

BAH MSD Project Objectives

Day 1

1. Understanding the business requirements and technical requirements
2. Understand the project environment tools and software requirements and use
3. Create an application design that aligns with fundamentals of DevOps and Microservices
4. List the components of the application, explain the significance of each component
5. Specify how the components will interact with each other
6. Create a workplan and backlog of tasks and document it
7. Create a local Git repository on the project vm
8. Create a GitHub online account and grant access to your project partner
9. Link online GitHub repository to your local repository on project vm
10. Install and/or verify E-git eclipse plugins in the project vm

Day 2

1. Design a build process using Gradle that will be used to build your REST services
2. Develop a REST service for Customers data, use hard coded customers data
3. Use Postman to test Customers data service end points
4. Develop a REST service for Events data, use hard coded events data
5. Use Postman to test Events data service end points
6. Develop a REST service for Registrations data, use hard coded registrations data
7. Use Postman to test registrations data service end points
8. Test the services end points using front end client
9. Prepare to explain and demonstrate the service and build process
10. Commit the completed API code (services code) and gradle build files to your local repository

Day 3

1. Implement the Customer data API with full CRUD capabilities using Spring Boot
2. Create an embedded database to store Customer data
3. Code the API to use data from embedded database
4. Use the gradle build process to build and run Customer data API
5. Use Postman to test service endpoints and verify CRUO capabilities of the service
6. Repeat task 1 through 5 to implement Event data API
7. Repeat task 1 through 5 to implement Registrations data API
8. Demonstrate the service and build process
9. Test the service end points using front end client
10. Commit the completed API code and gradle files to your local repository

Day 4

1. Create a new local Git repository for Security API
2. Implement the Security API using Spring Boot and JSON web token (JWT) library
3. Use the gradle build process to build and run Security API
4. Use Postman to test security service endpoints
5. Integrate Security API and Data API and test integration
6. Use the token generated by Security API to access Data API
7. Verify the integration between the two API using postman
8. Verify the integration between the two API using front end client
9. Demonstrate the integration and explain the flow to Instructor
10. Commit the completed API code and gradle files to your local repository

Day 5

1. Create a Docker image for Data API using a Dockerfile
2. Create a Docker image for Security API using a Dockerfile
3. Run the docker container for Data API and test its end points using postman
4. Run the docker container for Security API and test its end points using postman
5. Verify the API functions correctly with docker containerized apps
6. Create Kubernetes deployments for Authentication API and Data API docker images
7. Use Kubernetes commands to expose Authentication API and Data API services
8. Update front end client proxy configuration with exposed services addresses
9. Deploy React client application and test the application
10. Demonstrate the scalability and auto-failover capabilities of Kubernetes cluster

Day 6

1. Verify Kubernetes deployments, pods and services
2. Verify services instances logging
3. Demonstrate the deployments, logging to Instructor
4. Design a strategy to send logging data to a centralized logging system
5. Implement centralized logging capabilities
6. Set up API to generate monitoring metrics using Prometheus
7. Set up API to trace requests using Jaeger
8. Demonstrate Prometheus metrics to Instructor
9. Demonstrate trace requests to Instructor
10. Demonstrate centralized logging to Instructor

Day 7

1. Set up and configure Jenkins server on project VM
2. Start the Jenkins server and verify it is up and running
3. Create a sample Jenkins job and test it
4. Create a CI pipeline to automate API server build process
5. Test the pipeline and make sure it is running smoothly
6. Demonstrate the CI pipeline working to Instructor
7. Add to the pipeline with ability to create a Docker image for the API
8. Extend the pipeline with a testing environment and complete CD pipeline
9. Complete CD pipeline with integration and testing
10. Demonstrate the CI/CD pipeline to Instructor

Day 8

1. Make sure SecurityAPI is running
2. Test SecurityAPI using Postman
3. Make sure DataAPI is running
4. Test DataAPI using Postman
5. Make sure Operations on Docker containers for both API are running
6. Make sure operations on Pods are running
7. Make Sure Load balancing works
8. Explain the project components and answer questions
9. Work with team mate to Prepare Presentations
10. Deliver a great team presentation