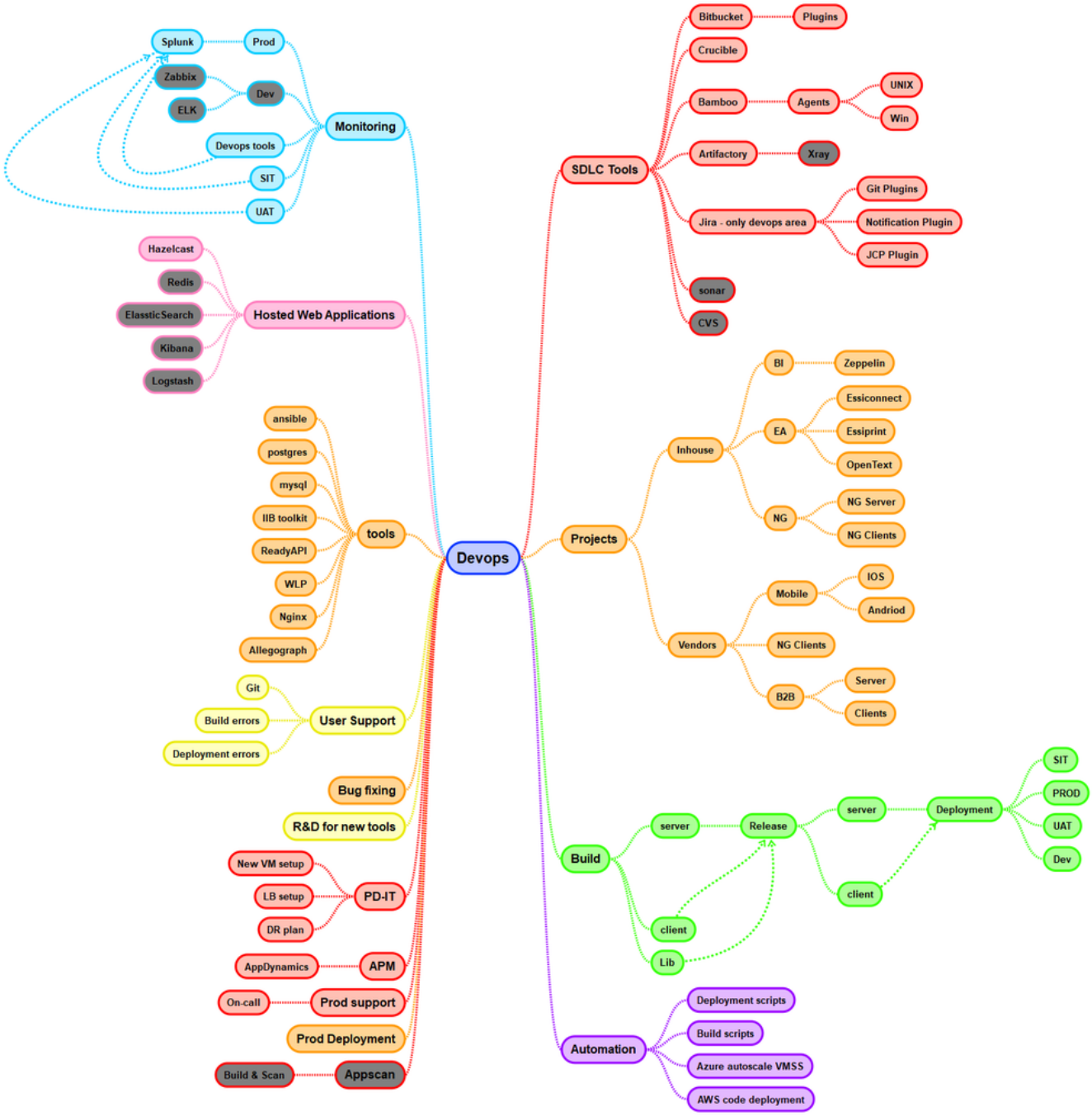
DevOps Tools

Overview

DevOps is a set of practices that automates the processes between software development and IT teams, in order that they can build, test, and release software faster and more reliably. The concept of DevOps is founded on building a culture of collaboration between teams that historically functioned in relative siloes. The promised benefits include increased trust, faster software releases, ability to solve critical issues quickly, and better manage unplanned work.

Tools we have

Devops Lanscape:



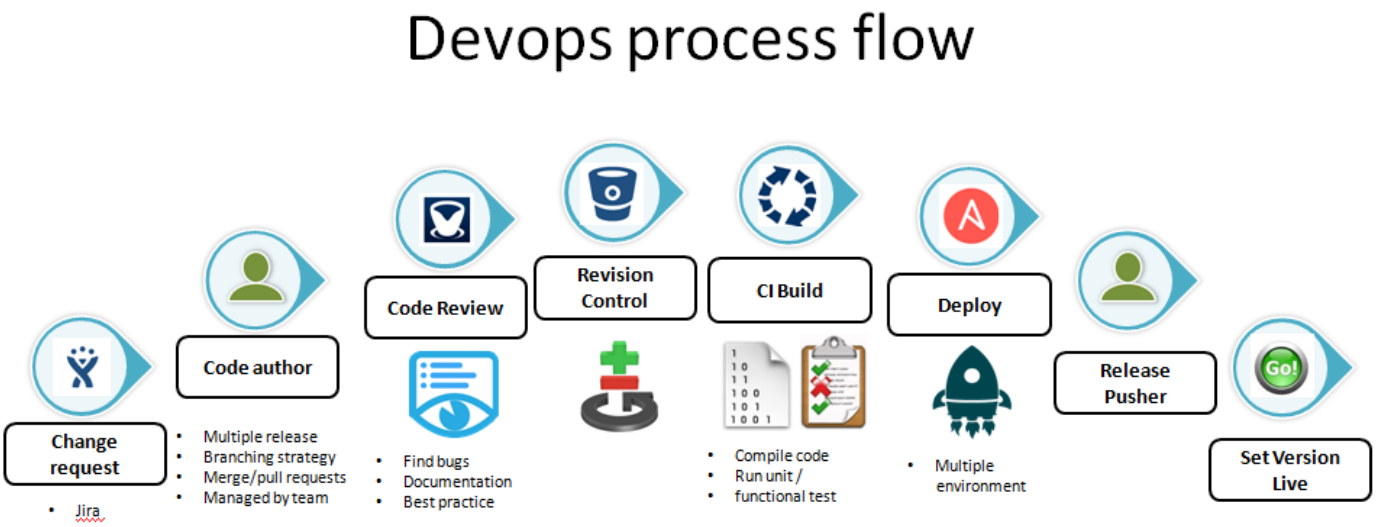
Devops process:

This process is designed to suit NG team needs, resources and support:

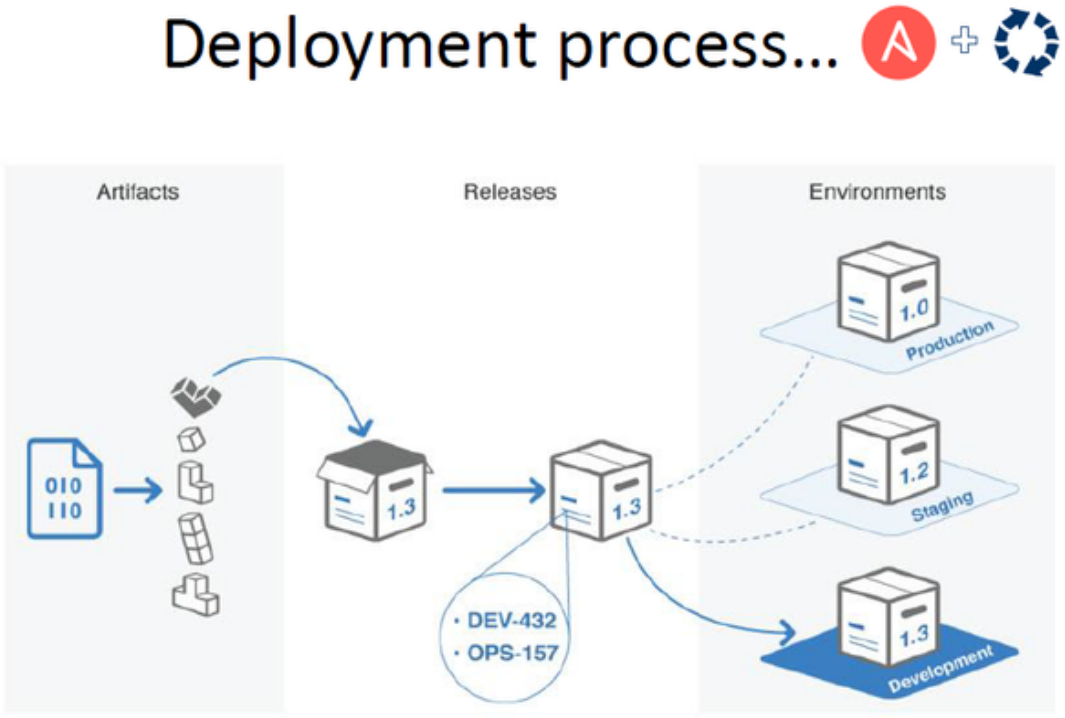
1. Lean and Agile team
2. Deploy applications in multiple environments
3. Self service model with role based access to build and deploy projects to multiple environments
4. Automation in Build, test and deployment to all environments
5. Reliable and consistent builds from development to production
6. Manage public and internal build dependencies across projects
7. Schematic versioning for all artifacts
8. Auditing and tractability from code change to deployment
9. Change control over source code commits
10. Rollback option to previous release in deployment environment
11. Increase visibility among team members and allow them to have a sense of purpose in their day-to-day work
12. Effective communication and collaboration with the teams to reduce bottlenecks in build, test and deployment process.

13. Support internal teams(DEV, QA, RELEASE, DEVOPS), remote teams and vendors.

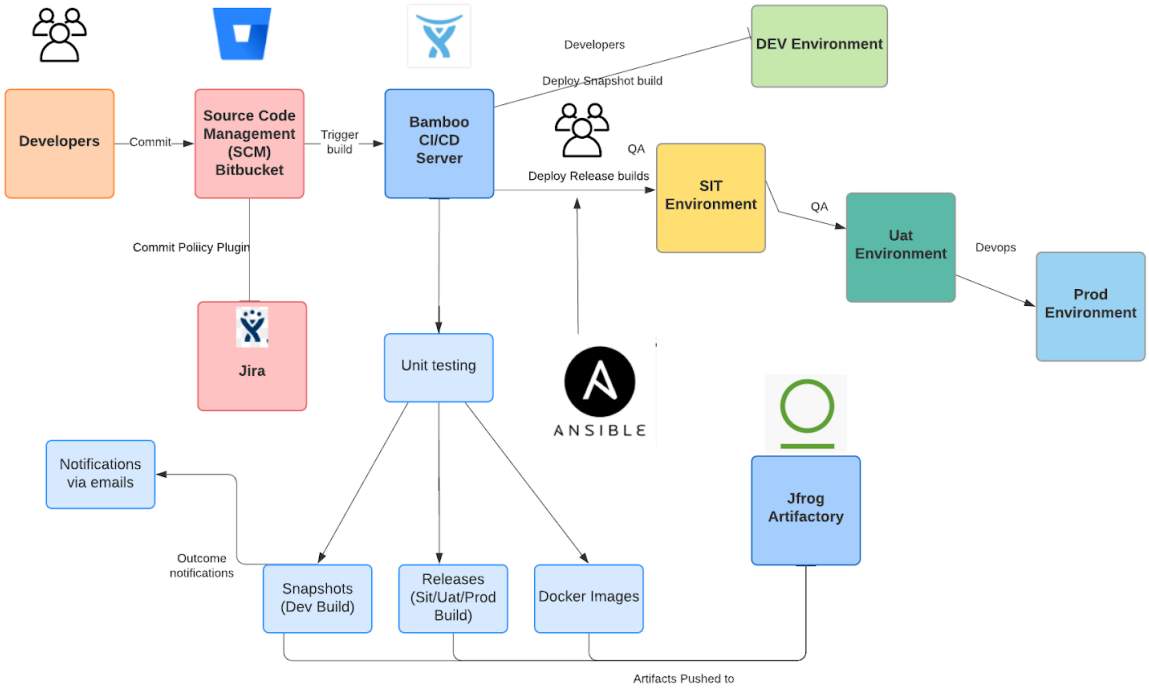
We need to improve the process thru continuous feedback from the teams.



Deployment process:

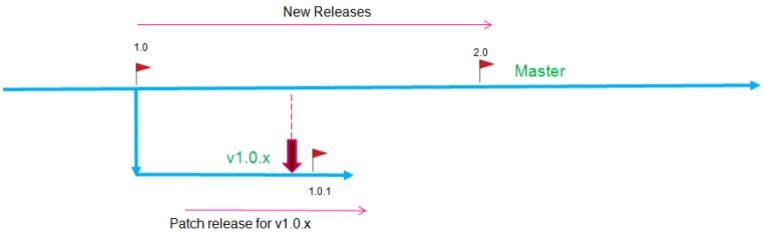


CI / CD Pipeline:



Branching strategy:

Trunk based development and branch for patch release



Master is the primary branch of a code line that evolves forever. Master branch provides a destination for all changes for **current sprint release**, patch fixes and represents the linear evolution of a project. Hot fix / Patch Release code lines are branched from the master branch, and work that occurs in branches is propagated back to the master or vice-versa using cherry pick method.

Jira commit policies:

Jira ticket is mandatory for every commit. When code is committed (master branch), Bitbucket will check the validity of the commit by verifying

1. Jira ticket is in working state
2. Jira ticket is valid for current sprint
3. If its a hot fix, the jira ticket has patch label (hot fix branch)

There are other policies for developer branch , feature branch etc.,

Builds:

SNAPSHOT: Build / Artifact that is under development. Continuous integration builds produce SNAPSHOT artifacts and its deployed to DEV env by developers for verification. Any artifact that has SNAPSHOT is un-tested code and should not be used to create RELEASE artifacts.

RELEASE: Final build / artifact that is created after code cut off or a hot fix change, It has unique version (including all dependencies) and does not change in the life cycle of application. Baseline is created after release build to support hot fix.

There are other build setup in BAMBOO like **Developer private branch** builds, **Nightly** builds, **Functional test** builds, **Sourcescan** builds etc., to support the team and project requirements.

Deployment:

Ansible is used as configuration management tool to deploy releases from Bamboo. Ansible playbook scripts are written by DEVOPS team.

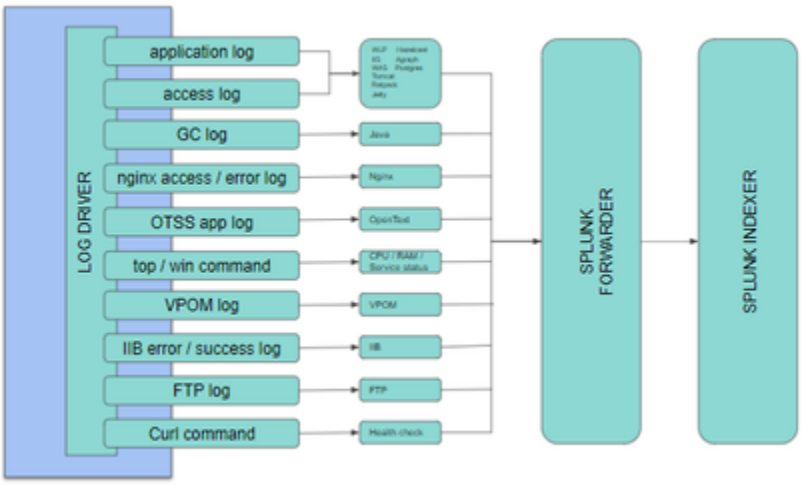
SNAPSHOT artifacts are deployed to DEV ENV

RELEASE artifacts are deployed to SIT env and promoted from SIT UAT PROD env

Release notes are prepared and verified using change log recorded in Bamboo for each release.

Deployment to PROD env is approved in Bamboo by NG release team and deployed by DEVOPS team.

Application monitoring and log aggregation: splunk



NEW TOOLS (R & D)

- Docker - App NG apps are containerized and used for functional testing
- Kubernetes - NG apps running in LAB K8S env
- Hashicorp Vault - Key store (WIP)

- Spring boot cloud - WIP
- Terraform - WIP