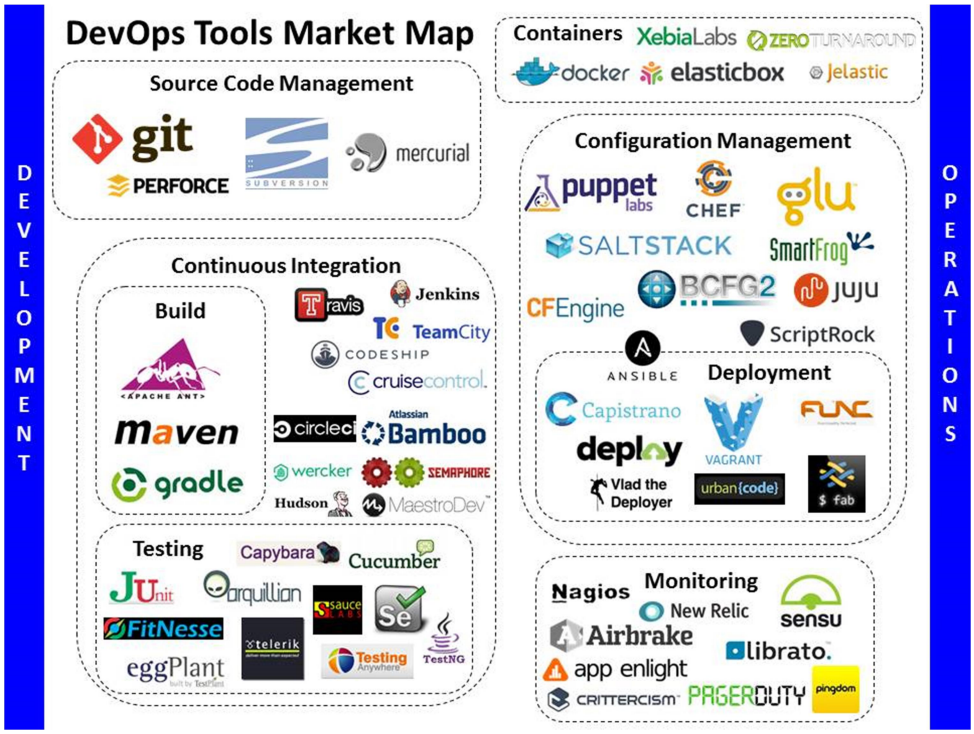
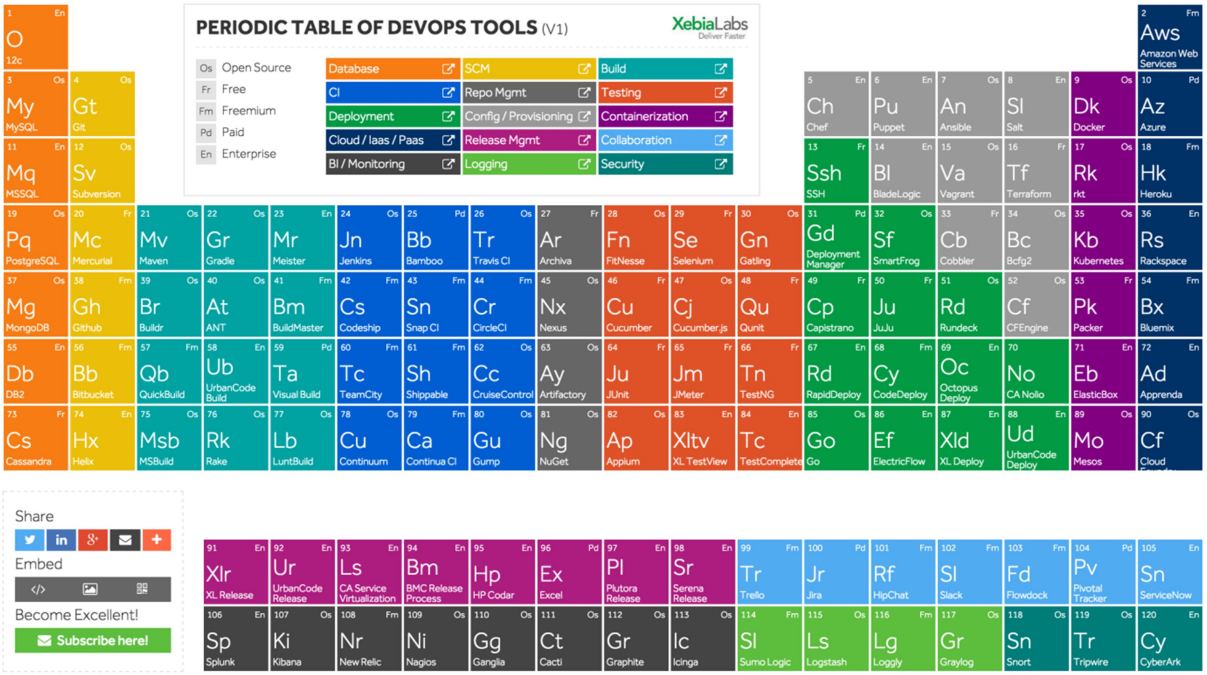
Proposed TOC/Agenda for DevOps with Jenkins 5 days Training: Accenture

1. **Introduction to DevOps (1/2 day)**
   - Comparison -> Waterfall, Agile & DevOps methodologies
   - Understanding the DevOps movement & culture
   - DevOps Lifecycle – All About 'Continuous'
     - Continuous Development
     - Continuous Testing
     - Continuous Integration
     - Continuous Deployment
     - Continuous Monitoring
     - Continuous Feedback
   - DevOps Principles, Team Topologies
   - Adoption Strategy & Milestone based planning – Process, Culture, people skills & tools, KPIs
   - DevOps Tools overview



- DevOps Periodic Table Walk through & criteria for choosing various Tools – Atlassian (BitBucket, Jira, Bamboo, Confluence) v/s AWS (CodeCommit, CodeBuild, CodeDeploy, CodePipeline)



- Some DevOps Terminologies – Blue Green Deployment, Rolling Updates, Mean Time to Recovery, Configuration drift, Snowflake Server
- Importance of Testing, Automation, % Coverage, TDD
- DevSecOps

2. Source Code Version Control Tools using GitHub (1 day)

- Comparison of various Source Code Tools – Client Server & Distributed attributes
- Various Git Flavors : GitHub, BitBucket, GitLab, GitStack etc & feature comparison
- Git 101 using free Cloud Server GitHub.com -> Server & Client set up – Registration on GitHub.com, installing Git Client (https://git-scm.com/download/win) on all participant's machine, SSH Key generation & updating it on Git server, sample repository set-up and basic commands
- Git Code Review using Git Pull requests – Lab exercise to create a Simple Branch for a feature and raise Pull requests for code review & subsequent merge
- Best practice while working with Git - Reset v/s Revert, amend, cherry pick do's & don'ts, rebase etc
- Introduction to Maven, sample maven based Java web application exercise to build, compile and package
- Git Working Model – Clone V/s Collaborative (pros & cons), GitHub Repository Organization Best practices, Overview of Typical Branching Model

3. **Continuous Integration & Jenkins (1 ½ days)**
    - Need for CI & Martin Fowler's definition of CI
    - Comparison of various CI Servers & introduction to Jenkins
    - Installation of JDK, Maven & Jenkins (Generic War File) on all participants machine
    - Lab exercise to set up simple Free Style Jobs and setting up a job to pull your GitHub repository (created earlier) repo , build & execute it
    - Jenkins Configuration - Integration with various LDAPs, Granular access permissions, SMTP
    - Popular plugins & features – Static Code Analysis (PMD, checkstyle), JUnit, Pipeline, Master-Slave, Deployment, back-ups
    - Continuous Integration: GitHub + Jenkins Integration using web-hooks to trigger automated build based on code push
    - Jenkins Pipelines, setting up sample pipelines using UpStream Jobs, Groovy DSL pipeline scripts to Git checkout sample maven based repository, Unit test, deploy it and send appropriate notifications

4. **Docker & Containers (1 ½ days)**
    - Docker Introduction, container analogy, difference between containers & VMs
    - Docker Workflows
    - Docker Architecture – Engine, Container, Docker host, Image, Registry
    - Docker Installation on participants laptops (Windows & Mac)
    - Docker 101 commands, attaching & detaching from containers, daemon containers, starting, stopping & removing containers
    - Sample exercises for running various containers & Applications – Alpine Linux, java, PHP
    - Dockerfile detailed syntax – FROM, COPY, ADD, ENV, USER, WORKDIR, VOLUME, EXPOSE, RUN, CMD, ENTRYPOINT, .dockerignore file
    - Building & storing images, pushing them to public repositories, Docker hub registration
    - Understanding Storage, Layers & inspecting Docker images with exercise
    - Linking Containers – Storing data, volumes, mounting volumes, read-only mounts, Docker networking, mapping & exposing ports
    - Weaving multiple containers with Docker Compose
    - Docker Orchestration with Swarm – Swarm init, join as worker/manager, scale service, drain a node, service update and shutdown service – Demo with simple nginx service
    - Docker swarm service rolling update and rollback exercise

5. **Infrastructure as Code (IaC) & Configuration As Code using ANSIBLE (1/2 day)**
    - Brief overview & comparison of various IaC Tools : Chef, Puppet, Ansible, Saltstack, Terraform, Python Boto3 Library
    - Demo of Python Boto3 SDK APIs to spin up AWS Instances, Upload files to S3 Buckets, list IAM users etc
    - Demo of ANSIBLE set-up required for controlling Linux AMI machines and running simple playbook to install NGINX server and inject index.html file of choice

Please Note: Few Topics can be dealt in detail or skipped as per need/interest

Lab Requirements:

- One Physical Machine per participant (16GB RAM, Core i5 CPU & 500GB Hard Disk)
- Virtualization should be enabled on all the physical machines (BIOS Level HyperV enabled)
- BaseOS Windows 10 or higher x64 Bit Operating System (Windows 10 pro)
- Install 7 Zip. https://www.7-zip.org/a/7z1801-x64.exe
- Install Virtual Box. https://download.virtualbox.org/virtualbox/5.1.28/VirtualBox-5.1.28-117968-Win.exe
- Install Docker Desktop for Windows https://docs.docker.com/docker-for-windows/install/
- Jenkins Windows MSI Installer 2.176.2 version (LTS version) : https://jenkins.io/download/
- Oracle Java SE Development Kit 8u221 https://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html
- Apache Maven 3.6.1 https://maven.apache.org/download.cgi
- Git Client for windows : https://git-scm.com/download/win
- Good & uninterrupted Internet connection