INTERNET2®

2016 TECHNOLOGY exchange

MIAMI FL

SEPTEMBER 25–28

RUGGED DEVOPS

Sara Jeanes and Nick Lewis

Program Managers, Internet2

Rugged DevOps

CONTENTS

- What is Rugged DevOps?
- Key Concepts
- The 3 Steps of Rugged DevOps
- A Few Examples

First, a couple questions for you

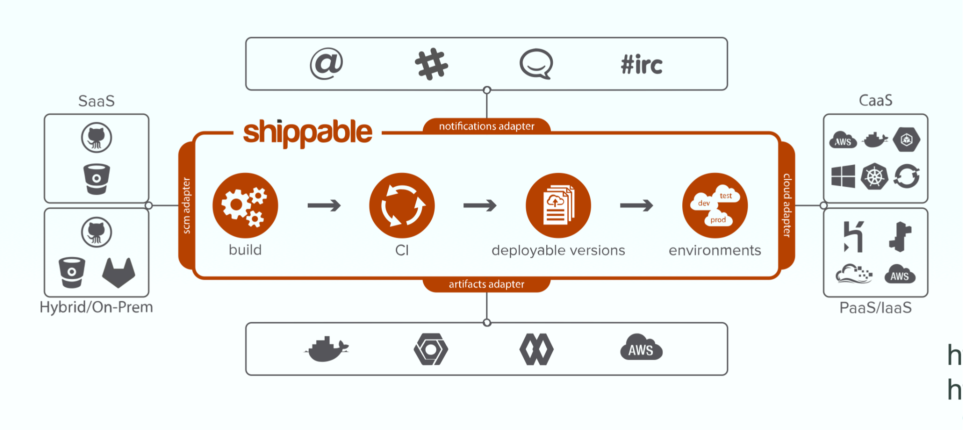
- Who is using DevOps on their campuses?
- Where are you using DevOps?
- If you’re not using DevOps, why not?
- Any good stories?

What is Rugged DevOps?

- Rugged + DevOps
- Rugged is
  - Durability
  - Visibility
  - Awareness
- Security as part of the application development and deployment pipeline, not a bolt-on
- Secure Mindset as Default

The DevOps Pipeline

– "DevOps is the practice of operations and development engineers participating together in the entire service lifecycle, from design through the development process to production support.”



– Continuous Integration

– Continuous Delivery

– Continuous Deployment

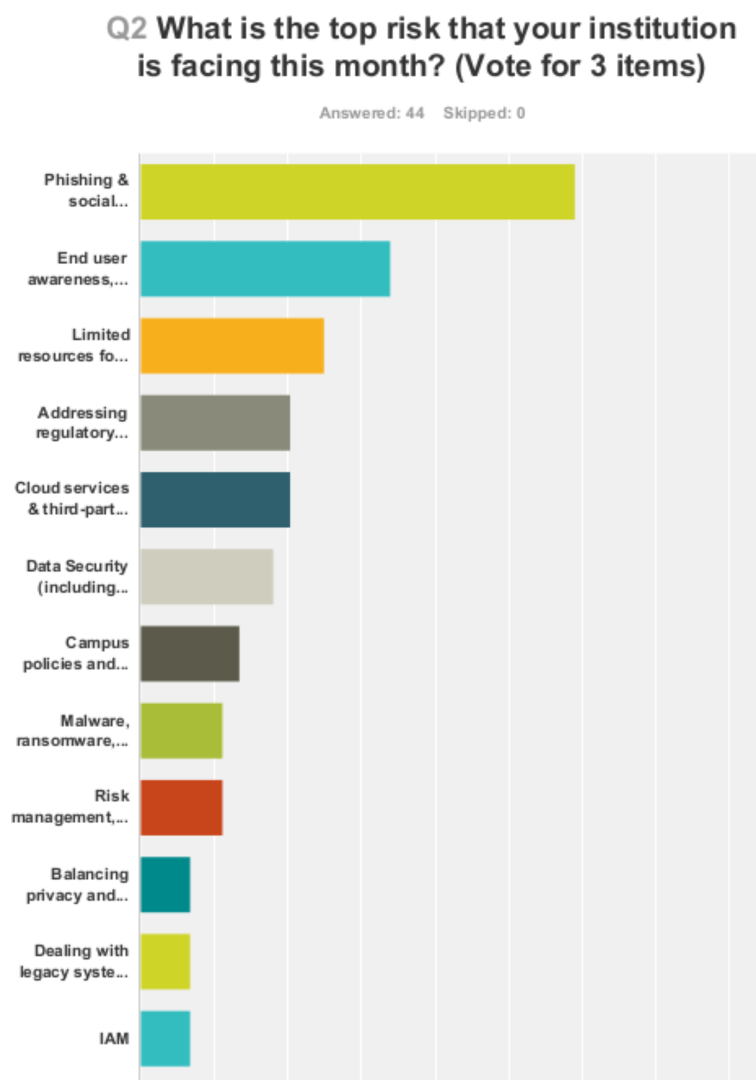
https://theagileadmin.com/what-is-devops
https://app.shippable.com/product.html

Information Security Priorities

Source: HEISC Risk Survey from 9/21/2016

What is the top risk that your institution is facing this month? (Vote for 3 items)

#3 - Limited resources for the security program (too much work, not enough time or people) – 25%



Q2 What is the top risk that your institution is facing this month? (Vote for 3 items)

Answered: 44 Skipped: 0

Why is DevOps important to information security?

- Continuous improvement is core to information security
- Traditional security approach is not meeting the needs of our enterprises and we still have data breaches
  - We will still have data breaches using DevOps
- Information Security is integrated throughout theDevOps pipelines
- This is still fairly new in many environments and getting involved early will ensure security is involved

Industry Reports on DevOps including Security

- Puppet Labs - 2016 State of DevOps Report
  - High performers spend 50 percent less time remediating security issues than low performers. By better integrating information security objectives into daily work, teams achieve higher levels of IT performance and build more secure systems.
  - https://puppet.com/system/files/2016-06/2016%20State%20of%20DevOps%20Report_0.pdf

- Securosis - Security goes from being "Dr. No" to just another set of tests which help validate code quality.
  - https://securosis.com/assets/library/reports/Security_Into_DevOps_Final.pdf

Rugged DevOps – Where is information security?

- Information Security is integrated throughout the DevOps pipelines
- You can train developers or IT staff to represent the Information Security office
  - They can contact you when they have a question or when they need assistance
- Integrate security checks into the pipelines so projects are checked for security as they get go through the pipeline
  - Example: Docker Security Scanning - https://blog.docker.com/2016/05/docker-security-scanning/
  - Other security scanning including static/dynamic code analysis, vulnerability scans, etc

3 Steps of Rugged DevOps

- Shift Left
  - “Focus on quality, work on prevention instead of detection, and begin testing earlier than ever before.”
- Think Micro
  - Serverless architectures and microservices can have a smaller attack surface
- Use the Same Language
  - Cultural security. Developers, SysOps, and SecOps using the same terms in one holistic process/view.

http://devops.com/2016/08/18/3-simple-steps-rugged-devops-101/
https://www.ibm.com/developerworks/community/blogs/rqtm/entry/what_is_shift_left_testing

What is Shift Left in Security?

- Integrating security early into the development and design of software or system, rather than retrofitting security in at the end
- Performing security testing early and preventing security issues. Potentially before someone external finds the issue!
- Test data to use during the testing phases
  - De-identify this data! Maybe put some malformed data in to see what happens

Misconceptions of DevOps

- It’s only for software development
- It’s just Dev and Ops – everyone else is not engaged
- Just integrate tools and we’re done
- DevOps won’t fit our culture
- It’s everything, everywhere!
- Magic fairy dust and profit

DevOps is too fast. Security is too slow.

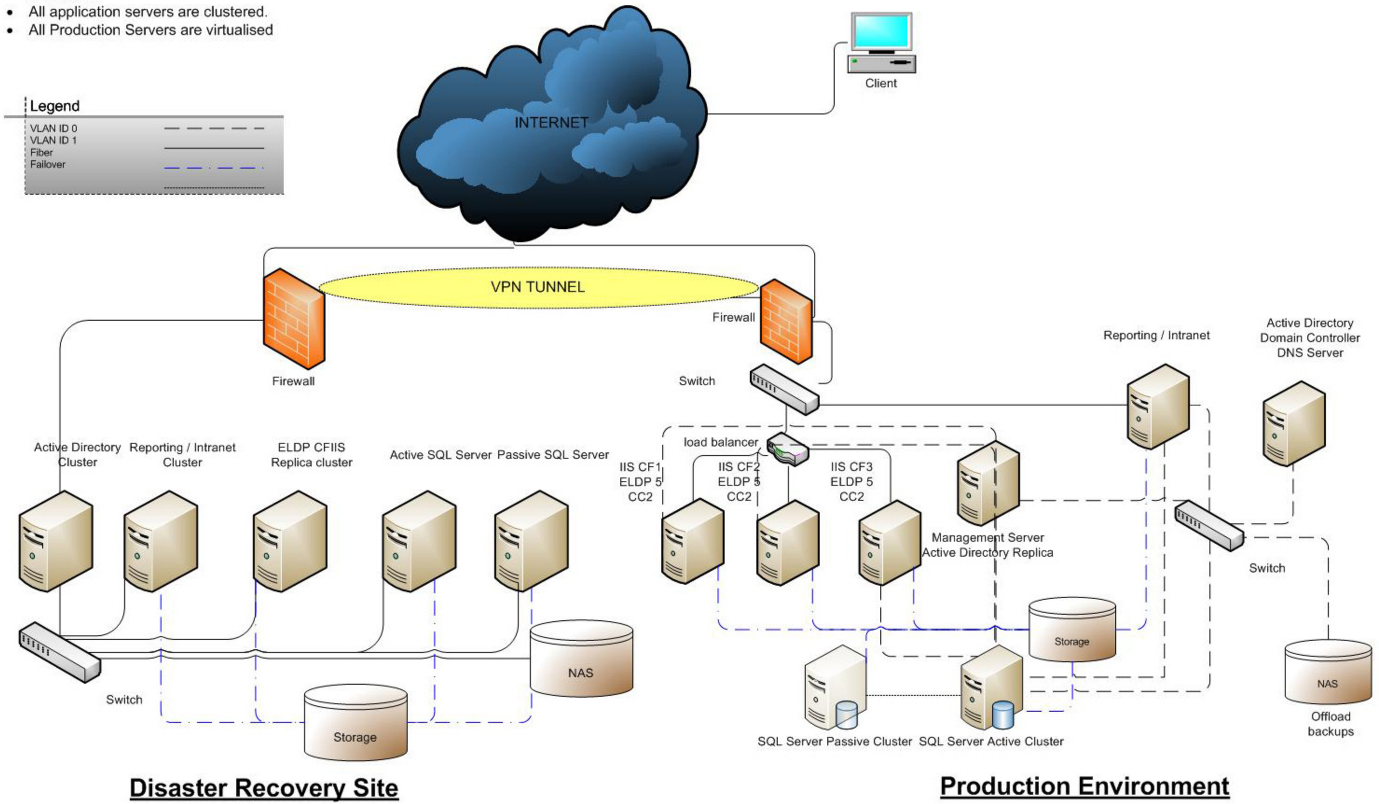
- Attackers are faster and more nimble than you
- How do you increase the speed of response?
- Have metrics and logging to use in incident response
- Some security checks do take a long time
  - Identify which checks take take an unacceptably long time and identify why
  - Potentially run those checks in a dedicated security environment to incorporate data back into DevOps processes

Security as a component of DevOps

- Security is integrated into Rugged DevOps in many different ways
- Focus is on process improvement where security, compliance, and privacy are requirements
- Use metrics to understand where you are spending your time and then figure out which time buckets can be reduced to save time
- Tools like Chef, Puppet and Ansible not only automate the building of infrastructure, but they also are used to enforce security policies. Scripts may pull certified ‘golden masters’, or they can automatically assemble secure application stacks by updating, patching, configuring and self-validating. – Securosis
- Test environments by default!

Example – Patching an ERP System

- Multiple VLANS are deployed and passed through the firewall clusters to enhance security.
- Only showing 2 VLANS here for the sake of simplicity
- All hardware is designed to be fully redundant.
- All application servers are clustered.
- All Production Servers are virtualised



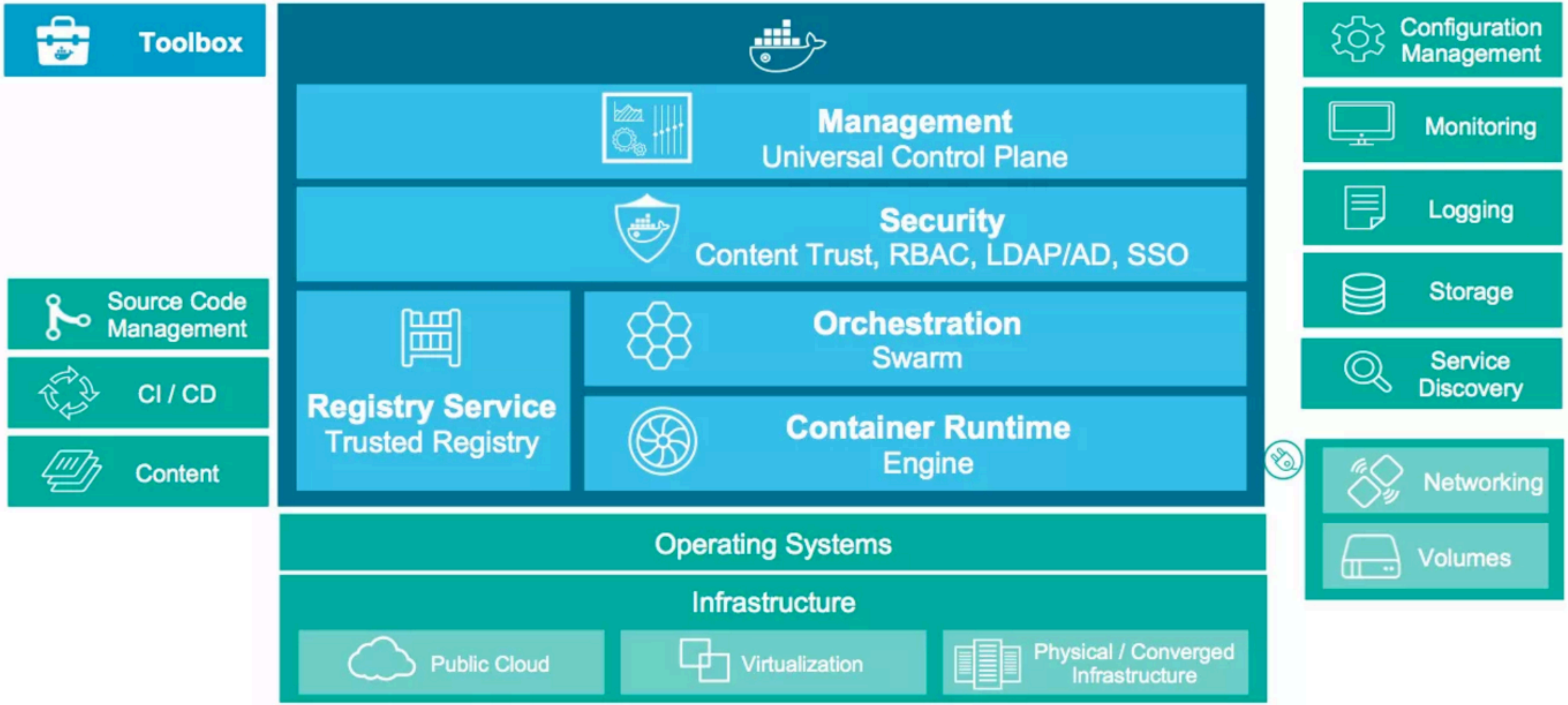
Source: https://www.skillcast.com/images/skillcastwebimg/Dataarchitecture.jpg

Example – Integrating vulnerability scanning into DevOps

- Traditionally vulnerability scanning is done by the security team on a periodic basis like every quarter, etc
  - Then distribute results to the proper group to respond to the findings
  - Sometimes these reports are long and contain findings that are low risk
- In a DevOps environment, running a vulnerability scan every quarter is too late. It needs to be done every iteration or update!
  - Prepare the team in advance including Ops, Dev, and Sec
  - Once the first couple times are done, it will get easier
  - Prioritize ruthlessly – start with the highs and work from there
  - Metrics – how long did it take to remediate the agreed upon vulnerabilities? How much effort?

Example – Distributed IT Self Service

- Container-as-a-Service model
  - Self Service Images of Common IT workloads
  - Framework for running Containers
  - RBAC
  - Version Control
- Docker Datacenter (for Example)



https://blog.docker.com/2016/02/docker-datacenter-caas/

How do I adopt DevOps on my campus?

- Talk to the other teams and see what they are working on
- Start small and build out
- Start in an environment with minimal security requirements
- Be willing to accept change
- Be willing to find new ways to accomplish the team goals
- Additional Motivators
  - More Secure by Default the Better
  - Professional Growth
  - Very Real Possibility of Cost Savings

Resources

- NET+ Services
  - Amazon Web Services by DLT
  - Splunk
  - (We welcome other suggestions)
- Other Related Services
  - Ansible, Shippable, Jenkins
- Articles
  - http://devops.com/2016/08/18/3-simple-steps-rugged-devops-101/
  - https://www.ibm.com/developerworks/community/blogs/rqtm/entry/what_is_shift_left_testing?lang=en
  - http://devops.com/2016/08/25/5-techniques-to-improve-software-hygiene/

Contact Us

- Sara Jeanes
  - sjeanes@internet2.edu
  - 🐦 @sarajeanes
- Nick Lewis
  - nlewis@internet2.edu
  - 🐦 @lewisnic
- Internet2.edu

INTERNET2®

2016 TECHNOLOGY exchange

MIAMI FL SEPTEMBER 25–28

RUGGED DEVOPS

Sara Jeanes and Nick Lewis

Program Managers, Internet2