Scaling Resilient Cloud Infrastructure

Surge 2016 – 09/19/16 – Tanusree McCabe

What does infrastructure mean to you?

Infrastructure can be…

- …static network (VPC, route tables, subnets, VGWs, IGWs, etc.)
- …dynamic network (security group)
- …network appliances (proxy, WAF, etc.)
- …monitoring (CloudWatch, etc.)
- …logging (CloudTrail, etc.)
- …supporting stack (EC2, ECS, S3, ELB, ACM, KMS, Route53, CloudFront, etc.)

Cloud infrastructure goals

- Scalable
  - Deploy across x # networks, accounts
  - Enable developers to be productive Day 1

- Resilient
  - Design for failure
    - Availability Zone goes down
    - Region goes down
    - API rate limits are hit
  - Graceful degradation
  - Self healing infrastructure

- Secure
  - Infrastructure itself incorporates safeguards
  - Pipeline to develop and deploy infrastructure as code is secure

Challenges

What challenges do you face in meeting these goals?

- Scalable
- Resilient
- Secure

Sample Challenges

- Scalable
  - Lack of orchestration
  - Reconciling human changes with automated changes
  - Lack of understanding
- Resilient
  - Persistent data
  - No backup/recovery plan
  - Lack of change / configuration / release management with adverse impact
- Secure
  - ‘wild wild west’
  - Attackers are in the cloud, too
  - Insider threat
  - Lack of automated controls framework

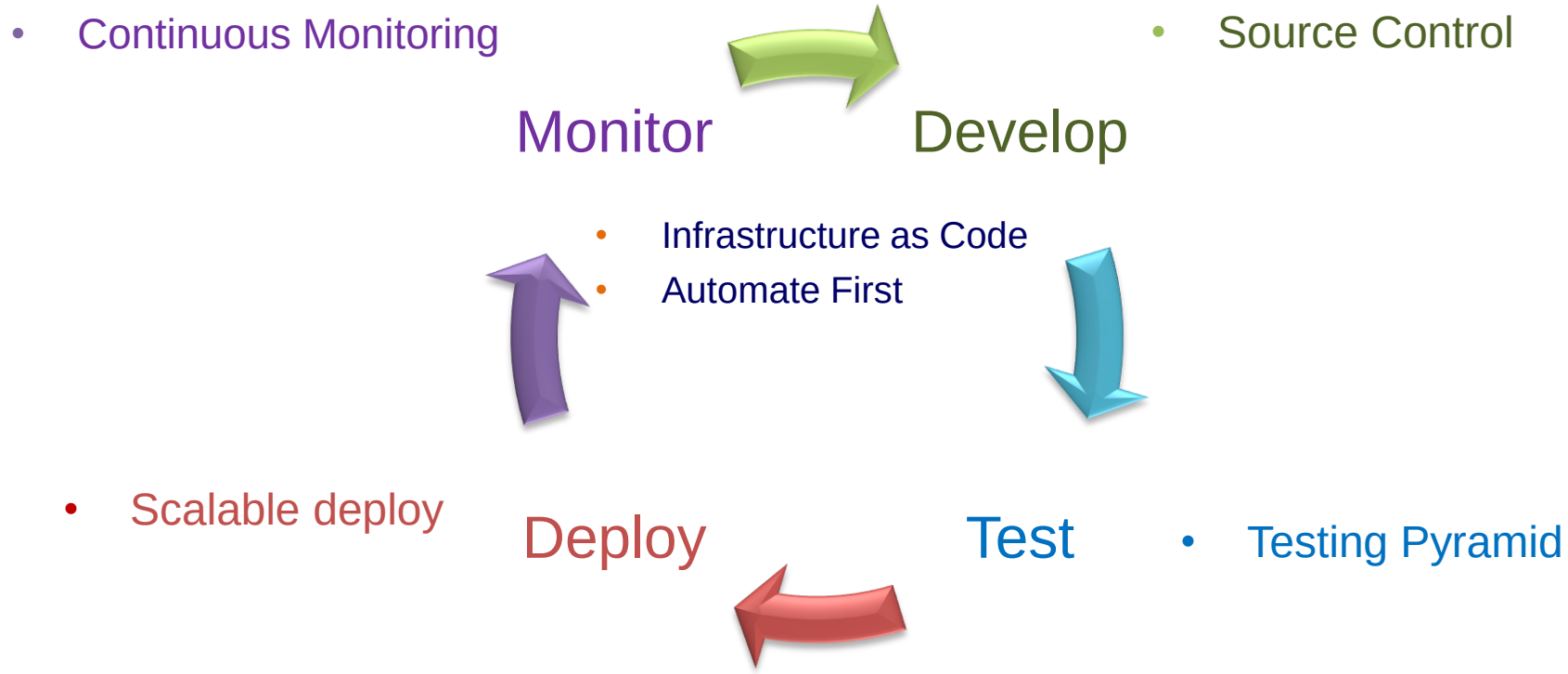
Possible Solutions

What is your automation approach?

- Automate or not automate?
  - Automate if:
    - It needs to be repeated
    - Saves time
    - Improves accuracy
    - Reduces risk
- Automate now or later?
  - Do you need to import manual changes? Is the target stable?
  - Do you have requisite skill sets?
- Centralized or decentralized?
  - Who controls the automation? How is access federated?
- Automate using what tool?
  - Quick analysis of alternatives – functional requirements, usability, cost, maintainability
- How will you build in quality control?
- How will you build in security?
  - Security of the infrastructure + security of how the infrastructure is delivered
  - Incidence response

Automation strategy / workflow using DevOps mindset

- Orchestrate infrastructure as code changes using CI/CD best practices
- Deliver infrastructure as code using application development best practices

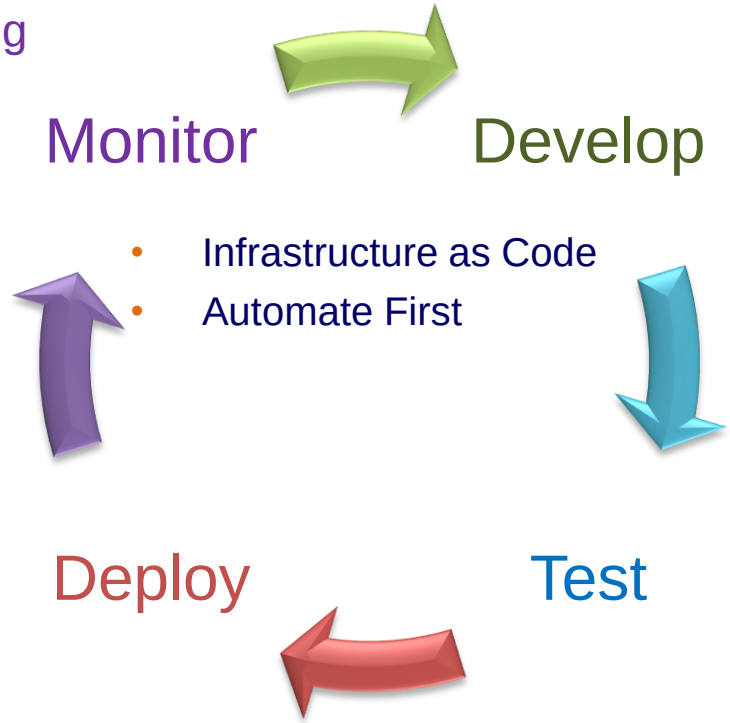


Automation strategy / workflow using DevOps mindset

- Configuration Management Overlay

- Continuous Monitoring
  - **Config changes**

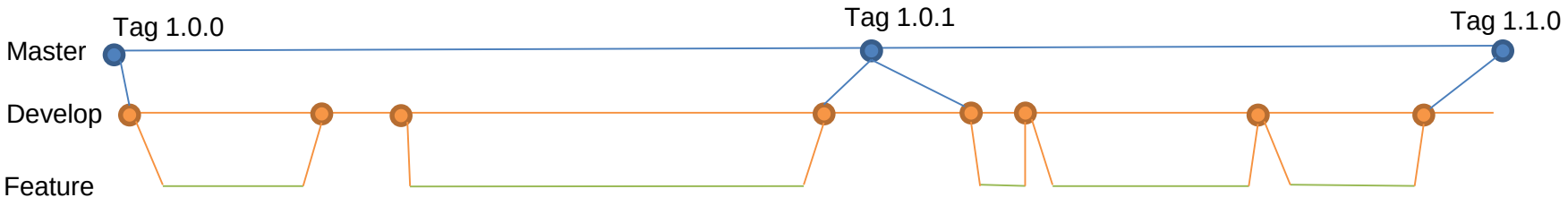
- Source Control
  - **Version Control / Tags**
  - **Metadata driven changes**



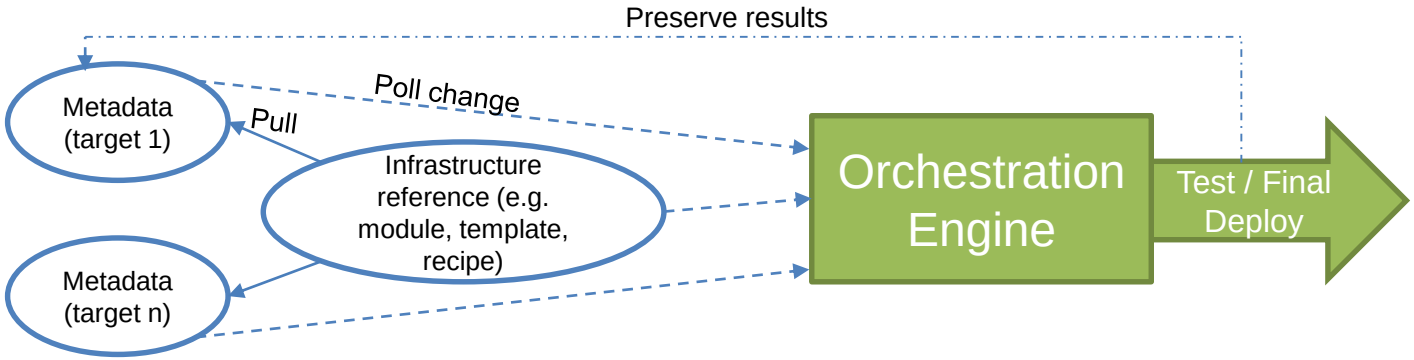
- Scalable deploy

- Testing Pyramid

Configuration Management – Details/Example



Example GitFlow model



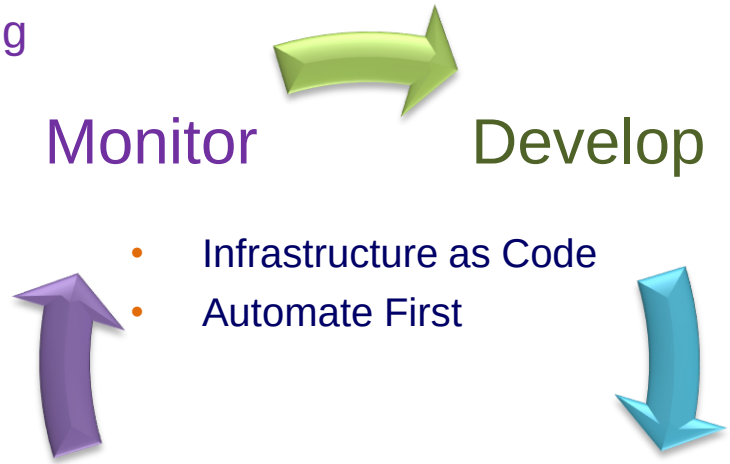
Example Repository Structure within example CI/CD process

Automation strategy / workflow using DevOps mindset

- Change Management Overlay

- Continuous Monitoring
  - Config changes

- Source Control
  - Version Control / Tags
  - Metadata driven changes



- Scalable deploy
  - **Automated / Standard change requests**

- Testing Pyramid
  - **Unit / integration**

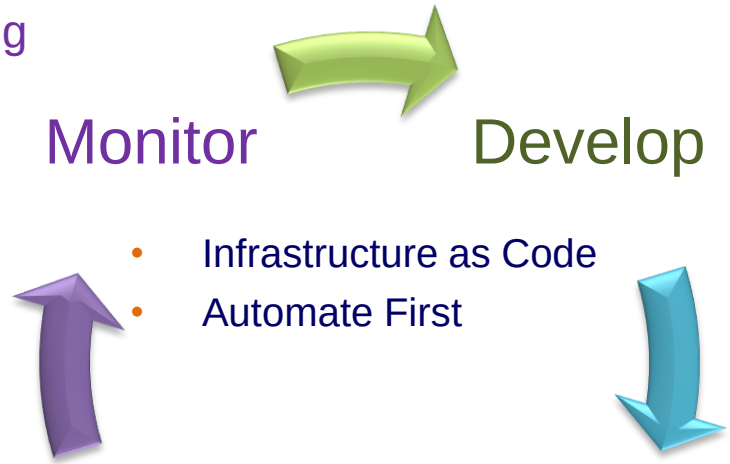
Change Management Details / Example

- Automated /standard change requests
  - CR Tool APIs e.g. Remedy
- Testing infrastructure
  - Tool specific ‘plans’ or ‘assertions’
    - CloudFormation change sets
    - Terraform plan
    - Ansible assertions
  - Actually deploy in a sandbox or lower environment
  - End to end regression testing

Automation strategy / workflow using DevOps mindset

- Release Management Overlay

- Continuous Monitoring
  - Config changes
  - **Metrics**
  - **Agile Stories**



- Source Control
  - Version Control / Tags
  - Metadata driven changes
  - **GitFlow**
  - **Peer Review**
  - **Traceability to Agile**

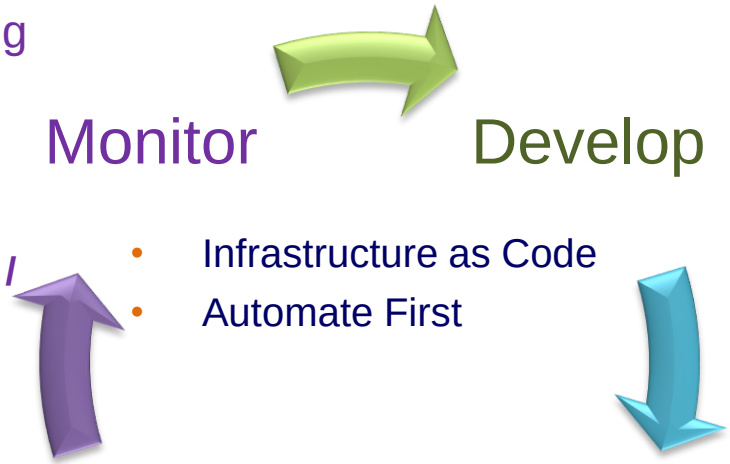
- Scalable deploy
  - Automated / Standard change requests
  - **Orchestration**

- Testing Pyramid
  - Unit / integration

Automation strategy / workflow using DevOps mindset

- Security Overlay

- Continuous Monitoring
  - Config changes
  - Metrics
  - Agile Stories
  - **Adaptive Security / Controls Framework**



- Infrastructure as Code
- Automate First

- Source Control
  - Version Control / Tags
  - Metadata driven changes
  - GitFlow
  - Peer Review
  - Traceability to Agile
  - **Secrets Management**

- Scalable deploy
  - Automated / Standard change requests
  - Orchestration
  - **Encryption**
  - **Log for Audit**

- Testing Pyramid
  - Unit / integration
  - **Security**

Security Details / Example

- Access
  - CI/CD authentication, authorization
- Network boundaries
  - Internet access (e.g. IGW), firewall policies (e.g. security groups), routing policies (e.g. route tables)
- Security boundaries
  - Scope of data, purpose of accounts, etc
- RTO / RPO
  - Infrastructure failover (multi-AZ, multi-region?)
- Encryption
  - Data in transit, Data at rest
- Logging

- ‘adaptive’ security instead of ‘reactive’ security: Monitor events, define triggers & rules, automate reactions/mitigations

- Balance freedom/flexibility for the productivity of your users with restraint

Security Details / Example – Automated Control Framework – EC2 Example

Sample Criteria	Y/N/M	Guidance	Compensating Control
Operates within VPC?	Y	Don't use default VPC, only use 'private' VPCs	Blacklist default VPCs in IAM. Monitor launches
Encrypts data at rest?	M	Use persistent EBS with CMK KMS, not instance stores. If using 3rd party AMI, generate encrypted EBS volume.	Monitor EBS volumes
Encrypts data in transit?	M	Use SSL/443 in security groups, web services, ELB listeners	Monitor for non-443; exception list
Accessible by security tool?	Y	Agents pre-baked into gold AMIs. Network open for security tools.	
Supports HA?	M	Use auto-scale groups for multi-AZ apps at minimum	Monitor for stand alone instances.
Supports multi-region DR?	M	Use multi-region failover architecture for platinum apps	
Supports backup & restore?	M	Abide by tagging for auto. EBS snapshot, AMI generation, and retention cleanup	Backup snaps/AMIs, clean-up per period
Encrypted backups?	M	Snaps are encrypted if EBS volumes are encrypted	Monitor snaps
Fine grained access controls?	Y	Supports IAM + Instance profiles	Monitor for EC2 operating without instance profiles

Security Strategy can encompass:

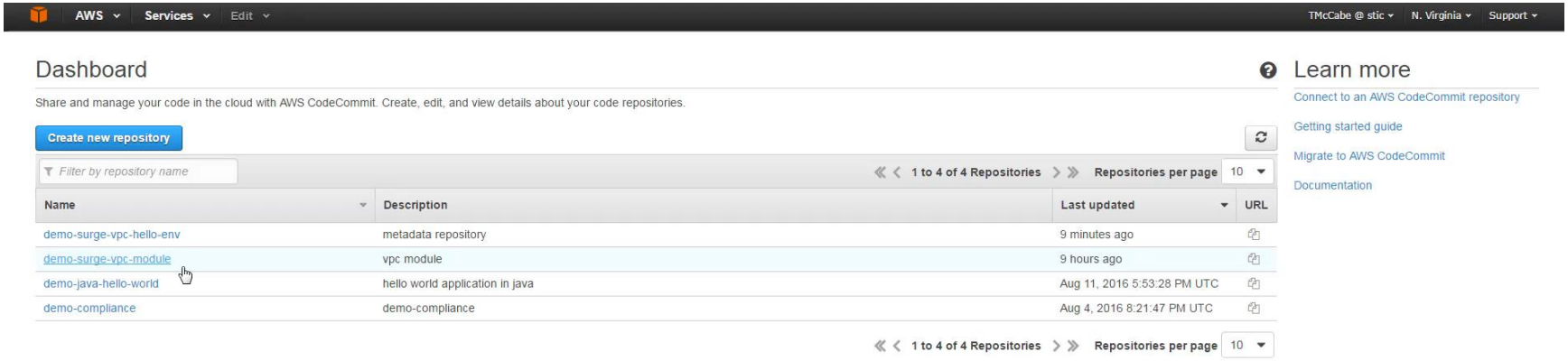
- Logging -> mine data from S3
  - CloudWatch Logs
    - EC2 logs
  - VPC Flow Logs
  - S3 Logs
  - Cloud Trail Logs
  - 3rd party solutions – e.g. ELK, Splunk
- AMIs / Static Scanning
  - Hardened images / gold images
  - Code scanning during CI/CD
- Run time compliance
  - Scan at run time
- Network control points
  - Proxy
  - WAF
  - IDP / IPS etc.

Security Strategy can encompass:

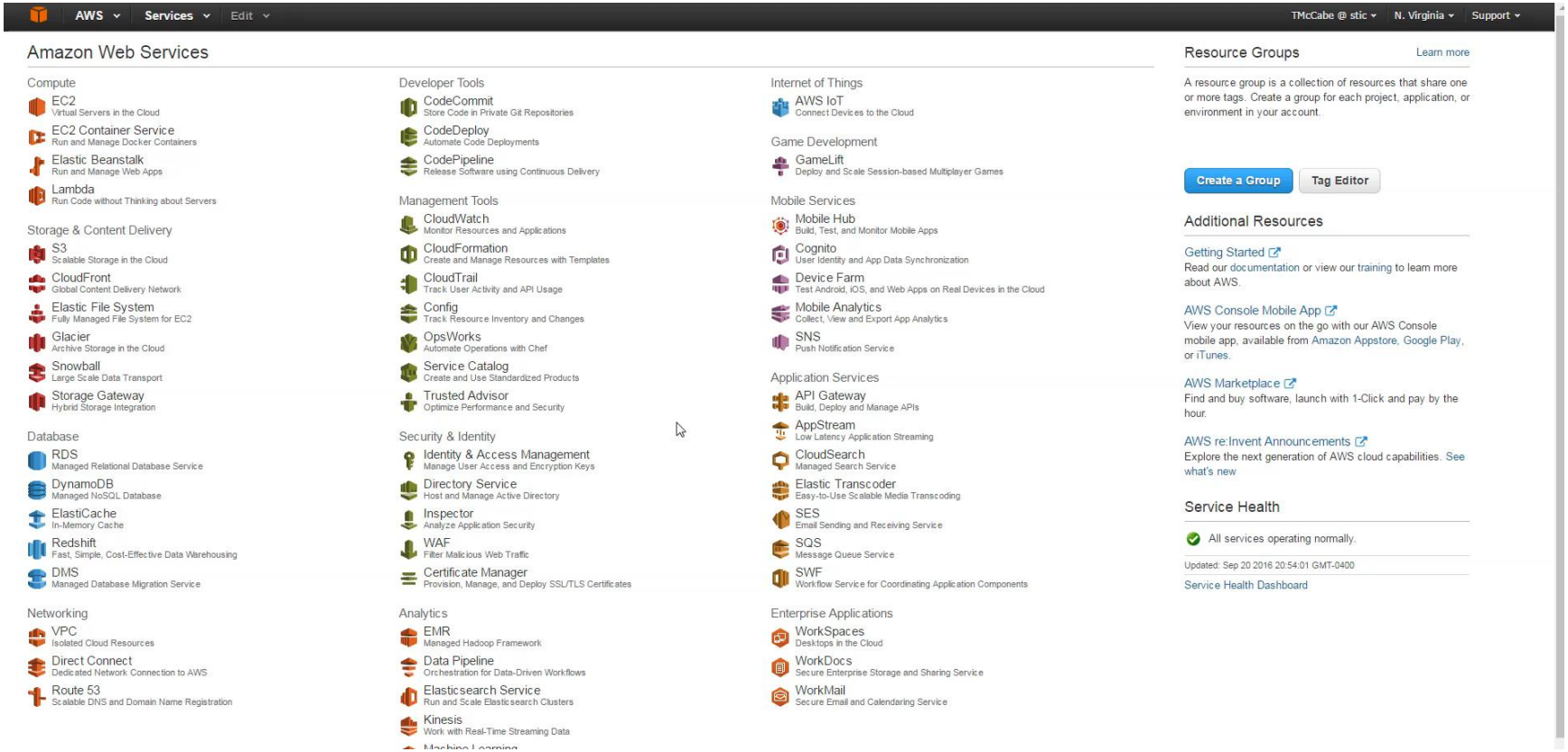
- Resiliency
  - Application resiliency (e.g. DR, HA, backup & restore, infrastructure automation)
- Forensics
  - Detect, isolate & analyze incidents
- Continuous monitoring
  - Monitoring tools
    - AWS CloudWatch + 3rd party tools e.g. NewRelic
  - Automated Compliance Framework
    - Instructive vs. punitive automated controls
    - AWS Lambda + 3rd party tools e.g. Cloud Passage
  - Automated Configuration Management
    - AWS Config / Config Rules, AWS Trusted Advisor + 3rd party tools e.g. Ansible, Salt, Puppet, Chef

Demo

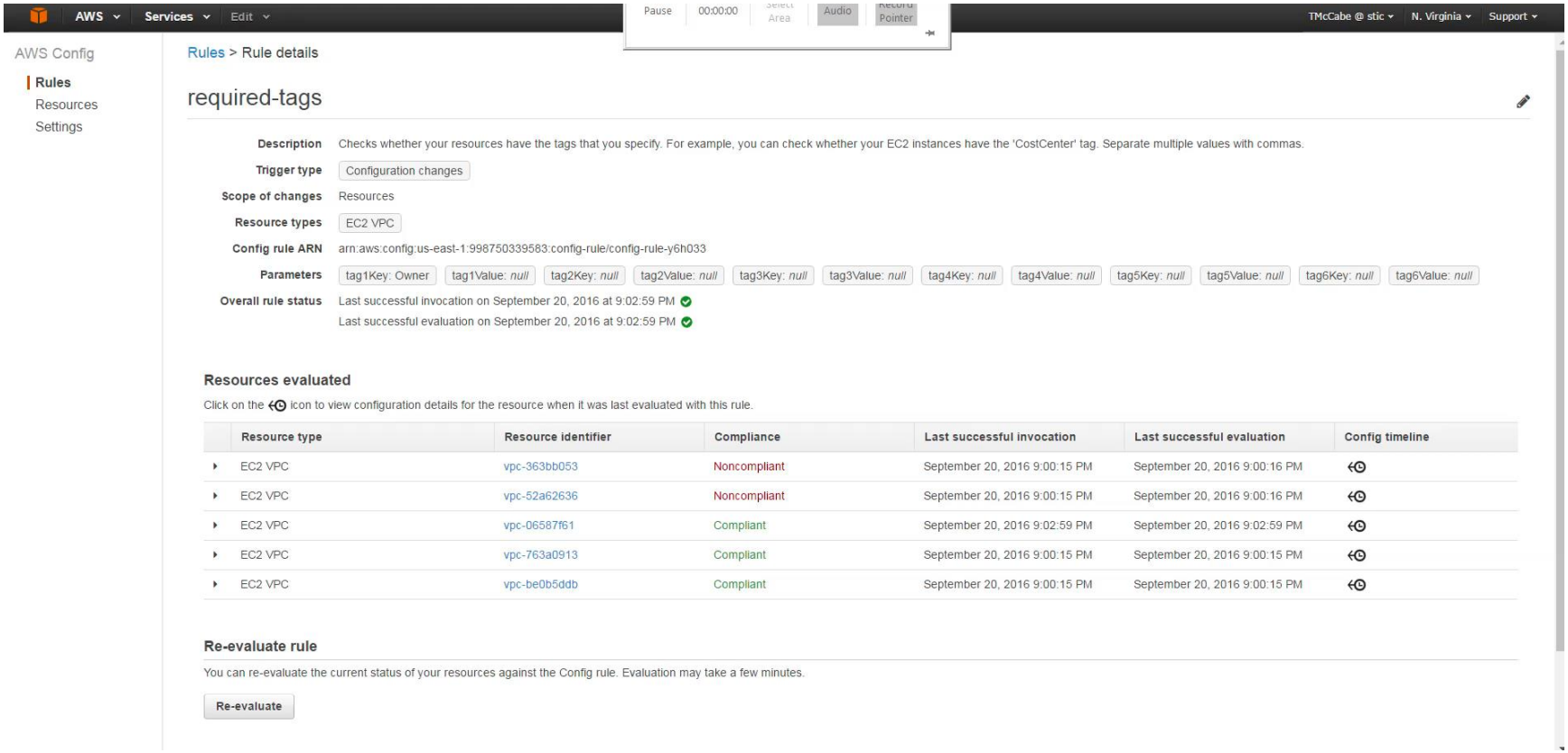
Demo (part 1)


AWS
Services
Edit
TMcCabe @ stic
N. Virginia
Support
Dashboard
Share and manage your code in the cloud with AWS CodeCommit. Create, edit, and view details about your code repositories.
Create new repository
Filter by repository name
1 to 4 of 4 Repositories
Repositories per page 10
Name
Description
Last updated
URL
demo-surge-vpc-hello-env
metadata repository
9 minutes ago
demo-surge-vpc-module
vpc module
9 hours ago
demo-java-hello-world
hello world application in java
Aug 11, 2016 5:53:28 PM UTC
demo-compliance
demo-compliance
Aug 4, 2016 8:21:47 PM UTC
1 to 4 of 4 Repositories
Repositories per page 10
Learn more
Connect to an AWS CodeCommit repository
Getting started guide
Migrate to AWS CodeCommit
Documentation

Demo (part 2)


AWS
Services
Edit
TMcCabe @ stic
N. Virginia
Support
Amazon Web Services
Compute
EC2
Virtual Servers in the Cloud
EC2 Container Service
Run and Manage Docker Containers
Elastic Beanstalk
Run and Manage Web Apps
Lambda
Run Code without Thinking about Servers
Storage & Content Delivery
S3
Scalable Storage in the Cloud
CloudFront
Global Content Delivery Network
Elastic File System
Fully Managed File System for EC2
Glacier
Archive Storage in the Cloud
Snowball
Large Scale Data Transport
Storage Gateway
Hybrid Storage Integration
Database
RDS
Managed Relational Database Service
DynamoDB
Managed NoSQL Database
ElastiCache
In-Memory Cache
Redshift
Fast, Simple, Cost-Effective Data Warehousing
DMS
Managed Database Migration Service
Networking
VPC
Isolated Cloud Resources
Direct Connect
Dedicated Network Connection to AWS
Route 53
Scalable DNS and Domain Name Registration
Developer Tools
CodeCommit
Store Code in Private Git Repositories
CodeDeploy
Automate Code Deployments
CodePipeline
Release Software using Continuous Delivery
Management Tools
CloudWatch
Monitor Resources and Applications
CloudFormation
Create and Manage Resources with Templates
CloudTrail
Track User Activity and API Usage
Config
Track Resource Inventory and Changes
OpsWorks
Automate Operations with Chef
Service Catalog
Create and Use Standardized Products
Trusted Advisor
Optimize Performance and Security
Security & Identity
Identity & Access Management
Manage User Access and Encryption Keys
Directory Service
Host and Manage Active Directory
Inspector
Analyze Application Security
WAF
Filter Malicious Web Traffic
Certificate Manager
Provision, Manage, and Deploy SSL/TLS Certificates
Analytics
EMR
Managed Hadoop Framework
Data Pipeline
Orchestration for Data-Driven Workflows
Elasticsearch Service
Run and Scale Elasticsearch Clusters
Kinesis
Work with Real-Time Streaming Data
Internet of Things
AWS IoT
Connect Devices to the Cloud
Game Development
GameLift
Deploy and Scale Session-based Multiplayer Games
Mobile Services
Mobile Hub
Build, Test, and Monitor Mobile Apps
Cognito
User Identity and App Data Synchronization
Device Farm
Test Android, iOS, and Web Apps on Real Devices in the Cloud
Mobile Analytics
Collect, View and Export App Analytics
SNS
Push Notification Service
Application Services
API Gateway
Build, Deploy and Manage APIs
AppStream
Low Latency Application Streaming
CloudSearch
Managed Search Service
Elastic Transcoder
Easy-to-Use Scalable Media Transcoding
SES
Email Sending and Receiving Service
SQS
Message Queue Service
SWF
Workflow Service for Coordinating Application Components
Enterprise Applications
WorkSpaces
Desktops in the Cloud
WorkDocs
Secure Enterprise Storage and Sharing Service
WorkMail
Secure Email and Calendaring Service
Resource Groups
Learn more
A resource group is a collection of resources that share one or more tags. Create a group for each project, application, or environment in your account.
Create a Group
Tag Editor
Additional Resources
Getting Started
Read our documentation or view our training to learn more about AWS.
AWS Console Mobile App
View your resources on the go with our AWS Console mobile app, available from Amazon Appstore, Google Play, or iTunes.
AWS Marketplace
Find and buy software, launch with 1-Click and pay by the hour.
AWS re:Invent Announcements
Explore the next generation of AWS cloud capabilities. See what's new
Service Health
All services operating normally.
Updated: Sep 20 2016 20:54:01 GMT-0400
Service Health Dashboard

Demo (part 2b)


AWS
Services
Edit
Pause
00:00:00
Audio
TMcCabe @ stic
N. Virginia
Support
AWS Config
Rules
Resources
Settings
Rules > Rule details
required-tags
Description
Checks whether your resources have the tags that you specify. For example, you can check whether your EC2 instances have the 'CostCenter' tag. Separate multiple values with commas.
Trigger type
Configuration changes
Scope of changes
Resources
Resource types
EC2 VPC
Config rule ARN
arn:aws:config:us-east-1:998750339583:config-rule/config-rule-y6h033
Parameters
tag1Key: Owner
tag1Value: null
tag2Key: null
tag2Value: null
tag3Key: null
tag3Value: null
tag4Key: null
tag4Value: null
tag5Key: null
tag5Value: null
tag6Key: null
tag6Value: null
Overall rule status
Last successful invocation on September 20, 2016 at 9:02:59 PM
Last successful evaluation on September 20, 2016 at 9:02:59 PM
Resources evaluated
Click on the icon to view configuration details for the resource when it was last evaluated with this rule.
Resource type
Resource identifier
Compliance
Last successful invocation
Last successful evaluation
Config timeline
EC2 VPC
vpc-363bb053
Noncompliant
September 20, 2016 9:00:15 PM
September 20, 2016 9:00:16 PM
EC2 VPC
vpc-52a62636
Noncompliant
September 20, 2016 9:00:15 PM
September 20, 2016 9:00:16 PM
EC2 VPC
vpc-06587f61
Compliant
September 20, 2016 9:02:59 PM
September 20, 2016 9:02:59 PM
EC2 VPC
vpc-763a0913
Compliant
September 20, 2016 9:00:15 PM
September 20, 2016 9:00:15 PM
EC2 VPC
vpc-be0b5ddb
Compliant
September 20, 2016 9:00:15 PM
September 20, 2016 9:00:15 PM
Re-evaluate rule
You can re-evaluate the current status of your resources against the Config rule. Evaluation may take a few minutes.
Re-evaluate

Conclusion / Takeaways

Practical Lessons Learned

- Define an automation strategy for your infrastructure
- Use a CI/CD pipeline; develop orchestration process first
- Incorporate best practices for developing secure code
- Define governance and implement automation for compliance
- Implement with scale in mind

Questions?