

IBM watsonx.ai With ARO on Azure Marketplace

Deployment Guide

July 2025

Overview.....	2
Cost and licenses.....	3
Architecture	3
Planning the deployment	5
Specialized knowledge	5
Technical requirements	5
Deployment options	6
Pre-requisites	6
Step 1. IBM watsonx.ai Subscription	6
Step 3. Storage Subscription	7
Step 4. Sign in to your Azure portal account	7
Launch the Deployment	7
Parameters for deploying IBM watsonx.ai into A NEW OR EXISTING VIRTUAL NETWORK.....	8
Basic configuration	8
Infrastructure configuration:	9
Virtual Network configuration:	9
OpenShift hosts configuration:	9
Storage Configuration:.....	11
watsonx configuration	11
IBM watsonx Services	12
Steps to Create Azure Service Principal Client ID and Client Secret	12
Generating the Portworx Spec URL for ARO cluster	14

Manage your cluster using the OpenShift Console	18
Login to watsonx web client.....	20
(Optional) Provide Boot Node SSH access	23
Scaling up your cluster by adding compute nodes	24
watsonx.ai services	29
watsonx.ai System and Services requirements	30
Steps to Install any other watsonx Service (Not available on Azure Marketplace template).....	30
Limitations	31
Additional resources	31
Document revisions	31

Overview

This deployment guide provides step-by-step instructions for deploying IBM watsonx.ai on ARO (Azure Red Hat OpenShift) Container Platform cluster on the Azure Cloud. This automatically deploys a multi-master, production instance of watsonx.

IBM watsonx.ai on Azure

IBM watsonx™ is an AI and data platform that includes three core components: watsonx.ai, watsonx.data and watsonx.governance. These three components are elegantly designed to help you scale and accelerate the impact of AI with trusted data across your business. The focus of this document is on how to install watsonx.ai on Azure.

IBM watsonx.ai is the next-generation enterprise studio for all AI builders to build, train, validate, tune and deploy AI models. It brings traditional machine learning and new generative AI capabilities powered by foundation models into a powerful studio that spans the AI lifecycle. IBM's open, hybrid, full stack approach includes a collection of foundation models including IBM-developed models that combine best-of-breed architectures with a rigorous focus on data acquisition, provenance, and quality, to serve enterprise needs.

The collection also includes third-party and select open-source foundation models from Hugging Face. Watsonx.ai also features a Prompt Lab and API to experiment with foundation models and build prompts for various use cases. The studio also includes a data science toolset to build machine learning models with code or with auto AI capabilities, as well as a collection of powerful capabilities such as visual data pipelines and flows, synthetic data generation and more - all running on a scalable, open and trusted, hybrid AI infrastructure.

This reference deployment provides Azure ARM templates to deploy watsonx.ai onto a new OpenShift cluster. This cluster includes:

- A Red Hat OpenShift Container Platform cluster created in a new VNet on Red Hat CoreOS (RHCOS) instances, using the [Red Hat OpenShift Installer Provisioned Infrastructure](#). See the [OpenShift Container Platform Installation overview](#) for details about the underlying OpenShift deployment architecture.
- A highly available storage infrastructure with Red Hat OpenShift Container Storage.
- Scalable OpenShift compute nodes running watsonx.ai services. For more information about watsonx.ai, see the [documentation](#).

Cost and licenses

The watsonx.ai environment is deployed by using Azure ARM template. You are responsible for the cost of the Azure services used for the infrastructure.

The Azure ARM template for this deployment includes configuration parameters that you can customize. You can use it to build a new VNet for your watsonx.ai solution on Azure cluster or deploy on an existing Azure VNet. Some of these settings, such as instance type, will affect the cost of deployment. For cost estimates, see the pricing pages for each Azure service you will be using. Prices are subject to change.

Pricing:

ARO Fee: <https://azure.microsoft.com/en-in/pricing/details/openshift/>

For more information about licensing terms, [watsonx.ai software license agreement](#).

Upgrading to the latest version of watsonx.ai indicates your acceptance of any new terms that may be applicable for the new version. To determine if new terms apply and to review them, please visit our [IBM Terms](#), and search for watsonx.ai

Architecture

Deploying the [Azure Marketplace template](#) for a new VNet with **default parameters** builds the following watsonx environment in the Azure Cloud to deliver the solution:

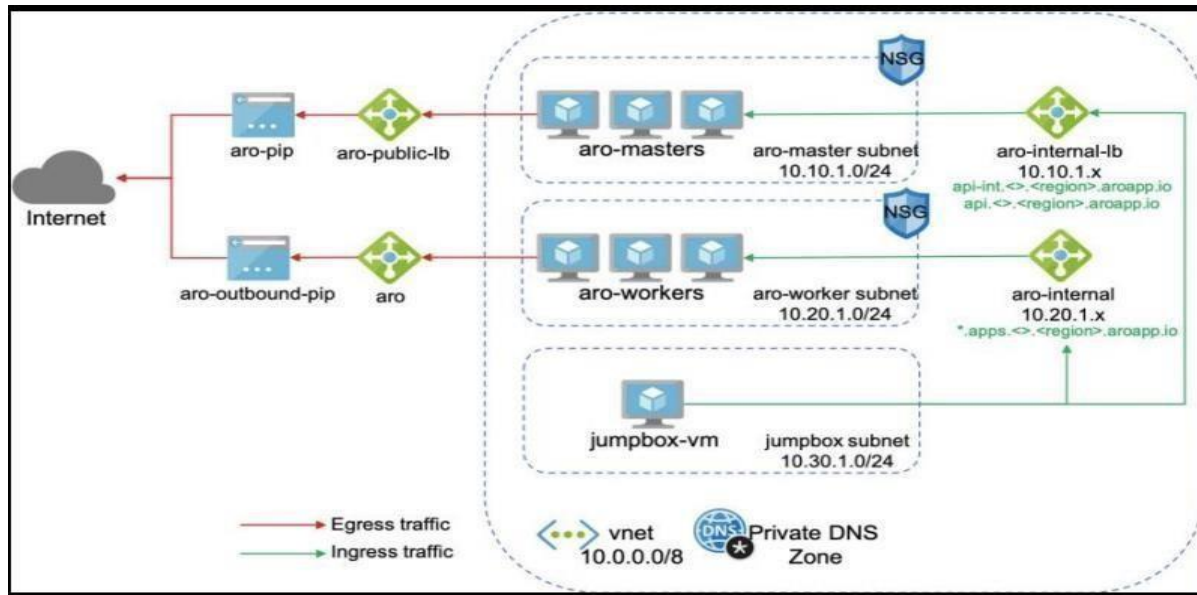


Figure 2: Deployment architecture for IBM watsonx.ai on Azure

The ARM template sets up the following:

- A highly available architecture that spans up to three Availability Zones. *
- A Virtual network configured with public and private subnets. *
- In the public subnets:
 - a bastion host to allow inbound Secure Shell (SSH) access to compute instances in private subnets.
- In the private subnets:
 - OpenShift Container Platform master instances.
 - OpenShift Container Platform (OCP) compute nodes that combined, contain watsonx.ai Collect, Organize, and Analyse services.
- An Azure Load Balancer spanning the public subnets for accessing watsonx.ai from a web browser.
- Storage disks with Azure Managed Disk mounted on compute nodes for ODF (OpenShift Data Foundation) v4.17 or Portworx Enterprise.
- An Azure domain as your public Domain Name System (DNS) zone for resolving domain names of the IBM watsonx.ai management console and applications deployed on the cluster.

* The template that deploys into an existing Virtual network skips the components marked by asterisks and prompts you for your existing Virtual network configuration.

watsonx.ai microservices are preconfigured on compute nodes. The following diagram shows the platform architecture.

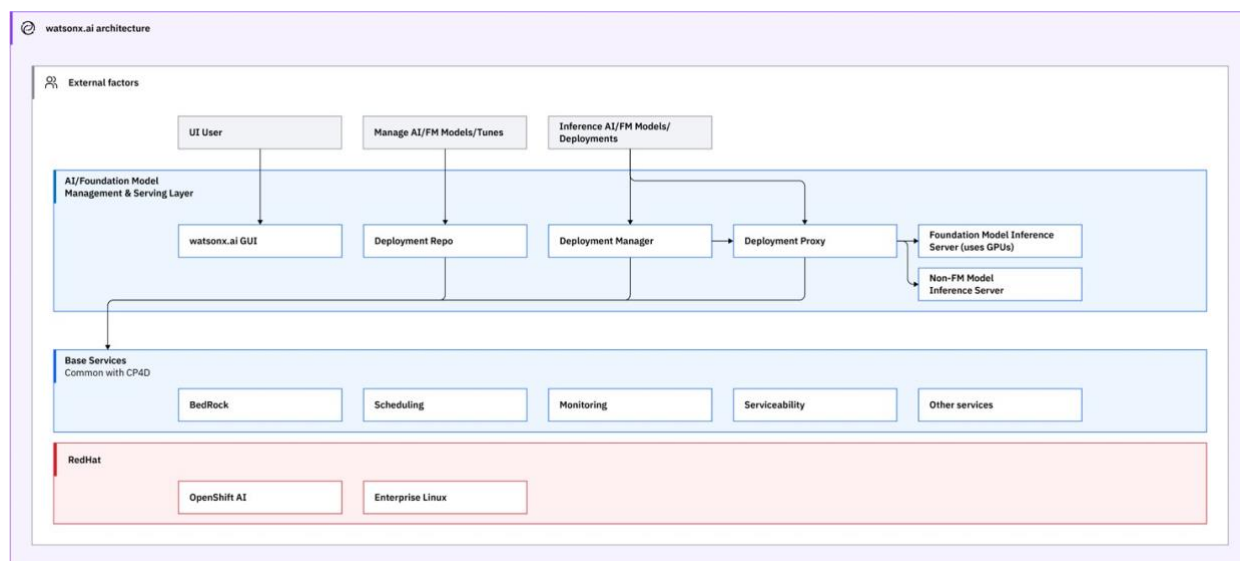


Figure 3: watsonx.ai architecture

Planning the deployment

Specialized knowledge

This deployment assumes basic familiarity with watsonx.ai service. If you're new to watsonx.ai and Red Hat OpenShift, see the [Additional resources](#) section.

This deployment also assumes familiarity with the OpenShift command line interface and Linux, in addition to a moderate level of familiarity with Azure services.

Technical requirements

[Red Hat Enterprise Linux CoreOS \(RHCOS\)](#) is used for the OpenShift compute node instances in this deployment.

Before you launch the template, your account must have resource quotas as specified in the following table.

Resources

If necessary, request [service quota increases](#) for the following resources. You might need to do this if an existing deployment uses these resources, and you might exceed the default quotas with this deployment. The [Service Quotas console](#) displays your usage and quotas for some aspects of some services. For more information, see the [Azure documentation](#).

Resource	This deployment uses
Virtual Network	1
Public IP addresses	3
Network Load Balancers	2
Standard_D4s_v3 virtual machines (Bootnode)	1
Standard_D8s_v3 virtual machines (Master nodes)	3
Standard_D16s_v3 virtual machines (Compute nodes)	4

	Standard D16s v3 virtual machines (Compute nodes) (only for ODF storage option)	3(ODF)
	Public IP addresses for ODF storage	2
Regions	This deployment includes 3 Availability Zones, which isn't currently supported in all Azure Regions.	
IAM permissions	To deploy the template, you must log in to the Azure portal with Azure Identity and Access Management (IAM) permissions for the resources and actions the templates will deploy. The <i>AdministratorAccess</i> managed policy within IAM provides sufficient permissions, although your organization may choose to use a custom policy with more restrictions.	

Deployment options

This template provides the following deployment options:

- **Deploy IBM watsonx.ai into a new Virtual Network** (end-to-end deployment). This option builds a new Azure environment consisting of the Virtual Network, subnets, NAT gateways, Network Security Groups (NSG), bastion hosts, and other infrastructure components, and then deploys watsonx.ai into this new Virtual Network.
- **Deploy IBM watsonx.ai into an existing Virtual Network.** This option provisions watsonx.ai in your existing Virtual Network infrastructure.

The template also lets you configure CIDR blocks, Virtual Machine types, and watsonx.ai settings, as discussed later in this guide.

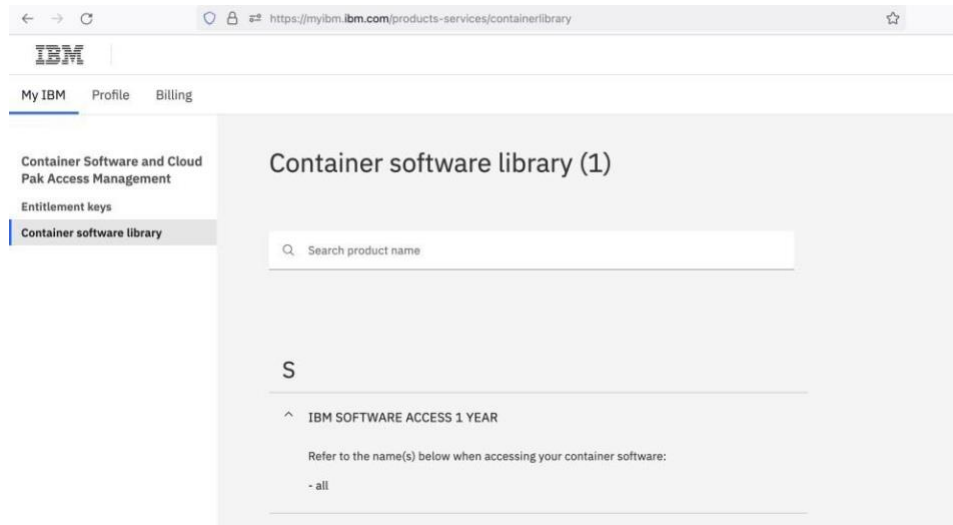
Pre-requisites

Ensure the following pre-requisites are in place with your existing watsonx.ai entitlements.

Step 1. IBM watsonx.ai Subscription

When you purchase IBM watsonx.ai from Marketplace, you will get the watsonx.ai entitlement Username and API Key. You should keep it handy as it's a required parameter in the ARM Template

Note: Please make sure you have valid entitlement key before using the offering, you can check your entitlement key is valid or not [here](#). Login to myibm.ibm.com and refer to the name(s) below when accessing your container software as shown in the below screenshot.



Step 2. Red Hat Subscription

Ensure that you have the Red Hat [OpenShift pull secret](#) with your purchase. You should keep it handy as this json is required parameter in the ARM Template. If you are deploying through Quickstart Template or from Azure Marketplace, you can directly enter the pull secret's json value.

You can select one of the three container storages while installing this Quick Start.

Step 3. Storage Subscription

- OpenShift Data Foundation (ODF): The Red Hat ODF license is linked as a separate entitlement to your RedHat subscription. This is the preferred option on Azure Marketplace.
- Portworx: When you select [Portworx](#) as the persistent storage layer, you will need to specify the install spec from your [Portworx account](#). You can generate a new spec using the [Spec Generator](#). Note that the Portworx trial edition expires in 30 days after which you need to upgrade to an Enterprise Edition. [Here are the steps to generate Portworx Spec.](#)

Step 4. Sign in to your Azure portal account

1. Sign in to your Azure account at <https://portal.azure.com/> with an Azure Identity and Access Management (IAM) user role that has the necessary permissions. For details, see [Planning the deployment](#), earlier in this guide.
2. Make sure that your Azure account is configured correctly, as discussed in [Technical requirements](#), earlier in this guide.
3. Use the Region selector in the navigation bar to choose the Azure Region where you want to deploy watsonx.ai on Azure. An IBM watsonx.ai high availability deployment is restricted to Azure Regions with at least three Availability Zones.
4. The following resources should be made available for watsonx.ai deployment
 - A new or an existing Key Vault location with Red Hat pull secret

Launch the Deployment

Note: The instructions in this section reflect the current version of the Azure portal. If you're using the redesigned portal, some of the user interface elements might be different.

You are responsible for the cost of the Azure services used while running this deployment. For full details, see the pricing pages for each Azure service you will be using for this deployment. Prices are subject to change.

1. Launch the Azure ARM template into your Azure account from the [Azure Marketplace](#) or directly or you can download the template and launch it separately from your account. watsonx.ai standard deployment takes about 4 hours.
2. Create a new Resource Group or specify existing one. This is where all the deployment resources will be stored. There will be two resource group created through ARM templates as specified in the parameters section below.
Note: Resource group should be unique, and same name shouldn't be specified it earlier. The ARM template may fail if the same name resource group created earlier in the past.
3. Check the Region that's displayed in the Region field and change it if necessary. This is where the network infrastructure for watsonx.ai will be built. The template is launched in the 'West US 2' Region by default.
4. Specify Service Principal App ID (Client ID) and Azure AD Client Secret. For more information check [how to create Service Principal](#) on Azure. It has steps to create service principal and get the client id and client secret.
5. For the Infrastructure settings, OpenShift settings, and watsonx.ai settings pages, review the parameters for the template. Provide values for the parameters that require input. For all other parameters, review the default settings and customize them as necessary.

In the following tables, parameters are listed by category and described.
When you finish reviewing and customizing the parameters, choose **Next**.

PARAMETERS FOR DEPLOYING IBM WATSONX.AI INTO A NEW OR EXISTING VIRTUAL NETWORK

Basic configuration:

Parameter label (name)	Default	Description
Subscription	Microsoft Azure Enterprise	The subscription is necessary to deploy cluster and create resources on the Azure portal. Make sure you have required administrator permission to create cluster resources, secrets, users, etc.

Resource Group	<i>Requires input</i>	Resource group is a container that holds related resource for an Azure Solution. Create a new resource group with the unique name (never used it earlier). There will be two Resource group created one with the name of the Resource Group and another one with the cluster name. The Cluster name Resource Group will hold all master and worker nodes related resources. E.g., Virtual Machines, Network Security Gateway (NSG), etc.
Region	West US 2	This is where the network infrastructure for watsonx will be built. The template is launched in the 'West US 2' Region by default.
Service Principal App ID	Azure —	This client ID will be used to create resources on the Azure portal. Check how to create Service Principal Azure App ID
Azure AD Client Secret	—	This client secret will be used to create resources on the Azure portal. Check how to get Azure Client Secret

Infrastructure configuration:

Parameter label (name)	Default	Description
Bootnode Public IP (Attach Public IP to BootnodeVM)	true	Bootnode can be accessible through SSH connection if bootnode has the public IP.
SSH public key	<i>Requires input</i>	Your machine's SSH public key will be added in the authorized_key in the bootnode so that it can be accessible through public IP.
Number of master nodes (NumberOfMaster)	3	The desired capacity for the OpenShift master instances. Must be an odd number. For a development deployment, 1 is sufficient; for production deployments, a minimum of 3 is required.
Number of compute nodes (NumberOfNodes)	4	The desired capacity for the OpenShift node instances. Minimum of 4 nodes required. Warning If the number of node instances exceeds your Red Hat entitlement limits or Azure virtual machine quotas, the stack will fail. Choose a number that is within your limits.
Bootnode VM size (BootnodeType)	Standard_D4s_v3	The virtual machine type for the OpenShift bootnode VM.
Master VM size (MasterInstanceType)	Standard_D8s_v3	The virtual machine type for the OpenShift master VMs.
Compute VM size (NodesInstanceType)	Standard_D16s_v3	The virtual machine type for the OpenShift compute VMs.

Virtual Network configuration:

Parameter label (name)	Default	Description
Virtual Network	<i>Requires input</i>	Create a new Virtual Network or select existing VNet
BootNode Subnet (Bootnode Subnet CIDR)	<i>Requires input</i>	Subnets for virtual network
Master Subnet (Master Subnet CIDR)	<i>Requires input</i>	Subnets for virtual network

Worker Subnet	<i>Requires input</i>	Subnets for virtual network
(Worker Subnet CIDR)		
Single or Multi Zone	Multi Zone	Deploy VMs to Single Zone region or Multiple Zone (Availability Zones). Recommended option is MultiZone. Please make sure that selected region supports AvailabilityZones: https://learn.microsoft.com/en-us/azure/reliability/availability-zones-service-support#azure-regions-with-availability-zone-support

OpenShift hosts configuration:

Parameter label (name)	Default	Description
RedHat subscription pull secret. (RedhatPullSecret)	<i>Requires input</i>	Enter json value of the RedHat Openshift Pull Secret , if you are deploying cluster through Azure Marketplace or internal GitHub repo.
		Note: If you are using internal GitHub repo to deploy the cluster then use Key vault path of OpenShift Installer Provisioned Infrastructure pull secret. e.g., <pre>"reference": { "keyVault": { "id": "/subscriptions/<YOUR-SUBSCRIPTION-ID>/resourceGroups/<RESOURCE-GROUP-OF-YOUR-KEY-VAULT>/providers/Microsoft.KeyVault/vaults/<YOUR-KEY-VALUE-NAME>" }, "secretName": "pullsecret" }</pre>
Cluster prefix (ClusterName)	<i>Requires input</i>	Custom cluster name for kubernetes.io/cluster/tags. The cluster name should be <i>unique</i> and should never be used earlier because this name will be also assigned it to another cluster resource group. This cluster resource group will hold all node related resources as mentioned in the Resource Group parameter.

		ARO cluster deployment needs Principal ID of the ARO Service Principal exist in your account in order to create ARO cluster in the given Azure account. Here is the command to fetch the Principal ID (ObjectId) of the ARO Service Principal, Azure CLI command: (Prereqs: Azure CLI installation in your computer.) <pre>`az ad sp list --filter "displayname eq 'Azure Red Hat OpenShift RP'" --query "[?appDisplayName=='Azure Red Hat OpenShift RP'].{name: appDisplayName, objectId: id}"`</pre> E.g.: <pre>[user]@[hostname] % az ad sp list --filter "displayname eq 'Azure Red Hat OpenShift RP'" --query "[?appDisplayName=='Azure Red Hat OpenShift RP'].{name: appDisplayName, objectId: id}"</pre> This command or command group has been migrated to Microsoft Graph API. Please carefully review all breaking changes introduced during this migration: https://docs.microsoft.com/cli/azure/microsoft-graph-migration <pre>[{ "name": "Azure Red Hat OpenShift RP", "objectId": "464114b9-XXXX-XXXX-a068-35ab76dXXXXX" }]</pre> Please make sure you have “Network Contributor” role assigned to ARO Resource Provider Principal ID. Note: If this Azure CLI command doesn’t work for you then you may need to register the required resource provider. Please follow instructions on this page and execute “<i>Get the service principal object ID for the Openshift resource provider – Azure CLI</i>” command: https://learn.microsoft.com/en-us/azure/openshift/quickstart-openshift-arm-bicep-template?pivots=aro-arm#register-the-required-resource-providers---azure-cli
Use Private or Public Endpoints (PublicCluster)	public	To Deploy a private cluster, select “private” and “public” for public cluster.
Enable Machine Autoscaler	false	Enable Machine Autoscaler to automate the scale up and down the cluster based on requirement dynamically.
Egress Outbound Type	Load Balancer	Choose value of Egress Outbound type values either Load Balancer and User Defined Routing.

Storage Configuration:

Parameter label (name)	Default	Description
Storage type for Cluster (StorageType)	<i>ODF or Portworx</i>	OpenShift Data Foundation (ODF), and Portworx storage options are available. ODF is recommended storage option.
IBM watsonx on ARO for Azure Marketplace Instance type (ODFInstanceType)	Standard_D16s_v3	Update this value if Storage type selected is ODF. The Virtual Machine type for the ODF instances. December 2024
Number of ODF nodes (NumberOfODF)	3	The desired capacity for the ODF instances. Minimum of 3 is required. You don't need to add these 3 ODF nodes in worker nodes count as our template will add 3 ODF nodes for you. Note: These 3 nodes will only be created for ODF but not for the Portworx storage option.
Portworx Spec URL	<i>Requires input (if Portworx is selected as a storage option.)</i>	Required field Only if you chose Portworx as storage option. Generated Spec URL for ARO cluster from Portworx spec generator: https://central.portworx.com/specGen/wizard . Here are the steps to generate Portworx Spec.
Portworx CSI	true	Required field Only if you chose Portworx as storage option. Portworx CSI option to enable CSI environment. Default value for Portworx CSI is true. It is recommended to enable Portworx CSI option.

watsonx configuration:

Parameter label (name)	Default	Description
IBM watsonx Entitled Registry API key Value (APIKey)	—	Enter the IBM Watsonx API key to access IBM Container Registry
License agreement (LicenseAgreement)	<i>Reject</i>	I have read and agreed to the license terms for IBM watsonx for that were provided to me at time of purchase. <i>You must accept the license to install watsonx services.</i>
OpenShift project (NameSpace)	zen	The OpenShift project that will be created for deploying watsonx It can be any lowercase string.
IBM watsonx version (CPDVersion)	5.2.x	The default version of watsonx to be deployed.
IBM watsonx Entitled Registry User (APIUsername)	cp	Enter the IBM watsonx Username value to access IBM Container Registry.

IBM

IBM watsonx Services:

Parameter label (name)	Default	Description
watsonx.ai	False	Choose True to install the IBM watsonx.ai service.

1. On the Openshift and Watsonx Settings page, select appropriate values from the above table.
2. On the **Review + Create** page, review and confirm the template settings. Make the necessary changes based on your requirement before deploying template.
3. Choose **Create** to deploy the templates.
4. On Azure Portal, monitor the status of the templates *Resource Group > Deployments*. When the status is succeeded for the watsonx.ai deployments, the watsonx.ai cluster is ready.
5. Use the URLs displayed in the *Resource Group > Deployments > AzureRMSamples > Outputs*. The URL for the “**watsonx web URL**” output key will navigate to the console login page.

Steps to Create Azure Service Principal Client ID and Client Secret

- Create an Azure Service Principal with Contributor and User Access Administrator roles.
- Create a Service Principal, using your Azure Subscription ID, named with a valid SP_NAME (e.g. AROClusterServicePrincipal) and save the returned json:

```
az login
az ad sp create-for-rbac --role="Contributor" --name="<SP_NAME>" --
scopes="/subscriptions/<subscription_id>"
```

- Assign the User Access Administrator role, using the AppId:□

```
az role assignment create --role "User Access Administrator" --assignee "<app_id>" --
scopes="/subscriptions/<subscription_id>"
```

Note: Save “*appId*” as a *ClientID* and “*password*” as a *Client Secret*. These values are needed for the deployment.

If you face any issues while creating Azure Service Principal Client ID, Please [refer](#) Microsoft Official Documentation to create Azure Service Principal Client ID and Client Secret

(Optional) Edit the Network Security Group

Optional: You might need to edit the Azure network security group to add IP addresses that can access the watsonx web client.

Navigate to Load Balancers on your Azure portal and filter on tags, for example *kubernetes.io/service-name: openshift-ingress/router-default*.

1. In Load Balancers, filter and select the security group.

Create load balancer

Subscription *

Resource group * [Create new](#)

Instance details

Name *

Region *

SKU * ☒ Standard ☐ Basic

Microsoft recommends Standard SKU load balancer for production workloads. Learn more about pricing differences between Standard and Basic SKU

Type * ☒ Public ☐ Internal

Tier * ☒ Regional ☐ Global

[Next : Frontend IP configuration >](#)

[Review + create](#) [< Previous](#) [Next : Frontend IP configuration >](#) [Download a template for automation](#) [Give feedback](#)

2. Select Security Group and modify the Inbound rules

vc-az-nsg Network security group

Search (Cmd+/) [Move](#) [Delete](#) [Refresh](#) [Give feedback](#)

Essentials [JSON View](#)

Resource group (Move) **vc-rg** Custom security rules: 1 inbound, 0 outbound

Location **West US 2** Associated with: 0 subnets, 1 network interfaces

Subscription (Move) **Microsoft Azure Enterprise**

Subscription ID: 6f046466-4b41-405b-b34c-c8992178e4fc

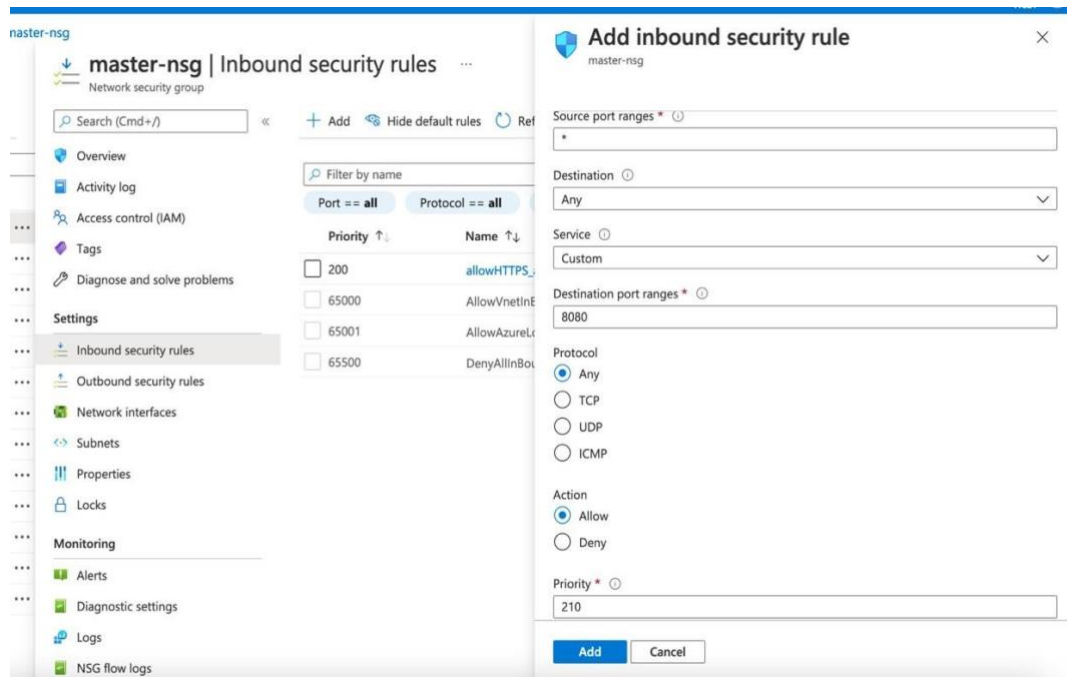
Tags (Edit) [Click here to add tags](#)

Filter by name

Port == all Protocol == all Source == all Destination == all Action == all

Priority	Name	Port	Protocol	Source	Destination
Inbound Security Rules					
Outbound Security Rules					
65000	AllowVnetOutBound	Any	Any	VirtualNetwork	VirtualNet
65001	AllowInternetOutBound	Any	Any	Any	Internet
65500	DenyAllOutBound	Any	Any	Any	Any

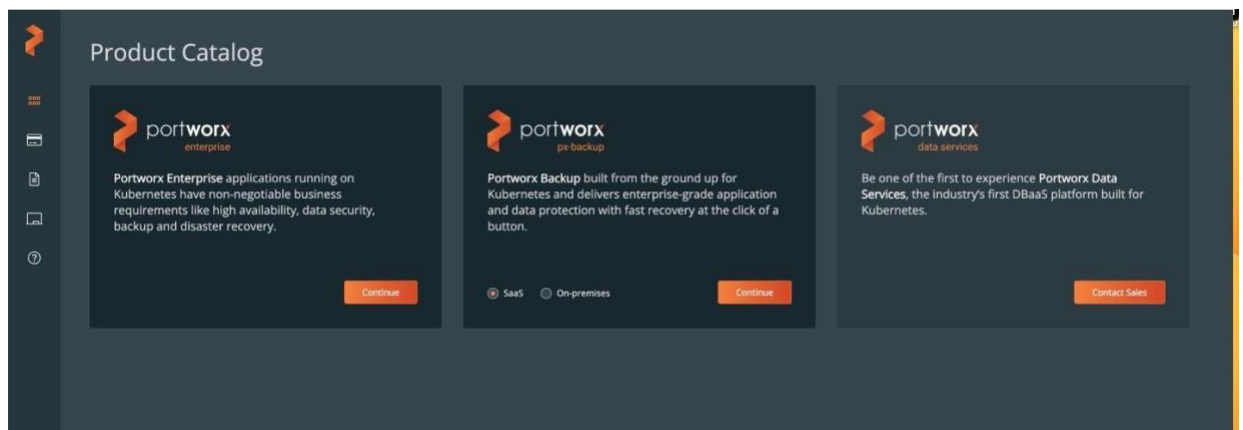
3. Choose **Add Rule** and fill in the rule details. For the rule **Type**, select either HTTP or HTTPS in the drop-down menu. Port 80 or 443 is filled in automatically. Add the network CIDR for the group of IP addresses that you want to permit HTTP or HTTPS access to the proxy nodes. To allow any IP address, use 0.0.0.0/0.



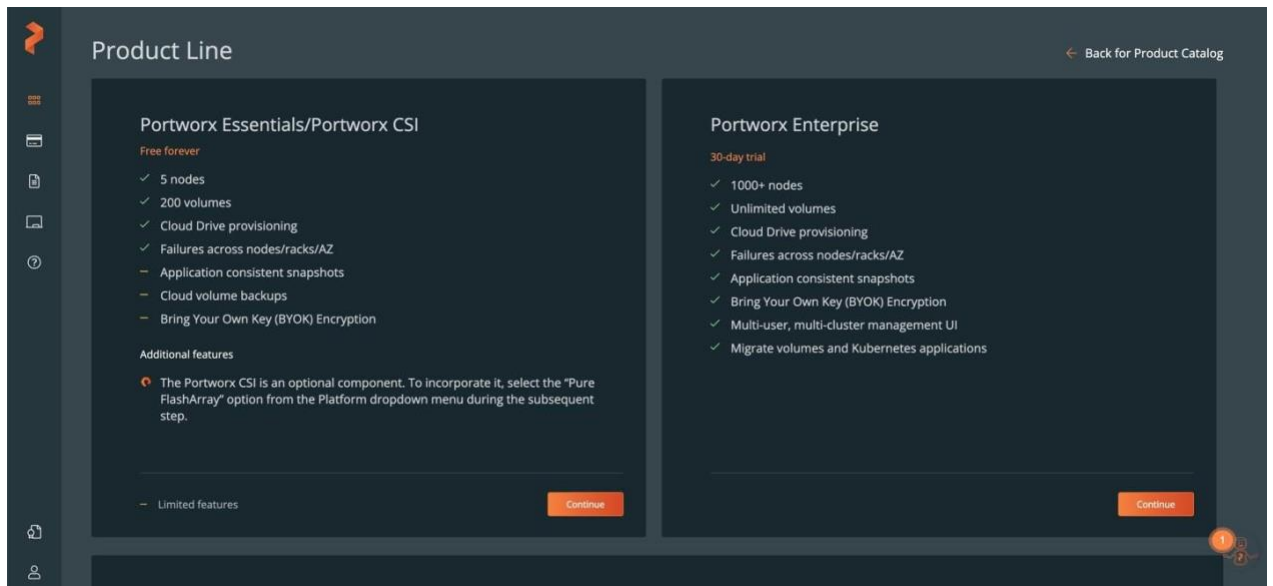
5. In the rule editor window, choose **Save**.

Generating the Portworx Spec URL for ARO cluster

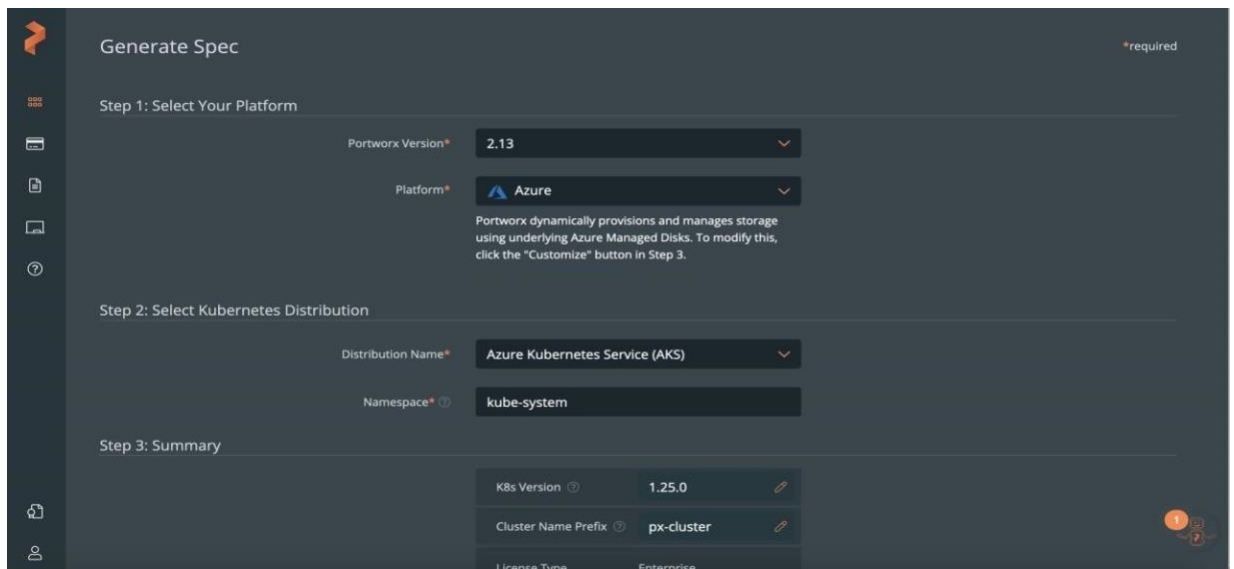
1. Launch the [spec generator](#)
2. Select **Portworx Enterprise** and press *Continue*:



3. Select **Portworx Essentials** or **Portworx Enterprise** and press *Continue*:



4. In generate Spec page, Select Portworx version **3.1**, Select Platform **Azure** and Select Kubernetes Distribution Name **Azure Red Hat OpenShift (ARO)**, Change Namespace name to **kube-system** and press **Customize**



5. Check Use the Portworx Operator box, select the Portworx version as **3.1**, enter Namespace as **kube-system**, select **Built-in** and then press **Next**. (Note: Please make sure all selected options in Step 4 are reflected on this page)

Basic Storage Network Customize

☒ Use the Portworx Operator ⓘ Portworx Operator only supports kubernetes versions 1.12 and up.

Portworx Version * ⓘ 2.13
View release notes

Kubernetes Version * ⓘ 1.25.0

Namespace * ⓘ kube-system
Provide an existing namespace, cannot be blank

etcd * ⓘ ☐ Your etcd details ⓘ ☒ Built-in ⓘ

Portworx will create and manage an internal key-value store (kvdb) cluster.

You can restrict the nodes that will run the key-value store by labelling your nodes with the label `px/metadata-node=true`. Only the nodes with the label will participate in the kvdb cluster. This allows you to use nodes with dedicated hardware for the key-value store.

For example: `kubectl label nodes node1 node2 node3 px/metadata-node=true`

← Back to Product Line Reset Next

6. Select *Cloud* as your environment. Click on *Azure* and select *Create Using a Spec* option for Select type of disk. Enter value for Size (GB) as **500** and then press *Next*.

Basic ✓ Storage Network ✓ Customize ✓

Kubernetes Version: 1.25.0
Built-in etcd

Port Range Start: 9001
Data Interface: auto
Mgmt Interface: auto

Running on

Select your environment * ☐ On Premises ☒ Cloud

Select Cloud Platform * Azure

Configure storage devices

If you are running AKS or your instances are part of VM Scale Sets, it is recommended to use the Portworx ASG feature, where Portworx manages the entire lifecycle of Azure Managed Disks and handles cases like instance termination. For ASG, Portworx will create disks using a spec.

Select type of disk * ☒ Create Using a Spec ⓘ ☐ Consume Unused ⓘ ☐ Use Existing Disks ⓘ

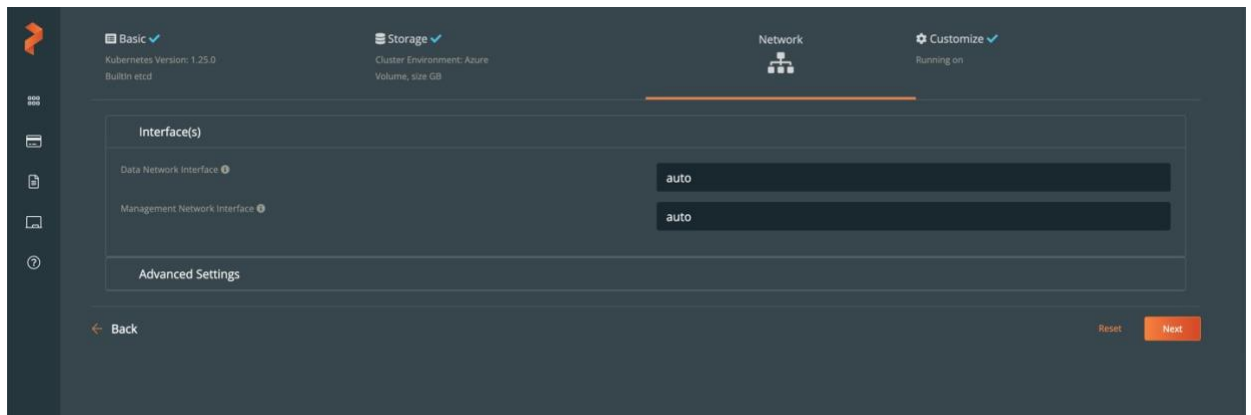
Select volume type * ⓘ Premium
Size (GB) * ⓘ 500
Encryption ⓘ None
Encryption Key ⓘ Not Applicable
Add/Delete Spec Object

Max storage nodes per availability zone (Optional) ⓘ

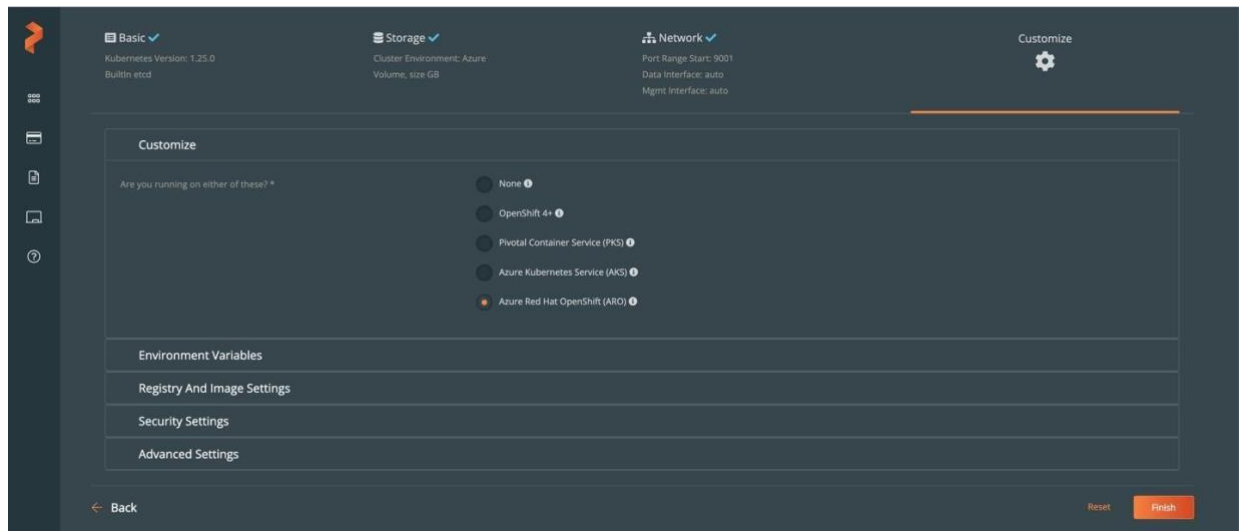
☐ Auto create journal device

← Back Reset Next

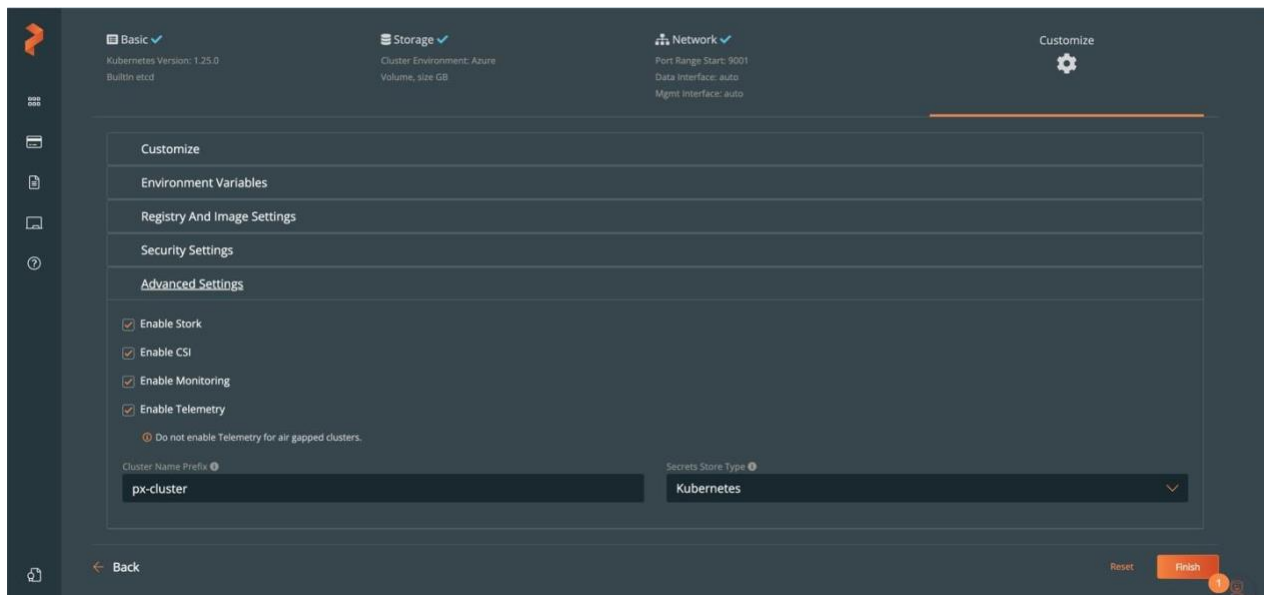
7. Keep the default option *auto* for the network interfaces and press *Next*:



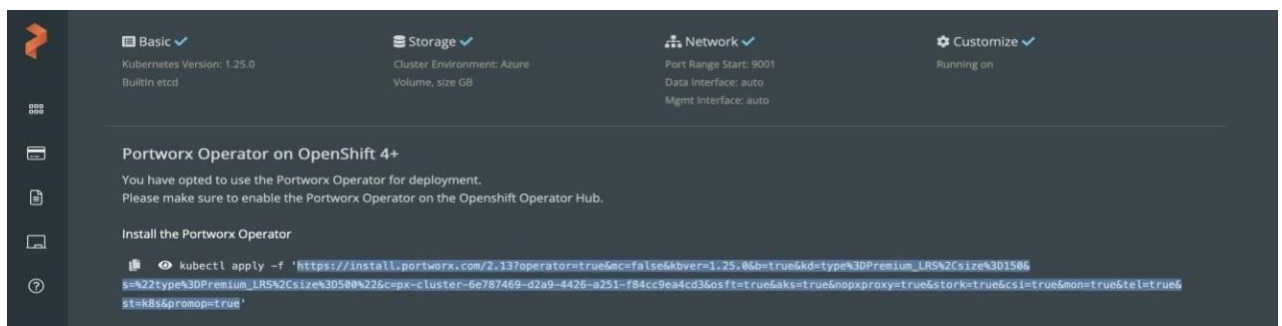
8. Select *Azure Red Hat OpenShift (ARO)* as Openshift environment, go to *Advanced Settings*:
(Note: if you don't find the ARO option in the Customize tab, please verify you have selected Portworx 2.12 or later versions as it is only available for Portworx 2.12 or later versions).



9. In the Advanced Settings tab: *Enable Stork, CSI, Monitoring and Telemetry* and press *Finish*:



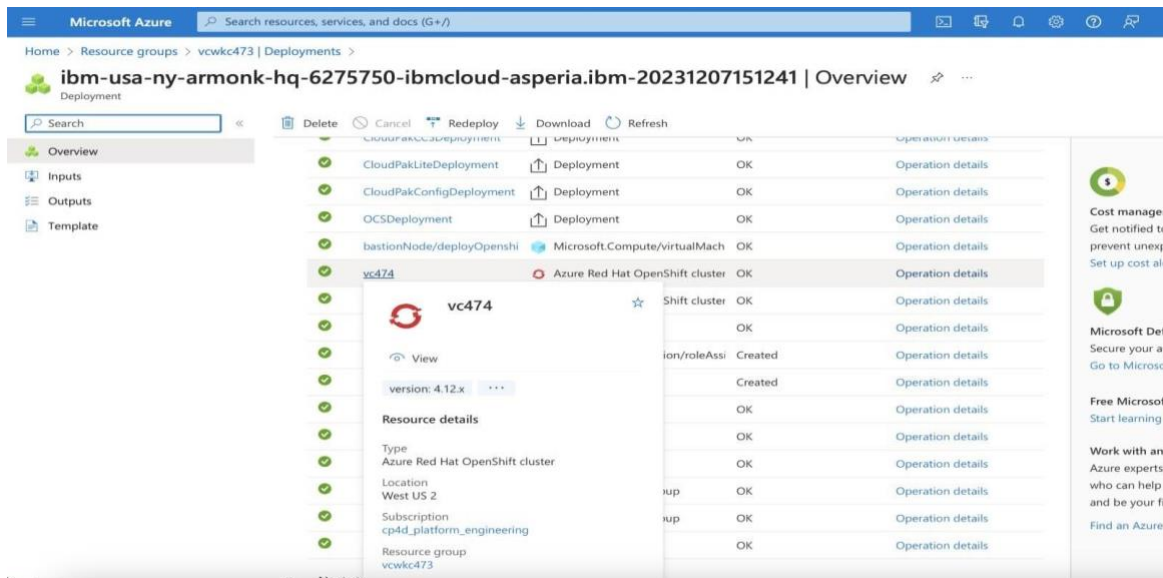
10. Copy the Spec URL and Save Spec in your Portworx account. (Note: Please copy only spec URL but not the whole command or single quote in order to avoid any errors.)



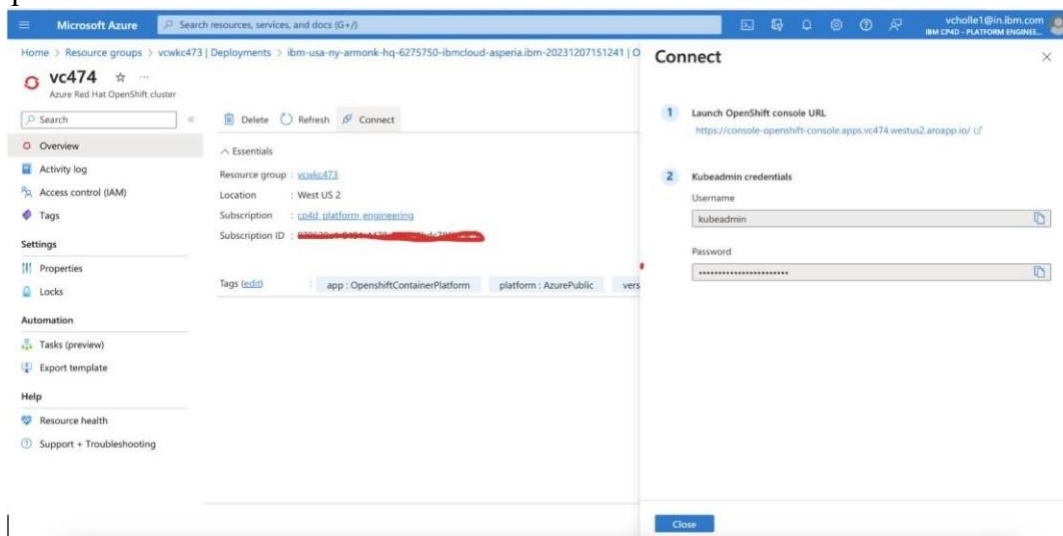
Manage your cluster using the OpenShift Console

1. To access the OpenShift console, go to the overview section of the root stack.
2. In the overview section you can get **OpenShift URL**, **username** and **password**, please select the Azure Red Hat OpenShift cluster deployment from the **overview** section of root stack as shown in the below screenshot:

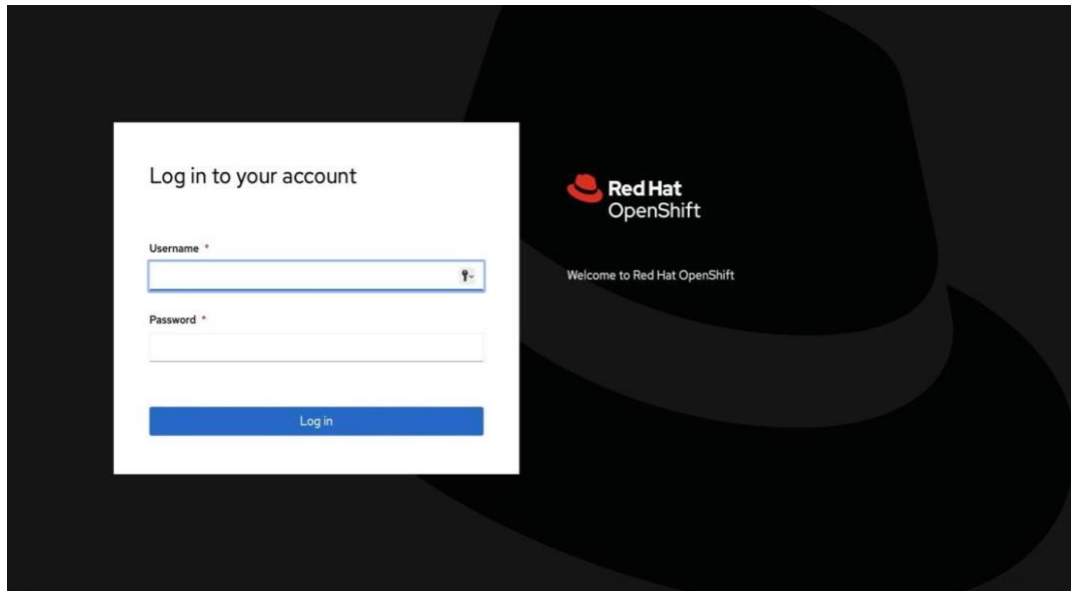
Home > Resource Groups > Cluster's Resource groups > select Deployments > ibm-usa-ny-armonk-hq-6275750-ibmcloud-asperia.ibm-20231207151241 > overview



3. Click on the “Connect” and get the **openshift URL, username & password** from the right panel.



4. Open the OpenShift Console URL in a browser and Login with the **username** and **password** from the previous step.



Login to watsonx web client

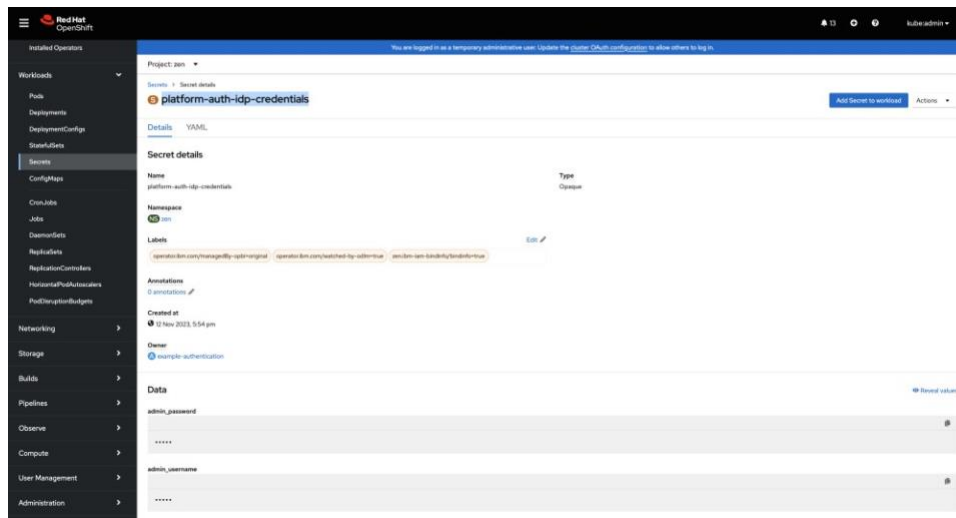
When the Azure ARM template has successfully created the stack, all server nodes will be running with the software installed in your Azure portal. In the following steps, connect to watsonx web client to verify the deployment, and then use the web client to explore watsonx features.

1. To access the watsonx web client, first get the console URL from the output for key **watsonx web URL**

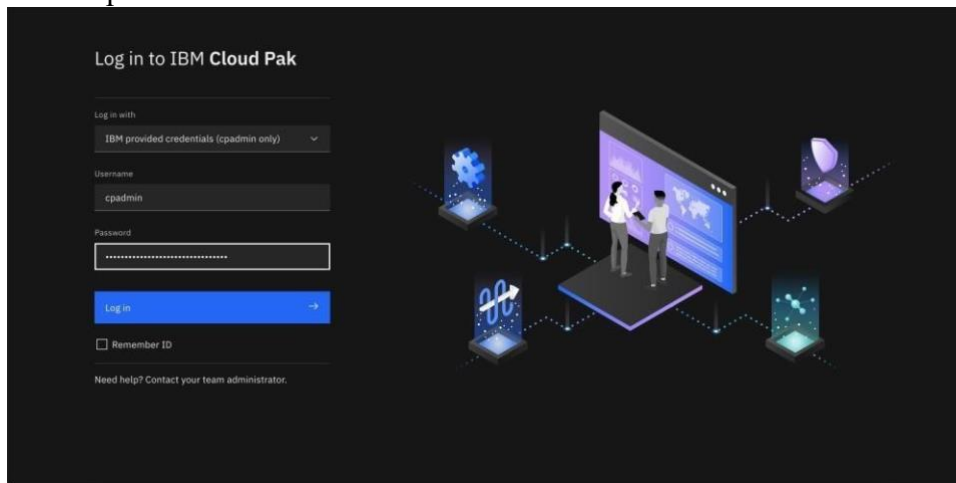
Home > Resource Groups > Cluster's Resource groups > select Deployments > ibm- alliance-global-1560886.cloud_pak_for_data-20211118184734 > Outputs



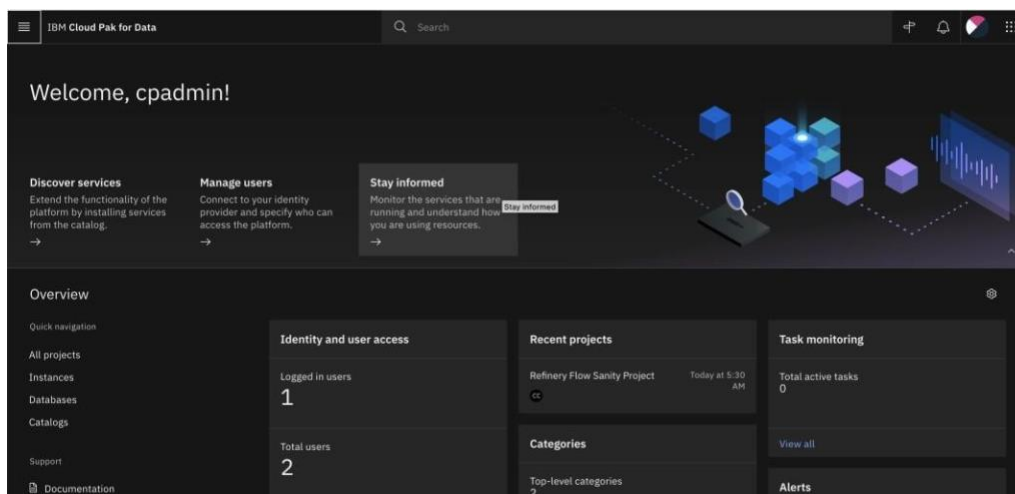
2. To get the password of watsonx web client, please login in to openshift console as mentioned in the previous step-6 and click on the workloads in the left pane **Workloads > secrets > click on platform-auth-idp-credentials > copy the admin_username & admin_password under Data.**



3. Log in to the watsonx web client by using the user “cpadmin” and the password from the previous step.



4. Once you log in, the welcome page is displayed.



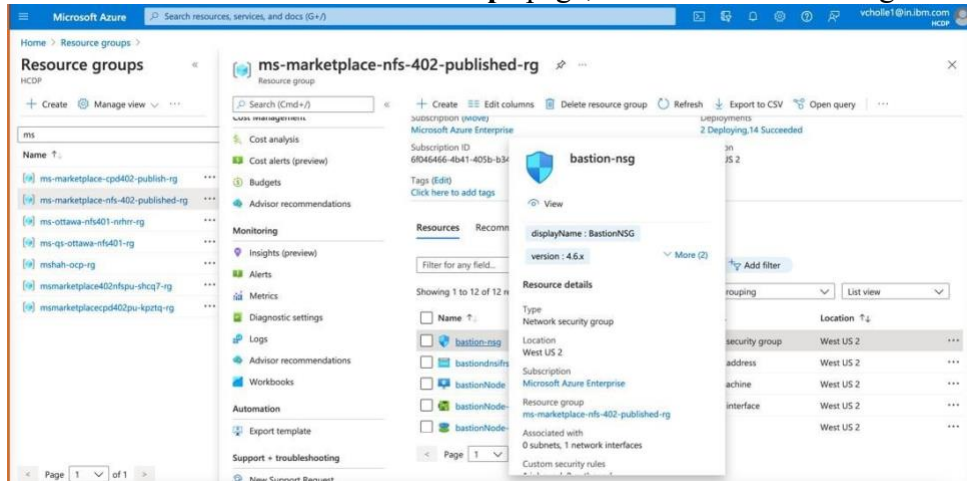
See [resources](#) on platform features and capabilities. For a list of supported browsers, see [Supported browsers](#).

(Optional) Provide Boot Node SSH access

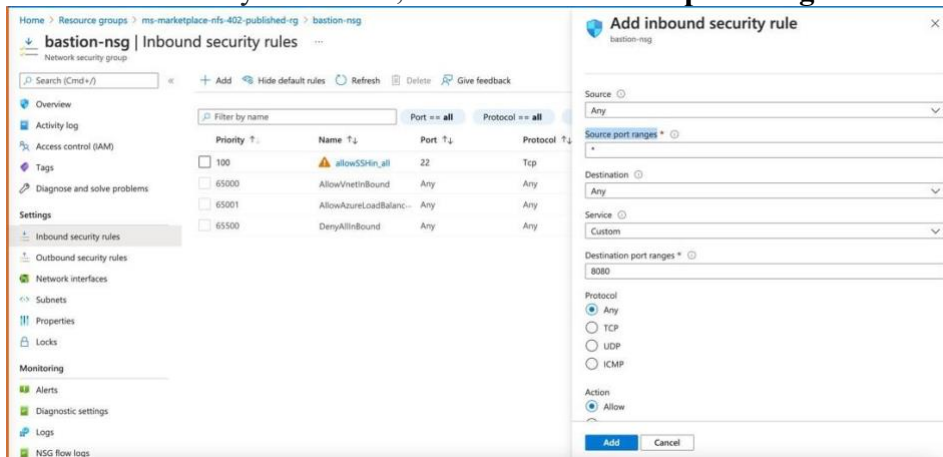
The boot node is used for certain command-line cluster administration tasks, such as adding compute nodes. SSH access to the boot node is required for some cluster administrators. After deployment, you only have access to the boot node. Provide the workstation IP address CIDR as the value of the network security group (nsg) inbound rule.

This section describes the steps to modify the network security group (nsg) inbound rules.

1. In the Azure cluster's **Resource Groups** page, select name containing **bastion-nsg**



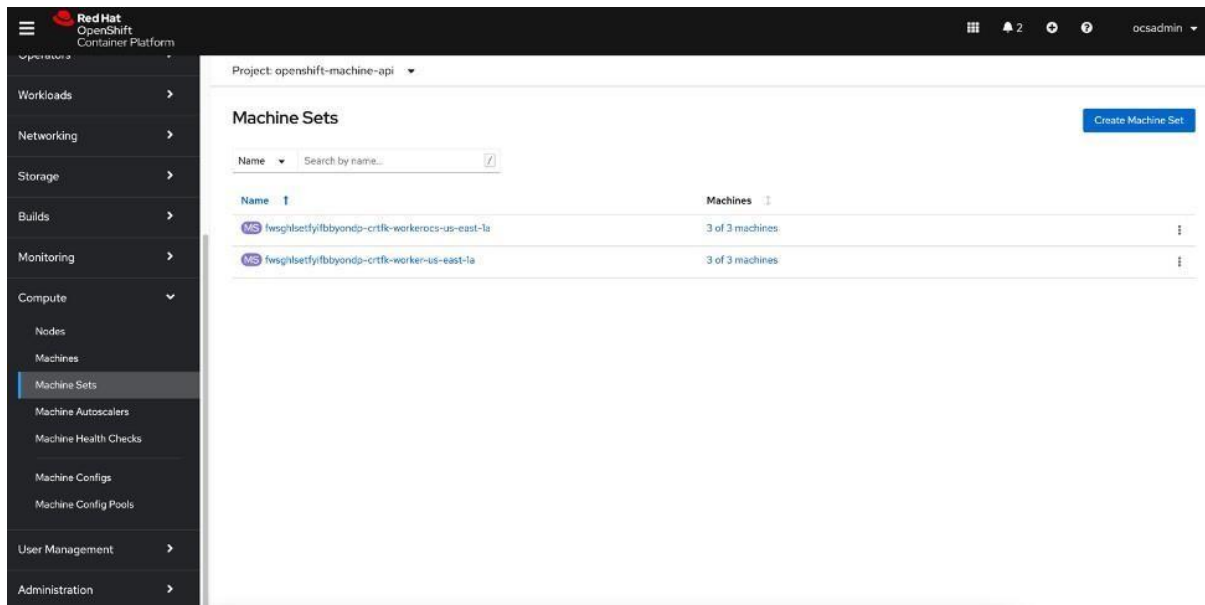
2. The security group window displays the ingress rules. Select the **Inbound** tab, and choose **Edit** to bring up the rule editor, choose **Add Rule**, and fill in the rule details. Add the network CIDR for the group of IP addresses that you want to permit SSH access to the boot nodes. To allow any IP address, use * in the “**Source port ranges**” field.



3. In the rule editor window, click on **Add**.

Scaling up your cluster by adding compute nodes

Login to your OpenShift Console, navigate Compute Machine Sets, each machine set can be scaled up.



- An Azure instance will be created, and Desired count and current count will get updated to replica value.
- After few mins once the node joins the cluster ready and available count will be updated to replica value

Note

1. If you choose to scale down your cluster or reduce the number of compute nodes, there is a risk of the cluster becoming unstable because pods will need to be rescheduled. Scaling down the worker nodes is not a recommended option.
2. Cluster auto scaler can overrule the scaling activity to maintain the required threshold.

Steps for the creation of GPU nodes and configuring it to the openshift cluster for watsonx offerings

Prerequisite:

1. Install az cli from [here](#) and do az login on the environment where following commands are executed
2. Make sure you have Cluster-Admin access to Cluster
3. [Click here](#) to update your pull secret to make sure you can install operators and connect to cloud.redhat.com.

To run the Foundation Models that are part of watsonx.ai, it is necessary to add GPU nodes to the OpenShift cluster.

Before starting, make sure that the subscription in use has a quota for GPU nodes. The GPU quota is set to zero by default. To request the quota on your Azure Cloud account please refer [here](#).

Assuming that your Azure account has sufficient quota for GPU workloads.

1. Choose the GPU node type based on your usage.
2. Set the node type. For example, considering the following GPU node type

```
NODE_TYPE="Standard_NC24ads_A100_v4"
```

3. Get the machineset from the existing ARO cluster and extract the needed values from it.

```
❏ MACHINESET=$(oc get machineset -n openshift-machine-api -o=jsonpath='{.items[0]}' | jq
  -r '[".metadata.name"] | @tsv')❏
❏ oc get machineset -n openshift-machine-api $MACHINESET -o json >
  gpu_machineset.json❏
❏ CLUSTER_LOCATION=eastus # This is the location/region where your ARO cluster is
  running❏
❏ SUBSCRIPTION_ID=$(az account show -o json --query 'id' -o tsv)❏
❏ CLUSTER_ID=$(jq -r '.metadata.labels["machine.openshift.io/cluster-api-cluster"]'
  gpu_machineset.json)❏
❏ OCP_VERSION=$(oc version -o json | jq -r '.openshiftVersion' | awk '{split($0,version,".");
  print version[1],version[2]}' | sed 's/ /.g')❏
❏ RESOURCE_GROUP=$(oc get machineset/${CLUSTER_ID}-worker-❏
  ${CLUSTER_LOCATION}1 -n openshift-machine-api -o
  jsonpath='{.spec.template.spec.providerSpec.value.networkResourceGroup}')❏
❏ export NODE_TYPE="Standard_NC24ads_A100_v4"❏
❏ IMAGE_OFFER=$(oc get machineset/${CLUSTER_ID}-worker-${CLUSTER_LOCATION}1 -n
  openshift-machine-api -o jsonpath='{.spec.template.spec.providerSpec.value.image.offer
  }')❏
❏ IMAGE_VERSION=$(oc get machineset/${CLUSTER_ID}-worker-${CLUSTER_LOCATION}1❏
  -n openshift-machine-api -o
  jsonpath='{.spec.template.spec.providerSpec.value.image.version}{"\n"}')❏
❏ ARO_RESOURCE_GROUP=$(oc get machineset/${CLUSTER_ID}-worker-❏
  ${CLUSTER_LOCATION}1 -n openshift-machine-api -o
  jsonpath='{.spec.template.spec.providerSpec.value.resourceGroup}{"\n"}')❏
❏ VNET_NAME=$(oc get machineset/${CLUSTER_ID}-worker-${CLUSTER_LOCATION}1 -n
  openshift-machine-api -o jsonpath='{.spec.template.spec.providerSpec.value.vnet}{"\n"}')❏
❏ SUBNET_NAME=$(oc get machineset/${CLUSTER_ID}-worker-${CLUSTER_LOCATION}1 -n
  openshift-machine-api -o
  jsonpath='{.spec.template.spec.providerSpec.value.subnet}{"\n"}')❏
```

```
❏ ZONE=$(az vm list-skus -l ${CLUSTER_LOCATION} --resource-type virtualMachines --size $NODE_TYPE --query '[0].locationInfo[0].zones[0]' -o tsv)
```

4. Determine the Gen2 ARO image SKU.

By default, ARO is deployed on Gen1 virtual machines using Gen1 images. All the compatible GPU virtual machine nodes listed above are Gen2 only. It is not possible to use a Gen1 image with a Gen2 virtual machine. This means that the default Gen1 ARO image will not work with these GPU nodes. There are however, Gen2 images available for ARO. The following steps explain how to identify the SKU for ARO gen2 images and use these in the GPU machine set.

Start by getting the current image SKU.

```
oc get machineset/${CLUSTER_ID}-worker-${CLUSTER_LOCATION}1 -n openshift-machine-api -o jsonpath='{.spec.template.spec.providerSpec.value.image.sku}'
```

The output will be similar to below:

aro_412

- Now find all available image SKUs for the same image version and set the appropriate image sku value.

```
❏ CURRENT_SKU=$(oc get machineset/${CLUSTER_ID}-worker-${CLUSTER_LOCATION}1 -n openshift-machine-api -o jsonpath='{.spec.template.spec.providerSpec.value.image.sku}'
```

```
❏ echo $CURRENT_SKU
```

```
❏ IMAGE_VERSION="$(az vm image list --all --publisher azureopenshift -o json | jq -r --arg SKU "$CURRENT_SKU" '.[] | select(.sku==$SKU) | .version')"
```

```
❏ IMAGE_SKU=$(az vm image list --all --publisher azureopenshift -o json | jq -r --arg VERSION $IMAGE_VERSION '.[] | select(.version==$VERSION) | .sku' | grep -v $CURRENT_SKU)
```

```
❏ echo $IMAGE_SKU
```

- Create the `gpu-machineset.yaml` file with the above collected variables.

```

cat > gpu_machineset.yaml <<EOF
apiVersion: machine.openshift.io/v1beta1
kind: MachineSet metadata: labels:
  machine.openshift.io/cluster-api-cluster: ${CLUSTER_ID}
  machine.openshift.io/cluster-api-machine-role: worker
  machine.openshift.io/cluster-api-machine-type: worker
name: ${CLUSTER_ID}-gpu-
${CLUSTER_LOCATION}${ZONE} namespace: openshift-
machine-api spec: replicas: 1 selector:
  matchLabels: machine.openshift.io/cluster-api-cluster:
    ${CLUSTER_ID}
  machine.openshift.io/cluster-api-machineset: ${CLUSTER_ID}-gpu-
${CLUSTER_LOCATION}${ZONE}
template:
  metadata:
    labels:
      machine.openshift.io/cluster-api-cluster: ${CLUSTER_ID}
      machine.openshift.io/cluster-api-machine-role: worker
      machine.openshift.io/cluster-api-machine-type: worker
      machine.openshift.io/cluster-api-machineset: ${CLUSTER_ID}-gpu-
${CLUSTER_LOCATION}${ZONE}
  spec:
    taints:
      - key: watsonxai value:
        gpunode effect:
          PreferNoSchedule
    lifecycleHooks: {}
    metadata: {}
    providerSpec:
      value:
        acceleratedNetworking: true apiVersion:
        azureproviderconfig.openshift.io/v1beta1
        credentialsSecret: name: azure-cloud-credentials
        namespace: openshift-machine-api
        diagnostics: {} image:
        offer: ${IMAGE_OFFER}
        publisher: azureopenshift

```

```

resourceID:      ""      sku:
${IMAGE_SKU}     version:
${IMAGE_VERSION}
kind:
AzureMachineProviderSpec
location: ${CLUSTER_LOCATION}
metadata:  creationTimestamp:
null
networkResourceGroup:
${RESOURCE_GROUP} osDisk: diskSettings: {}
diskSizeGB: 128 managedDisk:
  storageAccountType: Premium_LRS
  osType: Linux
publicIP: false
publicLoadBalancer:      ${CLUSTER_ID}
resourceGroup:
${ARO_RESOURCE_GROUP}      subnet:
${SUBNET_NAME} userDataSecret: name:
worker-user-data
vmSize: ${NODE_TYPE}
vnet: ${VNET_NAME} zone:
"${ZONE}"
EOF

```

7. Create a gpu machineset on the cluster by using the following command.

```
oc create -f gpu_machineset.yaml
```

Hint: Refer this Redhat [doc](#) to get the machineset file and update according to your cluster details just to avoid syntax errors.

8. Check the machineset and nodes status, you should see the new node in a running state

```
oc get nodes
oc get machinesets
```

9. Complete the below two steps for GPU nodes setup:

- [Install the Node Feature discovery Operator](#)
- [Install the GPU Operator](#)

Please refer [here](#) for any queries related to adding any infrastructure nodes to existing ARO cluster.

After the successful completion of the above steps, Go to project “zen”. Execute the below commands

```
oc patch watsonxai watsonxai-cr --type='json' -p='[{"op":  
"replace", "path": "/spec/tuning_disabled", "value": false}]'
```

Check the CR status of watsonx.ai service by executing this command:

```
oc get watsonxai
```

Example:

```
NAME VERSION RECONCILED STATUS AGE watsonxai-cr 8.3.0 8.3.0  
InProgress 60m
```

The CR status should be “InProgress”. After 15 to 20 minutes, The CR status should be back to “Completed” state. Execute the below command to check the CR status of watsonx.ai service.

Example:

```
$ oc get watsonxai  
NAME VERSION RECONCILED STATUS AGE watsonxai-cr 8.1.0 8.1.0  
Completed 9h
```

watsonx.ai services

You can browse the various services that are available for use by navigating to the [watsonx service catalog](#).

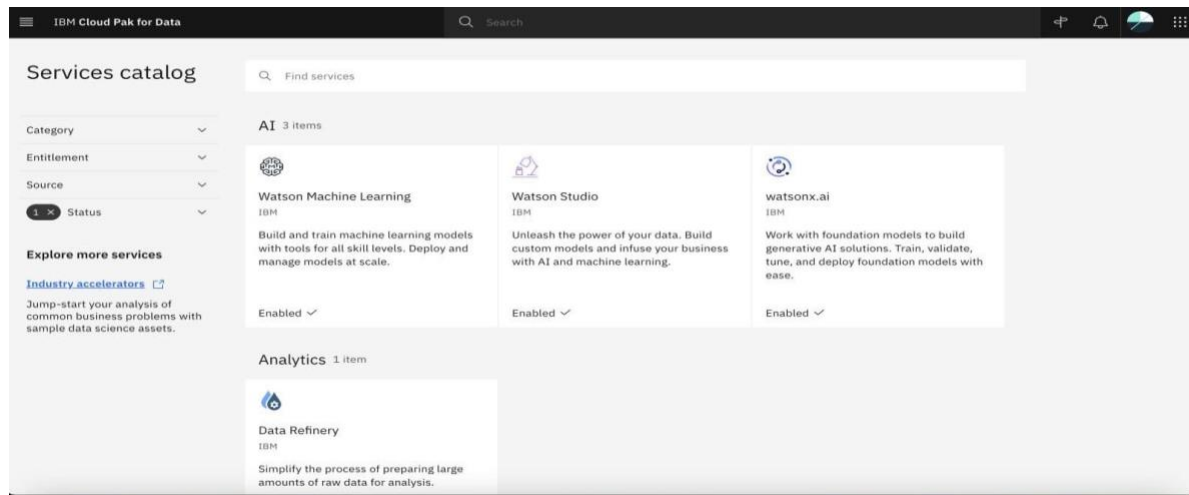


Figure: Services catalog page in watsonx

As part of this installation, the control plane is installed by default

watsonx.ai System and Services requirements

The system requirements for the components/services can be found at [Installing IBM watsonx.ai](#)

Steps to Install any other watsonx Service (Not available on Azure Marketplace template)

- Login into your bootnode server
 - Select your {CLUSTER_NAME} Resource Group from Azure Portal > Click on “Overview” page from the left panel > Click on “bastionNode” > Click on “Connect” from Overview page > Follow SSH command to bastionNode.
- Follow the [Prerequisite and Install steps for the selected service](#)
 - Downloaded CPD-CLI file location in the bootnode: “/mnt/cpd/templates/cpd-cli”
 - Here are necessary params value which might be useful for the CPD-CLI commands:
 - \${OCP_USERNAME}: Select your {CLUSTER_NAME} Resource Group from Azure Portal > Click on “Overview” page from the left panel > Click on your Azure Red Hat OpenShift Cluster > Click on “Connect” from the Overview page > Kubeadmin credentials.
 - \${OCP_PASSWORD}: Select your {CLUSTER_NAME} Resource Group from Azure Portal > Click on “Overview” page from the left panel > Click on your Azure Red Hat OpenShift Cluster > Click on “Connect” from the Overview page > Kubeadmin credentials.
 - \${OCP_URL}: Select your {CLUSTER_NAME} Resource Group from Azure Portal > Click on “Overview” page from the left panel > Click on your Azure Red Hat OpenShift Cluster > “API Server URL” from the Overview Page
 - \${VERSION}: “cpd-version” from “/mnt/cpd/templates/config.json” file path
 - \${PROJECT_CPD_INST_OPERATORS}: value “cpd-operator”
 - \${PROJECT_CPD_INST_OPERANDS}: value “zen” (Default value is “zen”, unless it is being changed during deployment time.)
 - \${STG_CLASS_BLOCK}: For ODF storage, value “ocs-storagecluster-ceph-rbd”
 - \${STG_CLASS_FILE}: For ODF storage, value “ocs-storagecluster-cephfs”

Note: These steps require knowledge of CPD CLI commands and Linux command line. If you need assistance for deployment, please create a support Ticket if you require assistance for any question. Support Links can be found in the [Learn more section of the Overview Page](#).

Upgrade watsonx services

See what new features and improvements are available in the [latest release of watsonx.ai](#)

- Login to your bootnode server.
 - Follow the [Operator Subscriptions upgrade instructions](#) for the services you are interested to upgrade.□
 - Apply upgrades to the custom resources for the [service](#) that you are interested in.□

Limitations

- Review the [known issues and limitations](#) for watsonx.ai□

Additional resources

Azure resources

- [Getting Started Resource Center](#)
- [Azure General Reference](#)

Azure services

- [Azure VM](#)□
- [Azure DNS](#)□
- [Azure Resource groups](#)

IBM Watsonx.ai documentation□

- [IBM Documentation](#)□
- [Red Hat OpenShift Container Platform](#)□

Document revisions

Date	Change
July 2025	Updated ODF v4.17 and watsonx v5.2.x
May 2025	Updated ARO version to 4.16
December 2024	Updated ARO version and IBM watsonx documentation with all latest v5.1.x references

July 2024	Updated ARO version and IBM watsonx with latest version references
May 2024	Updated IBM watsonx with latest version references
March 2024	Updated IBM watsonx version 4.8.x