

Creating a CI Pipeline with Compliance

Table of Contents

- [Overview](#)
- [Toolchain](#)
 - [Prerequisites:](#)
- [Process](#)
 - [Toolchain settings](#)
 - [Application](#)
 - [Inventory](#)
 - [Issues](#)
 - [Secrets](#)
 - [Key Protect](#)
 - [Secrets Manager](#)
 - [Evidence Storage](#)
 - [Cloud Object Storage Bucket](#)
 - [Deployment Target](#)
 - [Image Signing](#)
 - [TaaS Private Worker](#)
 - [Artifactory](#)
 - [DevOps Insights](#)
 - [SonarQube](#)
 - [Optional Tools](#)
 - [Final step](#)

[Edit topic](#)

Overview

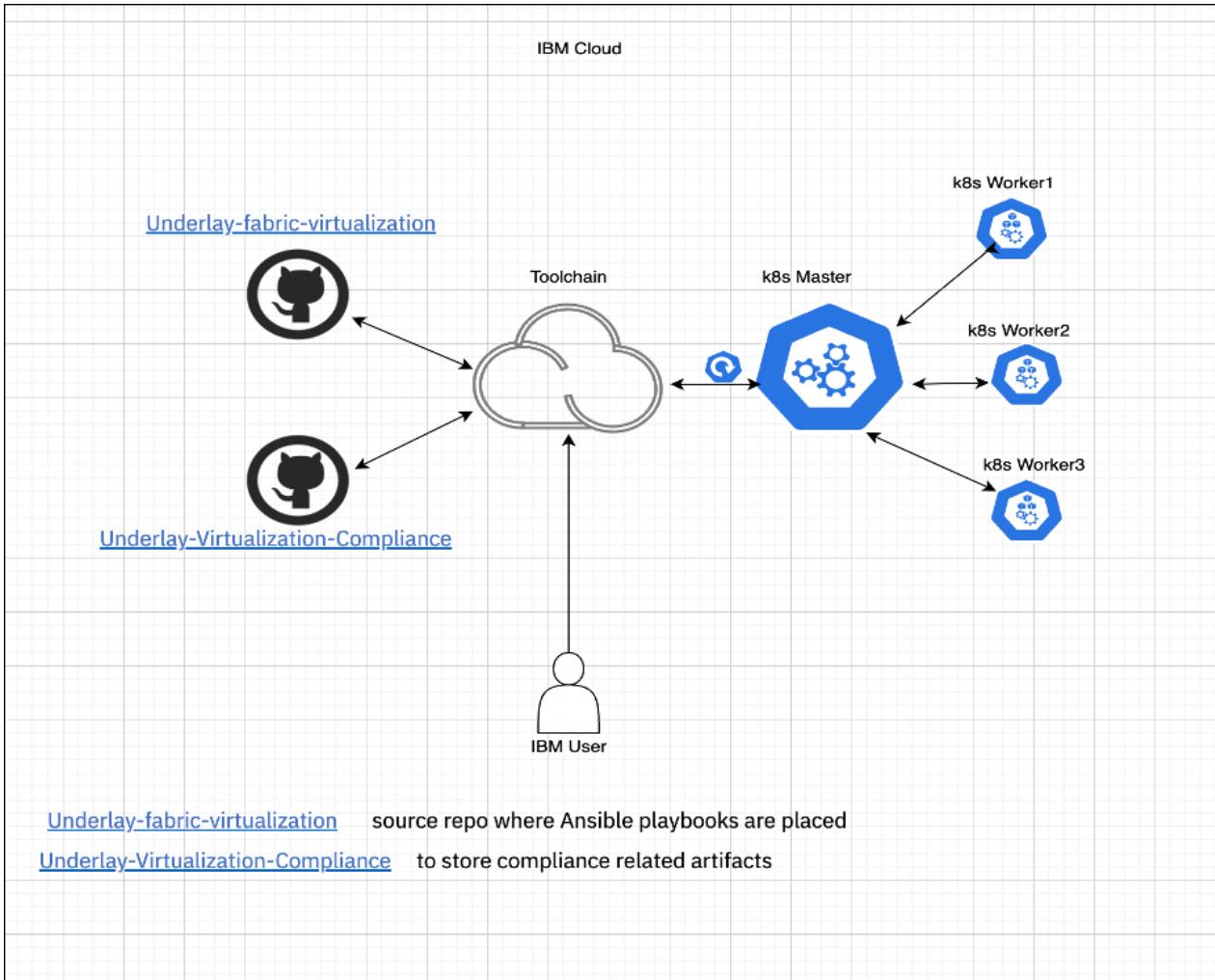
This document describes the procedure to create a Continuous Integration (CI) toolchain which deploys the plugins and resources on the destination Kubernetes cluster.

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



[Underlay-fabric-virtualization](#) source repo where Ansible playbooks are placed

[Underlay-Virtualization-Compliance](#) to store compliance related artifacts

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.

⚠ This is the public DevSecOps template. For **IBM INTERNAL** development, please use the One-Pipeline version instead.

- Now, the Toolchain template will open.
 - This template uses a guided experience. You will be asked, step by step, what tools should be included in your toolchain.
 - In most steps you will be asked to name each tool as they will appear in your toolchain.
 - 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**

Toolchain settings

Choose the settings for your toolchain, such as name and region.

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

Toolchain name ⓘ

rook-ceph-ci

Select region ⓘ

Dallas

Select a resource group ⓘ

nextgen-vmaas

I understand that all users with [access to this toolchain](#) can access artifacts that are derived from the toolchain repositories, such as build artifacts in the Delivery Pipeline. (Select the check box to continue.)

Back Continue Create toolchain

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](https://github.ibm.com/underlay-virtualization-compliance/rook-ceph-compliance-inventory.git) (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.

- Welcome
- Toolchain settings
- Application
- Inventory
- Issues
- Secrets
- Secrets Manager
- Evidence Storage
- Cloud Object Storage bucket
- Deploy
- Image Signing
- TaaS Private Worker
- Artifactory
- DevOps Insights
- SonarQube
- Optional tools
- Slack
- Summary

Issues

Switch to advanced configuration ⓘ

GitHub Enterprise Whitewater supports IBM confidential projects and makes social coding possible for teams at IBM.

Note: You must read and agree to the [terms of use](#) before adding this tool integration.

GitHub Whitewater Server

Whitewater GitHub Enterprise (<https://github.ibm.com>)

Authorized as Raghuram-M with access granted to underlay-fabric-virtualization, underlay-virtualization-compliance Whitewater GitHub Enterprise organization(s) [Manage Authorization](#)

Note: All users with [access to this toolchain](#) will be able to access artifacts derived from this repository, such as build artifacts in the Delivery Pipeline.

I understand

Repository type

Existing

Link to the repository that is specified in the Repository URL field.

Repository URL ⓘ

<https://github.ibm.com/underlay-virtualization-compliance/compliance-issues.git> x v

Git Integration Owner ⓘ

Raghuram-M

Enable GitHub Issues ⓘ

Track deployment of code changes ⓘ

[Back](#)
Continue
[Create toolchain](#)

- **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

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

Welcome

Toolchain settings

Application

Inventory

Issues

Secrets

- Key Protect
- Secrets Manager

Evidence Storage

- Cloud Object Storage bucket

Deploy

Image Signing

- TaaS Private Worker

Artifactory

DevOps Insights

Key Protect

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. Your toolchain stores references to the Key Protect secrets, not the literal secret values, which enables advanced capabilities like secret rotation. If you don't have an existing one, [create a new Key Protect instance](#) or for more details please visit the [Key Protect Documentation](#).

Important: This toolchain requires an [authorization policy](#) to be in place with this Key Protect instance to resolve secret references. This policy is created automatically when you create or save this tool integration.

Name ?
kp-compliance-secrets

Region
Dallas

Resource group
Default

Service name ?
Key-Protect-NG

Back Continue Create toolchain

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

Secrets Manager

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. If you don't have an existing one, [create a new Secrets Manager instance](#) or for more details please visit the [Secrets Manager Documentation](#).

Important: This toolchain requires an [authorization policy](#) to be in place with this Secrets Manager instance to resolve secret references. This policy is created automatically when you create or save this tool integration.

Enter a name for this Secrets Manager integration in your toolchain and select the region and resource group where your Secrets Manager instance exists.

Name (i)
sm-compliance-secrets

Region
Dallas

Resource group
nextgen-vmaas

Secrets Manager instance name (i)
Secrets Manager-VMaaS

Back Continue Create toolchain

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>

Evidence Storage

Switch to advanced configuration [\(i\)](#)

GitHub Enterprise Whitewater supports IBM confidential projects and makes social coding possible for teams at IBM.

Note: You must read and agree to the [terms of use](#) before adding this tool integration.

GitHub Whitewater Server

Whitewater GitHub Enterprise (<https://github.ibm.com>)

Authorized as Raghuram-M with access granted to underlay-fabric-virtualization, underlay-virtualization-compliance Whitewater GitHub Enterprise organization(s) [Manage Authorization](#)

Note: All users with [access to this toolchain](#) will be able to access artifacts derived from this repository, such as build artifacts in the Delivery Pipeline.

I understand

Repository type

Existing

Link to the repository that is specified in the Repository URL field.

Repository URL [\(i\)](#)

<https://github.ibm.com/underlay-virtualization-compliance/compliance-evidence.git> [X](#) [▼](#)

Git Integration Owner [\(i\)](#)

Raghuram-M

Enable GitHub Issues [\(i\)](#)

Track deployment of code changes [\(i\)](#)

Cloud Object Storage bucket

[Back](#) [Continue](#) [Create toolchain](#)

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.

The evidence
Continuous Int
in your toolcha

Refer to the CC
ID API key, click

Cloud Object Stora

VmaaS-Objec

Bucket name in yo

pr-ci-vmaas

Cloud Object Stora

public : us-ge

Service ID API Key

[cos-api-key](#)

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

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.

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.

Refer to the COS instance which you have provisioned to fill the following section. For the Service ID API key, click on the key icon and choose your key if it is located in a vault.

Cloud Object Storage Instance

Vmaas-Object-Storage

Bucket name in your Cloud Object Storage instance [\(i\)](#)

pr-ci-vmaas

Cloud Object Storage endpoint [\(i\)](#)

public : us-geo : s3.us.cloud-object-storage.appdomain.cloud

Service ID API Key [\(i\)](#)

nextgen-vmaas.COS-API-Key

X



[Learn more about creating a COS instance](#)

[View docs](#)

Back

Continue

Create toolchain

- 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 the api key you entered.

 Deployment Target

Prerequisite:
If you have not already done so, create a new [Kubernetes cluster](#). You will need the details of that instance to complete the fields below. It will take a few minutes to create. You can continue with this template but remember to re-select the API key below when the cluster is created.

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.

[View docs](#)

Enter a name for the app that you want to deploy.

App name (1)

Create a new cloud API key or choose an existing key (by clicking the key icon below).

IBM Cloud API key (1)

nextgen-vmaas.ngfabric_ibmcloud_apikey_1-fabricvmaas-prod-rw
X
🔗
New
+

Enter the details of your Kubernetes cluster.

Container registry region (1)	Container registry namespace (1)
Dallas	cr-south
Dev cluster region (1)	Resource group (1)
Dallas	Default
Cluster name (1)	Cluster namespace (1)
Fabric_Nextgen	default

After provisioning your toolchain with this CI template, you should Deploy your app using our CD template. We will also remind you at the end of this wizard. [Learn more](#)

Back Continue Create toolchain

- Click **Continue**.

Image Signing

GaraSign is a replacement for CISCO 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

Image Signing

Any images built by this toolchain and recorded in the inventory *must* be signed through CISO, before they can be deployed to production. To enable CISO image signing you need to have an [IBM CISO signing certificate](#).

The screenshot shows the 'Image Signing' configuration section. It includes fields for 'Gara Signing key' (containing 'nextgen-vmaas.GaraSign_Code_Signing_Key'), 'Gara Signing credentials' (containing 'nextgen-vmaas.GaraSign_Code_Signing_Credential'), and 'PRD/EAL' (with 'PRD' checked). Below these are instructions and navigation buttons: 'Back', 'Continue', and a large blue 'Create toolchain' button.

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

The screenshot shows the 'TaaS Private Worker' configuration dialog. It displays a message about image signing requirements and a 'TaaS Worker Service ID API Key' selection interface. The 'Secrets Manager' dropdown is set to 'sm-compliance-secrets', the 'Group name' dropdown is set to 'nextgen-vmaas', and the 'Secret name' dropdown is set to 'Tekton_Private_Worker_API_Key (Arbitrary)'. Navigation buttons include 'Back', 'Cancel', and a large blue 'OK' button.

TaaS Private Worker

Any images built by this toolchain and recorded in the inventory *must* be signed through CISO, before they can be deployed to production. To enable CISO image signing you need to have a [TaaS private worker](#) and an [IBM CISO signing certificate](#).

Private Worker Name ⓘ

private-worker

TaaS Worker Service ID API Key ⓘ

nextgen-vmaas.Tekton_Private_Worker_API_Key



New



Back

Continue

Create toolchain

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)

Artifactory

Switch to advanced configuration ⓘ

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.

Please go to your Artifactory instance and log in to provide your credentials.

The name of the Docker registry is `wcp-compliance-automation-team-docker-local`. Your credentials are stored in the `sm-compliance-secrets` secrets manager.

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: artifactory_apikey_ul-fabricvmaas-prod-rw (Arbitrary)

Cancel OK

 Artifactory

Switch to advanced configuration ⓘ

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.

Please go to your [Artifactory user profile](#) to retrieve your API key credentials.

The name of the Artifactory registry is provided as a convenience check for your credentials. If your credentials are invalid, then the repository name field will throw an error.

Docker registry

User ID ⓘ

ul-fabricvmaas-prod-rw@ibm.com

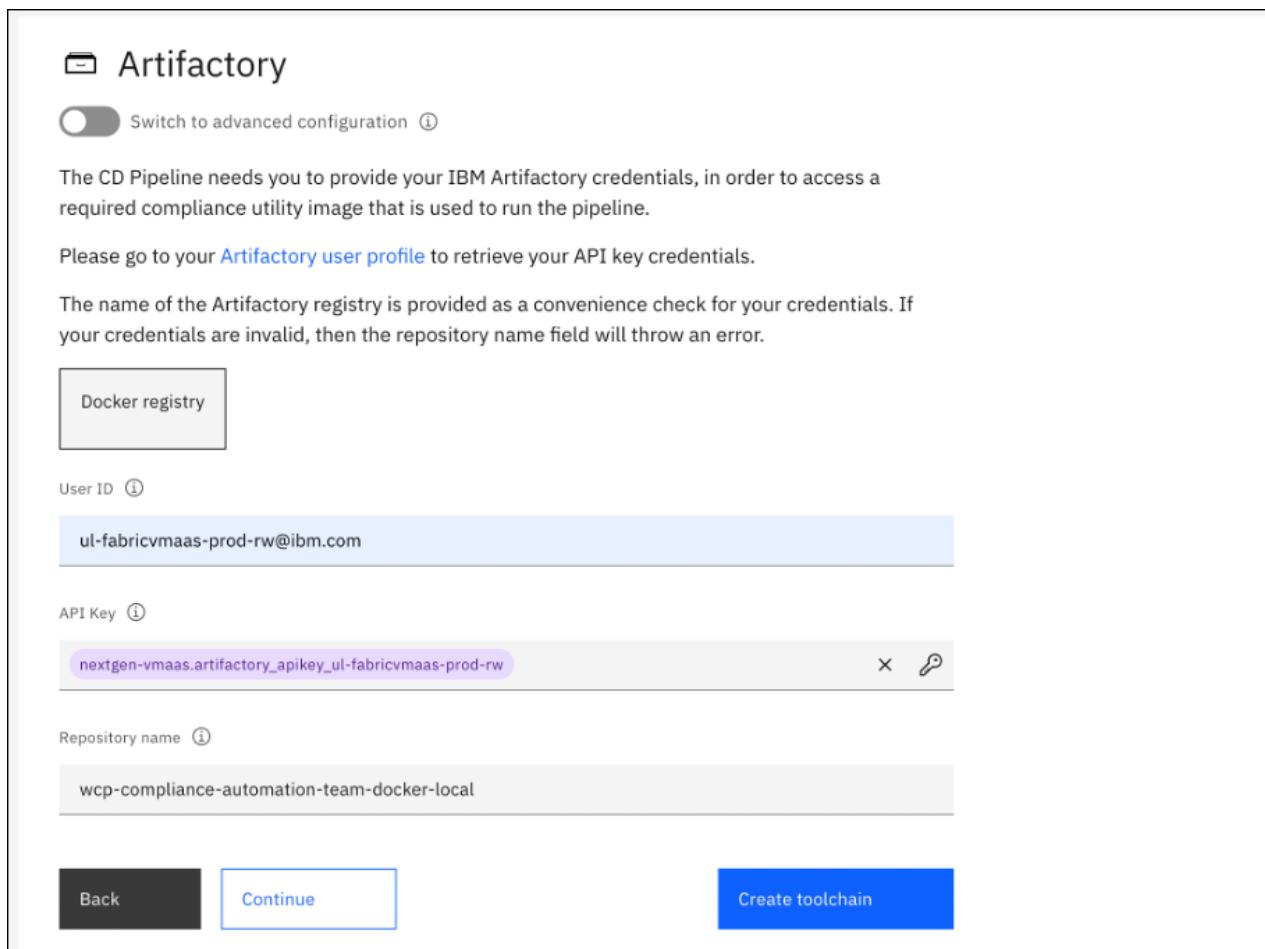
API Key ⓘ

nextgen-vmaas.artifactory_apikey_ul-fabricvmaas-prod-rw X 🔑

Repository name ⓘ

wcp-compliance-automation-team-docker-local

Back Continue Create toolchain



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