Cloud Security and Microservices

Disclaimer

“ The information presented in these slides is borrowed from DevSecOps community forum and is protected by © DevSecOps Foundation 2015-2016 ”

Agenda

- What is DevSecOps ?
- Why is this important to me as a software Developer ?
- Security in CICD and Microservices ?

What's happening in the world ?

- DEVOPS
- PUBLIC CLOUD
- AGILE
- SCRUM
- LEAN
- LOW-CODE
- NO-CODE
- NO OPS
- ...

Region | City

India	100
Israel	99
Finland	58
Ukraine	57
Netherlands	56
Australia	52
Sweden	48

Who's doing Enterprise DevOps ?

What hinders Secure innovation ?

1. Manual processes & meeting culture
2. Point in time assessments
3. Friction for friction's sake
4. Contextual misunderstandings
5. Decisions being made outside of value creation
6. Late constraints and requirements
7. Big commitments, big teams, and big failures
8. Fear of failure, lack of learning
9. Lack of inspiration
10. Management and political interference (approvals, exceptions)

Say What ??!!

“ THIS IS THE END OF SECURITY AS WE KNOW IT…”

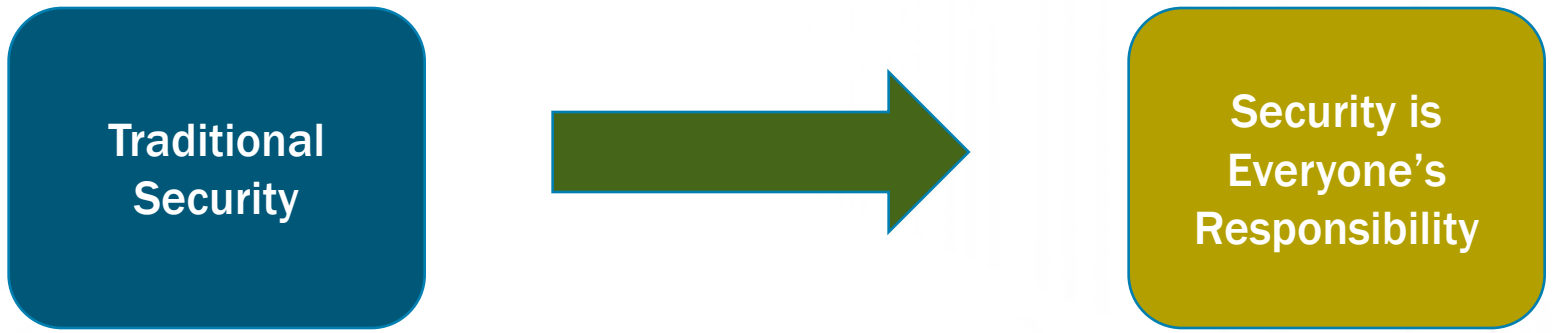
- **Joshua Corman**

(Director of the Cyber Statecraft Initiative at the Atlantic Council's)

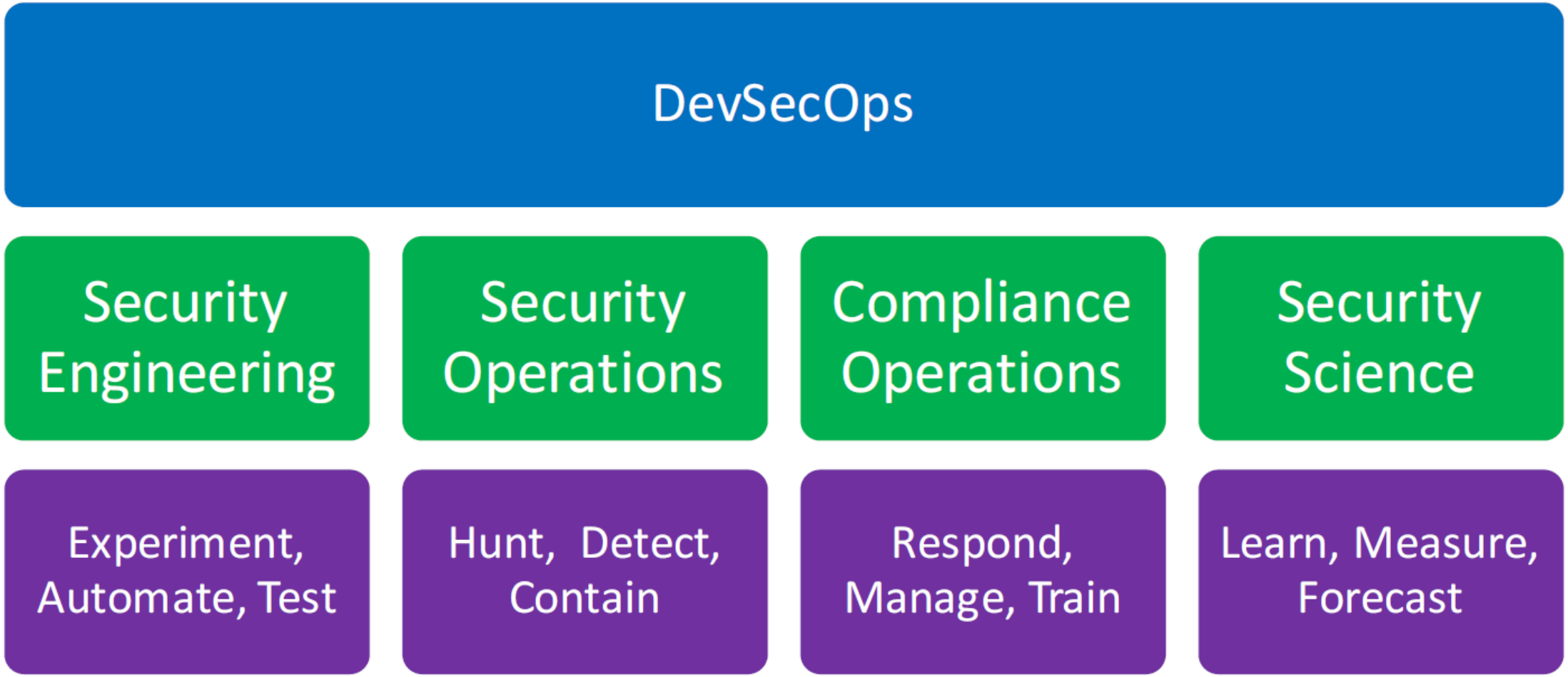
The Need for Change

- **Innovation** is a competitive advantage
- **Cloud** has leveled the playing field
- Demand for **Customer centric** product development
- Continuous delivery of features and changes
- New generation of workers desire collaboration
- **Speed and scale** are necessary to handle demand
- **Integration** over invention to speed up results
- Security breaches are on the rise
- People desire to work with **greater autonomy**...
- **Continuous Learning**... How can I do better? & better?

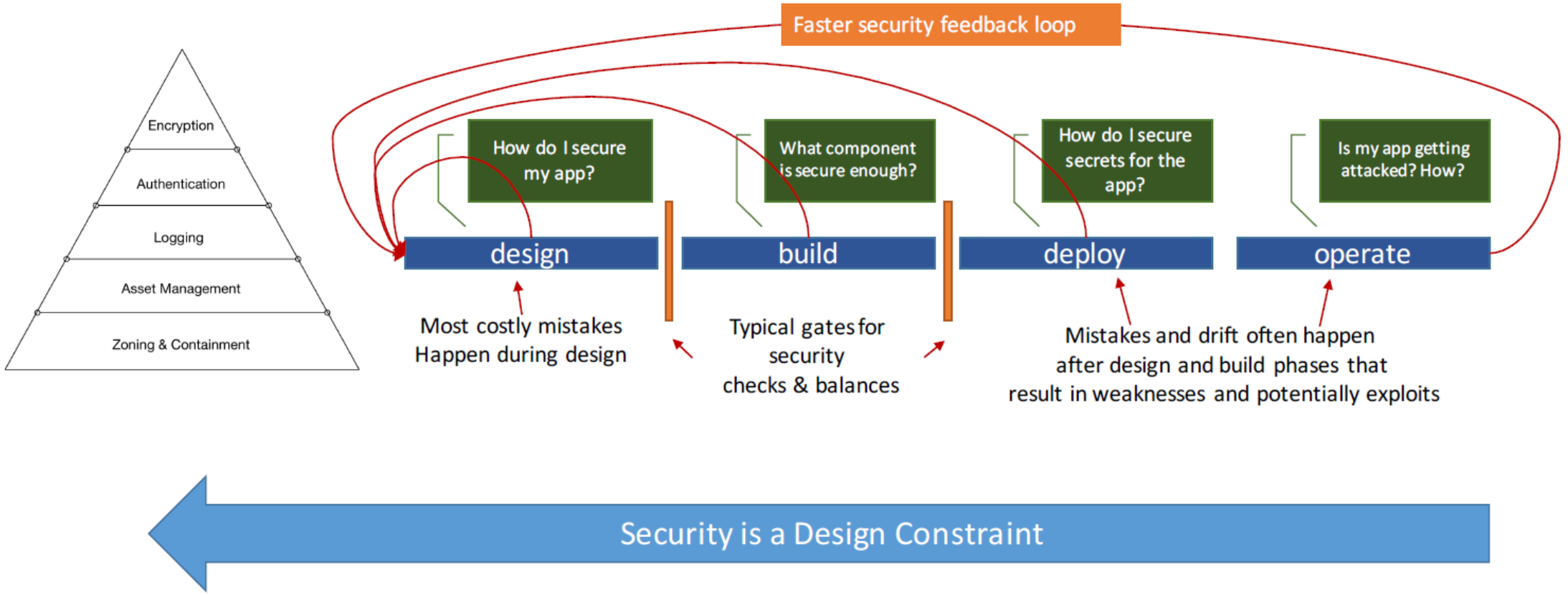
Culture Hacking



The Art of DevSecOps

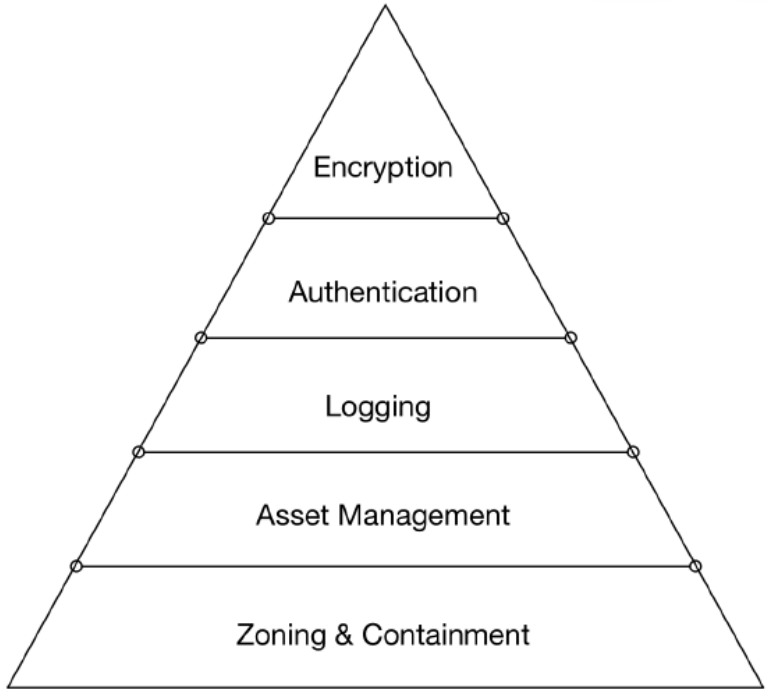


Shifting Security to the Left means built-in



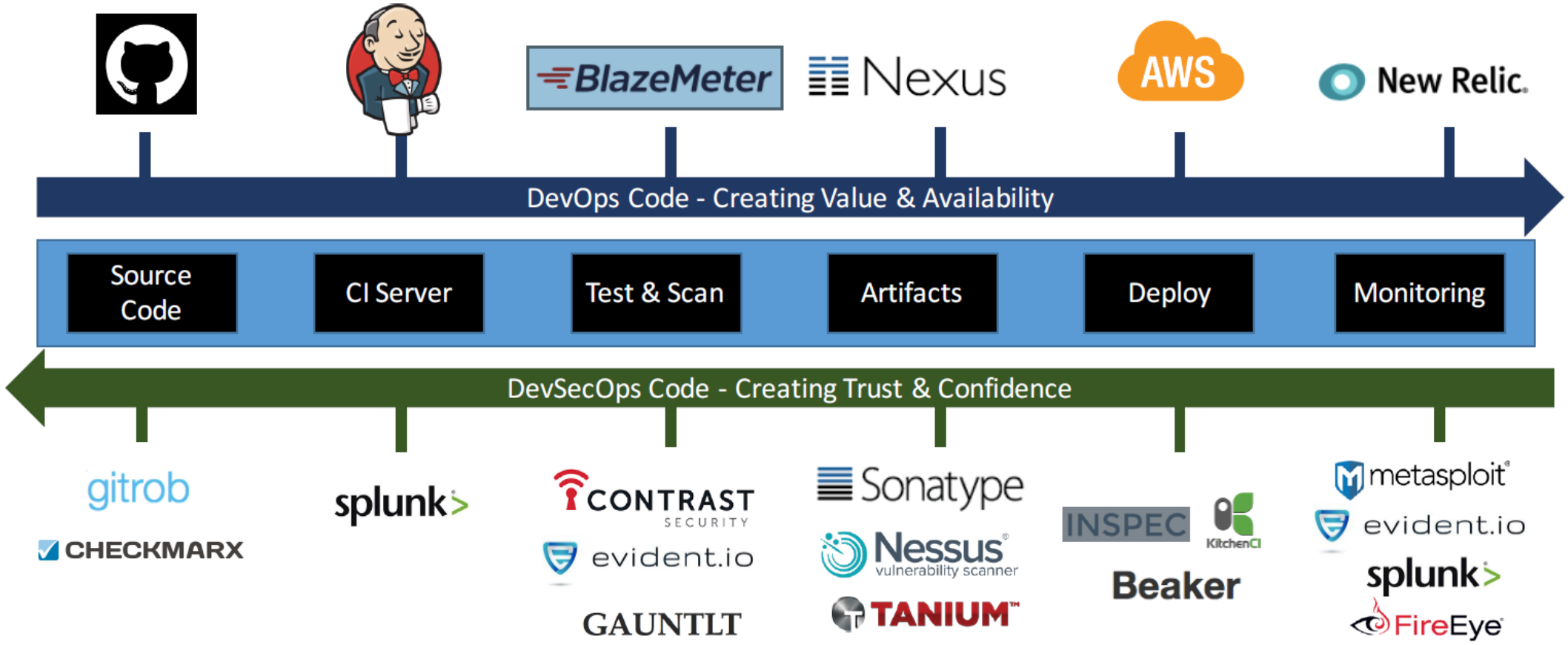
Security is and has always been a Design Constraint …

If you can remember 5 things, remember these ->

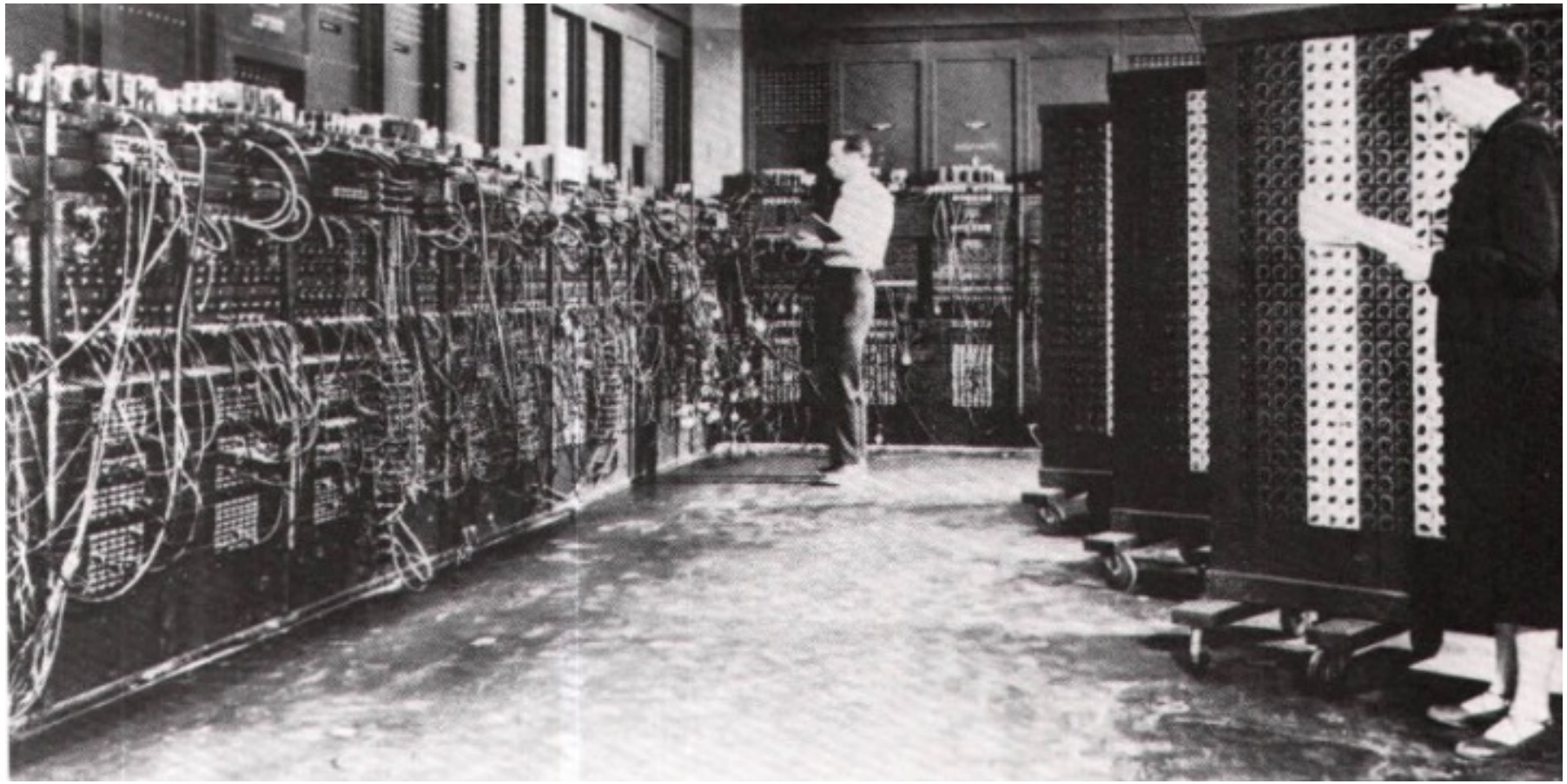


"Apps & data are as safe as where you put it, what's in it, how you inspect it, who talks to it, and how its protected…"

Example of Continuous Delivery + Security



Example of a Server Stack Two Generations back …



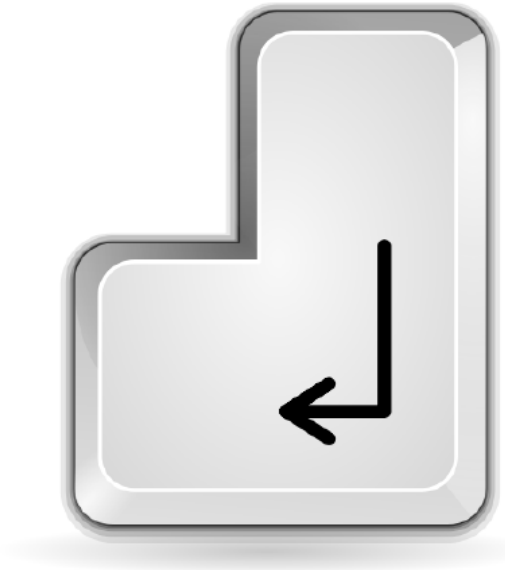
Glen Beck (background) and Betty Snyder (foreground) program ENIAC in BRL building 328. (U.S. Army photo)

Example of a Server Stack One Generations back …

Lawrence Livermore National Laboratory [Attribution], via Wikimedia Commons

How Stacks are managed Now...

ec2-run-instances ami-*12345678* -t t1.micro -k my-key-pair -g my-security-group



Software Defined Environment

- Virtualized/abstracted infrastructure managed by **software**, i.e. Configuration as Code:
  - Chef
  - Puppet
  - AWS CloudFormation
- The software being deployed **defines** the configuration and virtualized infrastructure requirements.
- The virtualized **infrastructure** extends past the data center to allow for multiple environments.

Benefits of SDEs

- Automatically adjusts to workload based on demand (autoscale)
- Centrally managed
- Everything as Code, underlying policies are code (JSON, YAML etc.)
- Better resource management
- Holistic overview of the environment
- Faster deployments
- Built-in audit trails and API endpoints

Speed

Ease

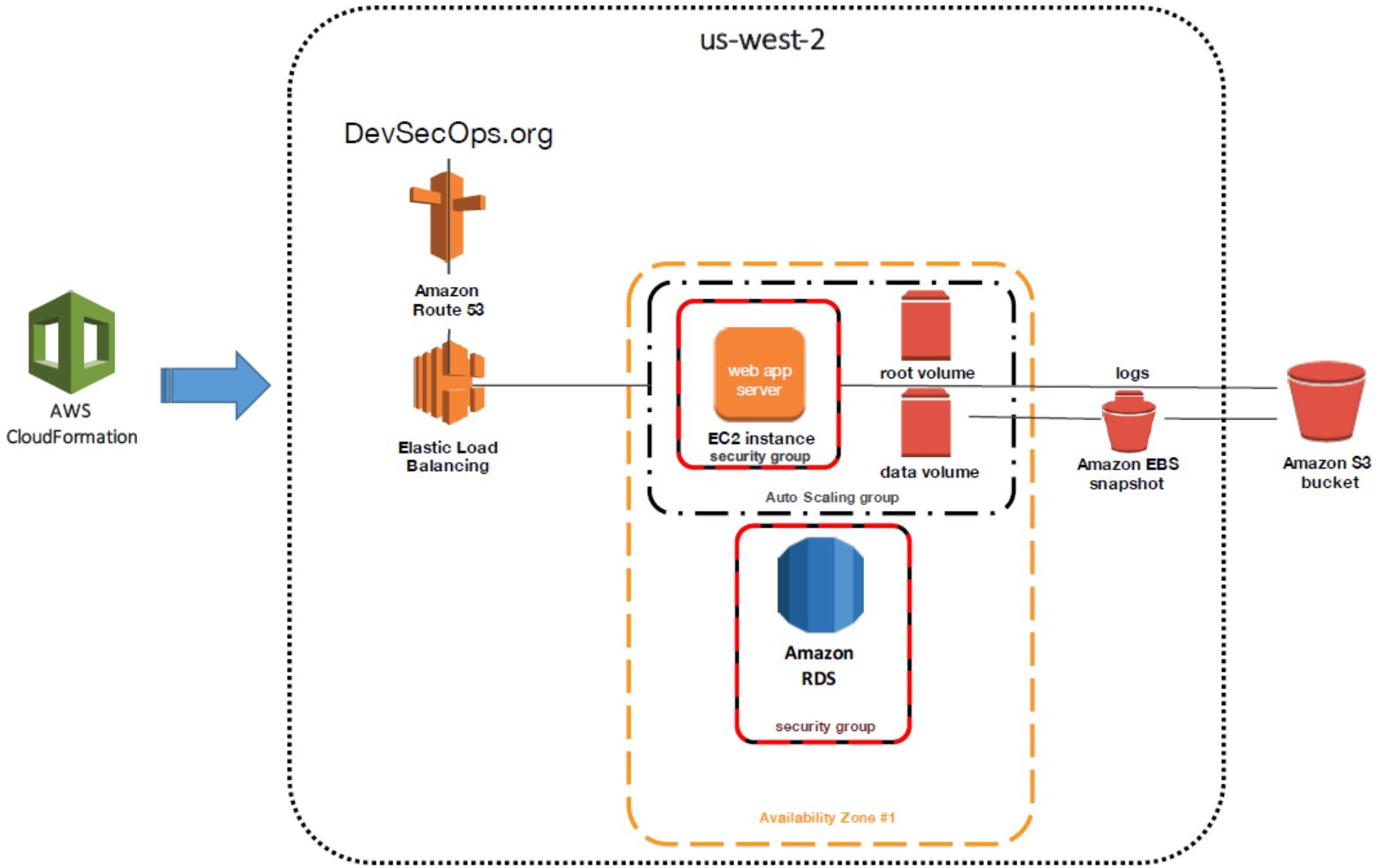
Security

Example for SDE

- Networking
- Programming
- Application
- Security
- Operating System
- Micro-services

```
"AWSTemplateFormatVersion" : "2010-09-09",

"Description" : "AWS CloudFormation Sample Template AutoScalingMultiAZWithNotifications"

"Parameters" : {
  "InstanceType" : {
    "Description" : "WebServer EC2 instance type",
    "Type" : "String",
    "Default" : "t2.small",
    "AllowedValues" : [  "m1.small", "m1.medium", "m1.large"]
,
    "ConstraintDescription" : "must be a valid EC2 instance type."
  },

  "OperatorEMail": {
    "Description": "EMail address to notify if there are any scaling operations",
    "Type": "String",
    "AllowedPattern": "([a-zA-Z0-9_\\-\\.]+)@((\\[[0-9]{1,3}\\.[0-9]{1,3}\\.[0-9]{1,3}\\
    "ConstraintDescription": "must be a valid email address."
  },

  "KeyName" : {
    "Description" : "The EC2 Key Pair to allow SSH access to the instances",
    "Type" : "AWS::EC2::KeyPair::KeyName",
    "ConstraintDescription" : "must be the name of an existing EC2 KeyPair."
  },

  "SSHLocation" : {
    "Description" : "The IP address range that can be used to SSH to the EC2 instances",
    "Type": "String",
    "MinLength": "9",
    "MaxLength": "18",
    "Default": "0.0.0.0/0",
    "AllowedPattern": "(\\d{1,3})\\.(\\d{1,3})\\.(\\d{1,3})\\.(\\d{1,3})/(\\d{1,2})",
    "ConstraintDescription": "must be a valid IP CIDR range of the form x.x.x.x/x."
  }
},
```

Example for SDE

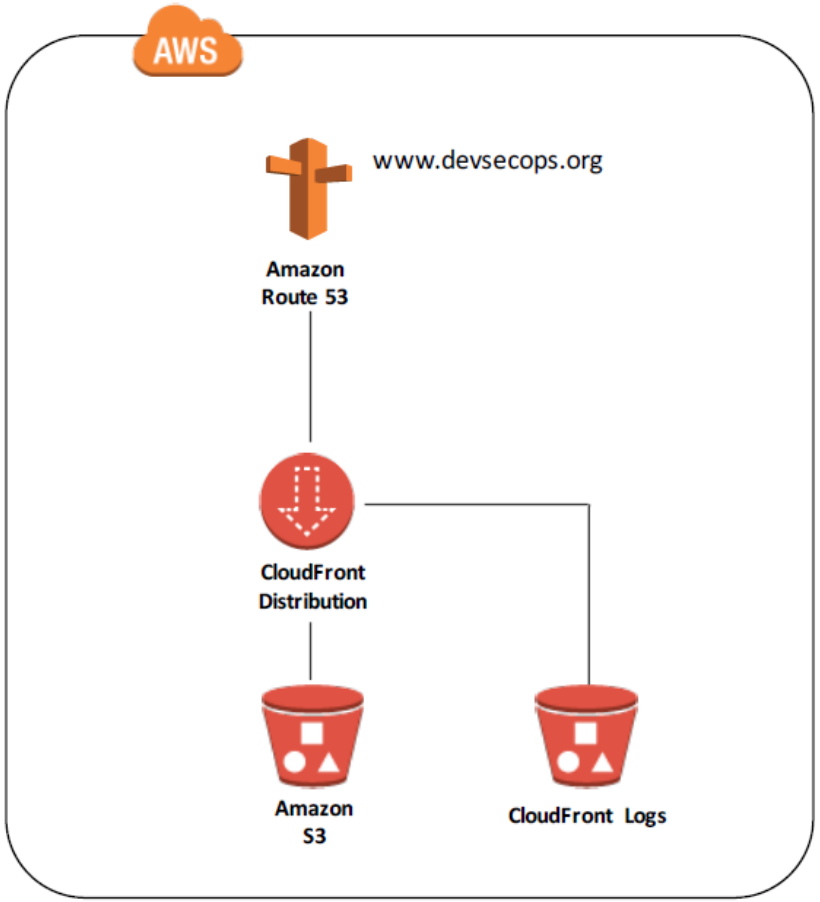


Micro - Services

- Decomposed Applications where each piece of functionality is its own service
- Scales by replicating these microservices across computing resources as needed
- Usually use a light weight communication protocol (HTTPS API)
- Commonly leverages a queue

Server less Microservice Architecture

- Managed services make it possible to run a full application that does not require physical server
- Each managed service is considered a microservice
- Multiple microservices can be put together to create a fully functional application
- Great for HTML 5 and Angular web applications



Security Considerations

- Is there visibility?
- Is there logging?
- Is there auditing the logging?
- Are there service logs?
- Are there API access logs?
- Is there encryption?
- Can customers control their own encryption keys?

Code “Sharing”

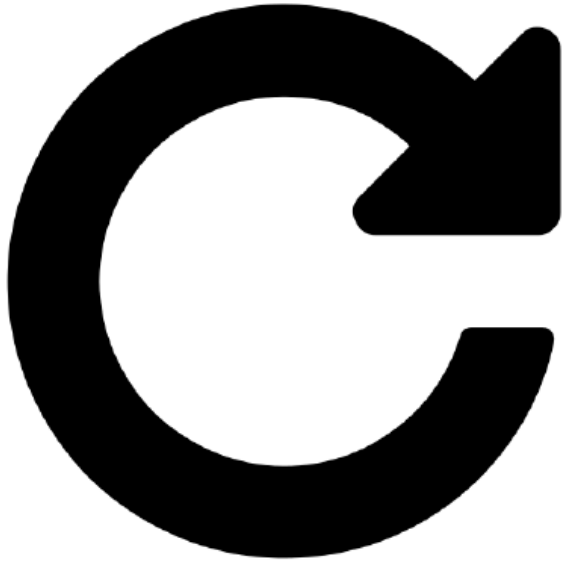
- GitHub makes copy paste easy
- It also makes it easy to paste in vulnerabilities
- Have you ever included a library without looking at it?

Bad Coding Practices

- Trusting the User
- Not Validating Input
- Hardcoding Secrets
- Trusting code without validating it
- Adding secrets to SCM (GitHub)
- What's your favorite?

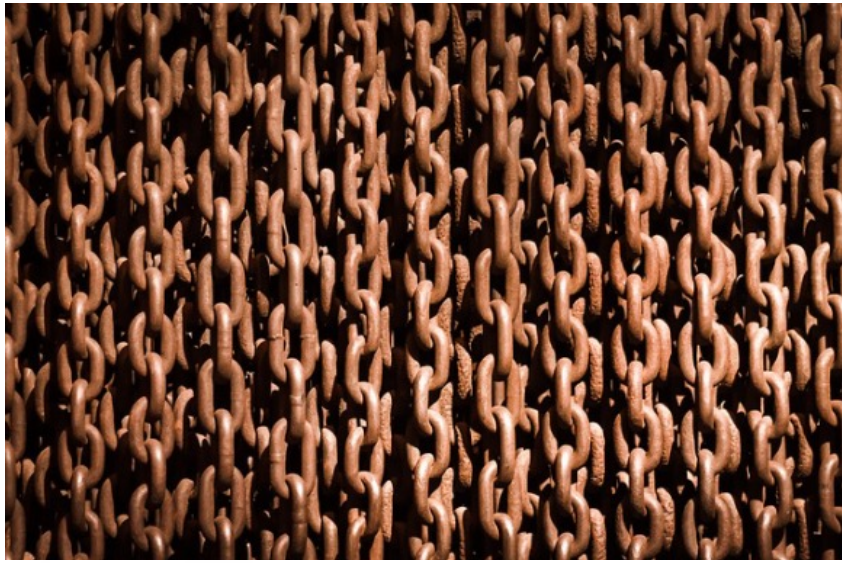
Intersection with DevOps

- Faster iterations can mean faster introduction of defects.
- Deployments now include infrastructure.
- Deployments now include application configurations.
- Anyone ever use Jenkins?
- BUT -> Faster iterations can mean faster fixes.



Software Supply Chain

- Better fewer suppliers
- Humans can’t move fast enough
- Automation is a must
- Be careful about selecting your dependencies
- The new hotness is not necessarily the most secure option



Top 10 Vulnerabilities

- Code Injection
- Broken Authentication and Session Management
- Cross Site Scripting
- Insecure Direct Object References
- Security Misconfiguration
- Sensitive Data Exposure
- Missing Function Level Access Control
- Cross Site Request Forgery
- Using Components with Known Vulnerabilities
- Invalidated Redirects and Forwards

Get Involved and Join the Community

- devsecops.org
- @devsecops on Twitter
- DevSecOps on LinkedIn
- DevSecOps on Github
- RuggedSoftware.org
- Compliance at Velocity