

References

- [What is DevOps?](#)
- [Bridging the gap between business and development](#)
- [Pair programming](#)

Getting Started

There are numerous options for managing source code during this OpenHack and in other DevOps activities. Many choose GitHub because of its relationship to Open Source, however the advanced DevOps, security and inner source capabilities may also help with your decision. Some will choose Azure DevOps, or a combination of GitHub and Azure DevOps. And others may choose other tools for work management, source control and/or CI/CD.

Note: If using GitHub or Azure DevOps, public repos and projects are recommended for the OpenHack as they offer additional benefits (e.g. additional concurrent CI/CD jobs, etc.)

- GitHub
 - [Setting up Git](#)
 - [Getting started with GitHub](#)
 - [Creating a GitHub Organization](#)
 - [Organizing work with Project Boards](#)
 - Managing work with [GitHub Issues](#) and [GitHub Projects](#)
 - [Improving code quality and security with GitHub Protected Branches](#)
 - [Linking GitHub Issues](#)
 - [Mastering GitHub Issues](#)
 - [Closing issues via Pull Requests](#)
- Azure DevOps
 - [Create an Azure DevOps organization](#)
 - [What is Azure Boards?](#)
 - [Integrating Azure Boards and GitHub](#)
 - [Improving code quality with Azure Repos branch policies](#)
 - [Link work items to support traceability and manage dependencies](#)
 - [Link work items to GitHub Commits and Pull Requests](#)

Note: You may have to review the permissions of each member of your team at the [organization and project level](#).

References

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

Note: Continuous Integration (CI) uses best practices including automated testing and ideally [trunk-based development](#). Automated testing will be covered later and will extend the CI setup completed here.

- [What is Continuous Integration?](#)
- [Azure DevOps Engineering Release Flow](#)

Work Management

- GitHub
 - [Organizing work with Project Boards](#)
 - [Managing Teams in GitHub](#)
 - Managing work with [GitHub Issues](#) and [GitHub Projects](#)
- Azure DevOps
 - [Azure Boards & GitHub](#)
 - [View and add work items using the Work Items page](#)

Branch protection

- GitHub
 - [GitHub - About protected Branches](#)
 - [GitHub Code Owners](#)
- Azure DevOps
 - [Configure Branch Policies](#)
 - [Code Reviewers](#)

3

References

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

- [What is Continuous Integration?](#)
- [Continuous Integration: What is CI? Testing, Software & Process Tutorial](#)

Workflow orchestration

[What is Pipeline as Code?](#)

- GitHub:
 - [GitHub Actions](#)
 - [GitHub Actions Schema and Syntax Reference](#)
- Azure DevOps
 - [Azure Pipelines](#)
 - [Azure Pipelines Schema and Syntax Reference](#)
- Jenkins:
 - [Creating a Jenkins pipeline](#)
 - [GitHub Permissions and API token scopes for Jenkins](#)
 - [Known issue with Docker workflow plugin and multi stage builds](#)

Unit testing

- [Test Driven Development Essay](#)
- [Writing and running unit tests with GoLang](#)
- [Writing and running unit tests with Node.JS](#)
- [Writing and running units test with .Net Core](#)
- [Writing and running unit tests with Java](#)

4

- Orchestrators
 - [GitHub Actions](#)
 - [Azure Pipelines](#) Part of Azure DevOps, formerly known as Visual Studio Team Services (VSTS)
 - [Jenkins](#)
- Version Control
 - [GitHub](#)
- Docker Registry
 - [Azure Container Registry](#)
 - [GitHub Packages](#)

References

- Continuous Delivery
 - [What is Continuous Delivery?](#)
 - [Why continuous delivery?](#)
- Github
 - [GitHub Actions - Docker build and push](#)
 - [GitHub Actions - Azure WebApp](#)
- Azure Pipelines
 - [Azure Pipelines - Deploy a custom Linux container to Azure App Service](#)
 - [Azure Pipelines - Tutorial: Build a custom image and run in App Service from a private registry](#)
 - [Azure Pipelines - Docker task](#)
- Docker Registries
 - [GitHub Packages Documentation](#)
 - [Azure Container Registry documentation](#)

5

References

- [Blue/Green deployment](#)
- [Staging environments in Azure App Service](#)
- [Azure Slot Deployment with Blue-Green Deployment Model](#)

6

References

Azure resources

- Azure Monitor
 - [Overview of the Azure Monitoring solutions](#)
 - [Getting started with Azure Monitor Log Analytics](#)
 - [Monitoring Azure resources with Azure Monitor](#)
 - [Monitor Azure App Service performance](#)
 - [Monitor Azure SQL Database using Azure SQL Analytics](#)
 - [Create diagnostic setting to collect resource logs and metrics in Azure](#)
 - [Monitor apps in Azure App Service](#)

- [What is Application Insights?](#)
 - [Create, view, and manage log alerts using Azure Monitor](#)
 - [Overview of alerts in Microsoft Azure](#)
- Azure Logic Apps
 - [Overview - What is Azure Logic Apps?](#)
 - [Azure DevOps Connector for Azure Logic Apps](#)
- Kusto documentation and references
 -
 - The solutions in Azure will evolve to use [Azure Data Explorer query language](#) (also know as [Kusto](#))
 - Syntax for regular expressions supported by [Azure Data Explorer](#)
- Regular Expressions
 - You can test your regular expression on this site: [RegExr: Learn, Build, & Test RegEx](#)
 - You can use regular expressions in the `extract()` function in Kusto. For example, this code will extract and convert to a double the response time from the container output showed above:

```
1 | todouble(extract(@"CreateTripPoint ([0-9.]*)ms", 1, LogEntry))
```

- Application availability and responsiveness
 - [Availability alerts of any web site](#)

Additional Reading

- [Azure Monitor overview](#)
- [Application performance FAQs for Web Apps in Azure](#)

7

References

- Dependency Scanning
 - [GitHub Security Alerts](#)
 - [Automated Security Updates with GitHub](#)
 - [Whitesource Bolt with Azure Pipelines](#)
 - [OWASP Dependency Checker in the Azure DevOps marketplace](#)
- Code Coverage
 - [CoverAlls](#)
 - [CodeCov](#)
 - [Jacoco](#)
 - [CoverAlls for GitHub Actions](#)
 - [Azure Pipelines ecosystem \(per-language code coverage docs\)](#)
 - [GitHub Coverage Reporter for Jenkins](#)
- Static Code Analysis
 - [Linters in the GitHub Actions marketplace](#)
 - [Static Code Analysis tools \(Wikipedia\)](#)
 - [SonarCloud for GitHub Actions](#)
 - [Integrate Visual Studio Team Services with SonarCloud](#)
 - [Using SonarCloud in an Azure DevOps pipeline](#)

- Variant Analysis (Security)
 - [Variant Analysis for Security](#)
 - [Continuous security analysis \(with GitHub and SEMMLE\)](#)
- Integration Testing
 - [Integration testing](#)
 - [Integration tests vs Unit tests](#)
 - [Integration test with .Net Core](#)
 - [Integration testing in GO](#)
 - [Set up a go workspace](#)
 - [Integration testing with Node.JS](#)
- Load Testing
 - [Blazemeter with Azure Pipelines](#)
- Manual Approvals
 - [Setting up Webhooks for GitHub Actions](#)
 - [GitHub Deployments \(REST API\)](#)
 - [Release approval gates for Azure Pipelines](#)
 - [Manual Intervention task for Azure Pipelines](#)

8

Reference

- Topical References
 - [Route traffic with Azure App Service and Deployment slots](#)
 - [Explore how to progressively expose your Azure DevOps extension releases in production to validate, before impacting all users](#)
 - [Deployment considerations for DevOps](#)
 - [Canary Deployments Using Azure Web App Deployment Slots](#)
- GitHub
 - [GitHub Action Rollback Release](#)
 - [Context and expression syntax for GitHub Actions](#)
- Azure Pipelines
 - [How to implement rollback strategies in Azure Pipelines](#)