

# Capstone Overview

**Estimated time needed:** 5 minutes

In this Capstone project, you will apply many of the technologies and concepts you learned in the preceding courses to build and deliver a fully functional **Customer Accounts microservice**.

Starting in **Module 1**, you will begin this capstone journey by developing an Agile plan to build your RESTful microservice. You will create a GitHub repository and Kanban board to manage the project, build a user story template to write well-structured user stories, and populate your **Product Backlog** with all the stories needed to implement the Customer Accounts microservice. After completing your backlog, you will prepare a sprint plan by setting up sprints, estimating story points, assigning stories to the appropriate sprint, and building your **Sprint Backlog**. Throughout the Capstone, you will take screenshots and record GitHub URLs as evidence for final submission.

In **Module 2**, you will begin **Sprint 1** by configuring your project environment and developing the Customer Accounts microservice using **test-driven development (TDD)**. As you work on each story, you will move it across your Kanban board—from “Backlog” to “In Progress,” then to “Done,” and ultimately to “Closed.” You will create a development branch for your sprint work and push changes to GitHub by submitting pull requests. You will write test cases for the read, update, delete, and list functions for your RESTful Flask service and write just enough code to make each test pass. You will run **nosetests** to ensure all tests pass and use the coverage tool to maintain at least **95% test coverage**.

In **Module 3**, you will move into **Sprint 2**, where you will configure a **GitHub Actions continuous integration (CI) workflow** that runs automatically whenever a pull request or push is made to the main branch. As part of Sprint 2, you will build a workflow that performs linting with **Flake8**, runs your tests, checks code coverage, and validates code quality. You will then enhance your microservice by adding secure coding practices. This includes adding **Flask-Talisman** for security headers and **Flask-CORS** to establish Cross-Origin Resource Sharing policies. Following TDD, you will write failing tests, implement the required security features, ensure the tests pass, and merge your work into the main branch.

In **Module 4**, you will begin **Sprint 3**, where you will work on deployment-related user stories. You will create a **Dockerfile**, build a Docker image for the Customer Accounts service, and push the image to the IBM Cloud Container Registry. You will manually deploy your application to an OpenShift/Kubernetes cluster. You will also create a **PostgreSQL service** in OpenShift and write the Kubernetes deployment and service YAML manifests required to deploy your microservice. As with earlier sprints, you will commit, push, and merge your changes as you progress through each story.

In **Module 5**, you will extend your deployment work by creating a **Tekton continuous delivery (CD) pipeline** to automate the deployment of your microservice to Kubernetes. When triggered, this pipeline will clone your repository, lint and test your code, build your Docker image, and deploy the updated service to the cluster without manual intervention.

In **Module 6**, you will collect and organize all required evidence—screenshots, URLs, and outputs—from the hands-on labs for submission. Your work will be evaluated using **Option 1: AI-Graded Submission and Evaluation** or **Option 2: Peer-Graded Submission and Evaluation**.

Finally, in **Module 7**, you will complete the **Final Exam** to validate your understanding of the DevOps concepts and practices applied throughout the Capstone project.

Good Luck!



# Skills Network