Roadmap to DevSecOps Adoption @ EPA
April 2020
Presentation Agenda
Highlights of the DevSecOps Paper including strategies for near term implementation.

01 Current State
Interviews with EPA Product Teams

02 Common Takeaways
Requirements, Processes, and Systems’ Needs

03 Future State
Frictionless DevSecOps Platform using Containers for Product Teams

04 Realize the Teams through Alignment
Create alignment of the new teams via platform, technologies, process and culture

05 Next Steps
Learn by Doing as Agile teams - Incrementally building Proof of Concepts
What is DevOps?

DevOps (bringing Software Development and Operations together) is a cultural mind set that uses tools and processes to foster two main goals:

1. Continual experimentation in short cycles, taking risks, and learning from failure
2. Understanding that repetition and practice is the prerequisite to mastery

When security is added into the lifecycle it is known as DevSecOps.

A Deployment is the installation of a version of software to an environment (e.g., promoting a new build to production).
Engaging The Product Teams

Understanding the current state of delivery and operations at EPA, including the Product Teams’ perspectives and their needs to support delivery requirements.

Interviews with Product Teams

- 9 EPA teams interviewed comprised of multiple, cross-functional, small agile teams
- Mature in Agile and in Continuous Integration and Continuous Delivery (CI/CD) skills
- All have:
  - Pipelines in AWS or cloud.gov
  - Similar tech stack and open source tools
  - Similar deployment, testing and security processes
  - Similar roles and responsibilities distributed between EPA and contractor team members

Operation Concerns

- Address security scanning much earlier in the process, which is the nature of DevSecOps
- Reduce the manual processes that are involved with the deployment of the artifacts
- Dependency on specialized skills to perform the deployment impedes deployments

Physical and Policy Impediments

- No sandbox testing environments for external users that mimics the CDX data flow to NCC
- EPA policies and firewall rules are cost prohibitive to create secondary data flows for users for testing new features
- Staging environments do not match production environments - user experience for testing is not ideal
02 Common Takeaways

What the Platform Teams Want

**Deliver Software Quicker, Reliably and Securely:**
- Cloud platforms and DevOps environment that match the demands of customer requirements
- Deliver on demand
- Separate deployment from release
- Do blue/green releases

**Build a Community of Learning:**
- Spread learning through, dojos coaching and training
- Communities of Practice and Proofs of Concepts

**Cloud is a Differentiator:**
- Scale pipelines and create predictable delivery

**Replace the ADC Process with an Automated Deployment Process:**
- Want a higher predictability in delivery
- Greater transparency
- More frequent releases (Daily/Weekly)
-ACKnowledged the necessity of appropriate control gates

**Autonomy:**
- Create and manage own environments, tools, pipelines
- Schedule their own releases

DevOps does not Scale Easily

**Applications have Dependencies:**
- Multiple applications running on the same machine share libraries and components
- Difficult to upgrade software due to conflicts
- Hard to isolate without impacting other services
- VMs do not solve issue as OS being upgraded is shared

**CI/CD does not scale well due to complexity:**
- Tightly coupled applications use various versions of databases, webservers and OS
- Environments get out of sync; ops team must patch and maintain
- Abundance of SMEs needed to maintain
- Struggle to stay ahead of the security curve for all these new technologies

**Security is a late concern:**
- Latency with reviewing the apps for security vulnerabilities
- High risk of discovering issues or false positives - must be negotiated, corrected, rebuilt, and rescanned.

Product Teams want DevOps

However
Pipelines, Containers and the Software Factory

Key Features

Autonomous Product Teams
- Plan Releases using Agile Tools
- Deploy everything inside containers (excepts database)
- Ensure consistency across all environments
- Reduce/remove manual deployment
- Shift responsibility of tech stack to Product Team
- Platform Ops team is responsible for everything outside of the container – i.e. the host Ecosystem

Team Needed

- Product Teams use tools, processes, containers and pipelines to manage all aspects of DevSecOps, including security
- Platform Teams provide any and all services to assist the Product Team
- Enabling Teams provide special services and capabilities as needed for the Product Teams
- Complex Sub-System Teams provides deep technical support for creating the environment
Product Teams using DevSecOp Pipelines

Future State

Legend
- **UAT Testing**
- **DAST**
- **Security Testing**
- **Control Gates**
- **Testing**
- **Environments**

03

Development
- Build Application
- Software Component Analysis (SCA)
- Static Application Security Test (SAST)
- Unit Test and Automated Tests
- Publish Artifacts
- Deploy to Test
- Dynamic Application Security Testing (DAST)
- Deploy to Stage
- Aggregate Vulnerabilities Report Tool

Test
- Build Application
- Software Component Analysis (SCA)
- Static Application Security Test (SAST)
- Unit Test and Automated Tests
- Publish Artifacts
- Deploy to Test
- Dynamic Application Security Testing (DAST)
- Deploy to Stage

Stage
- Release Decision
- Nonfunctional Testing
- UAT Testing
- Release Decision

Production
- Release to Prod
- DAST/VA penetration testing (ongoing)
<table>
<thead>
<tr>
<th>Activity/Practice</th>
<th>Phase</th>
<th>Why?</th>
<th>Frequency</th>
<th>Tool(s)</th>
<th>Service Provider</th>
<th>Service Consumer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Plan features and capabilities for a release</td>
<td>Plan</td>
<td>Release Management</td>
<td>Every 2 weeks</td>
<td>JIRA, Confluence Slack</td>
<td>Enabling Team</td>
<td>Product Team</td>
</tr>
<tr>
<td>SCA – Software Component Analysis</td>
<td>Develop</td>
<td>Identify Dependencies</td>
<td>As needed for a release</td>
<td>Sonatype – Repository Manager, Nexus IQ</td>
<td>Platform Team</td>
<td>Product Team</td>
</tr>
<tr>
<td>Container Selection</td>
<td>Develop</td>
<td>Appropriate container base image for application</td>
<td>At start of new feature or product</td>
<td>Docker EPA Centralized Artifact Registry (EPA CAR) (hosted in GitLab)</td>
<td>Enterprise Services</td>
<td>Product Team</td>
</tr>
<tr>
<td>SAST – Complete code base check of integrated code</td>
<td>Build</td>
<td>Code Vulnerabilities</td>
<td>Daily/ weekly</td>
<td>Contrast Security</td>
<td>Product Team</td>
<td>Product Team</td>
</tr>
<tr>
<td>Container Image Scan</td>
<td>Test</td>
<td>Conduct container image scan OS-check</td>
<td>On Deploy</td>
<td>Twistlock</td>
<td>Platform Team</td>
<td>Platform Team</td>
</tr>
<tr>
<td>Aggregate Vulnerabilities Report</td>
<td>Release</td>
<td>Provide findings from various scanning tools</td>
<td>On Deploy</td>
<td>Code DX</td>
<td>Platform Team</td>
<td>Product Team</td>
</tr>
<tr>
<td>Activity/Practice</td>
<td>Phase</td>
<td>Why?</td>
<td>Frequency</td>
<td>Tool(s)</td>
<td>Service Provider</td>
<td>Service Consumer</td>
</tr>
<tr>
<td>-----------------------------------</td>
<td>-----------</td>
<td>----------------------------------------------------------------------</td>
<td>-----------------</td>
<td>-----------------------------------------------</td>
<td>--------------------------------------</td>
<td>------------------</td>
</tr>
<tr>
<td>Complete System DAST</td>
<td>Deploy</td>
<td>Onboarding to Host Ecosystem</td>
<td>Once per system release</td>
<td>Sonatype – Nexus IQ</td>
<td>Platform Ops</td>
<td>Platform Teams</td>
</tr>
<tr>
<td>Track and visualize metrics</td>
<td>Operate</td>
<td>Understand the metrics of the data</td>
<td>Ongoing</td>
<td>Grafana</td>
<td>Production Ops</td>
<td>Product Teams</td>
</tr>
<tr>
<td>Production Scanning and Assessment (DAST/VA)</td>
<td>Monitor</td>
<td>Application Security Assessment</td>
<td>Continuous or as required by security team</td>
<td>Contrast Security - Protect</td>
<td>Cloud Application Security Team</td>
<td>Production Ops</td>
</tr>
<tr>
<td>Monitor Events</td>
<td>Monitor</td>
<td>Event alerts that integrates with Grafana</td>
<td>Ongoing</td>
<td>Prometheus</td>
<td>Production Ops</td>
<td>Product Teams</td>
</tr>
<tr>
<td>CI/CD Pipeline Management</td>
<td>Software Factory</td>
<td>Automate the deployment of containers with integrated security</td>
<td>Daily</td>
<td>GitLab</td>
<td>Platform Team</td>
<td>Product Team</td>
</tr>
<tr>
<td>Container Orchestration</td>
<td>Release</td>
<td>Manage the release, management, scaling, networking, and availability of container-based applications.</td>
<td>Daily</td>
<td>AWS EKS or Rancher</td>
<td>Platform Team</td>
<td>Platform Team</td>
</tr>
<tr>
<td>04 Realize the Teams through Alignment</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Frictionless Support Teams</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Product Teams (Developers, Testers)</strong></td>
</tr>
<tr>
<td>• Aligned to a single, valuable stream of work, such as a product, service or a set of features</td>
</tr>
<tr>
<td>• Empowered to design, build and deliver customer value as quickly and securely as possible without having to handoff work to other teams</td>
</tr>
<tr>
<td>• Have complete autonomy over the deployment of their products in any environment (dev, test, stage, prod)</td>
</tr>
<tr>
<td><strong>Production Ops Team (Infrastructure, Hosting, Services)</strong></td>
</tr>
<tr>
<td>• Supports release, operate, and monitor phases of the DevSecOps lifecycle</td>
</tr>
<tr>
<td>• Goal is to make the autonomous Product Teams make use of the platforms to deliver features at a higher pace with reduced coordination and little friction</td>
</tr>
<tr>
<td>• Ensure the host ecosystem is secure through Container Orchestration Management and Deployment operations</td>
</tr>
<tr>
<td><strong>Enabling Team (Cloud Service Reps, Assisted Services)</strong></td>
</tr>
<tr>
<td>• Onboarding and setup new Cloud accounts for Product Teams</td>
</tr>
<tr>
<td>• Provide special services and capabilities as needed for the Product Teams</td>
</tr>
<tr>
<td>• Team of cloud representatives that provide assisted services that help Product Teams get code through the pipeline process.</td>
</tr>
<tr>
<td><strong>Complex subsystem Team (New services that need to be figured out)</strong></td>
</tr>
<tr>
<td>• Deep technical support for creating and maintaining the hosted ecosystem.</td>
</tr>
<tr>
<td>• AI, Machine Learning, Analytics</td>
</tr>
</tbody>
</table>
Form Agile Teams to Work on Proof of Concepts (POC) and Deploy Continuously

Iterate the Platform, Techniques, Process and Culture

Identify Platform Team
• Identify startup Platform Ops Team
• Select Product Team for Proof of Concept (POC).
• Define performance metrics for outcomes

Identify Technologies
• Create Thinnest Viable Platform (TVP) to support the POC
• Prioritize the creation of Continuous Application Delivery Pipeline
• Deploy tech to run as an Agile DevOps process

Identify Platform Services and Process
• Deploy new services continuously
• Separate ops concerns in the platform by roles and tasks
• Iterate on services needed
• Define SLAs

Define Platform Culture
• Share learning and spread practice
• Actively practice retrospectives
• Experiment with ideas, share in failure, celebrate team success
• Market the platform

Continuously Delivering Products and Platform Ops Teams

Continuous Exploration
Continuous Integration
Continuous Deploy
Questions?