White paper

# **Solution Guide:** Shift Security Left with Anchore Enterprise

anchore

# Contents


3 > **Introduction**

5 > **Implement an automated DevSecOps workflow with Anchore**

5 > Creating the Anchore Developer Policy Bundle

5 > Anchore and Jira Integration

6 > Container Image Build

7 > Anchore Image Analysis

8 > Automated Jira Issue(s)

8 > Policy Findings

9 > Action Workbench (Remediation Path)

10 > Allowlist

11 > **Conclusion**

## Introduction

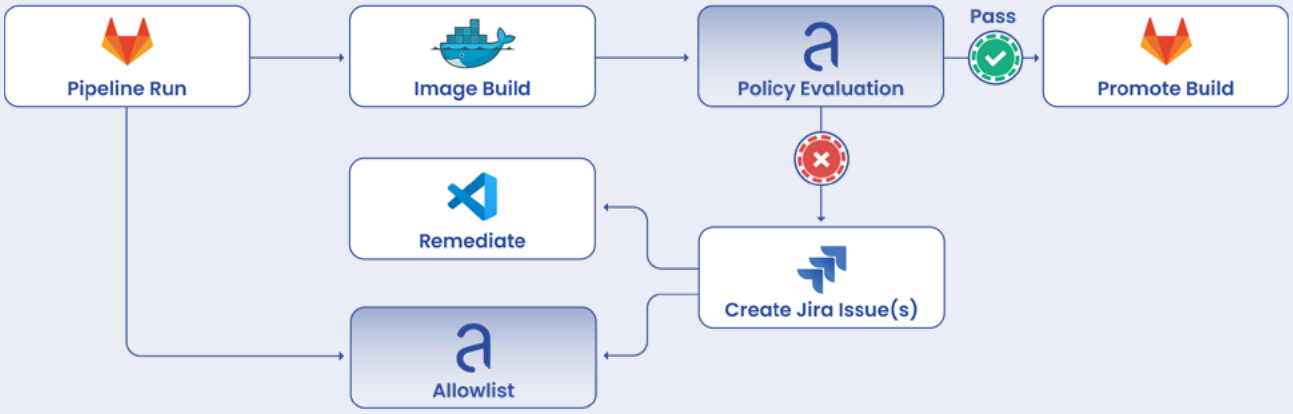
Shift left security is a secure development practice that integrates vulnerability assessment into continuous integration (CI) pipelines. Anchore Enterprise is helping organizations address this requirement. Developers gain early insights to potential vulnerabilities or policy violations, and are then able to prioritize findings or immediately remediate the issue. Anchore Enterprise injects policy compliance and vulnerability tracking right into the Software Development Life Cycle (SDLC).

In traditional security processes, issues are addressed after releases. Shifting left offers a number of advantages:

- Lower cost of remediation
- Faster time to market
- Improved overall security posture
- Increased user trust

Anchore Enterprise helps automate early vulnerability detection and remediation for security findings (e.g. CISA KEV, NVD Critical, etc.) through a superior user experience and persistent background services.

In this guide we present a battle-tested, shift-left developer workflow with Anchore Enterprise. The example will include commonly used tools by federal customers: GitLab as the continuous integration (CI) platform, Anchore Enterprise as the vulnerability scanner and compliance checker, and Atlassian Jira as the remediation tracking solution. Anchore Enterprise is agnostic to CI platform and remediation tracking tools, so this same automation can be achieved with your preferred tooling.




By removing the friction and delays of legacy security tooling, Anchore Enterprise contributes to the modern DevSecOps experience and allows for quicker remediations of security issues. Anchore can optionally be used as a pipeline blocking gate for both vulnerabilities and compliance checks. For example, if your policy does not allow critical vulnerabilities and/or NIST 800-53 violations, Anchore can cause your pipeline job to fail and the developer can use the results to remediate the issue(s). Policies are flexible and can be configured in great detail, allowing security teams to create policy checks that reflect the optimal balance between risk management and developer velocity.

**Requirements:**
To implement the workflow depicted above, you will need:

- A running instance of Anchore Enterprise with administrator privileges
  - Anchore Enterprise Deployment Guide
    **Note:** If you don't have an Anchore Enterprise license, you can contact our team to get started with a free evaluation.
- Access to a GitLab project and build runner
  - GitLab Runner server/image will require the anchorectl binary, see installation instructions for AnchoreCTL.
  - AnchoreCTL configuration should be set in your GitLab project
- Access to a container registry
  - Container registry information added to Anchore Enterprise
- Access to a Jira project with the ability to create new tickets

# Implement an automated **DevSecOps** workflow with **Anchore**

Below we cover how to work with Anchore policies, how to integrate Anchore into individual CI pipeline stages and how to remediate or justify policy violations.

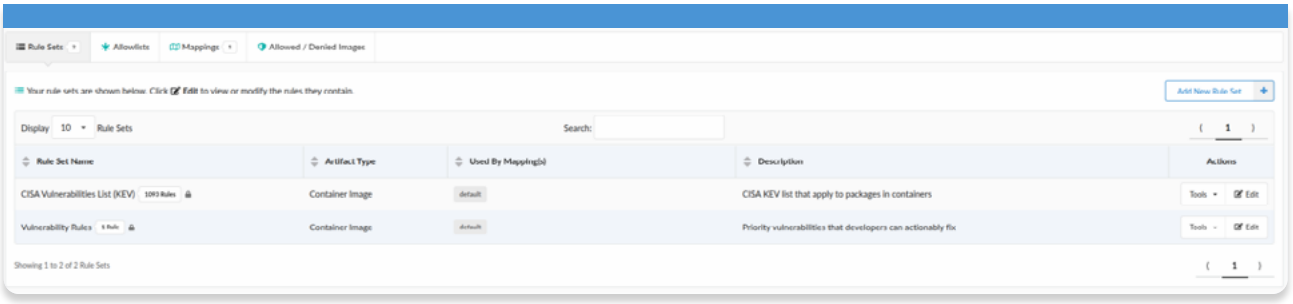
## Creating the Anchore Developer Policy Bundle

Anchore's Policy Engine is a powerful yet flexible feature. You can use and modify any of the prebuilt policy bundles (e.g., NIST 800-53, CIS, Iron Bank, FedRAMP) or you can create your own custom policy bundles meeting the unique requirements of your organization. It is recommended to copy an existing policy if you need to make changes to a prebuilt policy bundle.

In this example, we are using a policy bundle called anchore-developer-bundle.
This bundle is made up of only two policy rules:

- Trigger on any CISA KEV findings
- Trigger if there are critical vulnerabilities with an available fix.

You can download a working version of this policy file here.



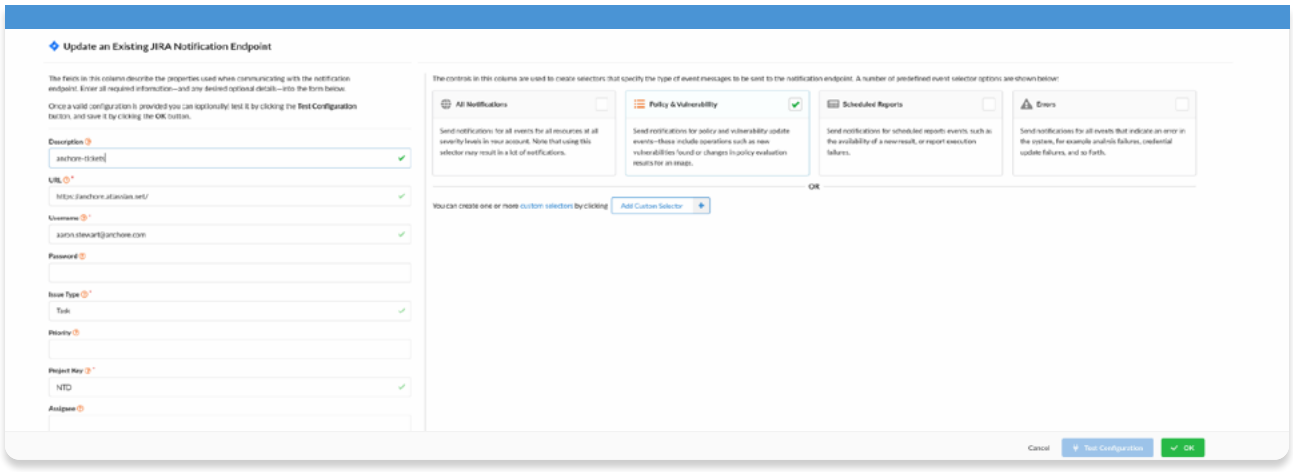

In the Anchore Enterprise UI, access Policies in the left sidebar and then Import Policy to upload the json file and create the new policy. Reference image below for the Rule Sets you should see.

## Anchore and Jira Integration

Anchore Enterprise natively integrates with Jira to automate the creation of Jira issues when policy violations are triggered in the CI pipeline.

You'll need an API token and a user that has access to create tickets in your project. If you're using Atlassian's SaaS solution, you can create a token here.

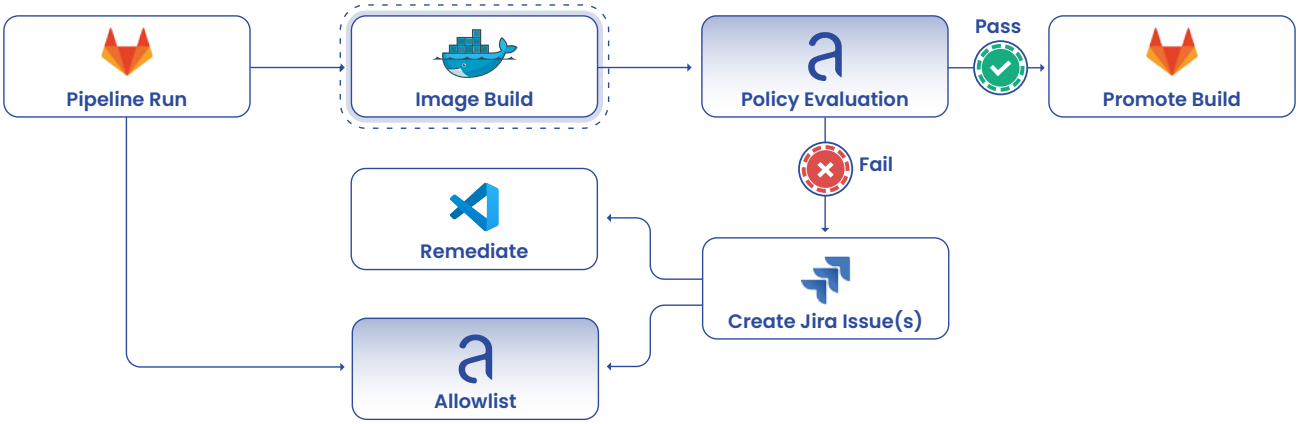
In the Anchore Enterprise UI, go to Events & Notifications > Manage Notification Endpoints.
In the Jira section, add a new endpoint using your user, token, and project information.
Check Policy and Vulnerability box to automatically create tickets for failed policy checks.




Test your configuration and click OK.

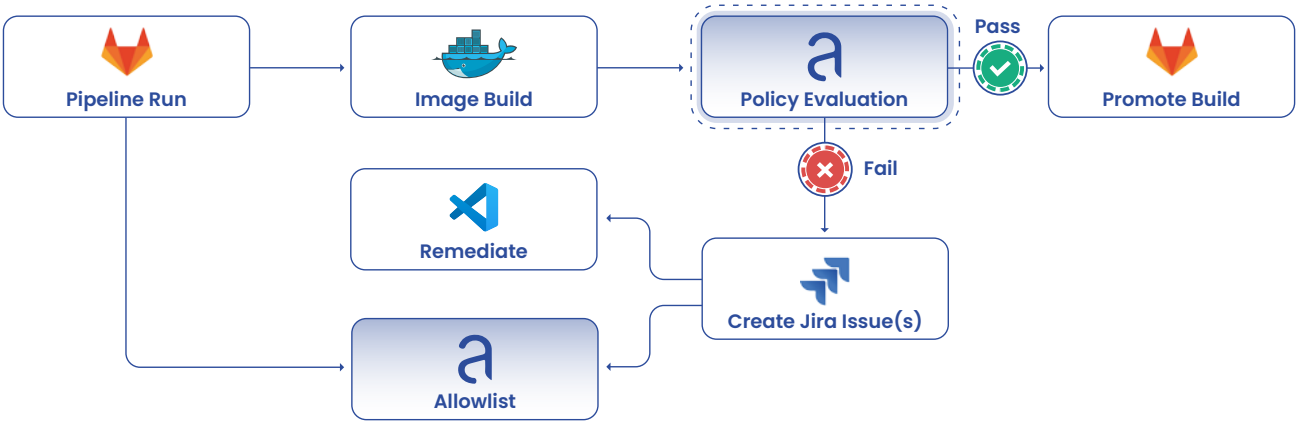
## Container Image Build

Container images can be built with your preferred tool and pushed to any Docker v2 registry that Anchore Enterprise can access. The sample code block shows an image build, tag, and push using Docker.




```
variables:
  REGISTRY: image-registry.mydomain
  IMAGE_TAG: $REGISTRY/my-app:$CI_PIPELINE_ID
stages:
  - build
  - anchore
build_tag_push:
  stage: build
  image: docker:latest
  services:
    - docker:dind
  before_script:
    - docker login $REGISTRY -u $REGISTRY_USER -p $REGISTRY_PASSWORD
  script:
    - docker build -t $IMAGE_TAG .
    - docker push $IMAGE_TAG
```

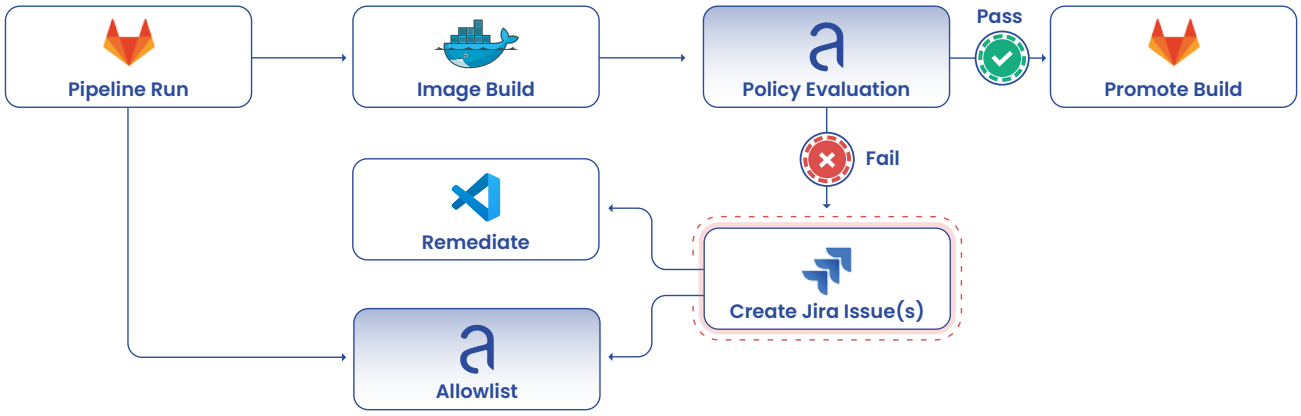
## Anchore Image Analysis



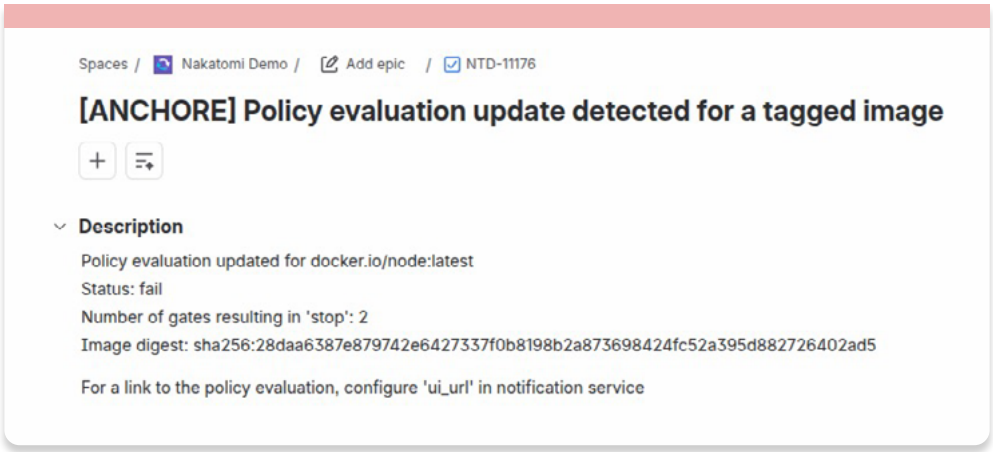

This stage uses the **anchorectl** tool to add the built image to Anchore Enterprise. Anchore generates an SBOM for the image and then performs the analysis based on the SBOM. If a policy violation in the *anchore-developer-bundle* is detected then Anchore returns a non-zero exit code, causing the job to fail. The Dockerfile is also being referenced in this example since that is a common compliance requirement.

```
anchore-image-scan:
  stage: anchore
  image: my-image-with-anchorectl:latest
  needs: ["build_tag_push"]
  script:
    # Ensures registry is added to Anchore Enterprise
    - >-
      anchorectl registry add $REGISTRY
      --type "docker_v2"
      || echo "Registry already exists..."

    # Adds the built image from the registry
    - >-
      anchorectl image add $IMAGE_TAG
      --dockerfile Dockerfile
      --from registry
      --no-auto-subscribe
      --wait

    # Performs policy compliance check and fails the CI job if check fails
    - >-
      anchorectl image check $IMAGE_TAG
      -p anchore-developer-bundle
      --detail
      --fail-based-on-results
```

## Automated Jira Issue(s)

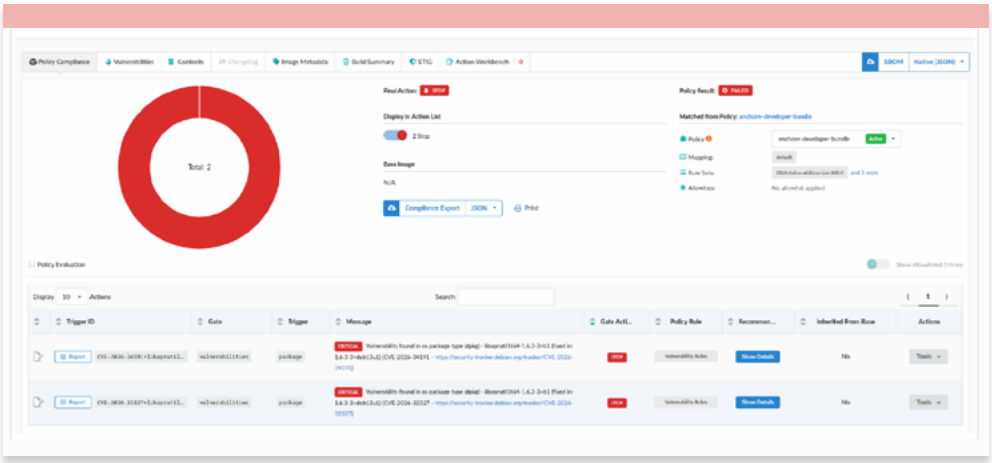



If a CI job fails due to a failed compliance check, a Jira issue will be created in the designated project.



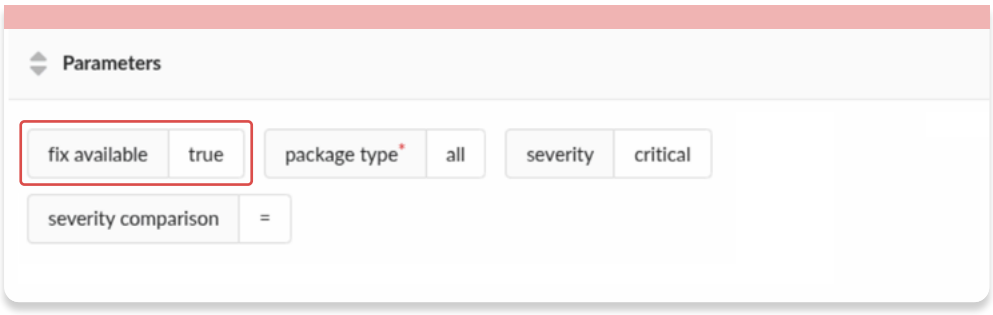

## Policy Findings

Further details about the failure can be found in the Anchore Enterprise UI by navigating to the Policy Compliance tab under the specified image.

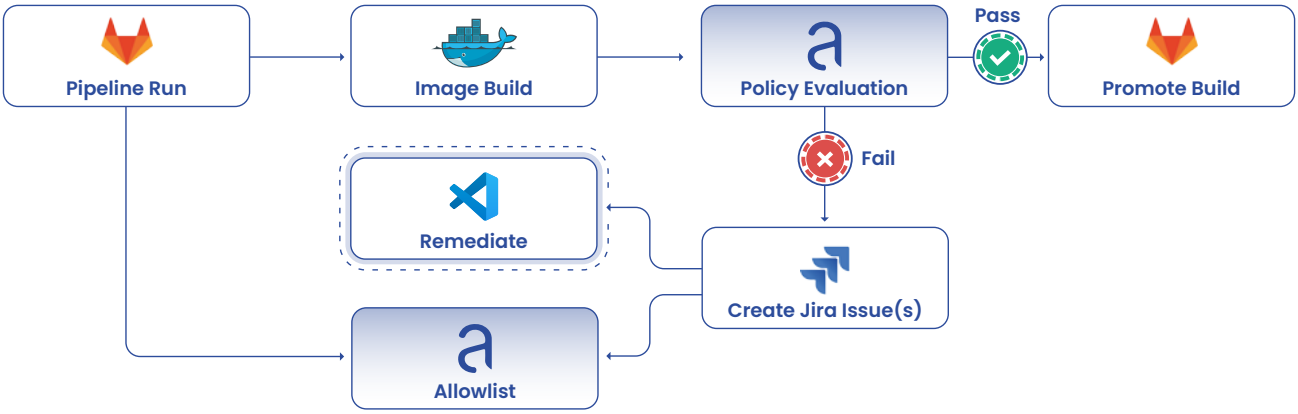


All of the **STOP** actions that get triggered by the **anchore-developer-bundle** will have a fix available, because this requirement is specified in the policy itself. It is also possible that a single package vulnerability could trigger more than one policy violation, depending on how rulesets are configured.

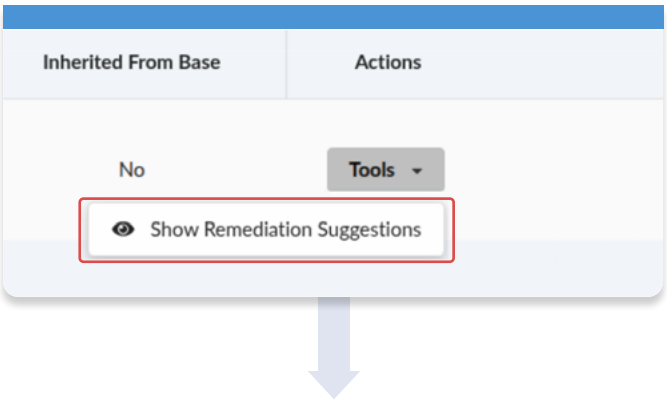
From this point you can use either the Action Workbench to initiate a remediation path, or you can add the vulnerability to the Allowlist.



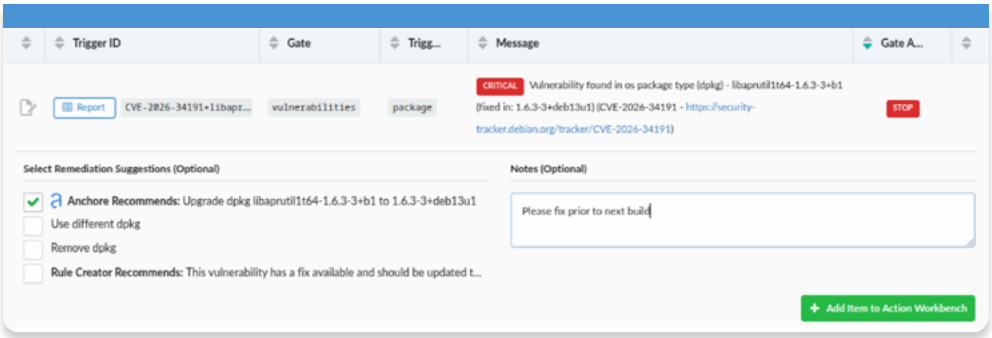

## Action Workbench (Remediation Path)




For each Policy finding, you can add a Remediation Suggestion to the Action Workbench by selecting Tools > Show Remediation Suggestions in the Actions column.

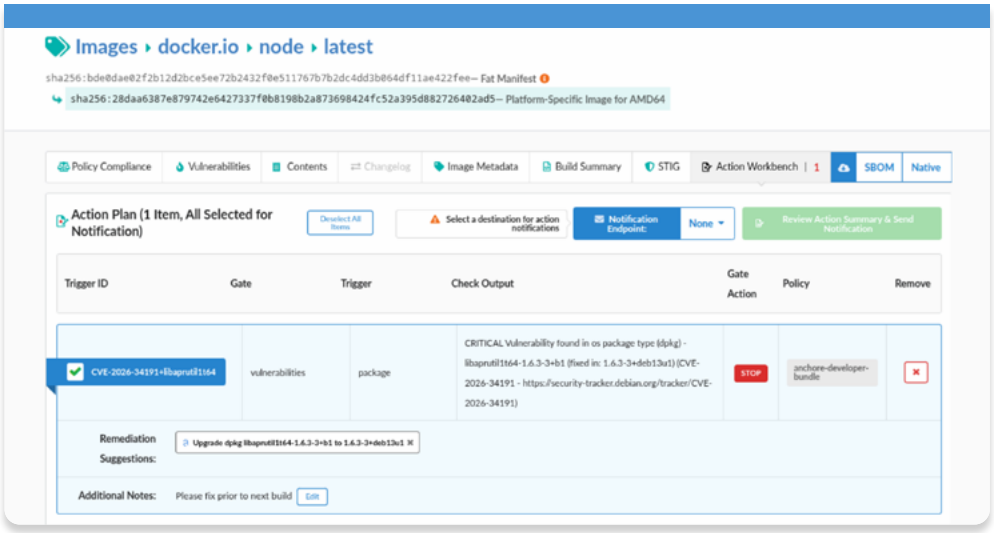



From there, you can select recommendations, provide a brief description, and then add the suggestions to the Action Workbench.

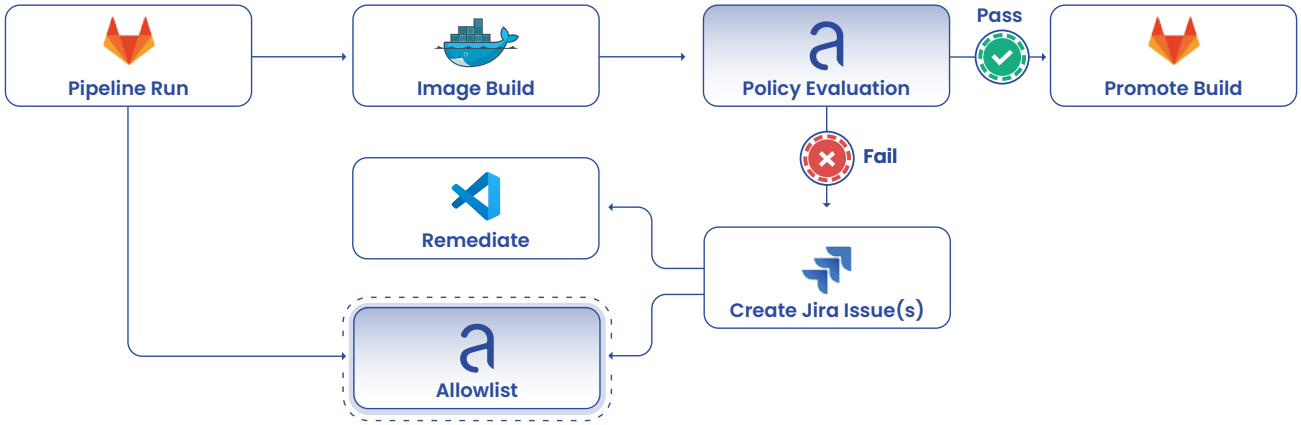


This populates the Action Workbench tab for the image.

The Action Workbench can be used as a working list between developers and security analysts to swiftly resolve policy violations.

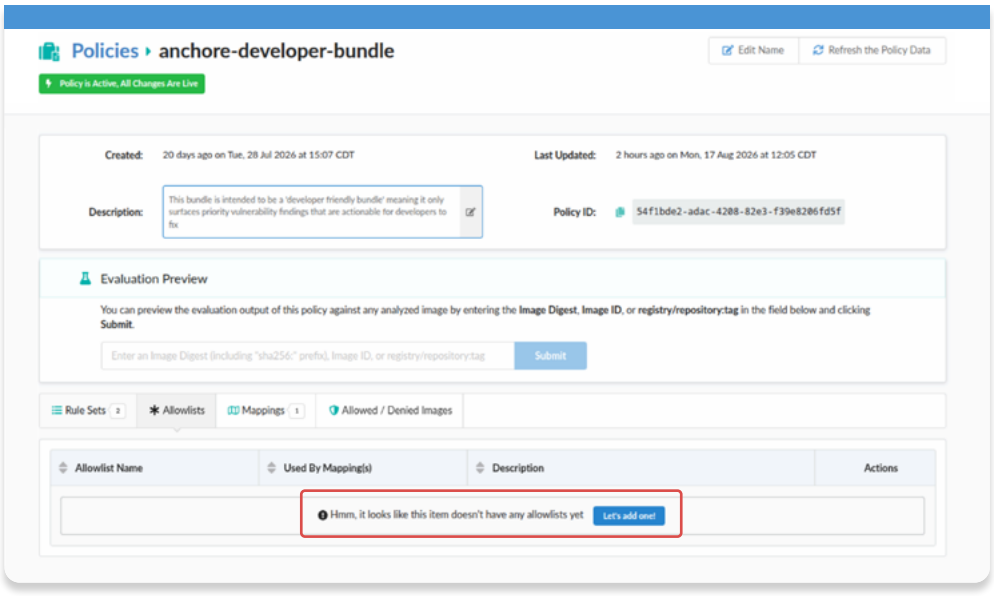



## Allowlist




If applying the security fix causes breaking changes to the application, then a justification can be added to an Allowlist. Once a finding is added to an Allowlist in Anchore, the **STOP** action will instead return a GO action in the policy evaluation and no longer cause the CI job to fail.

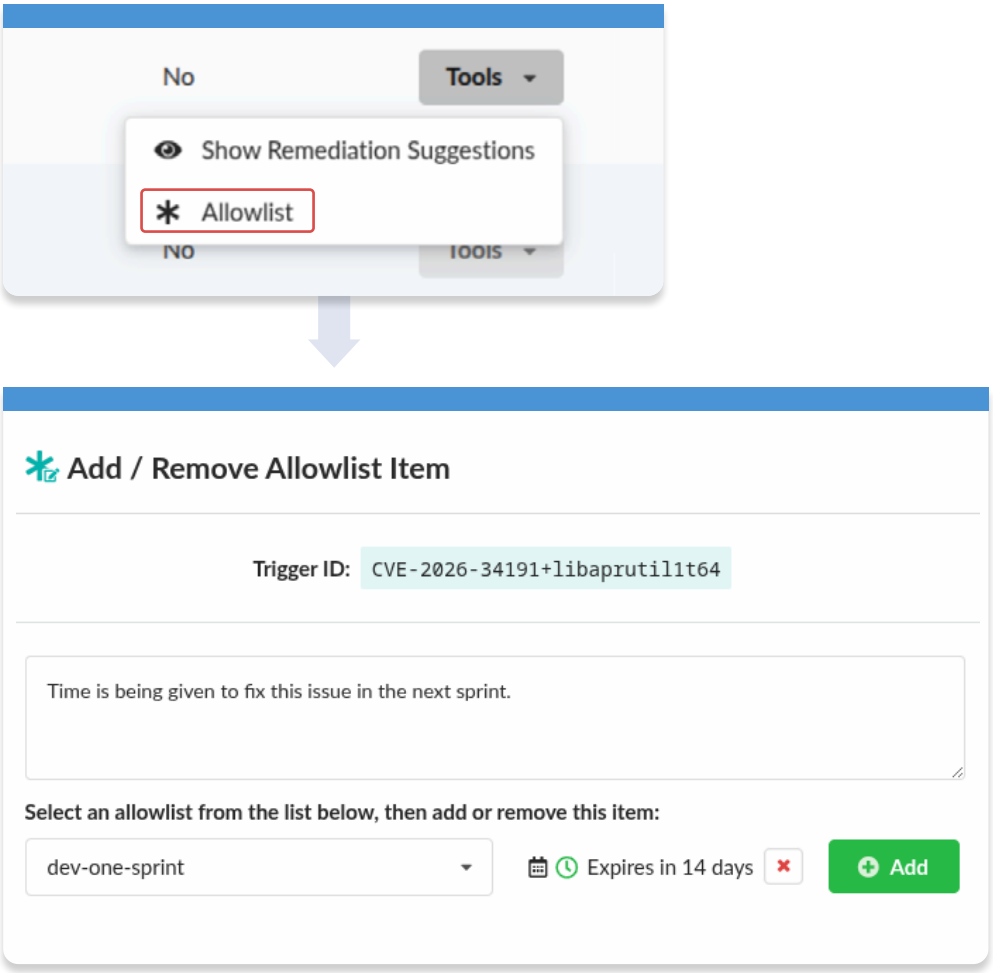
Allowlists are defined in the Policy. To add an Allowlist to the **anchore-developer-bundle**, navigate to Policies > anchore-developer-bundle and select the Allowlists tab. From here you can add an Allowlist and name it appropriately.



Now in addition to Remediation Suggestions, there is the option to use the Allowlist for policy violations.

Add a description for the Allowlist item, select the appropriate list, and add an expiration date (or remove the expiration to indefinitely accept the violation).

The CI job can now be run again and will not be failed due to this violation until after the expiration date is met.




## Conclusion

By integrating Anchore Enterprise into CI/CD processes, vulnerabilities and compliance violations can be addressed early in the development process. Anchore Enterprise bridges the gap between software engineers and cybersecurity engineers by providing a shared platform that helps identify, track, and remediate security vulnerabilities.

For further details on integrating Anchore Enterprise with your CI/CD pipeline, refer to the official documentation or contact our team.

### References


Anchore GitLab CI/CD Integration

AnchoreCTL

Anchore Examples Repository

GitLab CI/CD YAML syntax reference

Docker CLI reference

## About Anchore

Anchore enables organizations to speed digital transformation and reduce risks by streamlining the development of secure and compliant cloud-native applications. Anchore's solutions integrate with existing DevOps toolchains to automate security and compliance checks throughout the software development lifecycle. Organizations can reduce costs and accelerate time to market by remediating security and compliance issues early and continuously.

To learn more about Anchore's solutions, visit **Anchore.com**

Book a demo

anchore

©2026 Anchore

sales@anchore.com

anchore.com