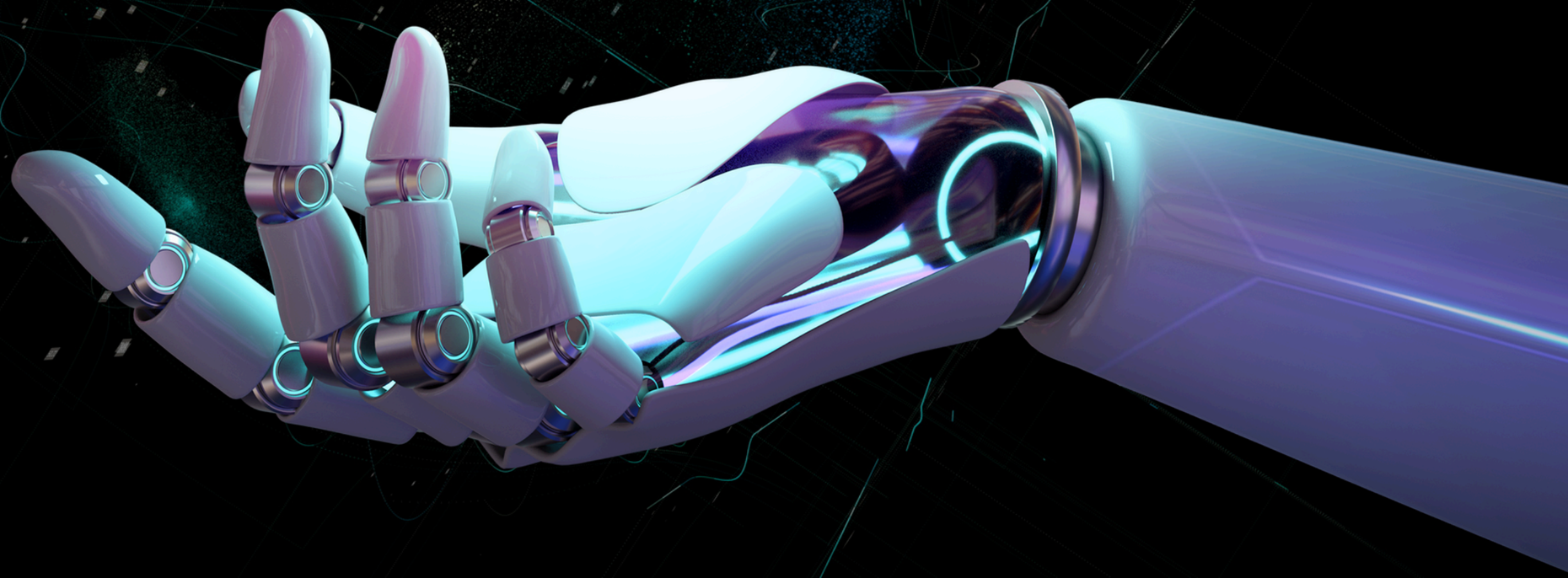
Code, Commit, Secure

DEVSECOPS IN ACTION



Michael Tayo
/michaeltayo

About

B.S. Information Technology, **Marquette University**
M.S Information Security & Assurance, **Strayer University**

Principal Security Engineer, U.S. Bank, AVP, Cloud and Application Security – Enterprise Security Architecture

---

Research: Cloud Security, DevSecOps, Cyber Fusion, Security Operations, Automation, Leadership and Product Management

---

Author: Anomali – Collaborative Security to Defend the Modern Landscape

---

Hobbies: Traveling, Music Concerts, Playing Sports

Michael Tayo
/michaeltayo

Disclaimer

The opinions expressed in this presentation are my own and do not necessarily reflect the opinions of my employer or any other organization.

This presentation is for educational purposes only.

I am not affiliated with any of the products or services mentioned in this presentation, and I have not been paid to endorse any of them.

All tools and resources shown in this presentation are for reference purposes only.

I encourage you to do your own research before making any decisions.

Objectives & Key takeaways:

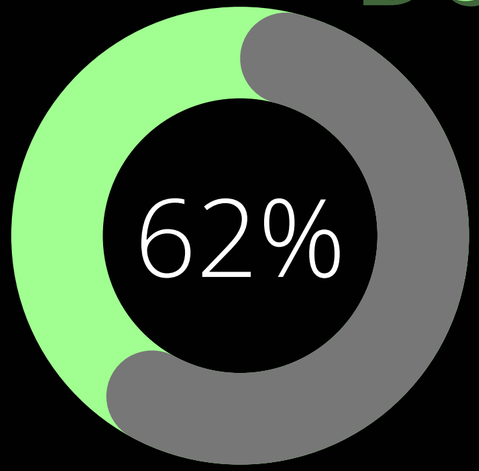
Comprehensive Understanding of DevSecOps

Practical Demonstrations and Real-World Examples

Actionable Best Practices

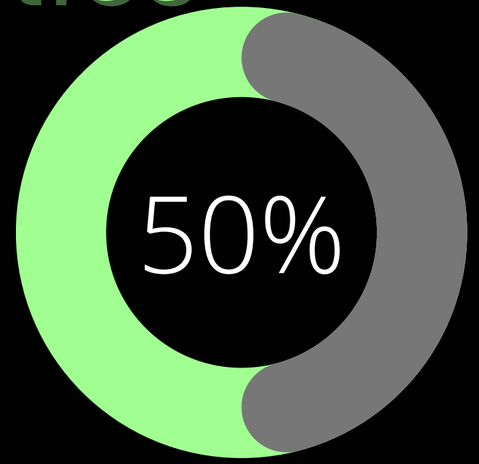
Table of contents

DevSecOps Statistics



Increase in Security Automation

62% of organizations reported using automated security tools within their DevOps pipelines, up from 30% in 2018
- Puppet 2021



Improved Security Posture

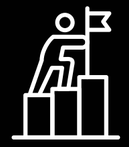
Organizations that have fully integrated DevSecOps practices experience 50% fewer security breaches and **can remediate vulnerabilities 90% faster** compared to those with minimal integration
- Sonotype 2021

What's the Dev**Sec**Ops Mission?

... creating **targeted customer value** through **secure iterative innovation** at **speed & scale.**

What is DevSecOps?

It is a practice that integrates security measures into every stage of the software development lifecycle (SDLC), from initial design through to deployment and beyond.



Think of DevSecOps as a way to make sure that the code developers write is secure right from the start, rather than adding security checks at the end.

Development, Security, & Operations

Importance in Today's Software Development

Early Detection of Vulnerabilities

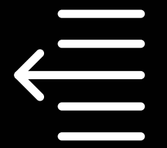
Speed and Efficiency

Cost-Effective

Enhanced Collaboration

Regulatory Compliance

Key Principles of DevSecOps



Shift Left Security

Automation

Collaboration

Continuous Monitoring

Immutable Infrastructure

Traditional Development vs. DevSecOps

Aspect	Traditional Development	DevSecOps
Security Integration	Late in SDLC	Early and continuous
Responsibility	Separate security team	Shared responsibility
Vulnerability Detection	Late-stage or post-deployment	Early and continuous
Development Speed	Slower due to late-stage fixes	Faster due to early issue resolution

Traditional Development vs. DevSecOps (cont'd)

Aspect	Traditional Development	DevSecOps
Cost of Fixing Issues	High due to complexity of late fixes	Lower, issues fixed early
Collaboration	Limited interaction	Close collaboration
Compliance	Periodic and manual	Continuous and automated
Tool Usage	Separate tools, minimal integration	Integrated toolchain, automation

DevSecOps

Plan and Develop

- [ ] Threat modelling
- [ ] IDE Security plugins
- [ ] Pre-commit hooks
- [ ] Secure coding standards
- [ ] Peer review

Commit the code

- [ ] Static application security testing
- [ ] Security unit and functional tests
- [ ] Dependency management
- [ ] Secure pipelines

Build and test

- [ ] Dynamic application security testing
- [ ] Cloud configuration validation
- [ ] Infrastructure scanning
- [ ] Security acceptance testing

Go to production

- [ ] Security smoke tests
- [ ] Configuration checks
- [ ] Live Site Penetration testing

Operate

- [ ] Continuous monitoring
- [ ] Threat intelligence
- [ ] Penetration testing
- [ ] Blameless postmortems

Source: **Microsoft**

Remember this...

Common Security Risks in DevOps

DEPENDENCY MANAGEMENT

Reliance on outdated or vulnerable third-party libraries can introduce security flaws.

STATIC ANALYSIS

Manual code reviews can miss critical security vulnerabilities in the source code.

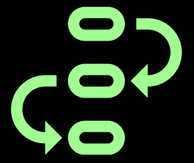
DYNAMIC ANALYSIS

Security vulnerabilities may go unnoticed until after deployment, posing significant risks.

INFRASTRUCTURE AS CODE

Inconsistent or insecure infrastructure configurations can lead to vulnerabilities.

Common Security Risks in DevOps (cont'd)



BUILD PIPELINES

Security checks are often manual and separate from the build process, leading to overlooked vulnerabilities.

ACCESS CONTROL

Inadequate access controls can lead to unauthorized access and data breaches.

DOCUMENTATION

Poor or outdated documentation can lead to insecure development practices and configurations.

ARTIFACT SIGNING

Unsigned artifacts can be tampered with, leading to security breaches.

Case Study

The Centers for Medicare & Medicaid Services (CMS)

The Problem

Addressing security as an afterthought or out of cycle activity created inefficiencies and additional risk

Case Study

The Centers for Medicare & Medicaid Services (CMS)

The Proposed Solution

Implement a security automation framework (SAF) into code pipelines.

- Infrastructure - creates the ability to destroy and re-create
- Code - enforces continuous security and quality checks with every code commit
- Security - provides configuration and vulnerability checks.

Case Study

The Centers for Medicare & Medicaid Services (CMS)

The Results

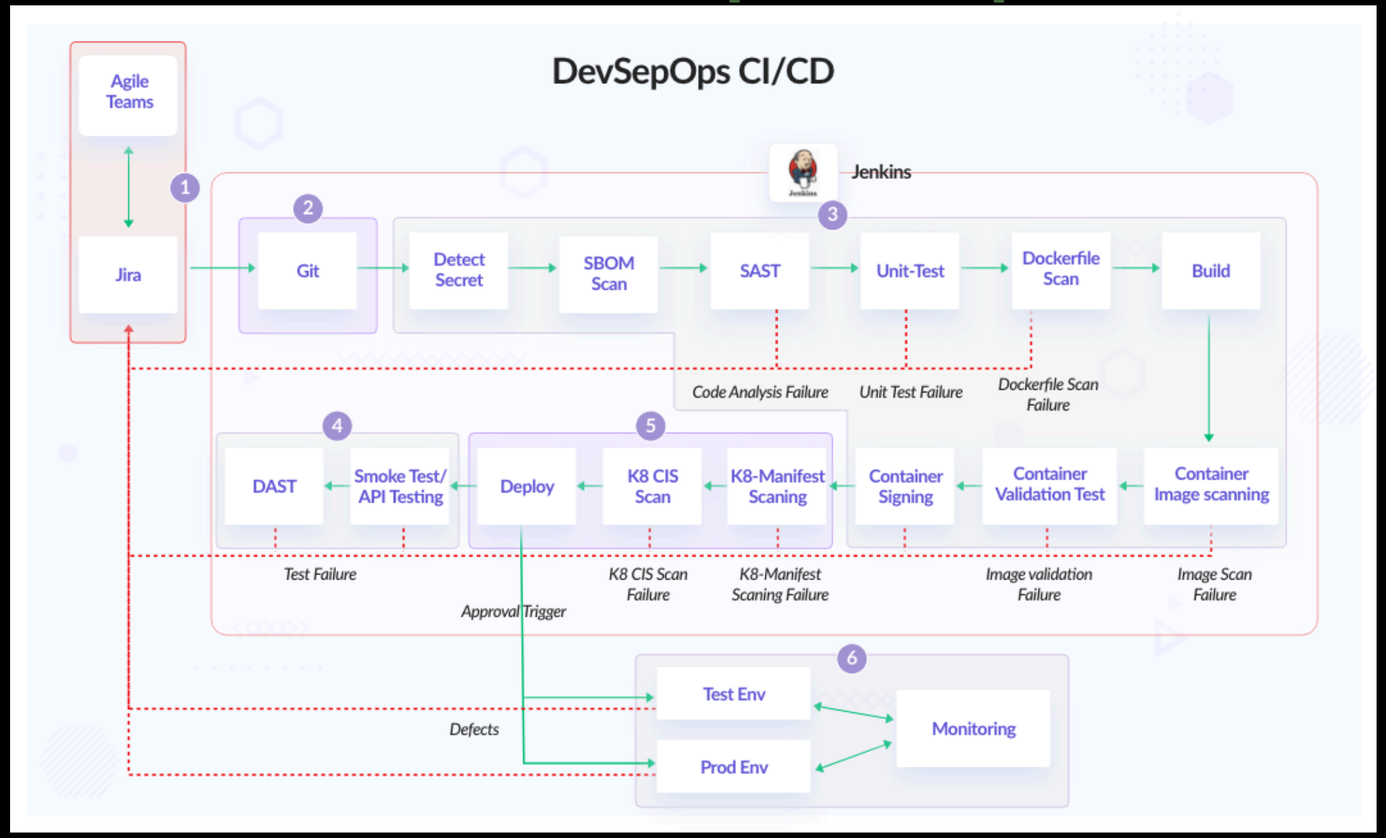
- Hardened operating system as a result of automation
- **Reduced burden** of security assessments
- **Continual improvement** of the security posture
- Increased product **quality**
- **Accelerated** software integration and deployment with precision
- Ability to validate configuration on demand
- Ability to automate product reconstruction for disaster recovery
- On-demand for testing and feature rollouts

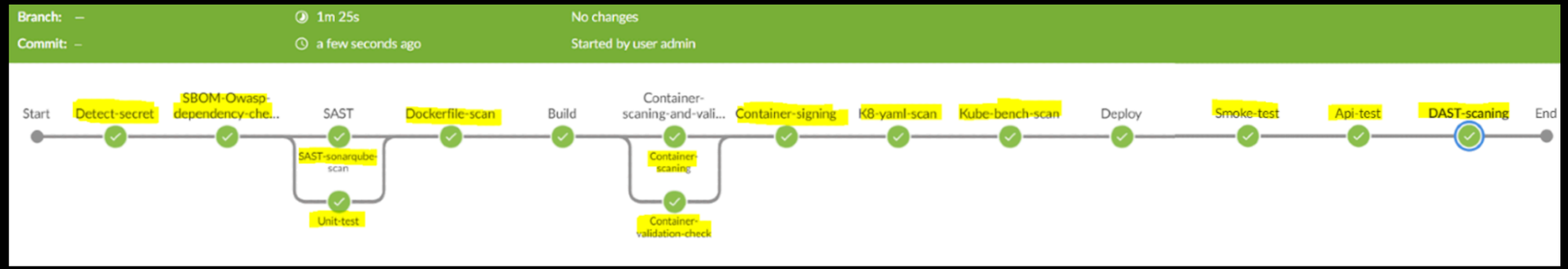
The DevSecOps Tool Marketplace

Remember: DevSecOps = Methodology

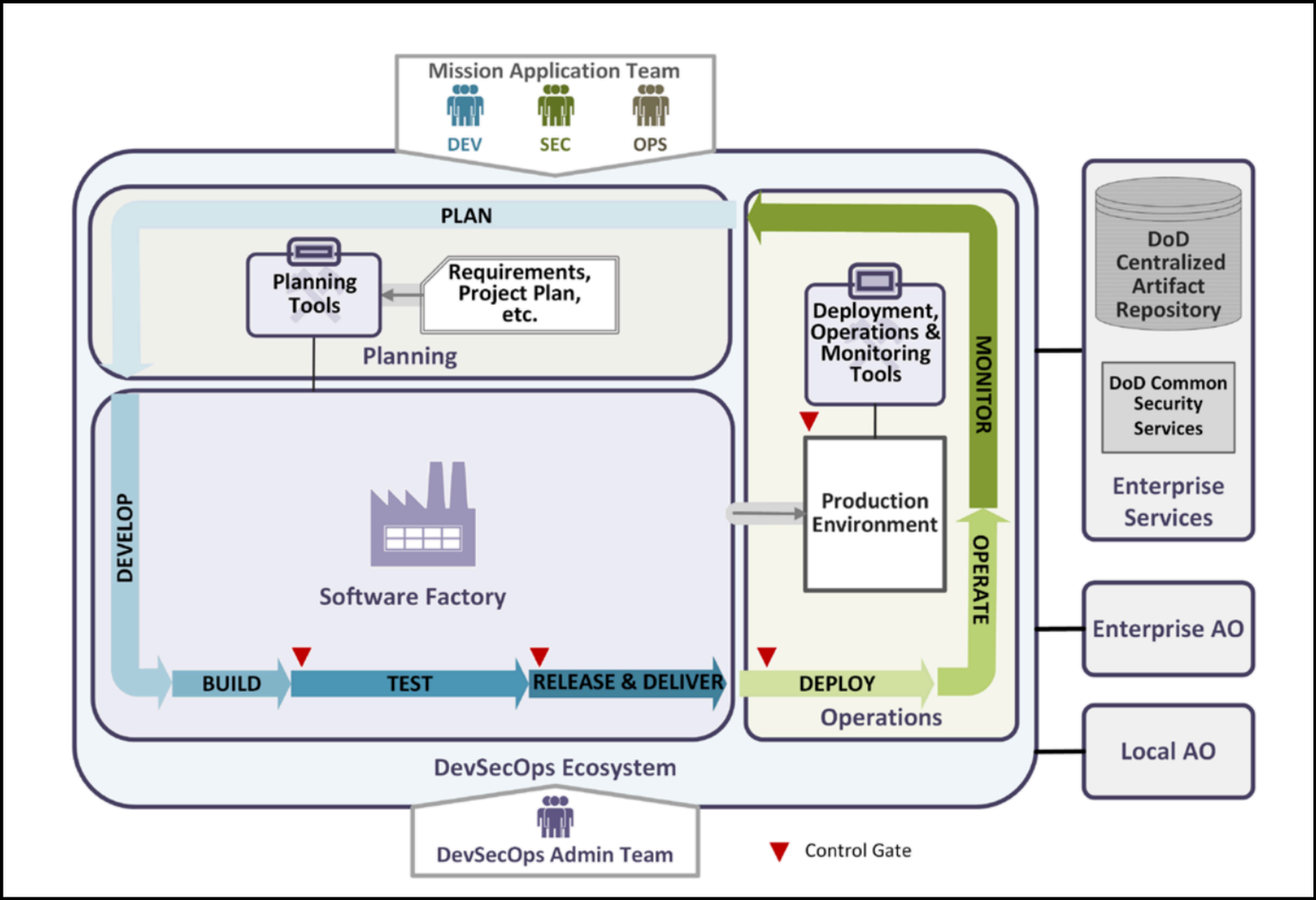
Enterprise DevSecOps Pipeline Example

Source: InfraCloud





DevSecOps Ecosystem



Source: **DoD Enterprise DevSecOps Reference Design**

Open Source Dev**Sec**Ops Pipeline

Source: **Michael Tayo (LinkedIn)**

Jenkins - Pipeline

Github - Source Code Manager

TruffleHog - Secrets Scanner (docker)

owasp/dependency-check - Software Composition Analysis (SCA)(bash script)

Sonarqube - SAST (docker container)

Maven - Build (running on instance)

Tomcat - Web HTTP server that runs Java code (running on instance)

Zap - DAST (running on docker)

Maven™

sonarqube

DEPENDENCY-CHECK

Secrets Scanning

Source Composition Analysis (SCA)

DevSecOps

Example Jenkins File

Static Analysis Security Testing (SAST)

Build

Deploy

Dynamic Analysis Security Testing (SAST)

webapp / Jenkinsfile

Michael Tayo and Michael Tayo put placeholder in jenkisnfile

Code | Blame | 60 lines (59 loc) · 1.62 KB | Code 55% faster with GitHub Copilot

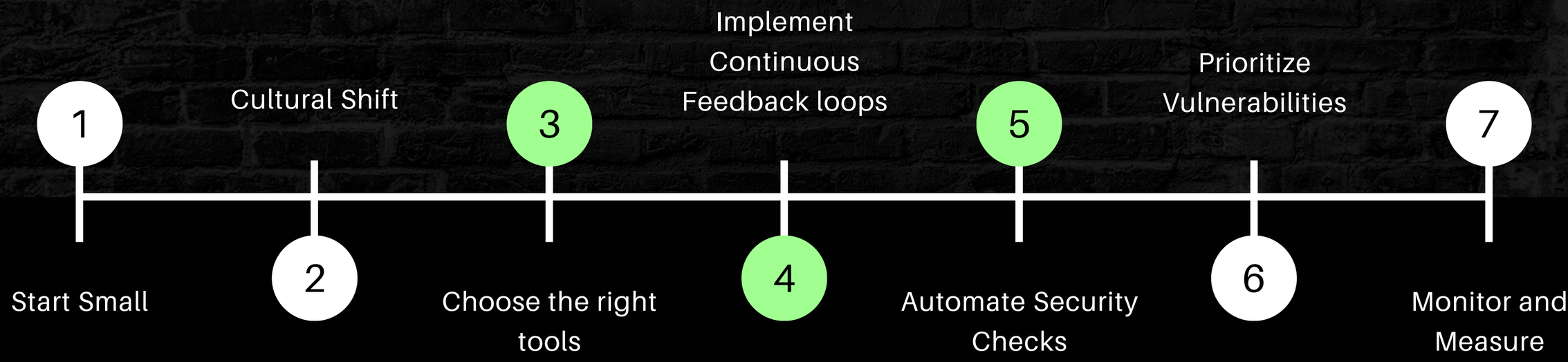
```
pipeline {
  agent any
  tools {
    maven 'Maven'
  }
  stages {
    stage ('Initialize') {
      steps {
        sh '''
            echo "PATH = ${PATH}"
            echo "M2_HOME = ${M2_HOME}"
        '''
      }
    }
    stage ('Check-Git-Secrets') {
      steps {
        sh 'rm trufflehog || true'
        sh 'docker run gesellix/trufflehog --json https://github.com/d0uble3L/webapp.git > trufflehog'
        sh 'cat trufflehog'
      }
    }
     stage ('Source Composition Analysis') {
      steps {
        sh 'rm owasp* || true'
        sh 'wget "https://raw.githubusercontent.com/d0uble3L/webapp/master/owasp-dependency-check.sh" '
        sh 'chmod +x owasp-dependency-check.sh'
        sh 'bash owasp-dependency-check.sh'
        sh 'cat /var/lib/jenkins/OWASP-Dependency-Check/reports/dependency-check-report.xml'

      }
     }
    stage ('SAST') {
      steps {
        withSonarQubeEnv('sonar') {
          sh 'mvn sonar:sonar'
          sh 'cat target/sonar/report-task.txt'
        }
      }
    }
    stage ('Build') {
      steps {
        sh 'mvn clean package'
      }
    }
  stage ('Deploy-To-Tomcat') {
            steps {
           sshagent(['tomcat']) {
                sh 'scp -o StrictHostKeyChecking=no target/*.war ec2-user@tomcat-server:/home/ec2-user/prod/apache-tomcat-9.0.74/webapps/webapp.war'
             }
           }
    }
    stage ('DAST') {
      steps {
        sshagent(['zap']) {
         sh 'ssh -o  StrictHostKeyChecking=no ec2-user@zap-server "docker run -t owasp/zap2docker-stable zap-baseline.py -t http://tomcat-server:8080/webapp/" || true'
        }
      }
    }
  }
}
```

Source: **Michael Tayo (LinkedIn)**

60 lines!!!!

Actionable Tips for Starting and Scaling DevSecOps



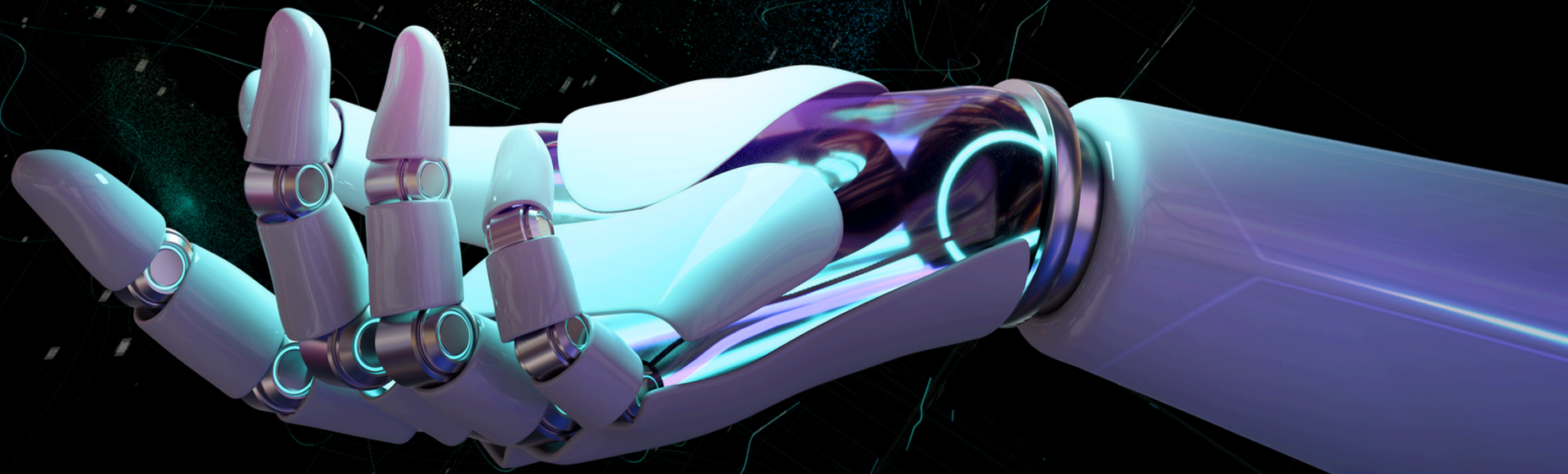
Summary

DevSecOps integrates security seamlessly into the development lifecycle, ensuring robust protection from the start.

Shift-left security, automation, collaboration, continuous monitoring, and immutable infrastructure are foundational to DevSecOps success.

Q and A

ASK AWAY!



TECHEX

THANK YOU

Michael Tayo
/michaeltayo