DevSecOps
What Why and How?

Anant Shrivastava
@anantshri
NotSoSecure Global Services

NOT SO SECURE
A Claranet Group Company

About: Anant Shrivastava

Director NotSoSecure Global Services

Sysadmin / Development / Security

Trainer / Speaker: BlackHat, Nullcon, RootConf, RuxCon, IPExpo, C0c0n

Project Owner: Android Tamer, Code Vigilant

Contributor: null, G4H, OWASP and more

https://anantshri.info (@anantshri on social platforms)

NOT SO SECURE
A Claranet Group Company

Agenda

- What is DevSecOps?
- Why do we need DevSecOps?
- How do we do DevSecOps?
- Integrate Security in DevOps Pipeline
- Tools of Trade
- Sample Implementation (On Prem and Cloud Native)
- Case Studies

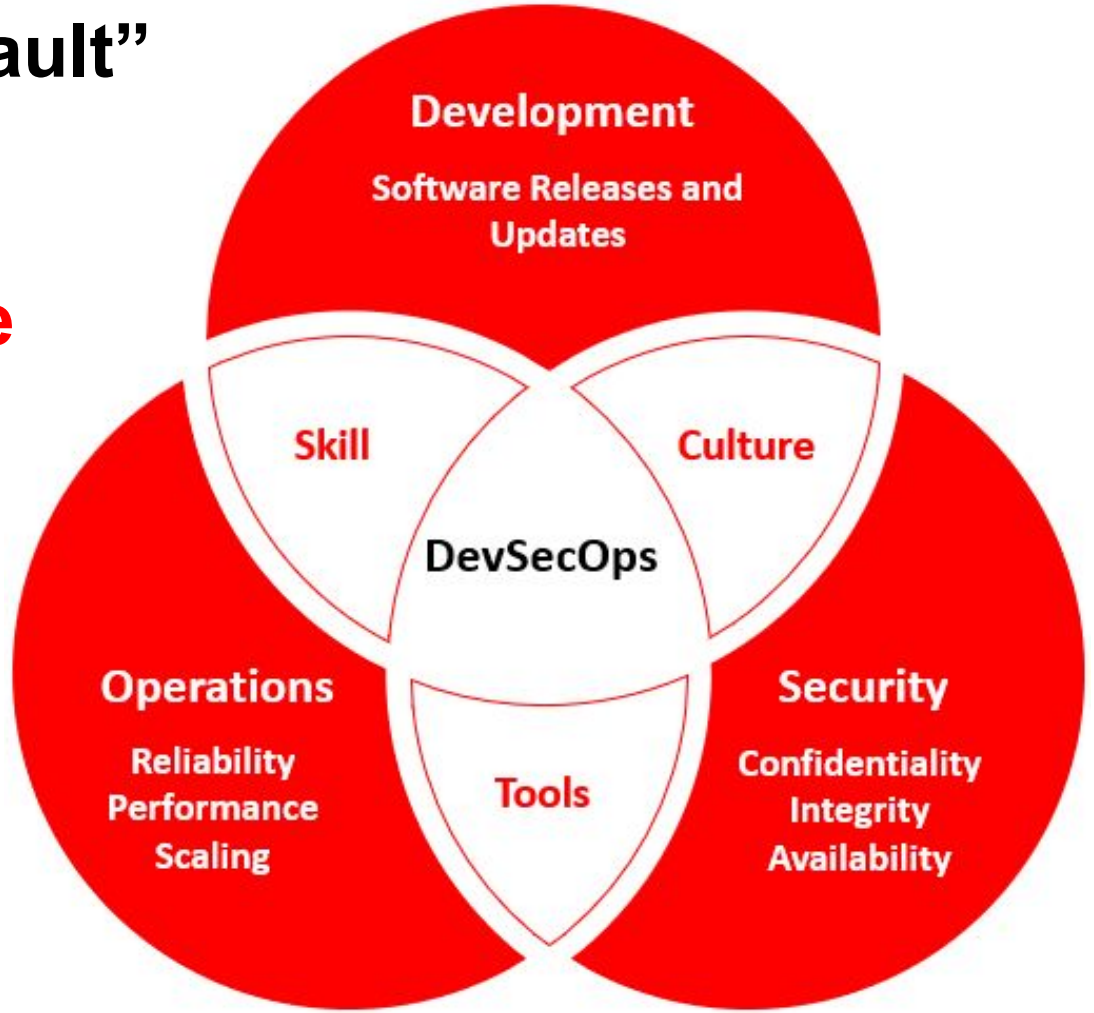
Disclaimer

- I will be listing a lot of tools, It’s not an exhaustive list
- I don't endorse or recommend any specific tool / vendor
- Every environment is different: Test and validate before implementing any ideas

What is DevSecOps?

Effort to strive for "Secure by Default"

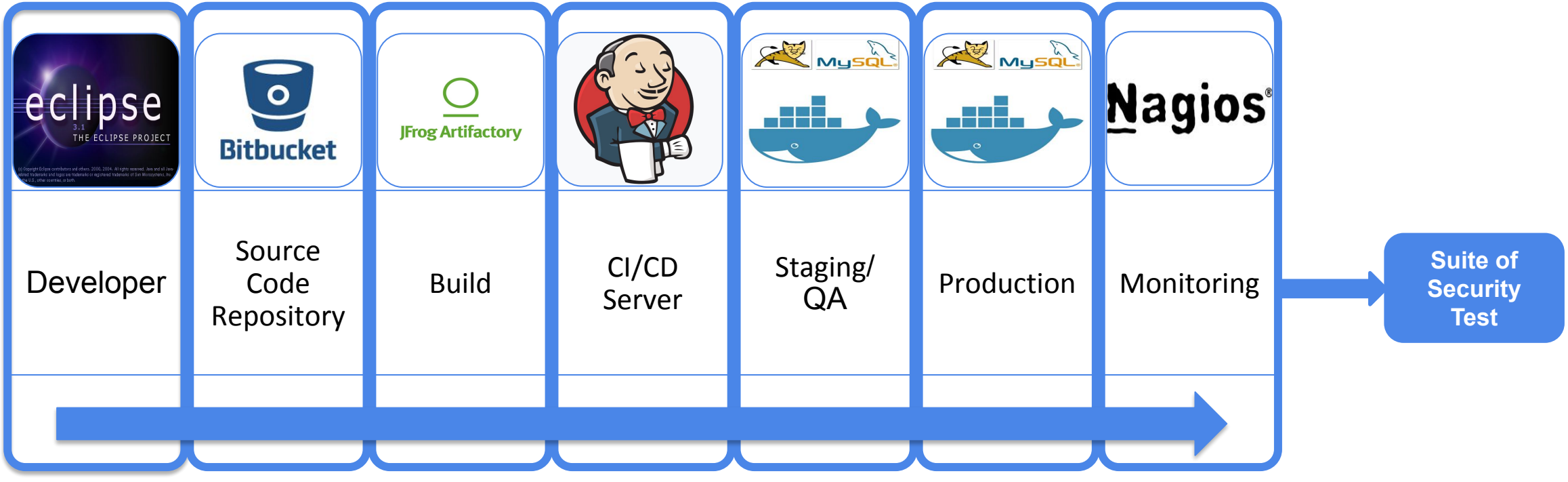
- **Integrate Security via tools**
- **Create Security as Code culture**
- **Promote cross skilling**



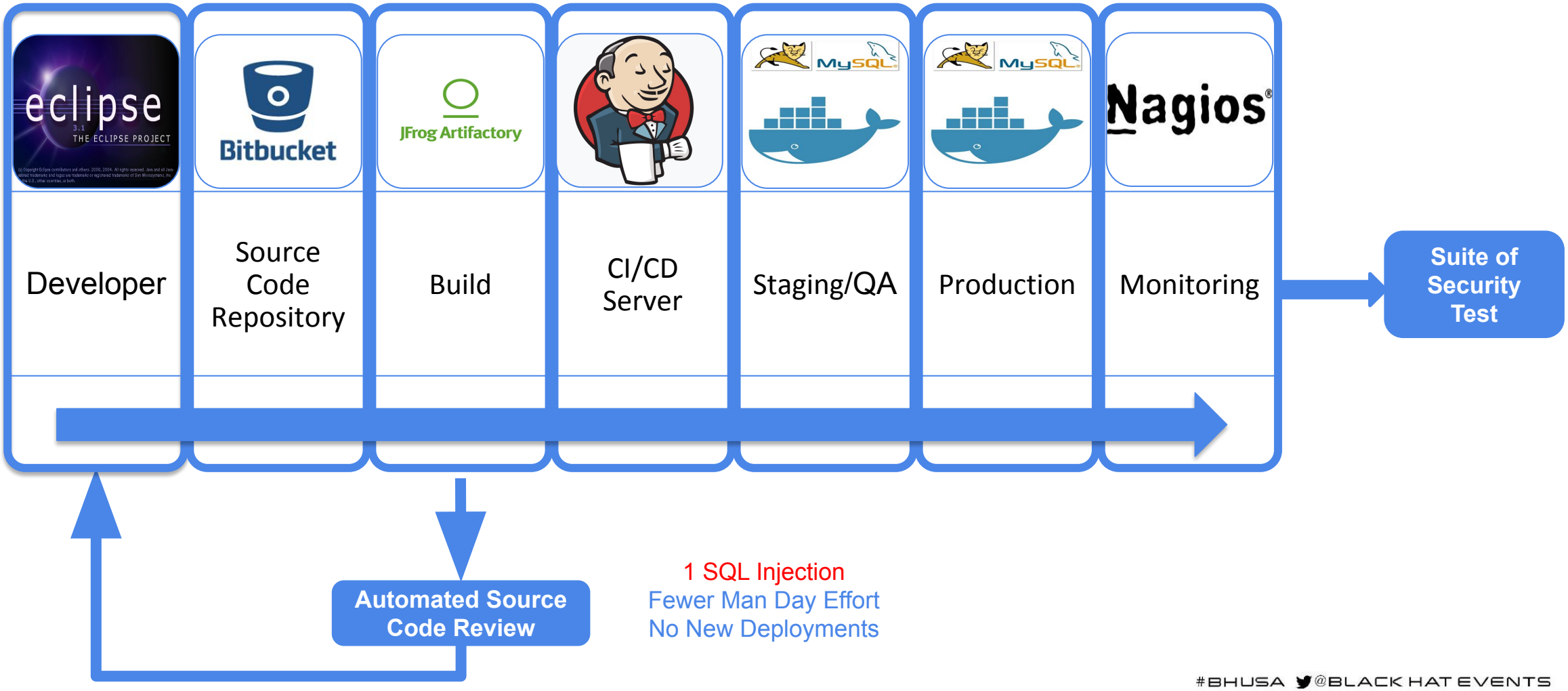
Why do we need DevSecOps?

- DevOps moves at rapid pace, traditional security just can't keep up
- DevSecOps makes it easier to manage rapid pace of development & large scale secure deployments
- DevSecOps allows for much smoother scaling of process
- Security as part of process is the only way to ensure safety

Shifting Left saves cost & time



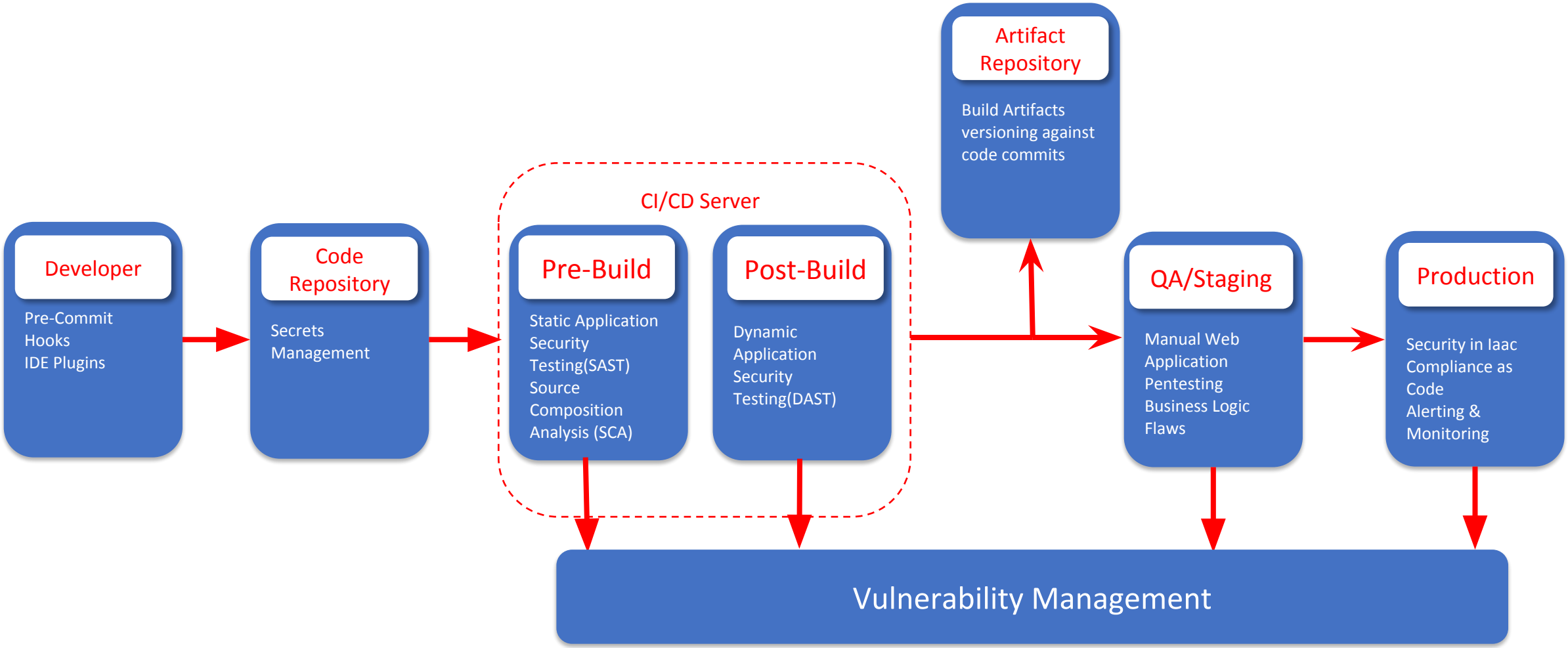
Shifting Left saves cost & time



How do we do DevSecOps?

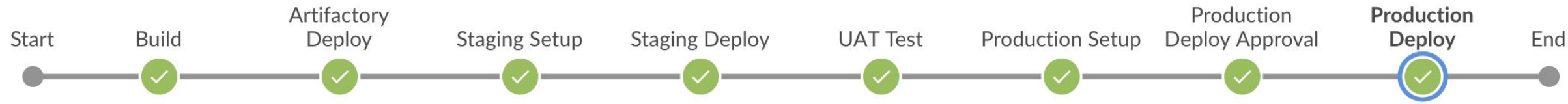
- DevSecOps is Automation + Cultural Changes
- Integrate security tools into your DevOps Pipeline
- Enable cultural changes to embrace DevSecOps

Injecting Sec in DevOps

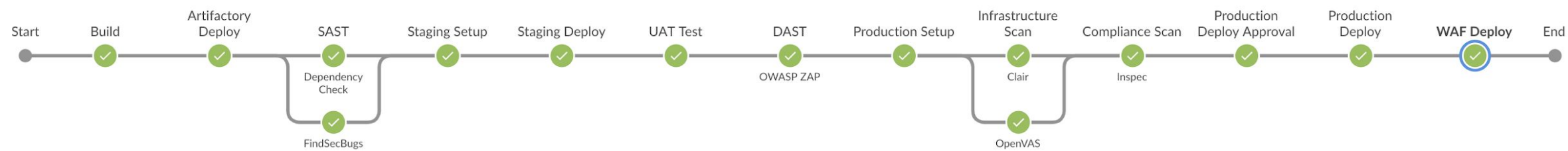


DevOps ---> DevSecOps

DevOps Pipeline



DevSecOps Pipeline



Pre-Commit Hooks

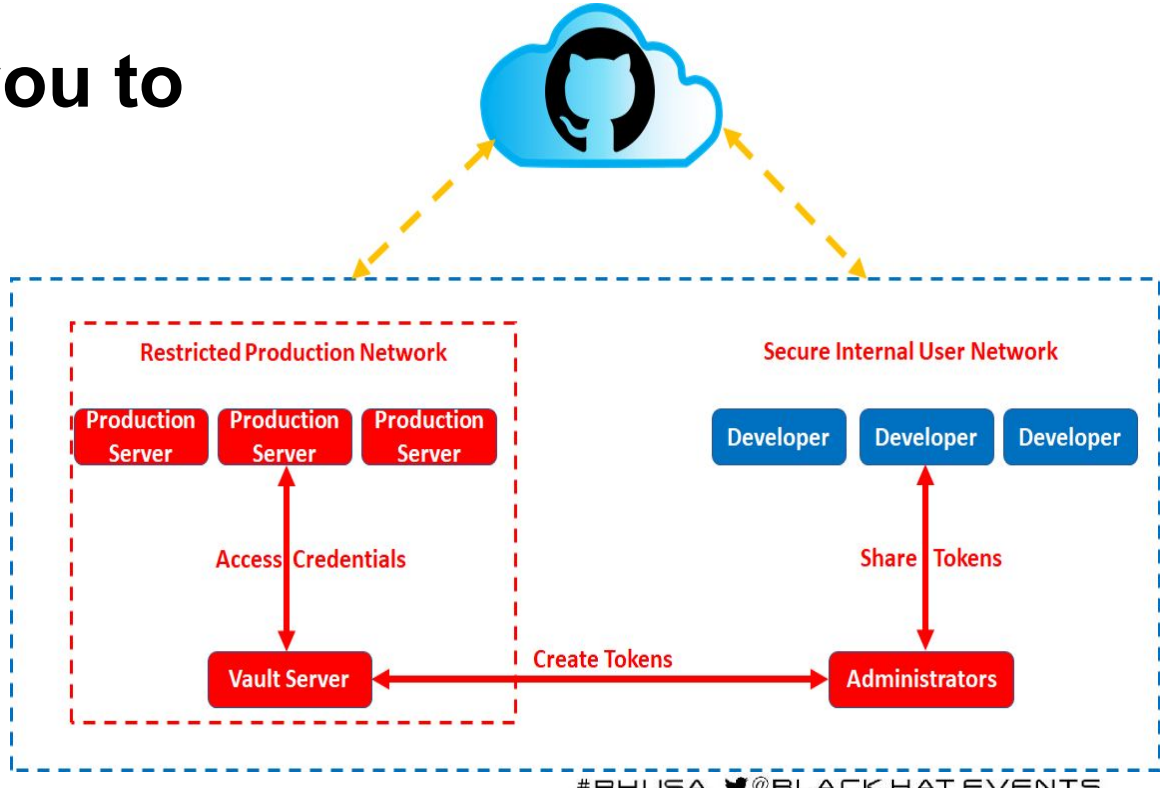
- **Sensitive information such as the access keys, access tokens, SSH keys etc. are often erroneously leaked due to accidental git commits**
- **Pre-commit hooks can be installed on developer's workstations to avoid the same**
- **Work on pure Regex-based approach for filtering sensitive data**
- ***If developers want they can circumvent this step hence use it like a defense in depth but don't fully rely on it***

IDE Security Plugin

- **IDE Plugin's provide quick actionable pointer to developers**
- **It is useful to stop silly security blunders**
- **Work on pure Regex-based approach**
- ***If developers want they can circumvent this step hence use it like a defense in depth but don't fully rely on it***

Secrets Management

- Often credentials are stored in config files
- Leakage can result in abuse scenario
- Secrets Management allows you to tokenize the information



Software Composition Analysis

- We don’t write software's, we build on frameworks
- Biggest portion of software is now third party libraries
- Major languages provide module managements
  - PIP, NPM, Gems, go get, perl cpan, php packager and more
- Software Composition Analysis performs checks to identify vulnerable/outdated 3rd party libraries

Static Analysis Security Testing

- White-box security testing using automated tools
- Useful for weeding out low-hanging fruits like SQL Injection, Cross-Site Scripting, insecure libraries etc
- Tool by default configured with generic setting, needs manual oversight for managing false-positives

Dynamic Analysis Security Testing

- Black/Grey-box security testing using automated tools
- SAST may not get full picture without application deployment
- DAST will help in picking out deployment specific issues
- Results from DAST and SAST can be compared to weed out false-positives
- Tools may need prior set of configuration settings to give good results

Security in Infrastructure as Code

- Infrastructure as a code allows you to document and version control the infra
- It also allows you to perform audit on the infrastructure
- Docker / K8s infra relies on base images
- Environment is as secure as the base image
- Base images need to be minimal in nature and need to be assessed to identify inherited vulnerabilities

Compliance as Code

- Compliance could be industry standard (PCI DSS, HIPAA, SOX) or org specific
- Compliance is essentially a set of rules and hence can be converted into written test cases
- Having written code format this can again be version controlled

Vulnerability Management

- All the tools discussed above result in report fatigue
- Every tool has a different style of presentation
- A central dashboard is required to normalize the data
- Vulnerability Management System can then be integrated to bug tracking systems to allow devs to work on items

Alerting and Monitoring

- Monitoring is needed for two end goals
  - Understand if our security controls are effective
  - What and where we need to improve

- To test Security control effectiveness:
  - When did an attack occur
  - Was it blocked or not
  - What level of access was achieved
  - what data was bought in and bought out

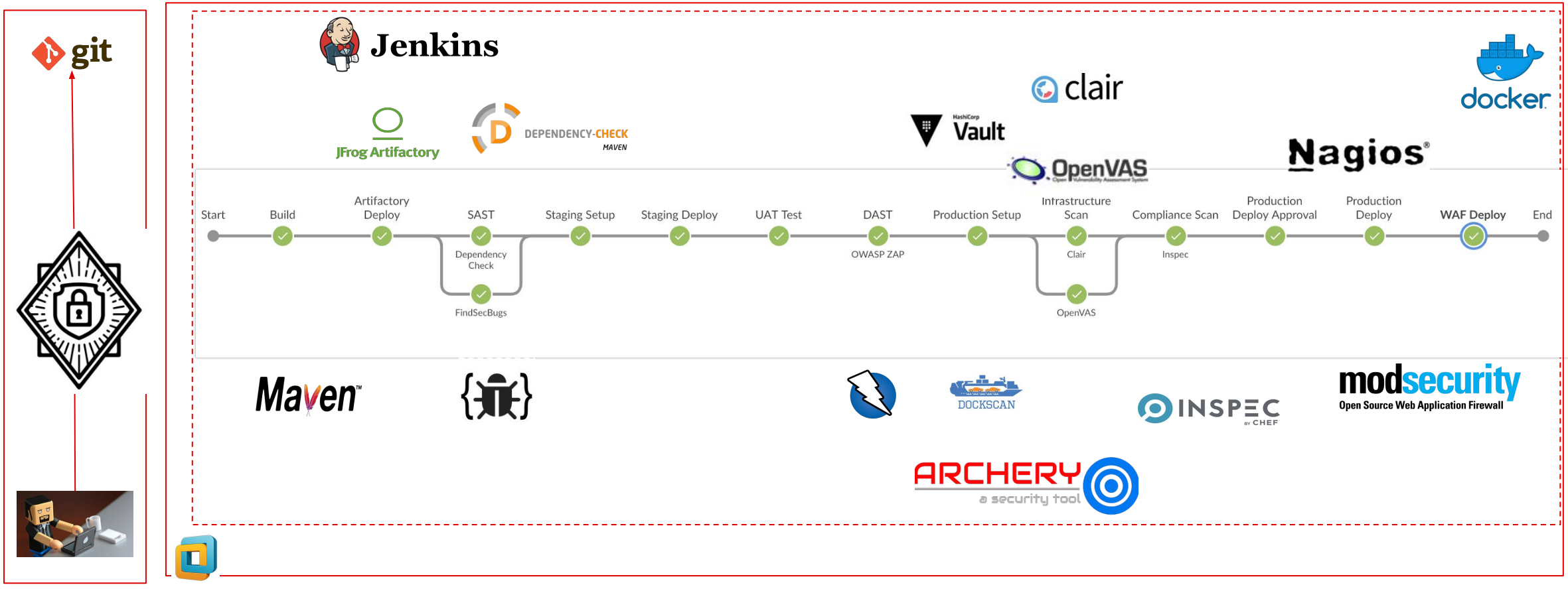
Asset Monitoring

- With recent advancements assets now should include anything and everything where organization data resides
- With rapid development & provisioning the asset inventory can't be a static inventory
- We need to monitor the assets constantly both on premise and Cloud

Reference: https://redhuntlabs.com/blog/redifining-assets-a-modern-perspective.html

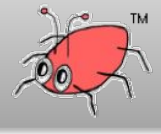
Sample Implementation - Java

A simplistic flow of DevSecOps Pipeline incorporating various stages



https://www.youtube.com/watch?v=7ILrzYLpr14

Tools of The Trade

Threat Modelling Tools	  ThreatSpec Microsoft Threat Modeling Tool
Pre-Commit Hooks	   truffleHog Git Hound
Software Composition Analysis	  Retire.js
Static Analysis Security Testing (SAST)	   
IDE Plugins	 CAT.net   
Secret Management	 Keywhiz  Confidant

Preference given to open-source tools; we don't endorse any tool

Tools of The Trade

Vulnerability Management	ARCHERY, JACK HAMMER, DEFECT DOJO
Dynamic Analysis Security Testing (DAST)	arachni, w3af, Wapiti
Security in Infrastructure as Code	OpenVAS, anchore, clair, DOCKSCAN, OpenSCAP
Compliance as Code	KitchenCI, INSPEC, DevSec Hardening Framework, Docker Bench for Security
WAF	modsecurity, NAXSI

Preference given to open-source tools; we don't endorse any tool

To be or Not to Be in Pipeline

- API / command line access
- Execution start to final output should be 15 minutes max
- Tools should be Containerizable / scriptable
- Minimal licensing limitations (parallel scans or threads)
- Output format parsable / machine readable (no to stdout, yes to json / xml)
- Configurable to counter false negatives / false positives

Pipeline Optimization

- Pipeline to be tweaked based on Milestone (Initiative/Epic/Story)
- Remember initial onboarding is tedious
- Ensure dataset dependent tool get frequent data refresh
- Sample optimization
  - Only CSS Changes: no need for SCA
  - Only pom.xml or gradle changes: no need of SAST
  - If Infra as code has zero changes skip or fast track infra scan
- Ensure to run full (non optimized) pipeline periodically

Does Programming Language Matter

- Different programming languages need different tools for static analysis and software composition analysis
- Some tools support multiple languages like sonarqube
- Others are focused on one language

Language Specific Tools

Languages	Software Composition Analysis	Source Code Static Analysis
JAVA	DEPENDENCY-CHECK, ClearlyDefined, OSS Review Toolkit	sonarqube, graudit
PHP	OSS Review Toolkit, sonatype	sonarqube, graudit
Python	DEPENDENCY-CHECK, Safety, sonatype	Bandit, graudit, sonarqube
.NET	DotNET Retire, SafeNuGet, OSS Review Toolkit	PUMA SCAN, DotNet Security Guard
Ruby/Rails	OSS Review Toolkit, sonatype	Brakeman, sonarqube, graudit
Node JS	ClearlyDefined, OSS Review Toolkit	npm-check, NodeJsScan, sonarqube

Preference given to open-source tools; we don't endorse any tool

DevSecOps Lab - Ruby

git

Jenkins

clair

docker

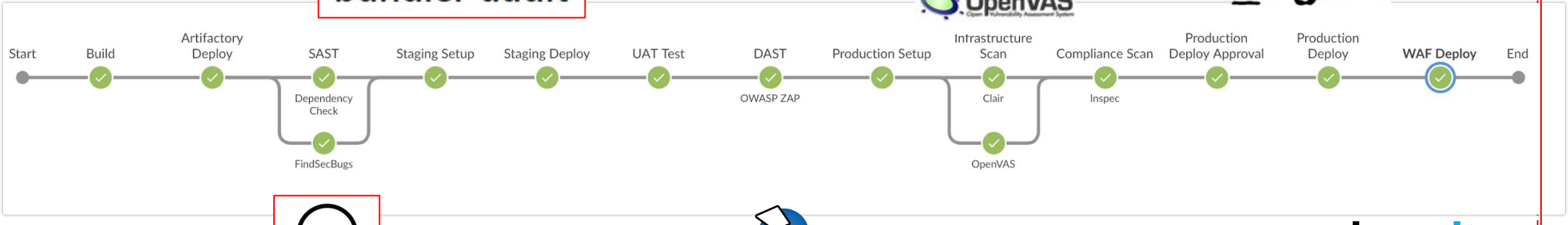
JFrog Artifactory

bundler-audit

HashiCorp Vault

OpenVAS

Nagios



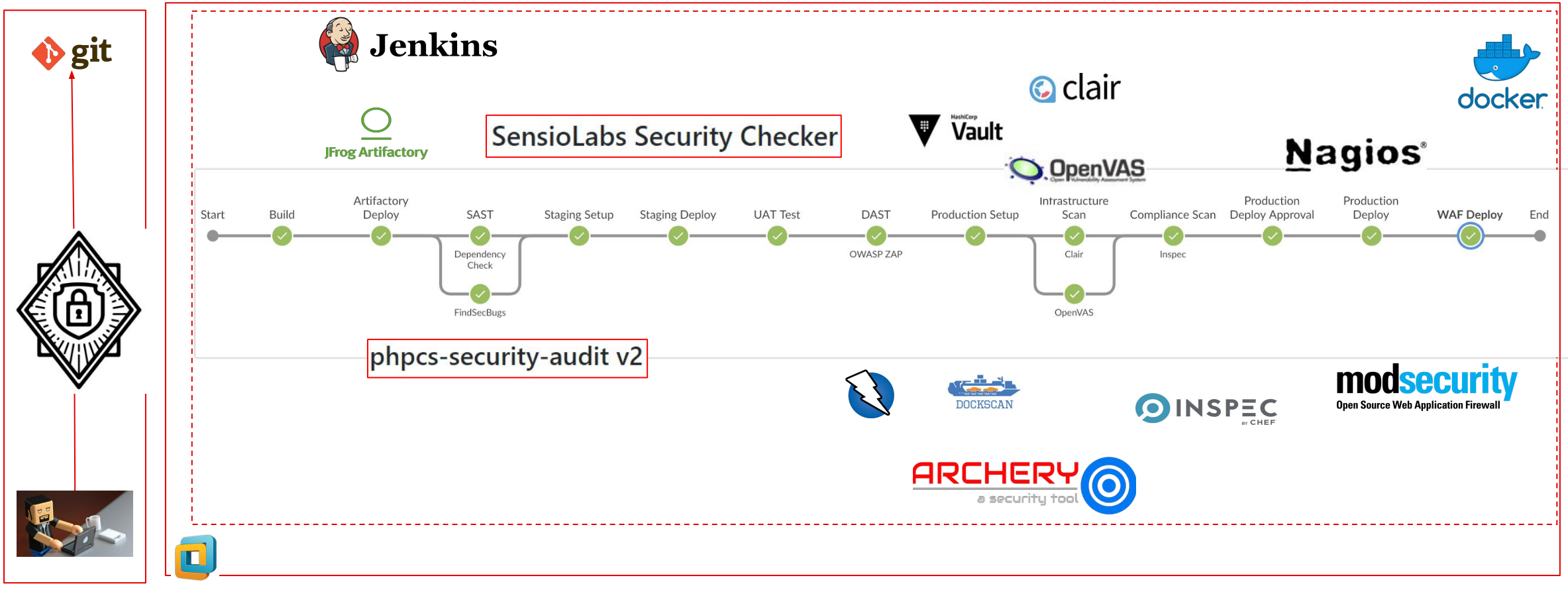
BRAKEMAN

INSPEC by CHEF

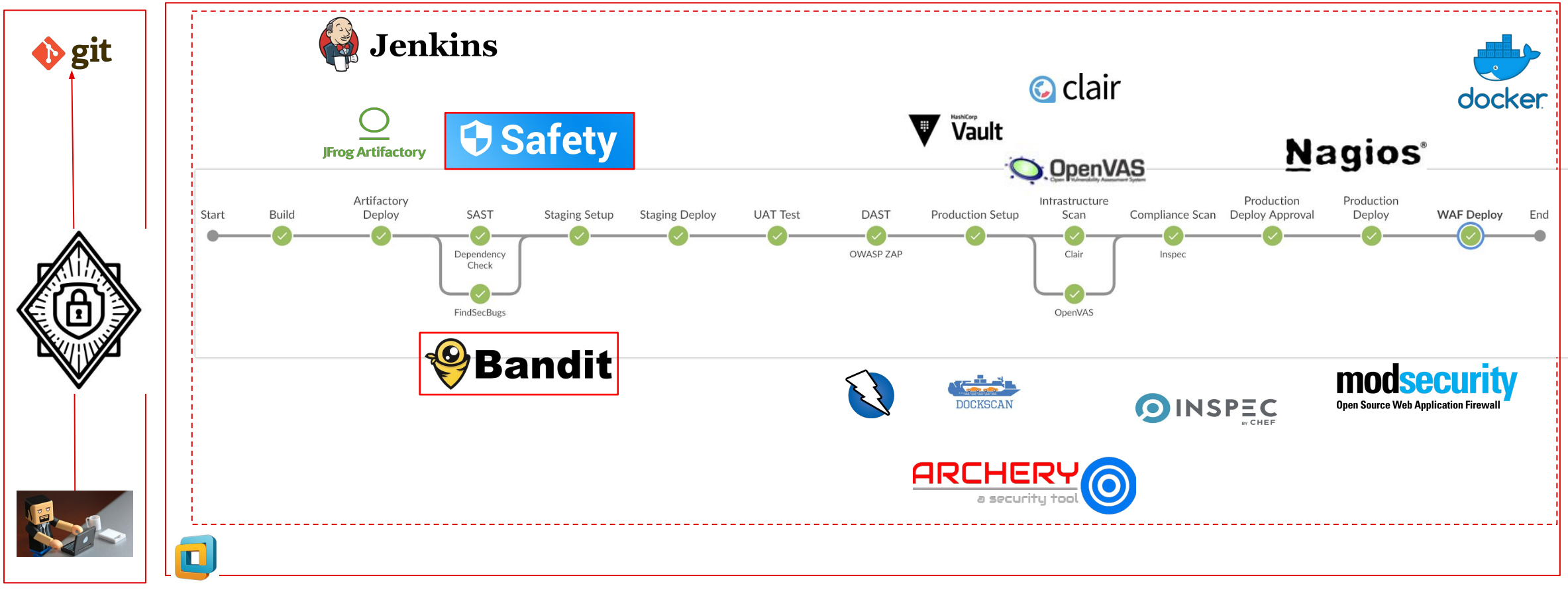
modsecurity
Open Source Web Application Firewall

ARCHERY
a security tool

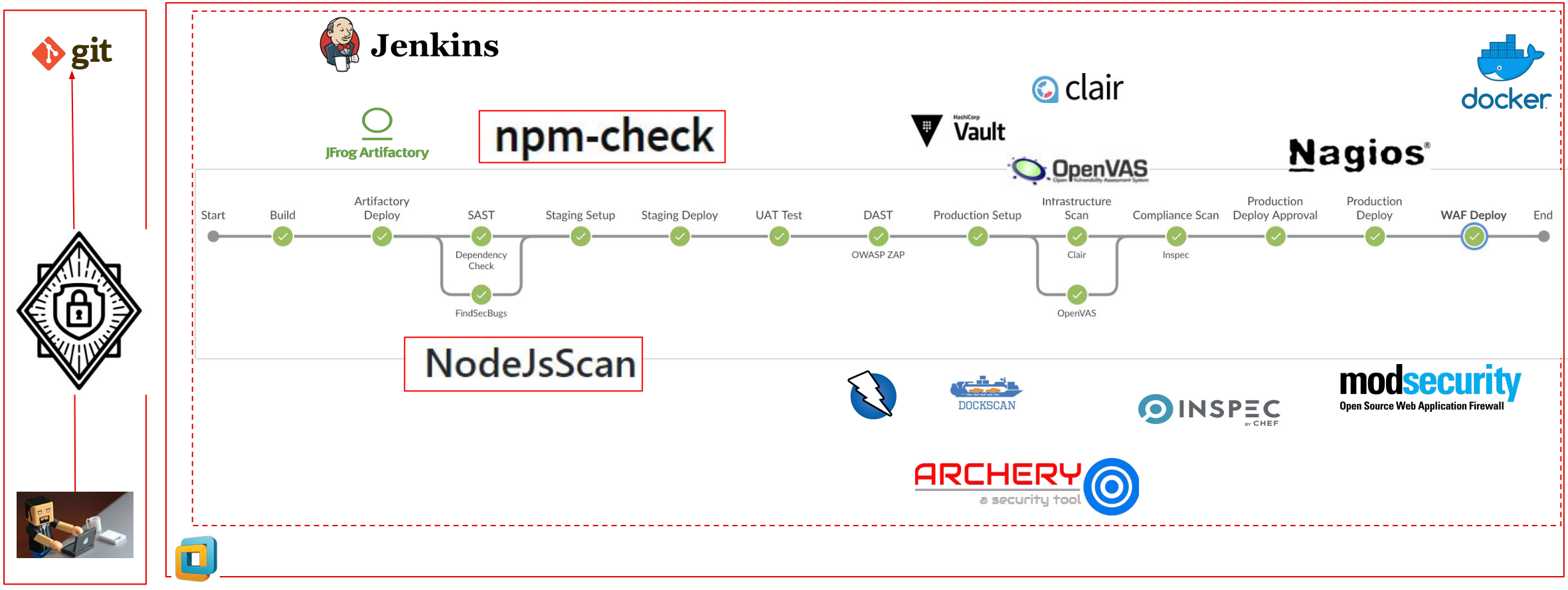
DevSecOps Lab - PHP



DevSecOps Lab - Python



DevSecOps Lab - NodeJS



What about Cloud

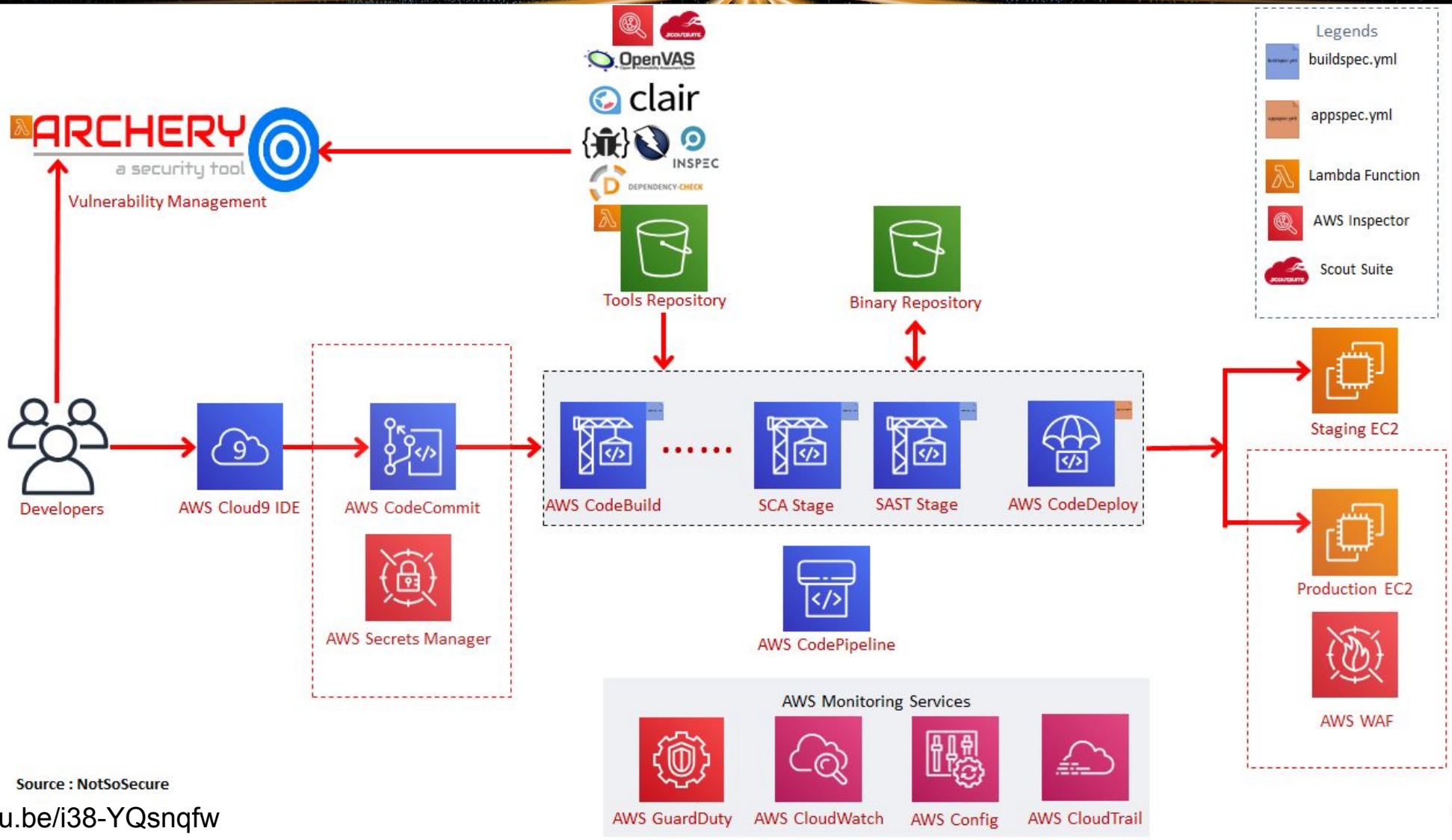
- **The Threat Landscape changes**
  - **Identity and Access Management**
  - **Asset Inventory**
  - **Billing**

- **Infrastructure as Code allows quick audit / linting**
- **Focus more on:**
  - **Security groups**
  - **Permissions to resources**
  - **Rogue / shadow admins**
  - **Forgotten resources (compromises / billing)**

Cloud Native Approach to Security

- Different Service Providers Approach Security Differently
- All of them provide some of the ingredient in-house
- Irrespective of Cloud provider some tools will still need to be sourced
  - Static Code Analysis Tool
  - Dynamic Code Analysis Tool
  - Software Composition Analysis
  - Vulnerability Management Tool

AWS Cloud Native DevSecOps



https://youtu.be/i38-YQsnqfw

Cloud Native Dev[Sec]Ops

	Conventional Infra	AWS	Azure	GCP
Source Code Management	Bitbucket, Github, Gitlab etc..	AWS CloudCommit	Azure Repos	Cloud Source Repositories
Infrastructure As a Code	Chef, Puppet, Ansible more..	Amazon CloudFormation	Azure DevTest Labs	Cloud Code
CI/CD Server	Jenkins, Bamboo, Gitlab, Travis CI, Circleci more	AWS CodeBuild AWS CodeDeploy AWS CodePipeline	Azure Pipelines, Azure Test Plans	Cloud Build, Tekton
Artifactory Repository	jFrog Artifactory, Sonatype Nexus, more..	Amazon S3	Azure Artifacts	Cloud Firestore
Stg/Prod Servers	VMWare, On-premises servers	EC2 ECS (Elastic Containers) EKS (Elastic Kubernetes)	Virtual Machines, Azure Lab Services, Azure Kubernetes Service (AKS)	Compute Engine, App Engine, Shielded VMs
Monitoring & Alert	Nagios, Graphite, Grafana	AWS CloudWatch	Azure Monitor, Network Watcher	Access Transparency
Firewall	Modsecurity	AWS Firewall Manager, AWS WAF	Azure Firewall	Application Gateway
DLP	MyDLP, OpenDLP	Amazon Macie	Azure Information Protection	Cloud Data Loss Prevention
Threat Detection	Snort, Kismet	Amazon GuardDuty	Azure Advanced Threat Protection	Event Threat Detection (beta)
Vulnerability Scanning	OpenVAS, Nessus	Amazon Inspector	Azure Security Center	Cloud Security Scanner
Secrets Management	Hashicorp Vault, Docker Secrets	AWS Secrets Manager	Azure Key Vault	Secrets management

Cultural Aspect

- Automation alone will not solve the problems
- Encourage security mindset especially if outside sec team
- Cultivate/Identify common goals for greater good
- Build allies (security champions) in company
- Focus on collaboration and inclusive culture
- Avoid Blame Game

Security team should try to eliminate the need of dedicated security team

Security Champion

- Bridge between Dev, Sec and Ops teams
- Single Person per team
- Everyone provided with similar cross skilling opportunities
- Incentivize other teams to collaborate with Sec team
  - Internal Bug bounties
  - Sponsor Interactions (Parties / get-togethers)
  - Sponsor cross skilling trainings for other teams

Security Enablers

People

- Build relationships between teams, don't isolate
- Identify, nurture security conscious individuals
- Empower Dev / ops to deliver better and faster and secure, instead of blocking.
- Focus on solutions instead of blaming

Process

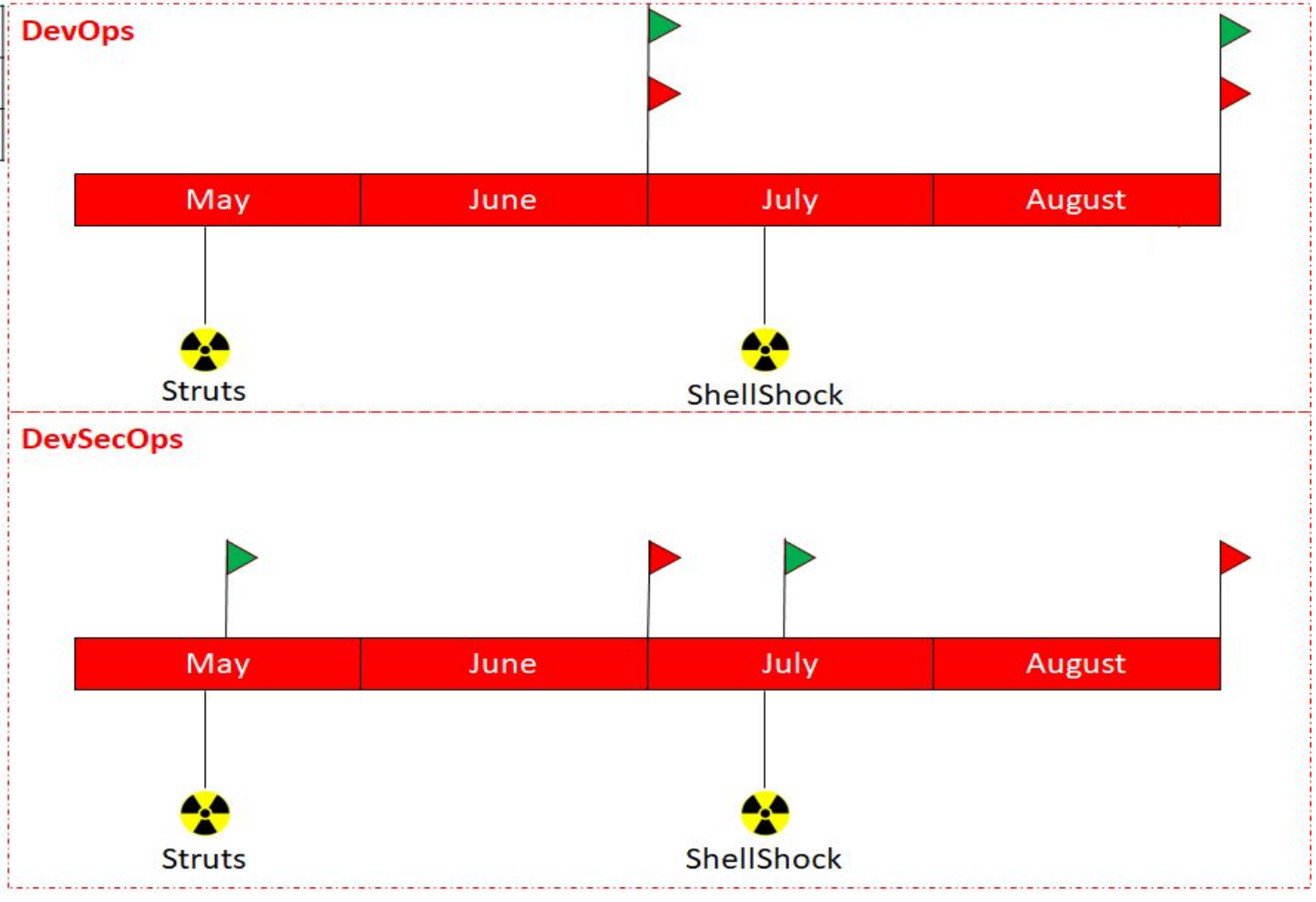
- Involve security from get-go (design or ideation phase)
- Fix by priority, don't attempt to fix it all
- Security Controls must be programmable and automated wherever possible
- DevSecOps Feedback process must be smooth and governed

Technology

- Templatize scripts/tools per language/platform
- Adopt security to devops flow don't expect others to adopt security
- Keep an eye out for simpler and better options and be pragmatic to test and use new tools

Generic Case Study

▶	Manual Pentest
☢	Zero Day
▶	Zero Day Resolved



Case Studies – Fannie Mae

DevSecOps @ Fannie Mae – The Strategy

Integrate with Culture

- Run as ONE (Security + DevOps as a singled purpose team)
- Training development teams to develop Secure code
  - OWASP Brown Bags and On Demand Training Courses
  - Secure Code Examples in GIT REPO show how to write secure code
- Empowering Developers/ Engaging Business Partners
  - Verification of Fortify "Clean Scans"
  - Periodic "To-the-Right" Application Static and Dynamic Tests

Make Security Easy

- Tracking security issues in the same systems developers are using
  - Integrated Fortify with SonarQube
  - Integrated Fortify with SSC
  - Application Security Issues Defect Tracking (Jira)
- Integrating preventive security controls/tools in the development phase
  - HP-Secure Assist
  - Find Security Bugs
  - Sonatype IQ Plugin

Automate Everything

- Automating as many security tests as possible to run alongside other tests
  - Integrating SAST tools (HP-SA, Find Bugs, Find Security Bugs, Fortify)
  - Future> Use DAST tool
- Detecting when applications are relying on libraries that have known vulnerabilities
  - Integrating Sonatype with fortify to detect third party libraries that have known vulnerabilities

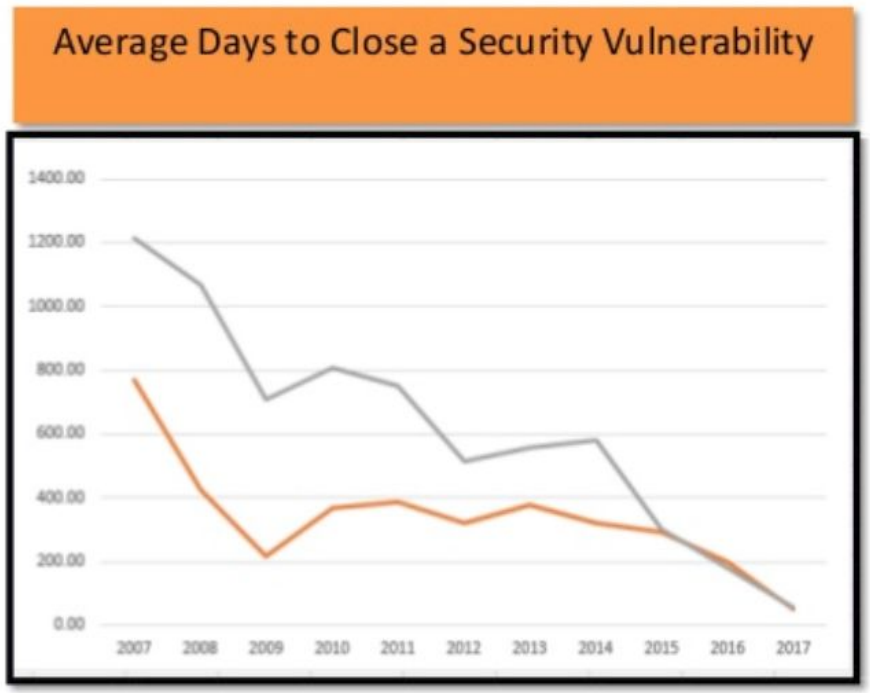
https://www.slideshare.net/DevSecOpsDays/fannie-mae-devsecops-journey-with-chitra-elango-and-john-willis

Case Studies – Fannie Mae

DevSecOps @ Fannie Mae – The Results

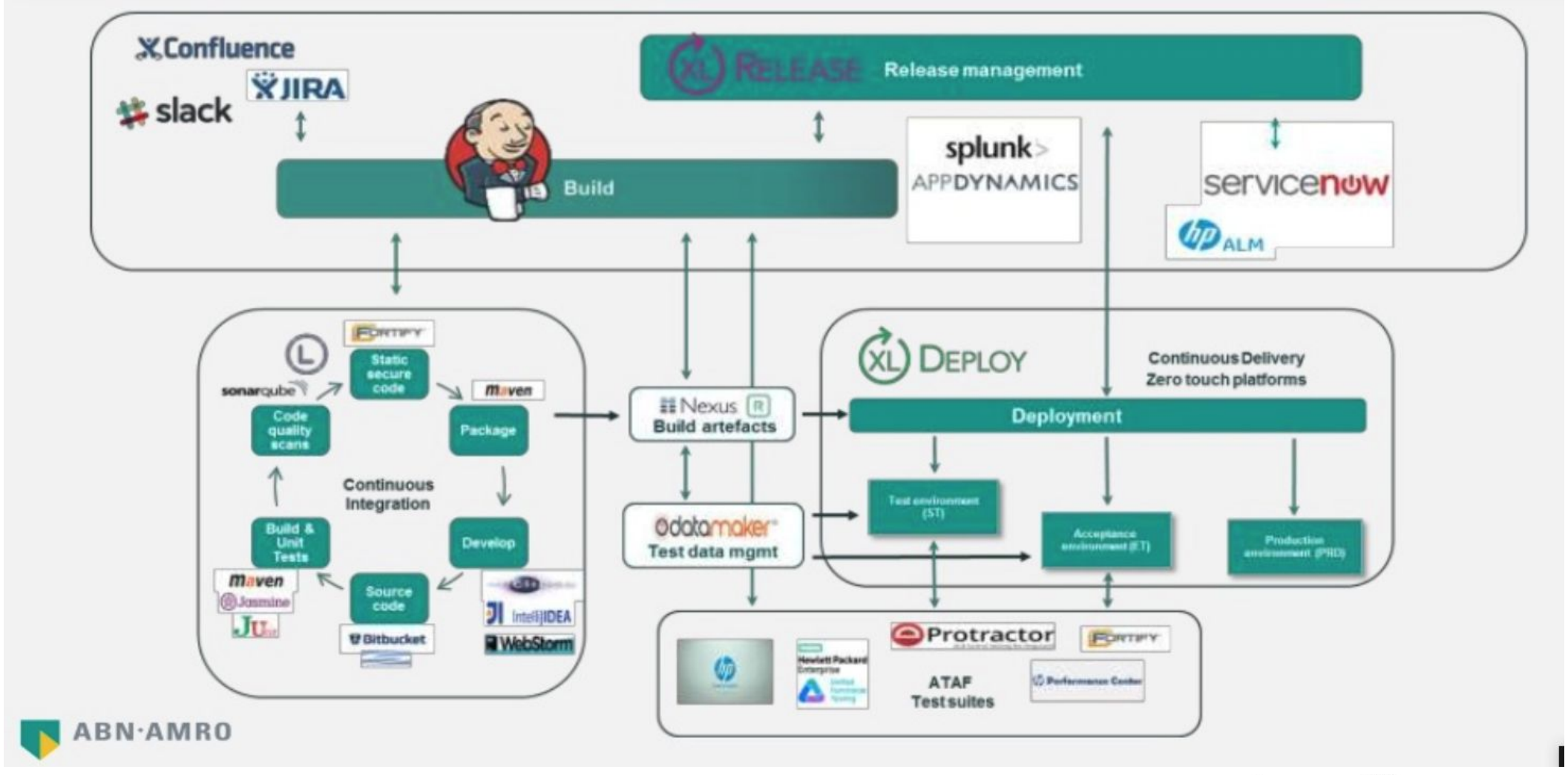
Delivering the Promise

- Average days to close a vulnerability improved by 74%
- Automated code quality scanning shows overall security code scores has increased by 10%
- More than 60% of application teams are performing security tests before release
- Critically vulnerable open source components (CVE 7.5+) downloaded has decreased from 18% to 6.25%
- ~ 55% of technical debt and security defects identified as a result of periodic testing have been dispositioned
- ~ 77% of older technical debt and security defects have been remediated, have a remediation plan in place, or have been addressed through managed retirements of assets



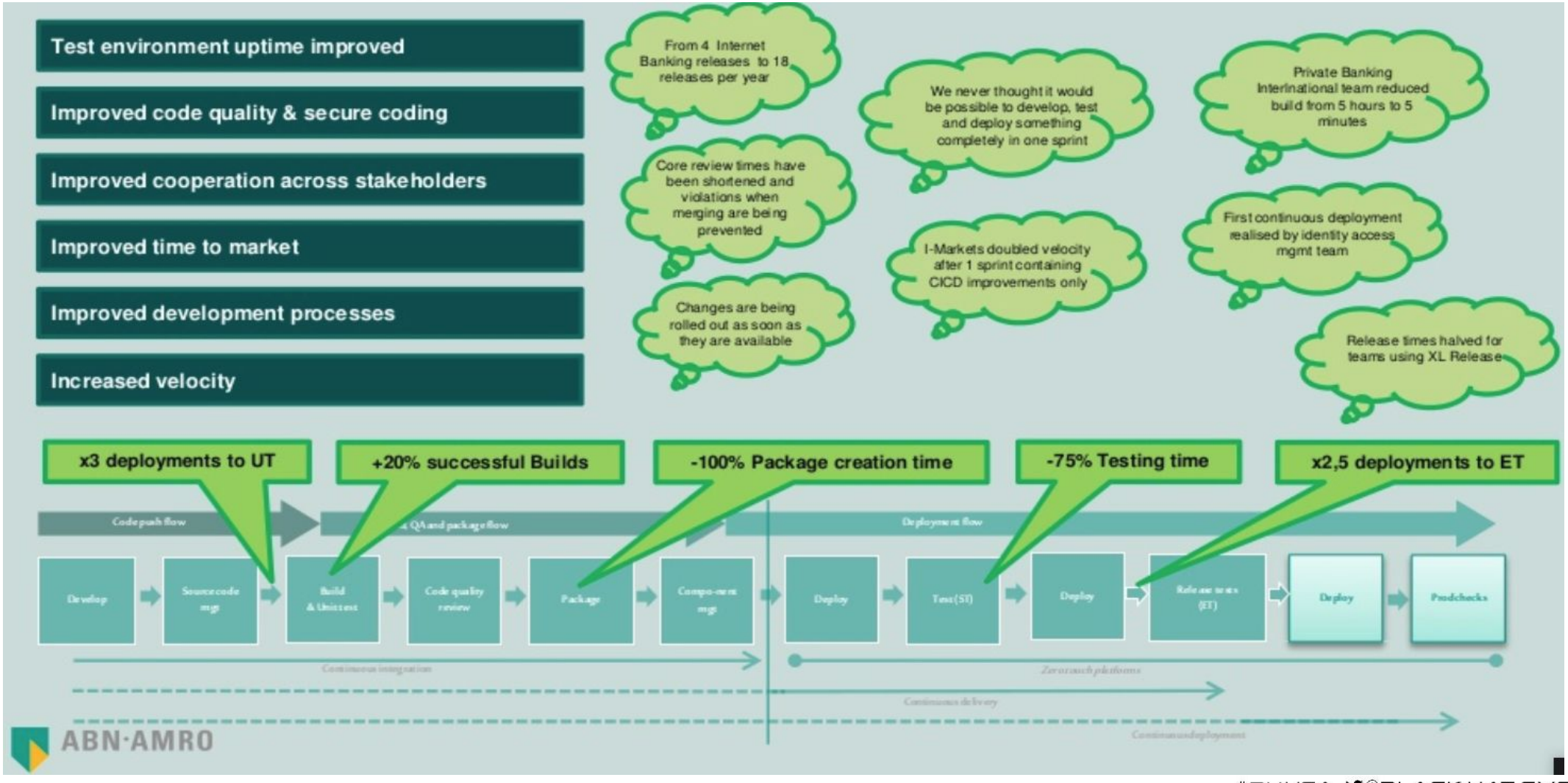
https://www.slideshare.net/DevSecOpsDays/fannie-mae-devsecops-journey-with-chitra-elango-and-john-willis

Case Studies – ABN Amro



https://www.slideshare.net/derweeksglobal/abn-amro-devsecops-journey

Case Studies – ABN Amro



https://www.slideshare.net/derweeksglobal/abn-amro-devsecops-journey

Negative Case Studies

bleepingcomputer.com/news/security/7-percent-of-all-amazon-s3-servers-are-exposed-explaining-recent-surge-of-data-leaks/

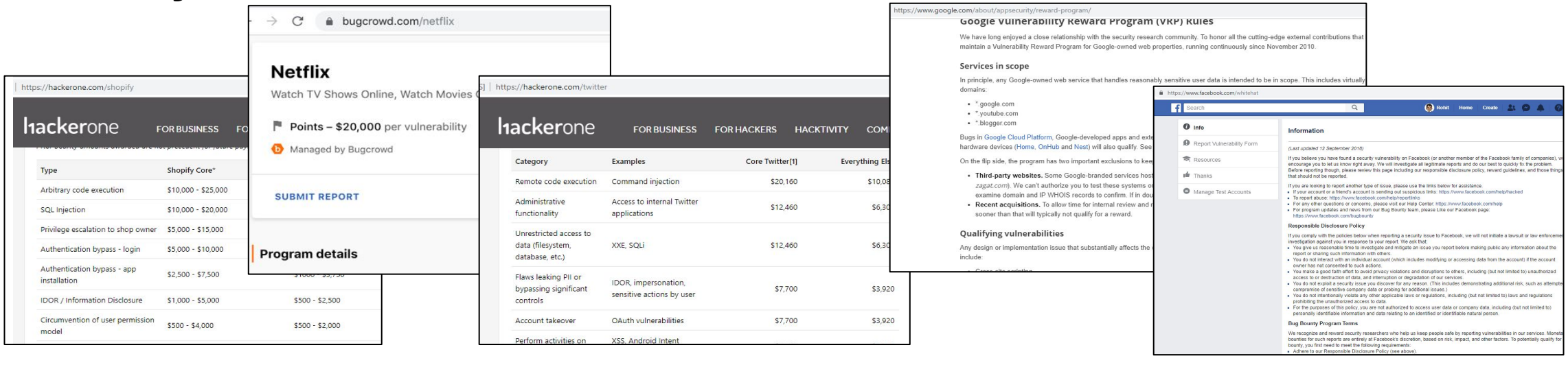
- Top defense contractor Booz Allen Hamilton leaks 60,000 files, including employee security credentials and passwords to a US government system.
- Verizon partner leaks personal records of over 14 million Verizon customers, including names, addresses, account details, and for some victims — account PINs.
- An AWS S3 server leaked the personal details of WWE fans who registered on the company's sites. 3,065,805 users were exposed.
- Another AWS S3 bucket leaked the personal details of over 198 million American voters. The database contained information from three data mining companies known to be associated with the Republican Party.
- Another S3 database left exposed online leaked the personal details of job applications that had Top Secret government clearance.
- Dow Jones, the parent company of the Wall Street Journal, leaked the personal details of 2.2 million customers.
- Omaha-based voting machine firm Election Systems & Software (ES&S) left a database exposed online that contained the personal records of 1.8 million Chicago voters.
- Security researchers discovered a Verizon AWS S3 bucket containing over 100 MB of data about the company's internal system named Distributed Vision Services (DVS), used for billing operations.
- An auto-tracking company leaked over a half of a million records with logins/passwords, emails, VIN (vehicle identification number), IMEI numbers of GPS devices and other data that is collected on their devices, customers and auto dealerships.

Cloud Assets Misconfiguration

Prevention: Continuous monitoring and review of cloud assets and config

Is it Enough?

- Rite of passage by periodic pen test and continuous bug bounty
- It's not just important to get feedback but to also action on them
- Risk Acceptance Documentation should be the worst case scenario not your first bet



Who Watches the Watcher

- Did we secure the security controls
- DevSecOops: If attacker controls security tools / build chain It has limitless power
- Ensure the same practice is followed back again for these tools
- Security role doesn't means you get to circumvent the rules
- Follow basic security hygiene we always keep talking about
  - Secure configuration
  - Patching Policy

References

- https://www.blackhat.com/docs/us-17/thursday/us-17-Lackey-Practical%20Tips-for-Defending-Web-Applications-in-the-Age-of-DevOps.pdf
- https://www.sonatype.com/hubfs/2018%20State%20of%20the%20Software%20Supply%20Chain%20Report.pdf
- https://snyk.io/opensourcesecurity-2019/
- https://scaling-threat-detection.awssecworkshops.com/
- https://www.veracode.com/state-of-software-security-report

Key Takeaways

- Security is everyone responsibility
- Embrace security as an integral part of the process, use feedback to refine the process
- DevSecOps is not a one size fit all: your mileage will vary

NOT SO SECURE
A Claranet Group Company

@anantshri
anant@notsosecure.com