Lab: Implement your own secure DevOps pipeline (DevSecOps)

Lab scenario

MyHealthClinic are shifting left and some of the developers have already made their own pipelines with no guidance. Bertie and Maxine review these pipelines and can see that they don't meet best practices and do not adhere to zero trust principles, they need to help the developers improve the security of the pipelines.

Commented [JK1]: Change scenario to MyHealthClinic

Objectives

After completing this lab, you will be able to:

- Configure the MyHealthClinic project on Azure DevOps.
- Integrate software composition analysis on Azure DevOps Pipeline
- Configure the Microsoft Security DevOps Azure DevOps extension.
- Configuring pipeline security and settings

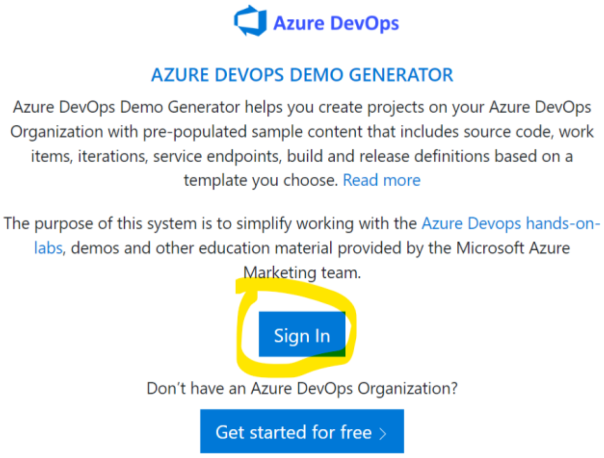
Estimated Time: 1 Hour

Lab Environment

- Azure DevOps

Exercise 1: Configure the MyHealthClinic project on Azure DevOps

1. Navigate to https://azuredevopsdemogenerator.azurewebsites.net . This utility site will automate the process of creating a new Azure DevOps project within your account that is prepopulated with content (work items, repos, etc.) required for the lab.
2. Sign in using the Microsoft account associated with your Azure DevOps subscription.



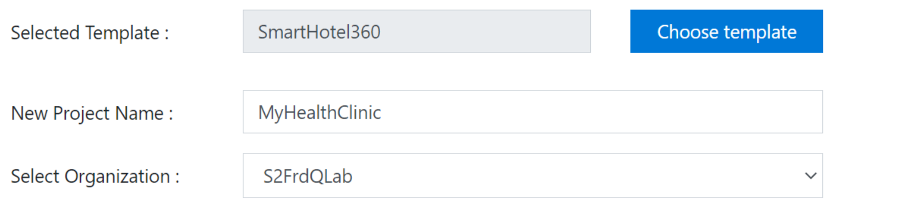
Azure DevOps

AZURE DEVOPS DEMO GENERATOR

Azure DevOps Demo Generator helps you create projects on your Azure DevOps Organization with pre-populated sample content that includes source code, work items, iterations, service endpoints, build and release definitions based on a template you choose. Read more

The purpose of this system is to simplify working with the Azure Devops hands-on-labs, demos and other education material provided by the Microsoft Azure Marketing team.

3. Select your Azure DevOps organization and enter the project name **“MyHealthClinic”** then Click **Choose Template**.

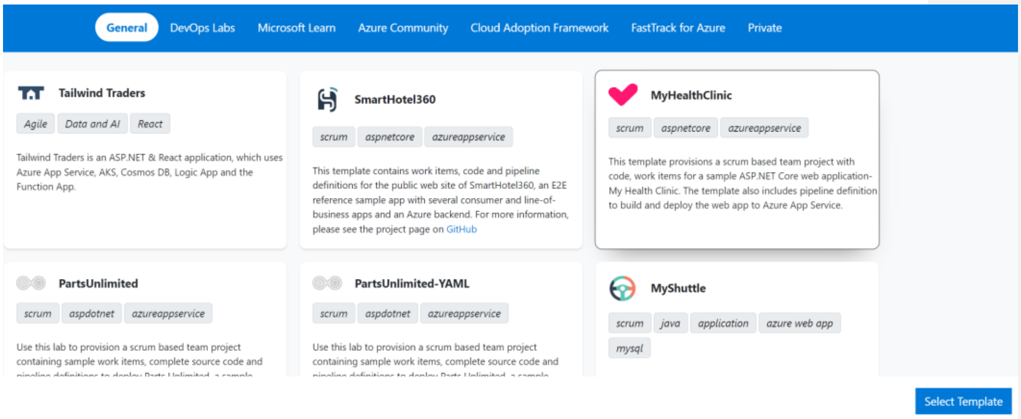


Selected Template : SmartHotel360

New Project Name : MyHealthClinic

Select Organization : S2FrdQLab

4. Select the **MyHealthClinic** template and click Select Template.



Tailwind Traders

Agile | Data and AI | React

Tailwind Traders is an ASP.NET & React application, which uses Azure App Service, AKS, Cosmos DB, Logic App and the Function App.

SmartHotel360

scrum | aspnetcore | azureappservice

This template contains work items, code and pipeline definitions for the public web site of SmartHotel360, an E2E reference sample app with several consumer and line-of-business apps and an Azure backend. For more information, please see the project page on GitHub

MyHealthClinic

scrum | aspnetcore | azureappservice

This template provisions a scrum based team project with code, work items for a sample ASP.NET Core web application- My Health Clinic. The template also includes pipeline definition to build and deploy the web app to Azure App Service.

PartsUnlimited

scrum | aspdotnet | azureappservice

Use this lab to provision a scrum based team project containing sample work items, complete source code and pipeline definitions to deploy Parts Unlimited, a sample

PartsUnlimited-YAML

scrum | aspdotnet | azureappservice

Use this lab to provision a scrum based team project containing sample work items, complete source code and pipeline definitions to deploy Parts Unlimited, a sample

MyShuttle

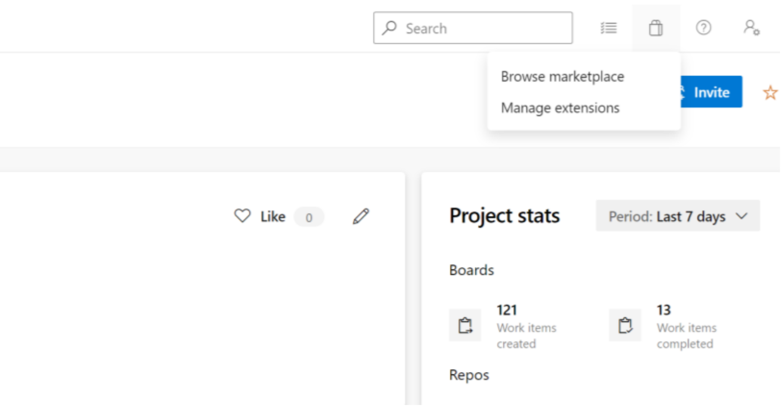
scrum | java | application | azure web app | mysql

5. Click **Create Project** and wait for the process to complete.

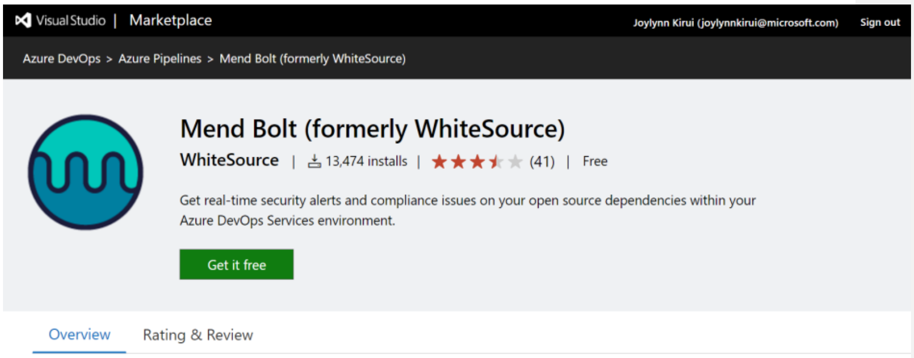
Exercise 2: Integrate software composition analysis on Azure DevOps Pipeline

In this exercise, we will use **Mend Bolt (formerly WhiteSource)** to automatically detect vulnerable open source components, outdated libraries, and license compliance issues in the code.

1. Activate Mend Bolt extension – on your Azure DevOps portal with the **MyHealthClinic** project open, click on the marketplace icon > **Browse Marketplace**.



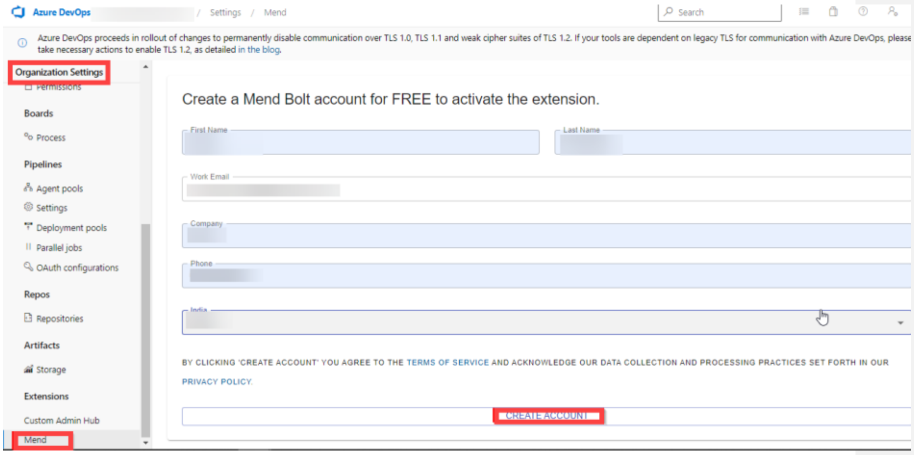
2. On the Marketplace, search for **Mend Bolt (formerly WhiteSource)** and open it. Mend Bolt is the free version of the previously known WhiteSource tool, which scans all your projects and detects open source components, their license and known vulnerabilities.
3. On the **Mend Bolt (formerly WhiteSource)** page, click on **Get it for free**.


Visual Studio | Marketplace
Joylynn Kirui (joylynnkirui@microsoft.com)
Sign out
Azure DevOps > Azure Pipelines > Mend Bolt (formerly WhiteSource)
Mend Bolt (formerly WhiteSource)
WhiteSource | 13,474 installs | (41) | Free

Get real-time security alerts and compliance issues on your open source dependencies within your Azure DevOps Services environment.

Get it free
Overview
Rating & Review

4. On the next page, select the desired Azure DevOps organization and Install. Proceed to organization once installed.
5. In your Azure DevOps navigate to **Organization Settings** and select **Mend** under **Extensions**. Provide your Work Email, Company Name and other details and click **Create Account** button to start using the Free version.



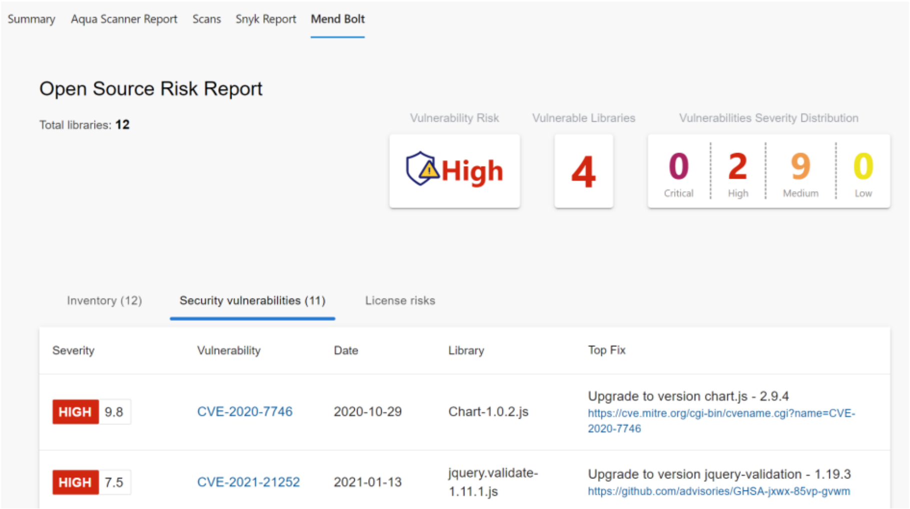
Azure DevOps proceeds in rollout of changes to permanently disable communication over TLS 1.0, TLS 1.1 and weak cipher suites of TLS 1.2. If your tools are dependent on legacy TLS for communication with Azure DevOps, please take necessary actions to enable TLS 1.2, as detailed in the blog.

Create a Mend Bolt account for FREE to activate the extension.

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.

6. In this step, you will create and trigger a CI build pipeline within Azure DevOps project. You will use **Mend Bolt** extension to identify vulnerable OSS components present in this code.

    i. In the MyHealthClinic Azure DevOps project, in the vertical menu bar on the left side, navigate to the Pipelines>Pipelines section, click New Pipeline.
    ii. On the Where is your code? window, select Azure Repos Git (YAML) and select the MyHealthClinic repository.
    iii. On the Configure section, choose Starter pipeline. Click Show assistant and search for mend, select the Mend Bolt (formerly WhiteSource) task, this will add the Mend Bolt task to your starter yaml file.
    iv. Click Save and run and let the pipeline run. You can check progress by going to Pipeline-Pipelines and select the running pipeline.
    v. When done, you can view security vulnerabilities found by Mend Bolt, by clicking Mend Bolt



Open Source Risk Report

Total libraries: **12**

Vulnerability Risk	Vulnerable Libraries	Vulnerabilities Severity Distribution
High	4	0 Critical, 2 High, 9 Medium, 0 Low

Severity	Vulnerability	Date	Library	Top Fix
HIGH 9.8	CVE-2020-7746	2020-10-29	Chart-1.0.2.js	Upgrade to version chart.js - 2.9.4 https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2020-7746
HIGH 7.5	CVE-2021-21252	2021-01-13	jquery.validate-1.11.1.js	Upgrade to version jquery-validation - 1.19.3 https://github.com/advisories/GHSA-jxwx-85vp-gvwm

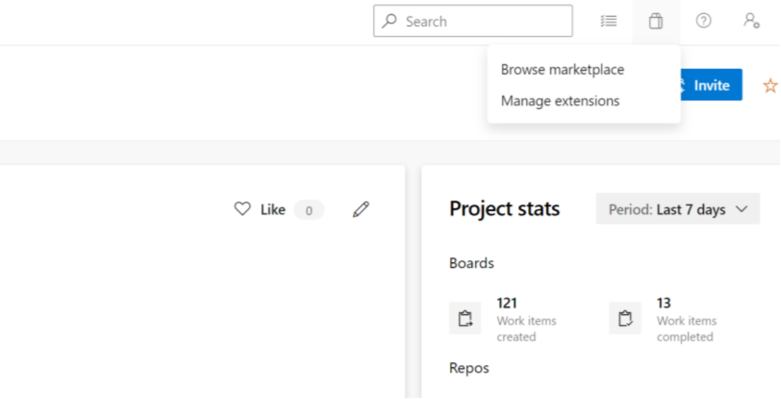
Exercise 3: Configure the Microsoft Security DevOps Azure DevOps extension.

In this task, you will configure Defender for DevOps. Microsoft Security DevOps is a command line application that integrates static analysis tools into the development lifecycle. Microsoft Security DevOps installs, configures, and runs the latest versions of static analysis tools (including, but not limited to, SDL/security and compliance tools).

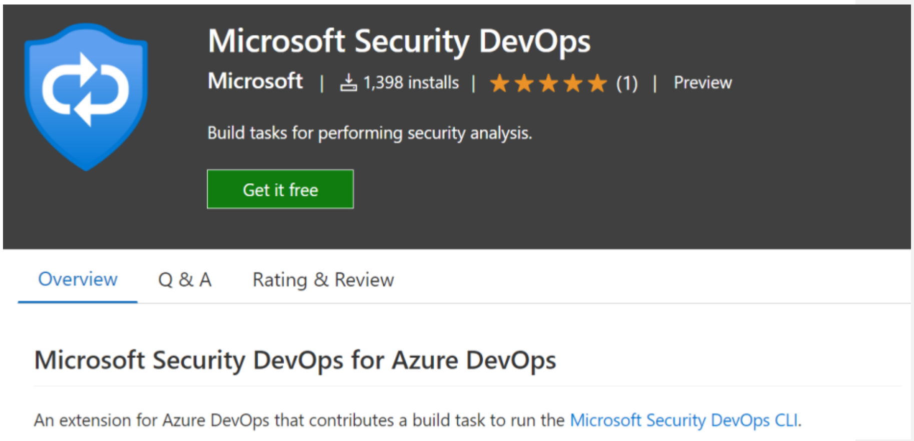
The Microsoft Security DevOps uses the following Open Source tools: Bandit, BinSkim, ESlint, Credscan, Template Analyzer, Terrascan and Trivy.

> **Note:** Admin privileges to the Azure DevOps organization are required to install the extension.

1. Activate Microsoft Security DevOps extension – on your Azure DevOps portal with the **MyHealthClinic** project open, click on the marketplace icon > **Browse Marketplace**.



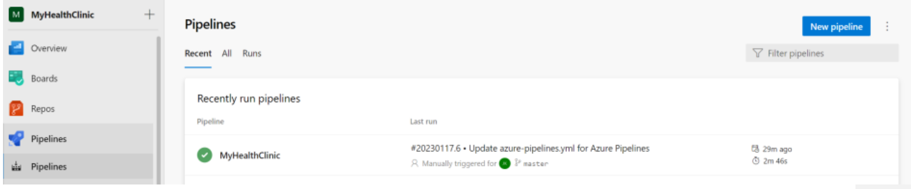
2. On the Marketplace, search for **Microsoft Security DevOps** and open it.
3. On the **Microsoft Security DevOps** page, click on **Get it for free**.


Microsoft Security DevOps
Microsoft | 1,398 installs | ★★★★★ (1) | Preview
Build tasks for performing security analysis.
Get it free
Overview
Q & A
Rating & Review

Microsoft Security DevOps for Azure DevOps

An extension for Azure DevOps that contributes a build task to run the Microsoft Security DevOps CLI.

4. On the next page, select the desired Azure DevOps organization and Install. Proceed to organization once installed.
5. Navigate to your MyHealthClinic projects, then Pipelines and Click New pipeline.

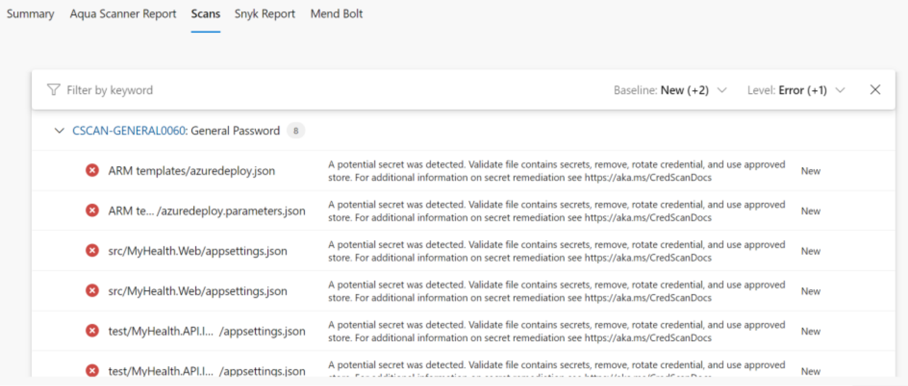


6. On the Where is your code? window, select Azure Repos Git (YAML) and select the MyHealthClinic repository.
7. On Add the following scripts as in into the yaml file.

```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: none
pool:
  vmImage: 'windows-latest'
steps:
- task: UseDotNet@2
  displayName: 'Use dotnet'
  inputs:
    version: 3.1.x
- task: UseDotNet@2
  displayName: 'Use dotnet'
  inputs:
    version: 5.0.x
- task: UseDotNet@2
  displayName: 'Use dotnet'
  inputs:
    version: 6.0.x
- task: MicrosoftSecurityDevOps@1
  displayName: 'Microsoft Security DevOps'
```

8. Click Save and run and let the pipeline run. You can check progress by going to Pipeline-Pipelines and select the running pipeline.
9. When done, you can view security vulnerabilities found by **Microsoft Security DevOps** , by clicking Scans.

Note: Install the SARIF SAST Scans Tab extension on the Azure DevOps organization in order to ensure that the generated analysis results will be displayed automatically under the Scans tab.



CSCAN-GENERAL0060: General Password 8

ARM templates/azuredeploy.json	A potential secret was detected. Validate file contains secrets, remove, rotate credential, and use approved store. For additional information on secret remediation see https://aka.ms/CredScanDocs	New
ARM te... /azuredeploy.parameters.json	A potential secret was detected. Validate file contains secrets, remove, rotate credential, and use approved store. For additional information on secret remediation see https://aka.ms/CredScanDocs	New
src/MyHealth.Web/appsettings.json	A potential secret was detected. Validate file contains secrets, remove, rotate credential, and use approved store. For additional information on secret remediation see https://aka.ms/CredScanDocs	New
src/MyHealth.Web/appsettings.json	A potential secret was detected. Validate file contains secrets, remove, rotate credential, and use approved store. For additional information on secret remediation see https://aka.ms/CredScanDocs	New
test/MyHealth.API.I... /appsettings.json	A potential secret was detected. Validate file contains secrets, remove, rotate credential, and use approved store. For additional information on secret remediation see https://aka.ms/CredScanDocs	New
test/MyHealth.API.I... /appsettings.json	A potential secret was detected. Validate file contains secrets, remove, rotate credential, and use approved	New

https://learn.microsoft.com/en-us/azure/defender-for-cloud/azure-devops-extension

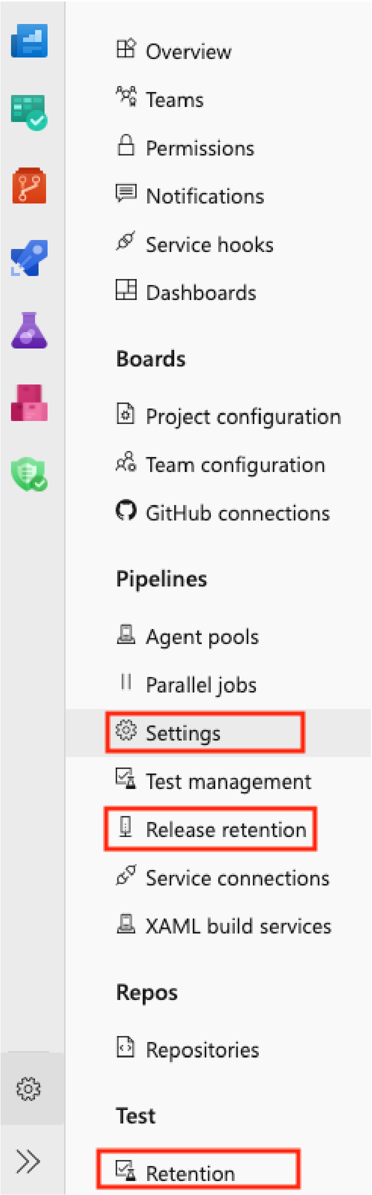
Exercise 4: Configuring pipeline security and settings

In this exercise, we will cover how you can configure retention policies and pipeline permissions as some of the security hardening on your pipeline. Retention policies let you set how long to keep runs, releases, and tests stored in the system. To save storage space, you want to delete older runs, tests, and releases.

The following retention policies are available in Azure DevOps in your **Project settings**:

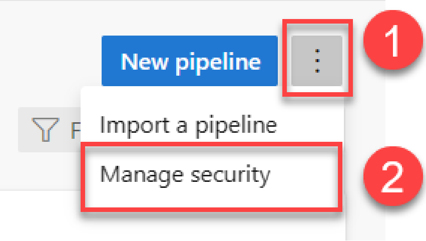
1. **Pipeline** - Set how long to keep artifacts, symbols, attachments, runs, and pull request runs.
2. **Release (classic)** - Set whether to save builds and view the default and maximum retention settings.
3. **Test** - Set how long to keep automated and manual test runs, results, and attachments.

1. Navigate to your project.
2. Go to the ⚙ Settings tab of your project's settings.
3. Select Settings or Release retention under Pipelines or Retention under Test.
    - Select Settings to configure retention policies for runs, artifacts, symbols, attachments, and pull request runs.
    - Select Release retention to set up your release retention policies and configure when to delete or permanently destroy releases.
    - Select Retention to set up how long to keep manual and automated test runs.



4. Now let's set pipeline permissions. Pipeline permissions and roles help you securely manage your pipelines. You can set hierarchical permissions at the organization, project, and object levels for all pipelines in a project or for an individual pipeline. You can update pipeline permissions with security groups or by adding individual users.
5. From within your project, select **Pipelines** > **Pipelines**

6. Select **More actions** ⋮ > **Manage security**.



7. Modify the permissions associated with an [Azure DevOps group](), for example, Build Administrators, or for an [individual user]().
8. Select **Allow** or **Deny** for the permission for a security group or an individual user, and then exit the screen.

   Your project-level pipelines permissions are set.