Development Workflow [Performed by an engineer]



Use Test-driven development



Commit changes to a local task branch



Local git hooks are executed



Push commit build workflow to a remote is triggered task branch on the task branch



Perform code review



Merge changes build workflow to the master branch is triggered on the master branch



Tag commit on master [timestamp-env] (1)



Deployment workflow is triggered to the [env] environment

Build Workflow [Triggered by pushing a commit to any branch - input: git commit hash]



Run static analysis



Run unit test



Build artefact



Prepare data



Run integration test



Run contract test



Run fitness function



Publish artefact [timestamp-hash]

Deployment workflo

Deployment workflow is triggered (2)

Deployment Workflow [Triggered by the Build Workflow or by tagging a commit on master branch] (3)



Backup data



Provision infrastructure



Deploy artefact (5)



Prepare data



Run smoke test



Send notification



Collect feedback



Run fitness function

Test Workflow [Triggered on demand or by time] (4)



Provision infrastructure



Deploy artefact (5)



Prepare data



Run smoke test



Run end-to-end test



Run performance test



Run security test



Run fitness function

Clean-up Workflow [Triggered on demand or by time]



Remove environment



Remove old backups



Remove old

Continuous Integration Pipeline Diagram

Last modified: 2020/10/05 19:41:24

Notes

- There is a fixed and known list of environments marked either as to be deployed on-commit to or on-demand
- 2. The Build Workflow triggers the deployment automatically to an environment that belongs to a task branch or is on the list of known environments but not marked for on-demand deployment. Environment creation for a task branch should be manual but once it is there the Deployment Workflow will be triggered automatically
- 3. The Deployment Workflow should create environment from scratch and set it to an initial state if it does not exist already for a task or any other branch. A subsequent deployment will be just an update till that environment is removed
- 4. Any test in the Test Workflow can be opted out if run manually or be implemented as a separate or independent pipeline execution.
- 5. Always the latest artefact for a commit is going to be deployed in case of multiple artefacts being produced for this single commit

Assumptions

- Sensitive information is held in a secret store, e.g. passwords and encryption certificates
- Product team is notified on any failure of the pipeline
- Each environment has a time-to-live information set and after its expiration all the allocated resources should be freed with an exception to a production environment
- All the significant pipeline events are recorded, e.g. deployment

Questions

- Should the Deployment Rollback Workflow be implemented as another execution of the Deployment Workflow as a result of "reverting" the last commit(s)?
- Should the artefact deployment strategy be in scope for the consideration of the CI pipeline to enable e.g. A/B testing by switching on/off features and user journeys?
- Do we need to extract the process of preparation of the test data from the main pipeline?
- What are good candidates for fitness functions?