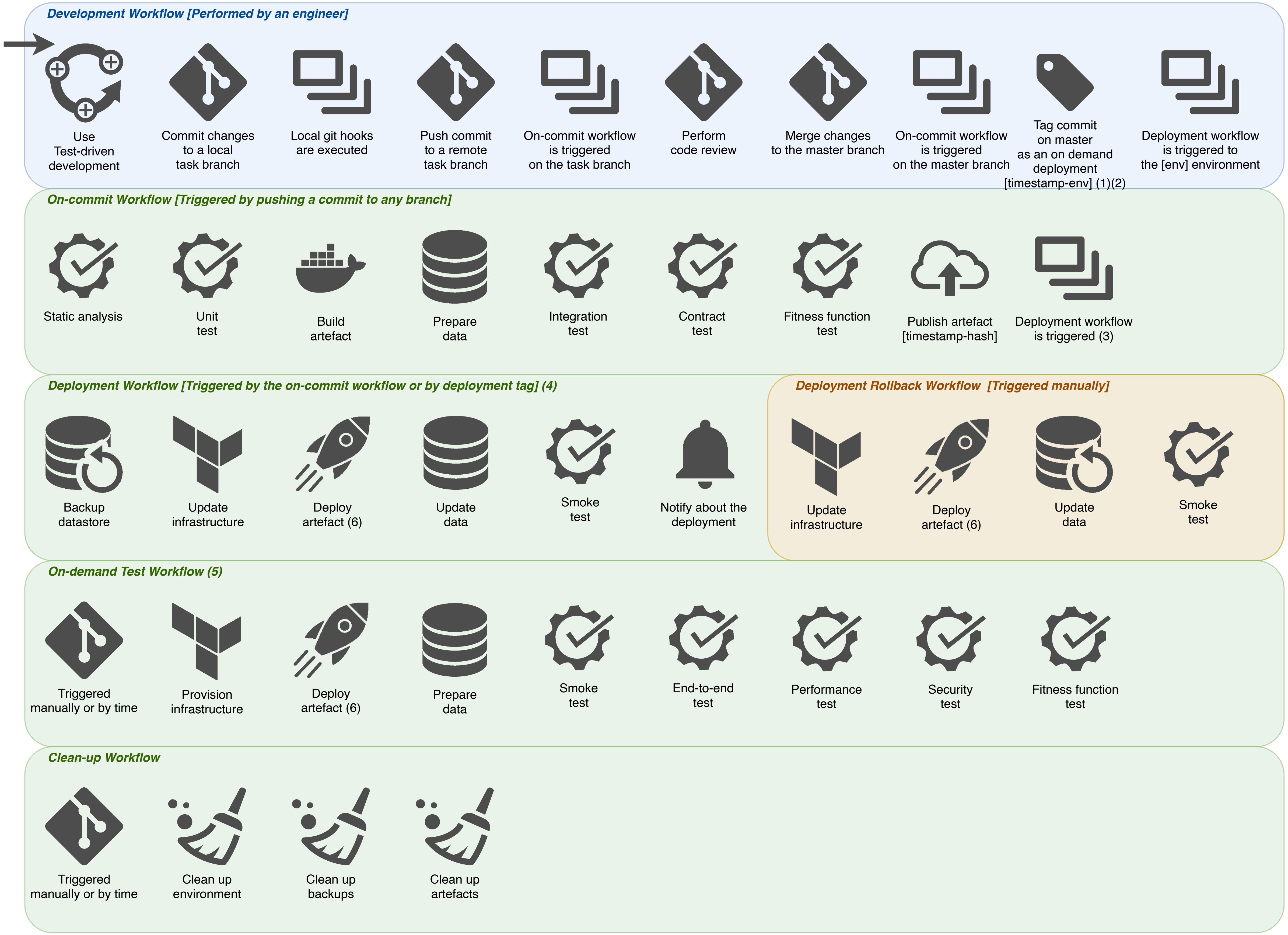

Development Workflow [Performed by an engineer]

- Use Test-driven development
- Commit changes to a local task branch
- Local git hooks are executed
- Push commit to a remote task branch
- On-commit workflow is triggered on the task branch
- Perform code review
- Merge changes to the master branch
- On-commit workflow is triggered on the master branch
- Tag commit on master as an on demand deployment [timestamp-env] (1)(2)
- Deployment workflow is triggered to the [env] environment

On-commit Workflow [Triggered by pushing a commit to any branch]

- Static analysis
- Unit test
- Build artefact
- Prepare data
- Integration test
- Contract test
- Fitness function test
- Publish artefact [timestamp-hash]
- Deployment workflow is triggered (3)

Deployment Workflow [Triggered by the on-commit workflow or by deployment tag] (4)

- Backup datastore
- Update infrastructure
- Deploy artefact (6)
- Update data
- Smoke test
- Notify about the deployment

Deployment Rollback Workflow [Triggered manually]

- Update infrastructure
- Deploy artefact (6)
- Update data
- Smoke test

On-demand Test Workflow (5)

- Triggered manually or by time
- Provision infrastructure
- Deploy artefact (6)
- Prepare data
- Smoke test
- End-to-end test
- Performance test
- Security test
- Fitness function test

Clean-up Workflow

- Triggered manually or by time
- Clean up environment
- Clean up backups
- Clean up artefacts

Notes

(1) There is a fixed and known list of environments marked either as to be deployed automatically to or on demand

(2) Only one tag per environment might be considered as these tags are used as helpers only

(3) The On-commit Workflow should trigger deployment 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 a deployment workflow should be triggered automatically

(4) 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

(5) Any test in the On-demand Test Workflow can be opted out if run manually

(6) 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

Continuous Integration Diagram

Last modified: 2020/10/04 15:17:47