Group Assignment: Cloud-Based Software Development and DevOps on Azure

> IE University - BCSAI - SDDO - 2025

Overview

Each group (up to 6 students) will plan, build, and deploy a small but functional cloud-native application on **Microsoft Azure** over **1-week Scrum sprints**. The goal is to simulate an end-to-end software and DevOps lifecycle: planning, coding, testing, deploying, and monitoring in a collaborative environment.

Duration

You will be conducting a demo & mini Sprint Review on 2025-12-04, which marks the deadline.

Team Structure

- 1 × Product Owner (rotating)
- 1 × Scrum Master (rotating)
- up to 6 × Developers/DevOps Engineers

Teams self-organize and assign responsibilities.

Note: you will have to play the role of stakeholders as well as developers during Sprint Reviews

Deliverable

A fully functional **minimum viable product (MVP)** hosted on Azure with automated CI/CD, basic monitoring, and documentation.

Minimum means minimum, **viable** means working. Prioritize building something simple that works over something complex that doesn't. If you're done by the end of week one, you can add more features or improve your code, but keep your MVP working.

Individually, each team member must **upload a document to Blackboard** stating the perceived **contributions of each team member** including themselves. Add relevant **links to the project** and **acknowledgment of relevant AI usage**.

Project Goal

Develop a **web-based application** deployed on Azure using modern DevOps practices.

DO NOT ITERATE UPON ONE OF YOUR INDIVIDUAL ASSIGNMENTS HERE, BUILD SOMETHING NEW TOGETHER

Core Requirements

1. Cloud Infrastructure

- Create a **resource group** and organize all project assets.
- Use at least **3 Azure services**, such as:
  - Azure App Service (web app hosting)
  - Azure SQL Database or Cosmos DB
  - Azure Storage (Blob/File)
  - Azure DevOps for pipelines and backlog management
  - Application Insights or Azure Monitor

2. Development and Functionality

- Develop a small REST or full-stack application.
- Backend: Python (Flask/FastAPI), Node.js (Express).
- Frontend optional or minimal (basic HTML/React).
- Testing

3. DevOps Pipeline

- Implement **CI/CD** with automatic build, test, and deployment.
- Version control using **Git** (Azure Repos).
- Define pipeline stages: *build → test → deploy → monitor*.

4. Monitoring, Logging, and Reliability

- Integrate **Application Insights** or equivalent monitoring.
- Create dashboards for key metrics (uptime, response time, error rate).

5. Documentation and Process

- README with architecture diagram, setup steps, and usage.
- Product Backlog, Kanban Boards, etc.
- Documents resulting of following the Scrum framework (Sprint backlog snapshots, Review and Retrospective outcomes and agreements, Definition of Done, ...)
- Retrospective summary.

Scrum Structure Example

THIS IS AN EXAMPLE, ADJUST AS NEEDED
you don't plan 4 sprints ahead in scrum

Sprint 0 (Preparation)

- Form teams and define roles.
- Select project idea and draft product backlog.
- Create Azure subscription access and initial repo.

Sprint 1

- Define MVP scope and architecture (services, CI/CD design).
- Set up Azure environment and repository structure.
- Deliver initial working skeleton app (Hello World deployment).

Sprint 2

- Implement core features.
- Add automated tests.
- Configure full CI/CD pipeline.

Sprint 3

- Add database integration, logging, and monitoring.
- Conduct code reviews and improve deployment automation.

Sprint 4

- Polish UI or API.
- Deliver documentation and retrospective.
- Demo & mini Sprint Review

Evaluation (100 pts)

Category	Points
Development and Functionality	25
Cloud Infrastructure	20
DevOps Pipeline 20	
Testing and Code Quality	15
Monitoring, Logging, and Reliability	10

Category	Points
Documentation and Process	10

Optional Extensions

- Use Infrastructure as Code (IaC) with **Bicep**, **Terraform**, or **ARM** templates.
- Add containerization with **Docker** or **Azure Container Apps**.
- Enable blue-green or rolling deployment strategies.