Creating a CI Pipeline with Compliance

Table of Contents

Edit topic

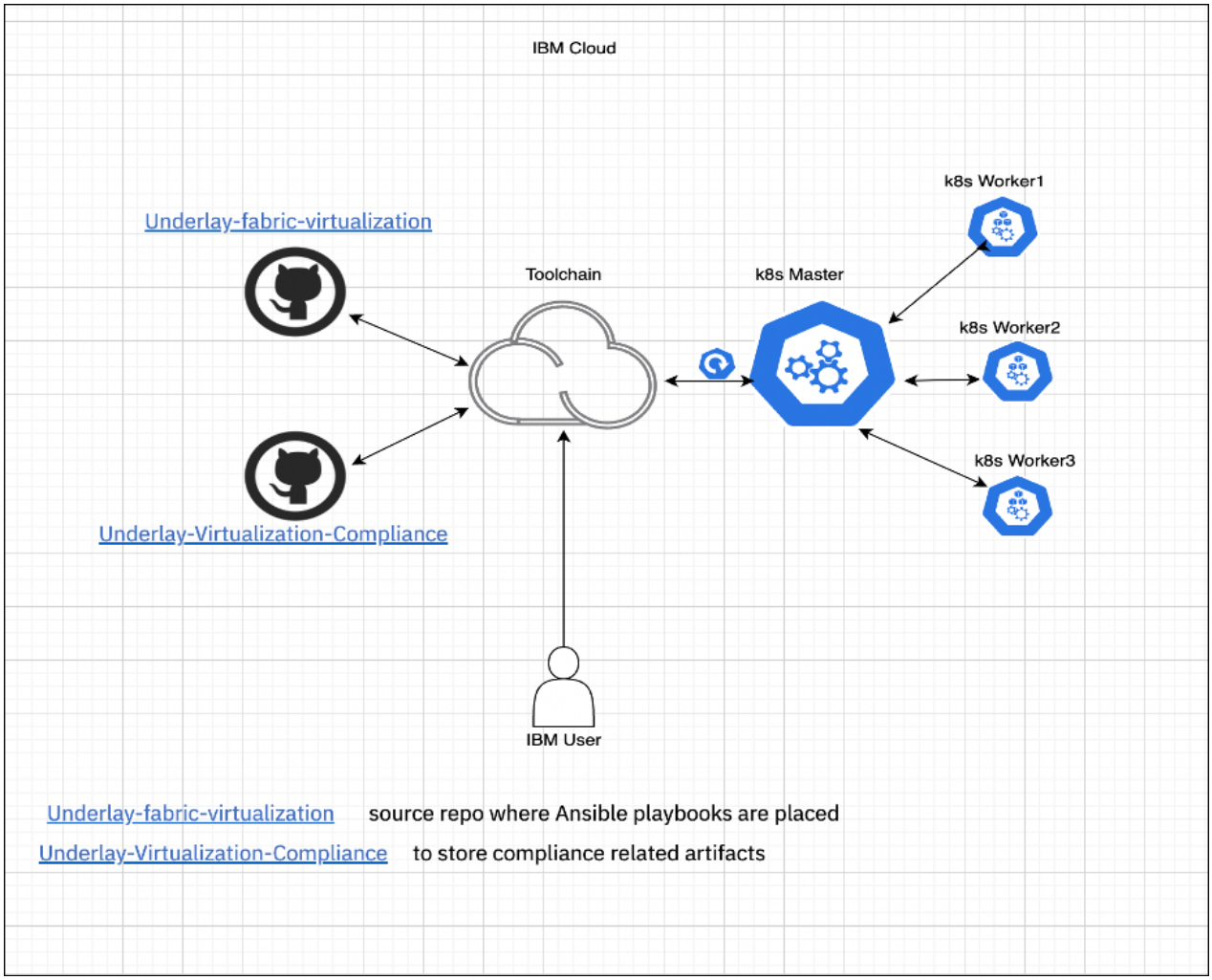
Overview

This document describes the procedure to create a Continuous Integration (CI) toolchain which deploys the plugins and resources on the destination Kubernetes cluster.
The plugins that are supposed to be deployed on Kubernetes cluster are stored in a repo as playbooks and these playbooks will be pulled by a toolchain as defined by meeting the compliance policies. Deliver a secure and compliant app to a Kubernetes cluster based on DevSecOps best practices and Continuous Integration (CI).

Toolchain

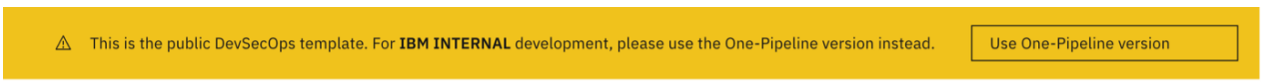
A series of tools that are integrated for development, deployment, and operating tasks is called a toolchain. Users can create toolchains that include IBM services, open-source tools, or third-party tools. Toolchains make the development and operations easier to manage, and enable issue tracking, source control, monitoring, and notifications. In other words, a suite of tools forms a toolchain. Refer to IBM's OnePipeline Documentation for more information.

Virtualization Toolchain Flow Diagram



Prerequisites:

- Get access to the following repos: - Underlay-fabric-virtualization: the source repo where the Ansible playbooks are placed. -Underlay-Virtualization-Compliance: the Org to store compliance related artifacts.
- Create a repo in Underlay-Virtualization-Compliance with proper naming conventions. For example, **kubevirt-compliance-Inventory**
  - Add ul_fabricvmaas_prod_rw as Collaborators for the newly created repo.
- In IBM console, switch to **1145597 - Nextgen-Fabric** account.
  - Navigate to **DevOps** and then select **Toolchains**.
  - Choose these field and values: Resource Group: nextgen-vmaas
    Location: Dallas
  - Click on **Create Toolchain** and select **CI- Develop a secure app with DevSecOps practices**
  - The following Pop-up appears, choose **One-Pipeline version** for Internal IBM Employees.



- Now, the Toolchain template will open.
  1. This template uses a guided experience. You will be asked, step by step, what tools should be included in your toolchain.
  2. In most steps you will be asked to name each tool as they will appear in your toolchain.
  3. The toolchain created from this template creates artifacts and evidence to be used with an associated Continuous Deployment toolchain.

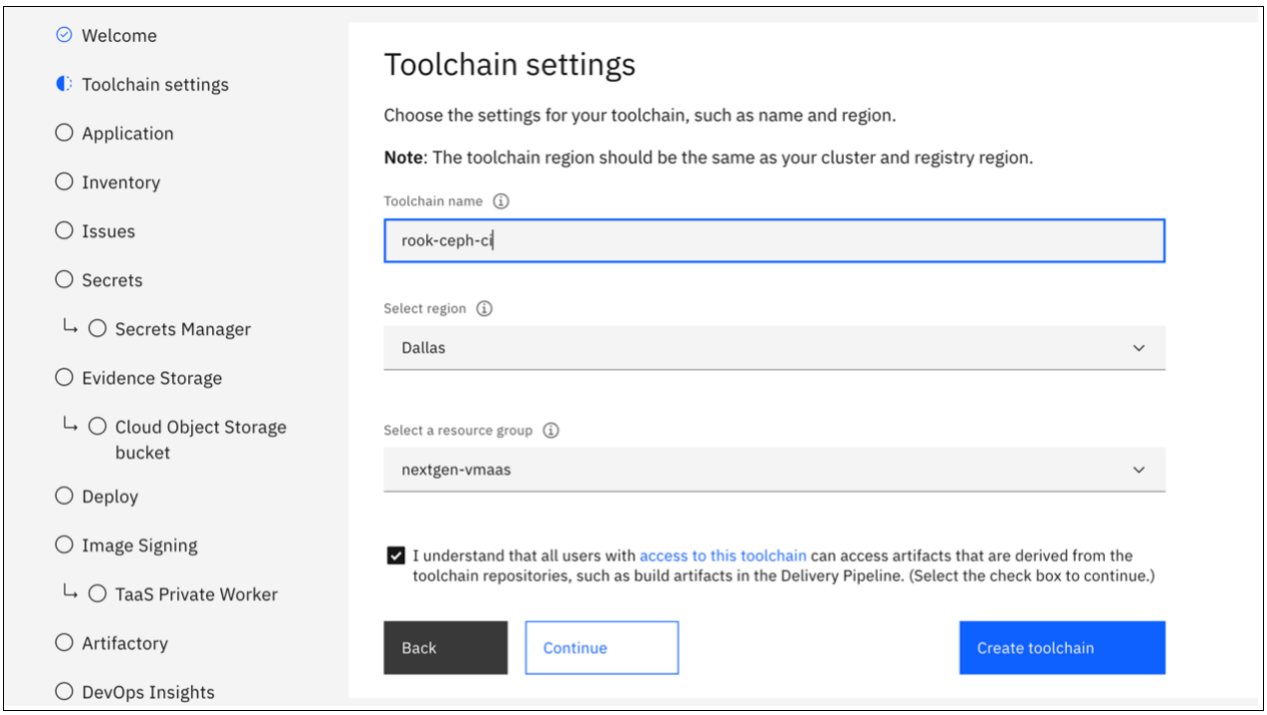
Process

Initiate the toolchain creation with the **Start** button and follow these steps:

Toolchain settings

Note: The toolchain region should be the same as your cluster and registry region.

- **Toolchain name**: “Please provide the name for the toolchain”
- **Select region**: Dallas (Select Dallas, where your toolchain to be created).
- **Select a resource group**: nextgen-vmaas (Select our group from the drop down).
- Click **Continue**

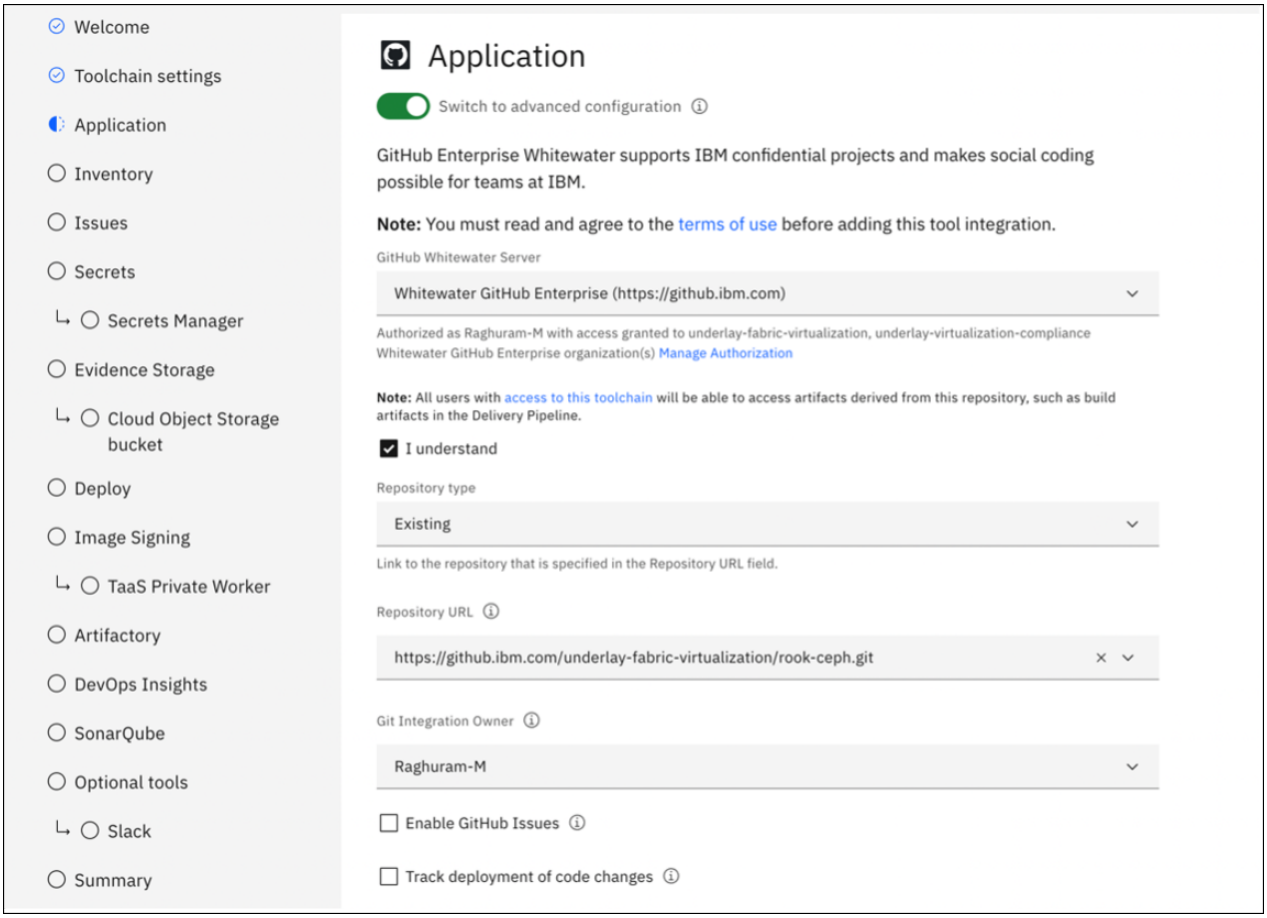


Note: The Git integration owner should be the VMaaS functional ID (FID) ‘ul-fabricvmaas-prod-rw’. However, if the toolchain is being created under another user’s account and this is not an option, it can be changed later. You can request the functional ID from a team admin with access to the FID’s login credentials.

Application

The application repository is the project or service source code that the CI toolchain and pipeline builds, tests, and delivers to target artifact registries. The repository also contains configuration YAML files and scripts for running the custom pipeline tasks.

- Select **Switch to advanced configuration**
- **Repository type**: Existing (Select Existing from drop down)
- **Repository URL**: Type the URL of the repository that you are linked to. (For example, https://github.ibm.com/underlay-fabric-virtualization/rook-ceph.git)

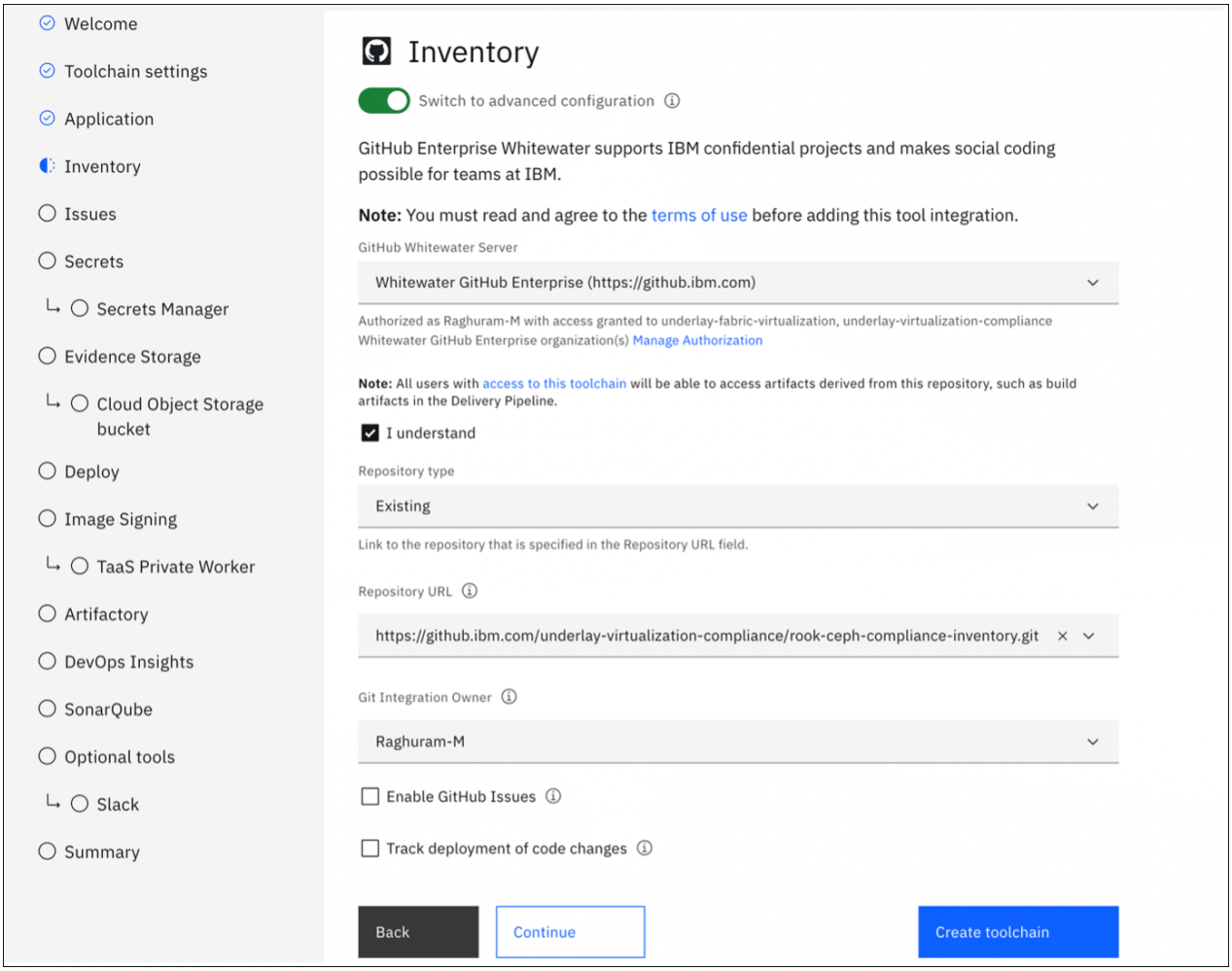


- Click **Continue**.

Inventory

The inventory repository records details of artifacts that are built by the CI toolchains.

- Select **Switch to advanced configuration**
- **Repository type**: Existing
- **Repository URL**: Type the URL from the repo Underlay-Virtualization-Compliance (For example, https://github.ibm.com/underlay-virtualization-compliance/rook-ceph-compliance-inventory.git)



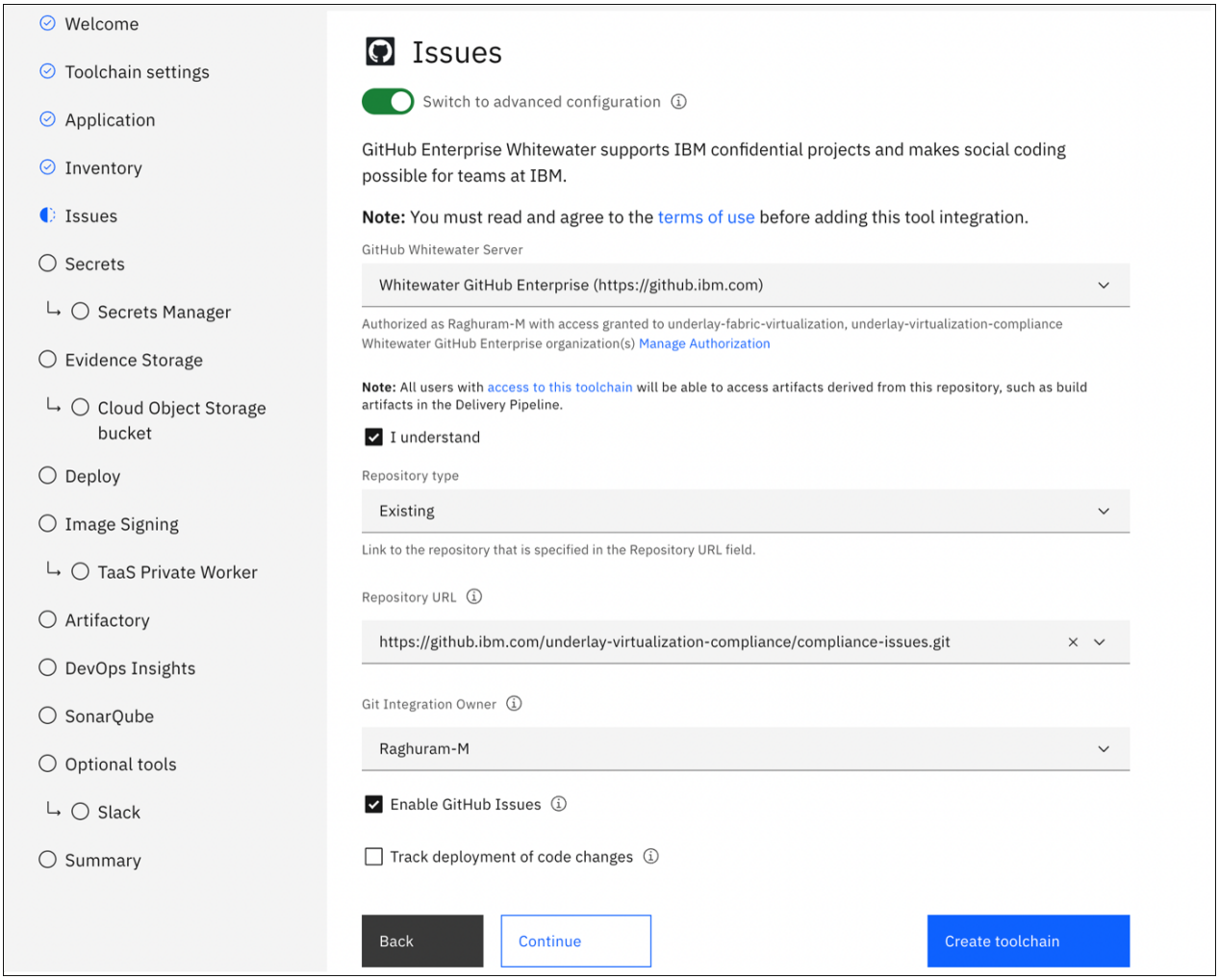
Note: The GIt integration owner should be our functional ID ‘ul-fabricvmaas-prod-rw’. However, if the toolchain is being created under another user’s account and this is not an option, it can be changed later. You can request the functional ID (FID) from a team admin with access to the FID’s login credentials.

- Click **Continue**.

Issues

The GitHub repository records issues that are found while the CI pipeline is running.

- Select **Switch to advanced configuration**
- **Repository type**: Existing
- **Repository URL**: https://github.ibm.com/underlay-virtualization-compliance/compliance-issues.git
- Select **Enable GitHub Issues** checkbox.



Note: The GIt integration owner should be our functional ID ‘ul-fabricvmaas-prod-rw’. However, if the toolchain is being created under another user’s account and this is not an option, it can be changed later. You can request the functional ID (FID) from a team admin with access to the FID’s login credentials.

- Click **Continue**.

Secrets

Select **Key Protect** and **Secrets Manager** tools

The VMaaS team is using the following Key Protect values:

Field	Value
Name	kp-compliance-secrets
Region	Dallas
Resource Group	Default
Service Name	Key-Protect-NG

Keys used for this toolchain are:

Key	Value
product-type	prd
gara-signing-key	sm-compliance-secrets.nextgen-vmaas.GaraSign_Code_Signing_Key

gara-signing-credential	sm-compliance-secrets.nextgen-vmaas

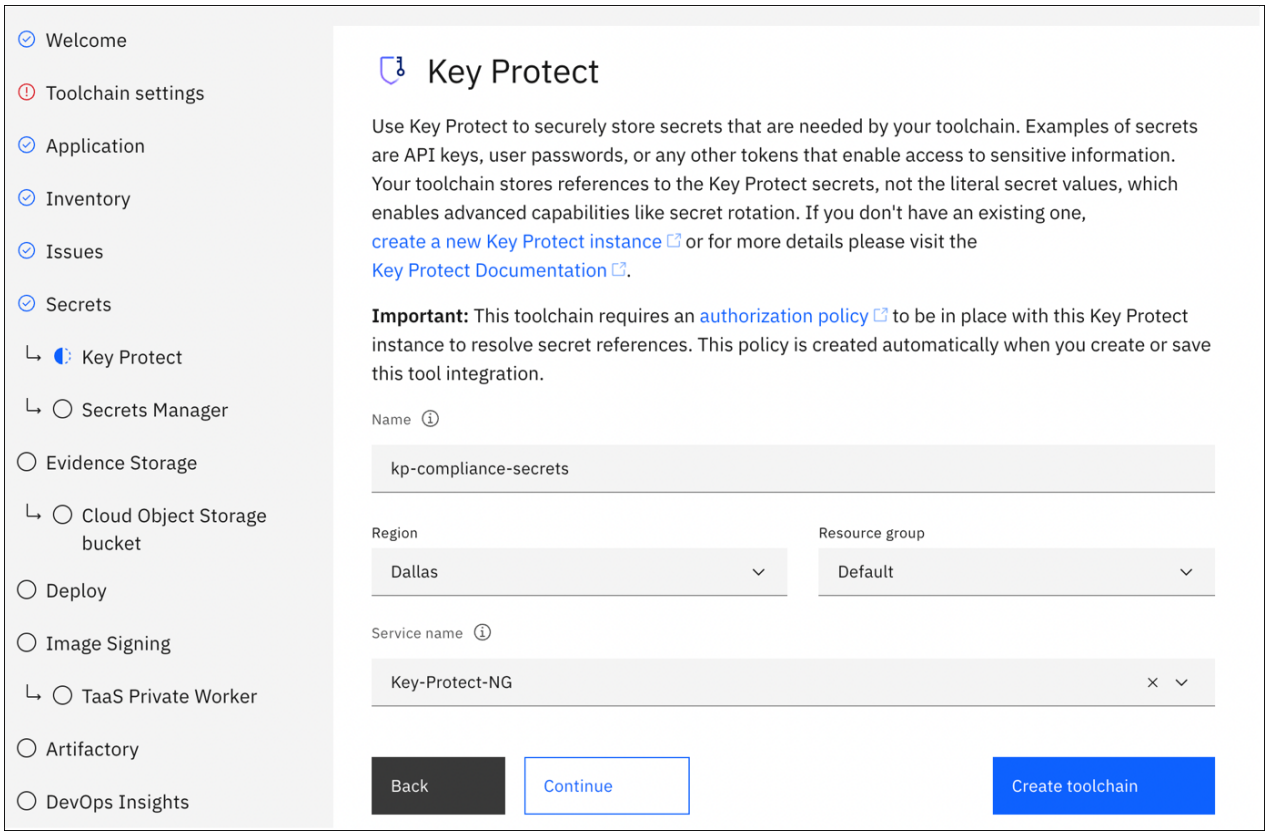
Secrets from Secrets Manager are:

Secrets	Description
ngfabric_ibmcloud_apikey_ul-fabricvmaas-prod-rw	An API key from the IBM Cloud account for functional ID ul-fabricvmaas-prod-rw (IBM Cloud API Key)
Tekton_Private_Worker_API_Key	Tekton Private API Key for TaaS Private Worker
artifactory_apikey_ul-fabricvmaas-prod-rw	API Key for Artifactory NA instance for the functional ID ul-fabricvmaas-prod-rw.
COS-API-Key	Service ID API Key for Cloud Object Storage bucket

Key Protect

Securely store and provide secrets for apps across IBM Cloud services. IBM Key Protect for IBM Cloud is a service for managing cryptographic keys, which are used to protect data. Use Key Protect to securely store secrets that are needed by your toolchain.

Examples of secrets are API keys, user passwords, or any other tokens that enable access to sensitive information



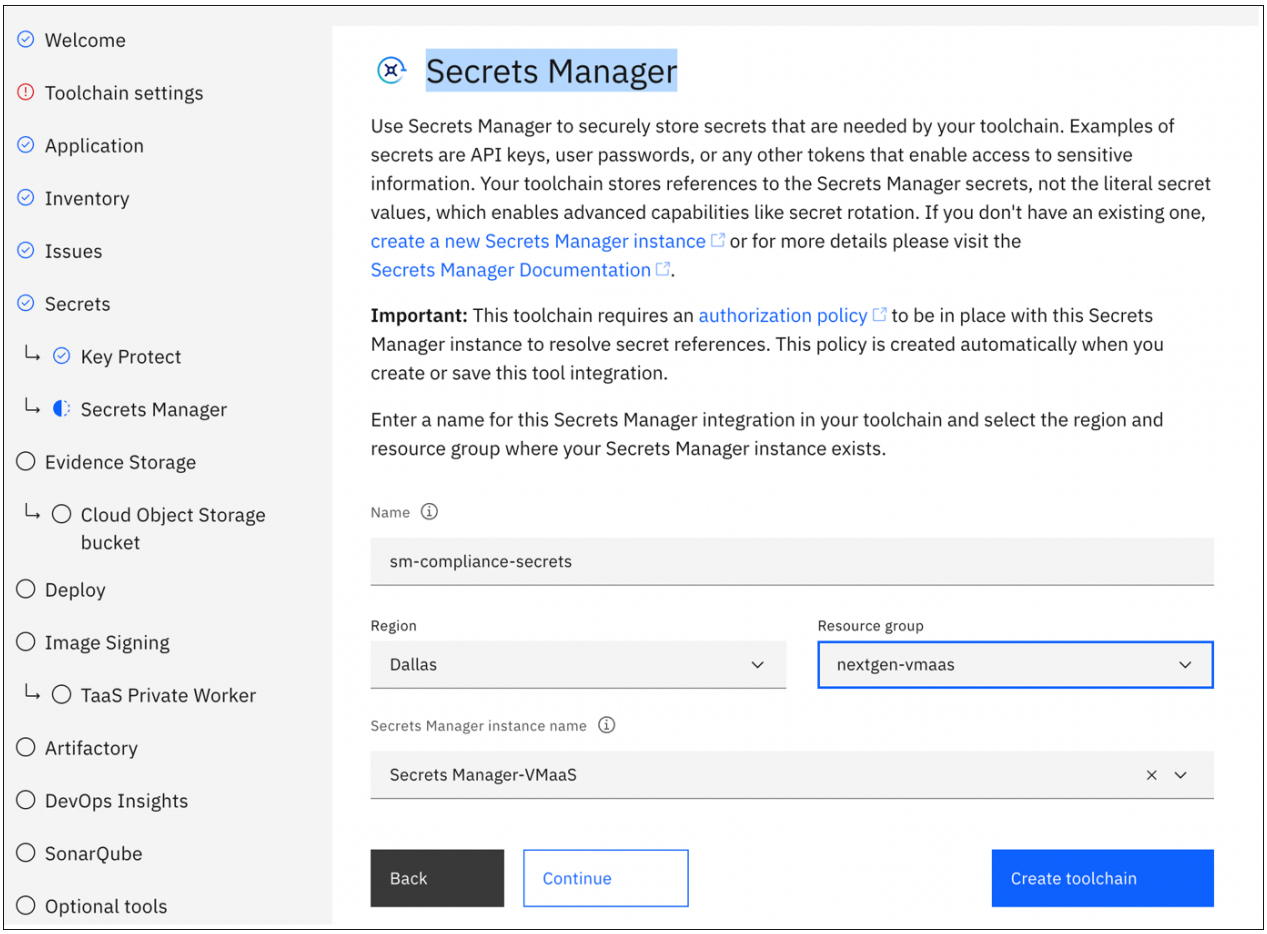
Secrets Manager

Securely store and apply secrets for apps across IBM Cloud services.

Use **Secrets Manager** to securely store secrets that are needed by your toolchain. Examples of secrets are API keys, user passwords, or any other tokens that enable access to sensitive information. Your toolchain stores references to the **Secrets Manager** secrets, not the literal secret values, which enables advanced capabilities like secret rotation.

- **Name**: sm-compliance-secrets

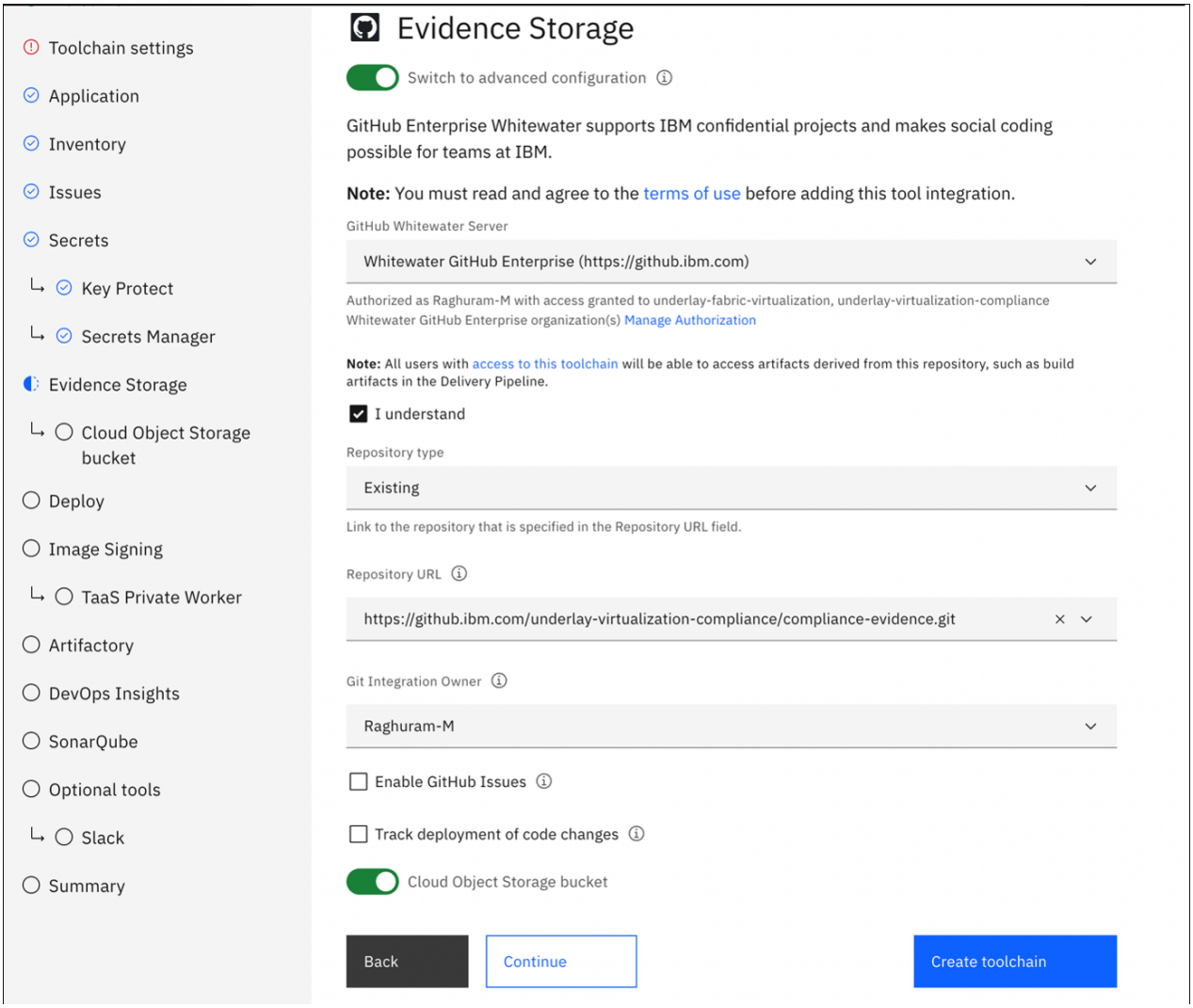
- **Region**: Dallas
- **Resource group**: nextgen-vmaas
- **Secrets Manager instance name**: Secrets Manager-VMaaS



Evidence Storage

The evidence repository records the summary of all the evidence that is collected during the CI process. This evidence is collected by the pipeline during the pipeline run.

- Select **Switch to advanced configuration**
- **Repository type**: Existing
- **Repository URL**: https://github.ibm.com/underlay-virtualization-compliance/compliance-evidence.git



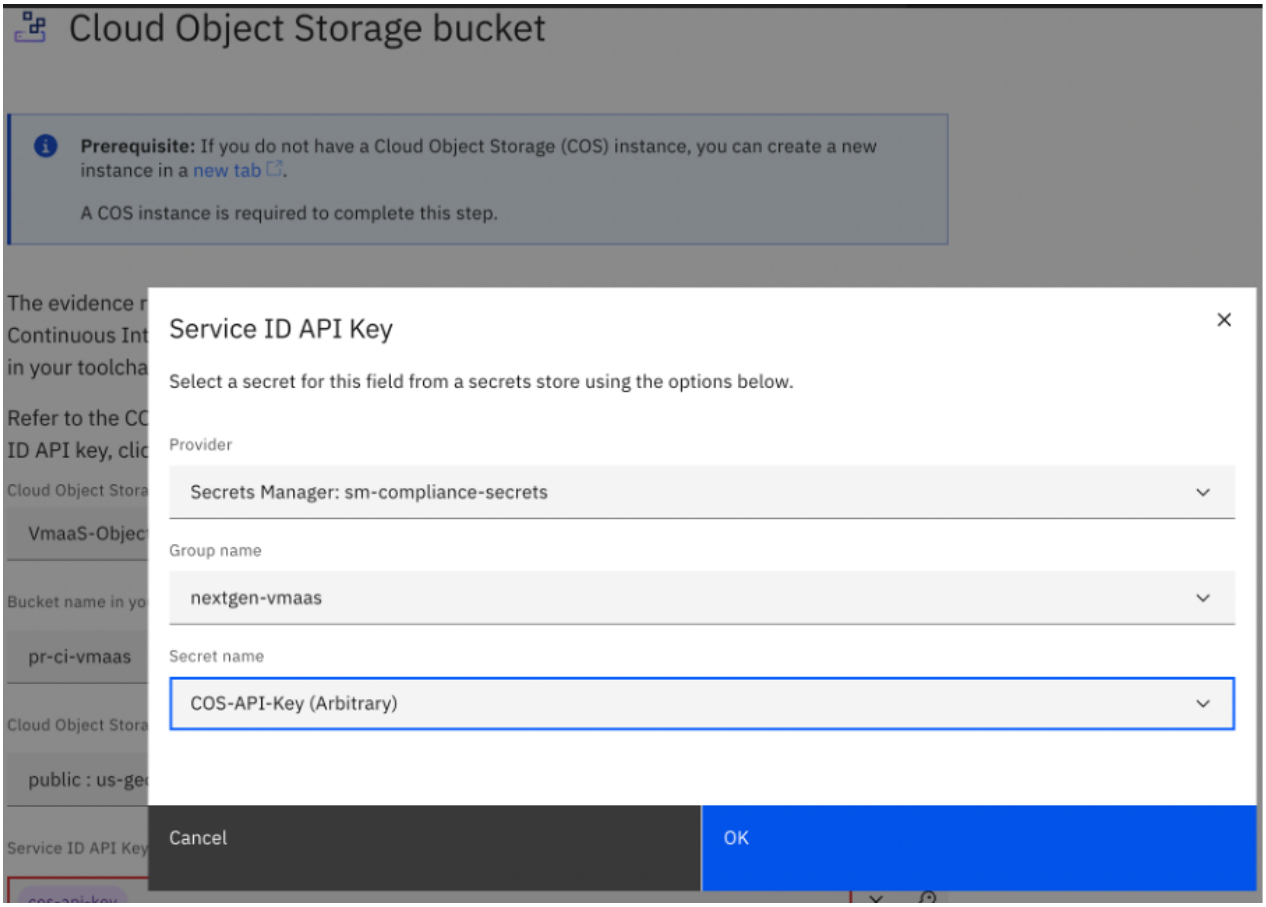
Note: The GIt integration owner should be our functional ID ‘ul-fabricvmaas-prod-rw’. However, if the toolchain is being created under another user’s account and this is not an option, it can be changed later. You can request the functional ID (FID) from a team admin with access to the FID’s login credentials.

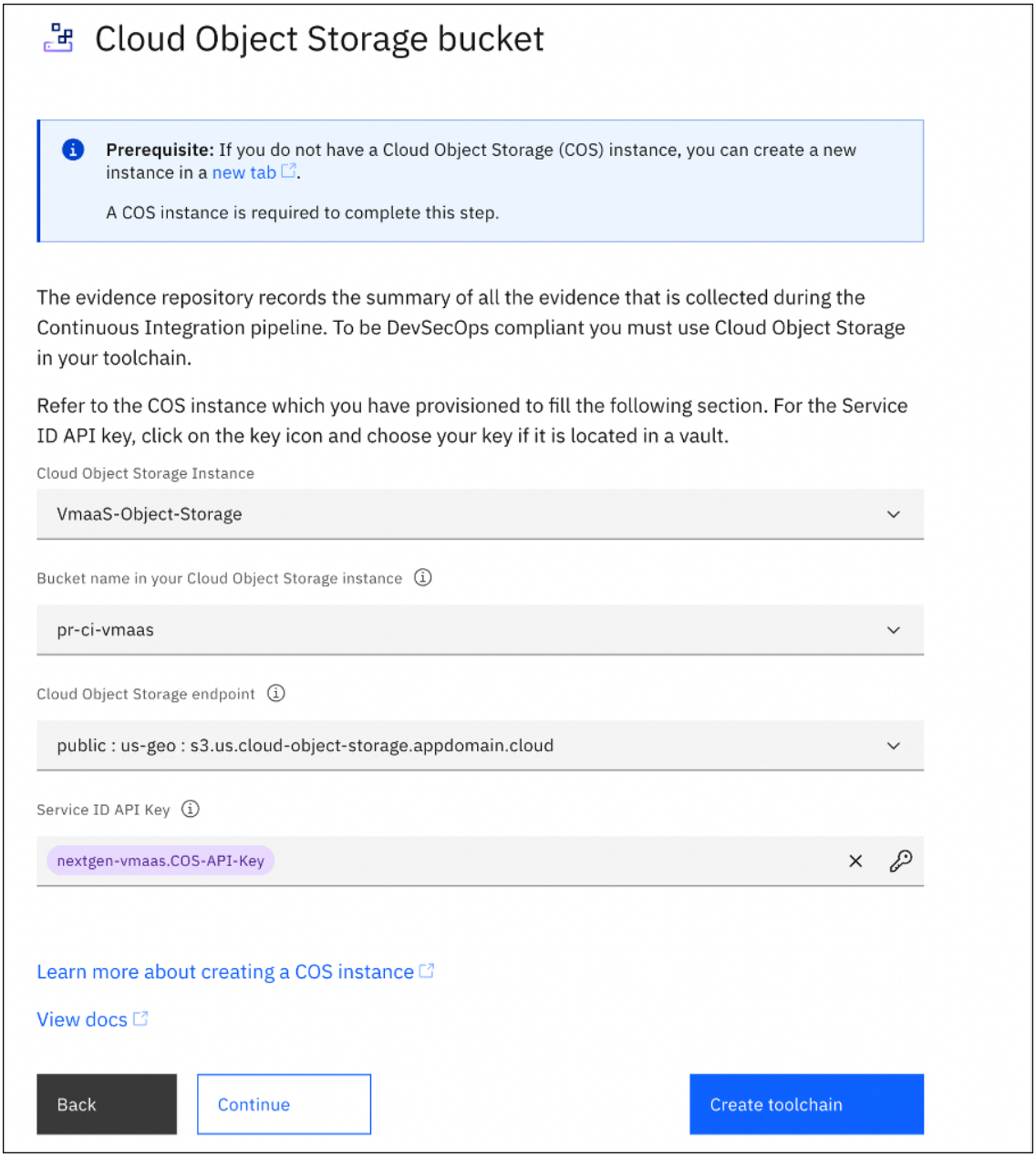
- Click **Continue**.

Cloud Object Storage Bucket

The evidence repository records the summary of all the evidence that is collected during the Continuous Integration pipeline. To be DevSecOps compliant, you must use Cloud Object Storage in your toolchain.

- **Cloud Object Storage Instance**: VmaaS-Object-Storage
- **Bucket name in your Cloud Object Storage instance**: pr-ci-vmaas
- **Cloud Object Storage endpoint**: “do not change default selection for now”
- **Service ID API Key**: COS-API-Key


Cloud Object Storage bucket
Prerequisite: If you do not have a Cloud Object Storage (COS) instance, you can create a new instance in a new tab.
A COS instance is required to complete this step.
Service ID API Key
Select a secret for this field from a secrets store using the options below.
Provider
Secrets Manager: sm-compliance-secrets
Group name
nextgen-vmaas
Secret name
COS-API-Key (Arbitrary)
Cancel
OK

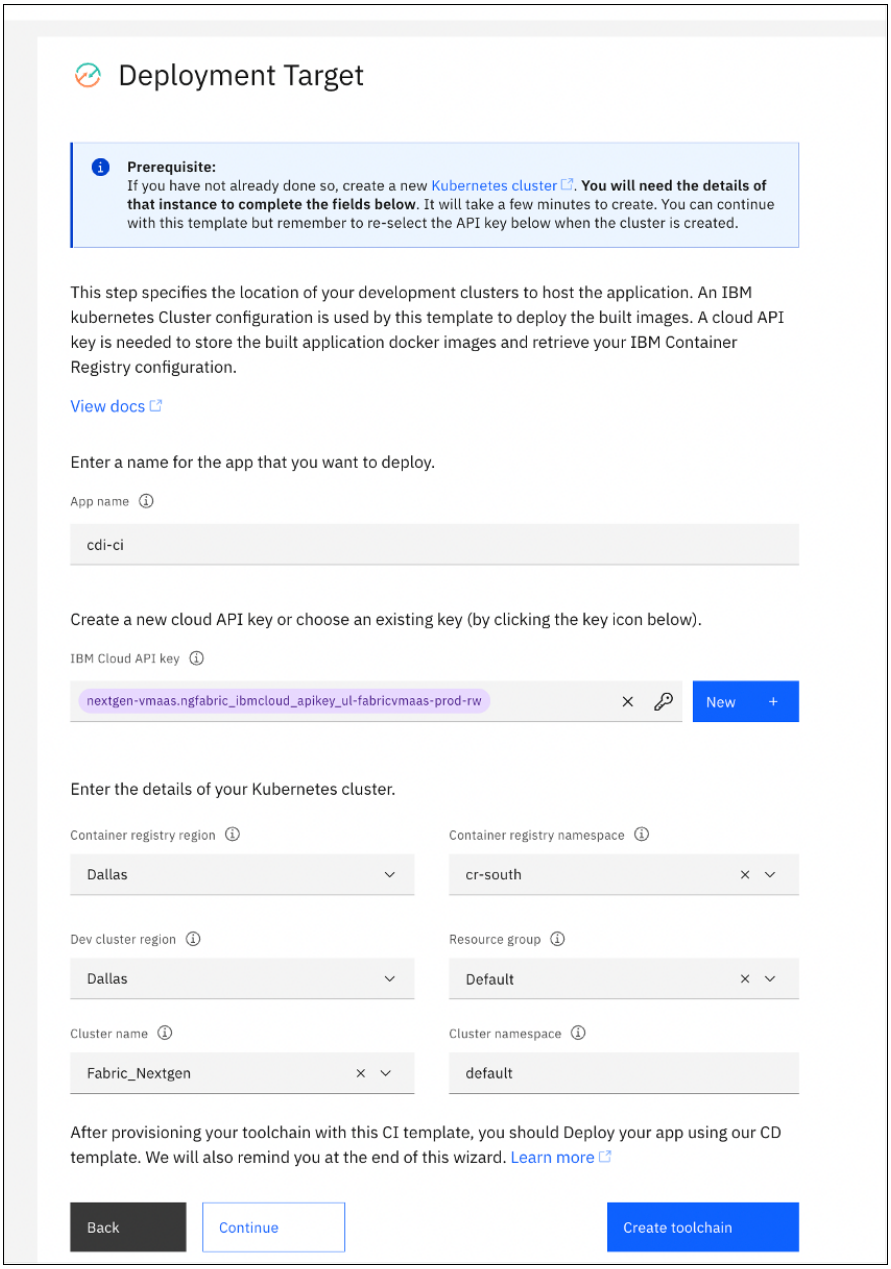


- Click **Continue**.

Deployment Target

This step specifies the location of your development clusters to host the application. An IBM Kubernetes Cluster configuration is used by this template to deploy the built images. A cloud API key is needed to store the built application docker images and retrieve your IBM Container Registry configuration

- **App name**: “Enter a name for the app that you want to deploy”
- **IBM Cloud API key**: ngfabric_ibmcloud_apikey_ul-fabricvmaas-prod-rw
  Details of your Kubernetes cluster will populate as per your api key you entered.



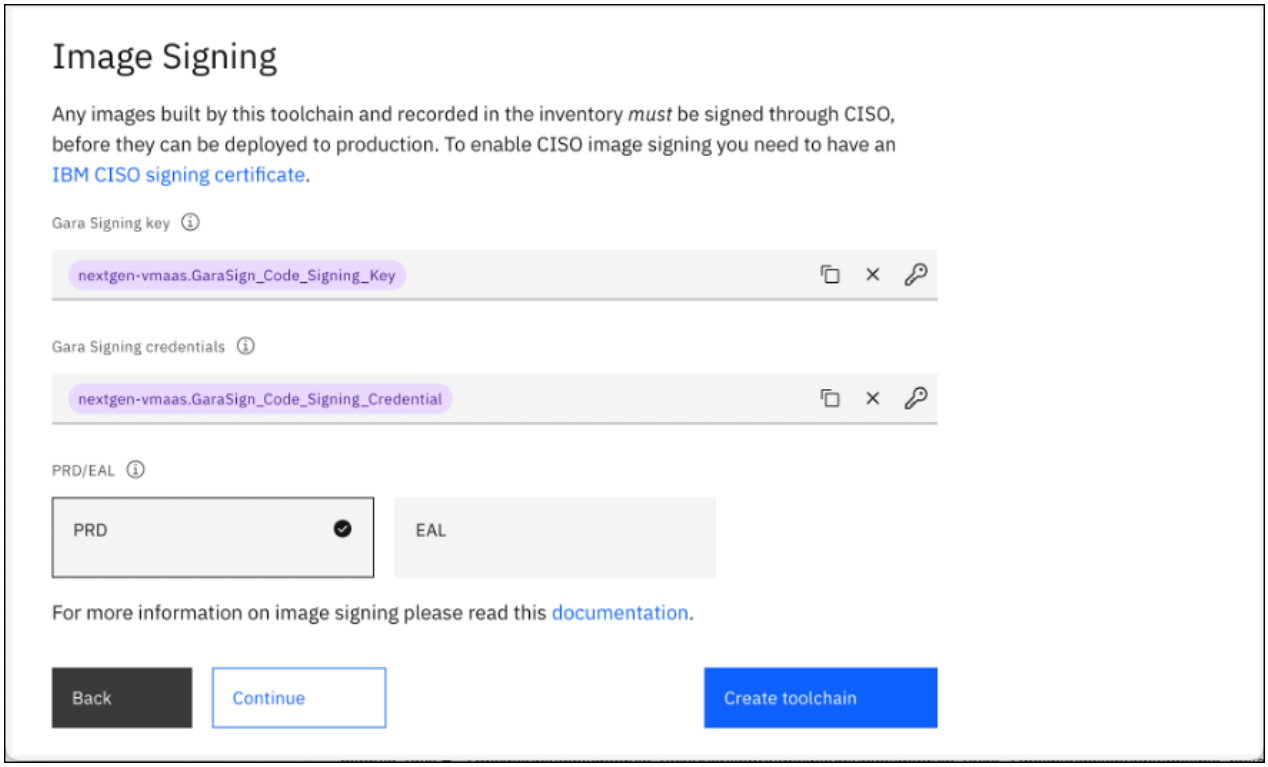
- Click **Continue**.

Image Signing

GaraSign is a replacement for CISO code signing service. Images are signed using GPG keys or GnuPG. GPG key is an implementation of a standard known as PGP (Pretty Good Privacy). The Compliance CI toolchain uses services provided by GaraSign to generate the GPG keys based on a securely stored certificate in GaraSign. In addition to requiring a GaraSign provided client, the GaraSign based certificate is further protected by restricting access to users who can access both the IBM internal network and the GaraSign key for their team. Access to the internal IBM network is provided by a worker from the TaaS worker pool.

The values are stored in SecretsManager under Resource group `nextgen-vmaas`,

- **product-type**: prd
- **gara-signing-key**: GaraSign_Code_Signing_Key
- **gara-signing-credential**: GaraSign_Code_Signing_Credential

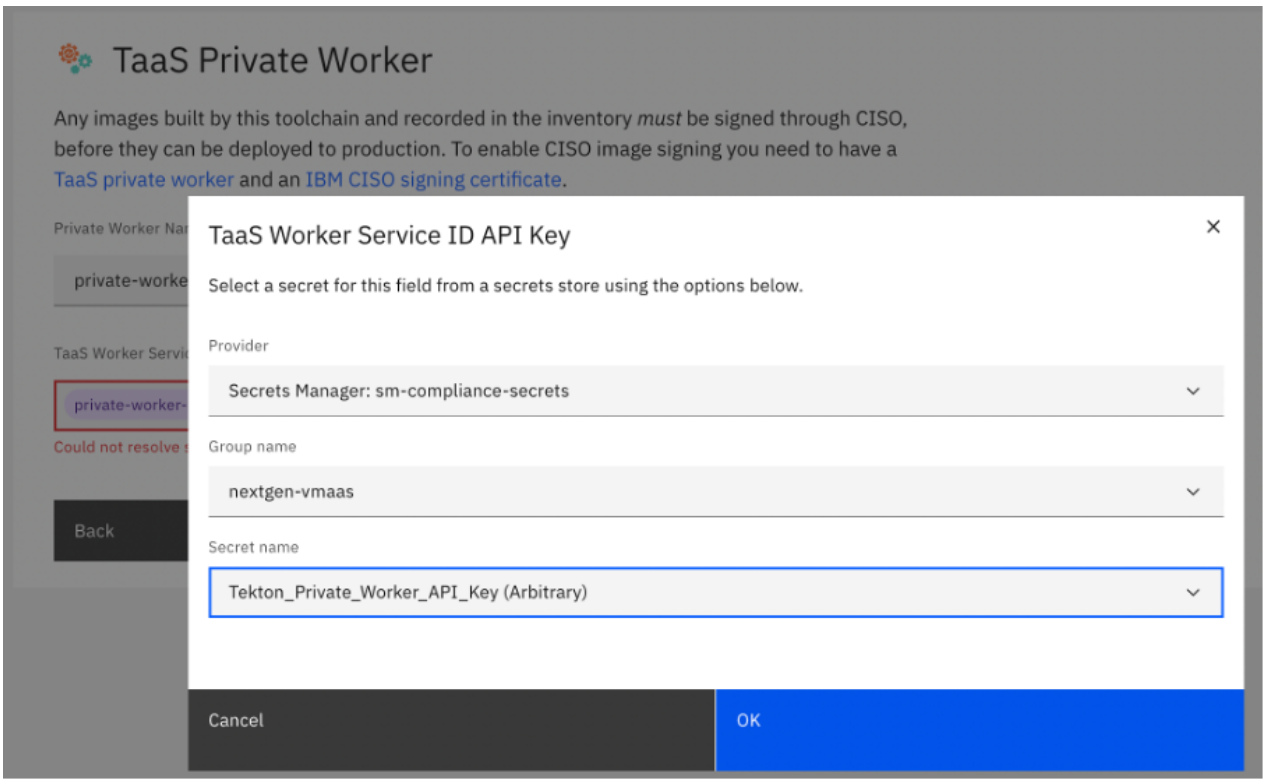


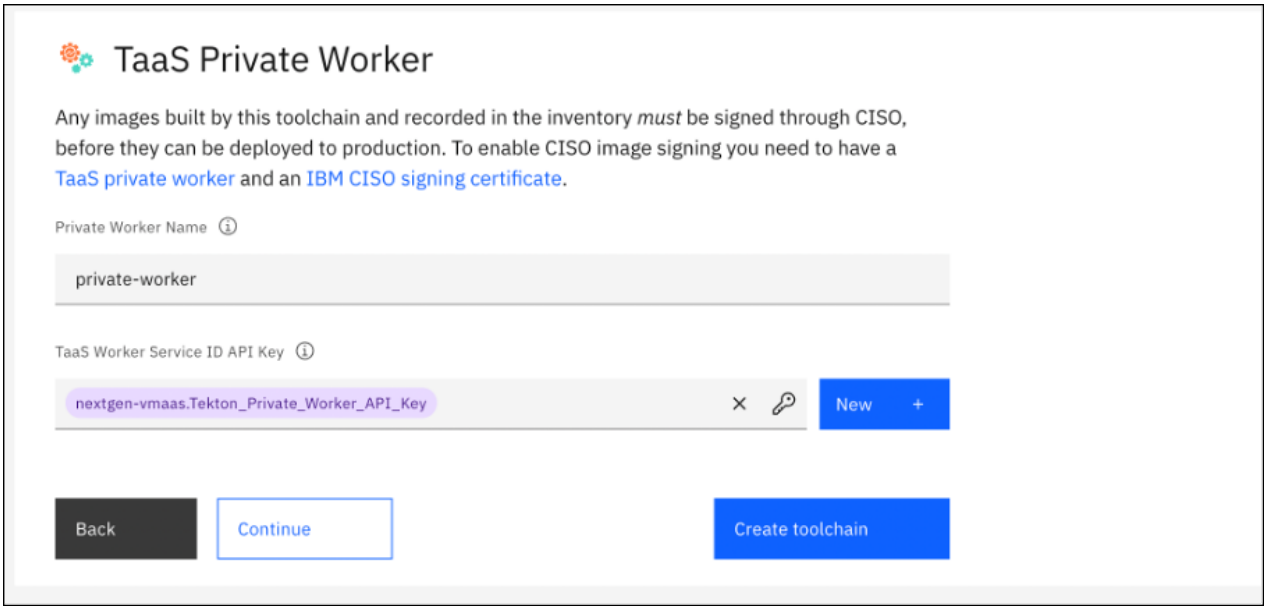
TaaS Private Worker

The private worker agents that are installed on private clusters request data only from the IBM-hosted private worker service. The data flow is one way and originates only from the agent.

With the release of the Private Workers, toolchains can be enhanced with a new private worker tool integration that allows pipeline stages to be configured to run on external Kubernetes environments. Supported platforms include the IBM Cloud Kubernetes Service, IBM Private Cloud, Docker on Desktop, or Red Hat OpenShift.

- **Private Worker Name**: private-worker
- **TaaS Worker Service ID API Key**: Tekton_Private_Worker_API_Key

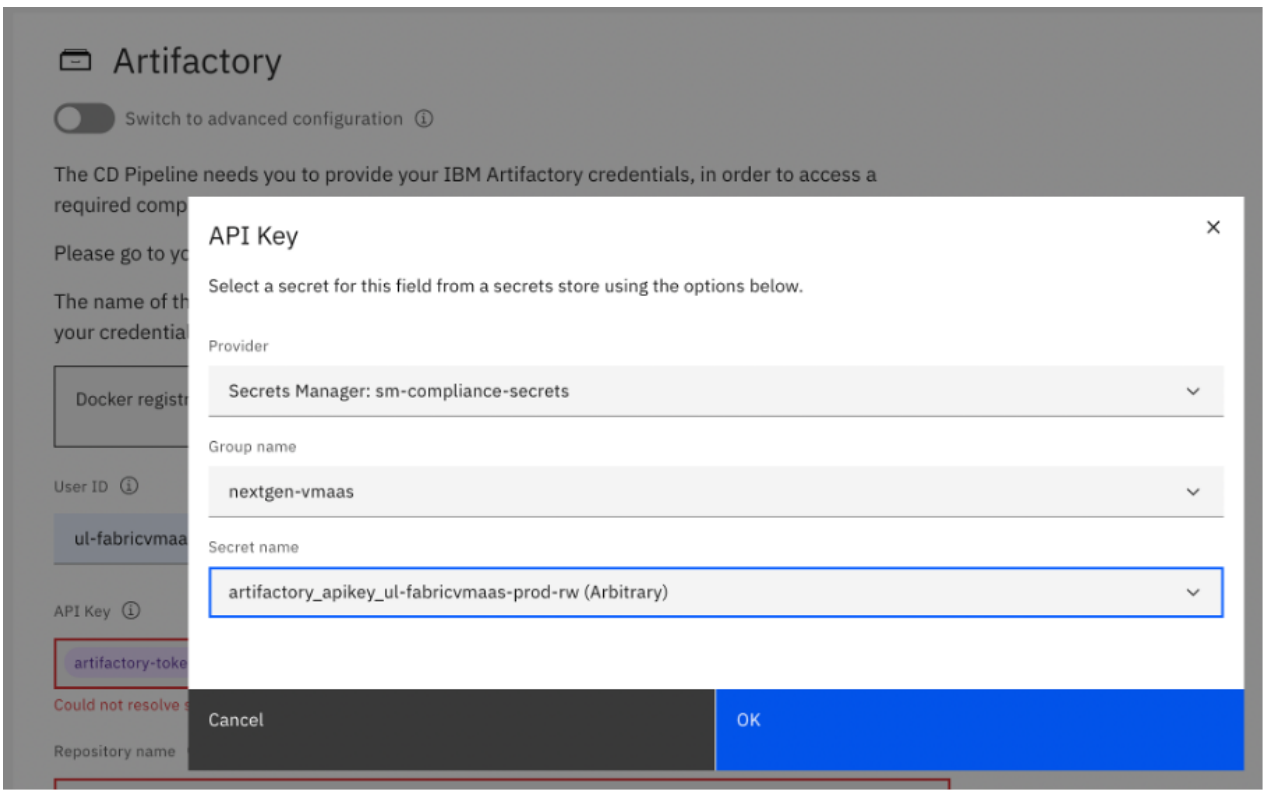


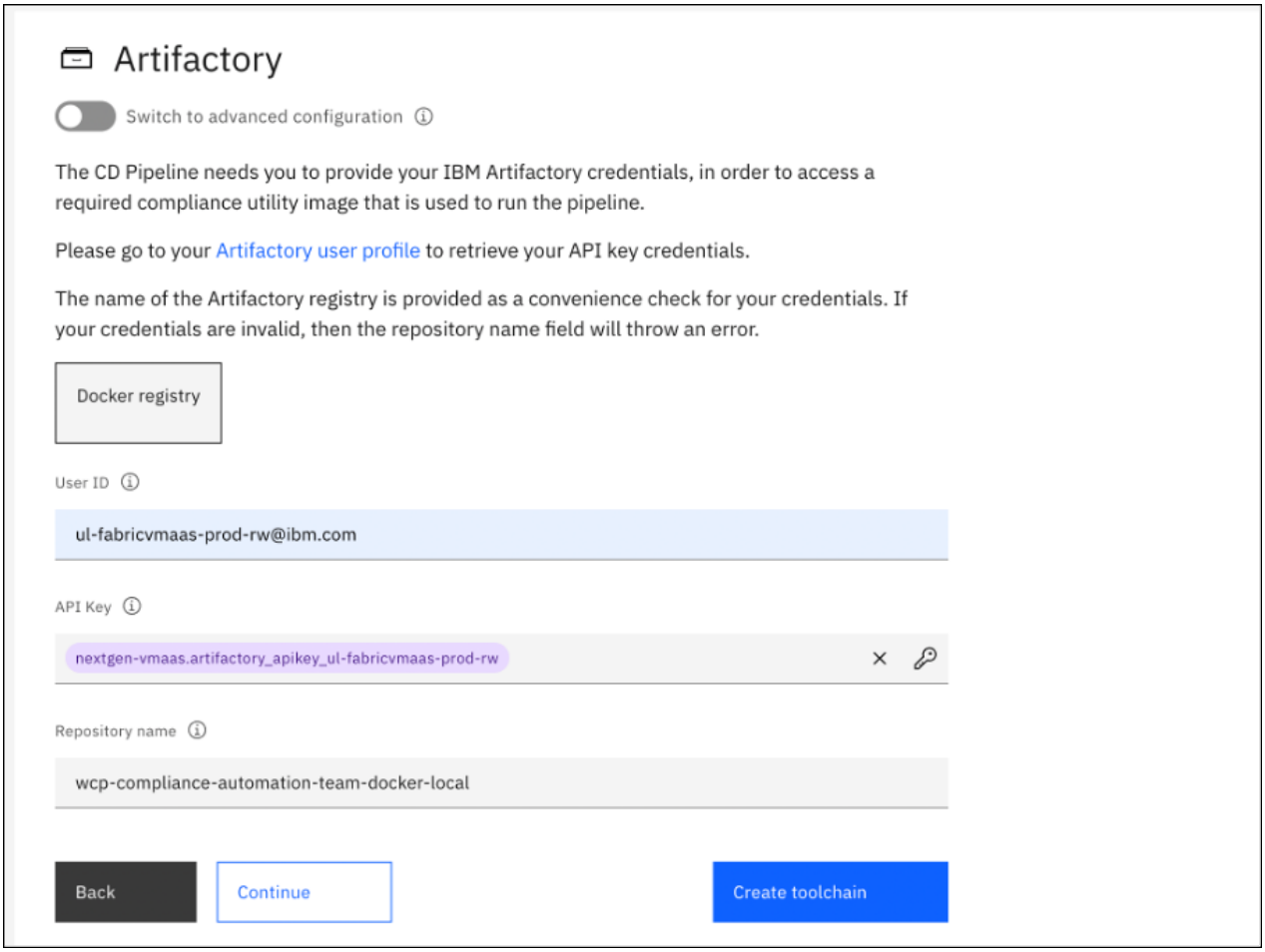


Artifactory

The CD Pipeline needs you to provide your IBM Artifactory credentials, in order to access a required compliance utility image that is used to run the pipeline.

- **User ID**: ul-fabricvmaas-prod-rw@ibm.com
- **API Key**: artifactory_apikey_ul-fabricvmaas-prod-rw
- **Repository name**: wcp-compliance-automation-team-docker-local (**do not** change this repo)





DevOps Insights

Collects and analyzes the results from unit and functional tests and code coverage tools to see if code meets criteria. Keep **default** option

SonarQube

An overview of the health and quality of source code, highlighting issues. **Default** Configuration

Optional Tools

Disable Slack option for now

Final step

- Click on **Create Toolchain**. Once completed, your toolchain will list in the UI