OPEN HACK ENVIRONMENT ☰   **OVERVIEW**   **OPEN HACK GUIDE**   **PROVIDE FEEDBACK**   **MESSAGES**
TEAM CHAT

<div style="border:1px solid orange; display:inline-block">1</div>

☐ **Mark Complete**

Establish your plan

In order to have a successful DevOps strategy, your team needs to have a plan.

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.

Challenge

Your challenge is to organize your team and establish a high level view of the workflow that you will implement during the rest of the event to achieve zero downtime deployment.

Based on your knowledge of the application and its architecture, your team must establish the fundamental processes that need to be followed in order to go from code to production. You are expected to focus only on a basic integration and deployment mechanism that takes into consideration the

architecture of the application.

Get familiar with the basic building blocks of a successful DevOps workflow and be aware that your current plan may change over the course of the event.

This is a good time to think of the following aspects:

- Each API will potentially have a different life cycle.
- Your process should aim at minimizing the time it takes to deploy a feature into production.
- You want to be able to prioritize your tasks.
- Small changes are easier to deploy.

Success Criteria

Present your workflow and initial organization of the team that you have collectively agreed on to your coach. Take time to explain in detail how you aim to validate one phase before you move to the next one.

Be ready to answer the following questions:

- What aspects of DevOps are you implementing?
- How are you splitting the work amongst the team?
- How do you keep track of your work in progress?
- How will you communicate amongst the team (Verbal, Teams, Slack, etc.)?

Your coach will discuss your process before unlocking the next challenge.

References

- What is DevOps? (https://docs.microsoft.com/azure/devops/learn/what-is-devops)
- Bridging the gap between business and development (https://www.developer-tech.com/news/2016/jan/29/devops-bridging-gap-between-business-and-development/)
- Pair programming (https://www.agilealliance.org/glossary/pairing/)

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 (https://help.github.com/github/getting-started-with-github/set-up-git#setting-up-git)
  - Getting started with GitHub (https://help.github.com/github/getting-started-with-github)
  - Creating a GitHub Organization (https://help.github.com/github/setting-up-and-managing-organizations-and-teams/creating-a-new-organization-from-scratch)
  - Organizing work with Project Boards (https://help.github.com/github/managing-your-work-on-github/about-project-boards)
  - Managing work with GitHub Issues (https://help.github.com/github/managing-your-work-on-github/about-issues) and GitHub Projects (https://github.com/features/project-management)
  - Improving code quality and security with GitHub Protected Branches (https://help.github.com/github/administering-a-repository/about-protected-branches)
  - Linking GitHub Issues (https://help.github.com/github/writing-on-github/autolinked-references-and-urls#issues-and-pull-requests)
  - Mastering GitHub Issues (https://guides.github.com/features/issues/)
  - Closing issues via Pull Requests (https://github.blog/2013-05-14-closing-issues-via-pull-requests/)
- Azure DevOps
  - Create an Azure DevOps organization (https://docs.microsoft.com/azure/devops/user-guide/sign-up-invite-teammates)
  - What is Azure Boards? (https://docs.microsoft.com/azure/devops/boards/get-started/what-is-azure-boards)
  - Integrating Azure Boards and GitHub (https://docs.microsoft.com/azure/devops/boards/github/?view=azure-devops)
  - Improving code quality with Azure Repos branch policies (https://docs.microsoft.com/azure/devops/repos/git/branch-policies)
  - Link work items to support traceability and manage dependencies (https://docs.microsoft.com/azure/devops/boards/queries/link-work-items-support-traceability)
  - Link work items to GitHub Commits and Pull Requests (https://docs.microsoft.com/azure/devops/boards/github/link-to-from-github?view=azure-devops)

Note: You may have to review the permissions of each member of your team at the organization and project level (https://docs.microsoft.com/azure/devops/organizations/security/index?view=azure-devops).