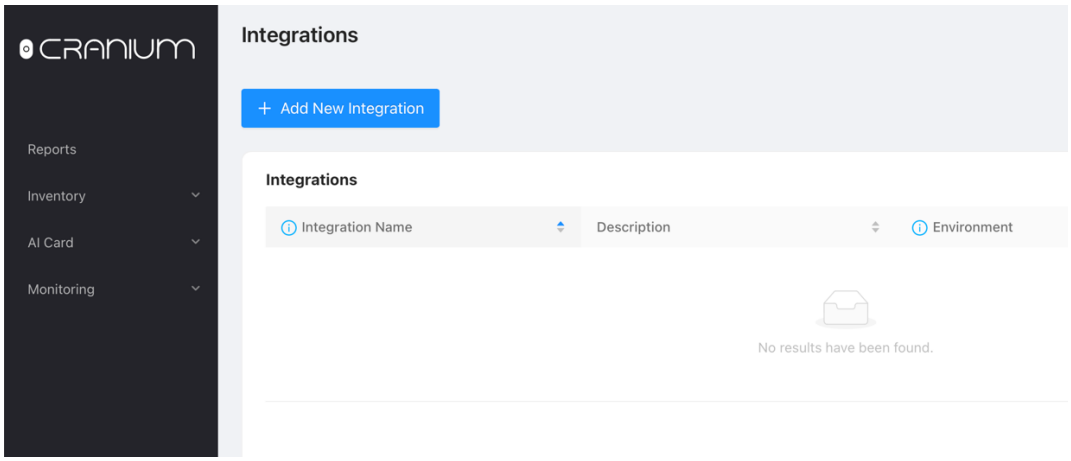
1. Login to Cranium by clicking the “Sign up now” button from your “Invitation to join Cranium” email



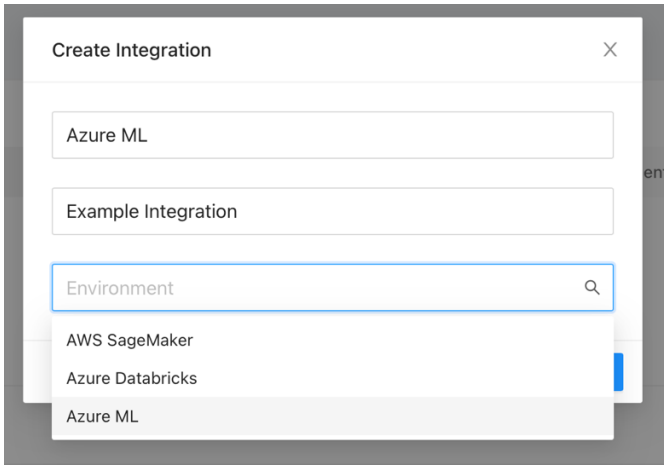
2. Complete the sign up instructions
3. Go to Settings > Integrations in the bottom left corner of the page



4. Click “Add New Integration”

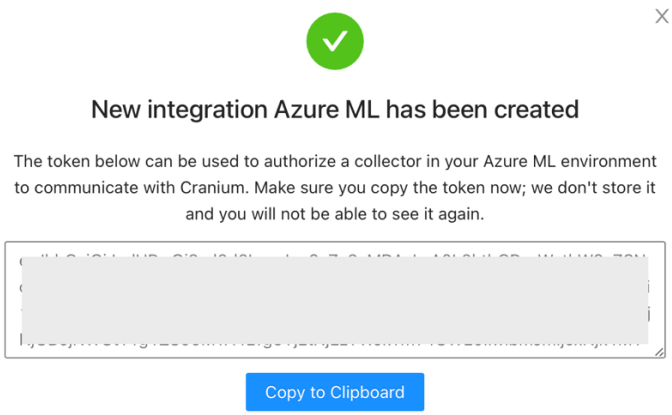


5. Give it any name and description you’d like and select “Azure ML” as the environment. (“Azure OpenAI“ will be added as an option in an upcoming release)



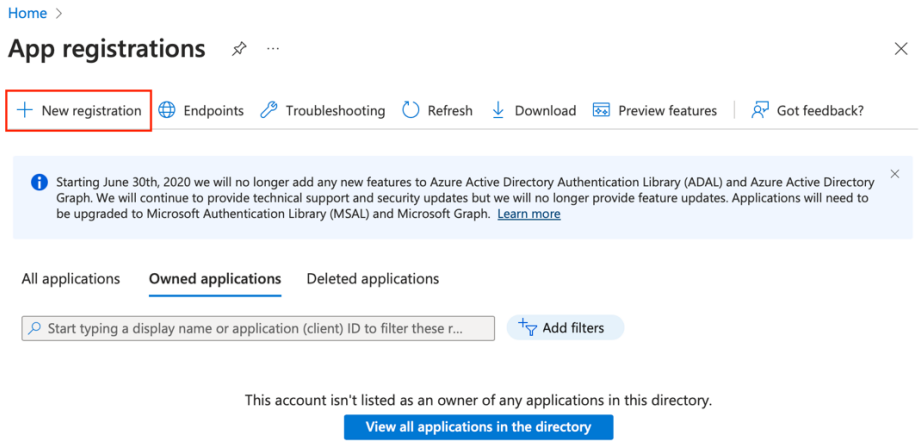
   - Note that this Environment label does not impact anything about the way the integration works and is only used as a helpful identifier

6. Click “Save” and “Copy to Clipboard” on the next screen
   - **IMPORTANT**: Save the copied token value somewhere to be used in Step 4

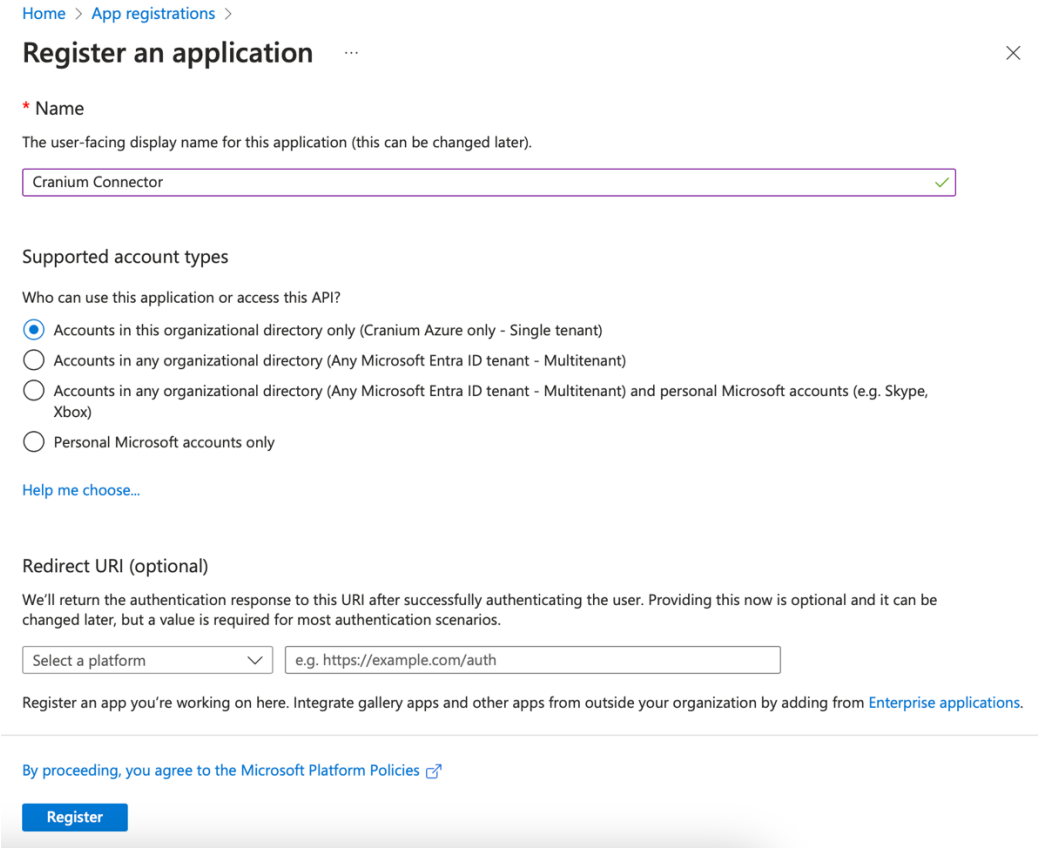


Step 2: Register a new app in Azure with Graph Application “User.Read.All” permissions and create a new secret and save the secret value

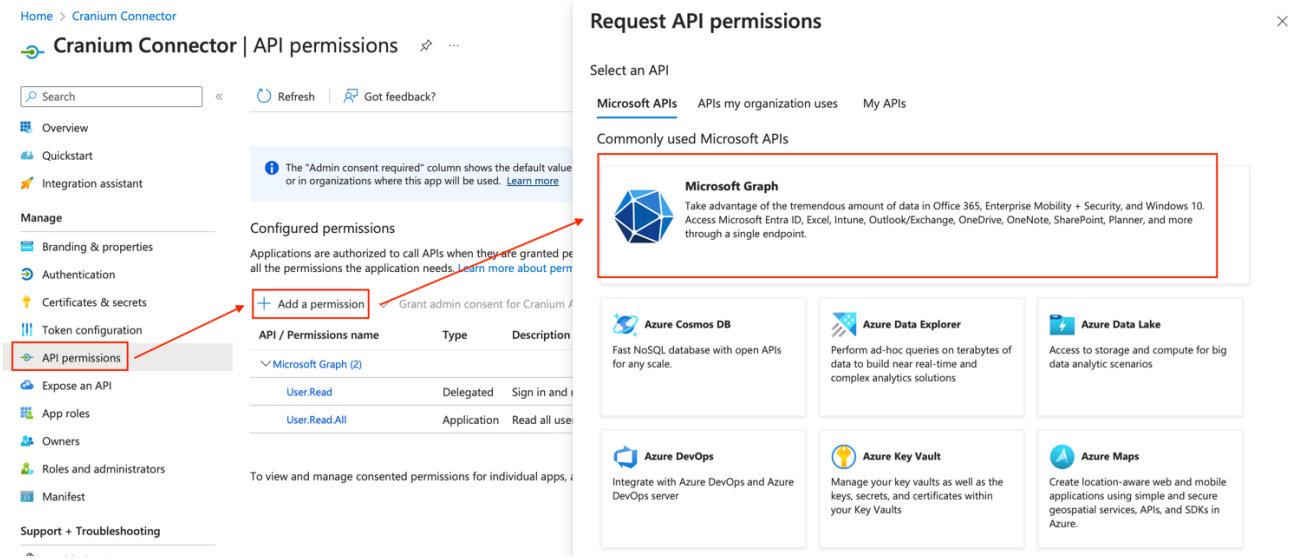
1. Go to the “App registrations” page and click “New Registration”



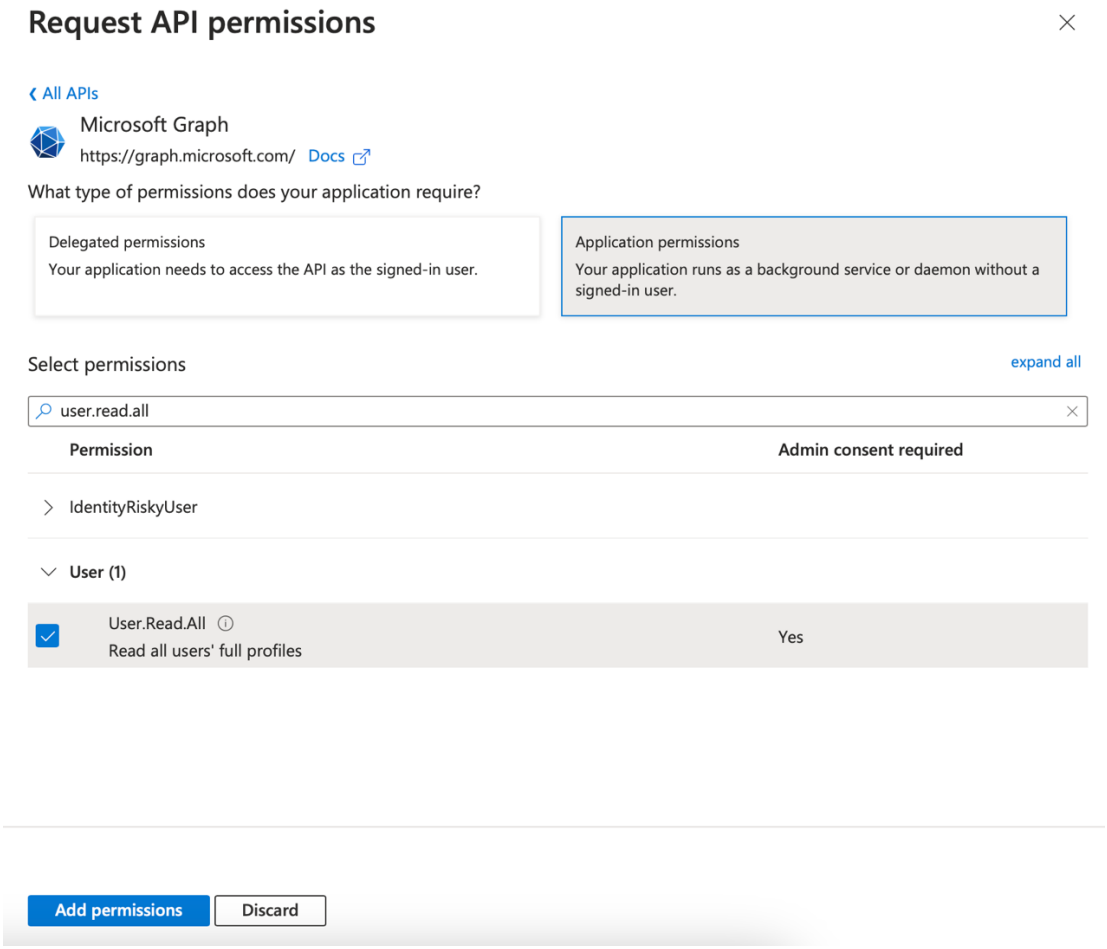
2. Give it any name you’d like, leave the defaults, and click “Register”



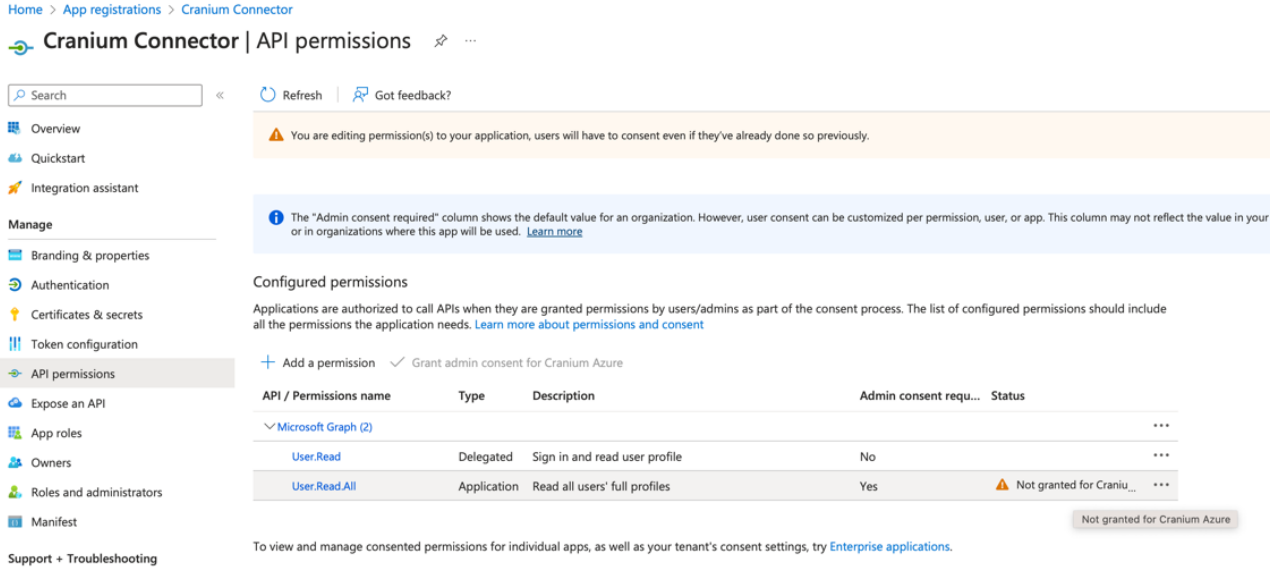
3. In “API permissions”, click “Add a permission”, and select "Microsoft Graph"



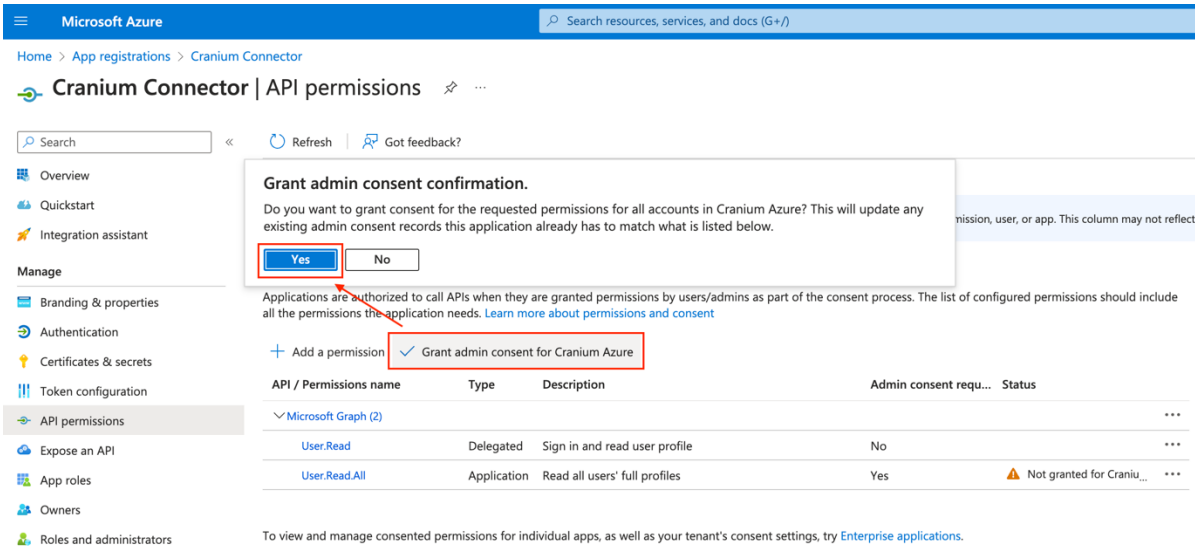
4. Select “Application permissions", find and select "User.Read.All" Permission and click "Add permissions"



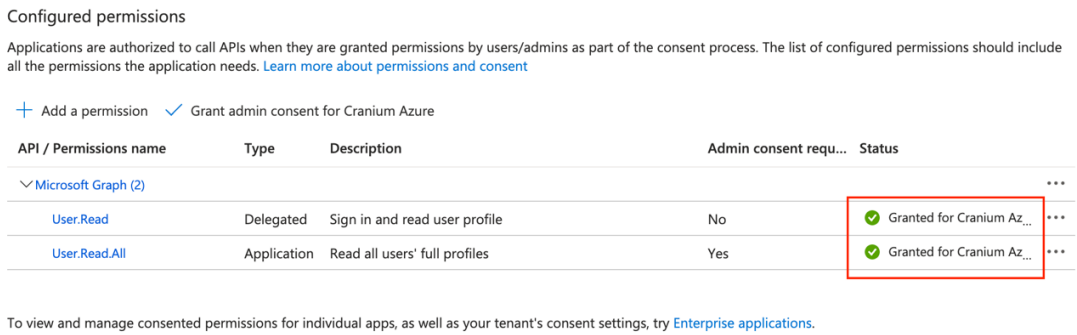
5. If this permission was created without admin privileges, then a warning will appear on the top of the screen and an admin will need to access the newly created Cranium Connector to grant access



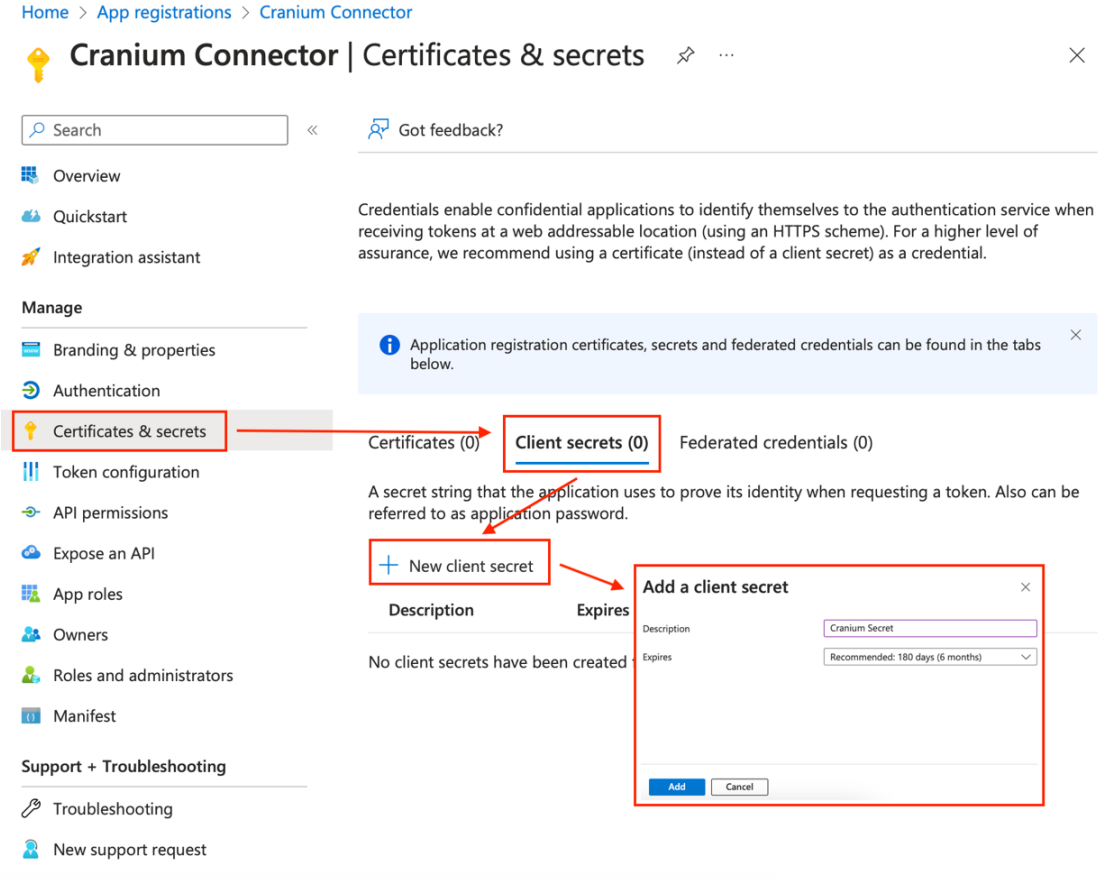
a. For Azure admins granting API permissions, navigate to “App registrations” and find the new Cranium Connector, click “Grant admin consent for ____”, and “Yes”



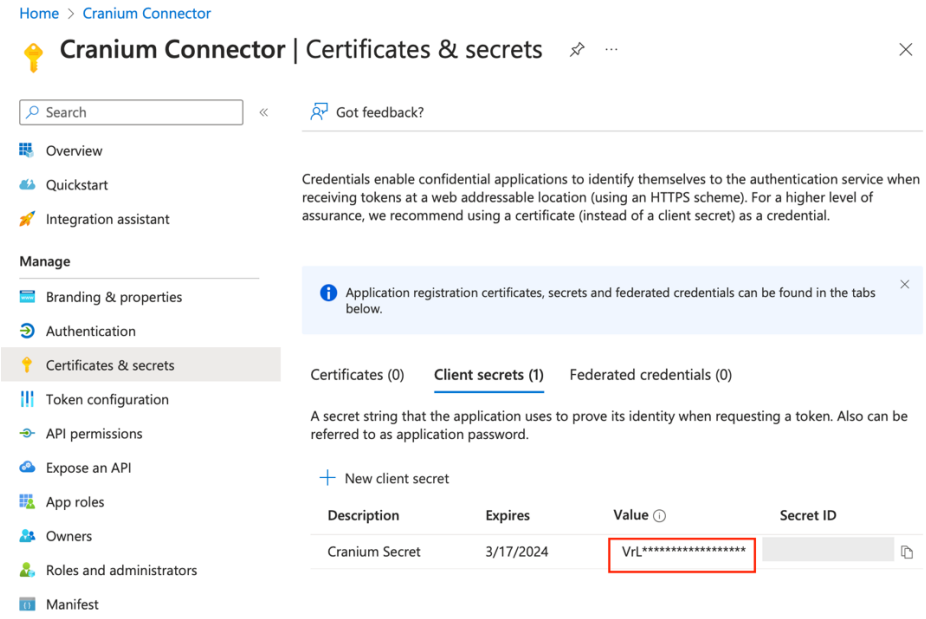
b. Confirm that permissions are granted under “Status”



6. In “Certificates & Secrets”, add a new Client secret with any description and save the secret value somewhere safe



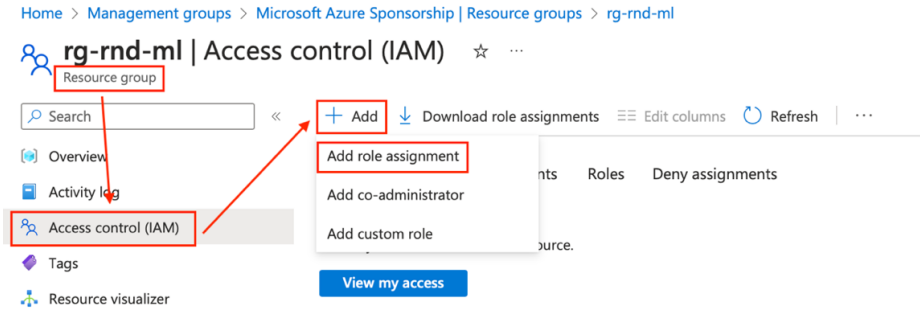
- **IMPORTANT**: Copy and save the client secret value somewhere to be used in Step 4



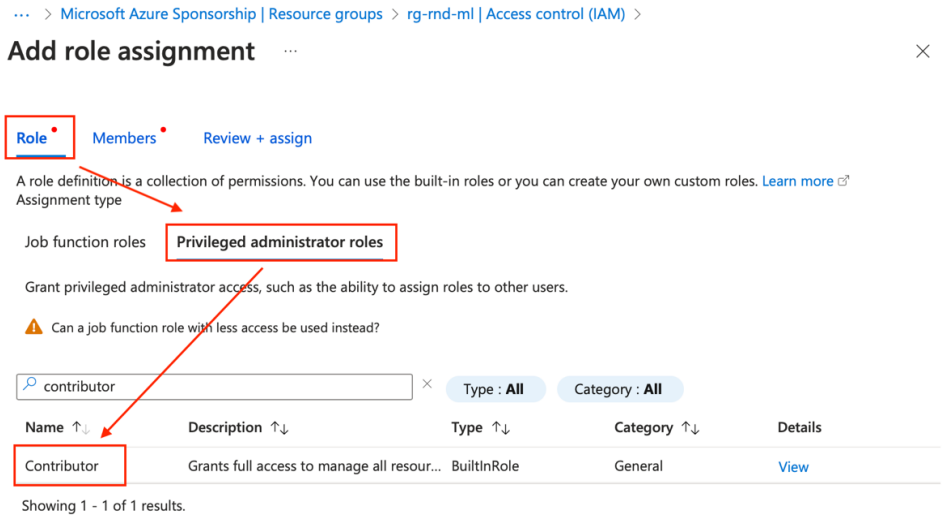
Step 3: Assign “Contributor” role for the Cranium Connector member to the Azure ML resource group or Azure OpenAI service

For Azure ML

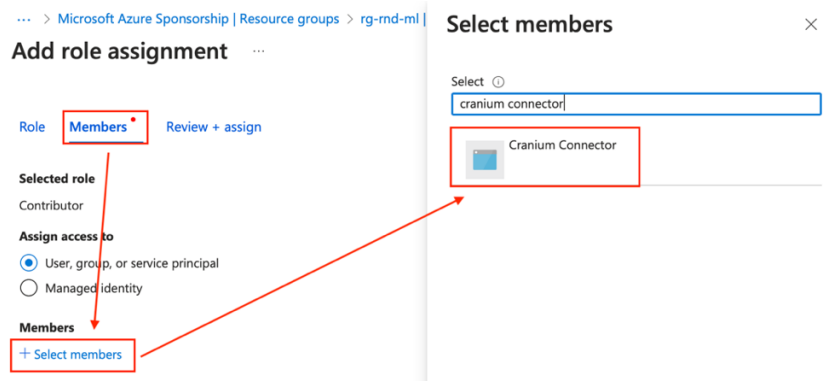
1. Navigate to the resource group, subscription, or management group to configure “Access control (IAM)” and “Add role assignment”



   - Cranium recommends following the principle of least privilege access by adding role assignments at the resource group level
   - Note that Access Control is only available to Azure users with "Owner" permissions
2. For the “Role”, navigate to “Privileged administrator roles” to find and select “Contributor”



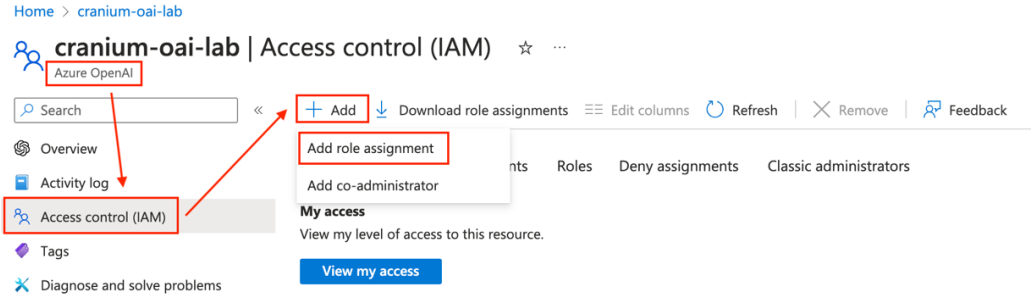
3. For the “Members”, click “Select members”, find and select the new Service Principal



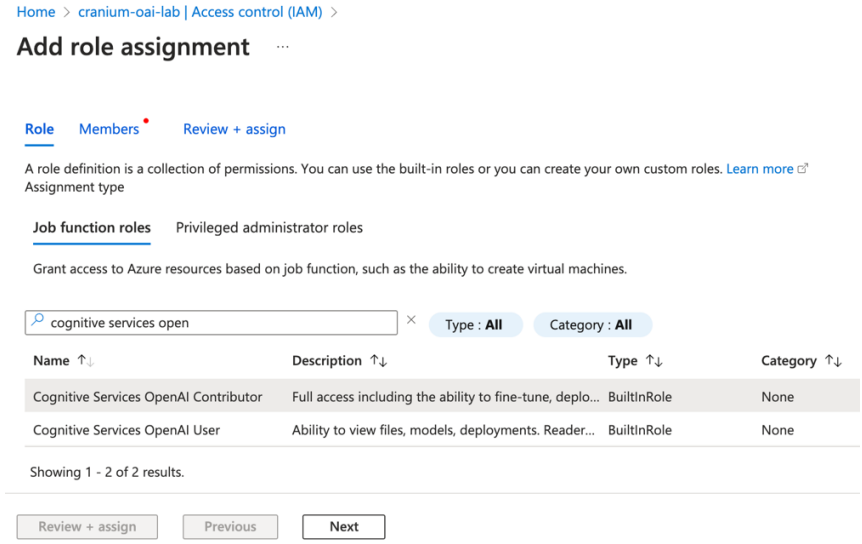
4. Review + assign

For Azure OpenAI

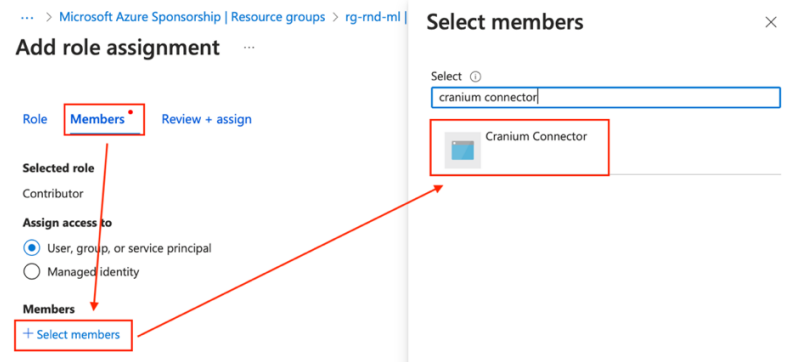
1. Navigate to the Azure OpenAI service to configure “Access control (IAM)” and “Add role assignment”



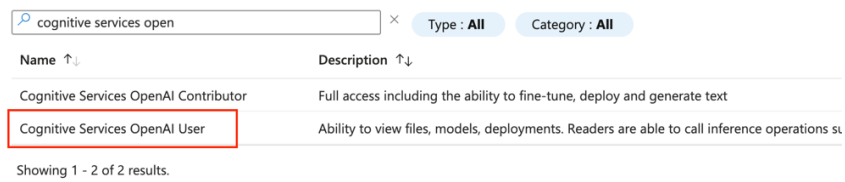
   - Note that Access Control is only available to Azure users with "Owner" permissions
2. For the “Role”, find the “Cognitive Services OpenAI Contributor” role and click “Next”



3. For the “Members”, click “Select members”, find and select the new Service Principal

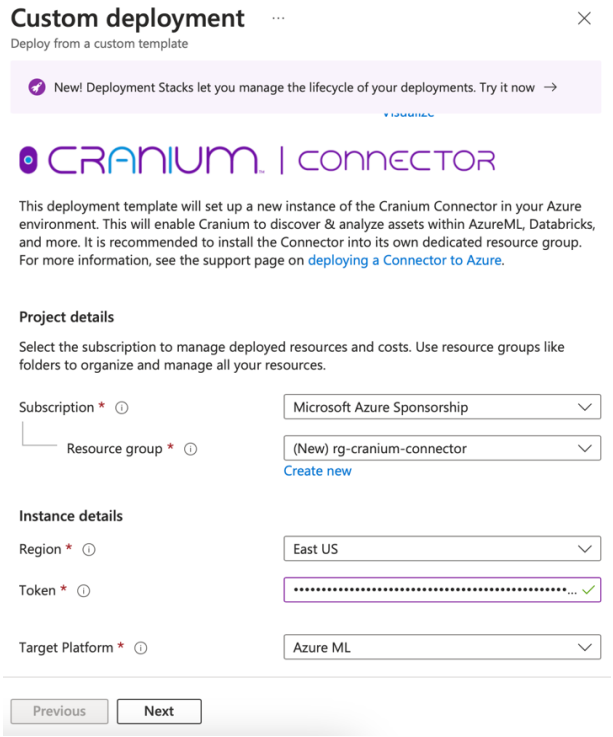


4. Review + assign
5. Complete steps 2-4 for the “Cognitive Services OpenAI User” role

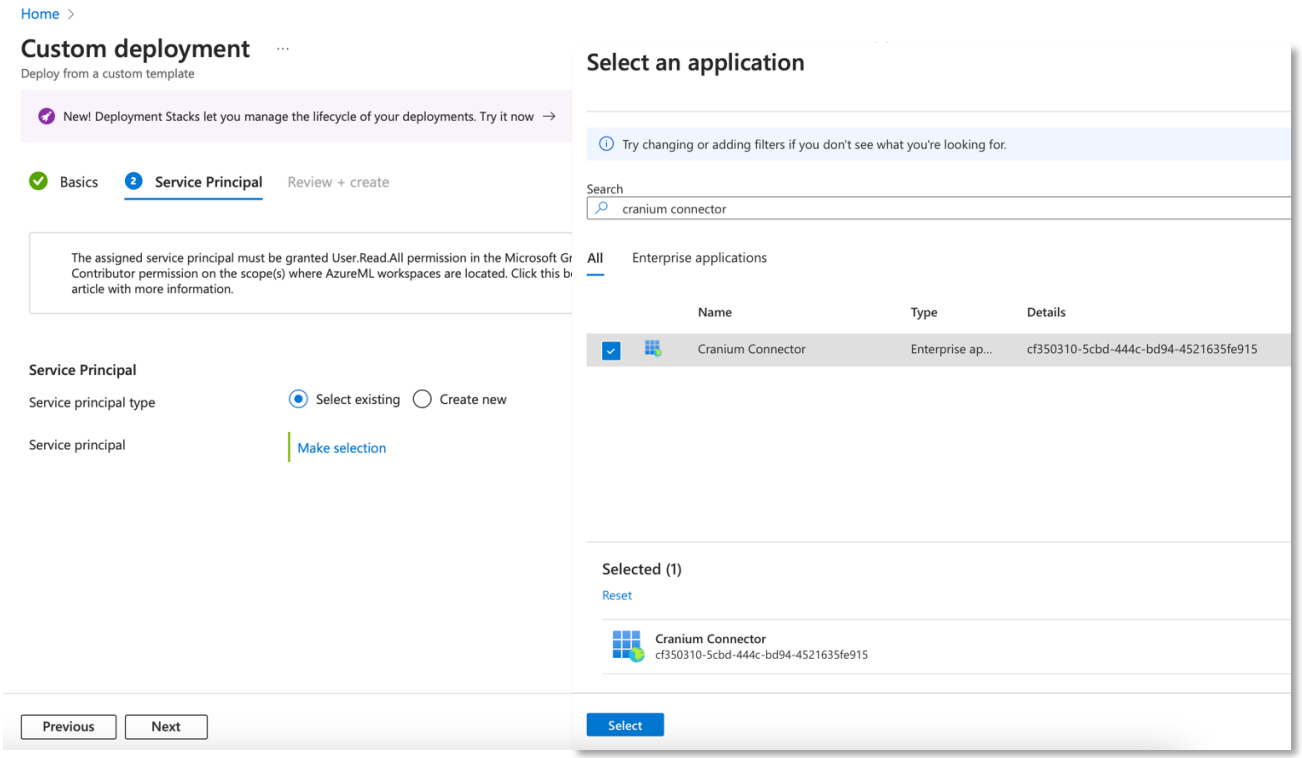


Step 4: Deploy the Cranium Connector ARM Template

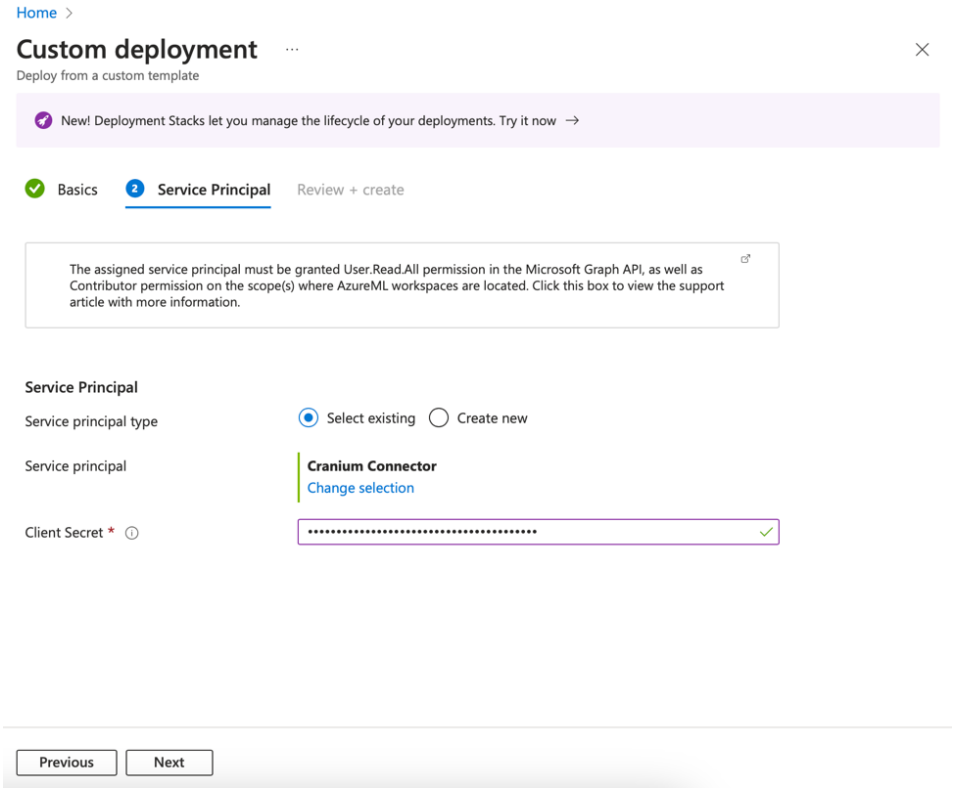
1. Open the Cranium Connector ARM Template: Deploy the Cranium Connector
2. Enter the Subscription and Resource group where you'd like to deploy
    - Cranium recommends setting up a new, dedicated resource group for the deployment
3. Fill the integration Token from step 1 and click “Next”



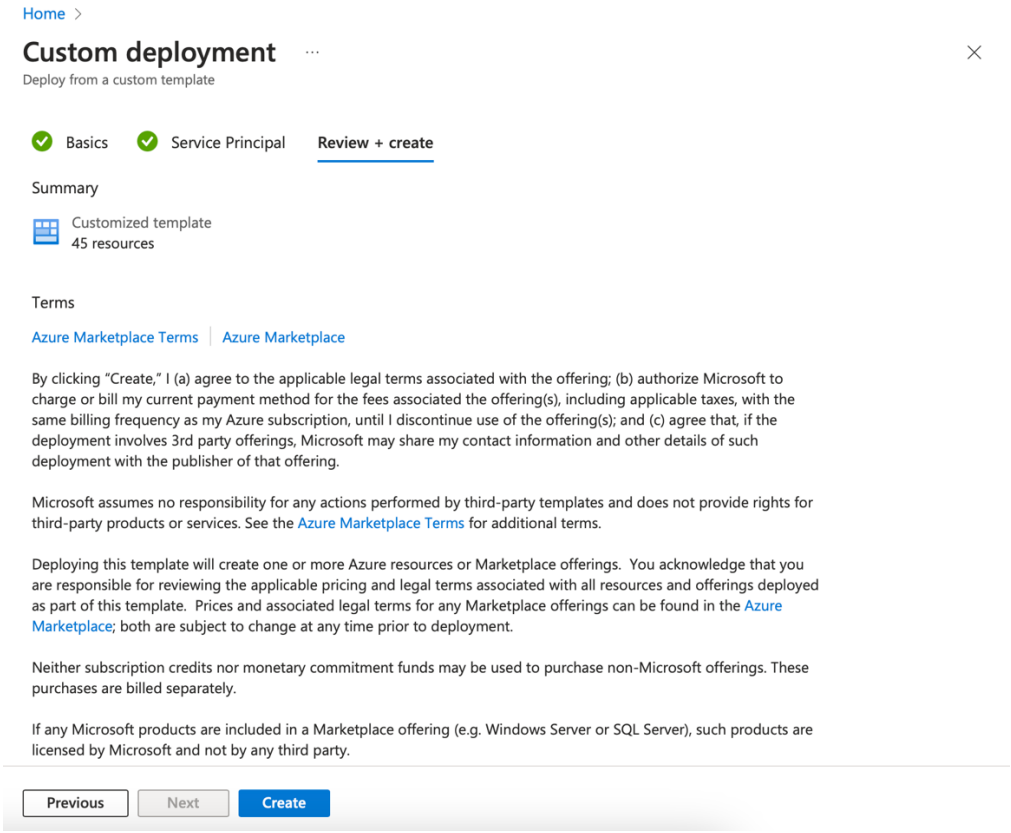
4. “Select existing” Service Principal and select your new Cranium Connector app



5. Enter the service principal Client Secret from step 2, click “Next”



6. Click "Create" to complete deployment!



   - Deployment should take 20-30 minutes

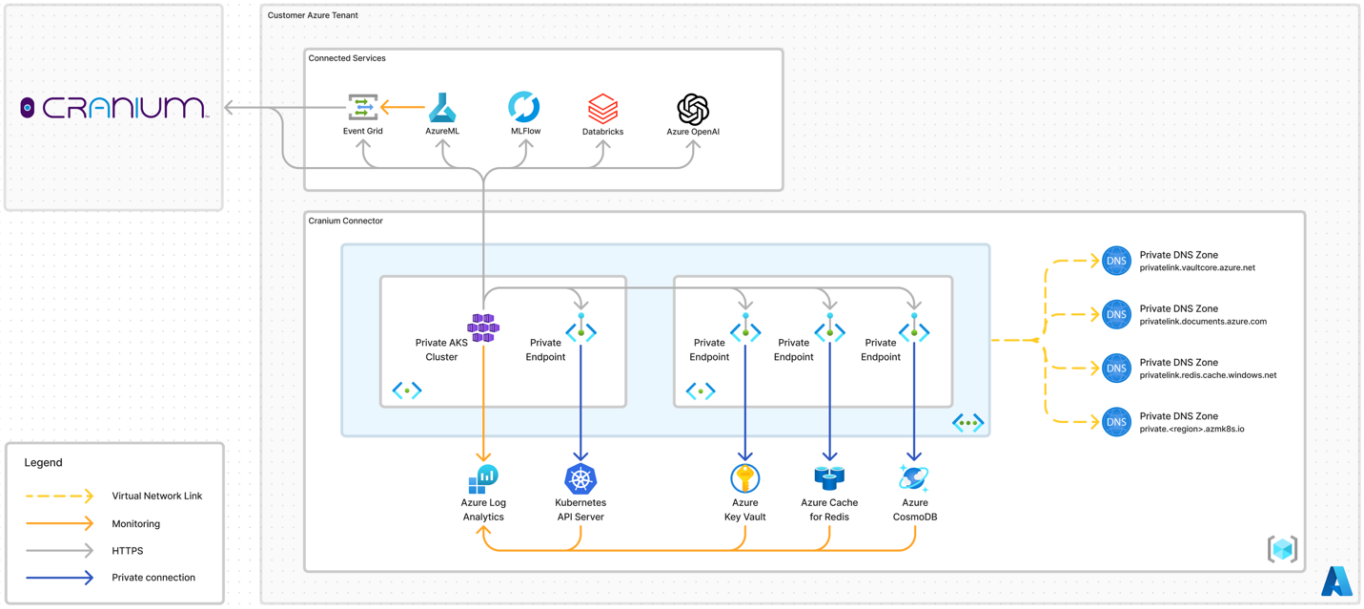
Getting Started with Cranium

The following links from the Cranium Knowledge Base can help when starting out with the Cranium platform after a connector has been deployed. Note that the Knowledge Base requires a Cranium account in order to access.

- Asset Overview
- Pipeline Overview
- Monitoring Overview
- Creating a Pipeline
- Creating Alert Rules

Architecture Diagram

The diagram below describes the networking and Azure components included in a connector deployment. The in-scope components are those within the “Cranium Connector” box. All other components shown are for reference as to the nature of network connections being made.



Asset Information Received from Connector

The table below lists the asset attributes sent from the Cranium Connector through Azure back to Cranium.

Asset	Cranium	Azure
Model	ID	Azure ID
	Name	Model -> Name
	Version	Model -> Version
	Description	Model -> Description
	Job ID	Model -> Job ID
	Experiment	Model -> Job ID -> Experiment
	Created By	Model -> Created By
	Created On	Model -> Created On
	Runs	--
	Datasets	--
	Event (Add Model version | Update version | Delete version)	--
Experiment	ID	Azure ID
	Name	Experiment -> Name
	Latest Job	Experiment -> Run
	Job types	Experiment -> Type
	Event (Created | Delete)	--
Run	ID	Azure ID
	Type (AutoML | Pipeline | Script)	Job -> Job Type
	Experiment	Job -> Experiment
	Duration	Job -> Runtime
	Compute Target	Job -> Compute -> Target
	Status (Completed | Failed)	Job -> Status
	Parent Run	Job -> Parent
	Created By	Job -> Created By
	Created On	Job -> Created On
	Datasets	--
	Models	--
	Result	--
Deployment	ID	Azure ID
	Name	Endpoint -> Display Name
	Type (Realtime | Batch)	Endpoint -> Type
	Compute Type	Deployment -> Compute -> Type
	Compute Target	Deployment -> Compute -> Target
	Created By	Deployment -> Created By
	Created On	Deployment -> Created On
	Models	--
	Event (Add | Delete)	--
Dataset	ID	Azure ID
	Name	Dataset -> Name
	Version	Dataset -> Version
	Description	Dataset -> Description
	Storage Type	Data Store Type (Azure Blob Storage, Azure File Share, etc)
	Type (Table | File | Folder)	--
	Created By	Dataset -> Creaded By
	Created On	Dataset -> Created On
	Runs	--
	Models	--
	Event (Add | Update version | Delete)	--
User	ID	Azure Graph -> User -> ID
	First Name	Azure Graph -> User -> Given Name
	Last Name	Azure Graph -> User -> Surname
	Email	Azure Graph -> User -> Email
	Actions	--

Analysis Information Received from Connector

The table below lists the analysis fields and metrics sent from the Cranium Connector through Azure back to Cranium.

Asset	Analysis Type	Calculated Fields & Metrics
Model	Metrics	Accuracy Precision Recall F-score ROC AUC Normalized RMS Error Spearman Correlation R2 Score Normalized Mean Absolute Error
Dataset	Structure	Name Path Type Size
	Features	Name Type missing/mistyped values
	Numeric Features	Mean Median Minimum Maximum Distribution Correlation
	Categorical Features	Unique value hashed Frequency
	Data/Feature Shift	JS test WS distance KS test
Detection	Poison Detection	Result Confidence Poisoned dataset IDs Model IDs Poisoned data primary keys Clean dataset IDs
	Backdoor Detection	Result Confidence Dataset IDs Model IDs Calculated trigger heatmap image