# Red Hat Trusted Application Pipeline 1.5

## Configuring Azure Pipelines

Learn how to configure Azure CI for secure CI/CD workflows.

Last Updated: 2025-04-30

# Red Hat Trusted Application Pipeline 1.5 Configuring Azure Pipelines

Learn how to configure Azure CI for secure CI/CD workflows.

## Legal Notice

## Abstract

This document provides instructions on setting up Azure CI to perform essential security tasks, such as vulnerability scanning, image signing, and attestation generation.

# Table of Contents

# PREFACE

If you’re using Azure Pipelines for your application, pipeline runs may fail due to missing secrets and environment variables. Without them, integrations with Quay, JFrog Artifactory, and Red Hat Advanced Cluster Security (ACS) won’t work, breaking security tasks like vulnerability scanning, image signing, and SBOM generation for compliance.

To prevent this, you need to securely store secrets and environment variables in Azure. This guide walks you through the process, ensuring your pipelines run smoothly and securely.

# CHAPTER 1. ADDING SECRETS AND VARIABLES TO AZURE PIPELINES FOR INTEGRATION WITH EXTERNAL TOOLS

This procedure explains how to add secrets and environment variables to Azure Pipelines and also lists which variables are required. All listed variables must be added to ensure that Azure Pipelines works correctly with RHTAP and related Red Hat products.

### Prerequisites

Before you configure Azure Pipelines, ensure you have the following:

- Admin access to your repository in Bitbucket or GitHub.
- Admin access to your Azure DevOps project and pipeline settings.
- **Container registry credentials** for pulling container images from Quay.io, JFrog Artifactory, or Sonatype Nexus.
- **Authentication details** for specific Azure Pipelines tasks:
  - **For ACS security tasks**:
    - ROX Central server endpoint
    - ROX API token
  - **For SBOM and artifact signing tasks**
    - Cosign signing key password, private key and public key
    - Trustification API and issuer URL, client ID, client secret, and supported CycloneDX version

> NOTE
>
> The credentials and other details are already Base64-encoded, so you do not need to encode them again. You can find these credentials in your **private.env** file, which you created during RHTAP installation.

### Procedure

1. Log in to https://dev.azure.com and open your Azure DevOps project.
2. In the left navigation panel, select **Pipelines**, then select **Library**.
3. Select **Variable group** to create a new variable group.
4. Enter a name for the variable group, for example, **rhtap**.
5. In the variable group editor:
   a. Select **Add** to add a new variable.
   b. In the **Name** field, enter the key. For example, **GITOPS_AUTH_PASSWORD**.
   c. In the **Value** field, enter the value used to authenticate with the GitOps repository for pushing updated image information.

d. Select the **Keep this value secret** checkbox to mask the value in the UI and logs.

6. Repeat step 5 to add all required secrets:

**Table 1.1. Image registry and GitOps secrets**

| Variable | Description |
|---|---|
| **IMAGE_REGISTRY_PASSWORD** | Password for accessing your container image registry. |
| **GITOPS_AUTH_PASSWORD** | The token the system uses to update the GitOps repository for newly built images. |

**Table 1.2. Secrets required for ACS and SBOM tasks**

| Variable | Description |
|---|---|
| **ROX_API_TOKEN** | API token for accessing the ROX server. |
| **COSIGN_SECRET_PASSWORD** | Password for Cosign signing key. |
| **COSIGN_SECRET_KEY** | Private key for Cosign. |
| **TRUSTIFICATION_OIDC_CLIENT_SECRET** | Client secret used alongside the client ID to authenticate to the Trustification Bombastic API. |

6. Now add regular environment variables and don’t mask their values. In the variable group editor:

   a. Select **Add**.

   b. In the **Name** field, enter the key. For example, **IMAGE_REGISTRY_USER**.

   c. In the **Value** field, enter the value. In our example: a username for accessing your container image registry.

   d. Do not select the **Keep this value secret** checkbox.

7. Repeat step 6 to add all required environment variables:

**Table 1.3. Image registry and GitOps variables**

| Variable | Description |
|---|---|
| **IMAGE_REGISTRY_USER** | Username for accessing your container image registry. |

| Variable | Description |
|---|---|
| **GITOPS_AUTH_USERNAME** (optional) | Your OpenShift GitOps username. This variable is required for Azure to work with Bitbucket. By default, lines with this variable are commented in the **azure-pipelines.yml** file. To start using Bitbucket, uncomment all 5 instances of the line **# GITOPS_AUTH_USERNAME: $(GITOPS_AUTH_USERNAME)**. |

Table 1.4. Variables required for ACS and SBOM tasks

| Variable | Description |
|---|---|
| **ROX_CENTRAL_ENDPOINT** | Endpoint for the ROX Central server. |
| **COSIGN_PUBLIC_KEY** | Public key for Cosign. |
| **TRUSTIFICATION_BOMBASTIC_API_URL** | URL for Trustification Bombastic API used in SBOM generation. |
| **TRUSTIFICATION_OIDC_ISSUER_URL** | OIDC issuer URL used for authentication when interacting with the Trustification Bombastic API. |
| **TRUSTIFICATION_OIDC_CLIENT_ID** | Client ID for authenticating to the Trustification Bombastic API using OIDC. |
| **TRUSTIFICATION_SUPPORTED_CYCLONEDX_VERSION** | Specifies the CycloneDX SBOM version that is supported and generated by the system. |

**Optional**: Set the Rekor and TUF variables if your CI provider runners do not run on the same cluster as the RHTAP instance.

Table 1.5. Rekor and TUF variables

| Variable | Description |
|---|---|
| **REKOR_HOST** | URL of your Rekor server. |
| **TUF_MIRROR** | URL of your TUF service. |

8. Select **Save**.

9. To authorize pipelines to use this variable group:

   a. Select the **Pipeline permissions** tab.

   b. Select **Add pipeline**.

   c. Select the pipelines that require access to this variable group and select **Authorize selected pipelines**.

10. Optional: If you use a different name for the variable group other than **rhtap**, you must update the variable group name in the **azure-pipelines.yml** file.

```
variables:
   - group: <my-variable-group>
```

### Verification

1. Rerun the latest pipeline. If the secrets are applied correctly, the pipeline will complete successfully. After a successful run, verify that tasks such as RHACS or SBOM display the expected details.

*Revised on 2025-04-30 03:55:38 UTC*