CI/CD for OpenShift

WW Developer Advocate Team

IBM **Developer**

Contents

- Overview of CI/CD
- Architecture of CI/CD pipeline in OpenShift
- Jenkins in OpenShift
- Jenkins external to OpenShift
- CI/CD Pipeline Demo

Overview of CI/CD

Continuous Integration

- The process of having developers integrate code into a single code repository.

Continuous Delivery

- The process of automatically building and testing new versions of software so that it is ready to be deployed to production. The process of deployment is manual.

Continuous Deployment

- The process of automatically building, testing, and deploying new versions of software into production as soon as it passes all tests.

Tool stack for implementing CI/CD in OpenShift

Jenkins – Deployed in OpenShift as a service

SCM – With Jenkins SCM integration we can be notified of new commits for different branches. (Jenkins has plugins for **AccuRev, CVS, Subversion, Git, Mercurial, Perforce, Clearcase** and **RTC)**

Docker – Container images are built and published to Docker registry.

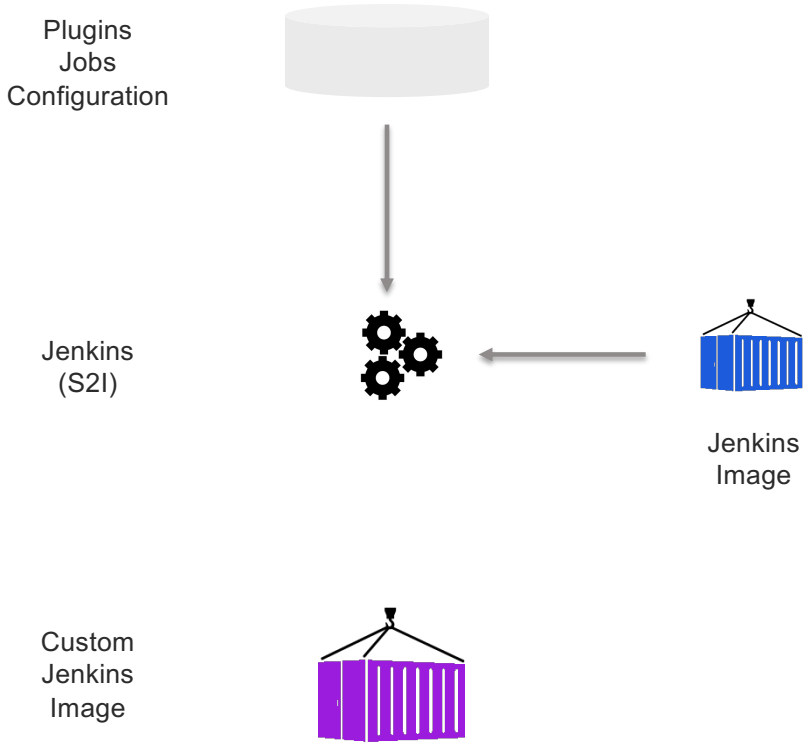
OpenShift – Jenkins will notify OpenShift of a new image which will begin a predefined deployment update strategy.


Jenkins

- Open source, server-based tool that builds and tests software continuously
  - Enables continuous integration and continuous delivery
  - Extendable via a rich set of plugins
    - Large ecosystem of existing plugins to integrate with on-prem and cloud based DevOps tooling
      - e.g. SCM integration, Kubernetes, Docker, Gerrit etc

Jenkins in OpenShift

- Certified Jenkins images with pre-configured plugins
- Jenkins S2I Builder for customizing the image
  - Add
    - Custom Jenkins Jobs definitions
    - Additional Plugins
    - Custom *config.xml* file
- OpenShift specific plugins to integrate authentication and CI/CD pipelines
- Dynamically deploy slave containers



Jenkins external to OpenShift

If a build server already exists outside of OpenShift, it is still possible to integrate and deploy to a cluster within OpenShift.

Here are 3 methods to accomplish this:

- Install **oc** on the external Jenkins server and use the **oc** mechanism to deploy apps to OpenShift
- Use REST APIs to call OpenShift
- Use the OpenShift Jenkins plugin to deploy apps to OpenShift

CI/CD Pipelines for Microservices

Pipeline as Code

- Treat pipeline as another piece of code. Commit Jenkinsfile to repo
- Will now have multiple repos and each may have different build/deploy processes

Automation of CI/CD pipeline is almost necessary to ensure smooth deployments of microservices

Zero downtime deployment will be handled by Kubernetes as a rolling update. Updates can also be rolled back to a previous revision.

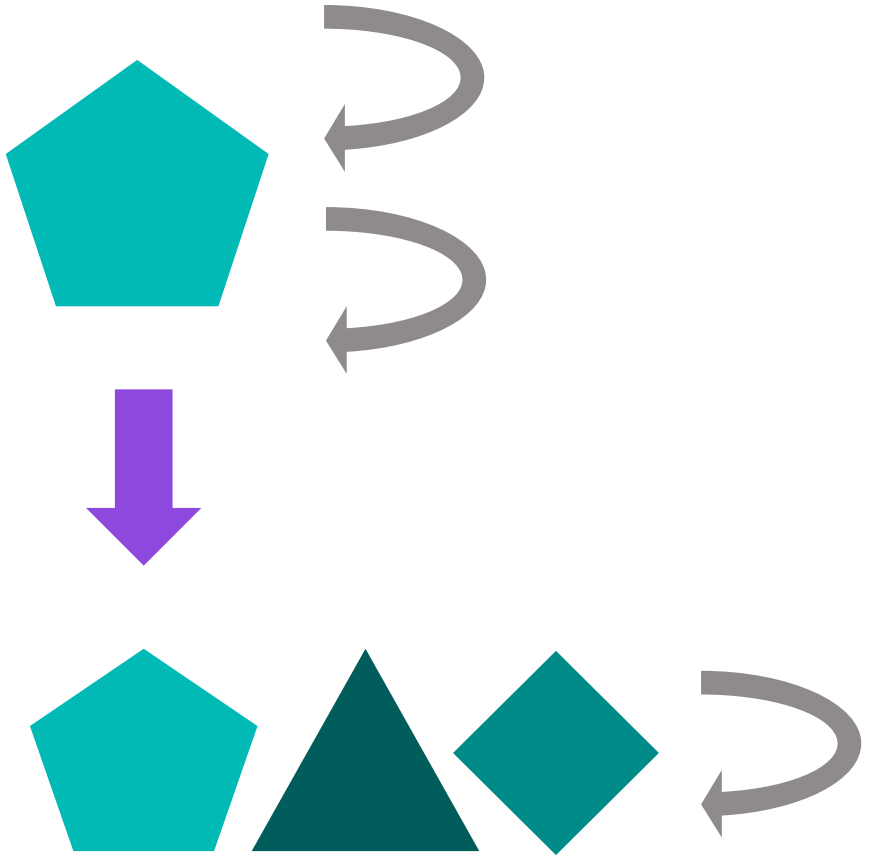
CI/CD Pipelines for Microservices

With microservices, each service can be tested separately.

- **Unit tests** can be done during the build stage with Maven along with a **static code analysis**

After publishing the microservice to the image registry, **integration testing** with the other components can begin.

- The newly published microservice is then tested against the latest stable versions of other components.



Reference links

- Microservice reference architecture documentation - https://www.ibm.com/cloud/garage/architectures/microservices/reference-architecture
- Microservice reference architecture application - https://github.com/ibm-cloud-architecture/refarch-cloudnative-kubernetes
- Pipeline as Code with Jenkins - https://jenkins.io/solutions/pipeline/

Further reading

- Azarny, Igor. “CI/CD for Containerized Microservices - DZone DevOps.” *Dzone.com*, 5 Oct. 2017, dzone.com/articles/cicd-for-containerised-microservices.
- Schenck, Don. “Get started with Jenkins CI/CD in Red Hat OpenShift 4” developers.redhat.com, 2 May 2019 https://developers.redhat.com/blog/2019/05/02/get-started-with-jenkins-ci-cd-in-red-hat-openshift-4/

Additional topics

- Istio – Allows for ease of management and additional functionality of the service mesh.
- External notifications? Jenkins has plugins for sending notifications through many different applications including Slack
- How does your application handle failures of individual services? This blog covers how to architect your microservice application to handle outages.
- Martin Fowler's blog on microservices has more information on microservice architecture