T-CAGE

Trillion’s Unique DevSecOps Platform

Introduction

T-CAGE or the **T**rillion, **C**ontinuous **A**gile **G**at**E**way is one of the first cloud agnostic DevSecOps, PaaS/SaaS platforms that utilizes technologies such as OpenShift, CloudForms, Ansible, Jenkins and Jira to enable seamless, easy Agile delivery. Unlike many traditional DevOps pipelines, T-CAGE has built in security scanning allowing security flaws to be caught and corrected as part of the development process.

Traditional DevOps

DevOps focuses on combining Software Development with software operations, utilizing automation and monitoring during all steps of the software development process. This grants numerous benefits including:

- **Increased Speed** – Teams are able to innovate and adapt better to changing markets and goals
- **Increased Reliability** – Application quality is constantly monitored and logged allowing for rapid application updates that still maintaining end user experiences
- **Improved Collaboration** – DevOps emphasizes teamwork, ownership and accountability allowing teams to better collaborate and share workflows.

Because of this, DevOps has become an industry standard and is used across industries to ensure timely and consistent software deployment.

T-CAGE DevSecOps

Within the traditional DevOps framework, security has become more of an afterthought, addressed at the end of the development cycle. Because of this, security vulnerabilities are not discovered until the very end of the process, if at all.

T-CAGE takes DevOps to the next level. By integrating security into the very fiber of the development process, T-CAGE minimizes security vulnerabilities and meshes security, development and business objectives. This creates an environment where *everyone* is responsible for security.

Benefits

The benefits of utilizing T-CAGE over a more traditional DevOps platform include:

- Cost Reduction – Security flaws are detected and fixed during development, thereby increasing speed of delivery
- Speed of Recovery – When there are security incidents, recovery time is decreased through use of templates

- Threat Hunting – By discovering threats and patching them early in the process, the final product is more secure and easier to market
- Secure by Design – Increased focus on security and automated security review empowers all developers to use secure design principles and take ownership of application security
- Security Auditing, Monitoring and Notification Systems – All security is managed in a way that it can be continuously enhanced to keep up with cyber threats

T-CAGE DevSecOps Pipeline

The T-CAGE DevSecOps Pipeline, depicted in *Figure 1*, provides a comprehensive solution for delivering secure and thoroughly tested code to Production. Following the Continuous Adaptive Risk and Trust Assessment (CARTA) approach the DevSecOps pipeline allows for continuously assessing the ecosystem risks and adapting as necessary, allowing for continuous Authority to Operate (ATO). Overall this reduces the time and cost by automating all testing and security scanning processes from the start of development.

Code(Development)

Code(Dev) begins with a developer creating a Feature Branch from the Master Branch on the Git repository. The Master Branch is locked, only allowing merges that have been peer reviewed and approved. On this Feature Branch the developer complete and test their work. Once the work is completed and checked in, the

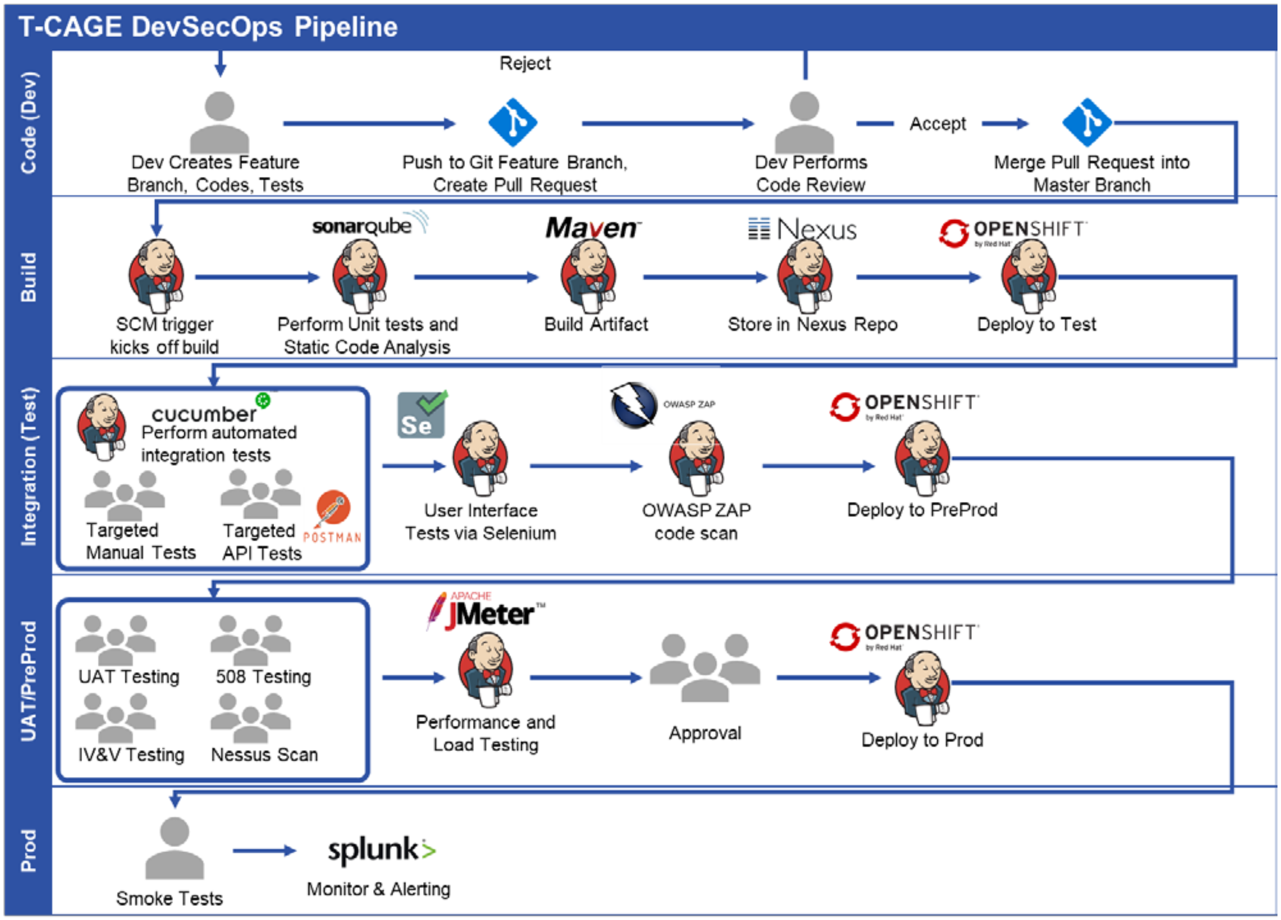

Figure 1: T-CAGE DevSecOps Pipeline

developer creates a Pull Request to have their code merged into the Master Branch. This initiates a code review, with n number of developers reviewing and approving/rejecting the changes. Once the code has been approved it is merged into the Master Branch, causing the Build phase to kick off.

Build
The Build phase is triggered by a code check in on the Master Branch. Jenkins is notified that a code change has been pushed onto the Master Branch, it checks out the code and begins its pipeline steps. The first step is running the unit tests and static code analysis using SonarQube. If the tests and analysis meet the defined passing threshold the next step occurs, else the build fails. Using Maven Jenkins builds the deployable artifact, once built it is then checked into the Nexus Repository. Finally, the compiled artifact is deployed into the OpenShift Test environment.

Integration (Test)
In the Integration(Test) phase Jenkins begins by running automated Integration Tests via Cucumber. If these pass the user interface test occur via Selenium. If the selenium tests pass the defined threshold then the OWASP ZAP security code scan occurs. While all these automated Jenkins tasks are running, targeted manual testing is performed by testers, and targeted API tests are performed utilizing the Postman software. When all testing is completed the artifact is deployed to the PreProd environment.

UAT/PreProd
In the UAT/PreProd phase UAT testing, 508 testing, IV&V testing and Nessus Scans are all manually performed by testers and security teams. Jenkins performs load testing via Jmeter. Once testing is complete, approval is needed before Jenkins will finally deploy the artifact to Production.

Production
In the Prod phase developers compete smoke testing. And continues monitoring and alerts are handled by Splunk.

Architecture

T-CAGE provides seamless DevSecOps while separating IaaS and Hypervisor with OpenShift and CloudForms, thus reducing the cost of maintenance of the application. In essence, T-CAGE makes our environment cloud agnostic. T-CAGE uses Ansible Playbooks to install and configure PostgreSQL/EnterpriseDB, Tomcat, Mongo DB/Alfresco, and Pentaho Data Integrator as infrastructure as code with Red Hat Linux, OpenShift pods, serviced with Kubernetes and Docker containers. Other tools and technologies— such as Jenkins, JIRA for Agile Lifecycle Management of the projects, Git, Bitbucket/GitLab for source code, Maven for the build process, Nexus for code repository, SonarQube for code quality, and Liquibase for database configuration management—are also used. T-CAGE allows maintenance of the same configuration across Dev, Test, Pre-Prod, and Prod environments. All scripts and software modules are checked into the repository for version management and source control. T-CAGE is a cloud and hypervisor agnostic platform.

Summary

T-CAGE combines Trillion’s extensive experience with Cyber Security and DevOps tocreate a robust development platform that comprehensively addresses many of the weaknesses inherent with more traditional DevOps processes. T-CAGE builds security into every step of the development lifecycle.