

Section 1: Process Flows using two or more visualizations/diagrams

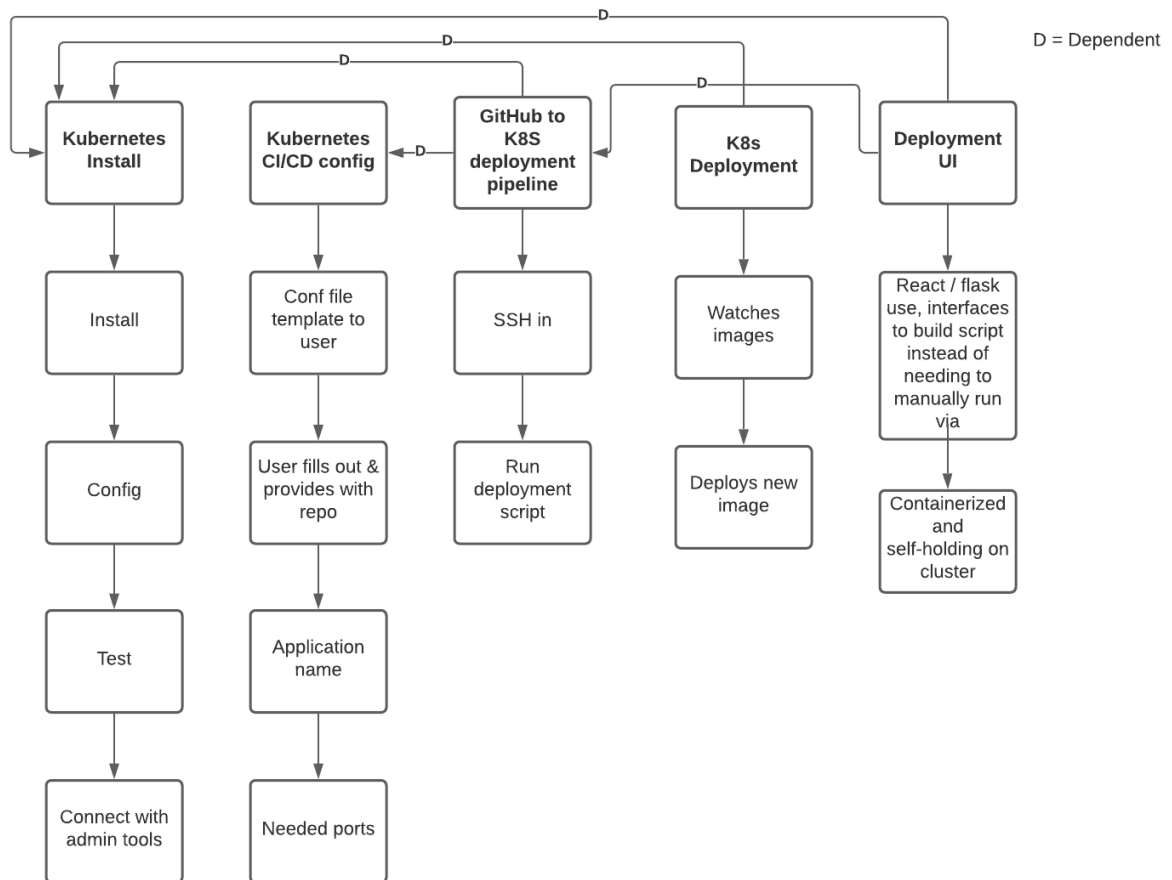


Figure 1: High level dependency chart of major tasks

The chart above is a dependency chart of our major tasks and their individual steps. The majority of the tasks are dependent upon the first major task, Kubernetes Installation, being complete. There are also several other dependencies, such as the GitHub to Kubernetes deployment pipeline being dependent on the Kubernetes CI/CD config, that need to be kept in mind as the team proceeds through each sprint.

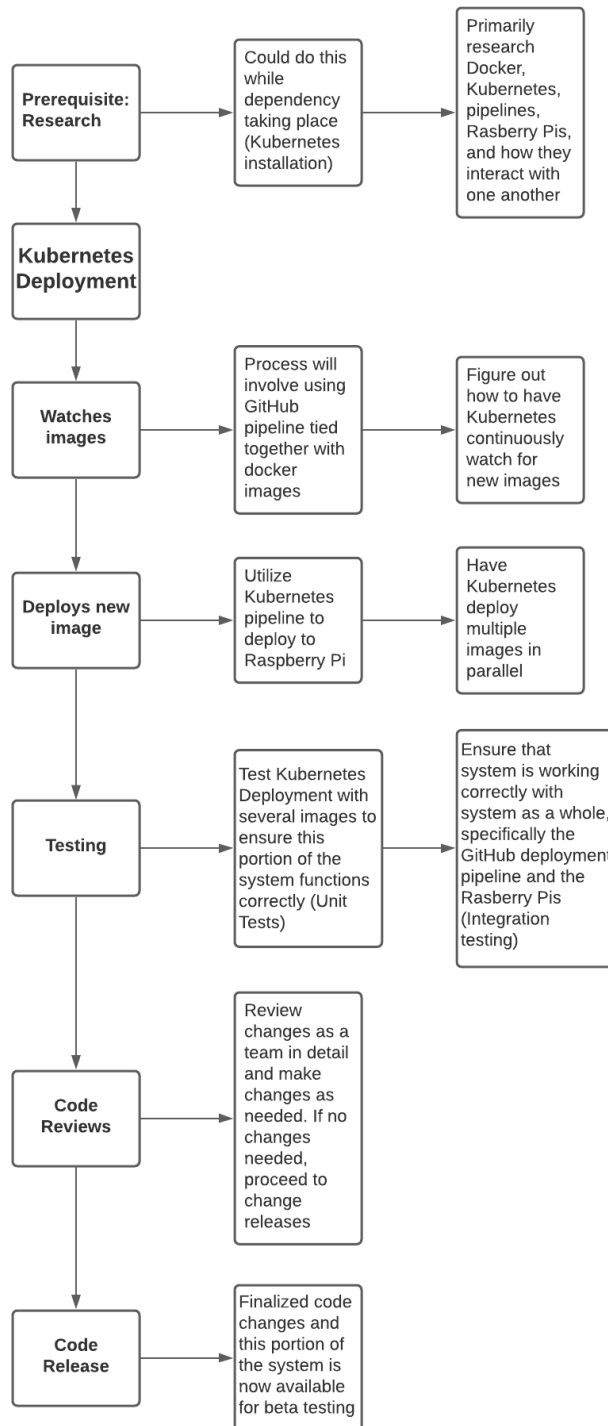


Figure 2: Individual project deliverables

The image above is a graphic outlining individual project deliverables. Arrows that point down describe the order in which the tasks are to be completed, whereas arrows to the right go into more detail as to what the individual task is.

Section 2: User Stories

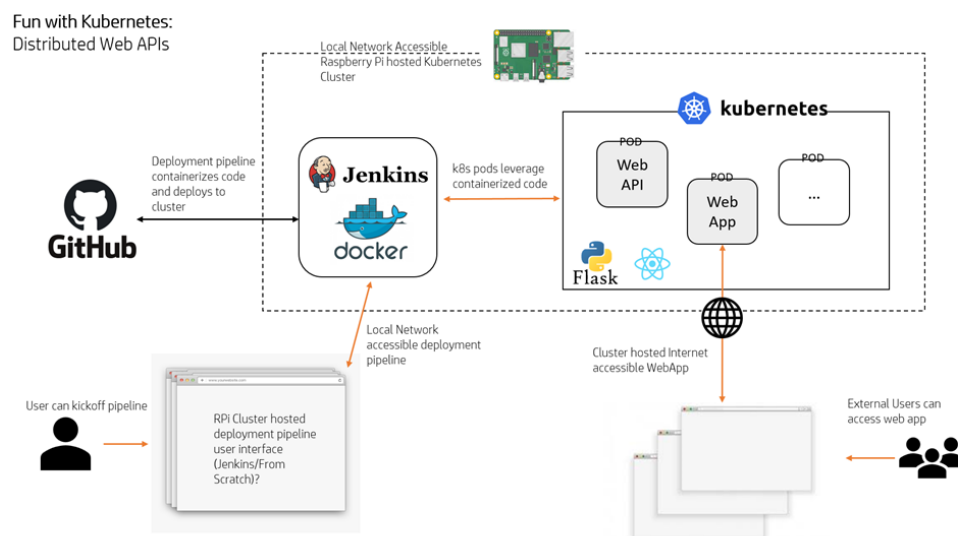
- User Story 1:
As a user, one needs to be able to go into the project dashboard and submit a config file to the pipeline so one can customize how the application works to tailor it to my use case.
- User Story 2:
As a user, one needs to be able to see the status of the project being uploaded to the pipeline so one can handle potential errors accordingly.
- User Story 3:
As a user, one needs to be able to track metrics, such as API requests or daily hits, through a user-friendly dashboard in order to analyze the application's performance.
- User Story 4:
As a user, one needs to be able to run my application locally through the Raspberry Pi network in order to debug and test the application before publicly deploying it.
- User Story 5:
As a customer, one needs to be able to access the deployed application from my personal computer in order to utilize the application deployed from the Raspberry Pi network.

Section 3: Personal iteration plan and estimations

Based on team discussions, implementation of these user stories (Figure 1) will begin Winter 2022. Many of the major tasks are dependent on research of topics such as Kubernetes, Docker, and deployment pipelines, so the team will use the remainder of the Fall 2021 term to focus on becoming familiar with these technologies. Several of the user stories also have dependencies. For example, both User Stories 4 and 5 are

dependent on the entire application pipeline being set up. If a team member is blocked on a current User Story, one should plan on assisting their teammate on the blocker rather than attempting to switch to another User Story, unless it proves to be inefficient for more than one teammate to work on that task. For the individual project deliverables (Figure 2), implementation will also begin Winter 2022. This is because these tasks are dependent on the Kubernetes Installation (Figure 1), which is expected to be completed by the end of Fall 2021. While Kubernetes is being installed, the prerequisite research needed for the Kubernetes Deployment can be completed (Figure 2).

Section 4: Solution Architecture



The architecture was designed by the project partner, Kyle Prouty. The design is intended to be a framework built on Raspberry Pis that developers can use to test their application on before deploying it onto their primary platform. The framework is versatile because of the deployment pipeline, which allows users to customize how the framework reacts if their code is pushed to GitHub, if tests fail, or other unexpected behavior occurs. Although Kyle designed this project to make local testing for developers easier, he also kept in mind that our team is comprised of students and chose relevant technologies for us to research and get experience with.