Role of SRE in DevSecOps

KALYAN DHOKTE

Practice Head, SRE Digital Engineering

Cognizant Technology Solutions.

DECEMBER 2022

Relationship of DevSecOps and SRE

SRE implements DevOps principals along with security practices for building DevSecOps ecosystem

DevOps Principles with Security practices	SRE Practices implements DevSecOps principals
 **Eliminate Silos** Collaborative approach for security	 Shared responsibility for Security. Embed security in dev teams
 **Accept failure as normal** Continuous state of compliance	 Blameless post-mortem and adopt security best practices and create a uniform security posture
 **Gradual Changes** Implement changes in production gradually	 Reduce cost of failure by incremental changes
 **Automate Everything** Automate security testing and scans	 Enforces automation for manual tasks(toil) Automating security monitoring and testing
 **Measure Everything** and security monitoring in real time	 Enforces robust monitoring of reliability and security through SLO/SLI and actionable alerts & triggers

Why are Modern Cloud technologies challenging?

Modern microservices cloud native architecture increases speed and scale along with complexity with improved cost efficiencies, accelerated innovation, faster time-to-market, and the ability to scale applications on demand

How speed and scale of microservices increase complexity?

Where Does complexity come from ?

- Continuous Delivery
- Service mesh
- Cloud Native - Serverless
- Cloud Containers Kubernetes
- API & Micro Services
- Service Mesh
- Release Engineering
- Continuous Integrations
- Infra as Code
- Blue/Green Deployment
- Automated Testing
- Observability

How it Changes in years with higher complexity?

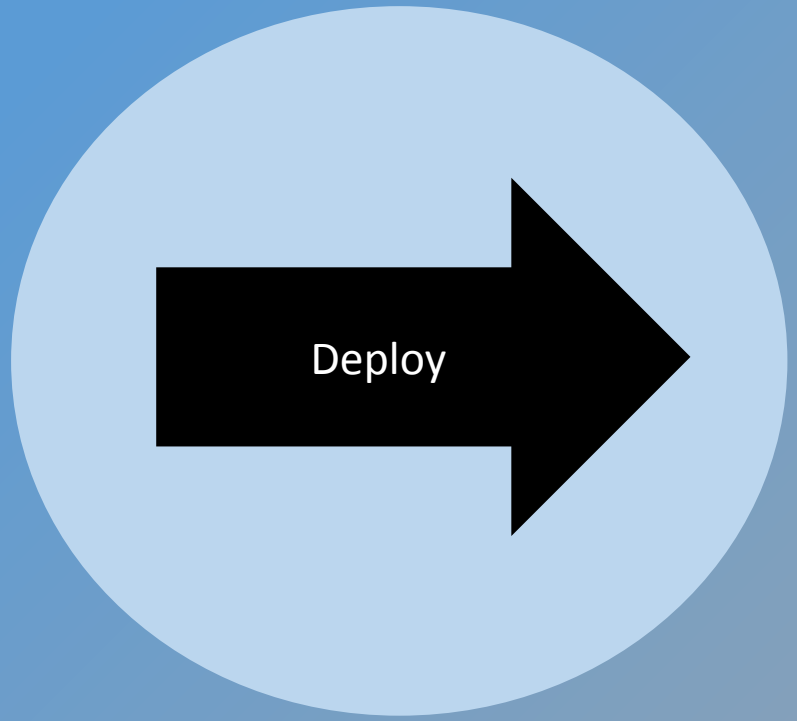
- Change in Deployment Tools
- Cloud Service Outage
- Delayed Feature
- Upgrade of Kubernetes
- Increased Number of Microservices
- Expired certificates
- Scalability issues
- Refactoring
- Regulatory Audit
- WAF Outage
- Firewall Outage
- Misconfigurations
- Security

How I navigate the complexity ?

By Apply DevSecOps and SRE practices

DevSecOps + SRE

- Flexibility to Change System Rapidly
- Apply security context to System Changes
- Resiliency & Reliability driven Development
- Building applications for failures'

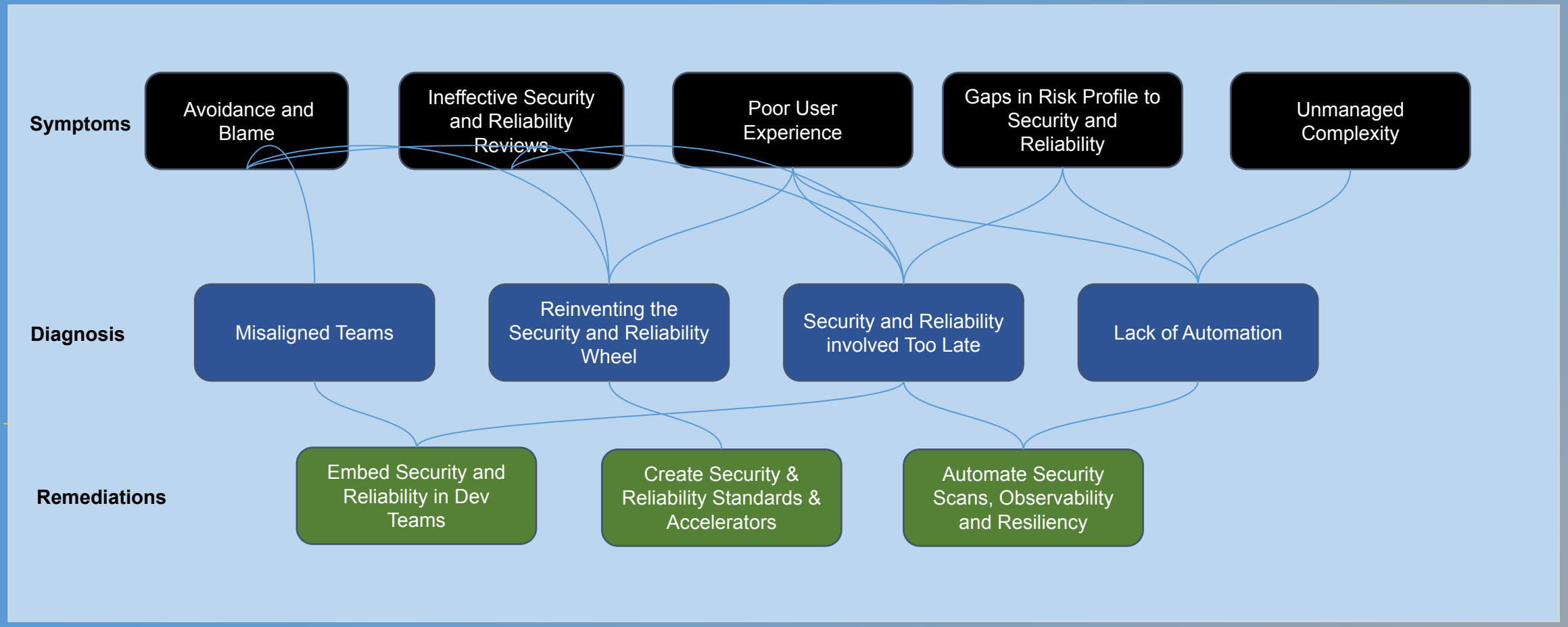


SRE

- Availability, SLO Tracking, Error budgets
- Observability includes Automated Security monitoring
- Securing Production System from Unauthorized Attacks
- Proactive Problem Prevention and Self Healing

Reliability and Security-First Mindset must be infused in development, deliveries and production

Common Symptoms for failing DevSecOps and SRE Models

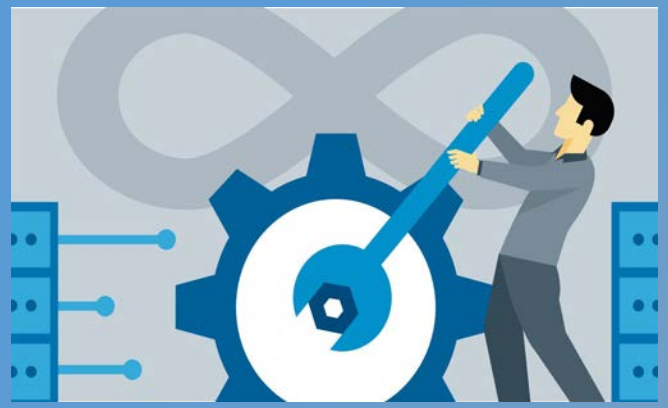


A healthy DevSecOps operating model embeds SREs into dev teams to help implement Security and reliability considerations early in the design process.

Common practices for success of DevSecOps and SRE Models

Embed Security in Development

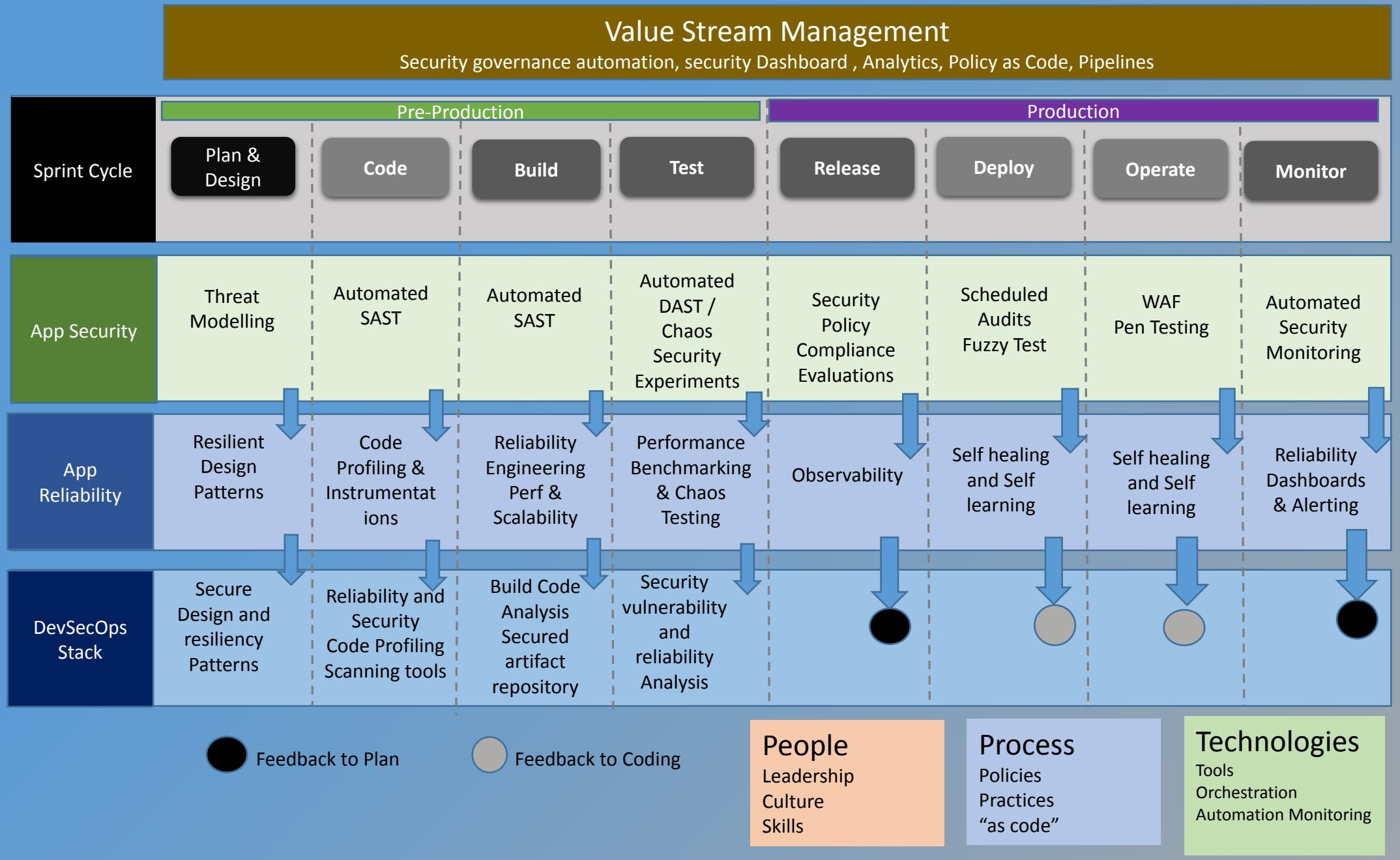
Reliability Driven Development



Automation
(Security & Reliability Testing)

Measure Everything
(Security & SLA Monitoring)

DevSecOps Operations Service Model with Application Reliability and Security Stack



Advanced DevSecOps and SRE Automation in Deployment pipeline

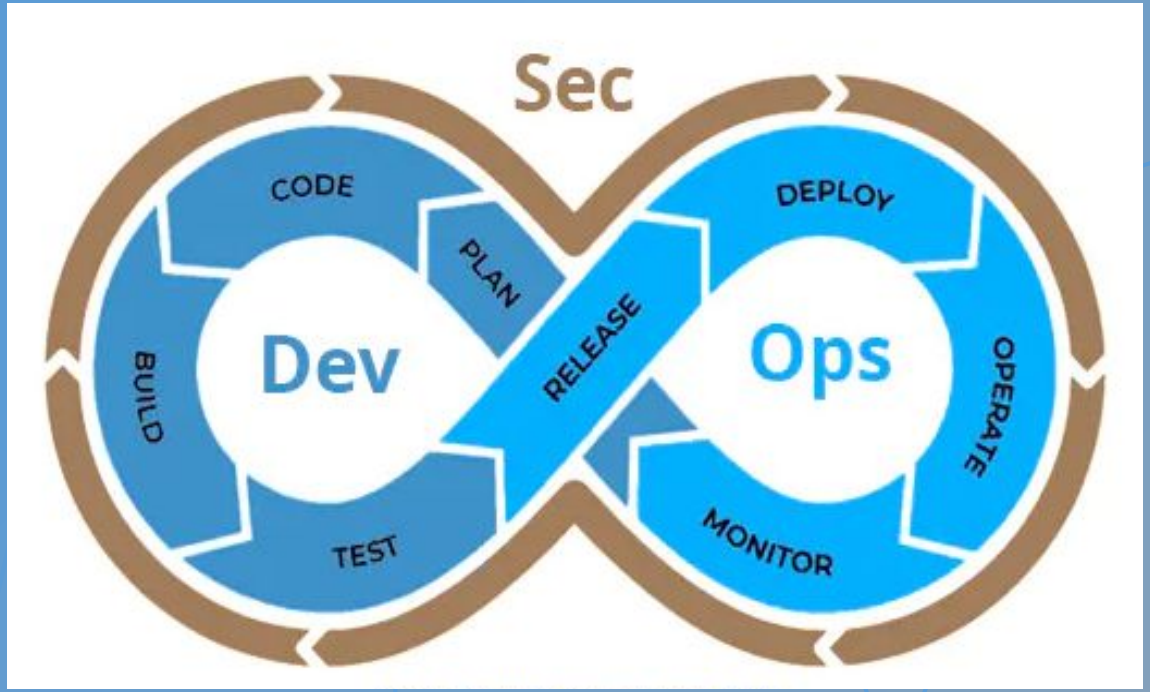
SREs monitors configuration drift and anomalies that triggers relevant responses back to CI/CD Pipeline

Automating Security Gates to stop the workflow from slowing down and backing in monitoring and analytics into the pipeline.

Automated code verification checks into DevSecOps frameworks

Reliability driven

Development - Code Profiling tool integrations



Automated Security

Monitoring and Analytics

SLO Tracking Reliability Gates and Operational Excellence Dashboards

Auto- auditing and compliance tools that streamlines compliance reporting

AI-backed threat analysis - to proactively identify code vulnerabilities

Automated Chaos Engineering in DevSecOps pipeline with tools like Gremlin, Chaos Sling

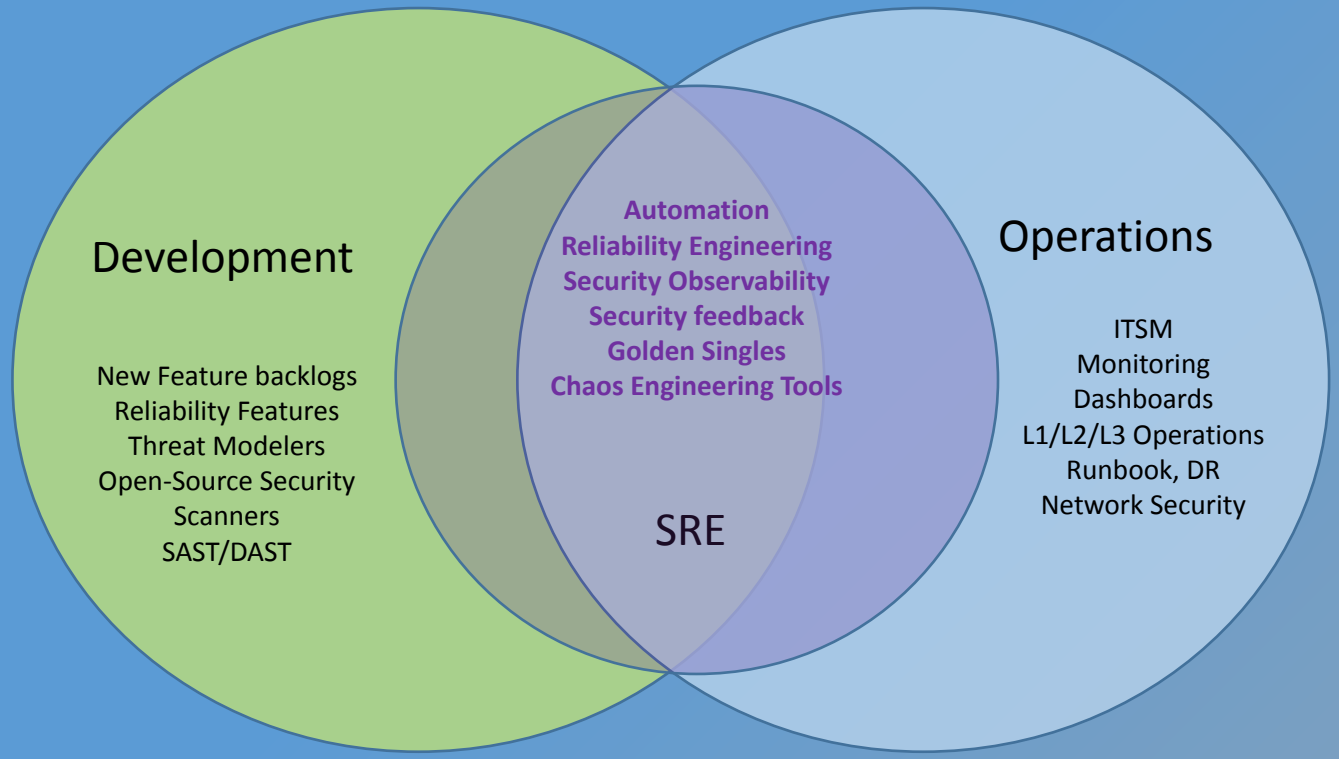
Define Security Policy and and maximize Compliance

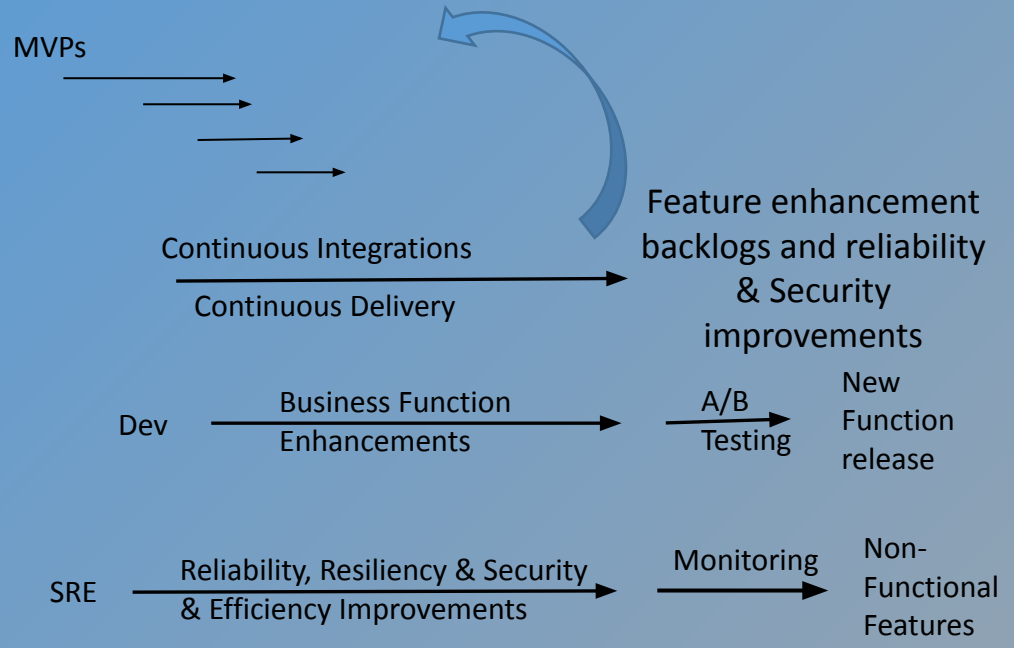
Deployment script validations against Security Policy Framework

Cost Savings Potential.

DevSecOps and SRE automations helps to lower likelihood of a catastrophic cybersecurity incident and the reduction in the number of operations staff

SRE Role in DevSecOps – Team Topology



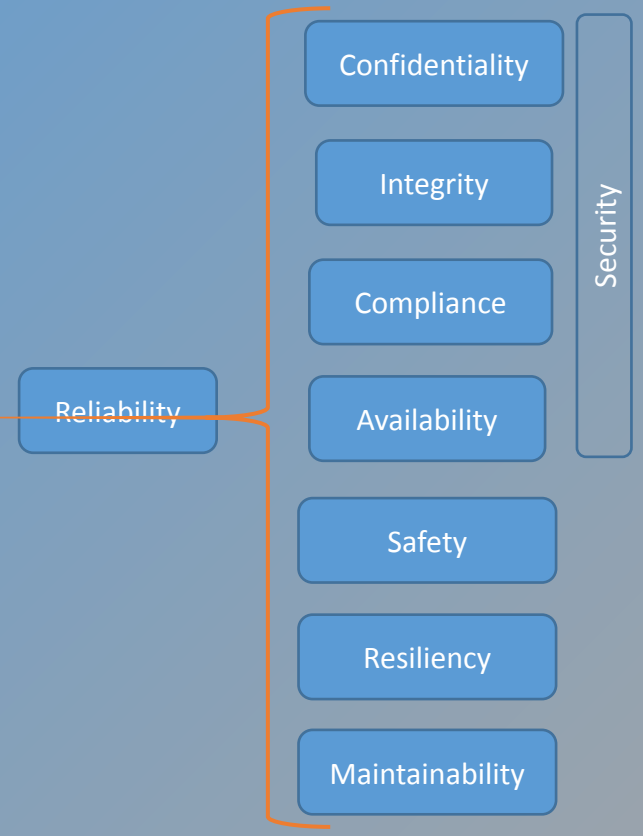


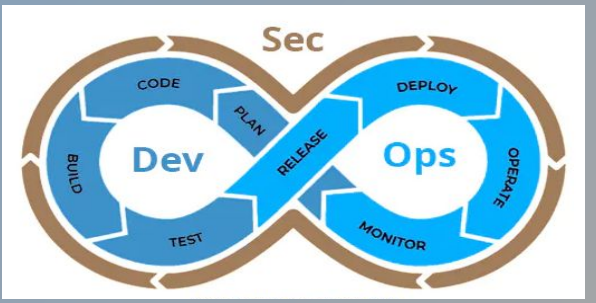
The role of SRE's is to collaborate, engage in value-added activities, and create results that contribute to measurable reliability improvements

SRE Role in DevSecOps – backlogs

SRE tasks in different phases for improving reliability and resiliency of applications, securing build pipeline, securing the deployment and Runtime Protection

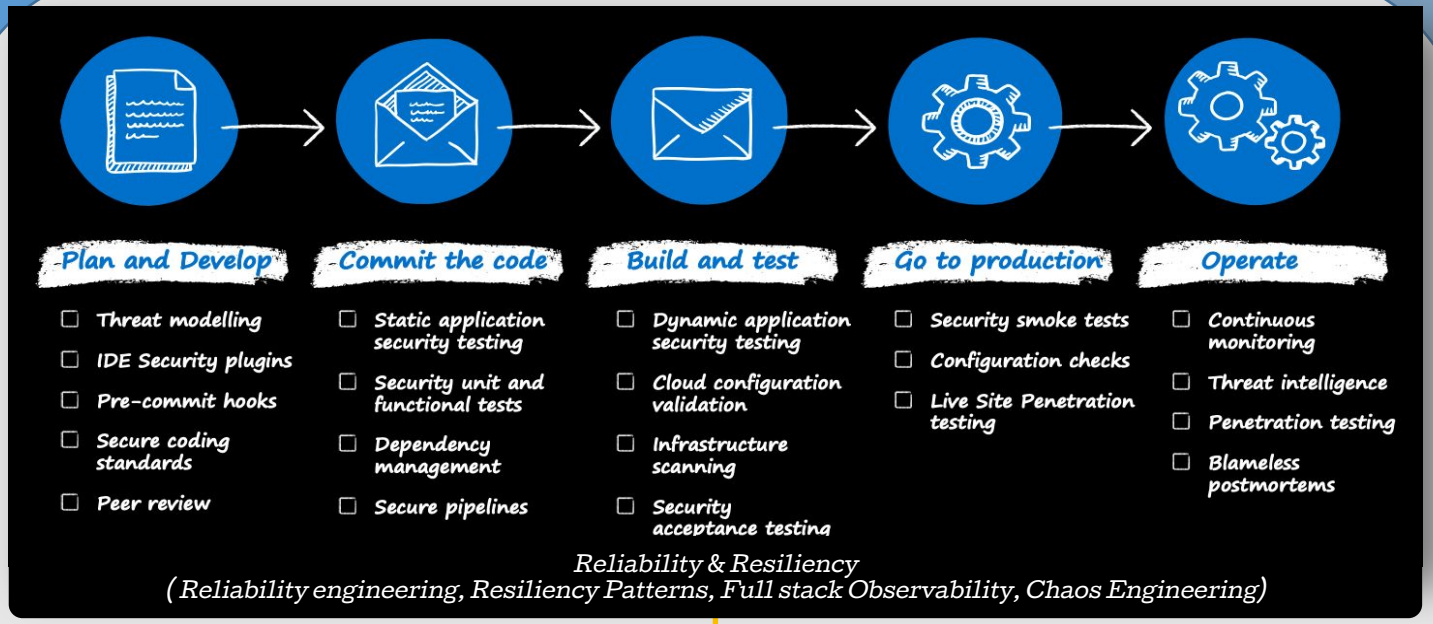
Area	SRE Role in DevSecOps Stack	Phases
Reliability	• Reliability & Security elements in Design and Architecture, NFRs driven product life cycle • Implement observability, Code profiling, scalability, recovery and self-healing capabilities • Predictive and Trend Analysis to measure SLIs and optimize it.	• Build, Deployment & Operations
Resiliency	• Risk based vulnerability assessment to proactively identify potential failure points • Design Fault tolerant capability using resilience frameworks • Chaos engineering to cloud security leverages feedback loops to execute, monitor, analyse and plan security fault injection	• Build, Deployment & Operations
Security	• Enable Security Code Scanning and Security Policy for PODs running on AKS Kubernetes • Configure and validate deployment scripts for Security Context • Configure ingress/ egress Network Policy and WAF Policy • Analysis of vulnerabilities detected via security fault injection has been used to harden the security of cloud resources • • Security Monitoring Dashboards (SEIM)	• Build, Deployment & Operations





Case Study : Building Security and Reliability In DevSecOps Pipeline

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: {{ .Values.serviceName }}
  namespace: {{ .Values.namespace }}
spec:
  minReadySeconds: 5
  progressDeadlineSeconds: 600
  replicas: {{ .Values.replicaCount }}
  revisionHistoryLimit: 10
  selector:
    matchLabels:
      app: {{ .Values.apiName }}
      name: {{ .Values.serviceName }}
  strategy:
    rollingUpdate:
      maxSurge: 1
      maxUnavailable: 1
    type: RollingUpdate
  template:
    metadata:
      annotations:
        apparmor.security.beta.kubernetes.io/pod: runtime/default
        container.apparmor.security.beta.kubernetes.io/service: runtime/default
        seccomp.security.alpha.kubernetes.io/pod: docker/default
        container.seccomp.security.alpha.kubernetes.io/service: docker/default
      labels:
        aadpodidbinding: azure-pod-identity-binding-selector
        app: {{ .Values.apiName }}
        name: {{ .Values.serviceName }}
    spec:
      automountServiceAccountToken: false
      securityContext:  # to be added
        runAsUser: 1000 # to be added
        runAsGroup: 3000 # to be added
        fsGroup: 2000 # to be added
        runAsNonRoot: true # to be added
        allowPrivilegeEscalation: false # to be added
        readOnlyRootFilesystem: true # to be added
        capabilities:
            drop:
              - ALL
      volumes:
        - name: secrets-store-inline
          csi:
            driver: secrets-store.csi.k8s.io
            readOnly: true
            volumeAttributes:
```



1

Threat Modelling Tool

2

YAML Validator Security Context Schema

3

Chaos Engineering for Resiliency and Security Experiments

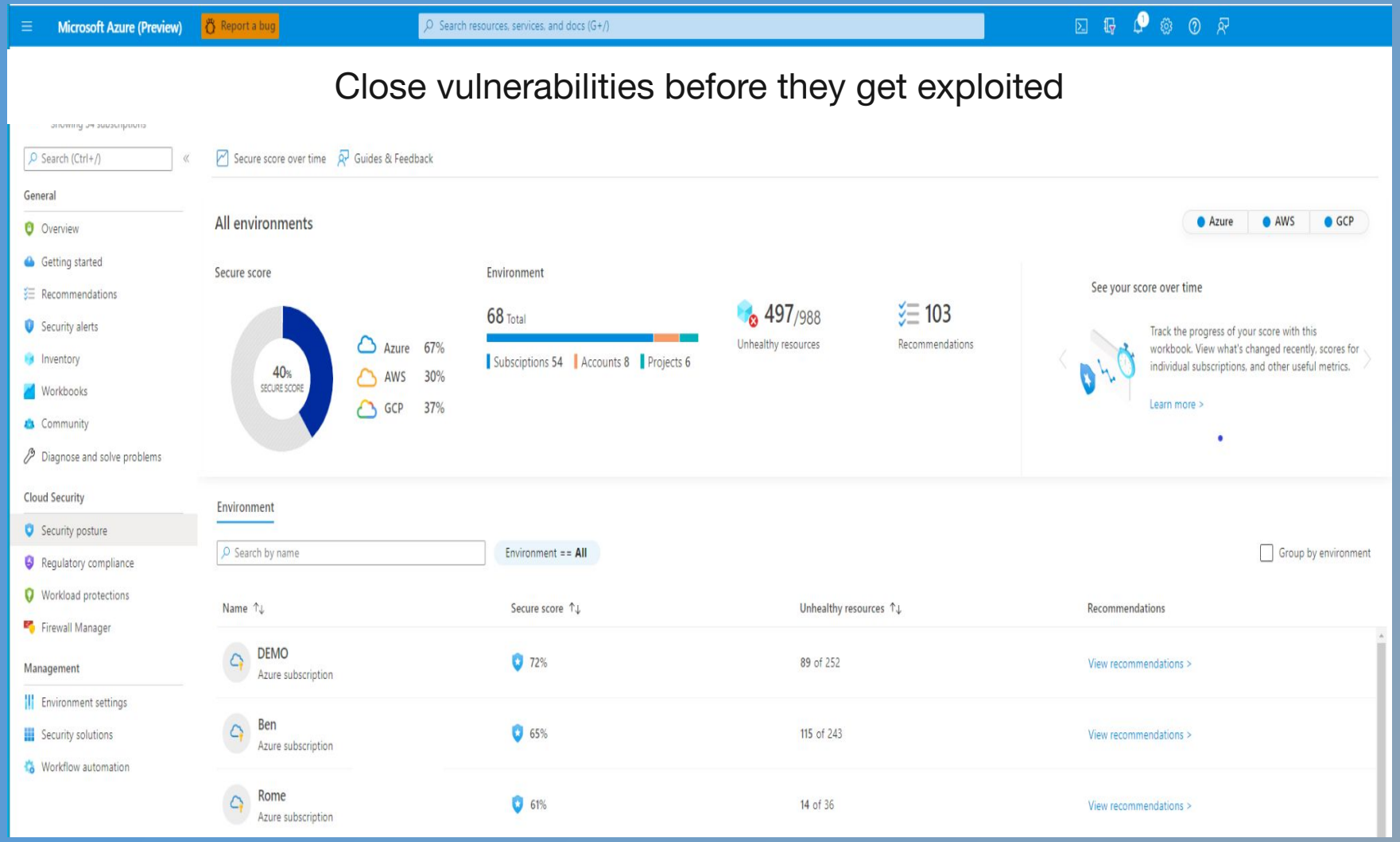
ChaoSlingr

Gremlin

Integrated YAML Schema Validation to check Security Context in Release Pipeline

Security Vulnerabilities
Fault Injection and Simulations for Reliability

Case Study : Building Security and Reliability In DevSecOps Pipeline

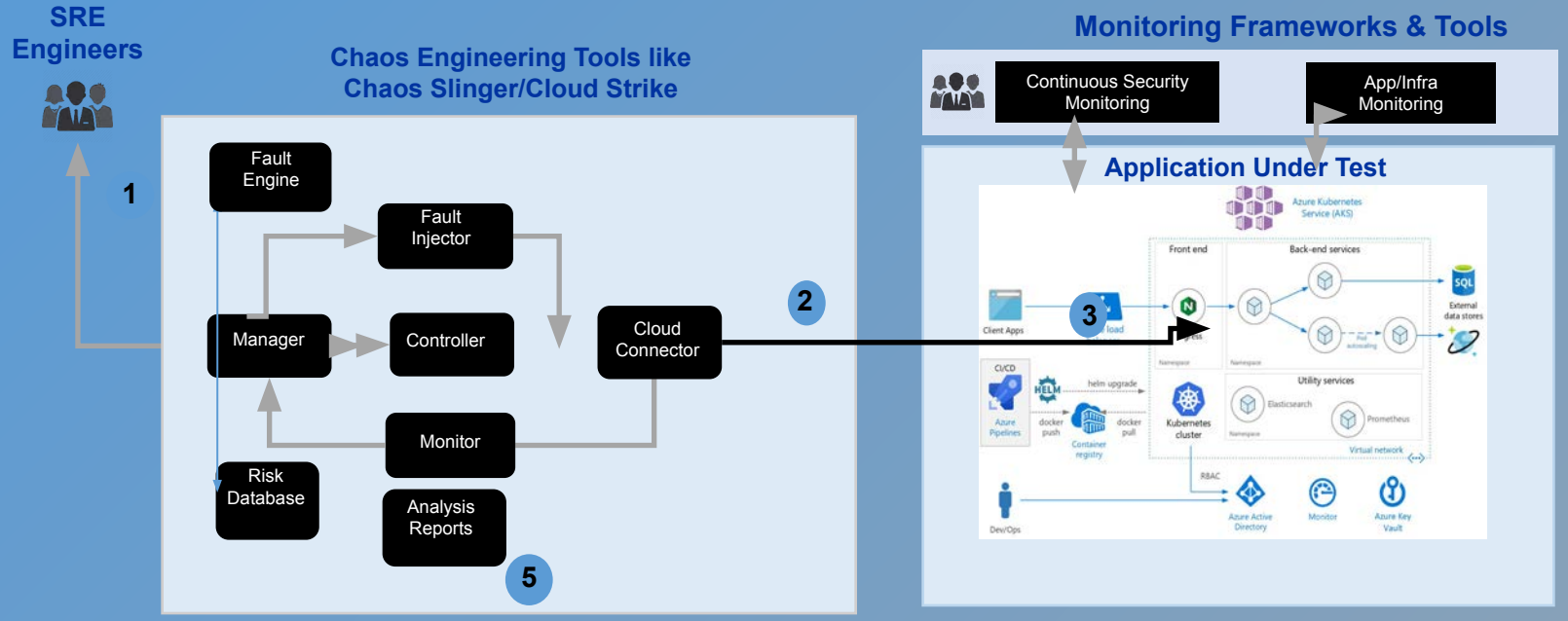


Enforce Security Policy

1. Know your Security Posture, Continuously assess and identify vulnerabilities
2. Harden resources and services with Security benchmark
3. Detect and resolves threats to resources and services

Case Study : Building Security and Reliability In DevSecOps Pipeline with SRE Capabilities

Open-Source Security Experimentation Tool



1 SRE engineers design security test scenarios based on Risk based Fault Vulnerability Assessment

2 Tool injects Security Faults Attack Points into cloud infrastructure

3 Test validates the rules that are designed to check if there are any security alerts

4 Security Vulnerability conditions are simulated with the help of tool Like - Misconfigured Access Control Policies, Over-privileged Users and Configuration-based vulnerabilities

5 SRE Engineer monitors application impact to the failures and validates fault tolerance capabilities of application. Possible Recommendations include updating security rules for security groups (cloud firewalls), restriction of access to overly permissive access control policies

Sr. No.	Test Category	Scenario
1	Spoofing of user identity and other entities	Compromise default privileged service and user accounts Like Azure Storage Account Keys
2	Tampering	Alter data in datastore (S3, RDS, Redshift
3	privacy breach or data leak	Misconfigured and default security groups and access lists
4	Denial of service	Destroy cloud services configuration, datastores and/or accounts
5	Elevation of privilege	Add users, assets or accounts to existing roles or groups with higher privileges
6	Validating baseline security requirements	Assign a public IP to a compromised / internal resource

Case Study : Building Security and Reliability In DevSecOps Pipeline with SRE Capabilities

Scenario - 1

Misconfigured Port Change

1. List available NSG (Network Security Groups) for Kubernetes Nodes
2. Select only those security groups that are tagged with Opt-in Tag for Chaos
3. Randomly Select NSGs
4. Apply Random Open or Close Action based on port configurations
5. Chaos Slinger – Applies the configured changes
6. Generator – Starts the experiment and performs port acquisition and changes in port
7. Tracker – Tracks changes. Verify events triggered in Security Center Alerts

Scenario - 2

S3 Bucket Permission changes

1. Create a new user
2. get a list of all the buckets in the cloud
3. select a random bucket from the set of buckets in the cloud
4. Configure Attack Points Chaos Tools
5. Simulate bucket unavailability e.g. by changing bucket ACL from ALLOW to DENY
6. Tool applies the configured changes
7. Tool starts Security experiments
8. Get real-time insights of chaos engineering experiments using AWS CloudWatch
9. Verify events triggered in Security Center Alerts

Testing is assessment or validation of an expected outcome, while Experimentation seeks to derive new insights and information that were previously unknown

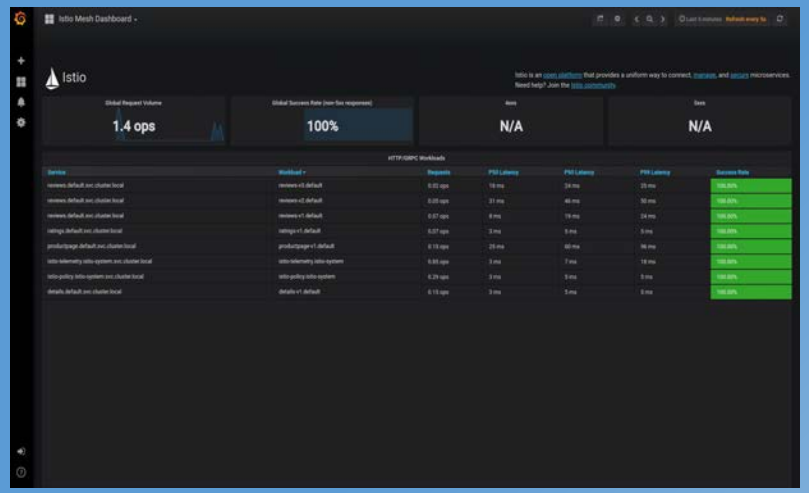
Case Study : Building Security and Reliability In DevSecOps Pipeline with SRE Capabilities

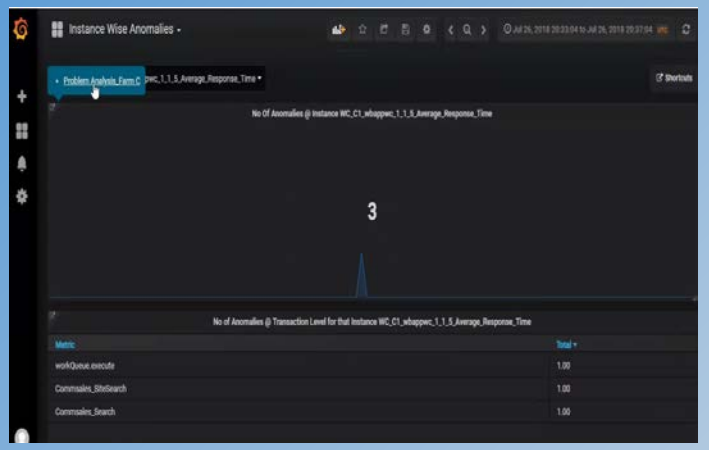
Customized SLO monitoring Dashboards

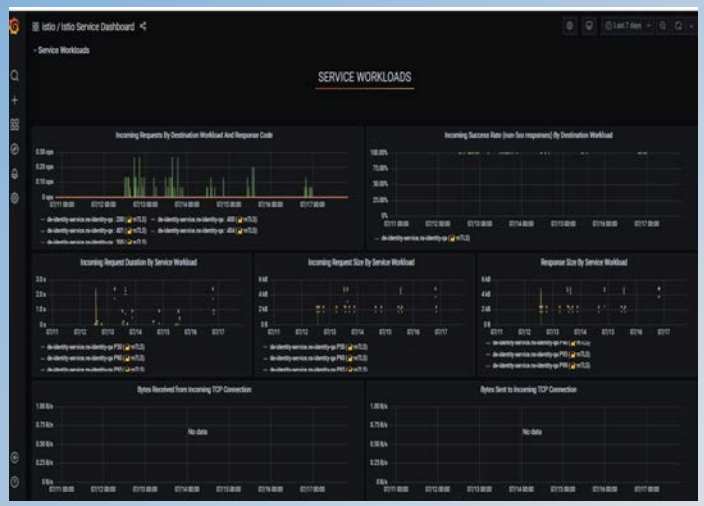
Use of Automated Root Cause Analysis using Anomaly detection and Suspect Ranking

Automated Machine Learning is applied using Python (Scikit-learn) algorithms to correctly point out the root cause of problems that occurred in the past

Automated RCA Dashboards







Summary

- ✔ Culture aspect of SRE to embed security along with reliability
- ✔ Implement SRE practices in DevSecOps like
  - Security Chaos Engineering
  - Improve Organization’s defense
  - Zero Trust Network
- ✔ Engineering Reliability and Security into design and deployment processes
- ✔ Eliminate reliability and security as a bottleneck with continuous delivery as a pipeline
- ✔ Bridge gaps with security practices while ensuring quick and safe deliverables
- ✔ Replace siloed teams with increased collaborations and shared security responsibility and reliability driven development

“SRE work is like being a part of the world’s most intense pit crew. We change the tires of a race car as it’s going 100mph.”

Andrew Widdowson, SRE at Google

Thank you


@kdhokte


https://www.linkedin.com/in/kalyan-dhokte-a443498/


@kdhokte#9987