

Lab: Sprint 3 Planning

Estimated time needed: 30 minutes

Welcome to the hands-on lab **Sprint 3 Planning**. With your first two sprints complete, it is now time to plan for Sprint 3. In this lab, you will create a Docker image of your microservice, deploy it to a Kubernetes/OpenShift cluster manually, and set up a continuous deployment (CD) pipeline to automate that deployment using Tekton.

Objectives

In this lab, you will:

- Create stories for Sprint 3
- Add the stories to the kanban board
- Add labels and estimates to make them sprint ready
- Assemble a sprint backlog

Note on Screenshots

Throughout this lab, you will be prompted to take screenshots and save them on your device. These will be required either to answer graded quiz questions or for submission under **Option 1: AI-Graded Submission and Evaluation** or **Option 2: Peer-Graded Submission and Evaluation** at the end of this course. Your screenshot must have either the .jpeg or .png extension.

To take screenshots, you can use various free screen-capture tools or your operating system's shortcut keys. For example:

- **Mac:** You can use Shift + Command + 3 ($\hat{S} + \hat{C} + 3$) on your keyboard to capture your entire screen, or Shift + Command + 4 ($\hat{S} + \hat{C} + 4$) to capture a window or area. They will be saved as a file on your desktop.
- **Windows:** You can capture your active window by pressing Alt + Print Screen on your keyboard. This command copies an image of your active window to the clipboard. Next, open an image editor, paste the image from your clipboard to the image editor, and save the image.

Scenario

Management has been very pleased with the changes you have been making. It's now time to create a sprint plan to implement the last two stories in your Product Backlog, which are "Containerize your microservice using Docker" and "Deploy your Docker image to Kubernetes."

One more thing. There is a new requirement. You did such a great job automating the continuous integration (CI) pipeline with GitHub Actions that all of the developers seem much happier because of it. Management has decided that if a little automation is good, then more automation would be better. They would like you to automate the deployment to Kubernetes using Tekton once you have figured out how to do it manually.

To satisfy these requirements, you will need to move the two existing stories from the `Product Backlog` and add one new story to your `Sprint Backlog`.

- Containerize your microservice using Docker
- Deploy your Docker image to Kubernetes
- Create a CD pipeline to automate deployment to Kubernetes

Then, you will make a sprint plan from them.

Story 1: Containerize Your Microservice Using Docker

This story should already be in your `Product Backlog` from the original sprint planning session. It's a good idea to review and confirm that you have a "sprint ready" story.

1. Check that the "Containerize your microservice using Docker" story in the `Product Backlog` column of your Zenhub kanban board looks something like this:

Note: You can cut-and-paste the markdown text below into your story:

```
Title: Containerize your microservice using Docker
**As a** developer
**I need** to containerize my microservice using Docker
**So that** I can deploy it easily with all of its dependencies
### Assumptions
* Create a `Dockerfile` for repeatable builds
* Use a `Python:3.9-slim` image as the base
* It must install all of the Python requirements
* It should not run as `root`
* It should use the `gunicorn` wsgi server as an entry point
### Acceptance Criteria
Given the Docker image named accounts has been created
When I use `docker run accounts`
Then I should see the accounts service running in Docker
```

2. Since this story adds no visible stakeholder value and is primarily to make development more efficient, assign this story a label of `technical debt`.
3. Give the story an estimate. Remember the values are Small = 3, Medium = 5, Large = 8, and Extra Large = 13.
4. Next, assign the story to Sprint 3.
5. Finally, move the story to the top of `Sprint Backlog`.

Story 2: Deploy Your Docker Image to Kubernetes

This story should already be in your Product Backlog from the original sprint planning session. It's a good idea to review and confirm that you have a "sprint ready" story.

1. Check that the "Deploy your Docker image to Kubernetes" story in the Product Backlog column of your Zenhub kanban board looks something like this:

Note: You can cut-and-paste the markdown text below into your story:

```
Title: Deploy your Docker image to Kubernetes
**As a** service provider
**I need** my service to run on Kubernetes
**So that** I can easily scale and manage the service
### Assumptions
* Kubernetes manifests will be created in yaml format
* These manifests could be useful to create a CD pipeline
* The actual deployment will be to OpenShift
### Acceptance Criteria
Given the Kubernetes manifests have been created
When I use the oc command to apply the manifests
Then the service should be deployed and run in Kubernetes
```

2. This story will make the service scalable and perform better and thus give the users a better experience. It could be seen as an enhancement. Assign the appropriate label to this story.

3. Give the story an estimate. Remember the values are Small = 3, Medium = 5, Large = 8, and Extra Large = 13.

4. Next, assign the story to Sprint 3.

5. Finally, move the story to Sprint Backlog, ranking it second.

Story 3: Create a CD Pipeline to Automate Deployment to Kubernetes

In this story, you will create a CD pipeline using Tekton to deploy to OpenShift. It will reuse the Kubernetes manifest that you created in the "Deploy to Kubernetes" story. This is why you didn't simply use the command line interface to deploy the service in that story.

1. Create the following story about creating a CD pipeline in the Product Backlog column of your Zenhub kanban board:

Note: You can cut-and-paste the markdown text below into your story:

```
Title: Create a CD pipeline to automate deployment to Kubernetes
**As a** developer
**I need** to create a CD pipeline to automate deployment to Kubernetes
**So that** the developers are not wasting their time doing it manually
### Assumptions
* Use Tekton to define the pipeline
* It should clone, lint, test, build, and deploy the service
* Deployment should be to OpenShift
* It can use a manual trigger for this MVP
### Acceptance Criteria
Given the CD pipeline has been created
When I trigger the pipeline run
Then I should see the accounts service deployed to OpenShift
```

2. You could make the case that this story is an enhancement because deploys will be faster so customers get new features faster, or technical debt because it doesn't add enough customer value. It's up to you. Assign the appropriate label to this story.

3. Give the story an estimate. Remember the values are Small = 3, Medium = 5, Large = 8, and Extra Large = 13.

4. Next, assign the story to Sprint 3.

5. Finally, move the story to Sprint Backlog, ranking it third.

You are now ready to start the sprint.

Evidence

For both **Option 1: AI-Graded Submission and Evaluation** and **Option 2: Peer-Graded Submission and Evaluation**.

Take a screenshot of your Kanban board showing the three user stories added to Sprint 3 and save it as sprint3-plan.jpeg or sprint3-plan.png.

Conclusion

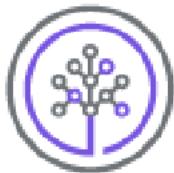
Congratulations! You have created your third sprint plan. You will use this in the next two labs to implement those three stories.

Next Steps

It's time to implement the first story in the sprint.

Author

[John J. Rofrano](#)



Skills Network