

Assignment 4

Suggest best practices for a CI/CD system for a micro-services-based application.

CI/CD process stands for Continuous Integration and Continuous Delivery. Continuous integration refers to code changes that are done and later combined with the main branch of already existing code. Microservice architecture – a variant of the service-oriented architecture structural style – arranges an application as a collection of loosely coupled services. This is in opposition to the previously popular monolith architecture in which a logically modular, component-based structure application is packaged and deployed as a monolith.

Some of the best practices suggested for a microservices based application are:

1. **Frequent software release:** Software release frequency can be increased if it is in a release ready state and has been tested vigorously under Alpha/Beta testing and similar.
2. **Less branching with daily commits:** this helps developers to spend more time on development rather than version controlling. The ideal use of GitOps would be to push commits directly to the main branch from the local branches daily (if possible). This helps developers to handle small sized code to integrate instead of big sized code.
3. **Data partitioning:** If each service is based on its own database then database schema changes and deployment becomes the responsibility of that service installation script. For the CI/CD pipeline, it means we need to be able to support multiple database deployment in our test and staging environment provisioning cycles.
4. **Readiness to apply microservices:** Microservices architecture is the best way forward to manage DevOps. Although re-designing the existing application can be time consuming and intimidating. What would help is to have an incremental approach where we are aware of

our mission and the new architecture is built around that so that we can gradually replace the old system with the new one.

5. **Security:** CI/CD system offers retrieval of the codebase and credentials to install in all sorts of development environments. Developers must consider separating their CI/CD systems and locating them at reliable and secure internal network. Consider putting strong two-factor authentications and access management systems to help restrict exposure to any external or internal threats.
6. **Polyglossia:** It is desirable to base the CI/CD flow for each service on the same template which includes the same building blocks and a well defined interface, i.e. each service should provide similar 'build' , 'test', 'deploy' and 'promote' endpoints for integration into the CI system. Additionally the interface for querying service interdependency should be clearly defined. This will allow for almost instant CI/CD flow creation for each new service and will reduce the load on the CI/CD team. Ultimately this should allow developers to plug-n-play new services into the CI/CD system in a fully self-service mode.