

[OPEN HACK ENVIRONMENT](#)   
[TEAM CHAT](#)

[OVERVIEW](#)

[OPEN HACK GUIDE](#)

[PROVIDE FEEDBACK](#)

[MESSAGES](#)

1 2 3

☐ Mark Complete

---

# Implement continuous testing

**Resist the initial temptation to rush on the keyboard! This is the time to pause and think as a team about your end-to-end deployment process that you would implement in this company.**

## Reminders:

- Use an InPrivate/Incognito window in your browser to avoid any confusion with any other credentials that you may use to access Azure resources.

---

Regardless of how carefully you write and review code, workflow automation not only increases velocity but also helps you avoid one-off issues, regressions and the "works on my machine" phenomenon.

## Challenge

You are expected to create workflows to automate the **building, unit testing and packaging** of at least one of the APIs of your application. You are not required to publish code coverage reports or resulting packages. Instead, focus exclusively on executing existing unit tests as part of your workflows. Carefully review the Readme for chosen API as it may contain additional instructions.

**Note:** When selecting GitHub Actions, Azure DevOps Tasks or Jenkins Plugins; be prudent and select ones that were created by the vendor as a first choice. If you are going to use a third party option make sure that you verify that you are able to satisfy the success criteria with your selection.

## Success Criteria

- With your coach present, execute the automation workflow and explain each step.
- Demonstrate that the workflow is triggered for an API, only when changes are made to code for that API.
- Demonstrate that the workflow(s) augment existing branch protections, by performing verifications before a Pull Request (PR) can be merged to your protected branch.
- Demonstrate that your pipeline runs, performs all of the required steps and then ultimately prevents the merging of the code when the workflow fails.
- Demonstrate that your workflow creates and issue/bug in your backlog for each failed build

**Note:** You are required to make a change that will cause one of your unit tests to fail. If you are uncertain, your coach will provide a breaking change to submit through your pipeline.

## References

A cheat sheet for the DevOps OpenHack is available at the end of the **Overview** page.

- [What is Continuous Integration?](https://docs.microsoft.com/azure/devops/learn/what-is-continuous-integration) (<https://docs.microsoft.com/azure/devops/learn/what-is-continuous-integration>)
- [Continuous Integration: What is CI? Testing, Software & Process Tutorial](https://codeship.com/continuous-integration-essentials) (<https://codeship.com/continuous-integration-essentials>)

# Workflow orchestration

What is Pipeline as Code? (<https://www.youtube.com/watch?v=C7NFwlmSAqU>)

- GitHub:
  - [GitHub Actions](https://github.com/features/actions) (<https://github.com/features/actions>)
  - [GitHub Actions Schema and Syntax Reference](https://help.github.com/actions/reference) (<https://help.github.com/actions/reference>)
- Azure DevOps
  - [Azure Pipelines](https://docs.microsoft.com/azure/devops/pipelines/get-started/what-is-azure-pipelines?view=azure-devops) (<https://docs.microsoft.com/azure/devops/pipelines/get-started/what-is-azure-pipelines?view=azure-devops>)
  - [Azure Pipelines Schema and Syntax Reference](https://docs.microsoft.com/azure/devops/pipelines/yaml-schema?view=azure-devops&tabs=schema%2Cparameter-schema) (<https://docs.microsoft.com/azure/devops/pipelines/yaml-schema?view=azure-devops&tabs=schema%2Cparameter-schema>)
- Jenkins:
  - [Creating a Jenkins pipeline](https://jenkins.io/doc/book/pipeline/) (<https://jenkins.io/doc/book/pipeline/>)
  - [GitHub Permissions and API token scopes for Jenkins](https://support.cloudbees.com/hc/articles/234710368-GitHub-Permissions-and-API-token-Scopes-for-Jenkins) (<https://support.cloudbees.com/hc/articles/234710368-GitHub-Permissions-and-API-token-Scopes-for-Jenkins>)
  - [Known issue with Docker workflow plugin and multi stage builds](https://issues.jenkins-ci.org/browse/JENKINS-44609) (<https://issues.jenkins-ci.org/browse/JENKINS-44609>)

## Unit testing

- [Test Driven Development Essay](http://agiledata.org/essays/tdd.html) (<http://agiledata.org/essays/tdd.html>)
- [Writing and running unit tests with GoLang](https://blog.alexellis.io/golang-writing-unit-tests/) (<https://blog.alexellis.io/golang-writing-unit-tests/>)
- [Writing and running unit tests with Node.JS](https://blog.risingstack.com/node-hero-node-js-unit-testing-tutorial/) (<https://blog.risingstack.com/node-hero-node-js-unit-testing-tutorial/>)
- [Writing and running units test with .Net Core](https://docs.microsoft.com/dotnet/core/testing/unit-testing-with-dotnet-test) (<https://docs.microsoft.com/dotnet/core/testing/unit-testing-with-dotnet-test>)
- [Writing and running unit tests with Java](https://www.vogella.com/tutorials/Mockito/article.html) (<https://www.vogella.com/tutorials/Mockito/article.html>)

OpenHack © Microsoft 2019. All Rights Reserved - Powered By [Opsgility Cloud Sandbox](#).  
[Privacy](#).

