DevSecOps

What is DevSecOps:

DevSecOps is the practice of integrating security testing at every stage of the software development process. It includes tools and processes that encourage collaboration between developers, security specialists, and operation teams to build software that is both efficient and secure.

DevSecOps in the SDLC:

The DevSecOps framework improves the SDLC by detecting vulnerabilities throughout the software development and delivery process.

DevSecOps Toolchain

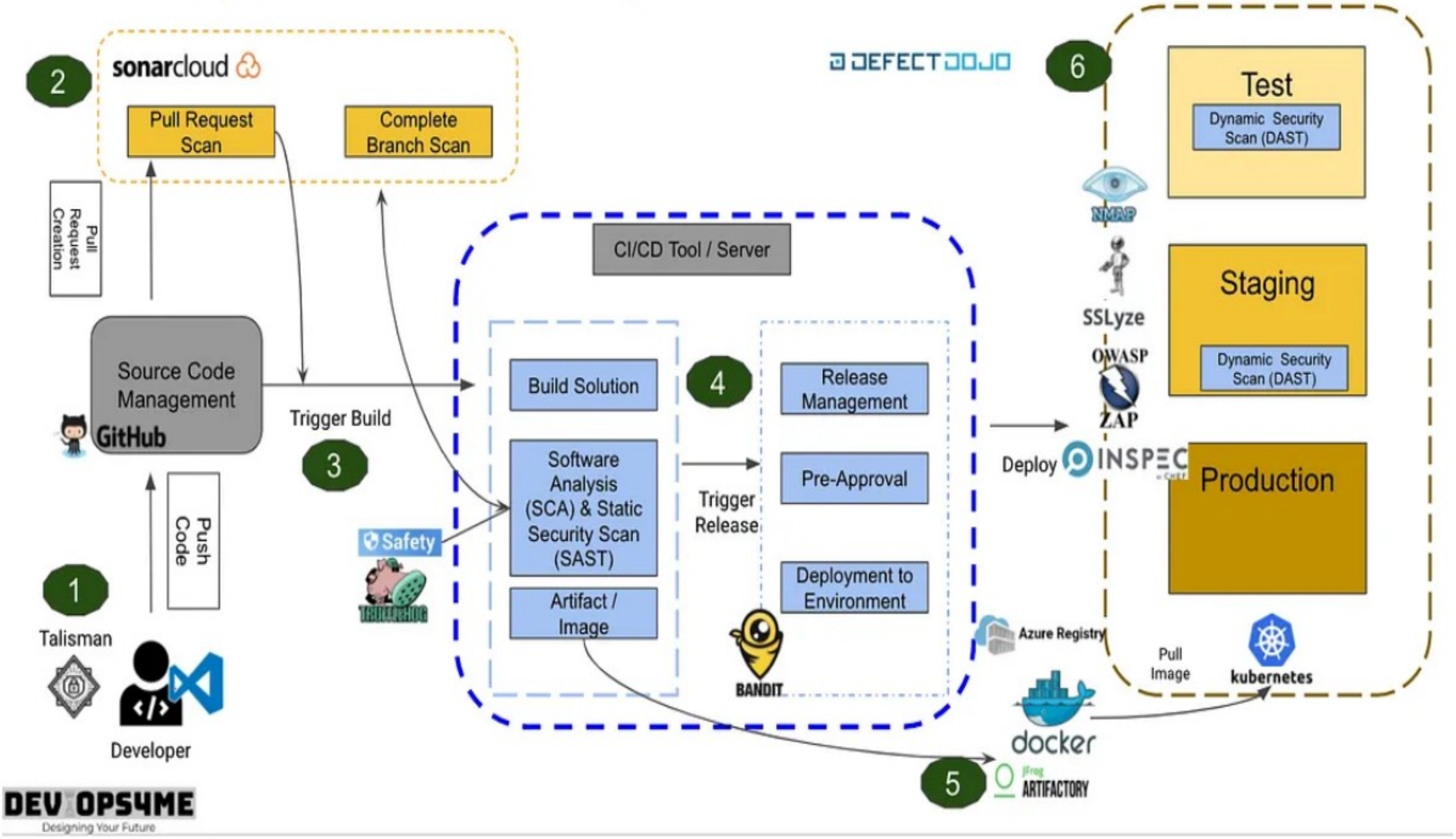
A DevSecOps toolchain is a series of integrated tools that automate and incorporate security practices throughout the development pipeline. Here is a list of some commonly used tools in a DevSecOps toolchain:

1. **Version Control and Source Code Management:**
    - Git, GitHub, Git Lab
2. **Dependency Management:**
    - OWASP Dependency-Check
3. **Static Application Security Testing (SAST):**
    - SonarQube
4. **Dynamic Application Security Testing (DAST):**

    - OWASP ZAP

5. **Software Composition Analysis (SCA):**
    - WhiteSource, Synopsys,Snyk
6. **Container Management and Security:**
    - Docker, Kubernetes,
7. **Secret Management:**
    - HashiCorp Vault, AWS Secrets Manager, Azure Key Vault
8. **Monitoring and Logging:**
    - ELK Stack (Elasticsearch, Logstash, Kibana), Splunk
9. **CI/CD:**
    - Jenkins, CircleCI, Travis CI, GitLab CI

Example of DevSecOps Workflow



Main stages of DevSecOps:

Planing:

- **Tools:** Jira (for user stories and acceptance criteria), Threat modeling tools (e.g., Microsoft Threat Modeling Tool).
- **Example:** Define user stories that include security acceptance criteria, such as "As a user, I want multi-factor authentication to enhance account security."

Development

- **Tools:** GitHub (for version control), SonarQube (for static code analysis), ESLint (for code quality).
- **Example:** Implement SAST tools like SonarQube in the CI pipeline to scan code for vulnerabilities with every commit.

Build

- **Tools:** Jenkins, Maven, OWASP Dependency-Check.
- **Example:** Use Jenkins to automate builds and include OWASP Dependency-Check to scan for vulnerabilities in dependencies.

Testing

- **Tools:** OWASP ZAP (for dynamic testing), Selenium (for automated testing), JUnit (for unit testing).
- **Example:** Automate DAST with OWASP ZAP to scan the application for vulnerabilities after deployment to a test environment.

Deployment

- **Tools:** Docker, Kubernetes, HashiCorp Vault (for secrets management).

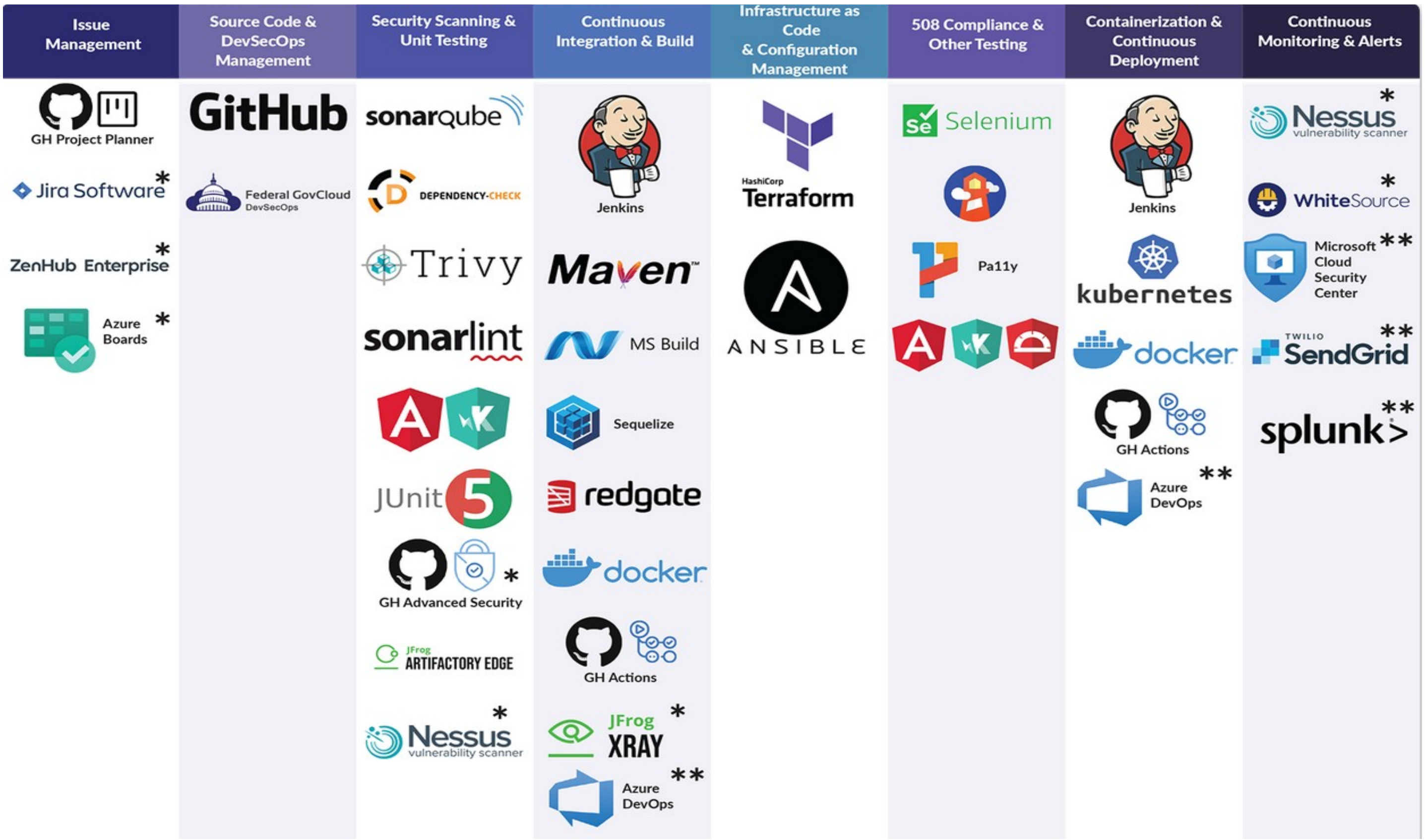
- **Example:** Deploy applications using Docker and Kubernetes, ensuring container images are scanned for vulnerabilities before deployment.

Operations

- **Tools:** ELK Stack (for logging), Splunk, Prometheus (for monitoring).
- **Example:** Set up ELK Stack to monitor application logs and detect potential security incidents in real-time.

Feedback and Continuous Improvement

- **Tools:** Jira (for tracking improvements), Security training platforms (e.g., OWASP resources).
- **Example:** Regularly update security policies and practices based on feedback from security incidents and testing results. Conduct regular training sessions for developers on secure coding practices.


Issue Management
GH Project Planner
Jira Software*
ZenHub Enterprise*
Azure Boards*
Source Code & DevSecOps Management
GitHub
Federal GovCloud DevSecOps
Security Scanning & Unit Testing
sonarqube
DEPENDENCY-CHECK
Trivy
sonarlint
JUnit 5
GH Advanced Security*
JFrog ARTIFACTORY EDGE
Nessus* vulnerability scanner
Continuous Integration & Build
Jenkins
Maven
MS Build
Sequelize
redgate
docker
GH Actions
JFrog XRAY*
Azure DevOps**
Infrastructure as Code & Configuration Management
HashiCorp Terraform
ANSIBLE
508 Compliance & Other Testing
Selenium
Pa11y
Containerization & Continuous Deployment
Jenkins
kubernetes
docker
GH Actions
Azure DevOps**
Continuous Monitoring & Alerts
Nessus* vulnerability scanner
WhiteSource*
Microsoft Cloud Security Center**
TWILIO SendGrid**
splunk>**