

Mend Bolt Tool

Mend Bolt (formerly WhiteSource Bolt) is an automated security and compliance tool designed for Azure DevOps. It scans open-source dependencies in a project to identify security vulnerabilities, outdated libraries, and licensing issues. Mend Bolt helps development teams maintain secure and compliant software by providing actionable insights directly within Azure DevOps.

Key Features

- **Automated Security Scanning:** Continuously scans open-source components to detect known vulnerabilities.
- **License Compliance Monitoring:** Identifies open-source licenses in use and checks for compliance risks.
- **Vulnerability Remediation:** Suggests updated versions or alternative dependencies to mitigate security risks.
- **Integration with CI/CD Pipelines:** Works within Azure DevOps pipelines, allowing security checks to be automated as part of the software development lifecycle.
- **Reports and Alerts:** Provides detailed reports on vulnerabilities, along with severity levels and remediation steps.

Use Cases

1. Secure Software Development

- Developers can integrate Mend Bolt into their Azure DevOps pipelines to ensure that third-party dependencies do not introduce security risks.
- Automated security scans help catch vulnerabilities early in the development cycle.

2. License Risk Management

- Organizations can prevent legal issues by ensuring that open-source licenses comply with company policies.
- The tool alerts teams if a dependency has an incompatible or high-risk license.

3. Continuous Monitoring in CI/CD Pipelines

- Teams implementing DevSecOps practices can use Mend Bolt to run security checks automatically during build and release stages.
- Security issues can be flagged before deployment, reducing production risks.

4. Compliance with Industry Standards

- Companies following security and compliance regulations (e.g., GDPR, ISO 27001) can use Mend Bolt to maintain adherence to best practices.

5. Risk Prioritization and Management

- Development teams can prioritize vulnerabilities based on severity and exploitability, focusing on the most critical issues first.

Benefits

- **Proactive Security:** Detects and fixes vulnerabilities before they reach production.
- **Time and Cost Savings:** Reduces manual security audits and lowers the cost of fixing security issues later in development.
- **Enhanced Compliance:** Helps organizations adhere to open-source licensing policies and industry regulations.
- **Seamless Integration:** Works natively with Azure DevOps, allowing teams to incorporate security without disrupting workflows.
- **Actionable Insights:** Provides clear guidance on how to fix security vulnerabilities and update dependencies.

Mend Bolt is a valuable tool for organizations looking to enhance the security of their software supply chain while ensuring compliance with open-source licensing requirements.

Summary

In this lab, we are integrating the **Mend Bolt** security tool into an Azure DevOps pipeline to scan for vulnerabilities in our application. Mend Bolt analyzes dependencies and checks for security risks.

Steps Overview

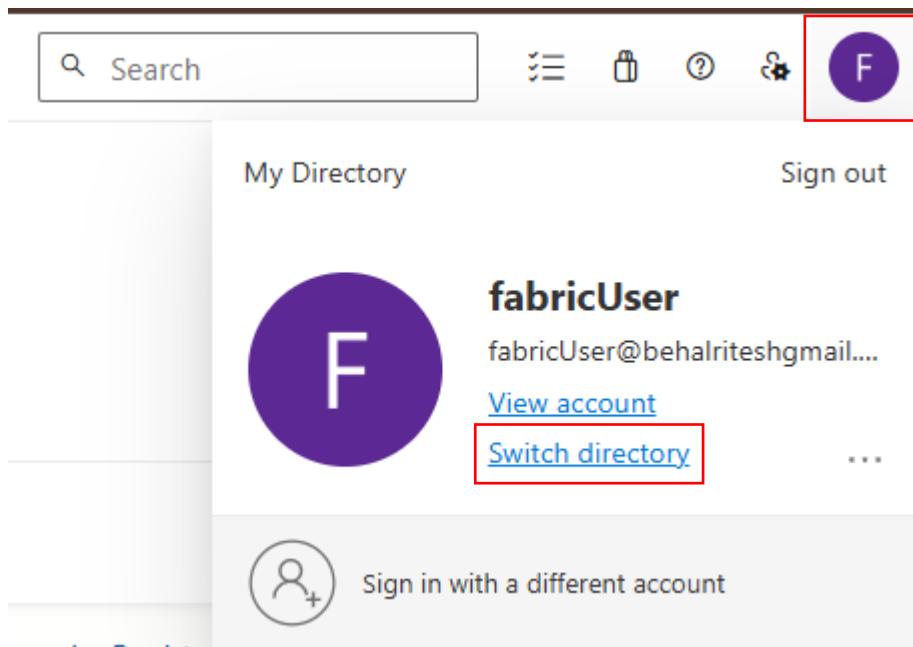
1. **Ensure Azure DevOps is linked to Microsoft Entra ID (Azure AD).**
2. **Install the Mend Bolt extension** from the Azure DevOps Marketplace.
3. **Create a new pipeline** for an application stored in Azure Repos.
4. **Modify the pipeline YAML** to include the Mend Bolt security scan task.
5. **Run the pipeline** and review the Mend Bolt security report.

End Goal

The objective is to **integrate security scanning** into the CI/CD pipeline, ensuring that the application dependencies are **free from vulnerabilities** before deployment.

To begin with the Lab

1. Now to use the mend bolt tool, your Azure DevOps account should be linked with your Microsoft Entra ID or Azure Active Directory.
2. Click on your user profile then choose to Switch directory.



3. Check that your current directory is the same as the Azure active directory. If not then you will see one more directory here. OR you will see a message to switch to another directory.

Switch to another directory X

You are currently connected to the My Directory Microsoft Entra directory. Select a directory.

Current directory

My Directory
behalriteshgmail.onmicrosoft.com
3d18f3ae-8875-4771-aaa9-a9afcd43d751

Directories

i You have no other directories to switch to.

4. Now go to Organization settings choose the extension tab then click on Browse Marketplace.

The screenshot shows the 'Organization Settings' page for 'fabricUser0529'. On the left, a sidebar lists 'General' settings like Overview, Projects, Users, Billing, Global notifications, Usage, Extensions (which is selected), and Microsoft Entra. The main area is titled 'Extensions' with tabs for 'Installed', 'Requested', and 'Shared'. A red box highlights the 'Browse marketplace' button in the top right corner. Below the tabs, it says 'No installed extensions were found matching your criteria.'

5. In the Marketplace search for Mend and choose the free version and install it.

The screenshot shows the Azure DevOps Marketplace search results for 'Mend'. The search bar contains 'Mend'. There are two results displayed:

- Mend Bolt** (WhiteSource) - FREE: This extension is highlighted with a red box. It has a 4-star rating and 24.6K downloads. The description states: 'Get real-time security alerts and compliance issues on your open source...'. It is marked as 'FREE'.
- Mend for Azure DevOps** (WhiteSource) - PAID: This extension has a 5-star rating and 4.3K downloads. The description states: 'Get real-time security alerts and compliance issues on your open source...'. It is marked as 'PAID'.

The screenshot shows the 'Mend Bolt' extension installation dialog. At the top, there are 'Organization' and 'Done' buttons. The main area is titled 'Select an Azure DevOps organization' with a dropdown menu containing 'fabricUser0529'. Below this is a large blue button labeled 'Install'. To the right, there is a 'Permissions' section stating: 'The extension uses the following permissions: Build (read)'. At the bottom, there is a 'Terms of Service' section with a link to the license and privacy policy.

6. After installing it you can see it in your extensions tab in the organization settings.

The screenshot shows the Azure DevOps Organization Settings page for 'fabricUser0529'. The left sidebar has 'Extensions' selected. The main area is titled 'Extensions' with tabs for 'Installed', 'Requested', and 'Shared'. One extension, 'Mend Bolt' by WhiteSource, is listed as 'Active'. Its description is: 'Get real-time security alerts and compliance issues on your open source'.

7. If you scroll to the bottom on the left pane, you will see the extension tab here and you have to open mend from here. Then you will see that it will ask you to create an account. Complete this process.

The screenshot shows the 'Mend' extension configuration page. The left sidebar has 'Mend' selected. The main area displays a form to create a Mend Bolt account, asking for First Name, Last Name, Work Email, Company, Phone, and Country. A note at the bottom states: 'BY CLICKING 'CREATE ACCOUNT' YOU AGREE TO THE TERMS OF SERVICE AND ACKNOWLEDGE OUR DATA COLLECTION AND PROCESSING PRACTICES SET FORTH IN OUR PRIVACY POLICY.' A 'CREATE ACCOUNT' button is at the bottom.

8. Now we are going to create a pipeline for our application stored in Azure Repos. Click on Create Pipeline.
9. Choose Azure Repos Git.

Azure DevOps fabricUser0529 / devProject / Pipelines

devProject +

Connect Select Configure Review

New pipeline

Where is your code?

- Azure Repos Git YAML
Free private Git repositories, pull requests, and code search
- Bitbucket Cloud YAML
Hosted by Atlassian
- GitHub YAML
Home to the world's largest community of developers
- GitHub Enterprise Server YAML
The self-hosted version of GitHub Enterprise

10. Then choose your Web application and move to review page.

✓ Connect Select Configure Review

New pipeline

Select a repository

Filter by keywords devProject X

- WebApp

11. Now while reviewing your YAML code for the pipeline in the Tasks section search for Mend and click on it.

New pipeline

Review your pipeline YAML

Variables Run

```
◆ WebApp / azure-pipelines.yml □
26   . . . solution: '$(solution)'
27   . . . msbuildArgs: '/p:DeployOnBuild=true /p:WebPublishMethod=Package /p:PlatformName=Any CPU /p:ConfigurationName=Release'
28   . . . platform: '$(buildPlatform)'
29   . . . configuration: '$(buildConfiguration)'
30
31     Settings
32       - task: VSTest@2
33         inputs:
34           . . . platform: '$(buildPlatform)'
35           . . . configuration: '$(buildConfiguration)'
```

Tasks

Mend

Mend Bolt (formerly WhiteSource)
Detect security vulnerabilities, problematic open s...

12. Then keep everything as it is and click on add to work with mend bolt too. Also, delete the task starting from line 31 then add the Mend bolt.

New pipeline

Review your pipeline YAML

Variables Run

```
◆ WebApp / azure-pipelines.yml □
26   . . . solution: '$(solution)'
27   . . . msbuildArgs: '/p:DeployOnBuild=true /p:WebPublishMethod=Package /p:PlatformName=Any CPU /p:ConfigurationName=Release'
28   . . . platform: '$(buildPlatform)'
29   . . . configuration: '$(buildConfiguration)'
30
31     Settings
32       - task: VSTest@2
33         inputs:
34           . . . platform: '$(buildPlatform)'
35           . . . configuration: '$(buildConfiguration)'
```

← Mend Bolt (formerly WhiteSource) ⓘ

Root working directory * ⓘ
\$(System.DefaultWorkingDirectory)

Project name ⓘ

Scan report timeout (minutes) ⓘ
10

Add

13. Here you can see that the mend bolt task has been added starting from line 31 to 33.

New pipeline

Review your pipeline YAML

WebApp / azure-pipelines.yml * □

```
Settings
24  - task: VSBuild@1
25    inputs:
26      solution: '$(solution)'
27      msbuildArgs: '/p:DeployOnBuild=true /p:WebPublishMethod=Package /p:PackageName=WebApp /p:PlatformName=Any CPU /p:ConfigurationName=Release'
28      platform: '$(buildPlatform)'
29      configuration: '$(buildConfiguration)'
30
Settings
31  - task: WhiteSource@21
32    inputs:
33      cwd: '$(System.DefaultWorkingDirectory)'
```

14. In the end just save and run your pipeline and wait for the job to get completed.

15. Below you can see that your pipeline run has been completed.

The screenshot shows the Azure Pipelines interface. On the left, there's a list of jobs for run #20250320.2 under the 'WebApp' project. The 'Job' step is expanded, showing its sub-tasks: Initialize job, Checkout WebApp@m... (with a warning), NuGetToolInstaller, NuGetCommand, VSBuild, WhiteSource, Post-job: Checkout W..., Finalize Job, and Report build status. Each task has a duration listed next to it. On the right, a detailed view of the 'Job' step is shown, with a green checkmark icon and the word 'Job'. The log output for the 'Job' step is displayed, showing the agent request, preparation parameters, live console data, and the start, async command, and finish logs for the Docker container.

Task	Duration
Initialize job	7s
Checkout WebApp@m...	8s
NuGetToolInstaller	2s
NuGetCommand	45s
VSBuild	1m 14s
WhiteSource	2m 14s
Post-job: Checkout W...	<1s
Finalize Job	<1s
Report build status	<1s

```
1 Pool: Azure Pipelines
2 Image: windows-latest
3 Queued: Today at 8:36 AM [manage_parallel_jobs]
4 Agent: Hosted Agent
5 Started: Today at 8:36 AM
6 Duration: 4m 34s
7
8 The agent request is already running or has already completed.
9 ▶ Job preparation parameters
44 Job live console data:
45 Starting: Job
46 Async Command Start: DetectDockerContainer
47 Async Command End: DetectDockerContainer
48 Async Command Start: DetectDockerContainer
49 Async Command End: DetectDockerContainer
50 Finishing: Job
```

16. You can see a new tab for Mend bold after the completion of your Job. Open it.

#20250320.2 • Set up CI with Azure Pipelines

Run new : WebApp

mend_scan_start_time=Thu, 20 Mar 2025 03_08_56 GMT mend_support_token=0rxke6lgdhdoxnggbfgjotrtrfbemzsy07leig9f7 build_report_status=ready

This run is being retained as one of 3 recent runs by master (Branch). View retention leases

Summary Code Coverage **Mend Bolt**

Manually run by  fabricUser

View 5 changes

Repository and version

WebApp master ↗ 768c3c81

Time started and elapsed

Today at 8:36 AM 4m 42s

Related 0 work items 0 artifacts

Tests and coverage 

Jobs

Name	Status	Duration
Job	Success	4m 34s

17. And according to mend there is no risk while running your pipeline.

#20250320.2 • Set up CI with Azure Pipelines

Run new : WebApp

mend_scan_start_time=Thu, 20 Mar 2025 03_08_56 GMT mend_support_token=0rxke6lgdhdoxnggbfgjotrtrfbemzsy07leig9f7 build_report_status=ready

This run is being retained as one of 3 recent runs by master (Branch). View retention leases

Summary Code Coverage **Mend Bolt**

Open Source Risk Report

Total libraries: 0

Mend could not detect any open-source components in this pipeline

No Risk

Vulnerability Risk Vulnerable Libraries

0

Vulnerabilities Severity Distribution

Critical	High	Medium	Low
0	0	0	0

Inventory (0) Security vulnerabilities (0) License risks

Library Licenses