CBOM - Azure DevOps Pipeline Documentation

This document provides a detailed step-by-step guide on setting up an **Azure DevOps (ADO) pipeline** for generating a **Crypto Bill of Materials (CBOM)** using **cbomkit_theia**.

The pipeline:
• **Builds a Python application** into a **Podman/Docker image**.
• **Generates a CBOM** (CycloneDX Crypto Bill of Materials) for security analysis.
• **Publishes the CBOM as an artifact** in ADO pipeline.

Pipeline Workflow

1. Cloned git repo and built the docker image and pushed it to docker hub
   a. git clone https://github.com/IBM/cbomkit-theia.git
   b. cd cbomkit-theia
   c. docker build -t cbomkit-theia .
   d. docker tag cbomkit-theia nihharika/cbomkit_theia:latest
   e. docker push nihharika/cbomkit_theia:latest

2. Pull the cbomkit Docker Image

• Fetches the latest cbomkit_theia image from Docker Hub.
    • docker pull nihharika/cbomkit_theia:latest

3. Build the Python Application as a Podman Image

• Builds the Python project and packages it as a Podman image.
    • podman build -t my-python-app .

4. Save the Python Project as a Tar File

• Exports the Python project container into a tar file.
    • podman save my-python-app > my-app.tar

5. Generate CBOM using cbomkit podman image

• Uses cbomkit to analyze the saved Docker image and generate a CBOM.

- podman run --rm -v $(pwd):/data nihharika/cbomkit_theia:latest image get /data/my-app.tar > enriched_CBOM.json

6. Publish CBOM as an Artifact

- Saves the CBOM JSON report in the Azure DevOps pipeline for future use.
    - CBOM Report (enriched_CBOM.json) stored in ADO.

Pipeline

```
trigger:
- main

pool:
vmImage: ubuntu-latest

steps:
- script: |
# Pull the cbomkit Docker image from Docker Hub
docker pull nihharika/cbomkit_theia:latest
displayName: "Pull cbomkit Docker image"

- script: |
# Build the Python podman image
podman build -t my-python-app .
displayName: "Build Python Podman image"

- script: |
# Save the Python project as a image
podman save my-python-app > my-app.tar
displayName: "Save Python project as tar"

- script: |
# Generate CBOM using cbomkit
podman run --rm -v $(pwd):/data nihharika/cbomkit_theia:latest image get
/data/my-app.tar > enriched_CBOM.json
displayName: "Generate CBOM"
```

```
- task: PublishBuildArtifacts@1
inputs:
pathToPublish: "enriched_CBOM.json"
artifactName: "CBOM"
publishLocation: "Container"
displayName: "Publish CBOM Artifact"
```

enriched_CBOM.json (Sample part)

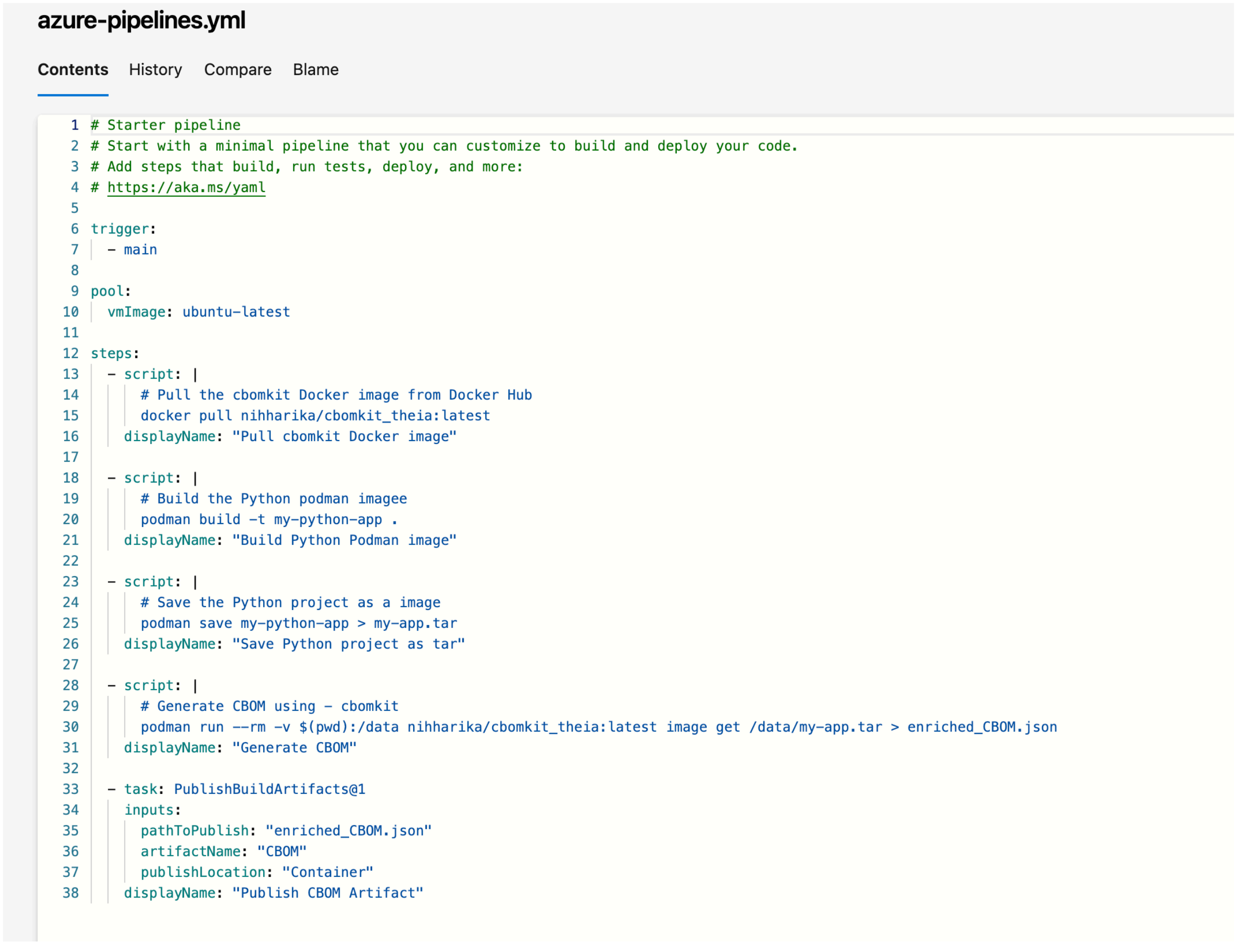
```
{
"$schema": "http://cyclonedx.org/schema/bom-1.6.schema.json",
"bomFormat": "CycloneDX",
"specVersion": "1.6",
"serialNumber": "urn:uuid:9c3b5251-0a41-4eb2-b48a-ead7617bc1a6",
"version": 1,
"metadata": {
"timestamp": "2025-02-09T09:56:00Z",
"tools": {
"services": [
{
"provider": {
"name": "IBM Research"
},
"name": "CBOMkit-theia",
"version": "0.9",
"services": [
{
"name": "Certificate File Plugin"
},
{
"name": "Secret Plugin"
},
{
"name": "java.security Plugin"
}
]
}
]
```

```
}
},
"components": [
{
"bom-ref": "1bf87d530a541e3e",
"type": "cryptographic-asset",
"name": "QuoVadis Root Certification Authority",
"evidence": {
"occurrences": [
{
"location": "/app/.local/lib/python3.9/site-packages/certifi/cacert.pem"
},
{
"location": "/etc/ssl/certs/QuoVadis_Root_CA.pem"
},
{
"location": "/usr/lib/ssl/certs/QuoVadis_Root_CA.pem"
},
{
"location": "/usr/local/lib/python3.9/site-
packages/pip/_vendor/certifi/cacert.pem"
}
]
},
"cryptoProperties": {
"assetType": "certificate",
"certificateProperties": {
"subjectName": "QuoVadis Root Certification Authority",
"issuerName": "QuoVadis Root Certification Authority",
"notValidBefore": "2001-03-19T18:33:33Z",
"notValidAfter": "2021-03-17T18:33:33Z",
"signatureAlgorithmRef": "f2075f8ea94ebfc1",
"subjectPublicKeyRef": "b396cff964ec91af",
"certificateFormat": "X.509",
"certificateExtension": ".pem"
}
}
},
{
"bom-ref": "f6ea5dc7eda1f5e0",
"type": "cryptographic-asset",
"name": "Certum Trusted Network CA 2",
"evidence": {
"occurrences": [
{
"location": "/app/.local/lib/python3.9/site-packages/certifi/cacert.pem"
},
{
"location": "/app/.local/lib/python3.9/site-
packages/pip/_vendor/certifi/cacert.pem"
},
{
"location": "/etc/ssl/certs/Certum_Trusted_Network_CA_2.pem"
},
{
"location": "/usr/lib/ssl/certs/Certum_Trusted_Network_CA_2.pem"
},
{
"location": "/usr/local/lib/python3.9/site-
packages/pip/_vendor/certifi/cacert.pem"
}
]
},
"cryptoProperties": {
"assetType": "certificate",
"certificateProperties": {
"subjectName": "Certum Trusted Network CA 2",
"issuerName": "Certum Trusted Network CA 2",
"notValidBefore": "2011-10-06T08:39:56Z",
"notValidAfter": "2046-10-06T08:39:56Z",
"signatureAlgorithmRef": "3e89c55bb29695e6",
"subjectPublicKeyRef": "5dfc017ea563147c",
"certificateFormat": "X.509",
"certificateExtension": ".pem"
}
}
},
{
"bom-ref": "6c62b6a6c713220f",
"type": "cryptographic-asset",
"name": "CFCA EV ROOT",
"evidence": {
```

```
"occurrences": [
{
"location": "/etc/ssl/certs/ca-certificates.crt"
},
{
"location": "/usr/lib/ssl/certs/ca-certificates.crt"
},
{
"location": "/usr/share/ca-certificates/mozilla/CFCA_EV_ROOT.crt"
}
]
},
"cryptoProperties": {
"assetType": "certificate",
"certificateProperties": {
"subjectName": "CFCA EV ROOT",
"issuerName": "CFCA EV ROOT",
"notValidBefore": "2012-08-08T03:07:01Z",
"notValidAfter": "2029-12-31T03:07:01Z",
"signatureAlgorithmRef": "f364f895f5391746",
"subjectPublicKeyRef": "940818c84cbe3145",
"certificateFormat": "X.509",
"certificateExtension": ".crt"
}
}
},
{
"bom-ref": "b7d9b9a5c679962a",
"type": "cryptographic-asset",
"name": "Telekom Security TLS RSA Root 2023",
"evidence": {
"occurrences": [
{
"location": "/app/.local/lib/python3.9/site-
packages/pip/_vendor/certifi/cacert.pem"
}
]
},
"cryptoProperties": {
"assetType": "certificate",
"certificateProperties": {
```

```
"subjectName": "Telekom Security TLS RSA Root 2023",
"issuerName": "Telekom Security TLS RSA Root 2023",
"notValidBefore": "2023-03-28T12:16:45Z",
"notValidAfter": "2048-03-27T23:59:59Z",
"signatureAlgorithmRef": "d82de78eae7f51e4",
"subjectPublicKeyRef": "e51d8c4aa00bc8db",
"certificateFormat": "X.509",
"certificateExtension": ".pem"
}
}
},
{
"bom-ref": "17de8f0adaf737ac",
"type": "cryptographic-asset",
"name": "SSL.com EV Root Certification Authority RSA R2",
"evidence": {
"occurrences": [
{
"location": "/usr/lib/ssl/certs/ca-certificates.crt"
},
{
"location": "/usr/share/ca-
certificates/mozilla/SSL.com_EV_Root_Certification_Authority_RSA_R2.crt"
},
{
"location": "/etc/ssl/certs/ca-certificates.crt"
}
]
},
"cryptoProperties": {
"assetType": "certificate",
"certificateProperties": {
"subjectName": "SSL.com EV Root Certification Authority RSA R2",
"issuerName": "SSL.com EV Root Certification Authority RSA R2",
"notValidBefore": "2017-05-31T18:14:37Z",
"notValidAfter": "2042-05-30T18:14:37Z",
"signatureAlgorithmRef": "f364f895f5391746",
"subjectPublicKeyRef": "709ebbb7fbafe785",
"certificateFormat": "X.509",
"certificateExtension": ".crt"
}
```

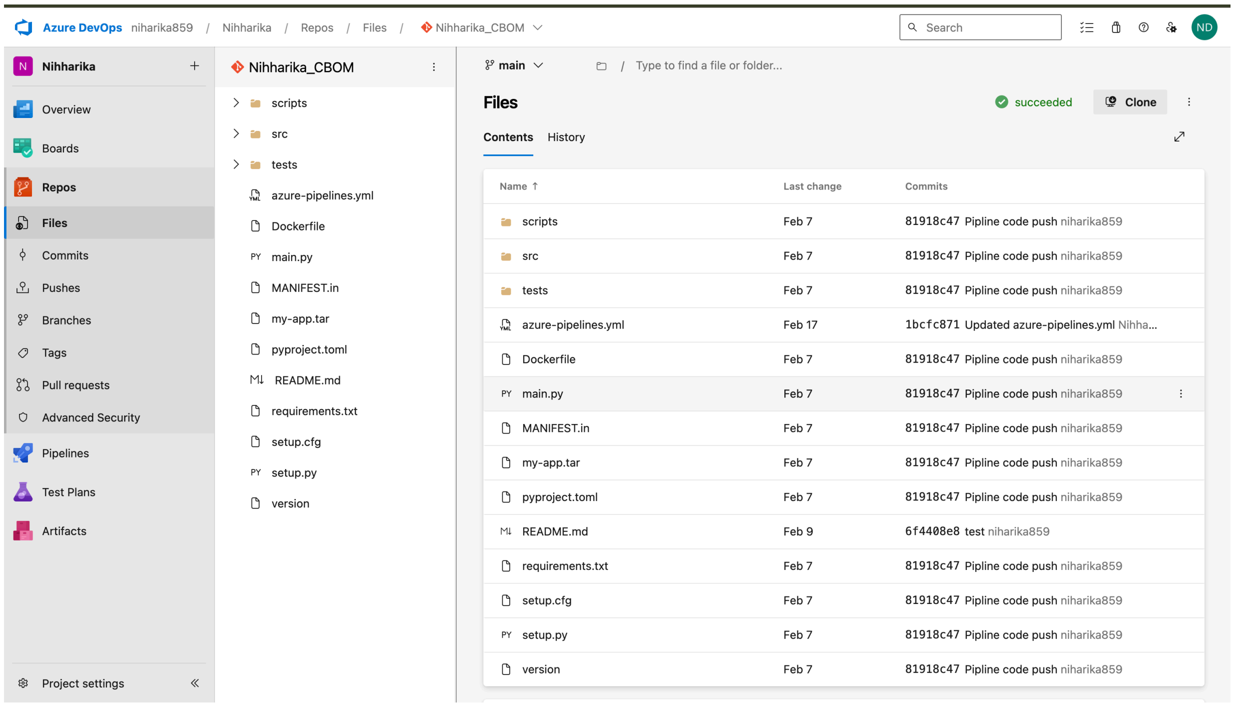
```
}
},
{
"bom-ref": "5a189aa2d585a95b",
"type": "cryptographic-asset",
"name": "Starfield Root Certificate Authority - G2",
"evidence": {
"occurrences": [
{
"location": "/etc/ssl/certs/ca-certificates.crt"
},
{
"location": "/usr/lib/ssl/certs/ca-certificates.crt"
},
{
"location": "/usr/share/ca-
certificates/mozilla/Starfield_Root_Certificate_Authority_-_G2.crt"
}
]
},
"cryptoProperties": {
"assetType": "certificate",
"certificateProperties": {
"subjectName": "Starfield Root Certificate Authority - G2",
"issuerName": "Starfield Root Certificate Authority - G2",
"notValidBefore": "2009-09-01T00:00:00Z",
"notValidAfter": "2037-12-31T23:59:59Z",
"signatureAlgorithmRef": "f364f895f5391746",
"subjectPublicKeyRef": "279acc699905ce4e",
"certificateFormat": "X.509",
"certificateExtension": ".crt"
}
}
},
```



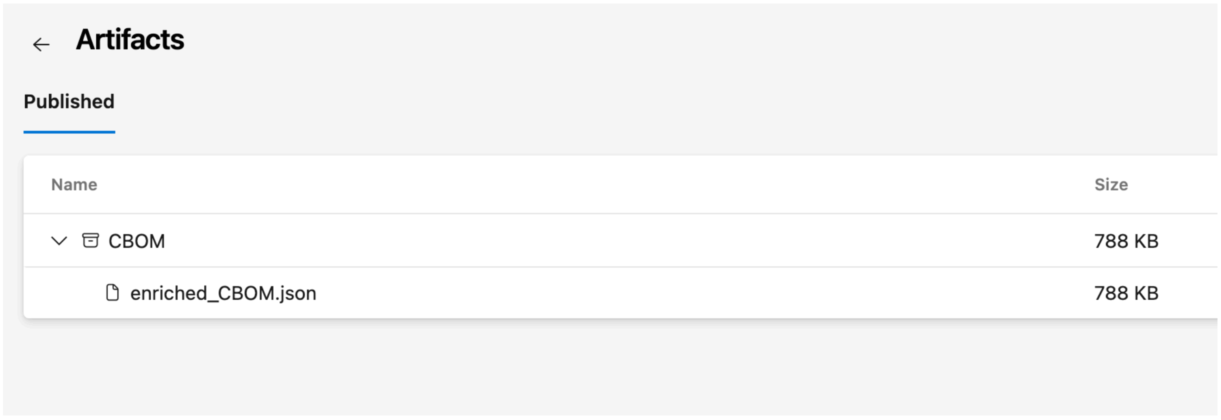
azure-pipelines.yml

Contents History Compare Blame

```yaml
# Starter pipeline
# Start with a minimal pipeline that you can customize to build and deploy your code.
# Add steps that build, run tests, deploy, and more:
# https://aka.ms/yaml

trigger:
  - main

pool:
  vmImage: ubuntu-latest

steps:
  - script: |
      # Pull the cbomkit Docker image from Docker Hub
      docker pull nihharika/cbomkit_theia:latest
    displayName: "Pull cbomkit Docker image"

  - script: |
      # Build the Python podman imagee
      podman build -t my-python-app .
    displayName: "Build Python Podman image"

  - script: |
      # Save the Python project as a image
      podman save my-python-app > my-app.tar
    displayName: "Save Python project as tar"

  - script: |
      # Generate CBOM using - cbomkit
      podman run --rm -v $(pwd):/data nihharika/cbomkit_theia:latest image get /data/my-app.tar > enriched_CBOM.json
    displayName: "Generate CBOM"

  - task: PublishBuildArtifacts@1
    inputs:
      pathToPublish: "enriched_CBOM.json"
      artifactName: "CBOM"
      publishLocation: "Container"
    displayName: "Publish CBOM Artifact"
```



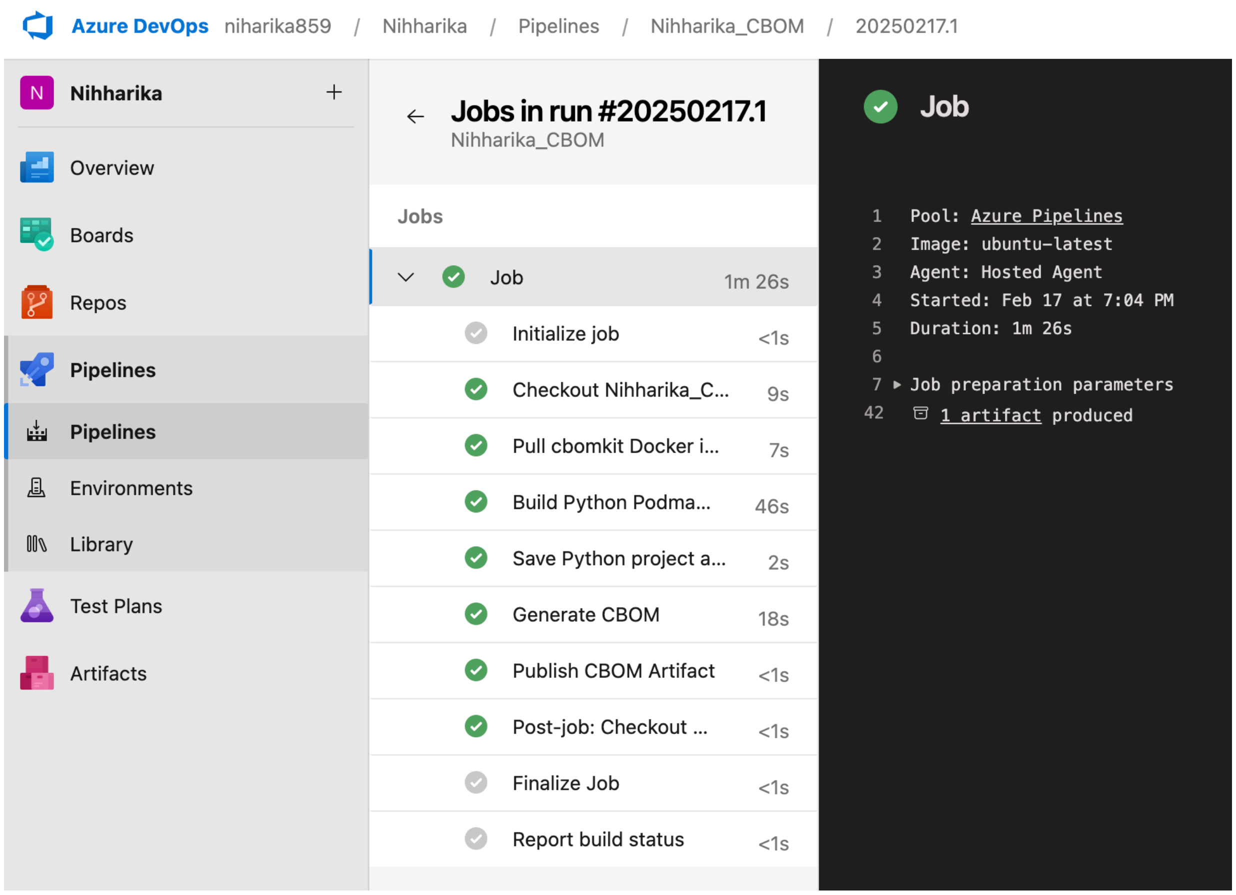
Name ↑	Last change	Commits
scripts	Feb 7	81918c47 Pipline code push niharika859
src	Feb 7	81918c47 Pipline code push niharika859
tests	Feb 7	81918c47 Pipline code push niharika859
azure-pipelines.yml	Feb 17	1bcfc871 Updated azure-pipelines.yml Nihha...
Dockerfile	Feb 7	81918c47 Pipline code push niharika859
main.py	Feb 7	81918c47 Pipline code push niharika859
MANIFEST.in	Feb 7	81918c47 Pipline code push niharika859
my-app.tar	Feb 7	81918c47 Pipline code push niharika859
pyproject.toml	Feb 7	81918c47 Pipline code push niharika859
README.md	Feb 9	6f4408e8 test niharika859
requirements.txt	Feb 7	81918c47 Pipline code push niharika859
setup.cfg	Feb 7	81918c47 Pipline code push niharika859
setup.py	Feb 7	81918c47 Pipline code push niharika859
version	Feb 7	81918c47 Pipline code push niharika859



Artifacts

Published

Name	Size
CBOM	788 KB
enriched_CBOM.json	788 KB



Azure DevOps niharika859 / Nihharika / Pipelines / Nihharika_CBOM / 20250217.1

Jobs in run #20250217.1
Nihharika_CBOM

Jobs

Job	1m 26s
Initialize job	<1s
Checkout Nihharika_C...	9s
Pull cbomkit Docker i...	7s
Build Python Podma...	46s
Save Python project a...	2s
Generate CBOM	18s
Publish CBOM Artifact	<1s
Post-job: Checkout ...	<1s
Finalize Job	<1s
Report build status	<1s

Job

```
Pool: Azure Pipelines
Image: ubuntu-latest
Agent: Hosted Agent
Started: Feb 17 at 7:04 PM
Duration: 1m 26s

▸ Job preparation parameters
  1 artifact produced
```