

MANICODE

SECURE CODING EDUCATION

Zero to DevSecOps:
Security in a DevOps World

A little background dirt…

@jimmesta

- 10 years of penetration testing, teaching, and building security programs
- OWASP AppSec California organizer and Santa Barbara chapter founder
- Conference speaker
- Been on both sides of the InfoSec fence
- Loves Clouds



COPYRIGHT ©2019 MANICODE SECURITY



COPYRIGHT ©2019 MANICODE SECURITY



COPYRIGHT ©2019 MANICODE SECURITY

We Have a “Situation”



COPYRIGHT ©2019 MANICODE SECURITY

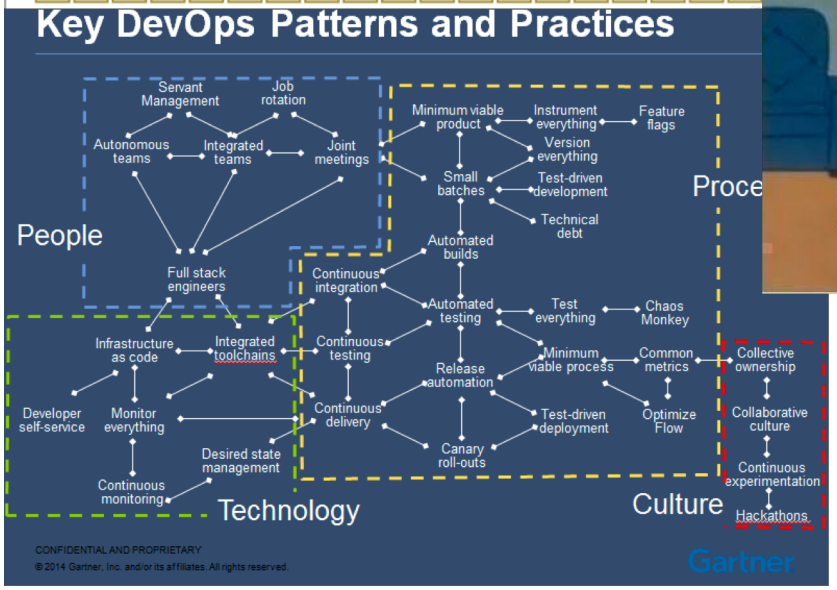
The Situation









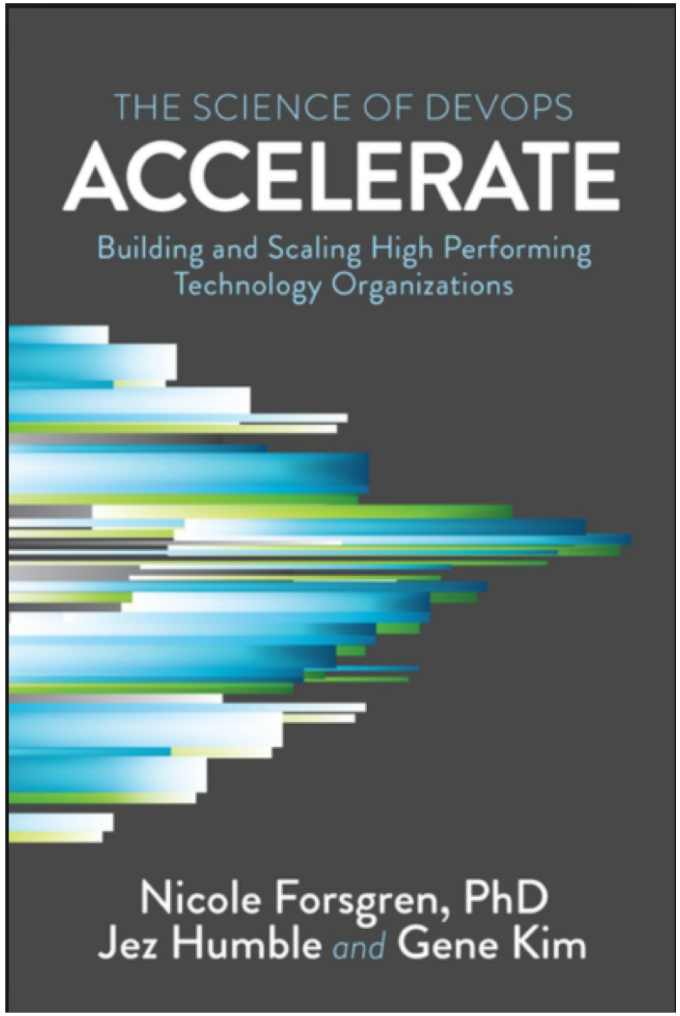






The (Actual) Current State of Affairs

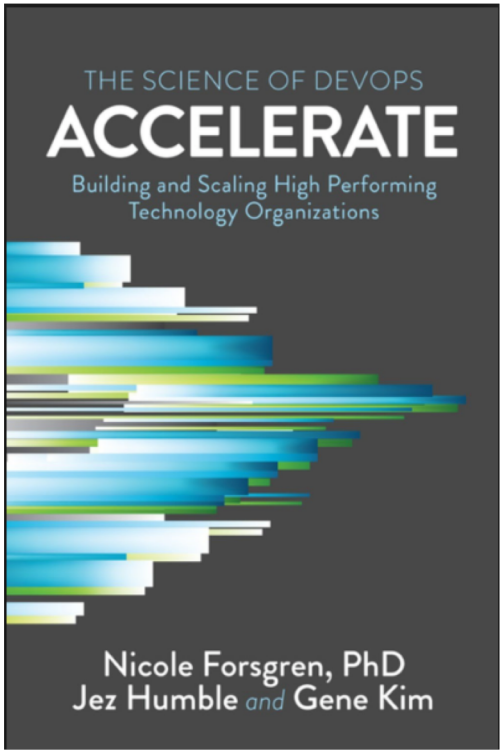
"Our research has uncovered 24 key capabilities that drive improvements in software delivery performance in a statistically significant way."



COPYRIGHT ©2019 MANICODE SECURITY

Continuous Delivery Capabilities

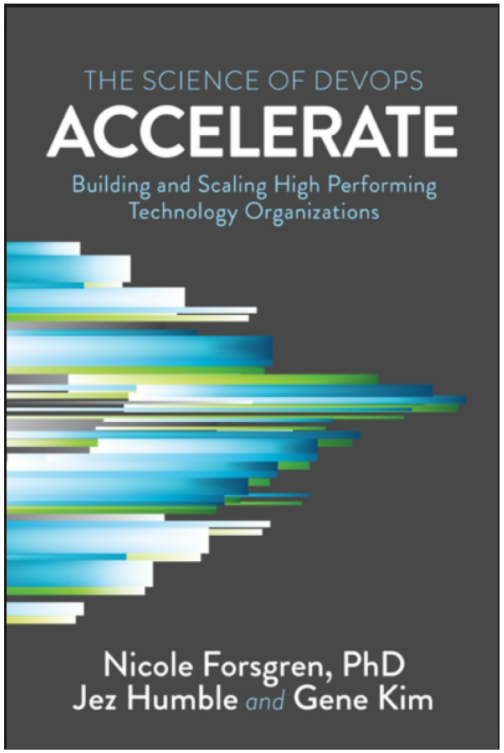
- Version Control
- Deployment Automation
- Continuous Integration
- Trunk-Based Development
- Test Automation
- Test Data Management
- ***Shift Left on Security***
- Continuous Delivery



COPYRIGHT ©2019 MANICODE SECURITY

Architecture Capabilities

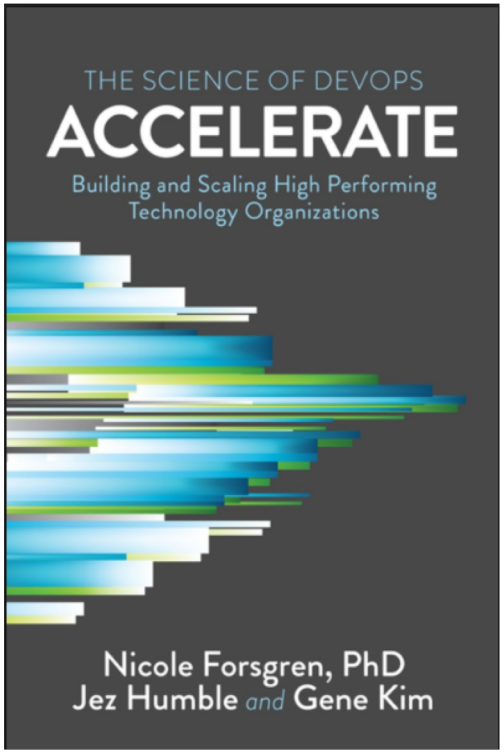
- Loosely Coupled Architecture
- Empowered Teams
- Customer Feedback
- Working in Small Batches
- Team Experimentation



COPYRIGHT ©2019 MANICODE SECURITY

Lean Management and Monitoring Capabilities

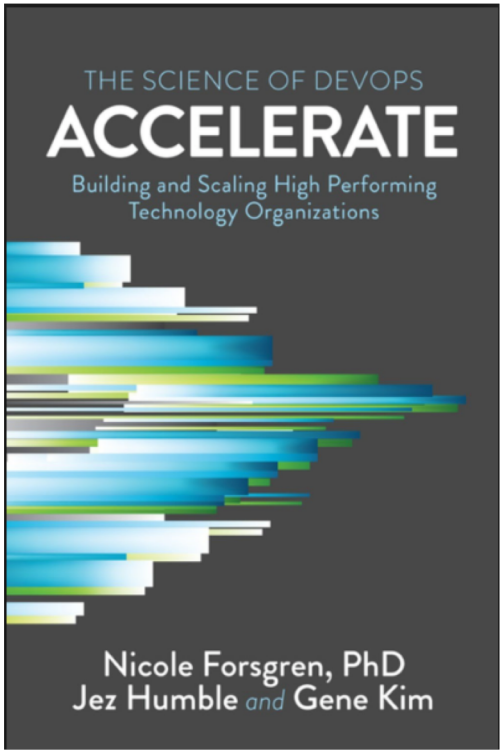
- Change Approval Process
- Monitoring
- Proactive Notification
- WIP Limits
- Visualizing Work



COPYRIGHT ©2019 MANICODE SECURITY

Cultural Capabilities

- Supporting Learning
- Collaboration Among Teams
- Job Satisfaction
- Transformational Leadership



COPYRIGHT ©2019 MANICODE SECURITY

High Performers vs. Low Performers

- 46x more frequent code deployments
- 440x faster lead time from commit to deploy
- 170x faster mean time to recover from downtime
- 5x lower change failure rate

COPYRIGHT ©2019 MANICODE SECURITY

High Performing Security Teams

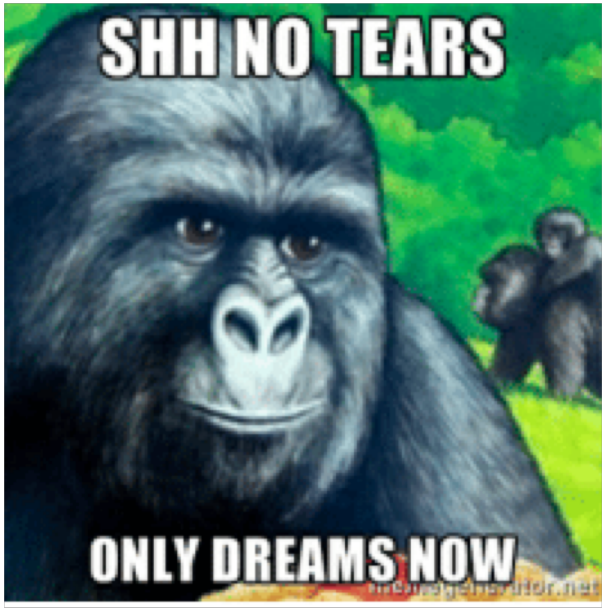
"High-performing teams were more likely to incorporate information security into the delivery process. Their infosec personnel provided feedback at every step of the software delivery lifecycle, from design through demos to helping out with test automation. However, **they did so in a way that did not slow down the development process**…"

COPYRIGHT ©2019 MANICODE SECURITY

Goals of this Course

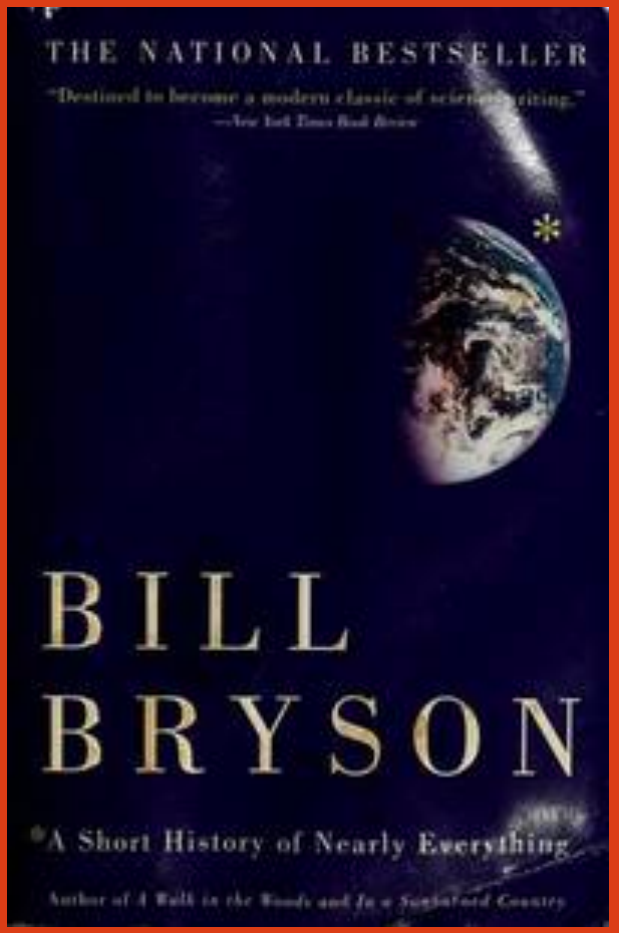
- Give you the tools to move the needle to “High Performer”
- Common consensus (or not?) on DevOps and DevSecOps
- Deploy software more confidently
- Understand cloud security topics
- Exposure to only the best memes

Most importantly, have some fun!



COPYRIGHT ©2019 MANICODE SECURITY

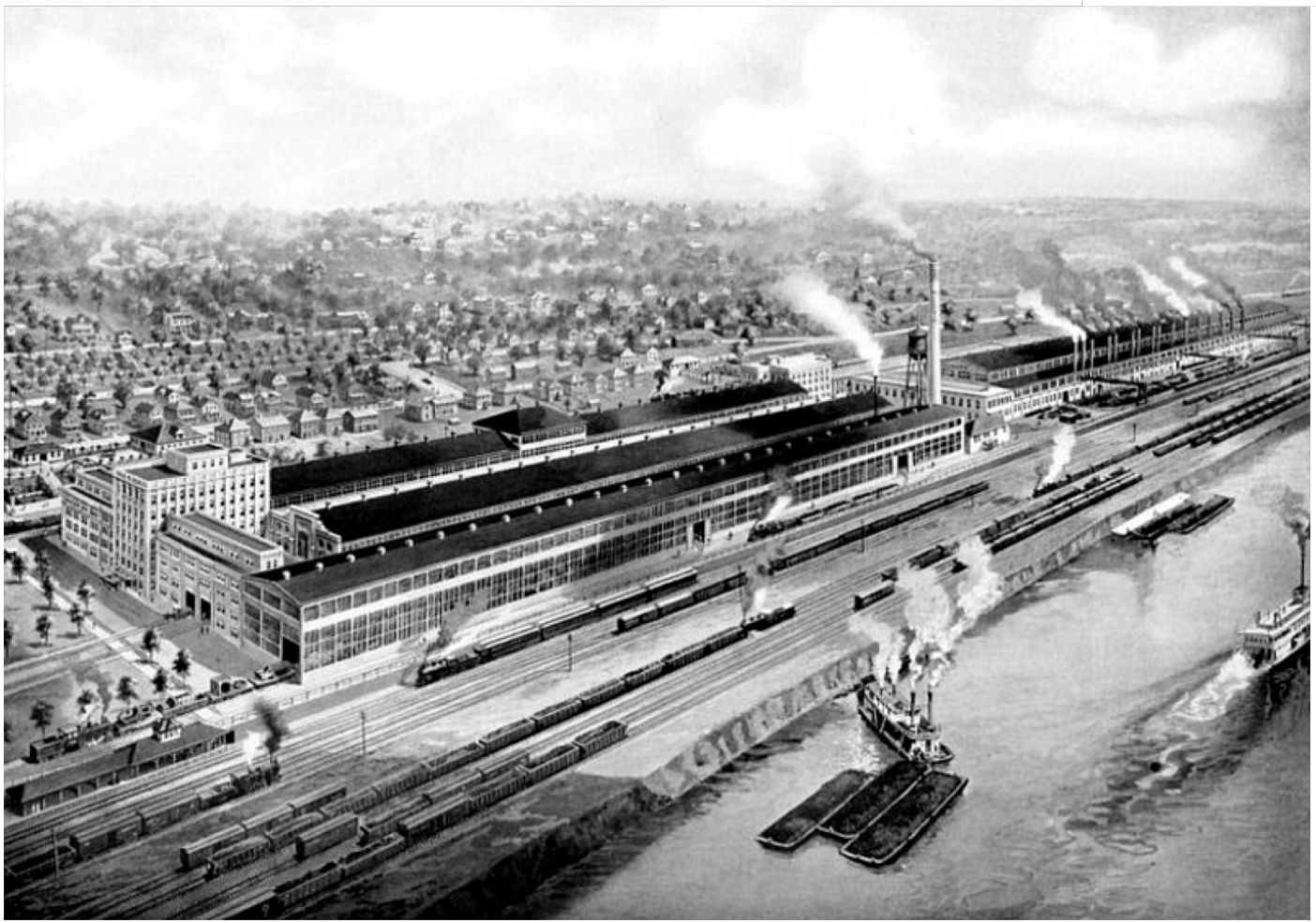
A Brief History of the SDLC



COPYRIGHT ©2019 MANICODE SECURITY

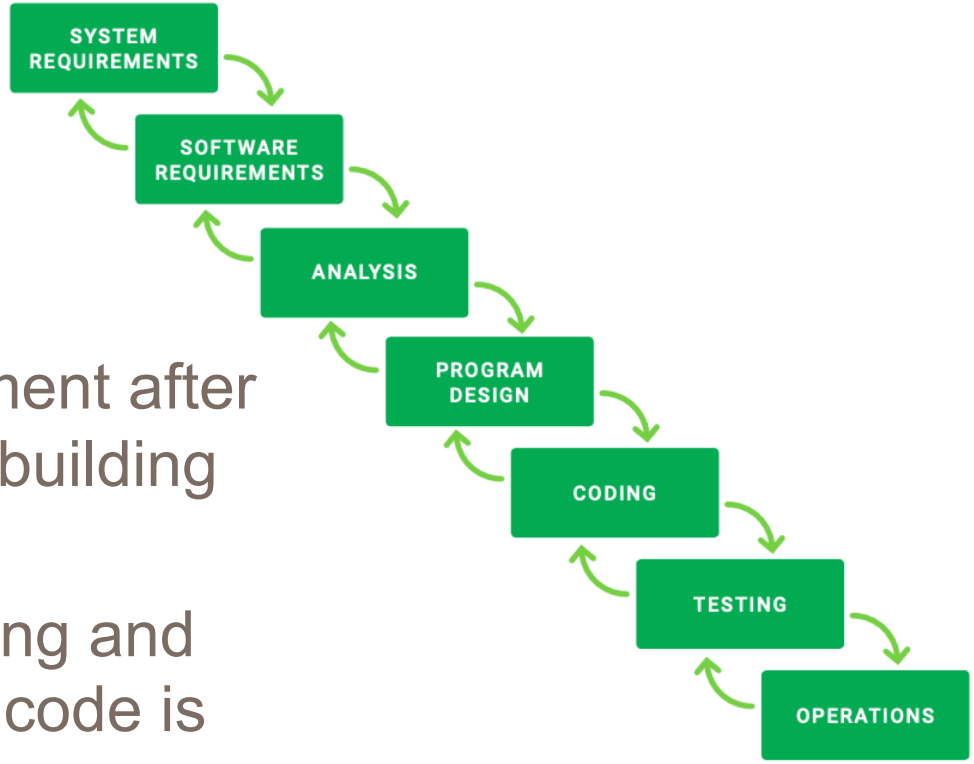
Part 1: The Waterfall Era





COPYRIGHT ©2019 MANICODE SECURITY

Part 1: The Waterfall Era

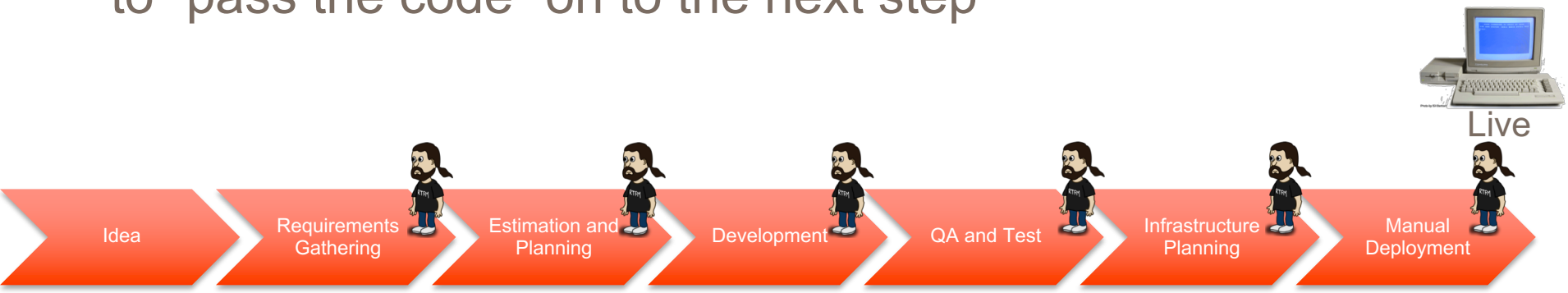


- Modeled software development after what we knew and learned building hardware
- Months (or years!) of planning and preparation before a line of code is written
- All good stories have to start *somewhere*

COPYRIGHT ©2019 MANICODE SECURITY

Traditional SDLC AKA “Waterfall”

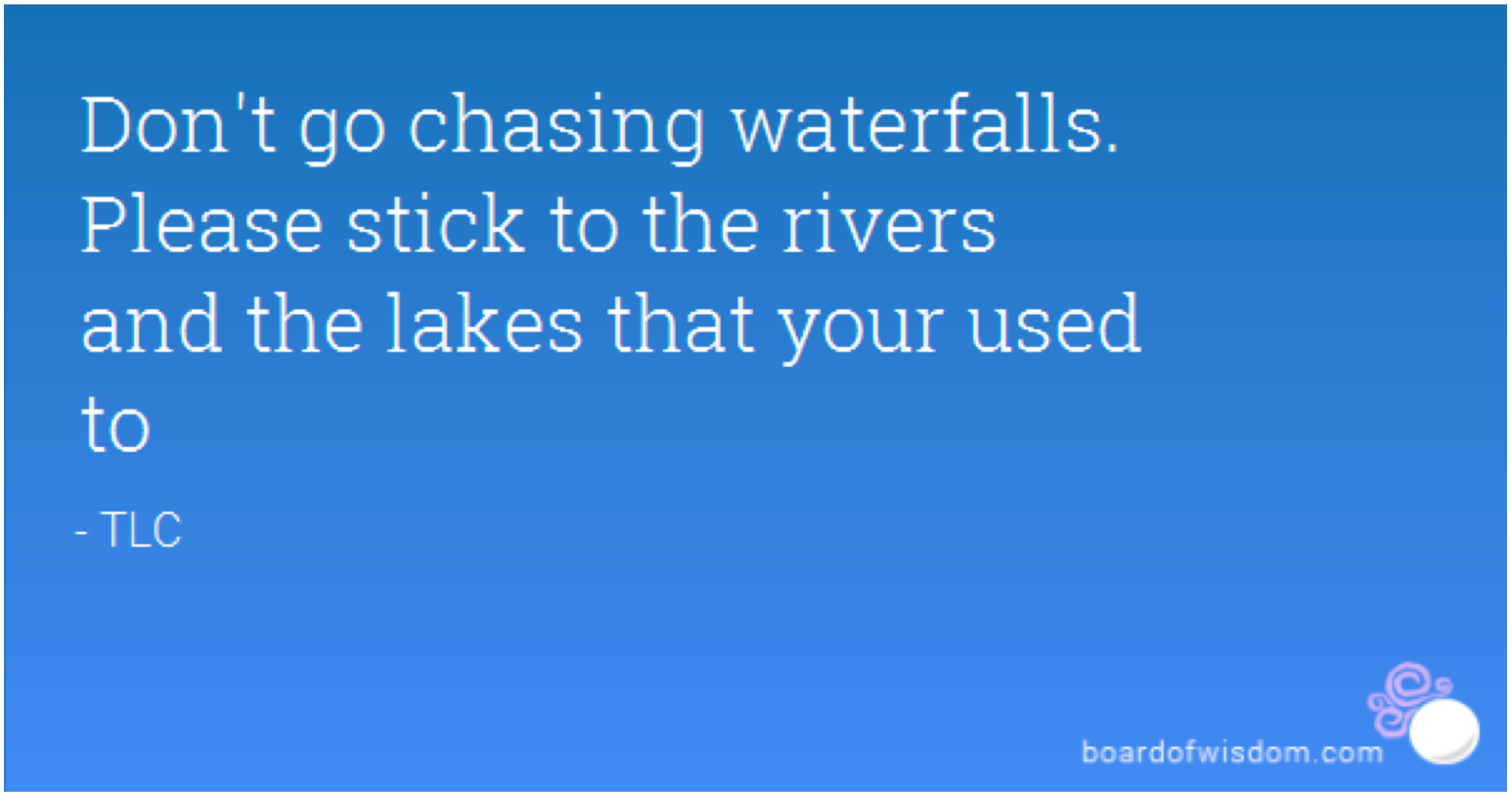
- **Optimizes for risk management.** Assuming the cost of a mistake is high and tolerance for risk is low
- Critical services still benefit from certain ”waterfall” methodologies
- Linear progression when deploying software
- Relies heavily on human intervention and interaction to “pass the code” on to the next step



COPYRIGHT ©2019 MANICODE SECURITY



COPYRIGHT ©2019 MANICODE SECURITY



COPYRIGHT ©2019 MANICODE SECURITY

Part 2: The Agile Enlightenment



Carol Morton/The Pittsburgh Press

Putnam McDowell, left, and Chester Engineers President Al Baily

Alive and well

Mestek is a new 'chapter' in Mesta story

By William H. Wylie
The Pittsburgh Press

MESTEK INC., once the mighty Mesta Machine Co., is alive and apparently well after emerging earlier this year from a bankruptcy ordeal that lasted nearly two years.

Things are going so much better that Putnam B. McDowell, who steered Mesta through the tricky Chapter 11 maze, said, "Now I can sit down and have a drink with some of those lawyers and we laugh. . . . It's like war stories."

But the Mesta bankruptcy was no laughing matter during the grueling days of 1983 and '84 when the fate of the once "Cadillac" of mill machinery builders was being litigated in Federal Bankruptcy Court here.

Asked if he ever had any doubts about getting out of Chapter 11 — less than 10 percent of the companies that file make it — he replied, "About every third day for a year something disastrous seemed about to happen . . . But I never lost my basic faith that somehow we could work it out."

Thousands of employees and retirees were hurt financially by Mesta's collapse. Jobs were lost and some pension benefits were reduced by the government's Pension Benefit Guaranty Corp., which took over the fund.

The West Homestead plant, which housed one of the world's largest foundries, and the New Castle facility were sold, sounding Mesta's last hurrah as a manufacturer.

After distribution of $25.1 million in cash, more than 1 million shares of common and preferred stock and warrants to purchase common stock, notes totaling $4.7 million and deferred payments of $1.5 million, creditors received about 30 cents on the dollar.

Mestek is a mere shadow of its former self, with approximately 220 employees, total assets of $10.7 million and estimated annual revenues of $15 million to $18 million.

That contrasts sharply with the 3,000 who worked for Mesta during its heyday, assets of $74 million and annual sales as high as $120 million.

If it weren't for two Mesta subsidiaries and a joint venture with one of Victor Posner's companies — none of the subsidiaries was involved in the bankruptcy — there might not be a Mestek. The holding company's principal sources of income are The Chester Engineers Inc., a Coraopolis-based engineering firm, and MCS Inc., a Monroeville computer company.

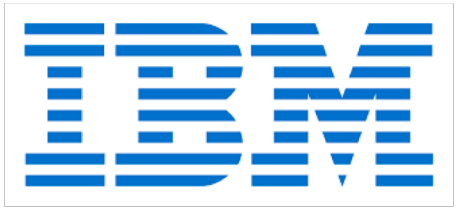
Mestek's 49 percent interest in Mesta Engineering Co., which is owned jointly with Pennsylvania Engineering Corp.,

Please see Mestek, C5

COPYRIGHT ©2019 MANICODE SECURITY

Part 2: The Agile Enlightenment

- Realization that software differs from hardware
- Competition emerges and first-to-market matters
- 90's was all about experimentations in effective software deployment
- Sprints, daily standups, retrospectives emerge
- Manual testing, QA, and deployment





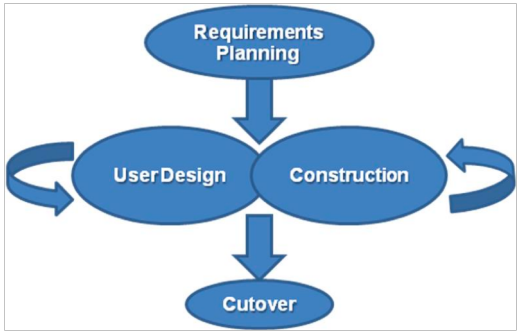
COPYRIGHT ©2019 MANICODE SECURITY

Part 2: The Agile Enlightenment

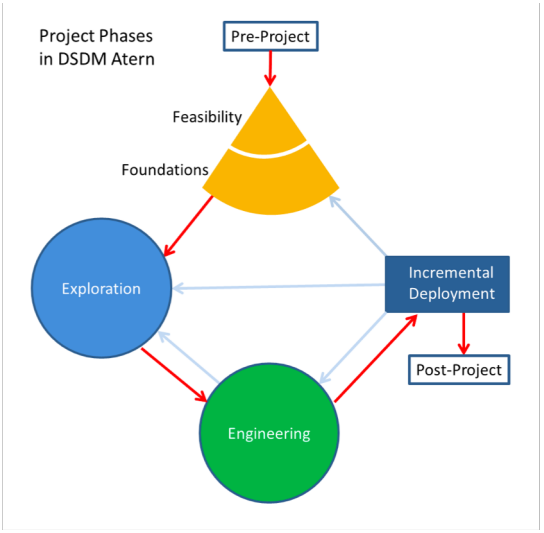


Extreme Programming





Rapid Application Development (RAD) Model



Dynamic Systems Development Method

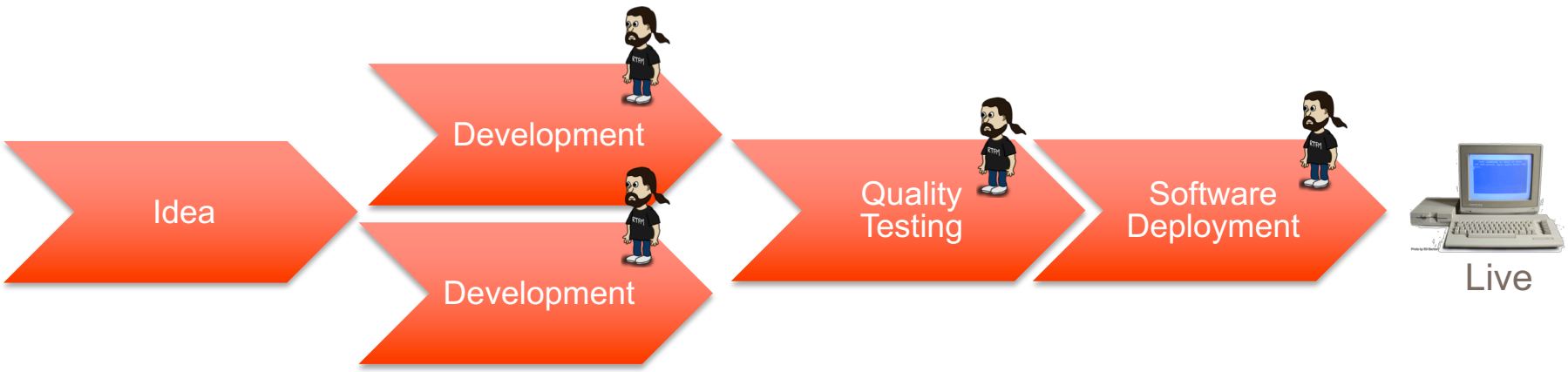


Scrum

COPYRIGHT ©2019 MANICODE SECURITY

Agile / Scrum / Extreme

- Begin optimizing for speed and agility
- **Incremental changes**
- Beginning of TDD, timeboxing, stories, pair-programming, etc.
- We begin *thinking* about and measuring the effectiveness of our SDLC



COPYRIGHT ©2019 MANICODE SECURITY

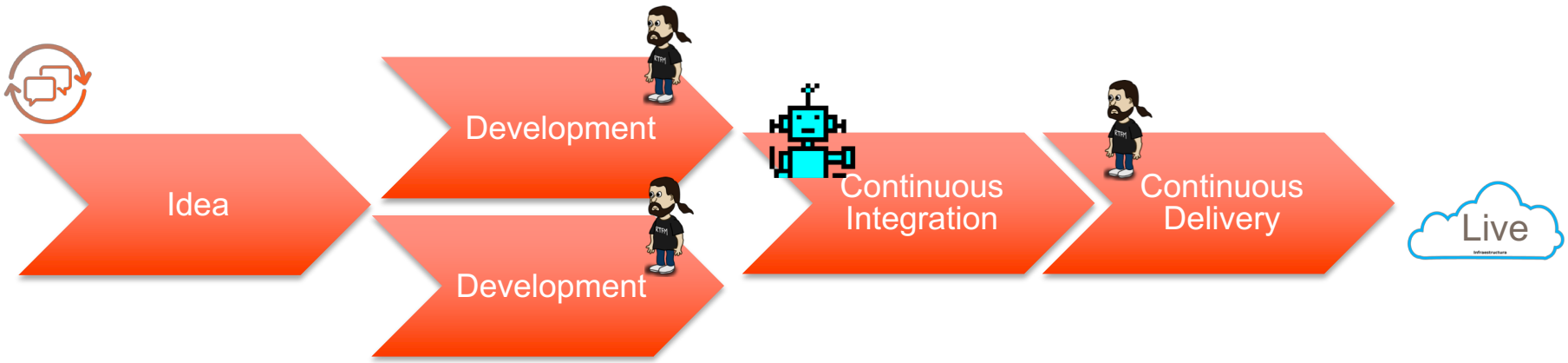
Part 3: Invasion of the Robots



COPYRIGHT ©2019 MANICODE SECURITY

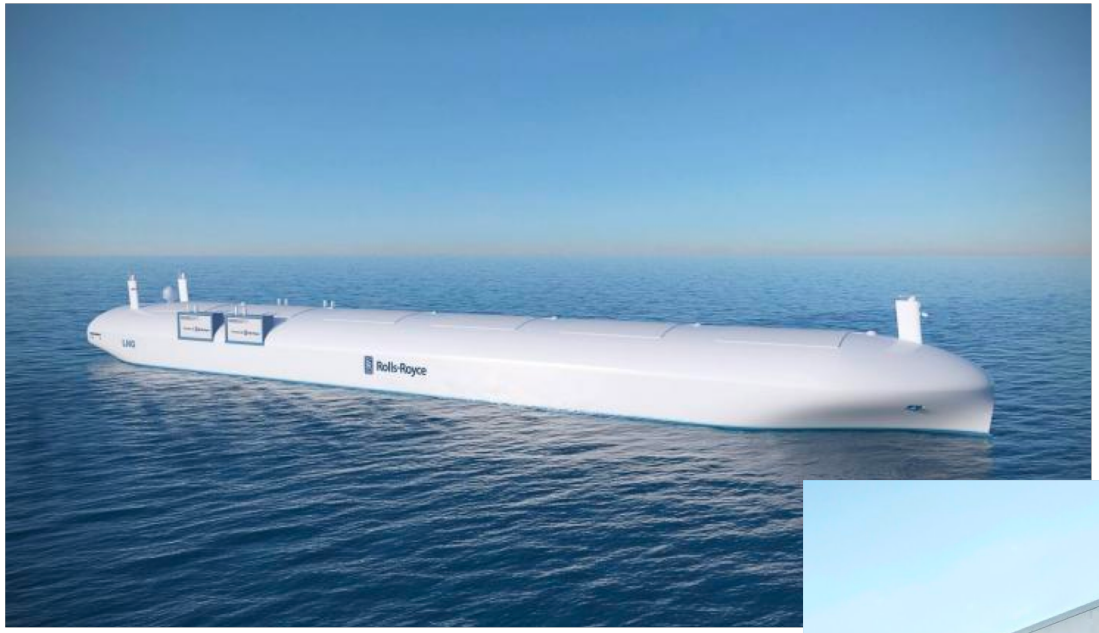
Continuous Integration and Delivery

- Optimizes for speed and agility. Assuming the cost of a mistake is low and tolerance for risk is high
- Parallel and incremental changes
- Automation and upfront work makes this possible
- **Self-testing code and early days of automated QA**



COPYRIGHT ©2019 MANICODE SECURITY

Part 4: The Current State of Affairs

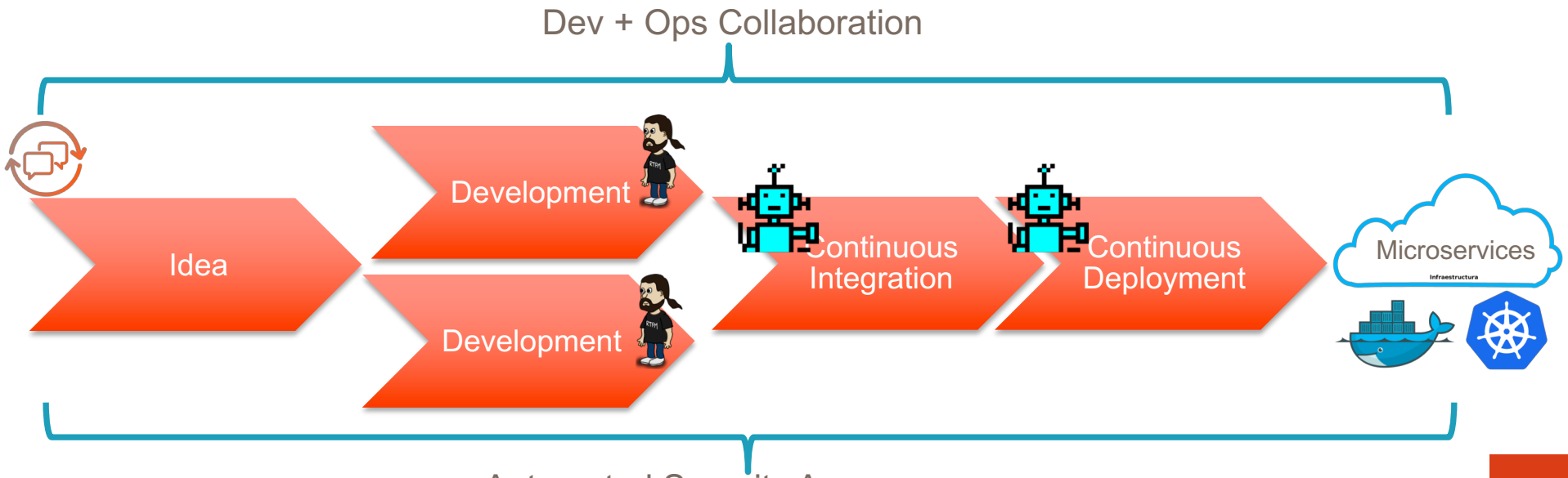




COPYRIGHT ©2019 MANICODE SECURITY

DevSecOps

- Cultural shift towards end-to-end ownership of code
- Zero-downtime, automated deployments
- Emergence of containers, serverless, and zero-downtime deployments
- ”Everything-as-Code” is the new standard
- **Security is no longer a blocker or silo**



COPYRIGHT ©2019 MANICODE SECURITY

DevSecOps Advantages

- Add customer value
- Puts security in everyone's job description
- Eliminate "black box" security teams and tools
- Ability to measure security effectiveness
- Reduce attack surface and vulnerabilities
- Increase recovery speed
- Save $$$
- Secure by default mentality

COPYRIGHT ©2019 MANICODE SECURITY

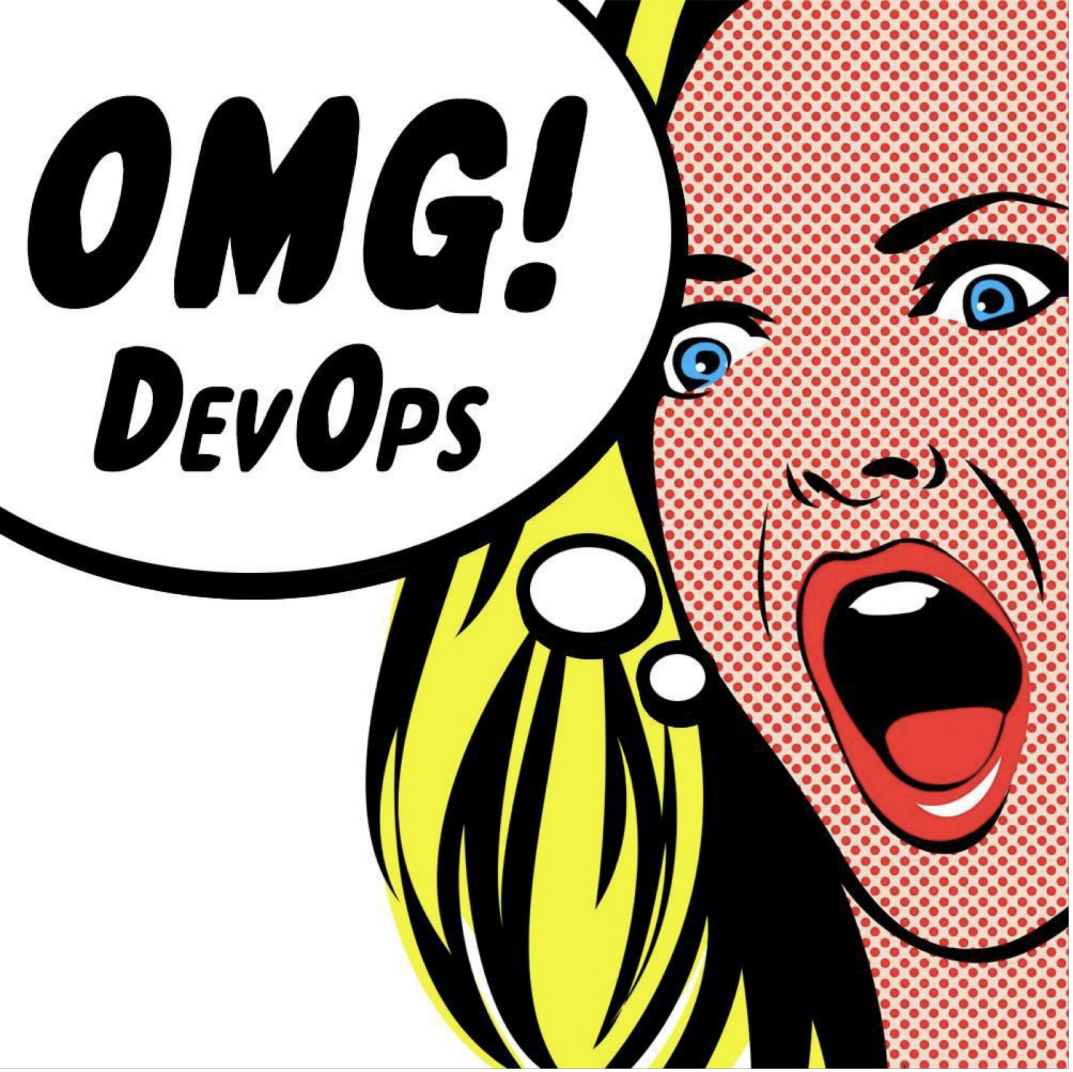
The Rest is History…



COPYRIGHT ©2019 MANICODE SECURITY

Introduction to DevOps

COPYRIGHT ©2019 MANICODE SECURITY

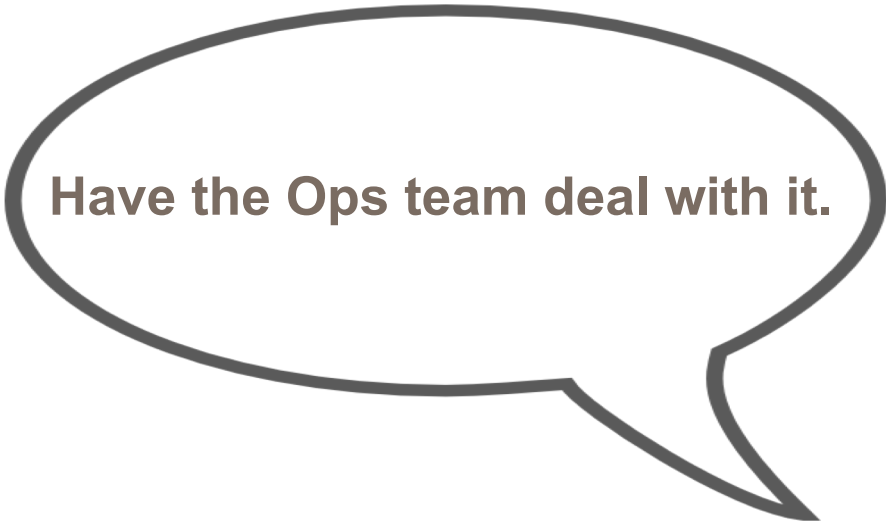


What *is* DevOps?

COPYRIGHT ©2019 MANICODE SECURITY

DevOps Anti-Patterns

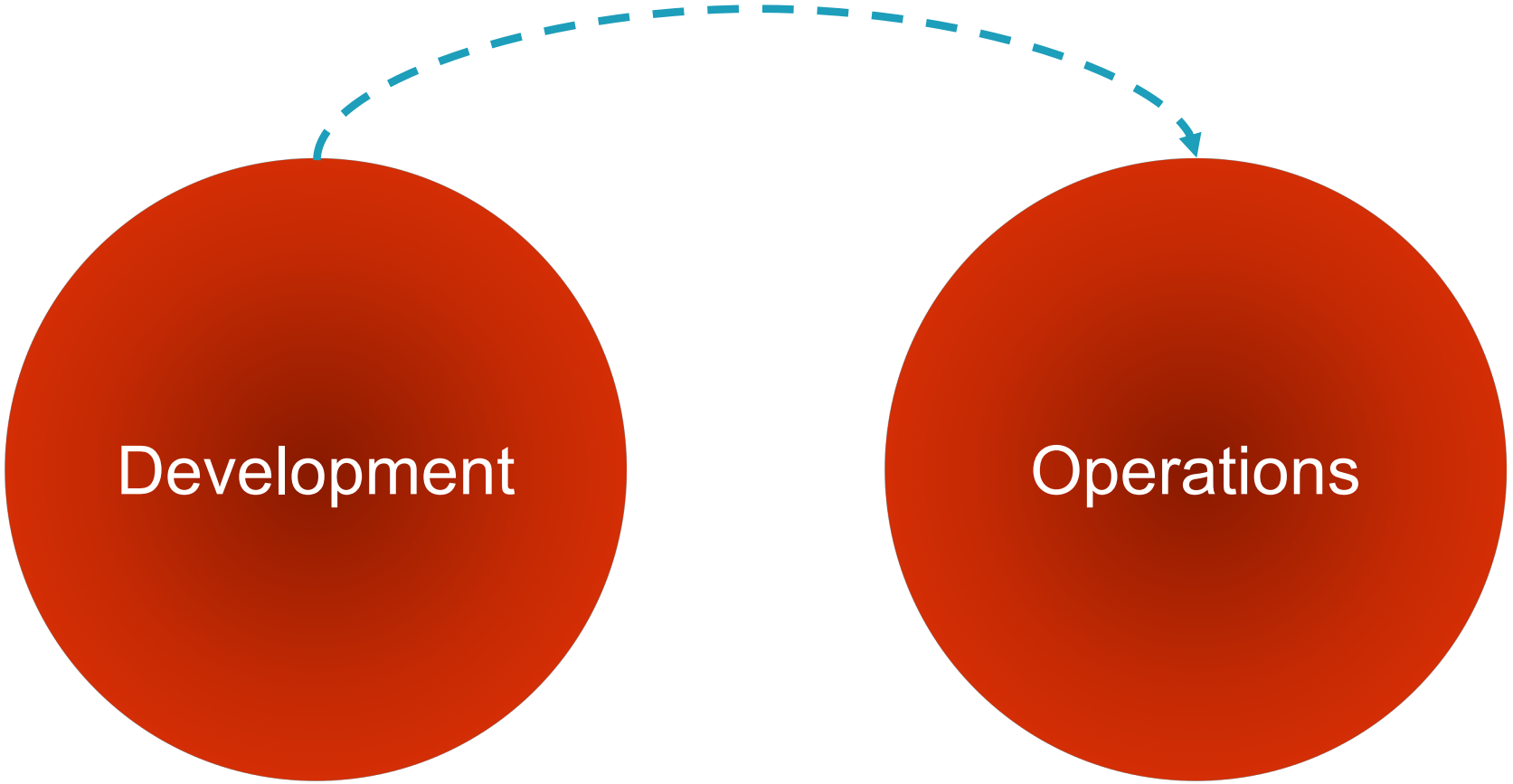






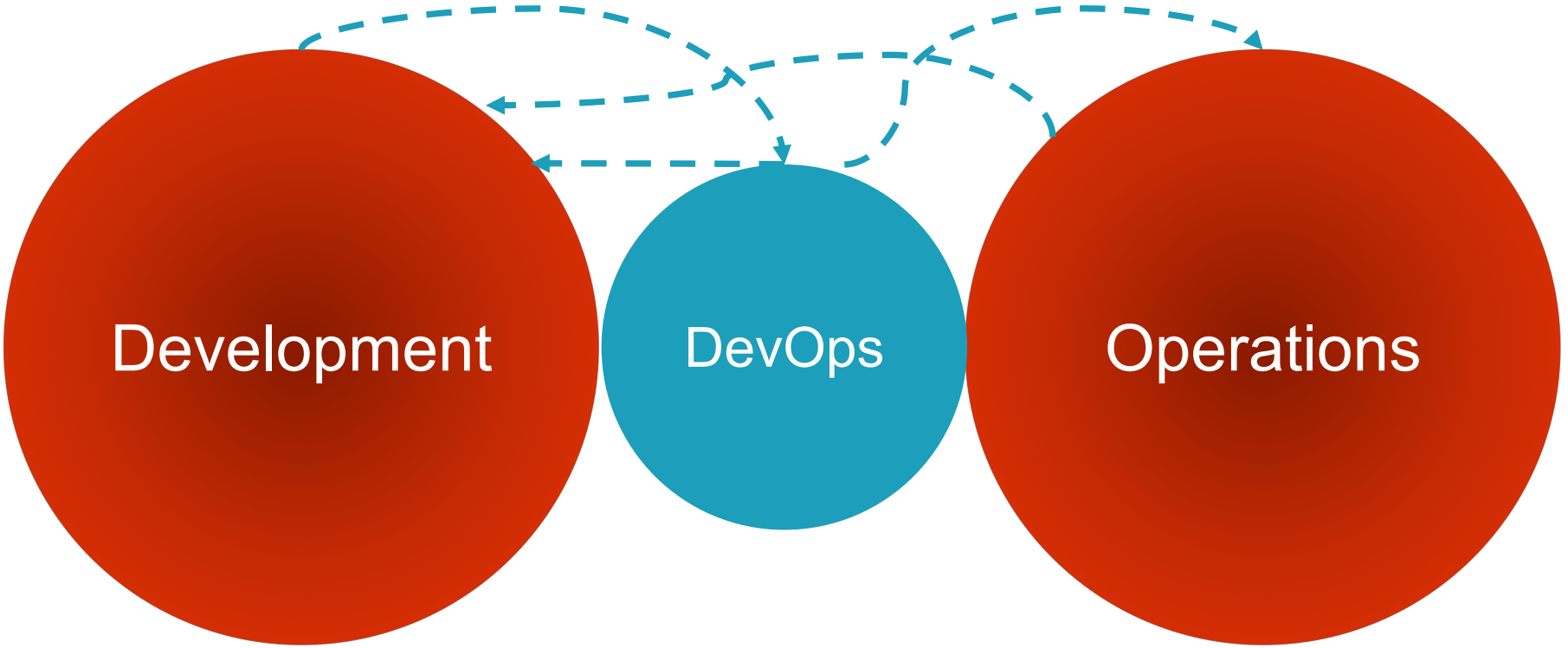
COPYRIGHT ©2019 MANICODE SECURITY

Anti-Pattern: “Throw it Over the Wall”



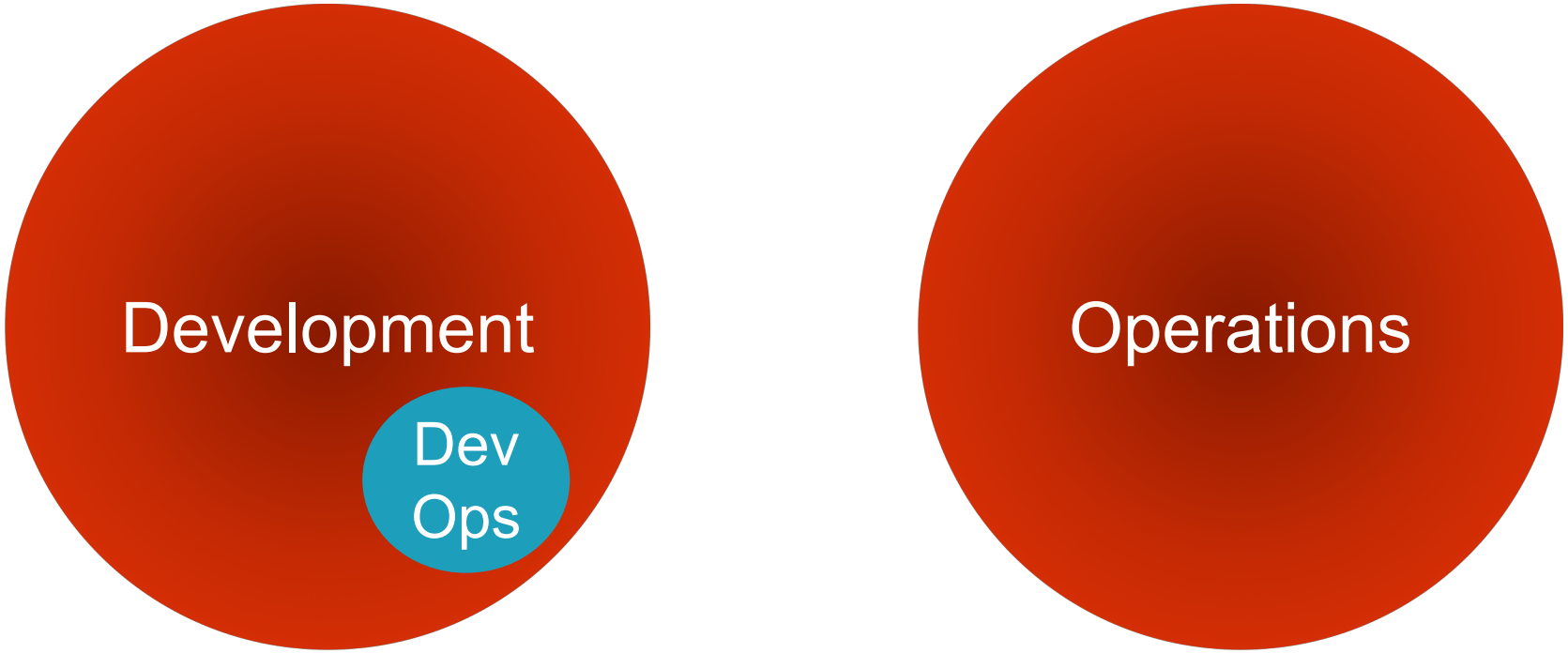
COPYRIGHT ©2019 MANICODE SECURITY

Anti-Pattern: “DevOps Team Silo”



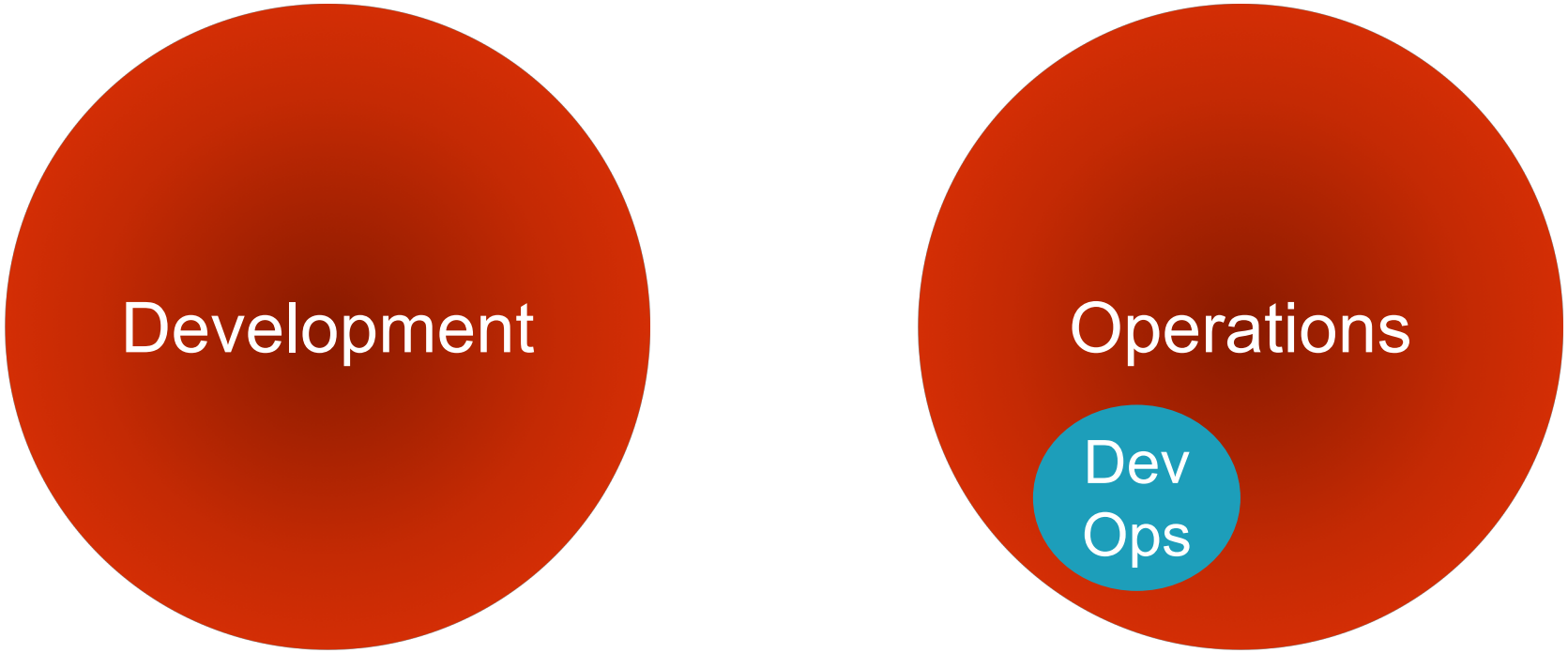
COPYRIGHT ©2019 MANICODE SECURITY

Anti-Pattern: “NoOps” Approach



COPYRIGHT ©2019 MANICODE SECURITY

Anti-Pattern: “Ops Will Handle it”



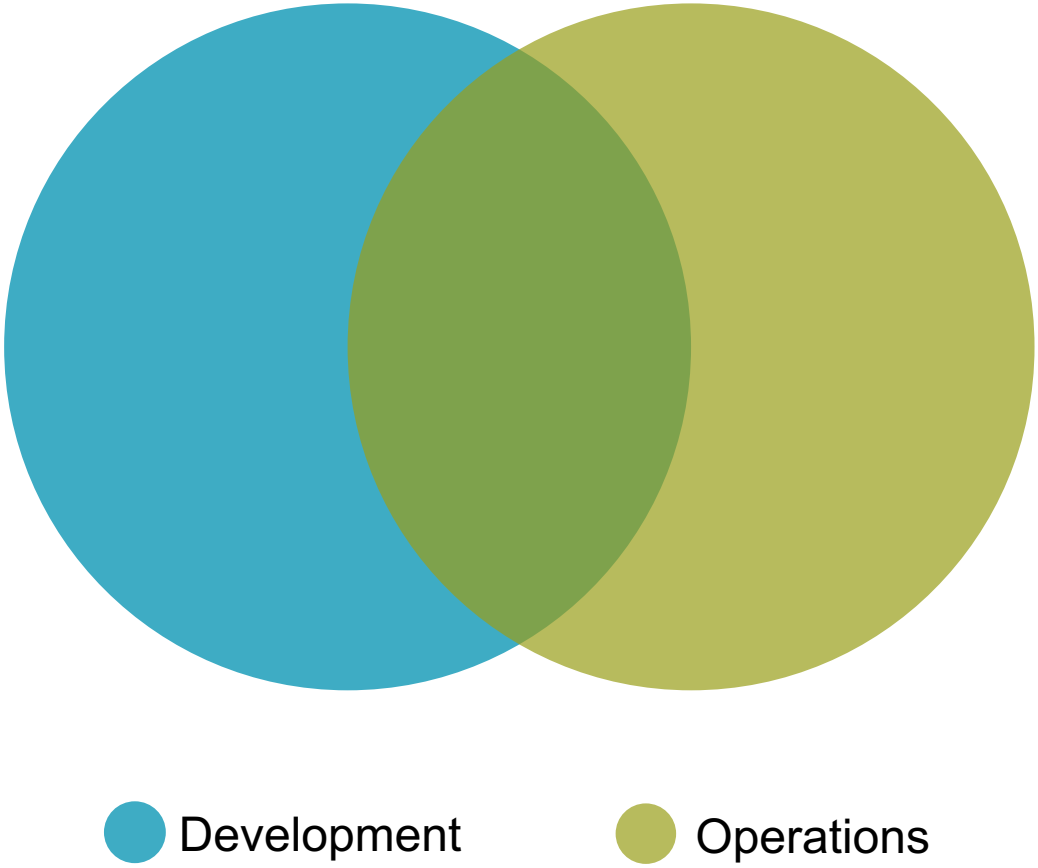
COPYRIGHT ©2019 MANICODE SECURITY

Anti-Pattern: “Ops Will Handle it”



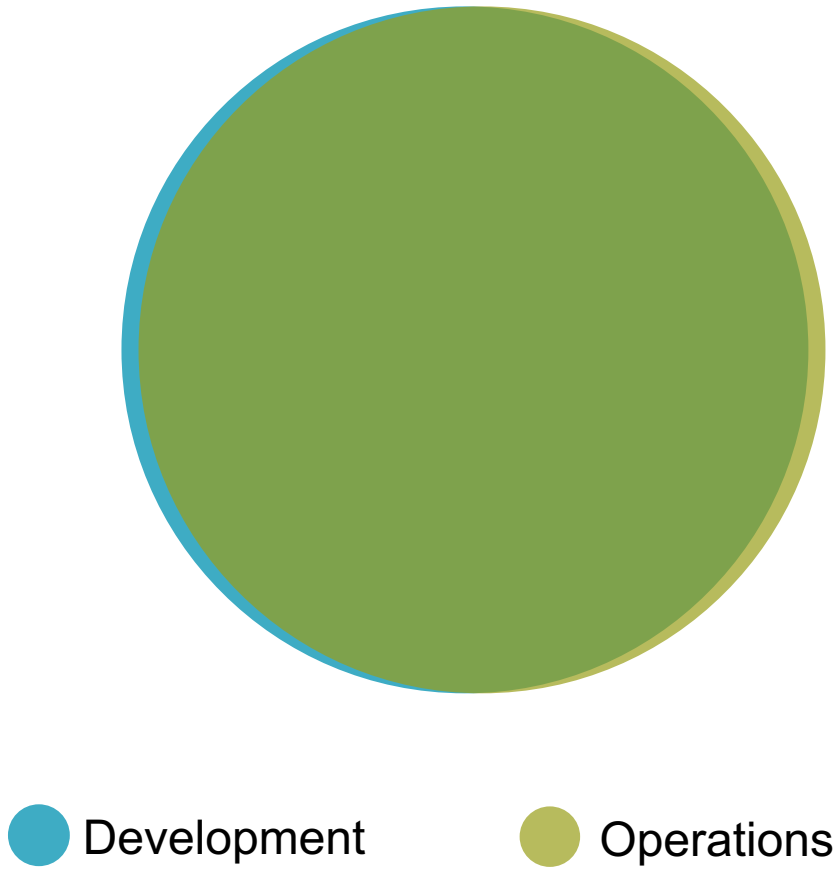
COPYRIGHT ©2019 MANICODE SECURITY

Development and Operations Collaboration



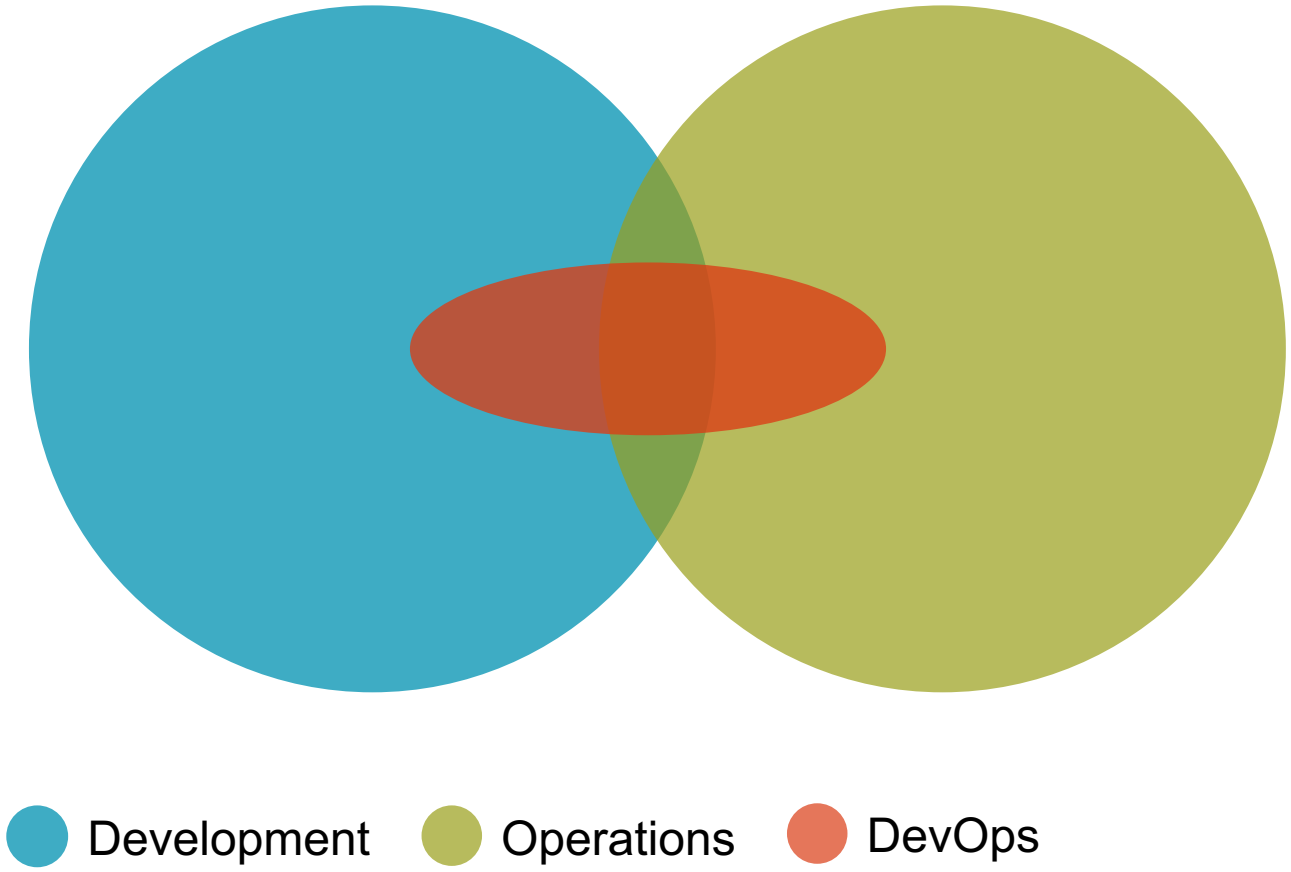
COPYRIGHT ©2019 MANICODE SECURITY

Dev and Ops Fully Shared Responsibilities



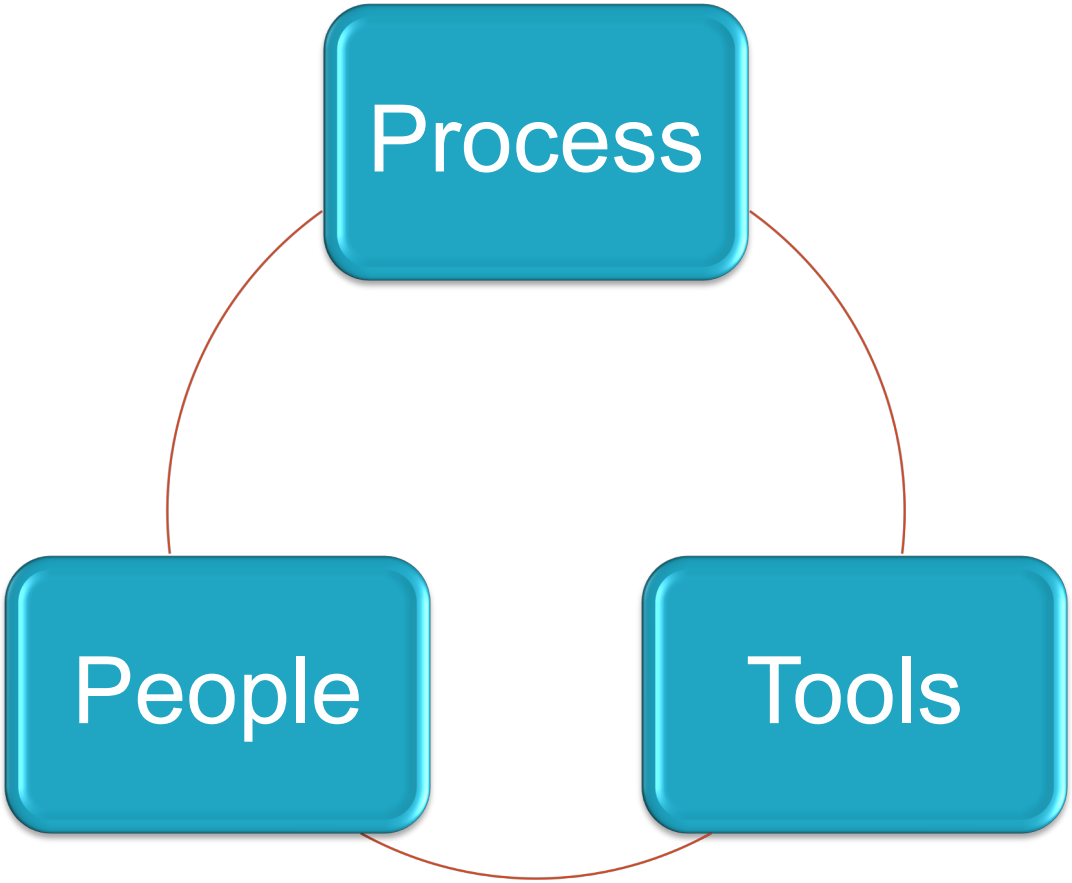
COPYRIGHT ©2019 MANICODE SECURITY

DevOps-as-a-Service



COPYRIGHT ©2019 MANICODE SECURITY

So…What *is* DevOps?



…and some buzzwords

COPYRIGHT ©2019 MANICODE SECURITY

Process

- Collaboration Across Teams
- Cross-Training
- Support Business Agility
- Breaking Down Silos
- Automation of Repetitive Tasks
- “Everything” as Code
- Test, Measure, and Monitor



People



Tools

COPYRIGHT ©2019 MANICODE SECURITY

Adding the “Sec” to DevOps

COPYRIGHT ©2019 MANICODE SECURITY



Windows for automated scanning and manual testing are shrinking

Continuous delivery scares security teams

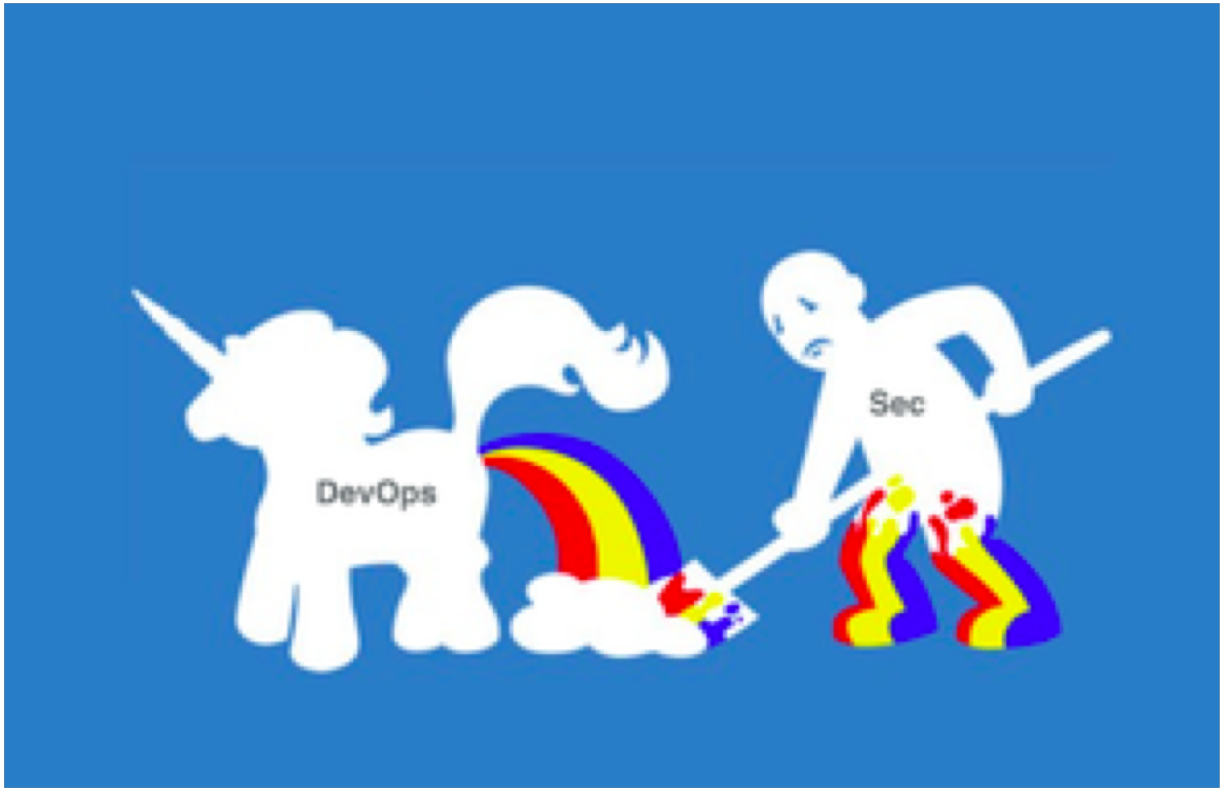
Framework, language, infrastructure fatigue

Security teams are vastly outnumbered

Automated detection of complex issues is hard work

Third-party code / libraries / APIs / tooling scattered *everywhere*

We want to turn this…



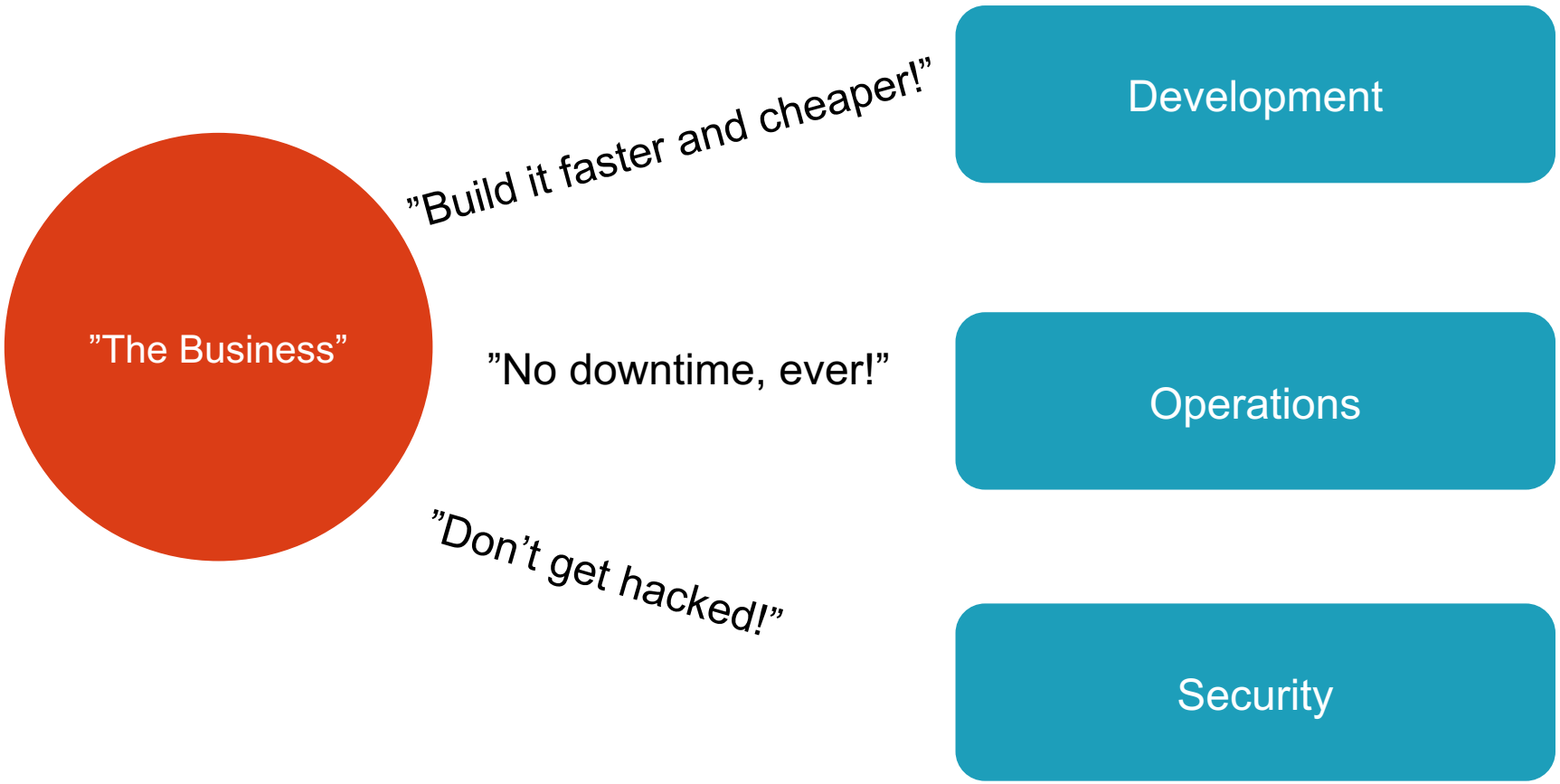
COPYRIGHT ©2019 MANICODE SECURITY

Into this!



COPYRIGHT ©2019 MANICODE SECURITY

Competing Forces



COPYRIGHT ©2019 MANICODE SECURITY

The Case for DevSecOps

- Software and product development is *rapidly* moving towards Agile, Scrum, DevOps
- The “perimeter” as we know it is going away
- Traditional security mechanisms are failing to keep up
- The demand for security aka “not getting hacked” is skyrocketing
- Security is becoming a marketing tool and selling point



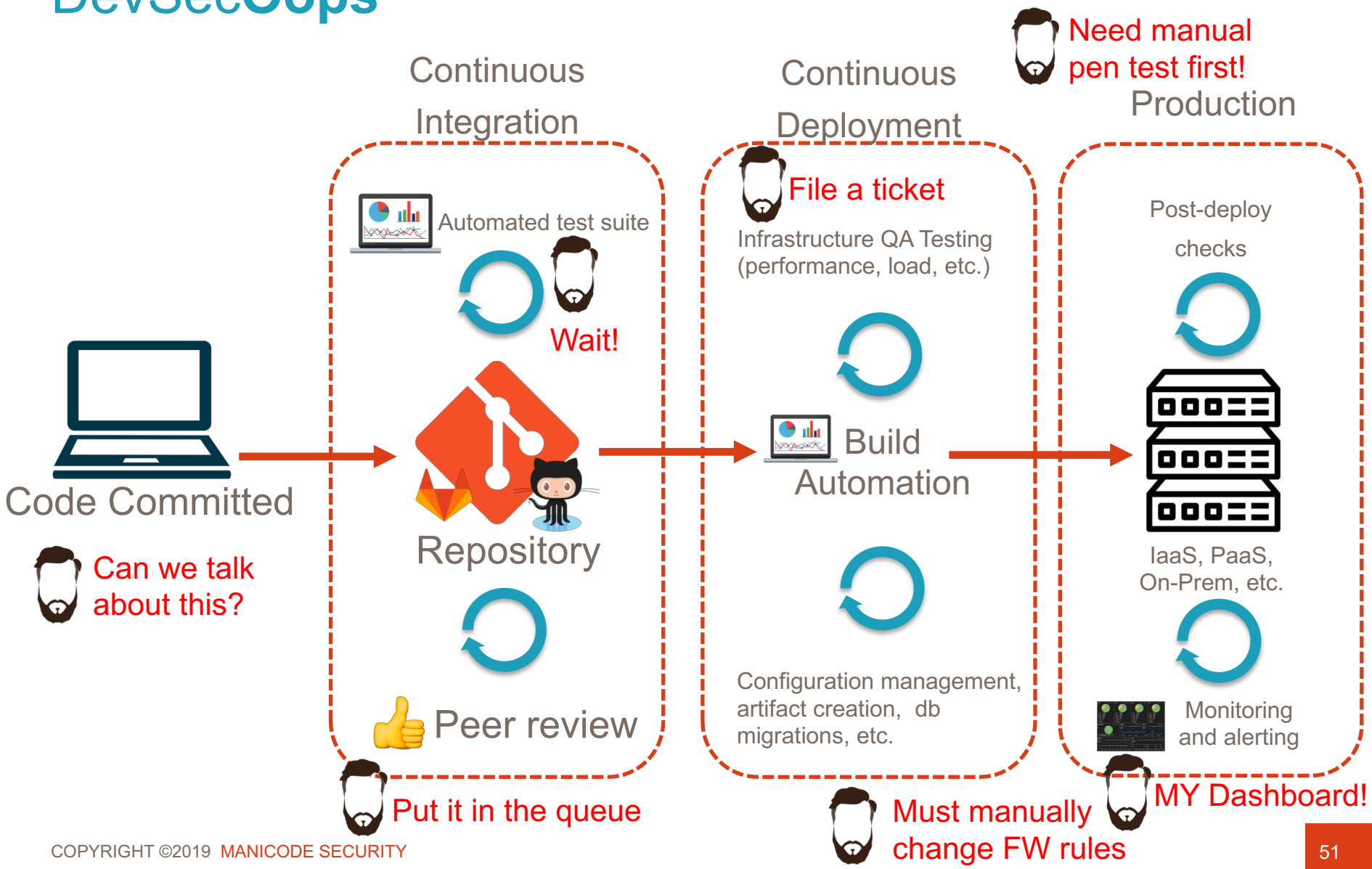
Traditional Security SDLC Integration

- Worked in a ”waterfall” world where time was not of the essence
- Many manual checkpoints and heavy domain expertise
- Often relies on opinion vs. science and data
- Tools tailored towards local operator machines
- Knowledge sharing and communication can be messy at best



COPYRIGHT ©2019 MANICODE SECURITY

DevSecOops

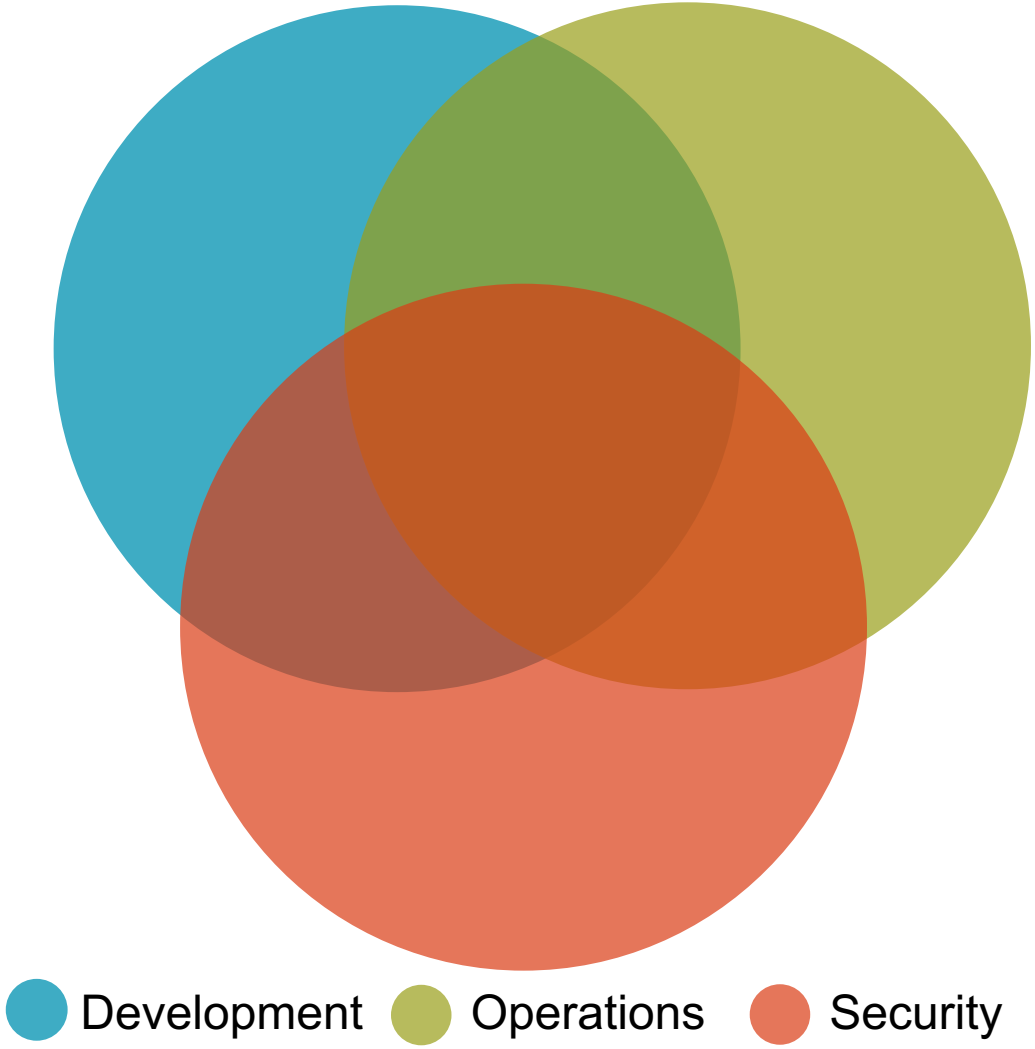


COPYRIGHT ©2019 MANICODE SECURITY

“DevSecOps is the process of incorporating and enforcing meaningful security controls without slowing down deployment velocity.”

COPYRIGHT ©2018 MANICODE SECURITY

DevSecOps Trinity of Success



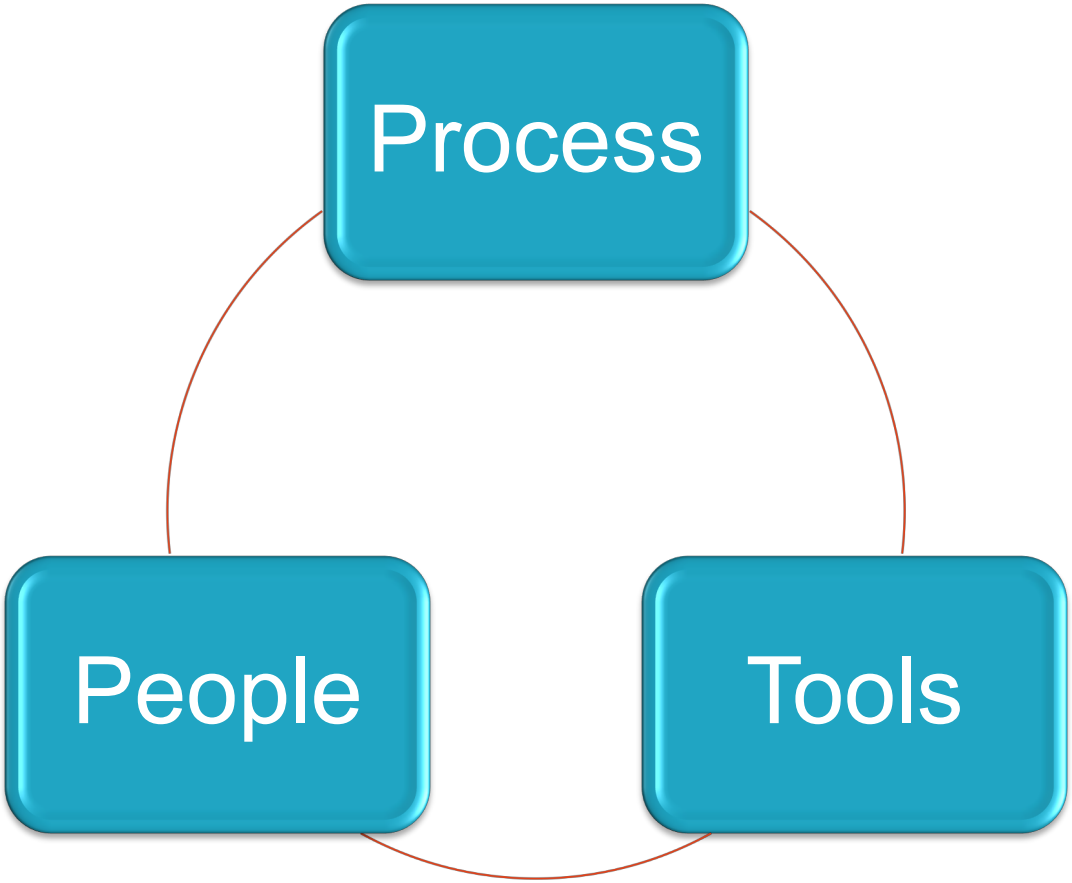
COPYRIGHT ©2019 MANICODE SECURITY

DevSecOps is a Journey…



COPYRIGHT ©2018 MANICODE SECURITY

So…What *is* DevSecOps?



…and some buzzwords

COPYRIGHT ©2019 MANICODE SECURITY

Enabling DevSecOps Through *People*

COPYRIGHT ©2019 MANICODE SECURITY

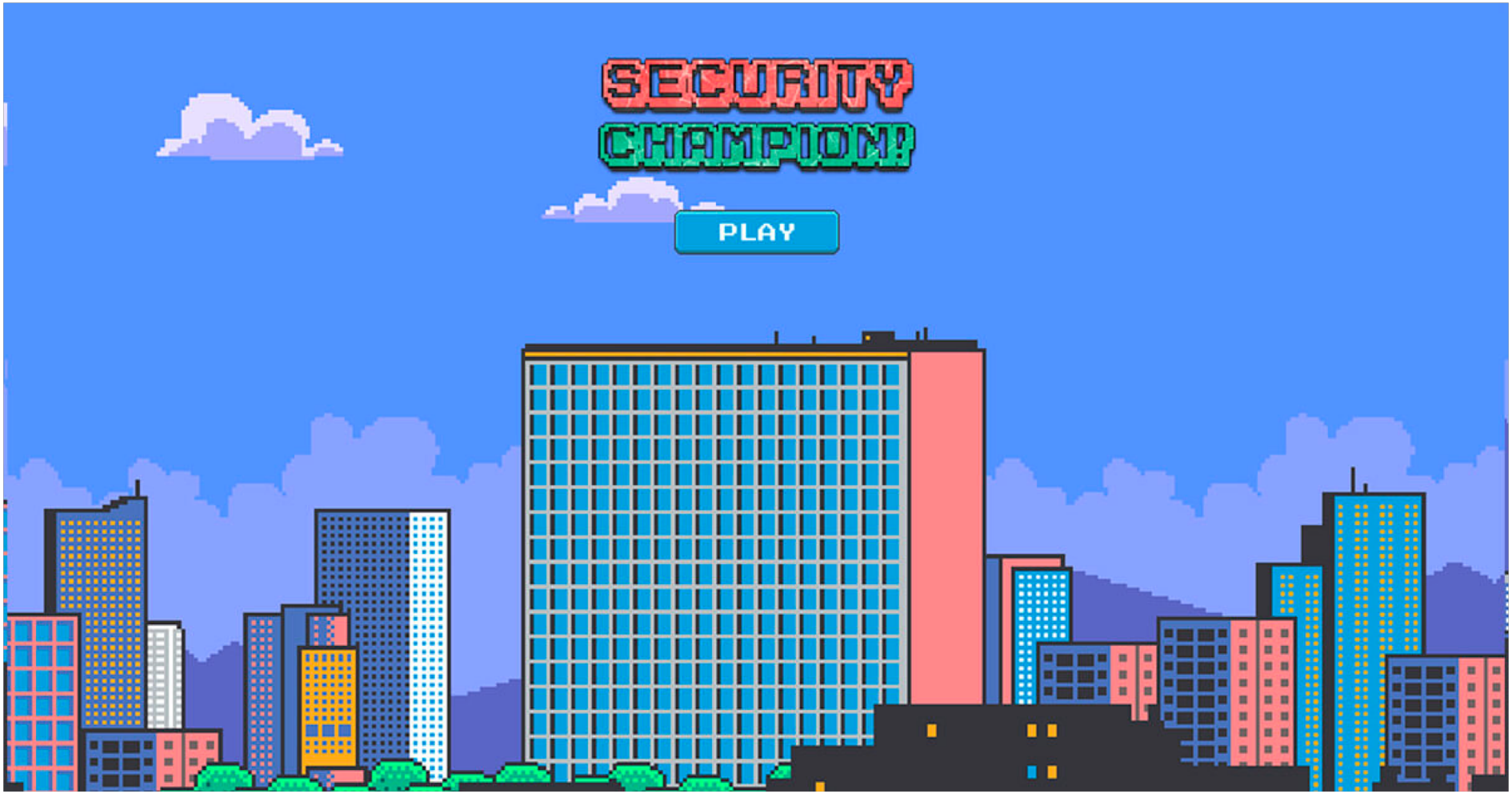
Break Down the Silos



COPYRIGHT ©2019 MANICODE SECURITY



Build a Team Beyond Yourself



COPYRIGHT ©2019 MANICODE SECURITY

Be Approachable



COPYRIGHT ©2019 MANICODE SECURITY

Train Others



COPYRIGHT ©2019 MANICODE SECURITY

Radical Transparency



COPYRIGHT ©2019 MANICODE SECURITY

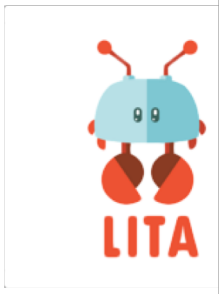
Communication and Collaboration

- Critical piece to the DevOps puzzle
- A culture of trust and empowerment makes for a healthy workplace
- Move towards shipping software faster and more confidently
- Embrace cross-team communication and training
- Feedback available from each step of the pipeline
- Security is a great fit in modern DevOps cultures











COPYRIGHT ©2019 MANICODE SECURITY

Case Study: The ”Two Pizza” Team



COPYRIGHT ©2019 MANICODE SECURITY

Enabling DevSecOps Through *Process*

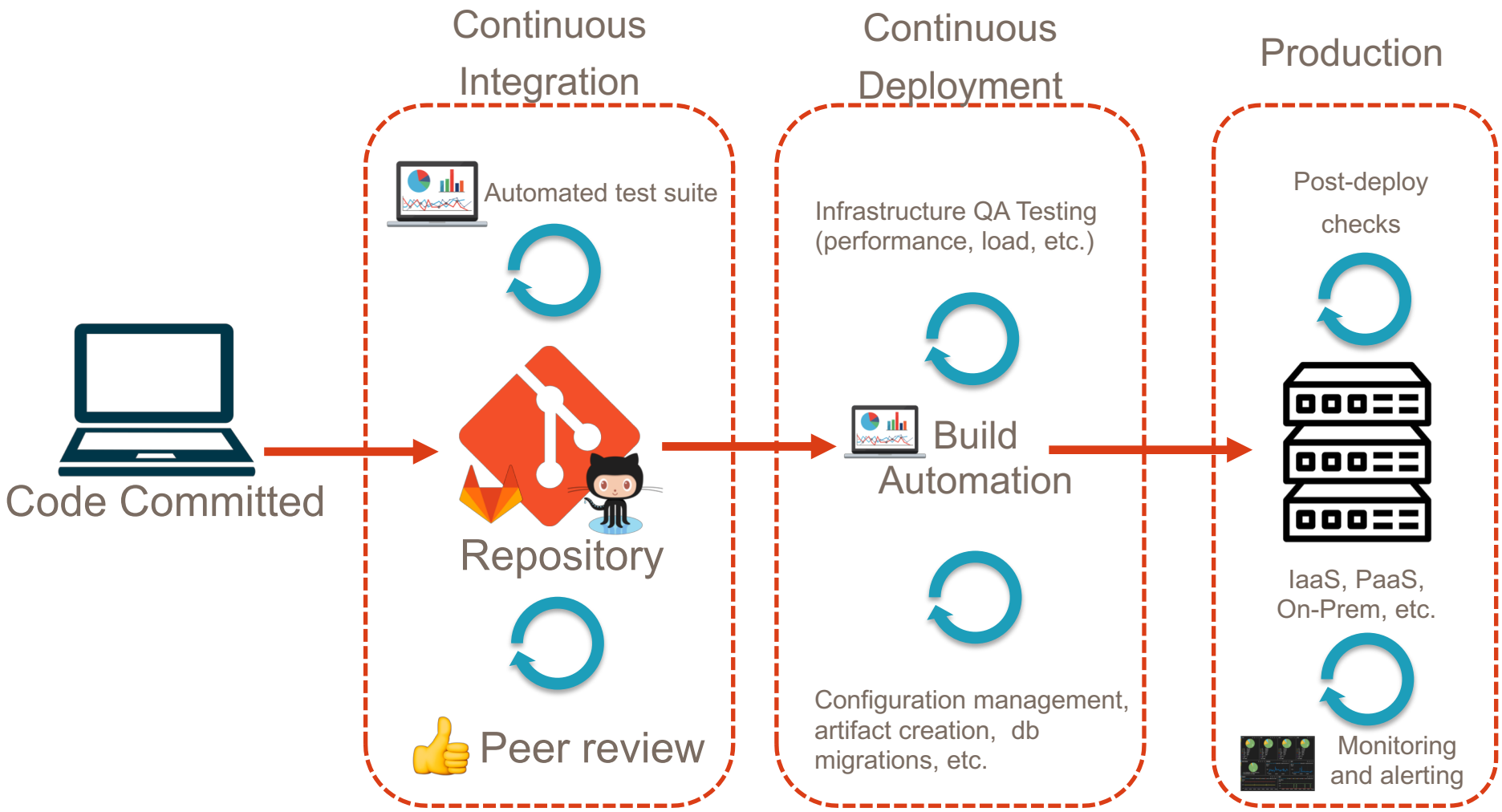
COPYRIGHT ©2019 MANICODE SECURITY

DevOps Processes

- Automate ***building*** the dev and production environment
- Automate software ***testing*** (including security)
- Automate ***deploying*** software and services
- Automate ***monitoring*** and ***alerting***
- ***Tune*** your tools to become more automated and hands-off
- Build the pipeline ***slowly*** and don't fear failure!
- Be careful with ***sensitive areas*** which are difficult to automate (access control, biz logic, complex actions)

COPYRIGHT ©2019 MANICODE SECURITY

DevOps Pipeline



COPYRIGHT ©2019 MANICODE SECURITY

Development

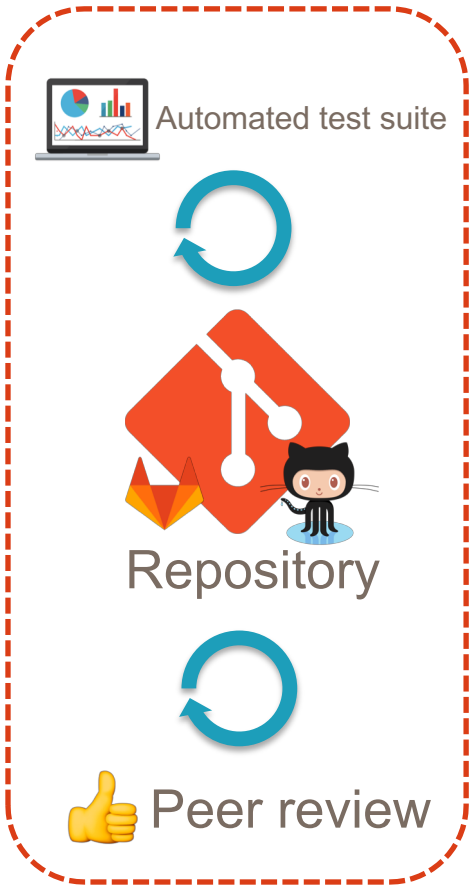


Code Committed

- Code is written by developers using an Agile methodology (use-cases, stories, etc.)
- Git branching methods to reduce releases into smaller deployable units
- Developers need an environment as similar to production as possible on their local machines
- Teams work fast and push code often
- Feedback loops and communication platforms are critical

COPYRIGHT ©2019 MANICODE SECURITY

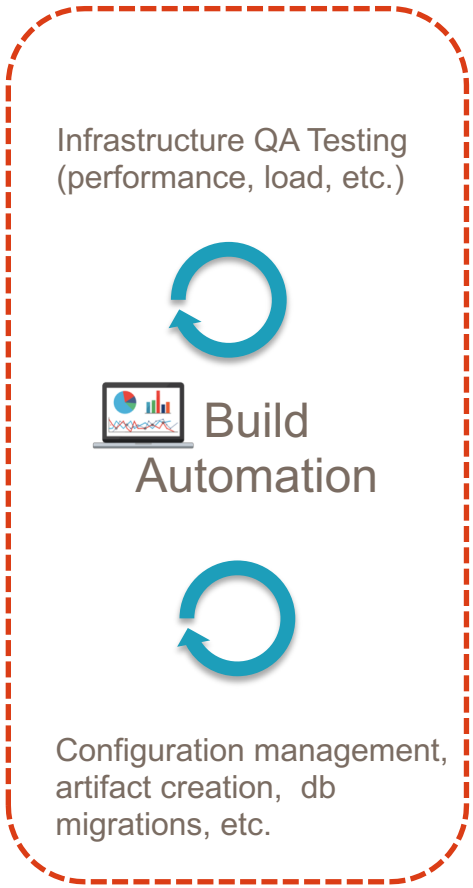
Continuous Integration



- Process of rapidly incorporating new features into software
- Code is committed by developers into a shared repository using some flavor of Git branching
- Automated checks are fired off ever time a commit is made
- Errors are detected quickly and the feedback loop is kept tight
- Code must be self-testing
- Keep builds fast and make CI output available to everyone

COPYRIGHT ©2019 MANICODE SECURITY

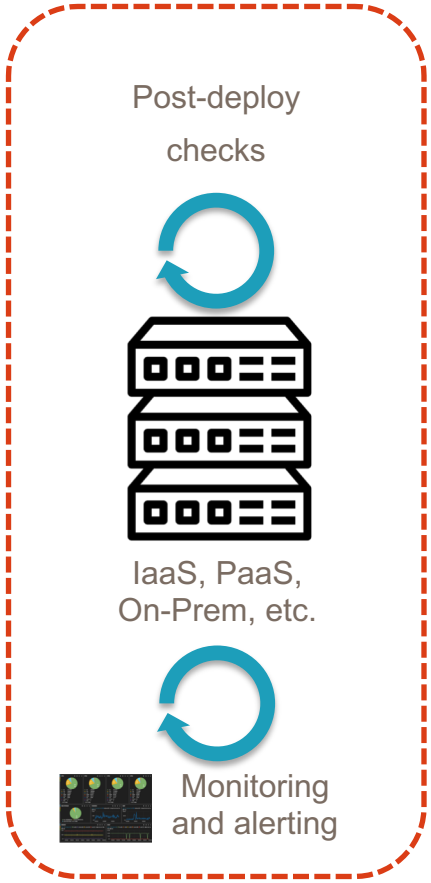
Continuous Deployment



- Process of rapidly releasing new software into production
- Only triggered after Continuous Integration steps have completed
- Can range from automatically constructing infrastructure, building container artifacts, or performing database migrations
- Includes testing infrastructure for performance and capacity
- Continuous Delivery implies a manual deployment while Continuous Deployment is automated end-to-end

COPYRIGHT ©2019 MANICODE SECURITY

Production



- Infrastructure that serves the software
- Can be bare metal, PaaS, SaaS, IaaS, or a hybrid
- Closely monitored by teams for performance, anomalies, security, etc.
- Should allow for zero-downtime deploys
- BC/DR and rollback procedures in place
- Programmable Infrastructure is an important piece to building a DevOps pipeline

COPYRIGHT ©2019 MANICODE SECURITY

DevSecOps Pipelines



COPYRIGHT ©2019 MANICODE SECURITY



COPYRIGHT ©2019 MANICODE SECURITY

Key Goals of DevSecOps Pipelines

- Optimize the critical resource: **Security personnel**
- Automate things that don't require a human brain
- Drive up consistency
- Increase tracking of work status
- Increase flow through the system
- Increase visibility and metrics
- Reduce any dev team friction with application security



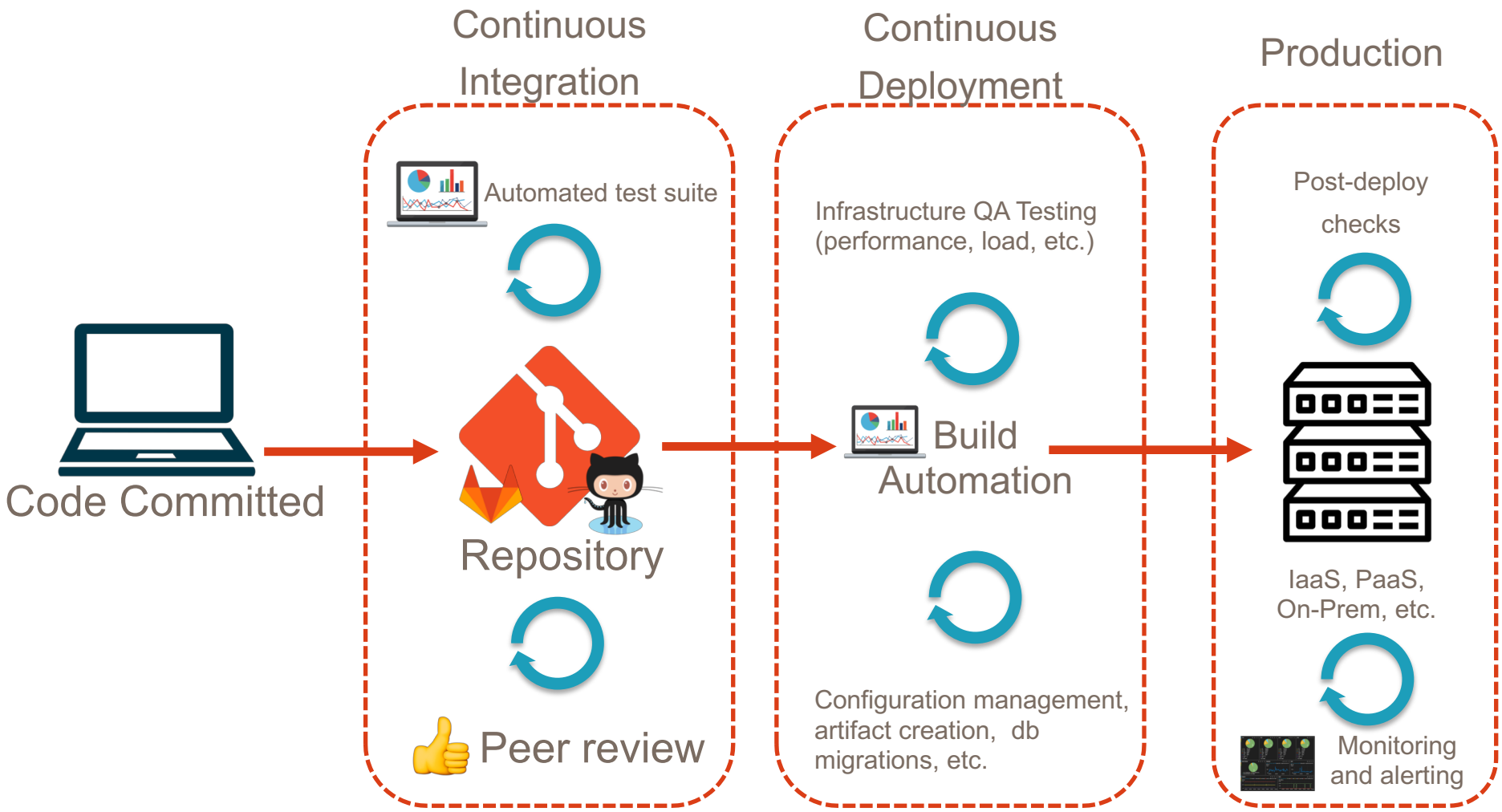
COPYRIGHT ©2018 MANICODE SECURITY

Why we like AppSec Pipelines

- Allow us to have visibility into WIP
- Better understand/track/optimize flow of engagements
- Average static test takes ...
- Great increase in consistency
- Easier re-allocation of engagements between staff
- Knowing who has what allows for more informed “cost of switching” conversations
- Flexible enough for a range of skills and app maturity

COPYRIGHT ©2018 MANICODE SECURITY

Pipeline Security



COPYRIGHT ©2019 MANICODE SECURITY

Development (Pre-Commit)



Code Committed

- Developer laptops are the first line of defense in a DevSecOps pipeline
- Moving security to the left prevents costly mistakes and vulnerabilities later
- Required Git pre-commit hooks can offer a simple, effective feedback loop
  - Static analysis scans in the IDE
  - Peer review from security engineers
  - Lightweight, threat modeling in sensitive areas

COPYRIGHT ©2019 MANICODE SECURITY

Git-Secrets

https://github.com/awslabs/git-secrets

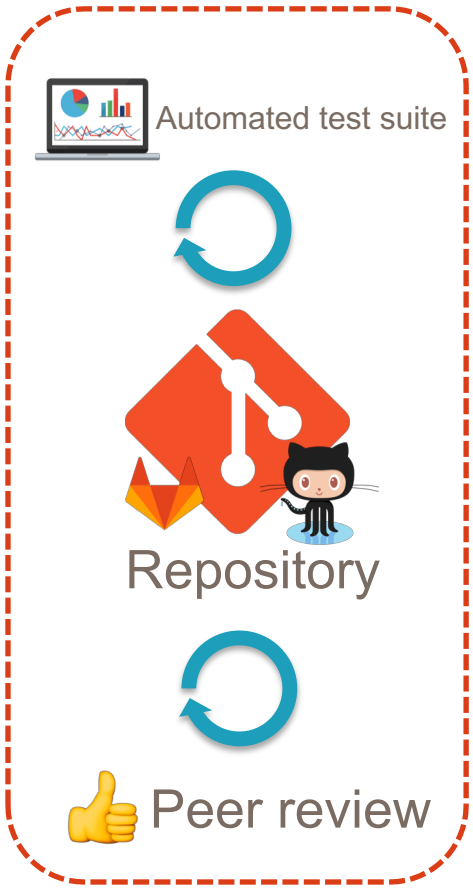
COPYRIGHT ©2019 MANICODE SECURITY

Brakeman Static Scanning (Git Pre-Commit Hook)



COPYRIGHT ©2019 MANICODE SECURITY

Continuous Integration (Commit Stage)



- Basic automated testing is performed after a commit is made
- Must be quick and offer instant feedback
- Key place to include security checks that run in parallel with integration tests, unit tests, etc.
  - Identify risk in third-party components
  - ***Incremental*** static security scanning
  - Alerting on changes to high-risk areas
  - Digital signatures for binaries

COPYRIGHT ©2019 MANICODE SECURITY

Continuous Integration (Commit Stage)







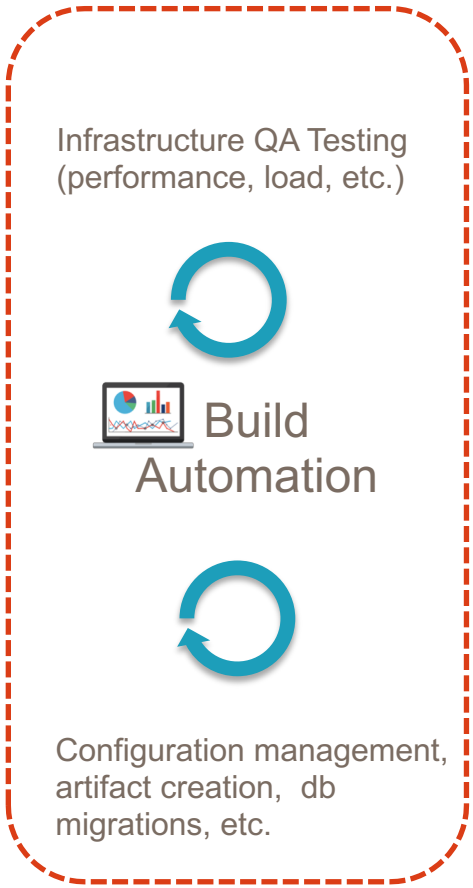




- CI server may include a dedicated security worker
- Third-party dependency checking performed in CI
  - OWASP Dependency Check
  - Node Security Project
  - Bundler-Audit
  - SRC:CLR
- Custom alerts set on repositories and sent to “on-call” security teams
  - Is someone changing pw hashing algorithm?
  - Is a new password policy enabled?

COPYRIGHT ©2019 MANICODE SECURITY

Continuous Deployment (Acceptance)



- Triggered by successful commit and passing build
- Utilize parallel, out-of-band processes for heavyweight security tasks
- IaaS and Config Management should provision latest, known-good environment state (as close to production as possible)
- Security checks during acceptance:
  - Comprehensive fuzzing
  - Dynamic Scanning (DAST)
  - Deep static analysis
  - Manual security testing

COPYRIGHT ©2019 MANICODE SECURITY

Continuous Deployment (Acceptance)



- Zap Baseline scan incorporated into CI stage of the deployment pipeline
- Runs a basic scan scan from a simple Docker run command
- By default will output all results of passive scan rules
- Highly configurable but still struggles in certain areas

https://github.com/zaproxy/community-scripts/tree/master/api/mass-baseline

COPYRIGHT ©2019 MANICODE SECURITY

Production (Post-Deployment)



- After all security checks have passed and deployment is complete
- Security teams job does not stop here:
  - Monitoring and Alerting
  - Runtime Defense (RASP)
  - Red Teaming
  - Bug Bounties
  - External Assessments
  - Web Application Firewalls
  - Vulnerability Management

COPYRIGHT ©2019 MANICODE SECURITY

Monitoring and Alerting







COPYRIGHT ©2019 MANICODE SECURITY

Web Application Firewall





modsecurity

Open Source Web Application Firewall

COPYRIGHT ©2019 MANICODE SECURITY

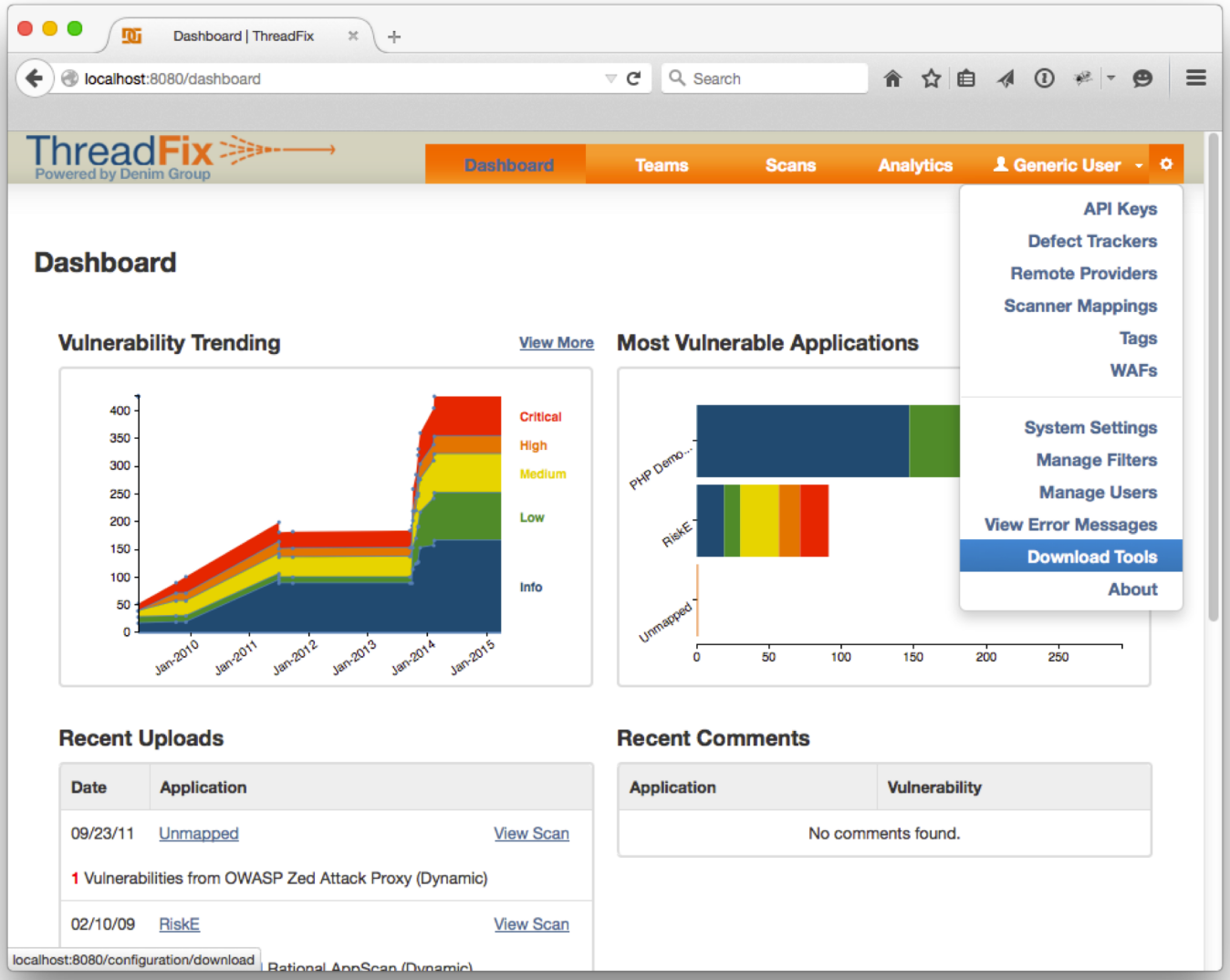
Case Study - Netflix



- The original DevSecOps Unicorn
- “Freedom and Responsibility” model in engineering
- Early cloud adopter
- No operations or systems engineers aka NoOps
- Heavy cross-team shared responsibility model
- Code is traced end-to-end
- Tooling to increase security visibility
- Compartmentalization

COPYRIGHT ©2019 MANICODE SECURITY

Vulnerability Management



COPYRIGHT ©2019 MANICODE SECURITY

Bug Bounties

bugcrowd



hackerone

COPYRIGHT ©2019 MANICODE SECURITY

How can you change your current pipeline to become more security-centric?

COPYRIGHT ©2019 MANICODE SECURITY

Enabling DevSecOps Through *Technology*

COPYRIGHT ©2019 MANICODE SECURITY

Where are we going?

- Infrastructure as a Service
- Identity and Access Management
- Network and Data Security
- Logging and Monitoring
- Containers and Microservices

COPYRIGHT ©2019 MANICODE SECURITY

Infrastructure



COPYRIGHT ©2019 MANICODE SECURITY

Infrastructure-as-a-Service

"Infrastructure as a Service, sometimes abbreviated as IaaS, contains the basic building blocks for cloud IT and typically provide access to networking features, computers (virtual or on dedicated hardware), and data storage space. Infrastructure as a Service provides you with the highest level of flexibility and management control over your IT resources and is most similar to existing IT resources that many IT departments and developers are familiar with today.



COPYRIGHT ©2019 MANICODE SECURITY

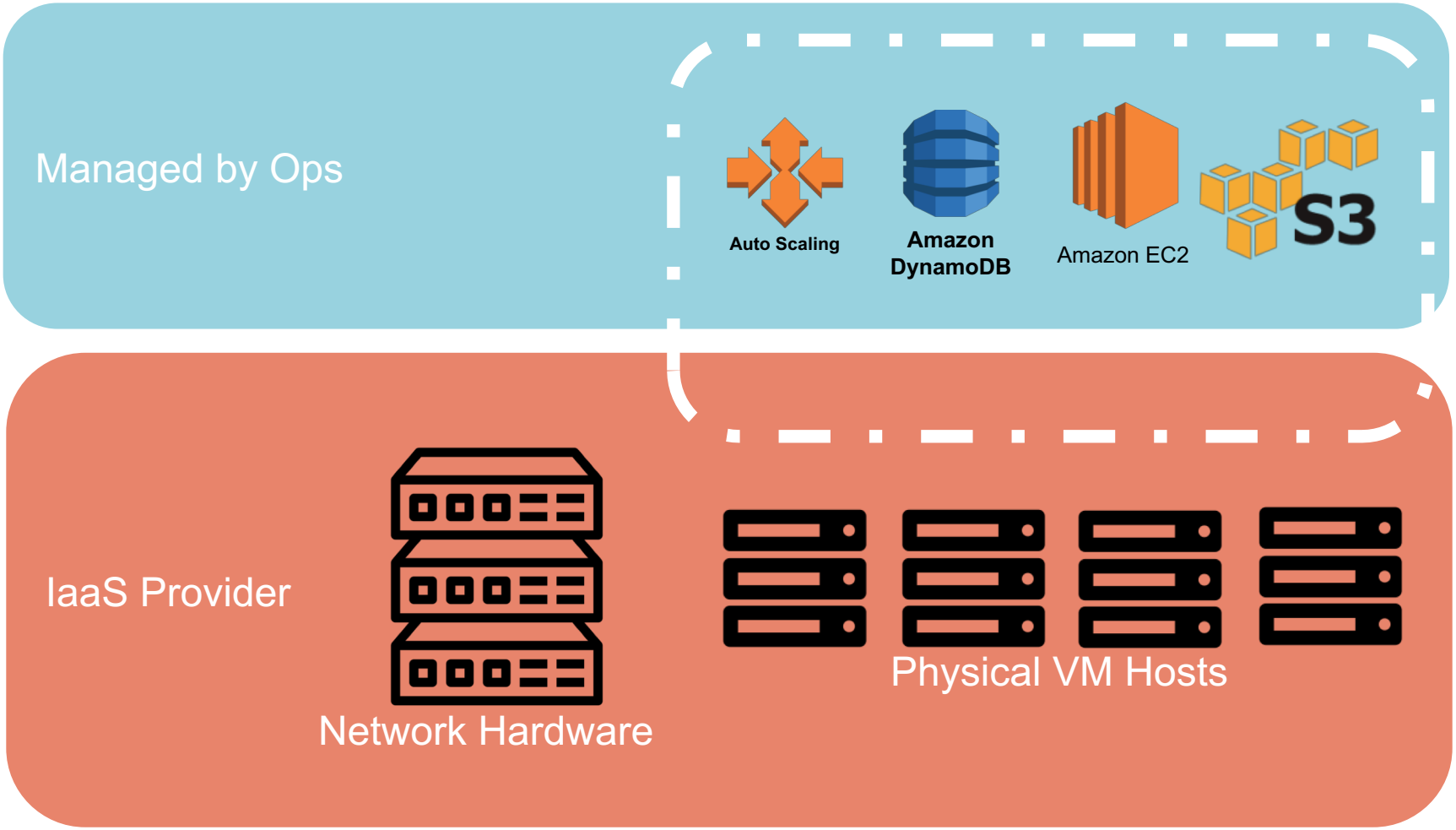
Infrastructure-as-a-Service (IaaS)

- Delivery of a complete computing foundation
  - Servers (virtualized, physical, or “serverless”)
  - Network
  - Storage
- Infrastructure is exposed to operators using a service
  - Programming network and infrastructure through APIs vs. buying and building physical hardware
- Can be operated by a third-party, hosted in-house (K8s), or a hybrid model

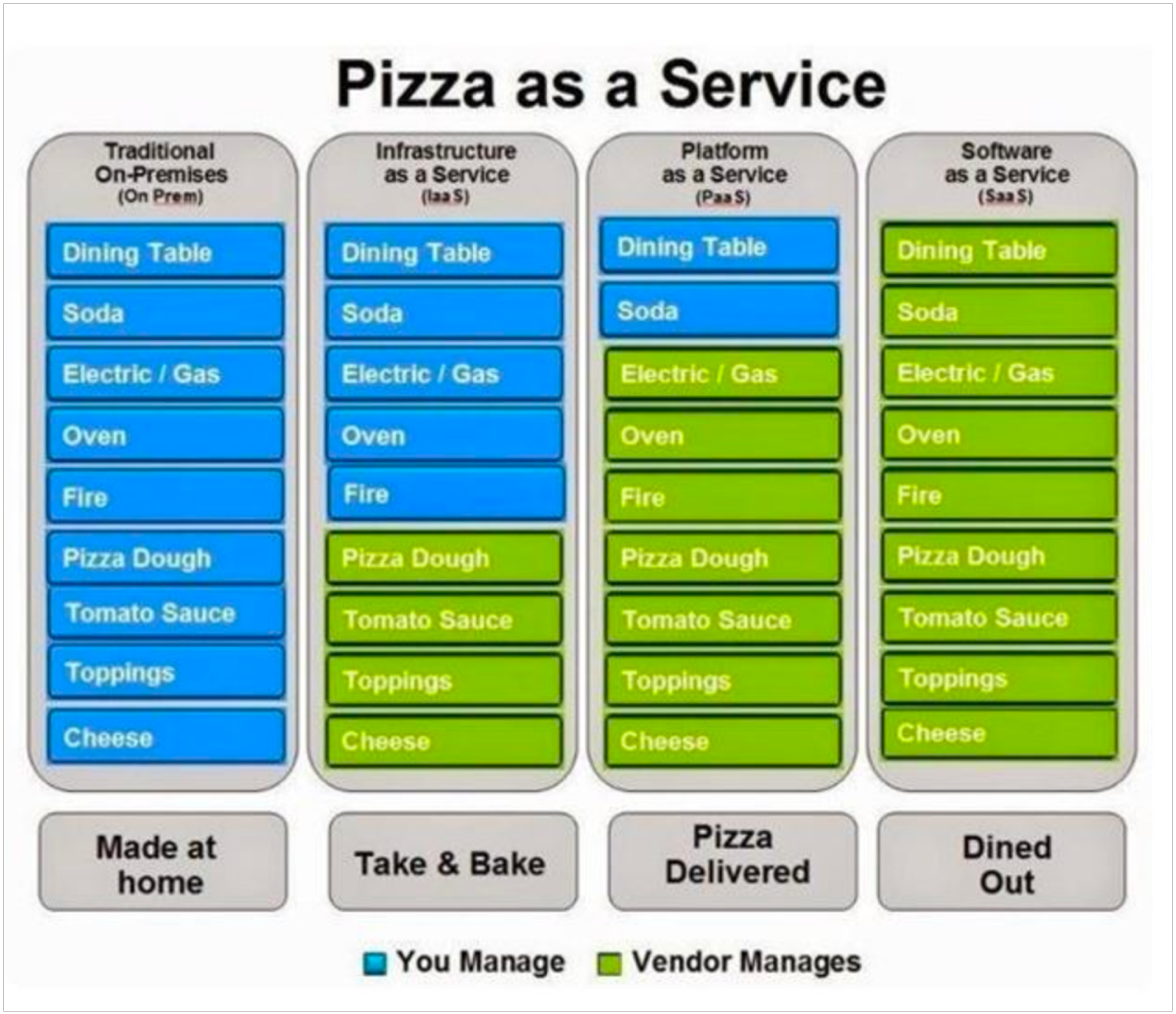


COPYRIGHT ©2019 MANICODE SECURITY

Infrastructure-as-a-Service



COPYRIGHT ©2019 MANICODE SECURITY



COPYRIGHT ©2019 MANICODE SECURITY

IaaS Benefits

- Allows organizations to scale quickly without the large capital expense of building a data center
- May be a cost effective approach compared to owning hardware
- If demand is volatile, IaaS can scale up and down elastically
- IaaS availability SLAs
- Many providers offer advanced services such as logging, monitoring, machine learning, appliances, etc. that integrate directly



COPYRIGHT ©2019 MANICODE SECURITY

IaaS Providers









Google Cloud Platform









COPYRIGHT ©2019 MANICODE SECURITY

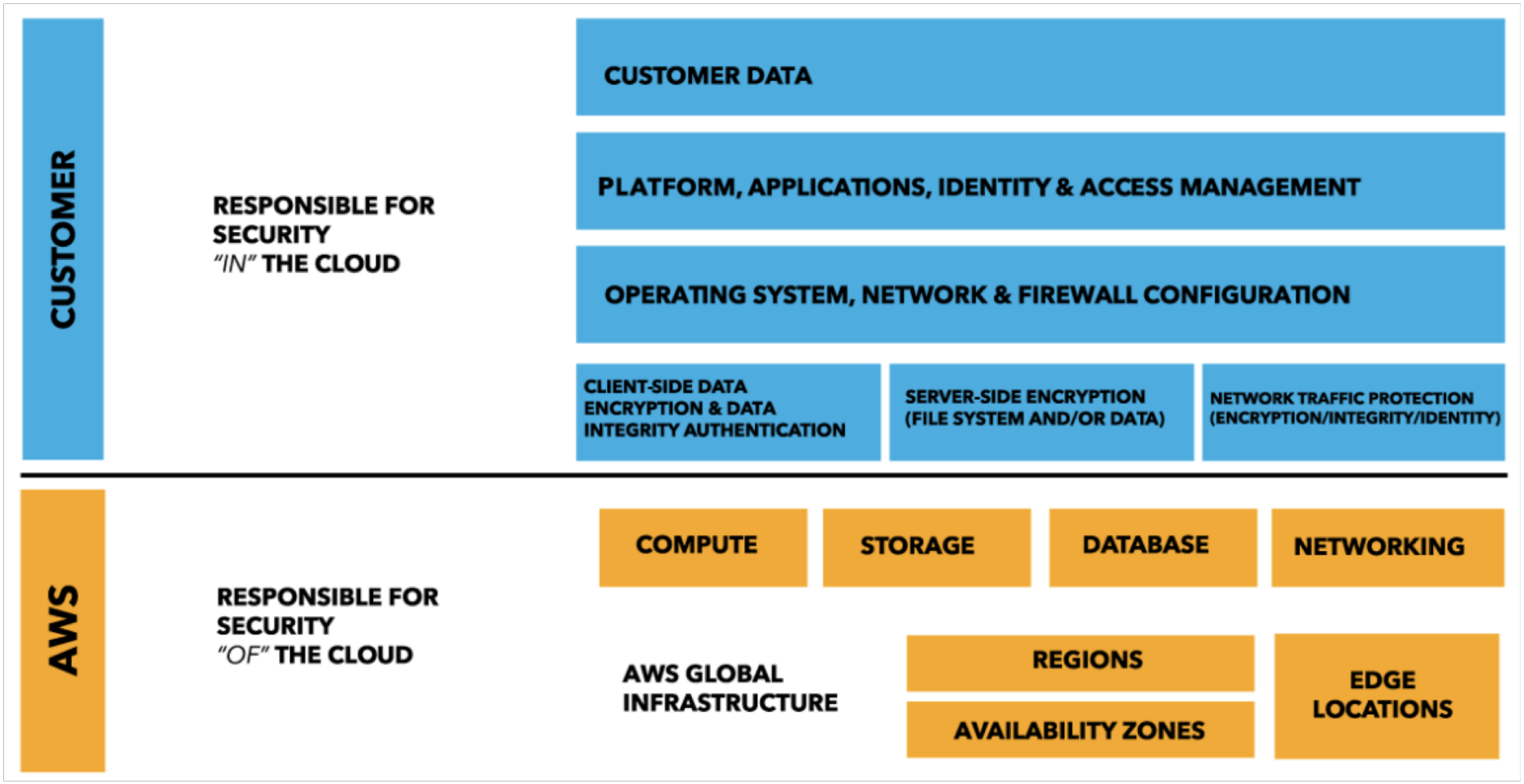
IaaS Security Considerations

- "The Cloud" doesn't *do* security for you – this is your responsibility
- Access Control
- Auditing Capabilities
- Insider Threats
- Compliance Requirements
  - SOC, PCI, HIPAA, etc.
- Historic Security Performance
- Encryption Capabilities
- Third-Party Certificates and Audits
- Secrets Storage, Built-in Security Features, etc.



COPYRIGHT ©2019 MANICODE SECURITY

AWS Shared Security Responsibility Model



COPYRIGHT ©2019 MANICODE SECURITY

The Cloud Won't Protect You

Security

5

Dow Jones index – of customers, not prices – leaks from AWS repo

S3 bucket was set to authenticate *all* AWS users, not just Dow Jones users

CNN tech BUSINESS CULTURE GADGE

Verizon confirmed on Wednesday the personal data of 6 million customers has leaked online.

The security issue, uncovered by research from cybersecurity firm UpGuard, was caused by a misconfigured security setting on a cloud server due to "human error."

The error made customer phone numbers, names, and some PIN codes publicly available online. PIN codes are used to confirm the identity of people who call for customer service.

198 million Americans hit by 'largest ever' voter records leak

Personal data on 198 million voters, including analytics data that suggests who a person is likely to vote for and why, was stored on an unsecured Amazon server.

COPYRIGHT ©2019 MANICODE SECURITY

Identity and Access Management

COPYRIGHT ©2019 MANICODE SECURITY

Identity and Access Management

Code Spaces goes FOREVER after attacker NUKES its Amazon-hosted data

Source-sharing site to close following total cloudpocalypse

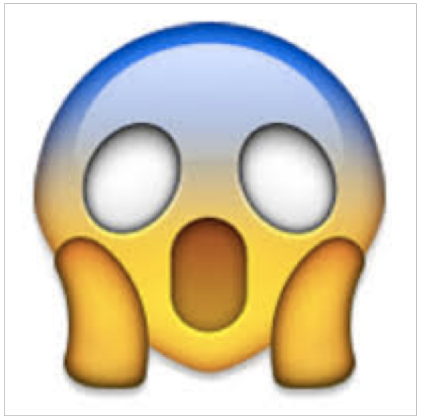
By Neil McAllister in San Francisco 18 Jun 2014 at 20:54 SHARE ▼



COPYRIGHT ©2019 MANICODE SECURITY

Identity and Access Management

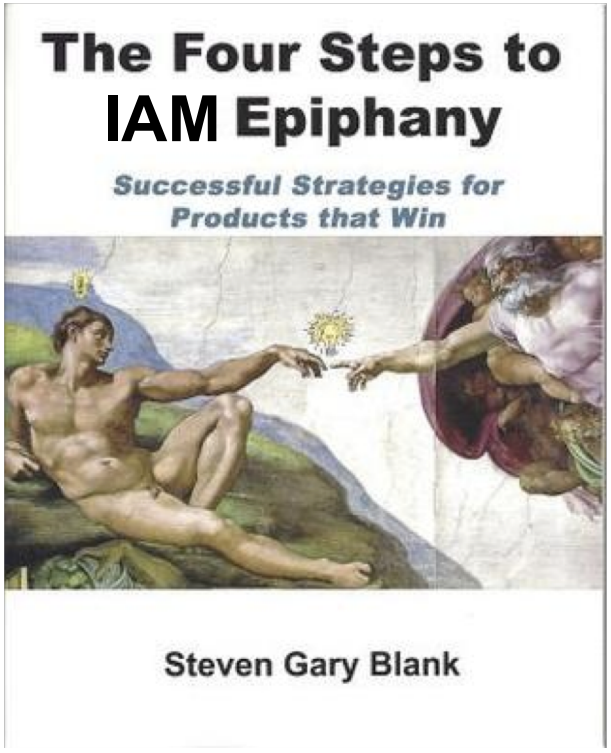
"We finally managed to get our panel access back but not before he had removed all EBS snapshots, S3 buckets, all AMI's, some EBS instances and several machine instances," the company wrote in a message posted to its homepage. **"In summary, most of our data, backups, machine configurations and offsite backups were either partially or completely deleted."**



COPYRIGHT ©2019 MANICODE SECURITY

The Four Steps to IAM Epiphany

- Lock Down Root Account
- Force Strong Authentication Mechanisms
- Properly Configure IAM Policies
- Monitor and Alert on Suspicious Behavior



COPYRIGHT ©2019 MANICODE SECURITY

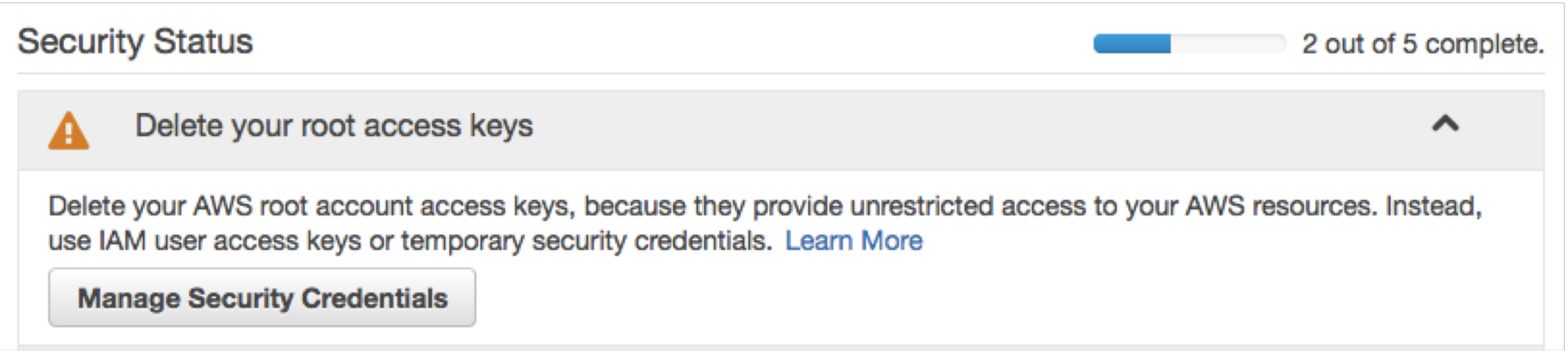
Lock Down Root Account



COPYRIGHT ©2019 MANICODE SECURITY

Lock Down Root Account

- Most Cloud environments have a Root account
- Grants full access to all your resources for all services, including billing information
- Permissions ***cannot*** be restricted for this key
- This IS the proverbial ”Key to the Kingdom”



COPYRIGHT ©2019 MANICODE SECURITY

Lock Down Root Account

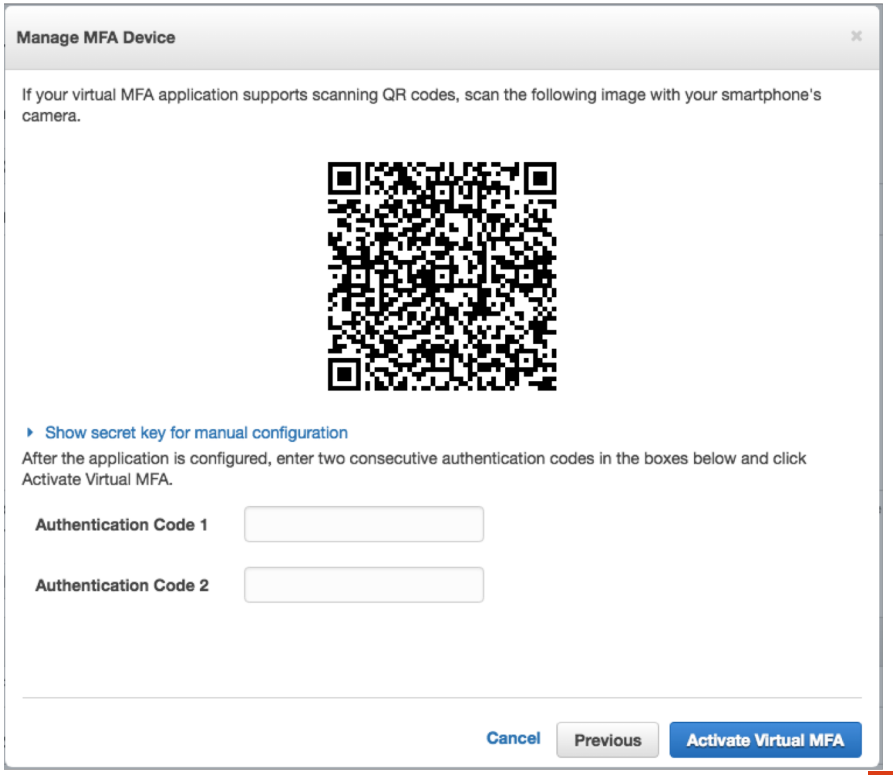
- Consider this account a “Break Glass” account
- Keep tabs on who has access to Root and audit this list regularly
- Strong password, rotated regularly, and stored in password manager
- Create and enforce written policy banning creation of Root access keys for SysAdmins
- Use a physical hardware token for MFA and place token in physical lockbox

COPYRIGHT ©2019 MANICODE SECURITY

Strong Authentication - MFA

- If credentials are compromised, require a second layer of authentication
- Can be hardware token or ”soft token”





COPYRIGHT ©2019 MANICODE SECURITY

Strong Authentication – API MFA

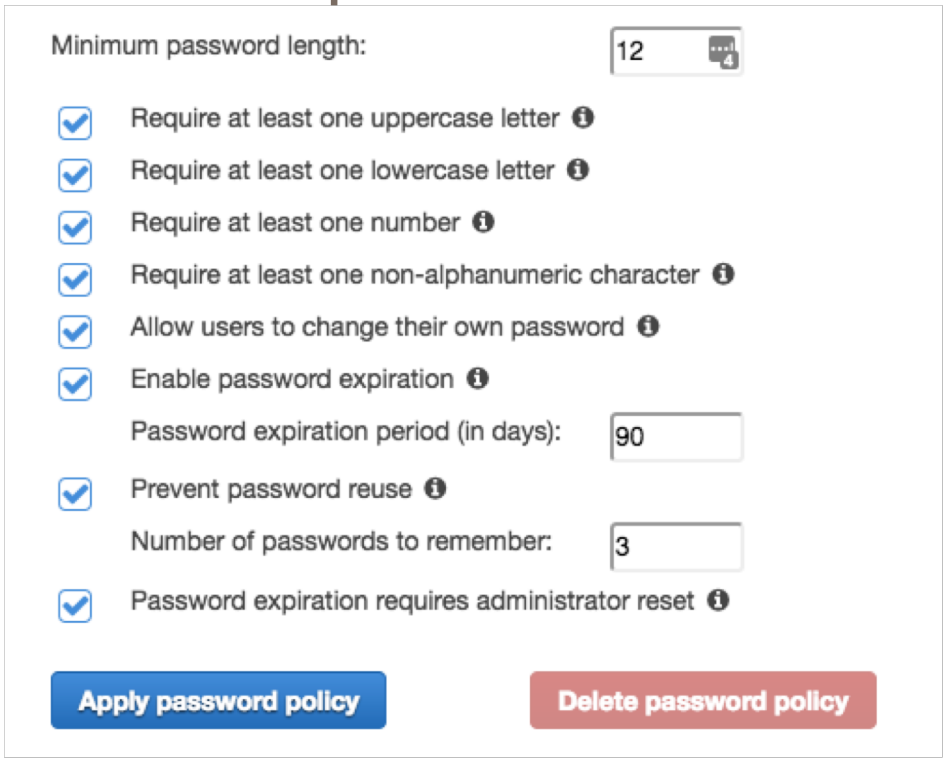
- With MFA enabled, operators must retrieve a ‘session-token’ which will grant access from the CLI for 12 hours (default)

```
→ ~ aws sts get-session-token --serial-number arn:aws:iam::771808488992:mfa/BillyTheDeveloper --token-code 925148 --profile developer
{
    "Credentials": {
        "SecretAccessKey": "SfcJcDa06j3BcvS4WhdV1jQqG1/taHkavRQE0khd",
        "SessionToken": "FQoDYXdzEH8aDGY6WLhhxLFSo/7XVSKwAeVIKukOkOdBVBBbpacyKGSgpImRTPDjbI8uwVZMSypINnrUIjCmWkFPw0vLWKLyXHUI8McvTls7nwBFjo9+XH0Y7
TdCH/ho/8Xrz974iygN64BnsizcMeCzlxq31NznWjs7HnQx9H37/qMr5dUIah0uMuLQ0WUs+HVyrvvRKqluPoMznS1jV0dst+CfaEYG/1W/bNtuMF/n6210mS58EzEZNlFsDb4Hw+eFnpfxE88
fKOqL9MsF",
        "Expiration": "2017-07-30T10:02:18Z",
        "AccessKeyId": "ASIAJ34333LO5POBDVQA"
    }
}
```

COPYRIGHT ©2019 MANICODE SECURITY

Strong Authentication – Password Requirements

- Granular password policy in AWS console controlled by administrators
- Align with internal policies and audit regularly



COPYRIGHT ©2019 MANICODE SECURITY

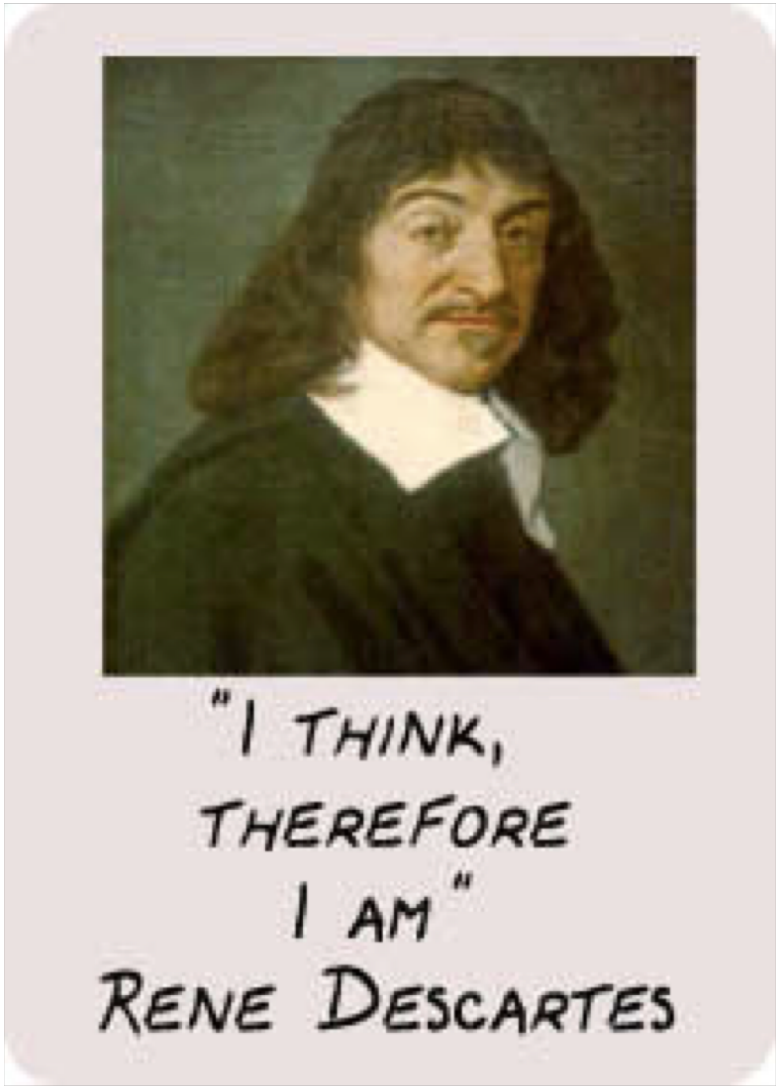
Strong Authentication – IP Restrictions

- Consider using IP restrictions when creating custom IAM policies
- Lock down to office IP range or VPN address

```
{
  "Version": "2012-10-17",
  "Statement": {
    "Effect": "Deny",
    "Action": "*",
    "Resource": "*",
    "Condition": {"NotIpAddress": {"aws:SourceIp": [
      "192.0.2.0/24",
      "203.0.113.0/24"
    ]}}
  }
}
```

COPYRIGHT ©2019 MANICODE SECURITY

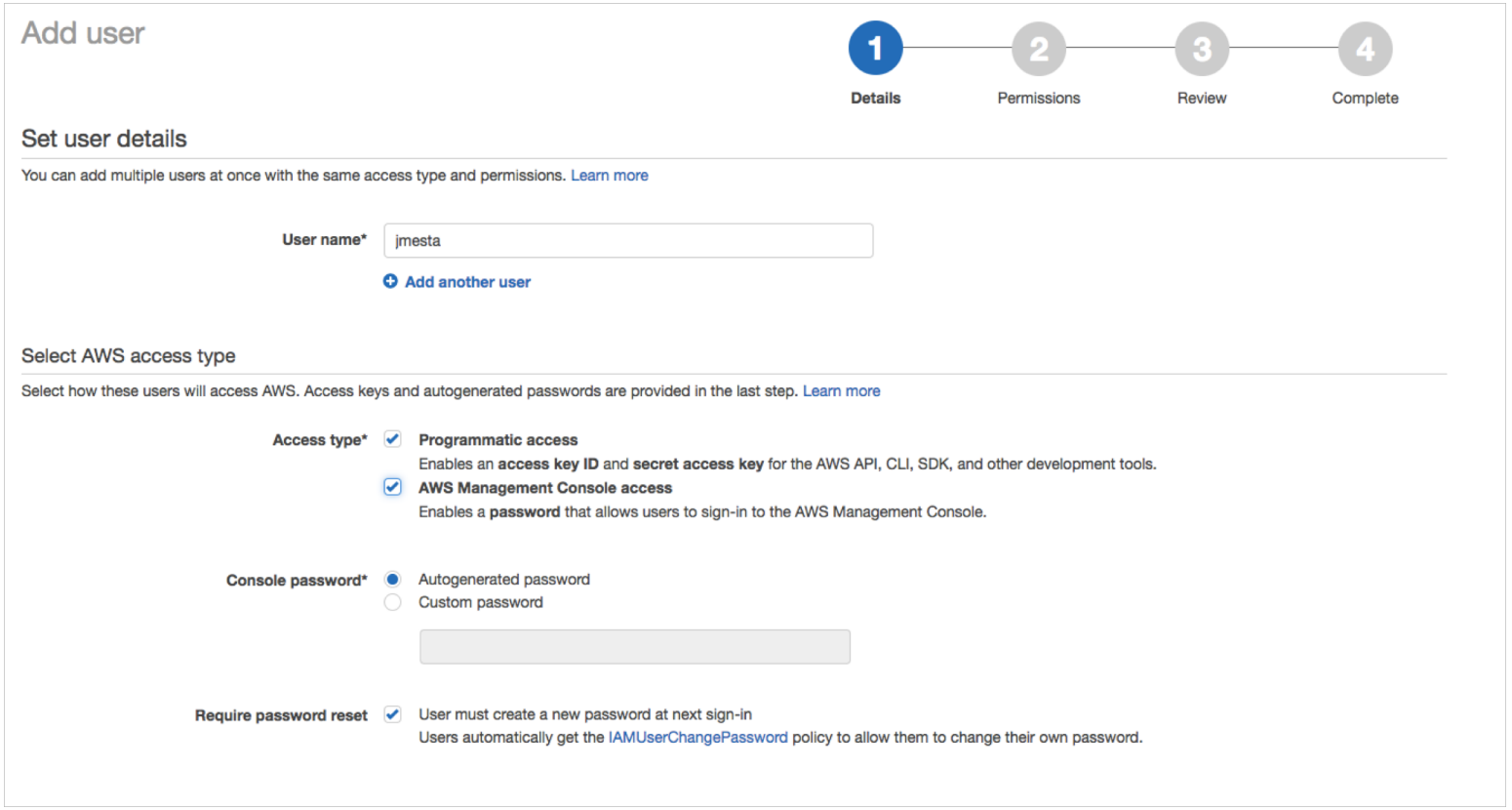
Identity and Access Management



COPYRIGHT ©2019 MANICODE SECURITY

IAM - Always Create Individual IAM Users

- It's 2019…We don't share credentials



COPYRIGHT ©2019 MANICODE SECURITY

IAM – Stock Policies

- These policies are updated by the cloud provider when new services are introduced
- Designed to support common tasks

Policy name	Type	Attachments	Description
AdministratorAccess	Job function	0	Provides full access to AWS services and resources.
AmazonAPIGatewayAdministrator	AWS managed	0	Provides full access to create/edit/delete APIs in Amazon API Gateway via the AWS Manage…
AmazonAPIGatewayInvokeFullAccess	AWS managed	0	Provides full access to invoke APIs in Amazon API Gateway.
AmazonAPIGatewayPushToCloudWatch…	AWS managed	0	Allows API Gateway to push logs to user's account.
AmazonAppStreamFullAccess	AWS managed	0	Provides full access to Amazon AppStream via the AWS Management Console.
AmazonAppStreamReadOnlyAccess	AWS managed	0	Provides read only access to Amazon AppStream via the AWS Management Console.
AmazonAppStreamServiceAccess	AWS managed	0	Default policy for Amazon AppStream service role.
AmazonAthenaFullAccess	AWS managed	0	Provide full access to Amazon Athena and scoped access to the dependencies needed to e…
AmazonCloudDirectoryFullAccess	AWS managed	0	Provides full access to Amazon Cloud Directory Service.
AmazonCloudDirectoryReadOnlyAccess	AWS managed	0	Provides read only access to Amazon Cloud Directory Service.

COPYRIGHT ©2019 MANICODE SECURITY

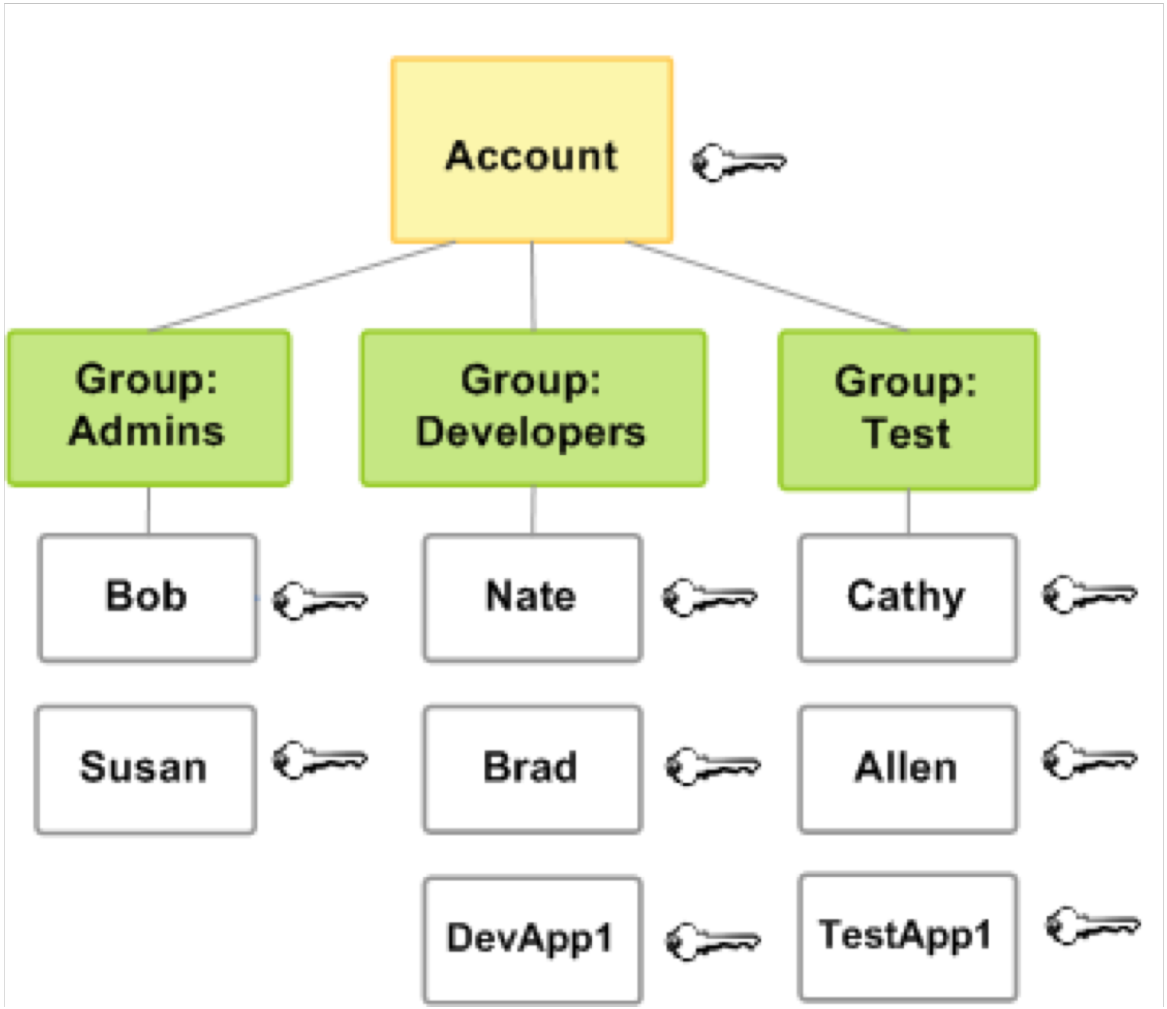
IAM – Use Groups to Assign Permissions

- Use groups to enforce least privilege access for function or business unit
- Easier to manage than individual IAM policies per user
- Attach default AWS policies as needed



COPYRIGHT ©2019 MANICODE SECURITY

IAM – Use Groups to Assign Permissions



COPYRIGHT ©2019 MANICODE SECURITY

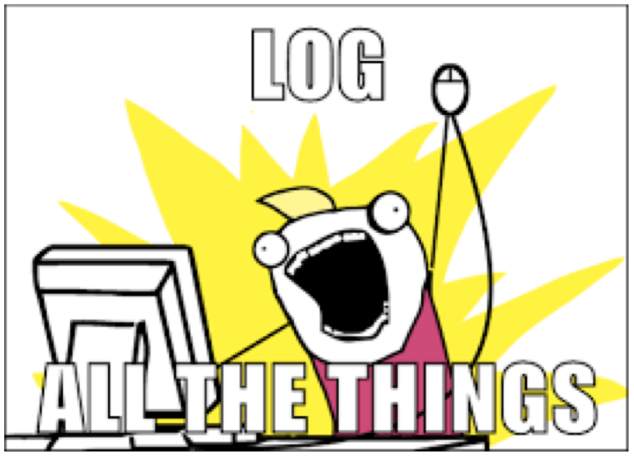
Monitoring and Alerting (IAM)



COPYRIGHT ©2019 MANICODE SECURITY

IAM - Monitoring and Alerting

- In the event of account compromise, how would we know what failed?
- Many IaaS providers have built in monitoring and alerting mechanisms out of the box
- Start simple and build your solution over time
- Focus on keeping the noise to a minimum and developing meaningful metrics



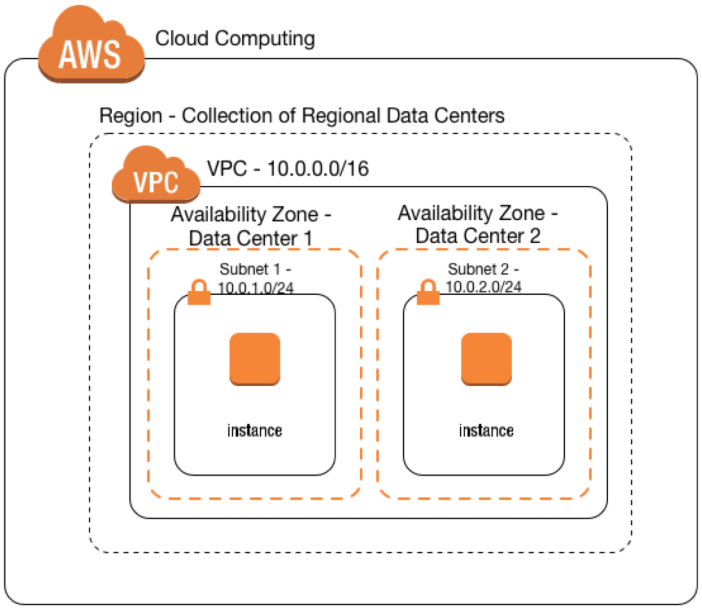
COPYRIGHT ©2019 MANICODE SECURITY

Network and Data Security

COPYRIGHT ©2019 MANICODE SECURITY

Network Security - VPC

- Provides set of contained subnets with a common CIDR block
- Like a “virtual data center” spread across multiple availability zones (AZ)
- Can be built in a variety of ways to accomplish security and scale



COPYRIGHT ©2019 MANICODE SECURITY

VPC Architecture

- Public-facing VPC
- Public and Private VPCs
- Private VPC with hardware VPN access
- Public / Private subnets with hardware VPN
- Software-based VPN
- SSH Bastion Box
- …

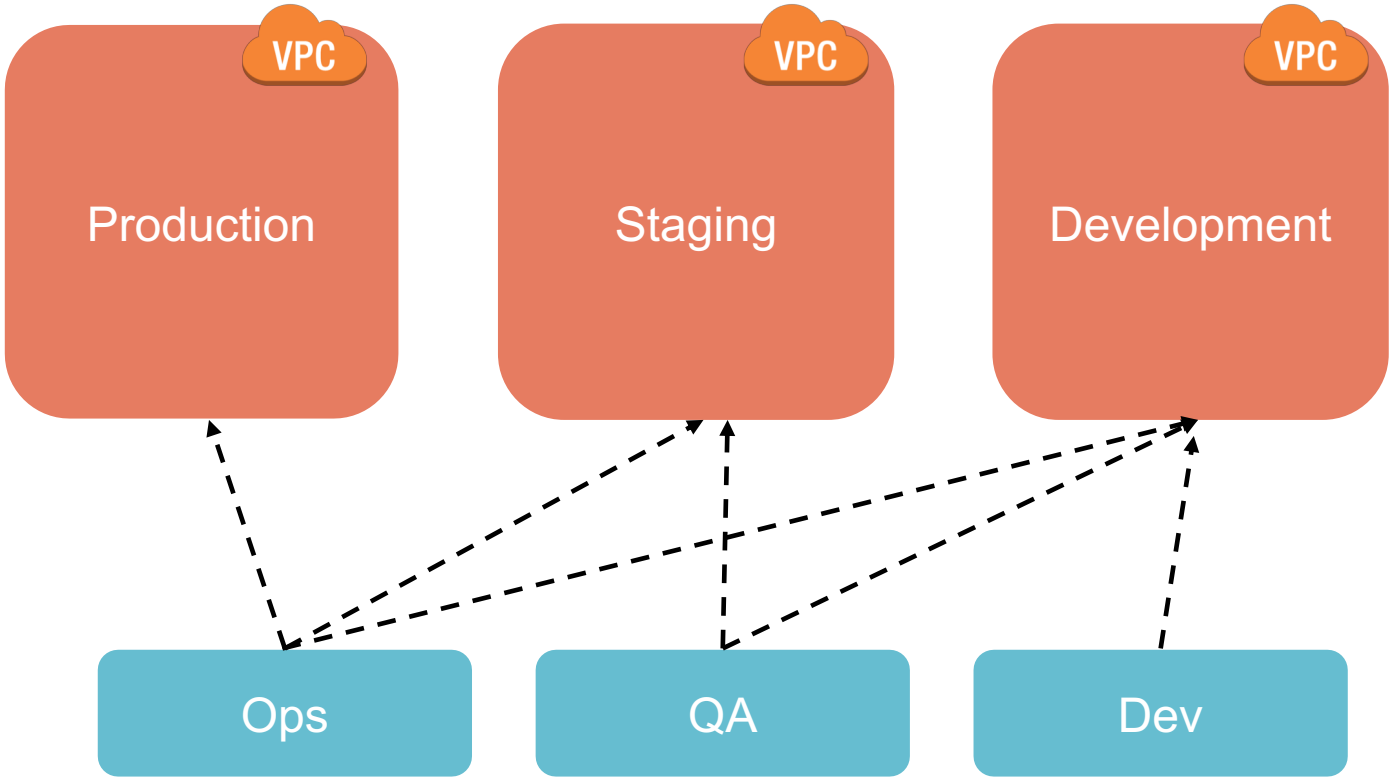
COPYRIGHT ©2019 MANICODE SECURITY

VPC Security

- Enable network logging(Flow Logs + CloudWatch)
- Configure site-to-site VPN to transfer data between regions or providers
- Consider implementing IDS / IPS
- Use private subnets and put instances behind load balancers when possible
- Outbound traffic should be proxied and restricted to only known ports / protocols (Squid, etc.)

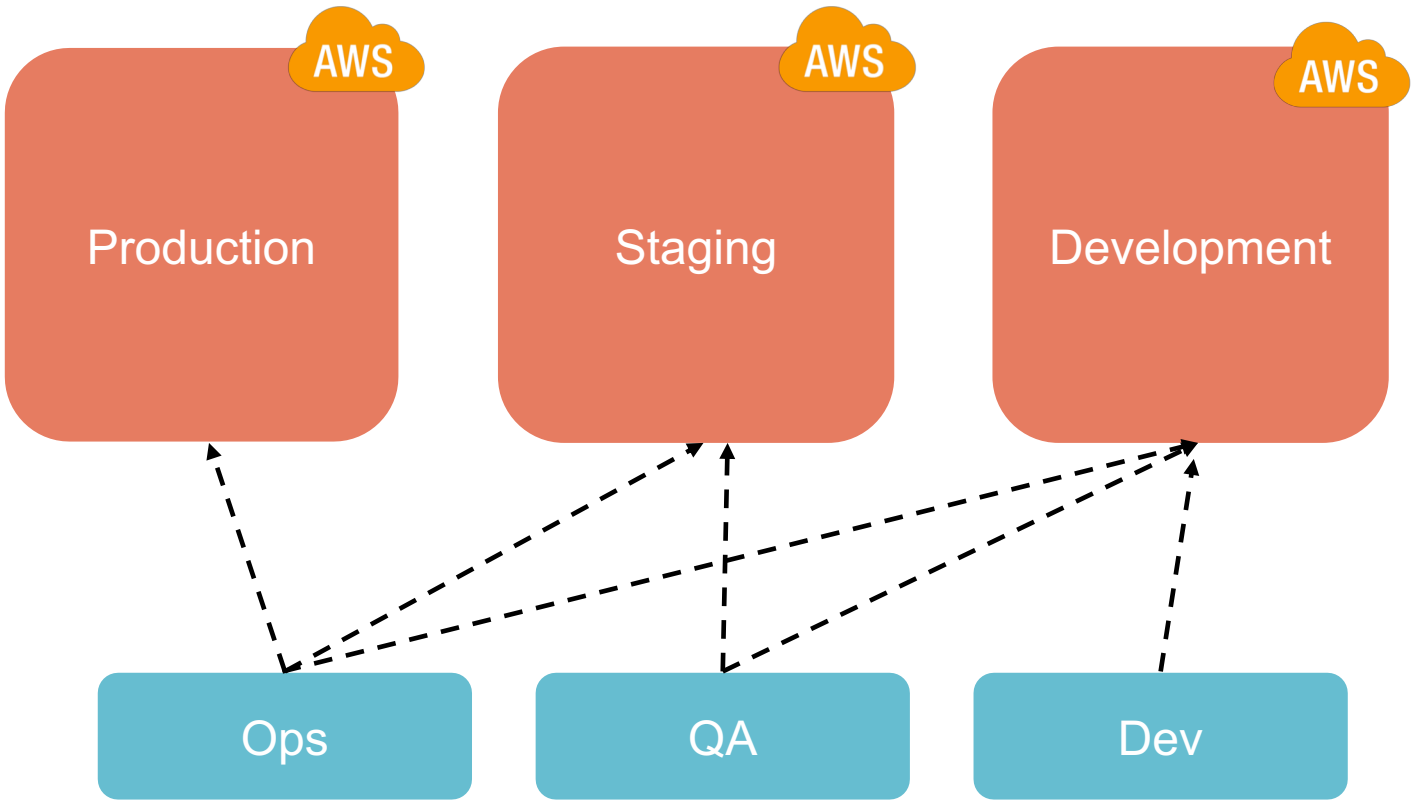
COPYRIGHT ©2019 MANICODE SECURITY

Network Segmentation – Separate VPC



COPYRIGHT ©2019 MANICODE SECURITY

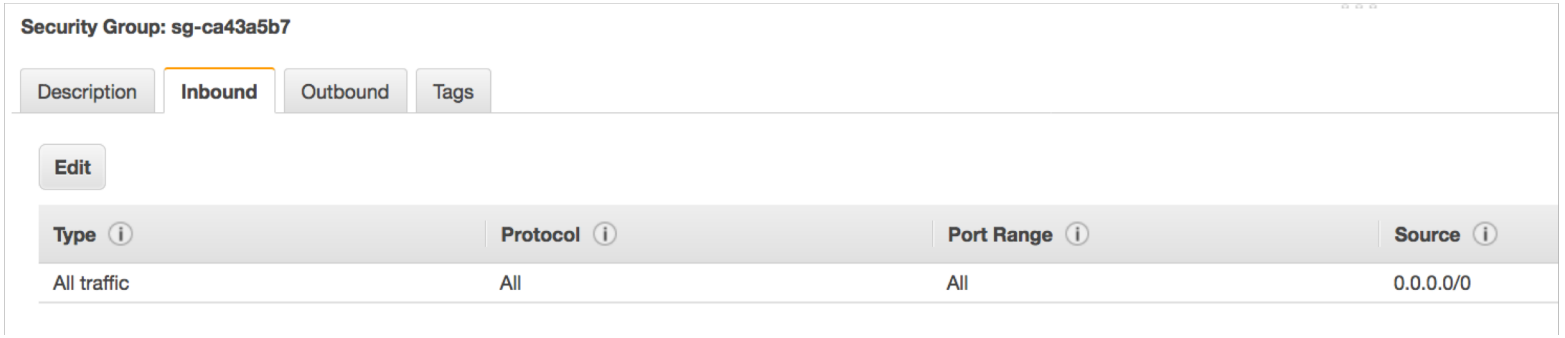
Network Segmentation - Account Isolation



COPYRIGHT ©2019 MANICODE SECURITY

Network Security – Security Groups

- ”ACLish” rules typically applied to firewalls or routers
- SG rules are applied directly to instances
- Audit often, enable monitoring and alerting
- Avoid incoming traffic from 0.0.0.0/0



COPYRIGHT ©2019 MANICODE SECURITY

Network Security – Security Groups

- In regulated environments where end-to-end encryption is required set up alerts for risky ports being opened
- Consider setting up alerts for suspicious behavior:
  - Port 21 was opened and closed in <30 minutes
  - The range of ports 100-200 were opened
  - X number of SGs were created or deleted in 24 hours
- Severely restrict who can create, modify, or delete SGs
- SGs should be version controlled and go through proper change management procedures

COPYRIGHT ©2019 MANICODE SECURITY

Data Storage

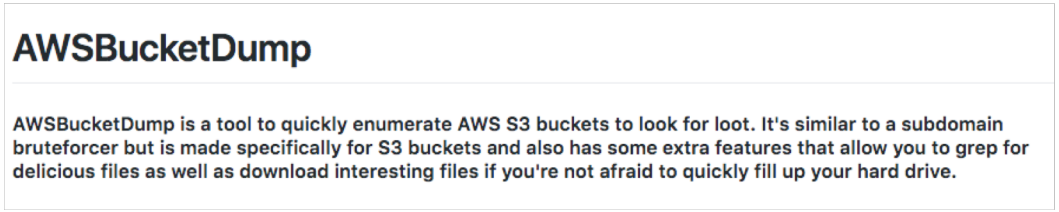


Lock your doors, people: Verizon breach on unsecured AWS server exposes 14M customer records

BY TOM KRAZIT on July 12, 2017 at 10:04 am

COPYRIGHT ©2019 MANICODE SECURITY

Data Storage



- Encryption of all instance volumes as well as object storage
- Proper authorization and authentication of databases and object storage mechanisms *cough* ***S3*** *cough*
  - This includes message queuing systems!
- Always use encrypted channels for data in transit from another cloud instance or from a local machine
- Retain access logs of object storage and alert on suspicious behavior
- Ensure default credentials, ports, etc. are locked down when deploying new infrastructure

COPYRIGHT ©2019 MANICODE SECURITY

Key Management

- Many cloud providers offer key management services built to store and distribute key pairs
  - AWS Key Management Service (KMS)
  - Google Cloud Key Management Service
- Ensure keys are backed by an Hardware Security Module (HSM)
- Keys can be used for a variety of tasks
  - SSH key pairs
  - Encrypting / Decrypting databases or volumes
  - Encrypting logs



COPYRIGHT ©2019 MANICODE SECURITY

Key Management Best Practices

- Ensure access to keys is logged and audit-friendly
  - Cloudtrail integration in AWS
- Apply least privilege / separation of duties for sensitive keys
  - IAM policy enforcement in AWS
- MFA applied for sensitive actions
- Rotate per your organizations policies
- Have an incident response procedure and practice it
  - What happens when a key is compromised?
- No keys in version control. Ever.

COPYRIGHT ©2019 MANICODE SECURITY

Distributing Secrets

- Software systems often need access to a shared credential to operate:
  - Database password
  - Third-Party API key
  - Microservices
- Secret management is full of opinions and could be a course itself
- Many options exist – Choose your own adventure!









COPYRIGHT ©2019 MANICODE SECURITY

Commandments of Sane Secret Management

- Secrets should not be written to disk in cleartext
- Secrets should not be transmitted in cleartext
- Access to secrets should be recorded
- Operator access to secrets should be limited
- Access control to secrets should be granular
- Secrets distribution infrastructure should be mutually authenticated
- Secrets should be version-controlled

COPYRIGHT ©2019 MANICODE SECURITY

HashiCorp's Vault

```
➜ devsecops vault server -dev
==> Vault server configuration:

                     Cgo: disabled
        Cluster Address: https://127.0.0.1:8201
             Listener 1: tcp (addr: "127.0.0.1:8200", cluster address: "127.0.0.1:8201", tls: "disabled")
              Log Level: info
                  Mlock: supported: false, enabled: false
       Redirect Address: http://127.0.0.1:8200
                Storage: inmem
                Version: Vault v0.7.3
            Version Sha: 0b20ae0b9b7a748d607082b1add3663a28e31b68
```

COPYRIGHT ©2019 MANICODE SECURITY

Which is the most secure way to pass secrets to an app running in a container?

1. Pass secrets as an environment variable
2. Mount volume in container that has secrets in a file
3. Build the secrets into the container image
4. Query a ”Secrets API” over your network
5. Other

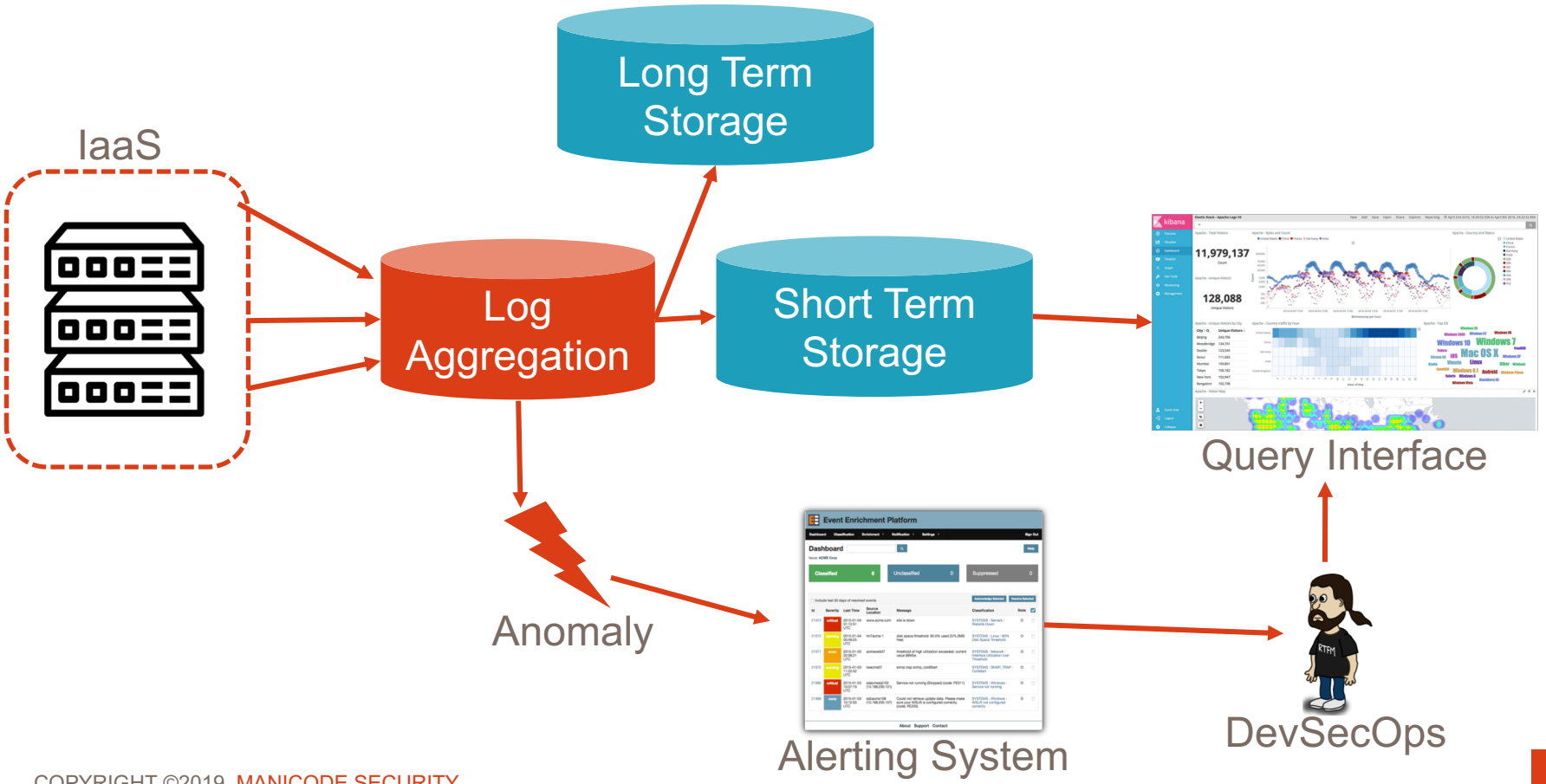
COPYRIGHT ©2019 MANICODE SECURITY

Logging, Monitoring, and Alerting

- Logs are a part of daily life in the DevOps world
- In security, we focus on particular logs to detect security anomalies and for forensic capabilities
- A basic logging pipeline can be shared between Developers, Operations, and Security teams:
  - **Log Aggregation:** Used to ingest logs from systems, applications, network components, etc.
  - **Long Term Storage:** Filesystem which retains logs for an extended period of time. Good for forensics or breach investigation.
  - **Short Term Storage**: Filesystem or DB which stores logs to be queried quickly and easily.
  - **Alerting:** Anomaly detection system which is responsible for sending alerts to teams when a deviation occurs

COPYRIGHT ©2019 MANICODE SECURITY

Logging and Monitoring Pipeline



COPYRIGHT ©2019 MANICODE SECURITY

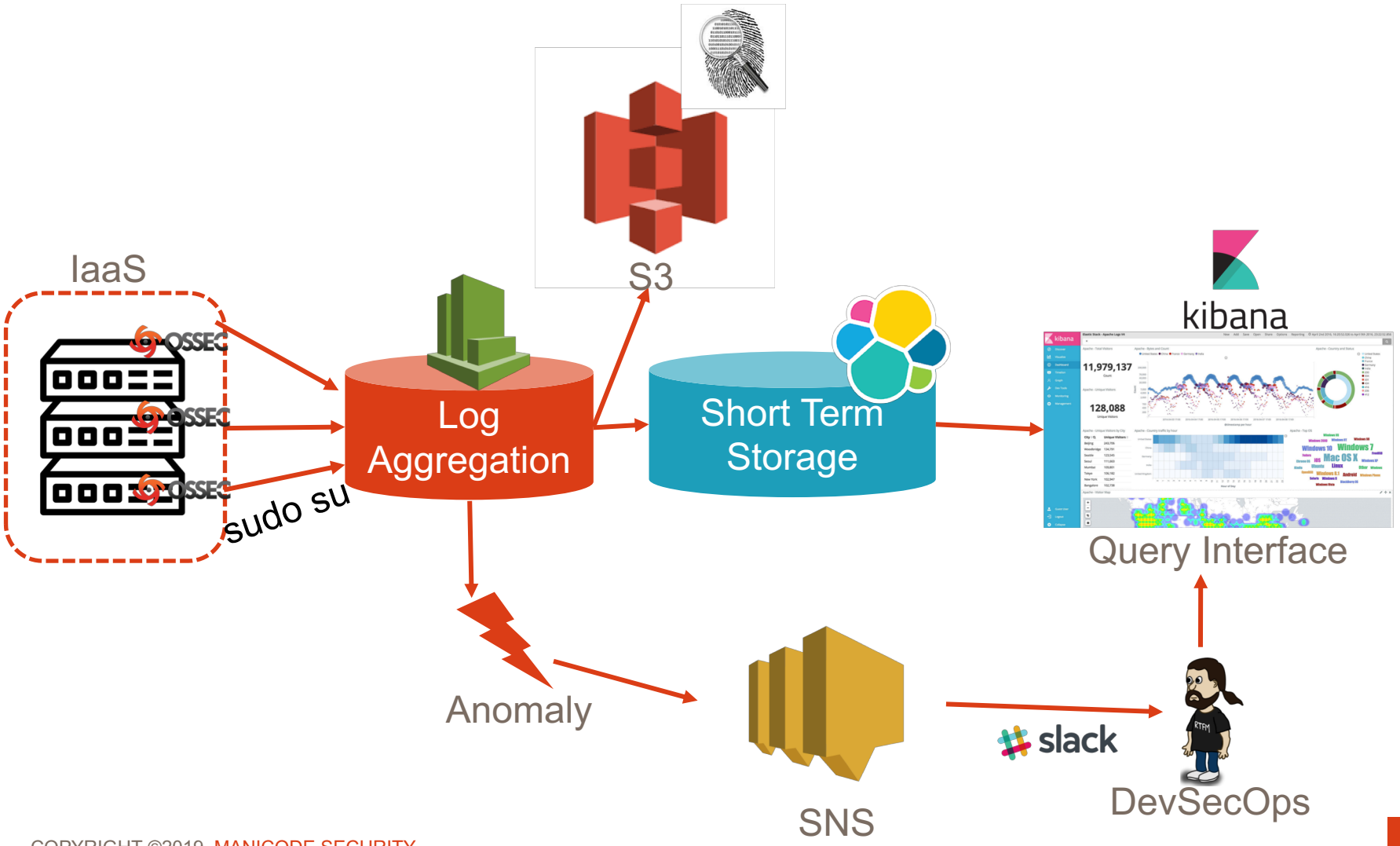
Monitoring and Alerting in Action

- OSSEC is a popular open source Host-Based Intrusion Detection System (HIDS)
- An OSSEC Agent is installed on each cloud instance throughout our IaaS
- Wired up with Elasticsearch and Kibana can make for a great (and affordable) DevSecOps dashboard
- Can alert on a wide variety of security-specific events:
  - Invalid SSH attempts
  - Successful sudo to ROOT executed
  - Interactive session opened
  - Package deleted
  - DB access from unsuspected system



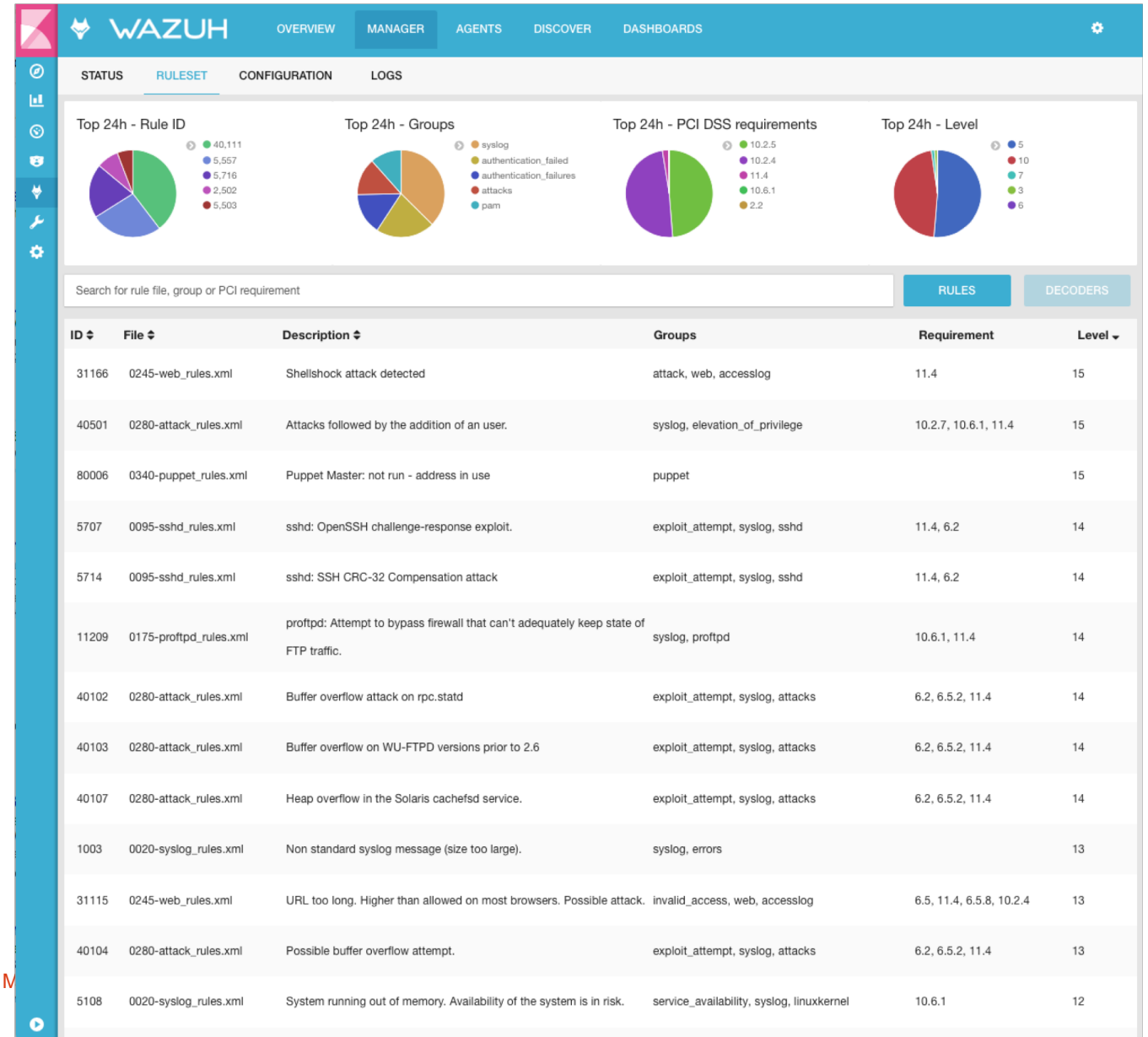
COPYRIGHT ©2019 MANICODE SECURITY

Monitoring and Alerting in Action



COPYRIGHT ©2019 MANICODE SECURITY

Open Source PCI Dashboard

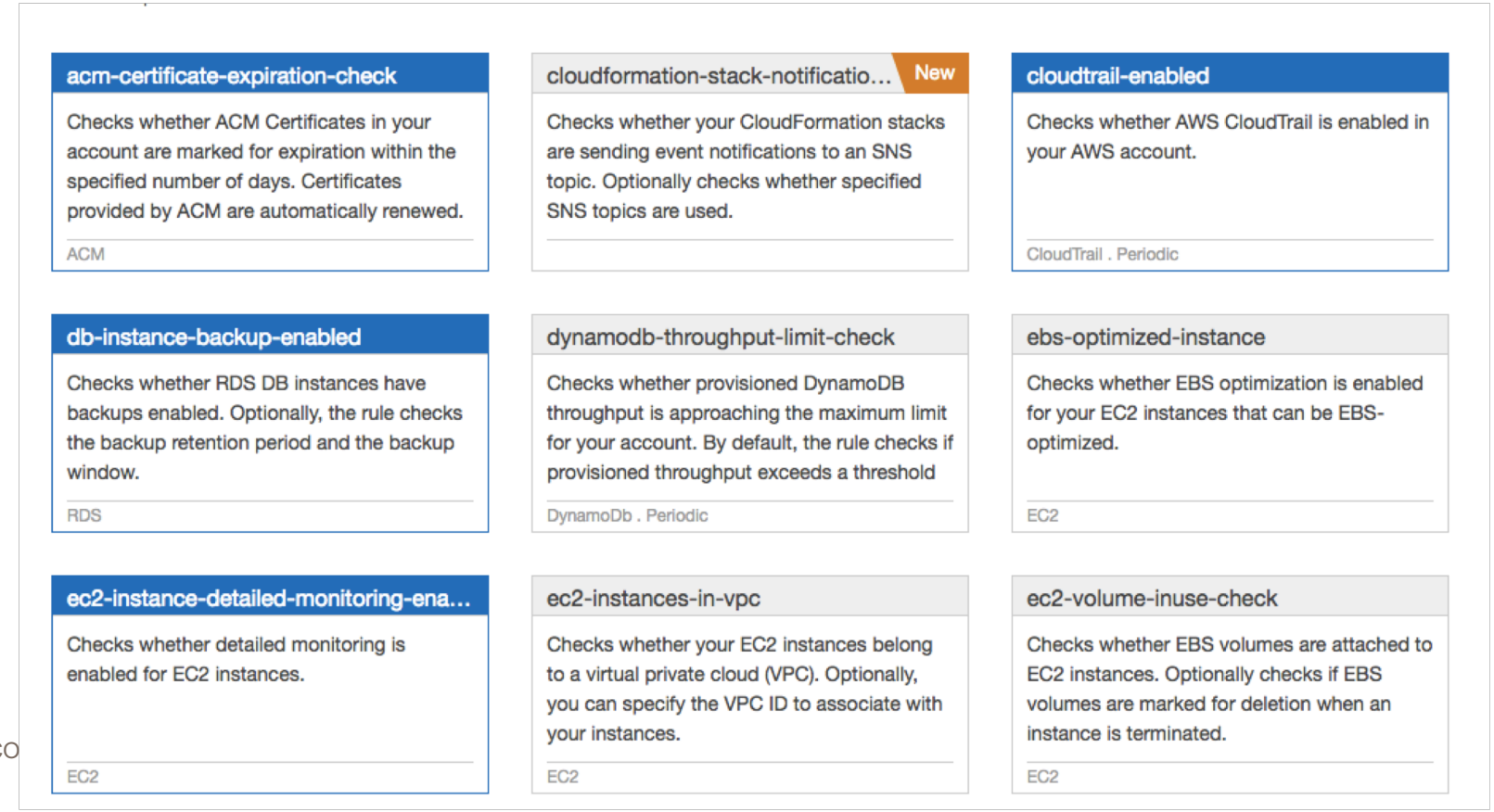


ID	File	Description	Groups	Requirement	Level
31166	0245-web_rules.xml	Shellshock attack detected	attack, web, accesslog	11.4	15
40501	0280-attack_rules.xml	Attacks followed by the addition of an user.	syslog, elevation_of_privilege	10.2.7, 10.6.1, 11.4	15
80006	0340-puppet_rules.xml	Puppet Master: not run - address in use	puppet		15
5707	0095-sshd_rules.xml	sshd: OpenSSH challenge-response exploit.	exploit_attempt, syslog, sshd	11.4, 6.2	14
5714	0095-sshd_rules.xml	sshd: SSH CRC-32 Compensation attack	exploit_attempt, syslog, sshd	11.4, 6.2	14
11209	0175-proftpd_rules.xml	proftpd: Attempt to bypass firewall that can't adequately keep state of FTP traffic.	syslog, proftpd	10.6.1, 11.4	14
40102	0280-attack_rules.xml	Buffer overflow attack on rpc.statd	exploit_attempt, syslog, attacks	6.2, 6.5.2, 11.4	14
40103	0280-attack_rules.xml	Buffer overflow on WU-FTPD versions prior to 2.6	exploit_attempt, syslog, attacks	6.2, 6.5.2, 11.4	14
40107	0280-attack_rules.xml	Heap overflow in the Solaris cachefsd service.	exploit_attempt, syslog, attacks	6.2, 6.5.2, 11.4	14
1003	0020-syslog_rules.xml	Non standard syslog message (size too large).	syslog, errors		13
31115	0245-web_rules.xml	URL too long. Higher than allowed on most browsers. Possible attack.	invalid_access, web, accesslog	6.5, 11.4, 6.5.8, 10.2.4	13
40104	0280-attack_rules.xml	Possible buffer overflow attempt.	exploit_attempt, syslog, attacks	6.2, 6.5.2, 11.4	13
5108	0020-syslog_rules.xml	System running out of memory. Availability of the system is in risk.	service_availability, syslog, linuxkernel	10.6.1	12

COPYRIGHT ©2019 M

Monitoring and Alerting – Config

- Alerts based on non-compliance in AWS resources
- Custom rules or out-of-the-box AWS rules



Monitoring and Alerting – Config

eip-attached

Checks whether all EIP addresses allocated to a VPC are attached to EC2 instances or in-use ENIs.

EC2

encrypted-volumes

Checks whether EBS volumes that are in an attached state are encrypted.

EC2

iam-password-policy

Checks whether the account password policy for IAM users meets the specified requirements.

IAM . Periodic

iam-user-group-membership-check

Checks whether IAM users are members of at least one IAM group.

IAM

iam-user-no-policies-check

Checks that none of your IAM users have policies attached. IAM users must inherit permissions from IAM groups or roles.

IAM

rds-multi-az-support

Checks whether high availability is enabled for your RDS DB instances.

RDS

rds-storage-encrypted

Checks whether storage encryption is enabled for your RDS DB instances.

RDS

restricted-ssh

Checks whether security groups that are in use disallow unrestricted incoming SSH traffic.

EC2

root-account-mfa-enabled

Checks whether the root user of your AWS account requires multi-factor authentication for console sign-in.

IAM . Periodic

COPYRIGHT ©2019 MANICODE SECURITY

Monitoring and Alerting – Config

Rules

Rules represent your desired configuration settings. AWS Config evaluates whether your resource configurations comply with relevant

Add rule

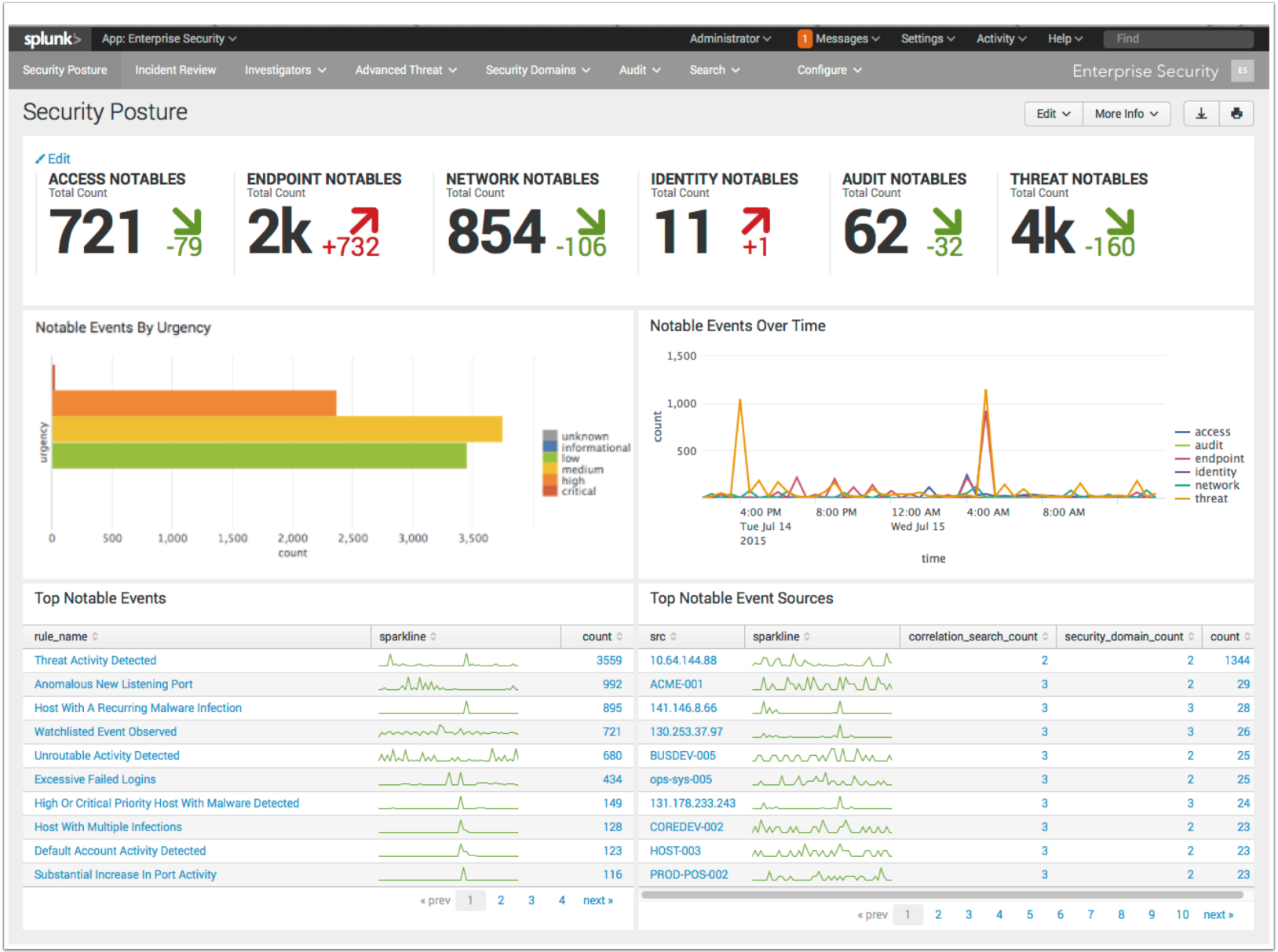
Rule name	Compliance
iam-user-no-policies-check	3 noncompliant resource(s)
s3-bucket-ssl-requests-only	2 noncompliant resource(s)
s3-bucket-logging-enabled	2 noncompliant resource(s)
iam-password-policy	1 noncompliant resource(s)
cloudtrail-enabled	Compliant
restricted-ssh	Compliant
root-account-mfa-enabled	Compliant
rds-storage-encrypted	No resources in scope
encrypted-volumes	No resources in scope
ec2-instance-detailed-monitoring-enabled	No resources in scope
acm-certificate-expiration-check	No results available

Monitoring and Alerting – Splunk

- Defacto log aggregation and analysis toolset
- Enables us to:
  - Monitor for security events across environments
  - Analyze anomalous behavior
  - Automate IR procedures
  - Visualize attack data
  - Correlate seemingly random data streams to actionable security events

COPYRIGHT ©2019 MANICODE SECURITY

Monitoring and Alerting – Splunk



Infrastructure Scanning Using Prowler

Prowler: AWS CIS Benchmark Tool

Table of Contents

- Description
- Features
- Requirements
- Usage
- Fix
- Screenshots
- Troubleshooting
- Extras

Description

Tool based on AWS-CLI commands for AWS account security assessment and hardening, following guidelines of the CIS Amazon Web Services Foundations Benchmark 1.1

Features

It covers hardening and security best practices for all AWS regions related to:

- Identity and Access Management (24 checks)
- Logging (8 checks)
- Monitoring (15 checks)
- Networking (5 checks)
- Extra checks (3 checks) *see Extras section

For a comprehesive list and resolution look at the guide on the link above.

With Prowler you can:

- get a colourish or monochrome report
- a CSV format report for diff
- run specific checks without having to run the entire report
- check multiple AWS accounts in parallel

COPYRIGHT ©2019 MANICODE SECURITY

Common Tools Found in DevSecOps Environments

















COPYRIGHT ©2019 MANICODE SECURITY

Infrastructure as Code

COPYRIGHT ©2019 MANICODE SECURITY

Building Infrastructure

- Is ***your*** infrastructure…
  - Self documenting?
  - Version controlled?
  - Capable of continuous delivery?
  - Integration tested?
  - Immutable?

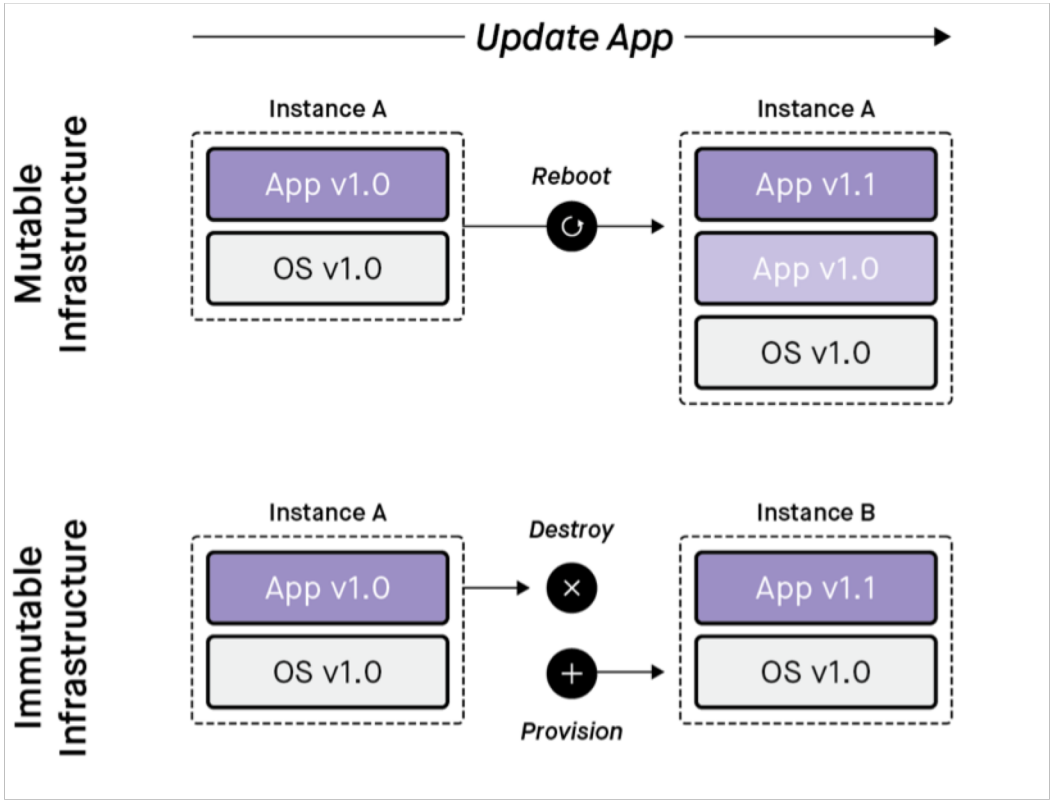


Remember: ”It’s all software"

COPYRIGHT ©2019 MANICODE SECURITY

Immutable Infrastructure

> "Immutable infrastructure is compromised of components which are replaced during deployment rather than being updated in place"



COPYRIGHT ©2019 MANICODE SECURITY

Security and Immutable Infrastructure

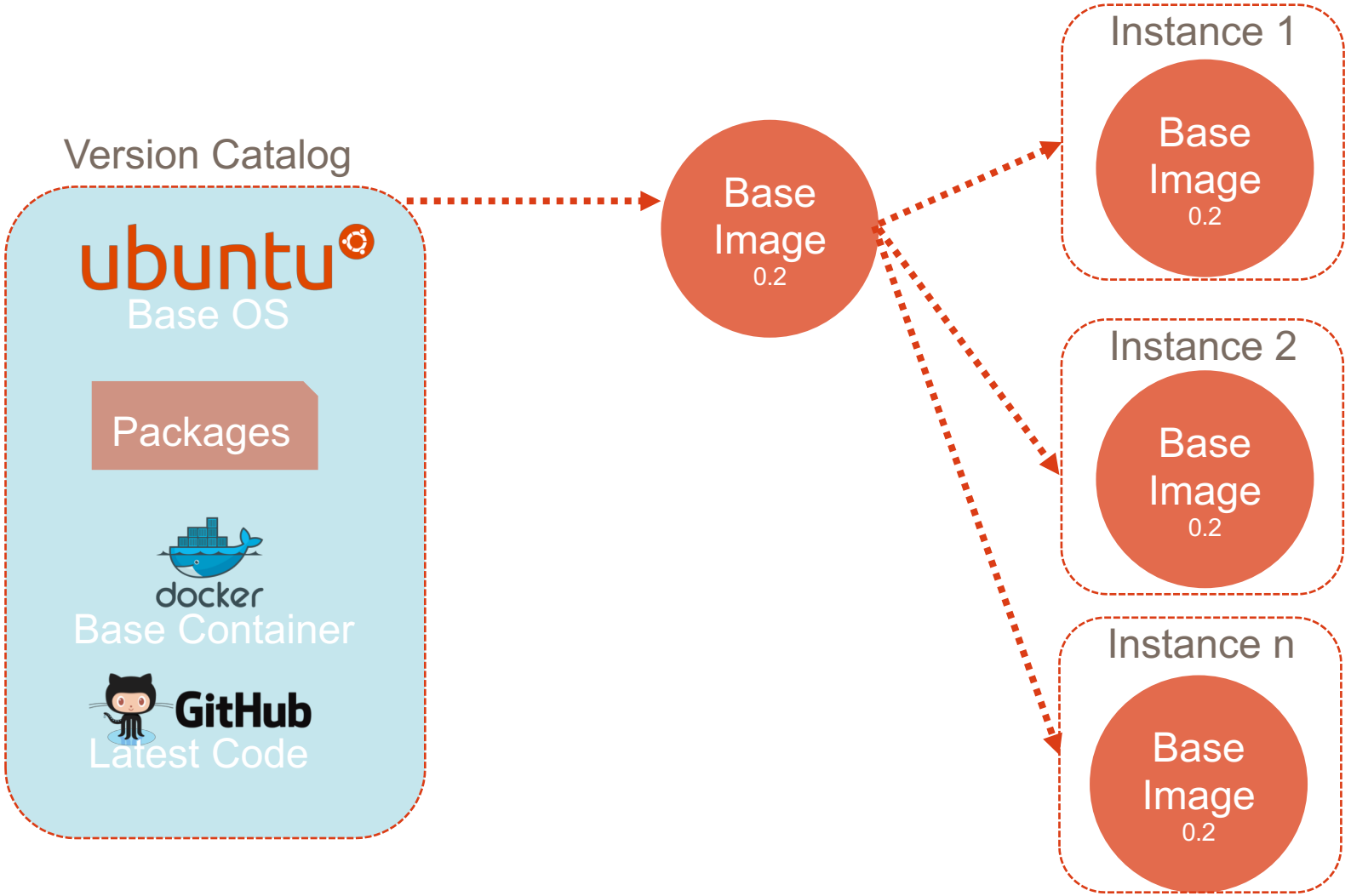
- An immutable infrastructure starts with a "Golden Image" in a version catalog
- Security teams have a central location to validate images as compliant and enforce OS hardening policies
- No more guesswork what is installed Automation can flag security anomalies vs. human intervention

"Push Security to the Left"



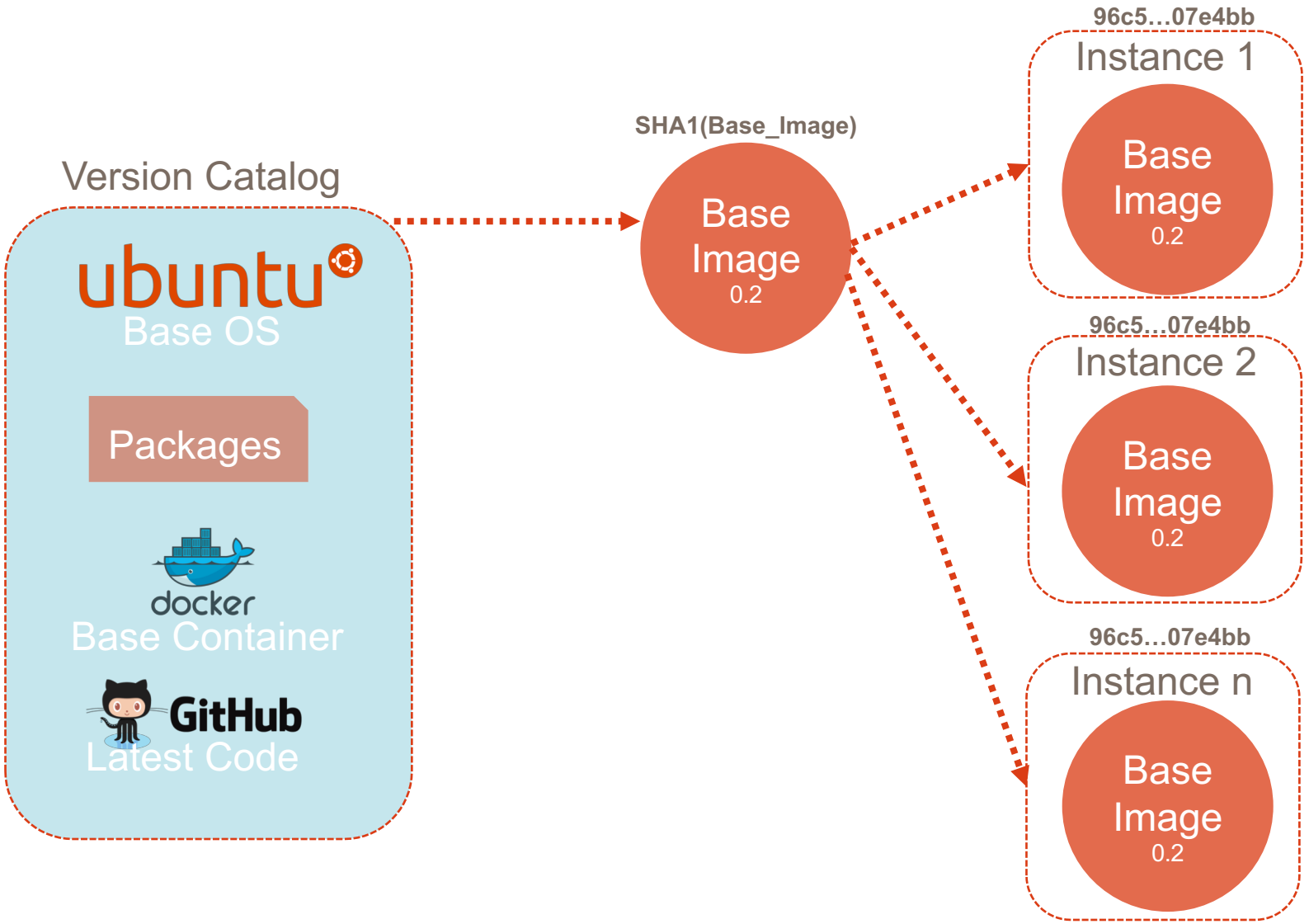
COPYRIGHT ©2019 MANICODE SECURITY

Simple Immutable Infrastructure



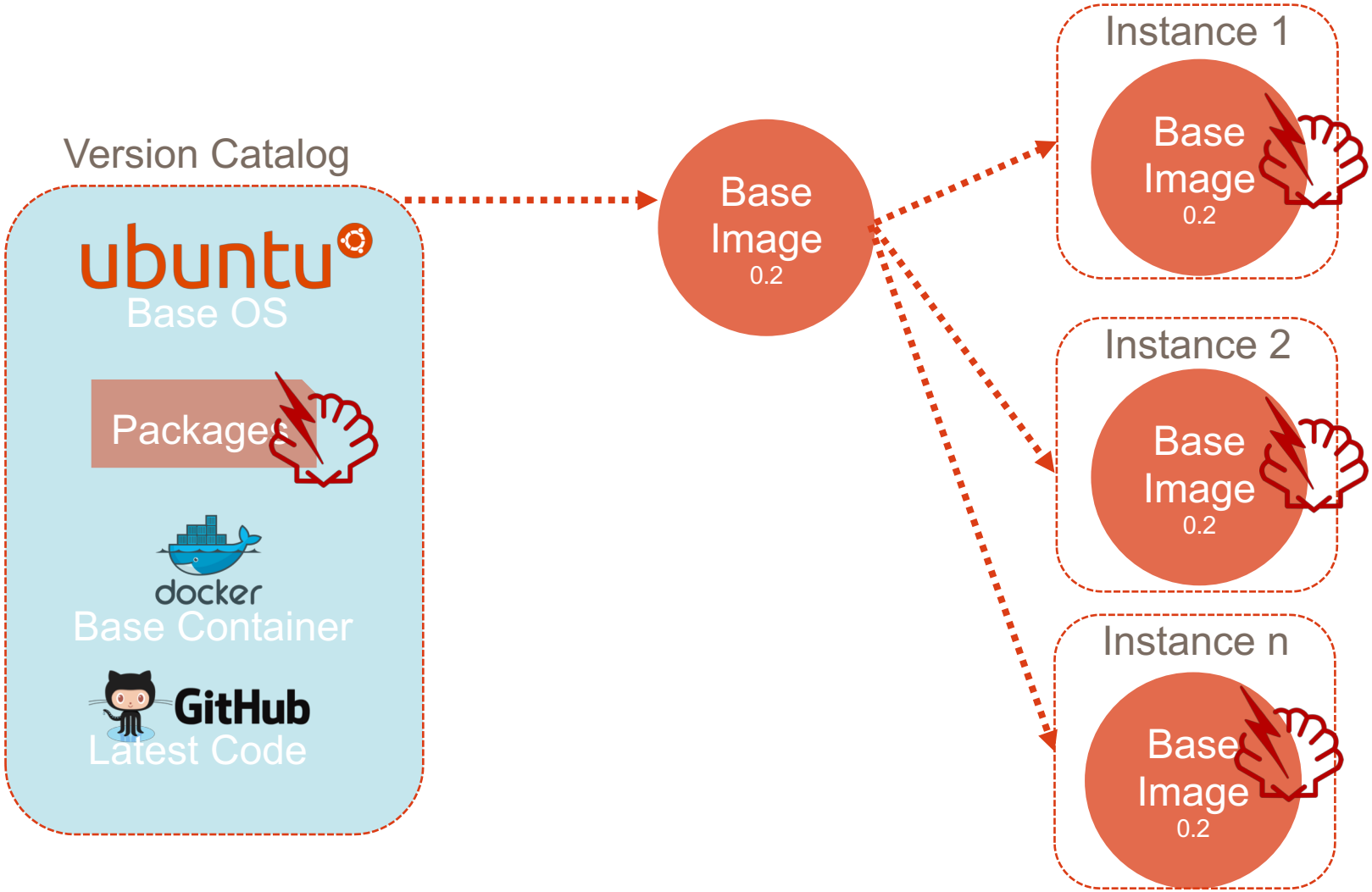
COPYRIGHT ©2019 MANICODE SECURITY

Proving Immutability



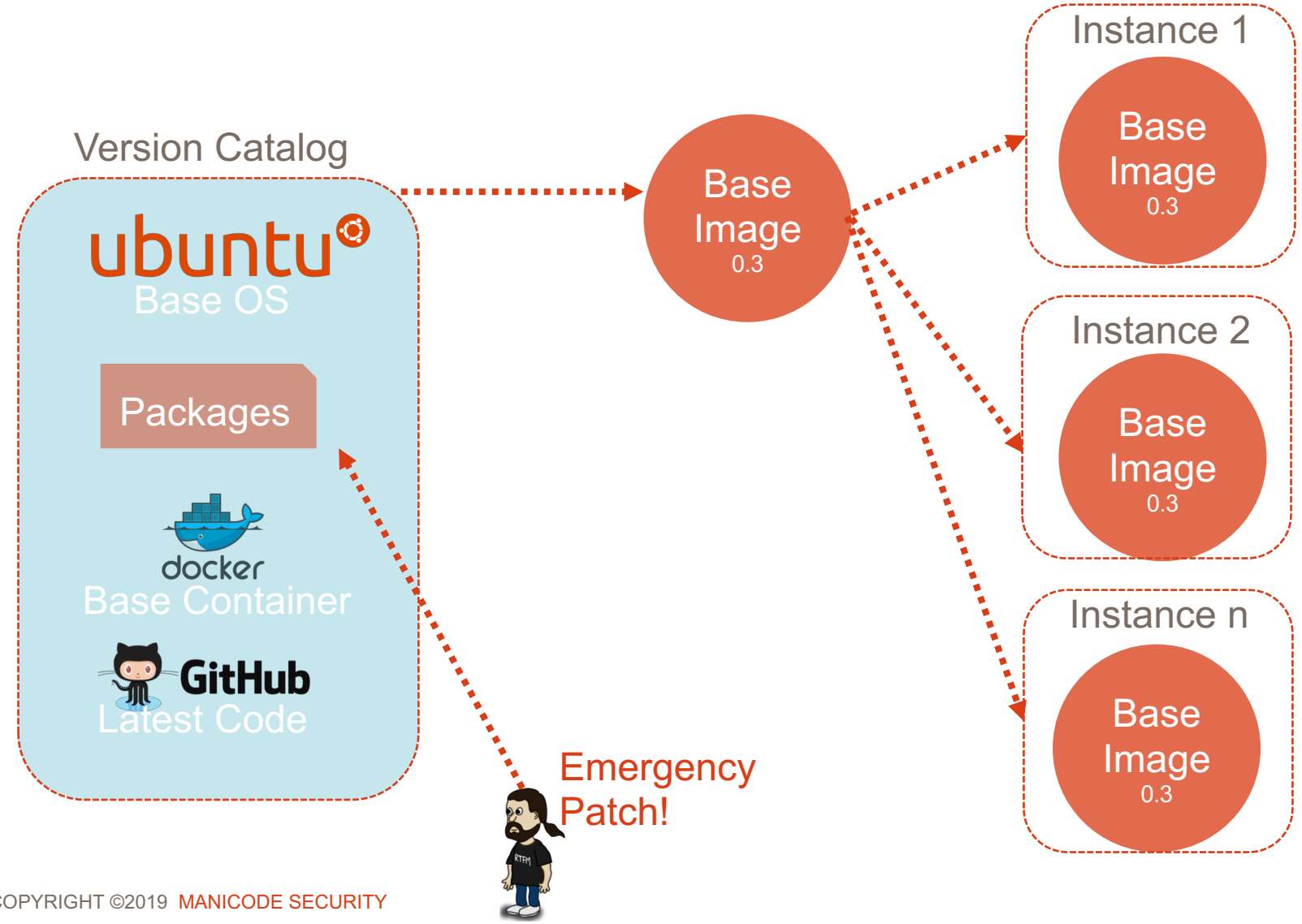
COPYRIGHT ©2019 MANICODE SECURITY

Shellshock?



COPYRIGHT ©2019 MANICODE SECURITY

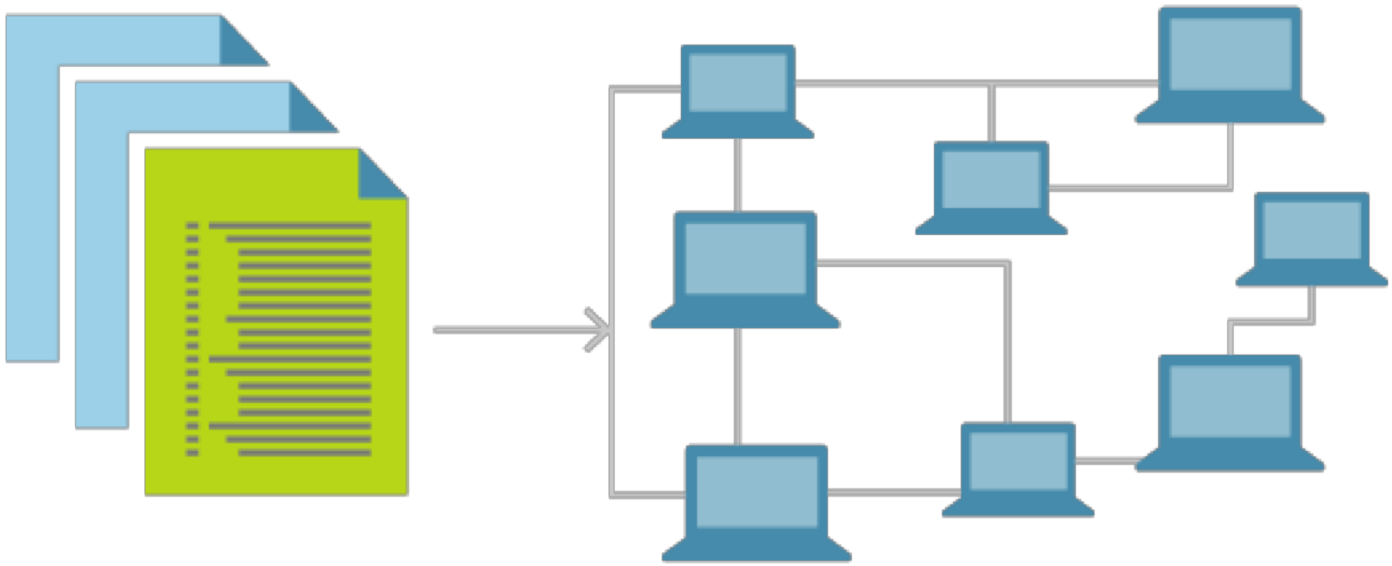
Shellshock?



COPYRIGHT ©2019 MANICODE SECURITY

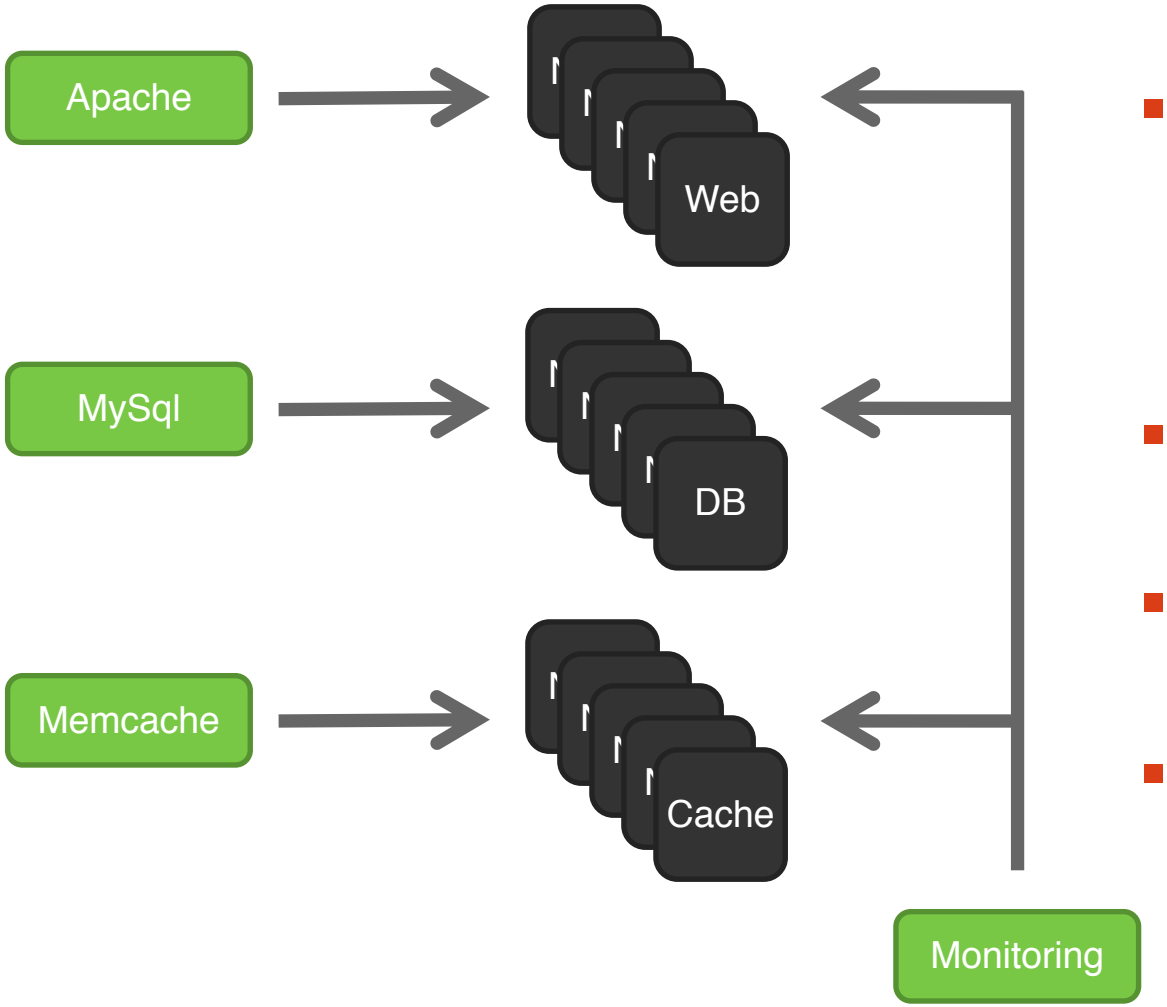
Infrastructure as Code

Infrastructure as code (IaC) allows for infrastructure to be deployed using a high-level, descriptive language. IaC treats the entire infrastructure as if it is software. Because, it's all software…



COPYRIGHT ©2019 MANICODE SECURITY

Grouping & Tagging

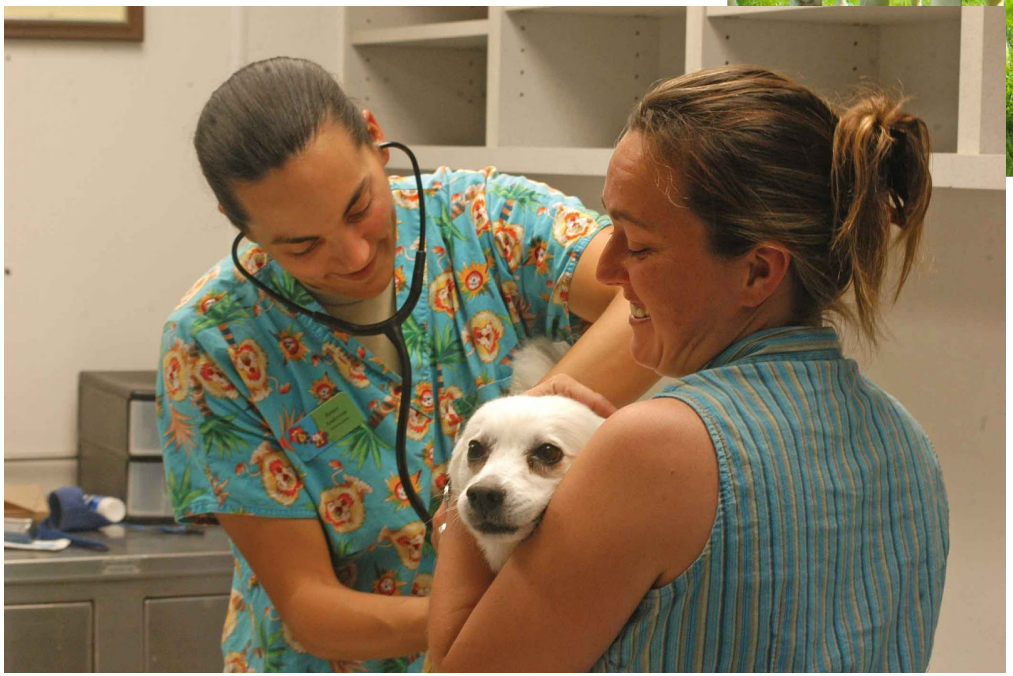


- Tagging your servers and containers applies the required set of automation
- A base set of for all servers
- Each server can have multiple tags
- Map tags to security requirements

COPYRIGHT ©2018 MANICODE SECURITY

Cattle, not pets.





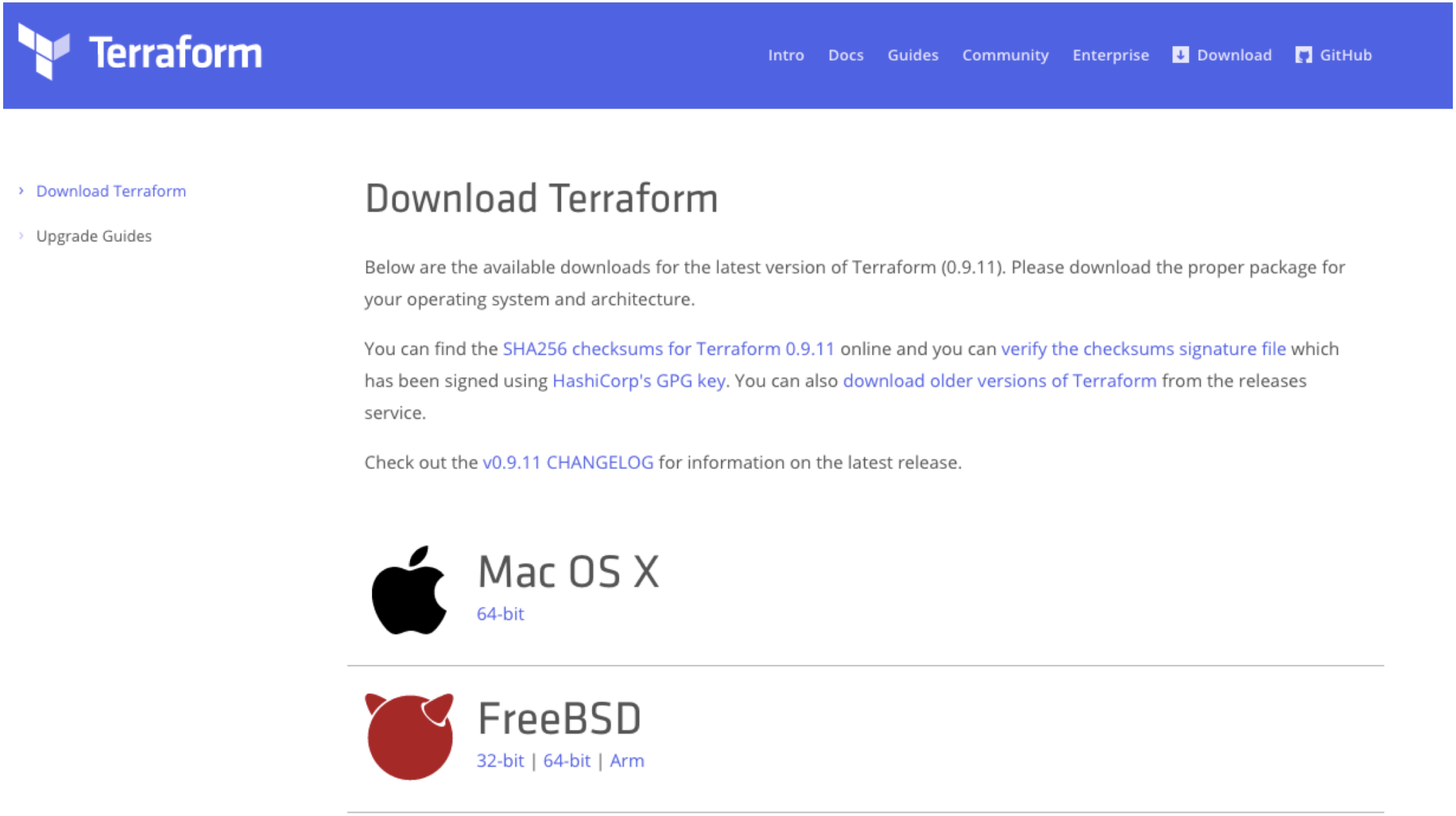
COPYRIGHT ©2019 MANICODE SECURITY

Security Wins

- Security team now has insight into the entire system
- Infrastructure is auditable and version controlled, just like source code
- Patching can be applied programmatically with a high level of certainty
- Alerting can be built for changes to specific areas of the infrastructure
  - A new firewall rule is created or deleted
  - Administrative user is created
  - New VPC rolled out
- Testing can occur much earlier in the pipeline

COPYRIGHT ©2019 MANICODE SECURITY

Infrastructure as Code - Terraform



COPYRIGHT ©2019 MANICODE SECURITY

Infrastructure as Code – K8s

```yaml
apiVersion: extensions/v1beta1
kind: Ingress
metadata:
  name: my-site-ingress
  namespace: my-site-prod
  annotations:
    kubernetes.io/tls-acme: "true"
    kubernetes.io/ingress.class: "gce"
    kubernetes.io/ingress.global-static-ip-name: my-site-external-ip
spec:
  tls:
  - hosts:
    - api.my.site
    - my.site
    secretName: my-site-cert
  rules:
  - host: api.my.site
    http:
      paths:
      - path: /*
        backend:
          serviceName: app-api
          servicePort: 80
  - host: my.site
    http:
      paths:
      - path: /*
        backend:
          serviceName: my-site-prod
          servicePort: 80
```

COPYRIGHT ©2019 MAN

Security Testing Infrastructure: Compliance as Code




```
describe package('telnetd') do
 it { should_not be_installed }
end

describe inetd_conf do
 its('telnet') { should eq nil }
end
```

COPYRIGHT ©2019 MANICODE SECURITY

Security Testing Infrastructure using GauntIt

```
# nmap-simple.attack
Feature: simple nmap attack to check for open ports

  Background:
    Given "nmap" is installed
    And the following profile:
      | name     | value       |
      | hostname | example.com |

  Scenario: Check standard web ports
    When I launch an "nmap" attack with:
      """
      nmap -F <hostname>
      """
    Then the output should match /80.tcp\s+open/
    Then the output should not match:
      """
      25\/tcp\s+open
      """
```

COPYRIGHT ©2019 MANICODE SECURITY

Security Testing Infrastructure using GauntIt

```
@slow
Feature: Run dirb scan on a URL

Scenario: Use dirb to scan a website for basic security requirements and the DIRB_WORDLISTS environment variable must be set
  Given "dirb" is installed
  And the following profile:
     | name                | value                          |
     | hostname            | http://localhost:8008          |
     | dirb_wordlists_path | Overwritten by $DIRB_WORDLISTS |
     | wordlist            | vulns/tests.txt                |
  When I launch a "dirb" attack with:
  """
  dirb <hostname> <dirb_wordlists_path>/<wordlist> -f
  """
  Then the output should contain:
  """
  FOUND: 0
  """
```

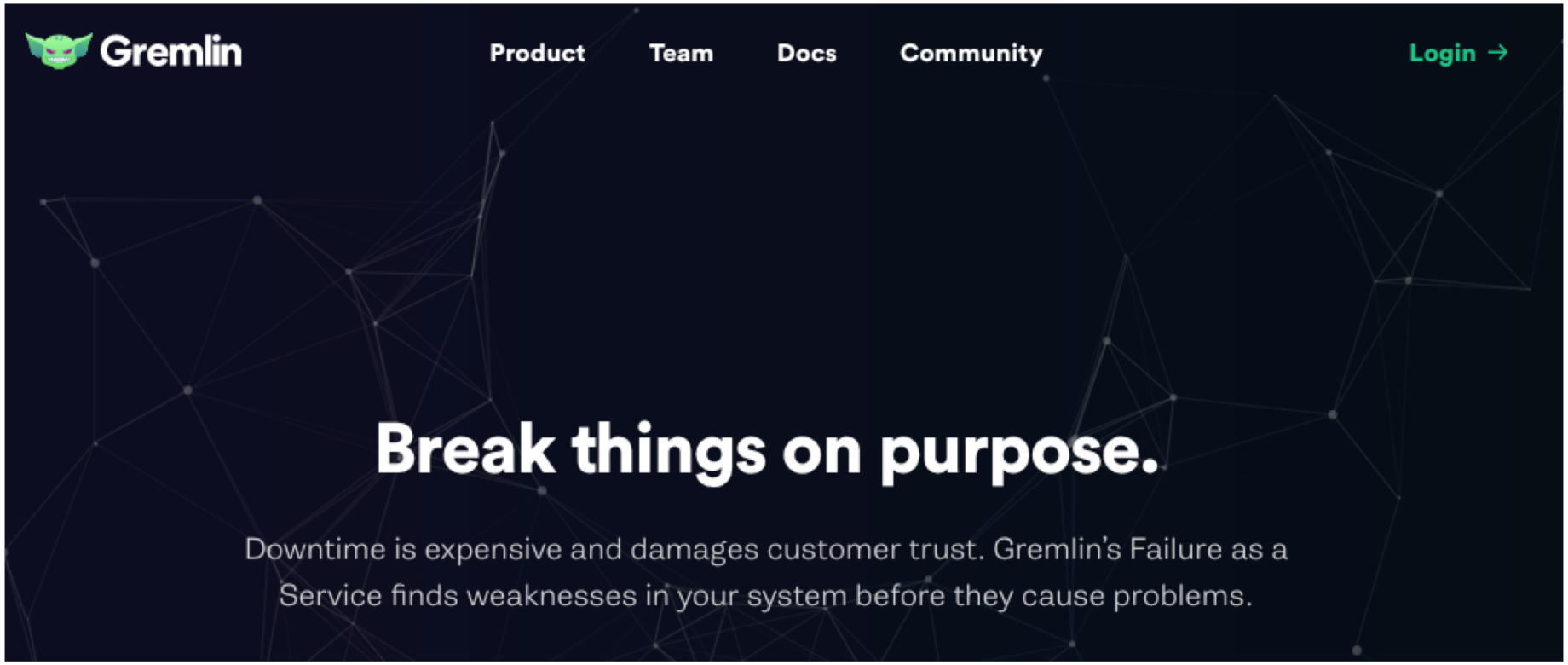
COPYRIGHT ©2019 MANICODE SECURITY

”Chaos” Testing Infrastructure using Chaos Monkey



COPYRIGHT ©2019 MANICODE SECURITY

”Chaos” Testing Infrastructure using Gremlin



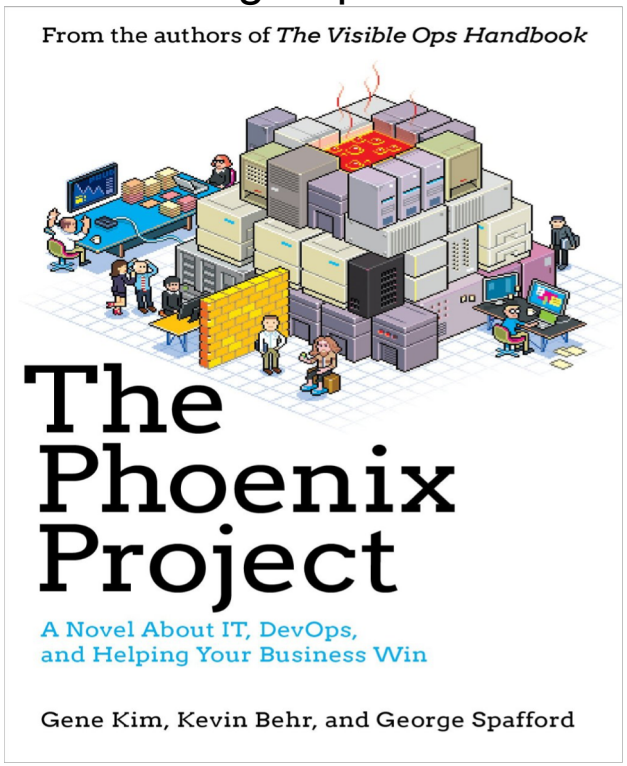
COPYRIGHT ©2019 MANICODE SECURITY

Where do we go from here?

COPYRIGHT ©2019 MANICODE SECURITY

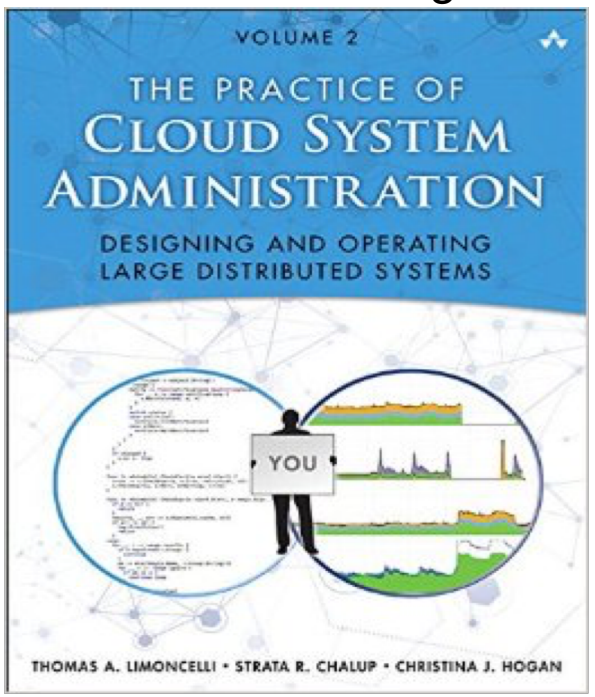
The Phoenix Project

Gene Kim, Kevin Behr and George Spafford



The Practice of Cloud System Administration

Thomas A. Limoncelli, Strata R. Chalup, Christina J. Hogan



COPYRIGHT ©2018 MANICODE SECURITY



It's been a pleasure.

jmesta@manicode.com

Jimmy Mesta Secure Coding Instructor *www.manicode.com*