+DevOps culture enables development, IT operations, quality engineering and security teams to coordinate and collaborate to produce better, more reliable products. By adopting a DevOps culture along with DevOps practices and tools, teams gain the ability to better respond to customer needs, increases confidence in the applications they build and ship to achieve business goals faster.

Software release process (a journey from git commit to production, not the entire software developmentb process) focuses on building the source code, testing and ultimately shipping it out to production environment(s). Release process constitutes the following four stages:

1. **Source**
    a. Check-in source code such as .py, .java, .cs files
    b. Peer review new code
2. **Build**
    a. Compile code
    b. Run unit tests
    c. Style checkers
    d. Create function deployment packages and container images
3. **Test**
    a. Integration tests with other systems
    b. Load testing
    c. UI tests
    d. Security testing
4. **Production**
    a. Deployment to production environments
    b. Monitor code in production to quickly detect errors

Effects of CI/CD

- Deployment frequency became hourly/daily activities.
- Change lead time became one - seven days as opposed to months/quarters.
- Change failure rate: 0 - 15%

Pillars of releasing modern applications

1. Continuous integration goals
    a. Automatically kick-off a new release when new code is checked in
    b. Build and test code in a consistent, repeatable environment
    c. Continually have an artifact ready for deployment
    d. Continually close feedback loop when build fails
2. Continuous deployment goals
    a. Automatically deploy new changes to staging environments for testing
    b. Deploy to production safely without impacting customers
    c. Deliver to customers faster: Increase deployment frequency, and reduce change lead time and change failure rate

3. Infrastructure as Code (IaC) goals
    a. Provision and manage IT infrastructure using source code, essentially treating your servers, databases, networks, and other infrastructure like code
    b. Make your infrastructure deployments fast, repeatable, and consistent
    c. Ability to standup and terminate production environment in a staging environment to enable continuous testing.

[You are expected to read and understand the inserted hyperlinks pointed to different blog posts and user guides in this document]

Pre-requisites

1. DG provided Windows system
2. Install chocolatey package manager
3. Install Jenkins using choco package manager

Assignment

1. Login and configure Jenkins server on your system. Configure Docker for Jenkins pipeline.
2. Architect a simple calculator application that supports addition, subtraction and product of two positive integers.
3. Develop the application package with simple unit test cases in Python 3.x
4. Automate the infrastructure creation with AWS CloudFormation templates. (use SAM templates for AWS Lambda functions)
5. Add web hooks to trigger a Jenkins build every time changes are pushed to a Git branch on GitHub.
6. Trigger the Jenkins pipeline job that uses Docker agent to:
    a. CI: Run the unit test cases and build the package
    b. CD: Deploy the code package to AWS cloud.

Deliverables

1. A document summarizing steps that you had followed in completing the assignment (a runbook of sorts and/or an architecture diagram)
2. Demo a running pipeline (from git commit to deployment to AWS cloud) to the DevOps and CI/CD learning panel.

Learning Resources

- DevOps 101 - https://www.coursera.org/learn/uva-darden-continous-delivery-devops

- DevOps
  - Intro to DevOps - https://www.youtube.com/watch?v=_I94-tJlovg
  - https://www.toptal.com/devops/what-the-hell-is-devops
  - https://www.edureka.co/blog/category/devops/

- Jenkins
  - Jenkins Architecture - https://www.simplilearn.com/tutorials/jenkins-tutorial/what-is-jenkins?
  - https://www.infoworld.com/article/3239666/what-is-jenkins-the-ci-server-explained.html
  - https://www.edureka.co/blog/what-is-jenkins/

- Docker
  - Docker Revolution - https://www.infoworld.com/article/3204171/what-is-docker-the-spark-for-the-container-revolution.html
  - https://www.edureka.co/blog/what-is-docker-container