

Jenkins pipelines

4 jenkins

1. ci - jenkins git - source to target
 - a. which profile to run from naven for the build, which tests to run, steps skipped based on the DB.
 - b. spot bugs - quality assurance, more for the dev
 - c. junits - errartic tests can be skipped (can be html reports - artifacts)
2. cd - jenkins srv -
 - a. production build - several versions == several branching
 - i. giddy = master + supported versions
 - ii. all are running the same pipelines, to seoearte master from versions and make it simpler for the release
 - b. non production build - for devs, same as a, but without the licensing (3 licensing servers) that are signed + minor changes - only for tests
 - c. saas
 - i. AZURE - make sure
 - d. appliance - the actual image, web admin and such
 - i. iso in azure and aws →
 - ii. appliance_ubuntu job - create the appliance
 - iii. web admin - port 444 - user=skyboxview password=skyboxview/welcomet0Jamic@
 1. moved from centosS to ubuntu - redhat stopped updating centosS.
 - iv. application - port 8443 - user=skyboxview password=skyboxview (on perm, saas is differnet)
3. automation - less our jobs
4. ti - volunarbility dictionary - from older pipelines to newer ones - accompany

jenkins git - after soucre to target approved - trigger "jira pipeline"

Jira

1. fixed version - which versions are affected and needs to be fixed
2. affected version - which version are affected, what was already pushed, was pushed there by "jira_pipelines" + orit + release mangers → "commit\$merge" button in the Jira at the top right corner, will run the cherry pick 2 pipeline on the diff from 1 and 2.
3. -
4. fixed in build - A the end of a production build a git tag is added, so that each build has its own tag for the commit that was built by it. Diff between two build tags, for each of the issues we update the new build that fixed the issue
5. latest jenkins build - by QA request, the latest build that has it fixed. each version that has a README - from git the issues, cross referncned from Jira queries so that certin criteria cross refrenced with salrforce (CRM) to create =the readme,

Mend - code scanning - not yes on the appliance, as it is not a container yet - if a voln is discovered it creates a Jira,

Qualice - on appliance, it is important for the client to have no vulnerabilities, old and new to create a diff for the release notes. the base of the iso holds the latest versions, better to have

secret server - holds passwords

- qualice - general user , get username password from secret server

o

container images - nexus \ ecr for AWS \ same as azure

start_server_validation - master skybox - the actual product, appliance is mostly the infrastructure, but for the client to get an appliance it includes the product itself

console - **research** - holds key values

artemis - research

swagger - html files that a client can access, documentation not additional features

GIT -