RSAConference2016
San Francisco | February 29 – March 4 | Moscone Center

Connect to Protect

SESSION ID: CSV-F03

DevSecOps in Baby Steps

Hart Rossman
Amazon Web Services Global Practice Manager- Security, Risk, and Compliance
@HartDanger

#RSAC

The Journey: Baby Steps to Parkour

- Getting to DevOps
- DevOps to DevSecOps
- Planning your Epics & Sprints
- Use Cases & Examples

In The Beginning There Was NoOps

Security program – Ownership as part of DNA

Distributed

Embedded

- Promotes culture of “everyone is an owner” for security
- Makes security stakeholder in business success
- Enables easier and smoother communication

Operating with Shared Responsibility

Responsibility & Accountability

- Own it.
- Govern it.
- Not my monkeys; not my circus.

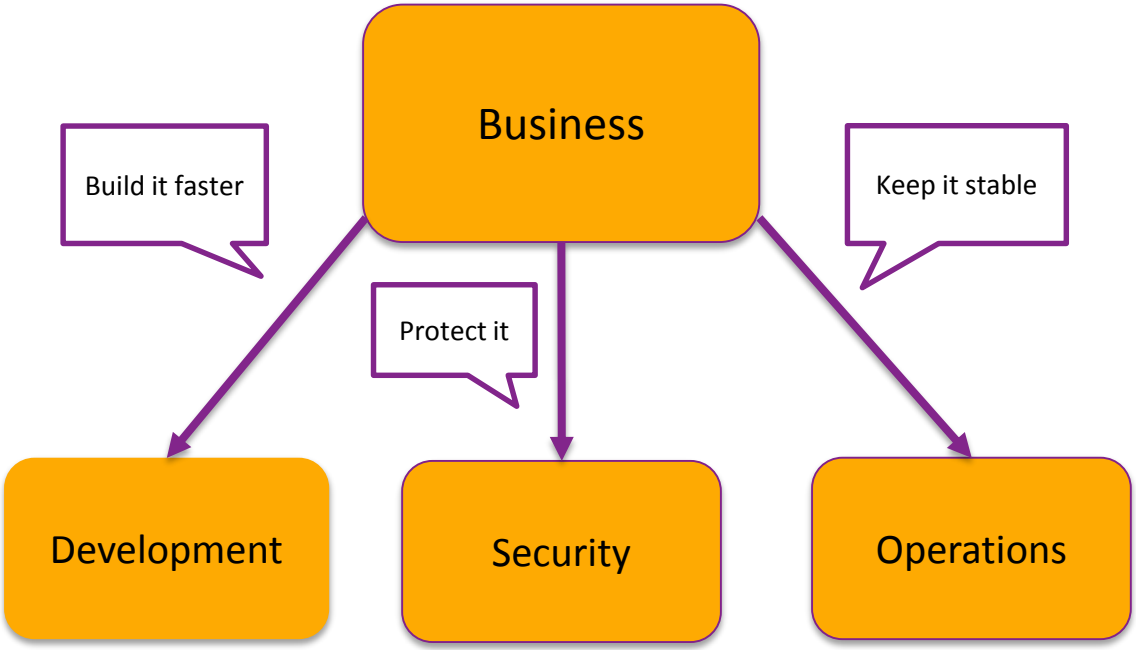
How do I know?

- Do I carry a pager for this service?
- Do I make the rules?
- Should I be consulted or informed?

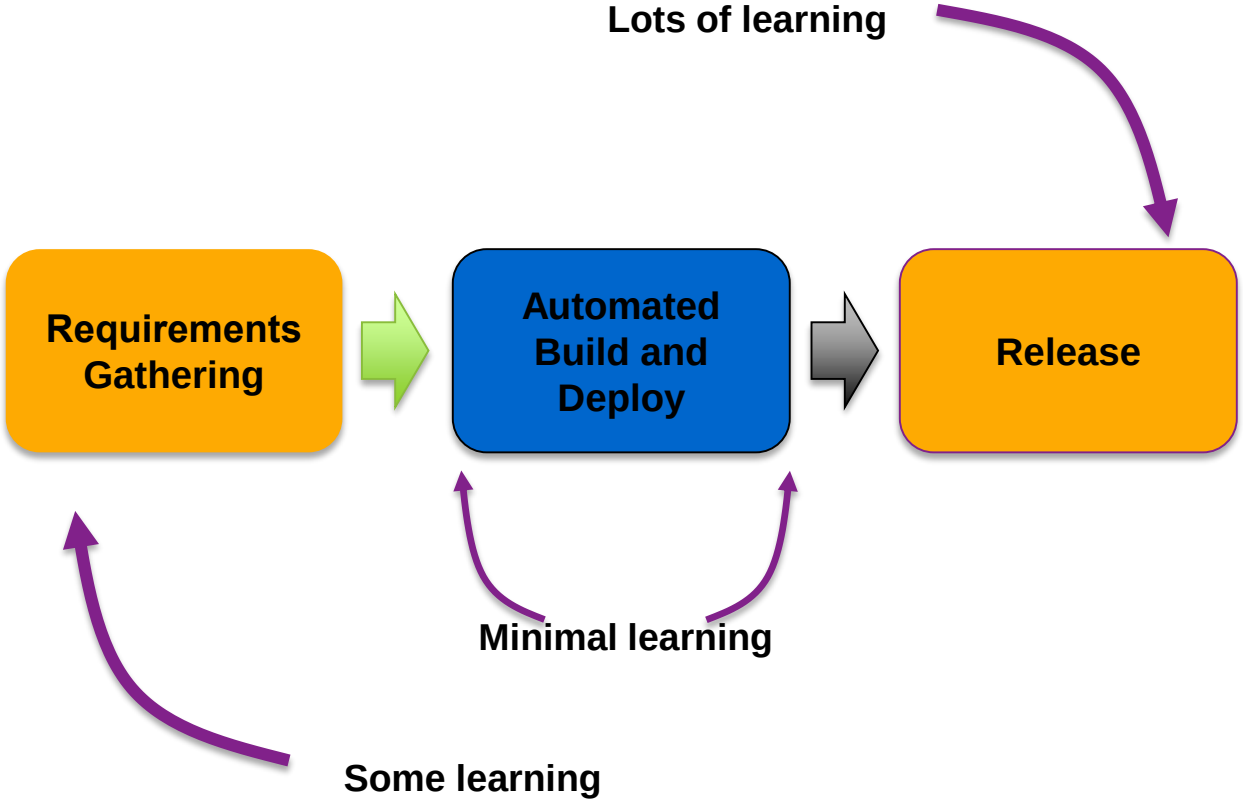
Security as code

1. Use the cloud to protect the cloud
2. Security infrastructure should be cloud aware
3. Expose security features as services via API
4. Automate everything so everything scales

Security as code: Innovation, stability, & security

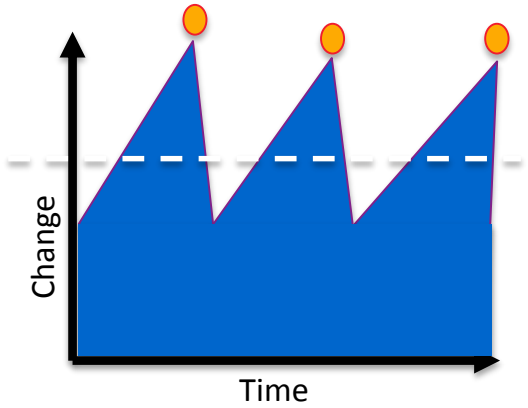


Security as code: A shorter path to the customer



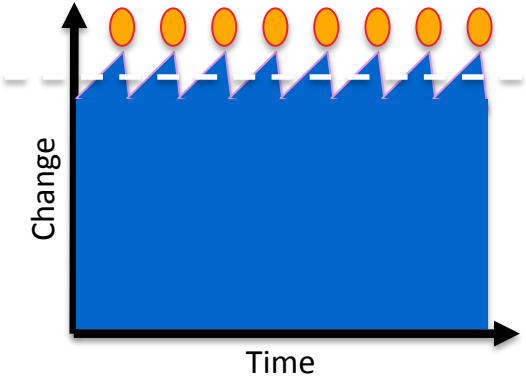
Security as code: Deploying more frequently lowers risk

Rare release events: "Waterfall methodology"



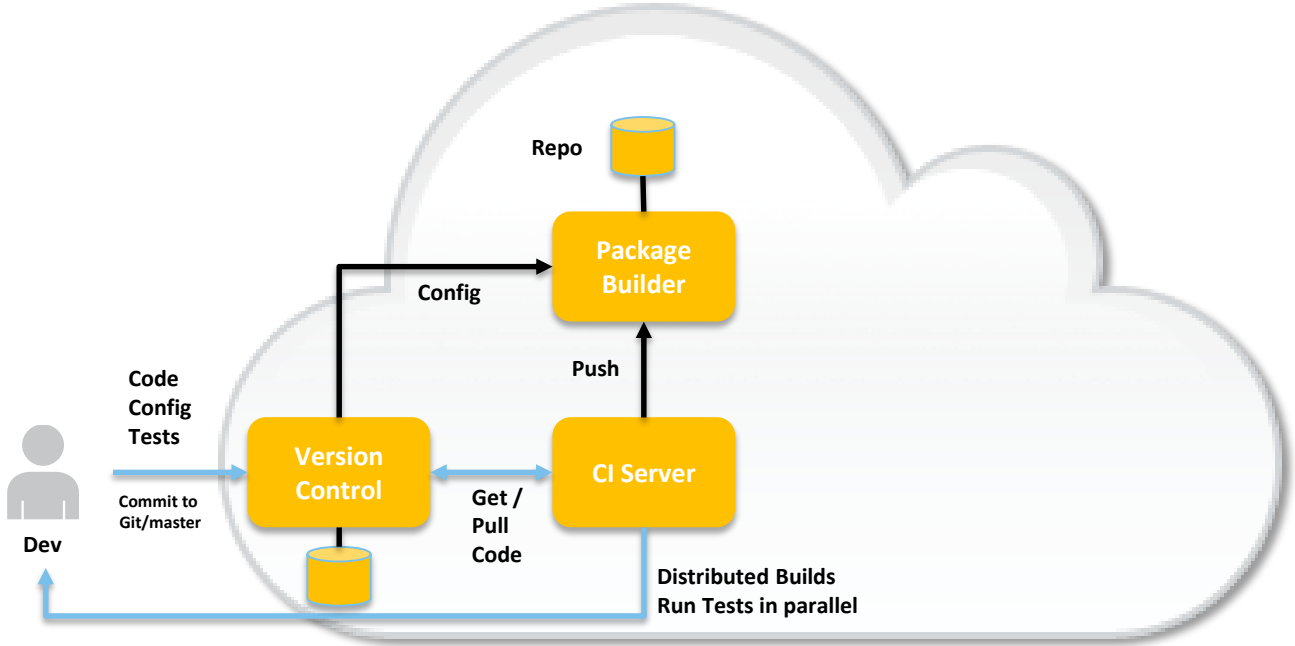
Larger effort
"Increased risk"

Frequent release events: "Agile methodology"



Smaller effort
"Minimized risk"

Continuous Integration



Send Build Report to Dev
Stop everything if build failed

What does CI give us?

Confidence that our code changes will build successfully

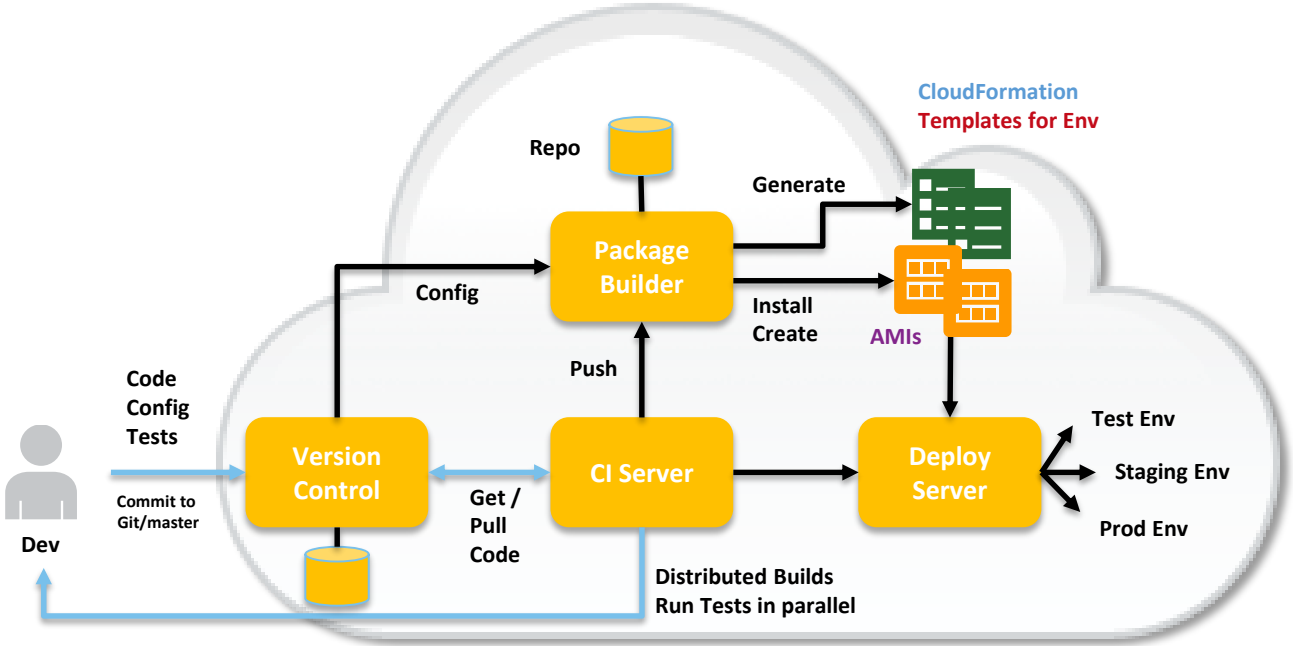
Increasing velocity of feedback cycle through iterative change

Bugs are detected quickly

Automated testing reduces size of testing effort

Very fast feedback on the things we can test immediately

Continuous Delivery



What does CD give us?

Automated, repeatable process to push changes to production

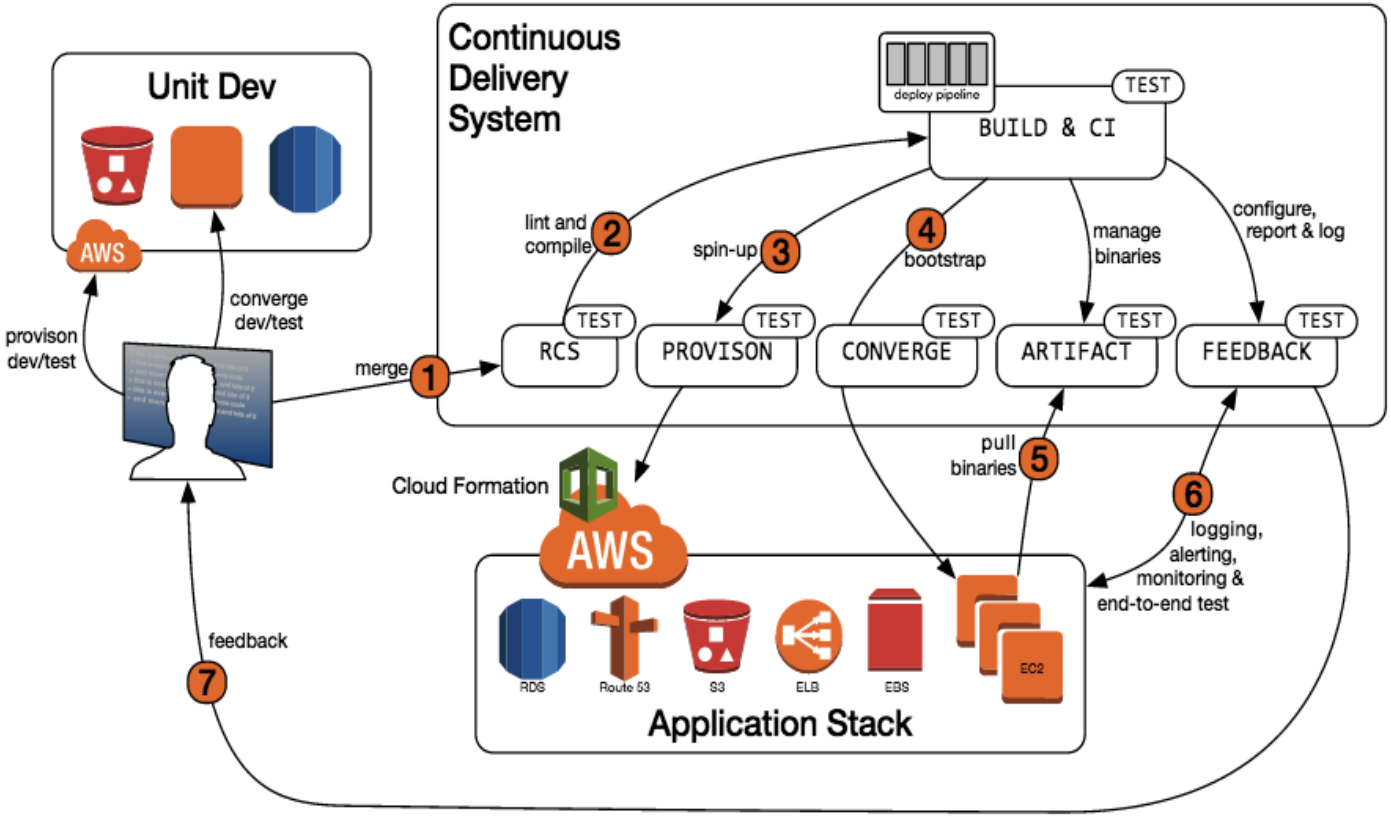
Hardens, de-risks the deployment process

Allows detection of failure as quickly as possible in the build process

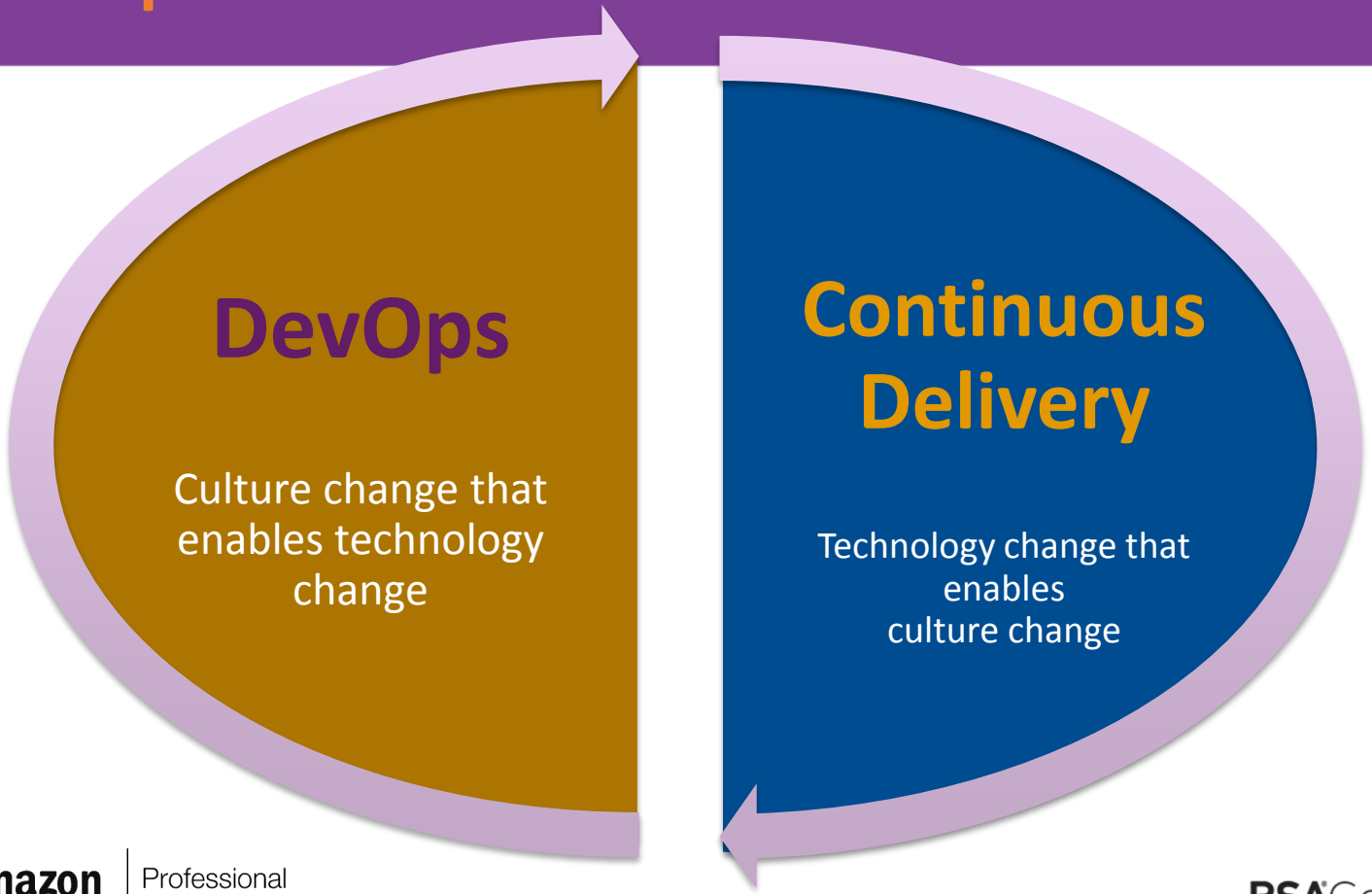
Supports A/B testing or “We test customer reactions to features in production”

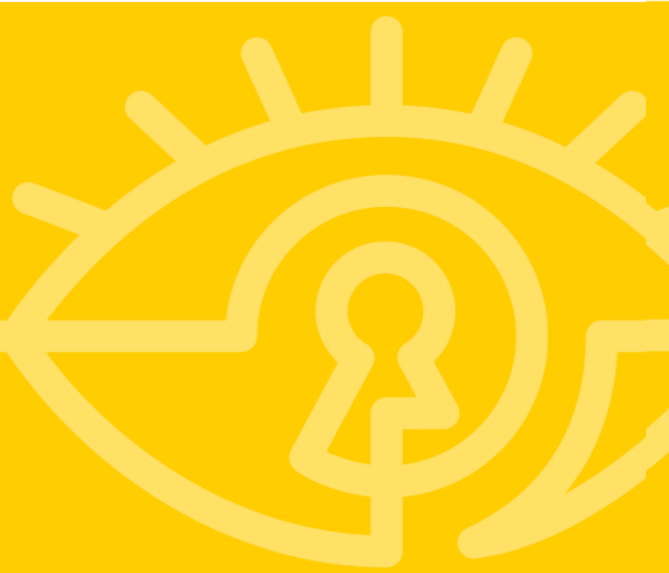
Gives us a breadth of data points across our applications

Continuous Delivery System



DevOps


DevOps
Culture change that enables technology change
Continuous Delivery
Technology change that enables culture change

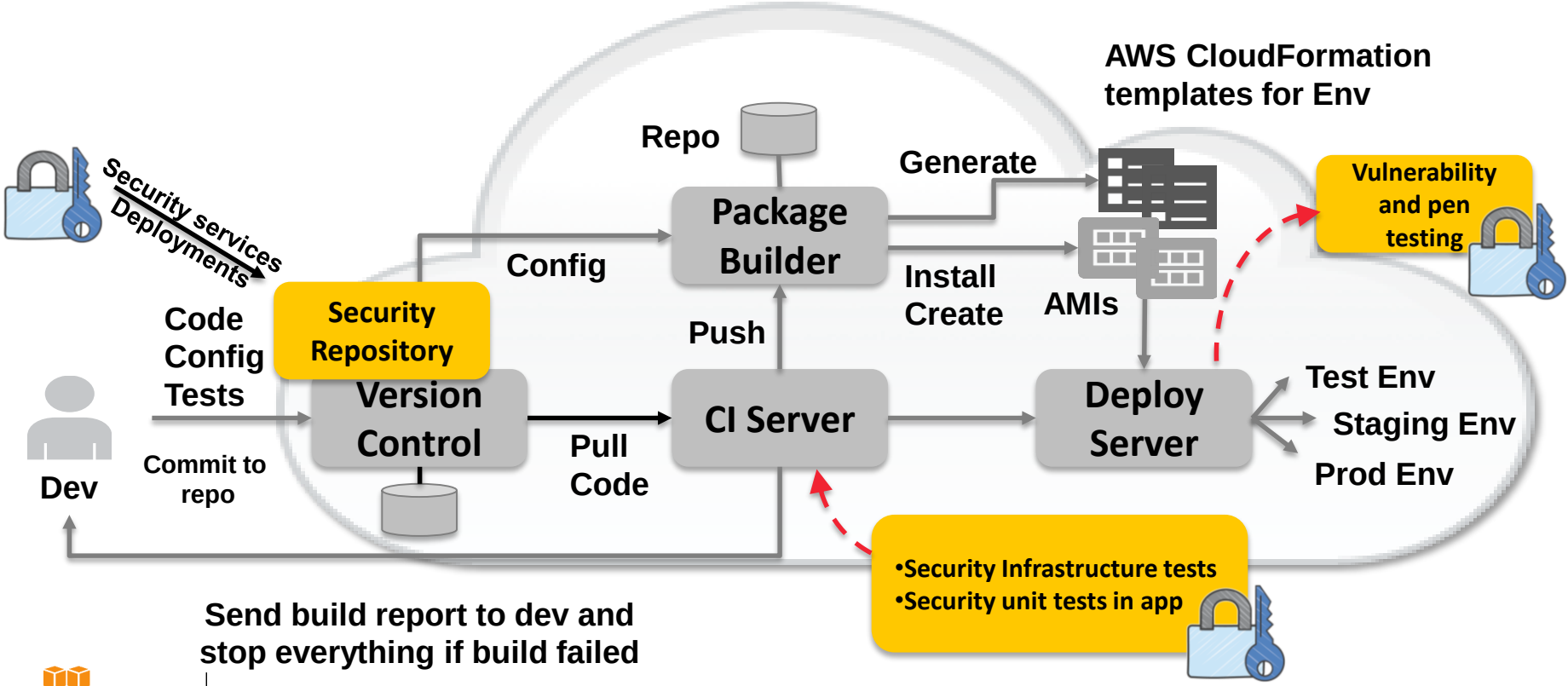


DevOps to DevSecOps

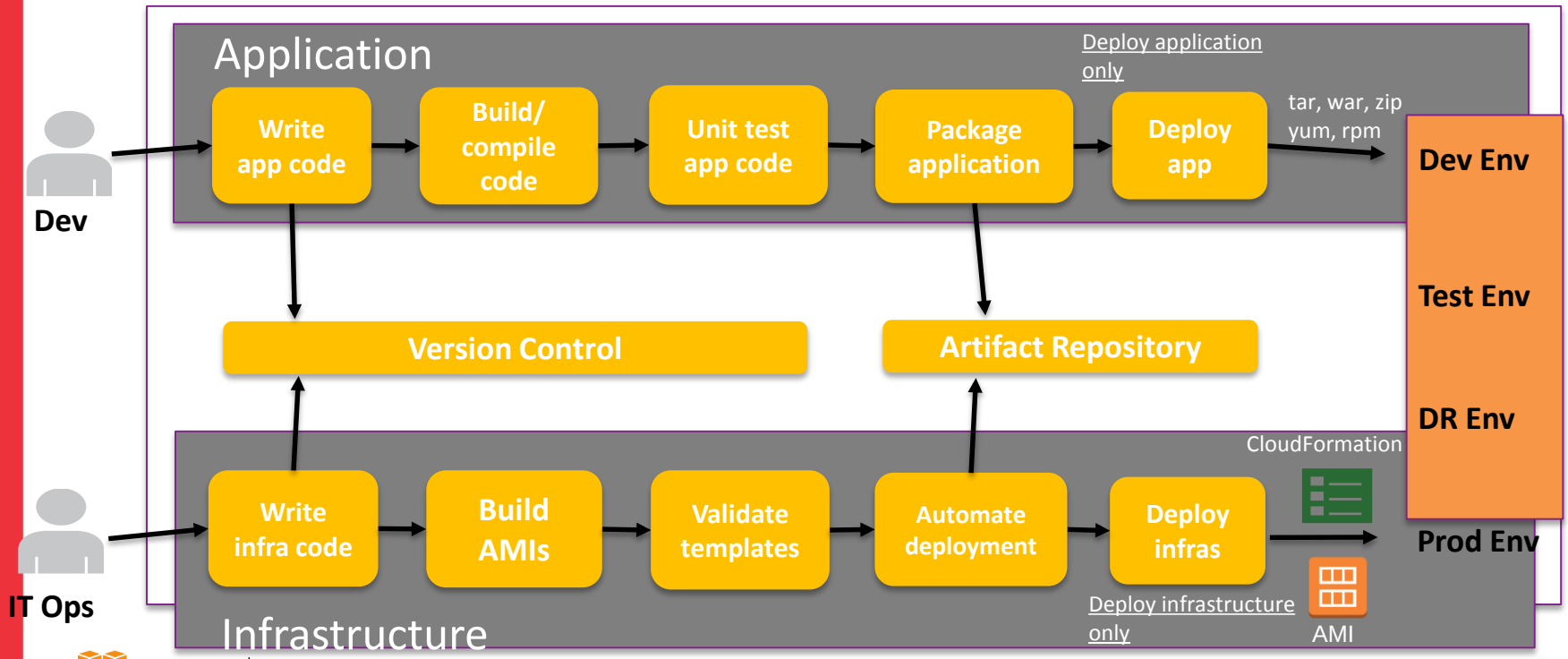
DevSecOps: Core Principles

1. Secure the toolchain
2. Armor up the workloads
3. Deploy your security infrastructure through the toolchain

DevOps→ DevSecOps



CI/CD and automation for security infrastructure



Building DevSecOps teams

- Make DevOps the security team's job.
  - No siloed/walled off DevOps teams.
- *Encourage {security} developers* to participate openly in the automation of operations code.
- *Embolden {security} operations* participation in testing and automation of application code.
- Take pride in how *fast and frequently* you deploy.

Planetary Scale from Day 1

Build Security Services

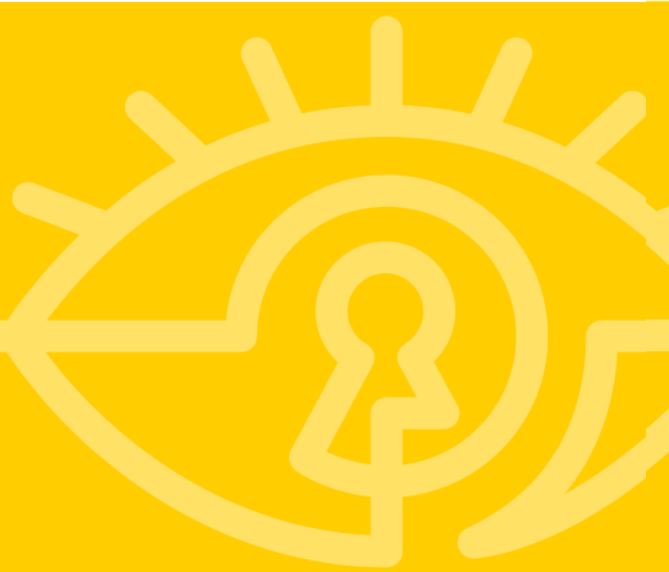
Expose features as API

Plan for Scale

Know your Customers

Utilize customer feedback to Iterate

Internalize your Metrics, let them guide you



Planning your Epics & Sprints

Security as code: Agile user stories

1. Epics vs. stories

An epic is delivered over many sprints; a user story is delivered in one sprint or less.

Icebox→ backlog → sprint

2. Product owner

The product owner decides the priority of each story, is responsible for accepting the story, and is responsible for defining the detailed requirements and detailed acceptance criteria for the story.

Security as code: Agile user stories

3. Persona (or role)

A persona/role is a fictitious user or actor within or of the system.

4. Acceptance criteria

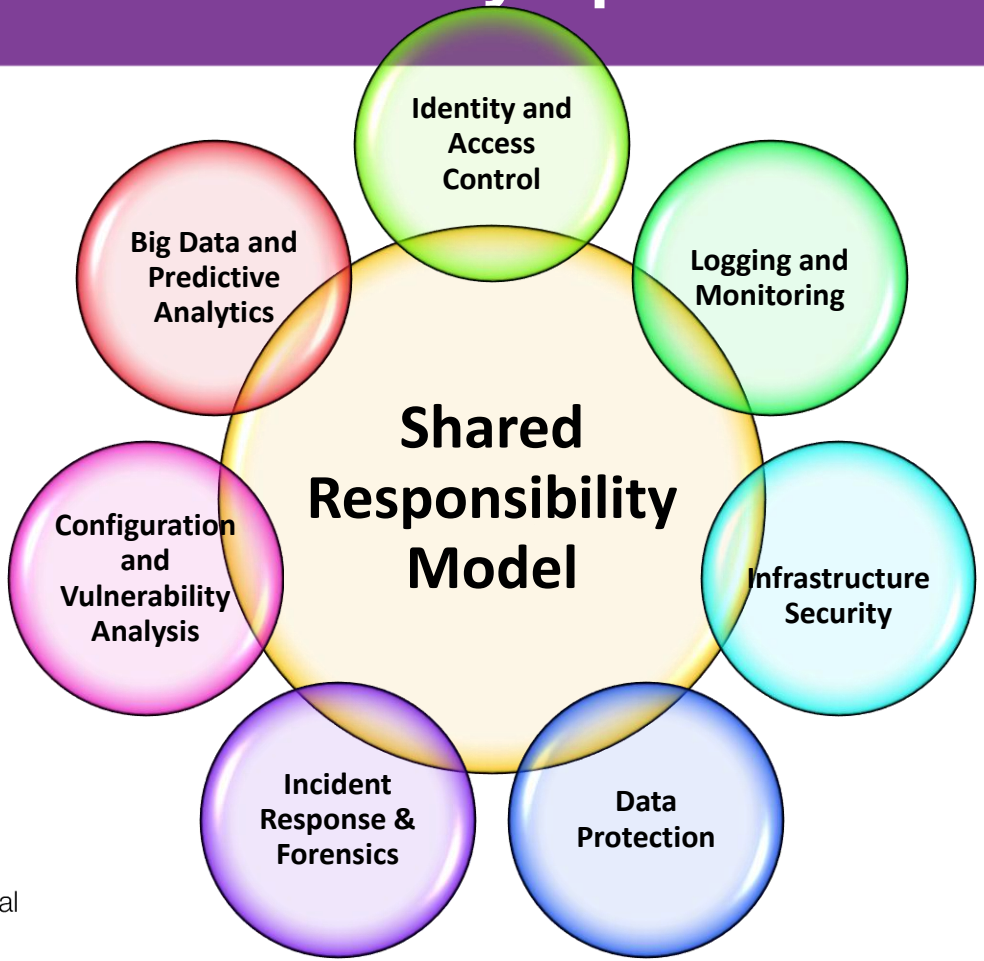
What does good look like? How will we know?

5. Summary format

Every story should have the same summary format:

As a (persona/role) I want (function) so that (benefit).

Security as code: Security Epics



Getting Started: IAM

Story: As a IAM administrator I want to continually reduce the scope of access for humans even as our platform grows. Passwords and access keys that have not been used recently might be good candidates for removal.

Sprint 1: Get credential reports and flag credentials not used in last 45 days.

Other sprint ideas can be found at: http://docs.aws.amazon.com/IAM/latest/UserGuide/best-practices.html

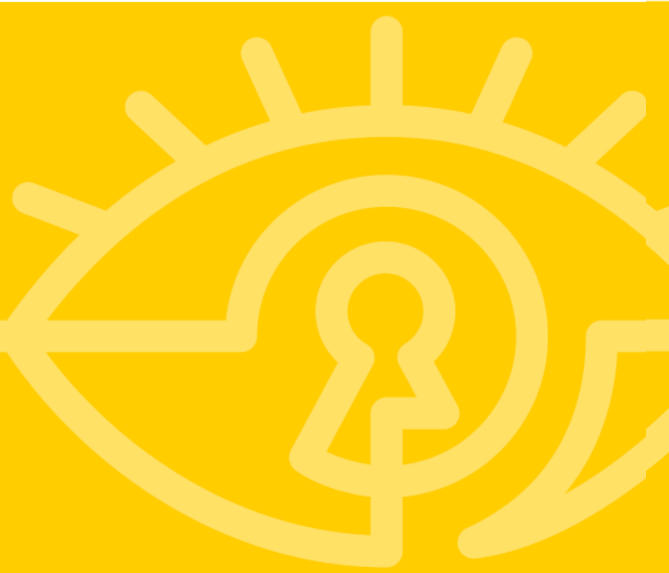
Getting Started: Logging & Monitoring

Story: As a security analyst I want to monitor interactions with AWS API so that we can baseline user behavior

Sprint 1: Enable AWS CloudTrail globally

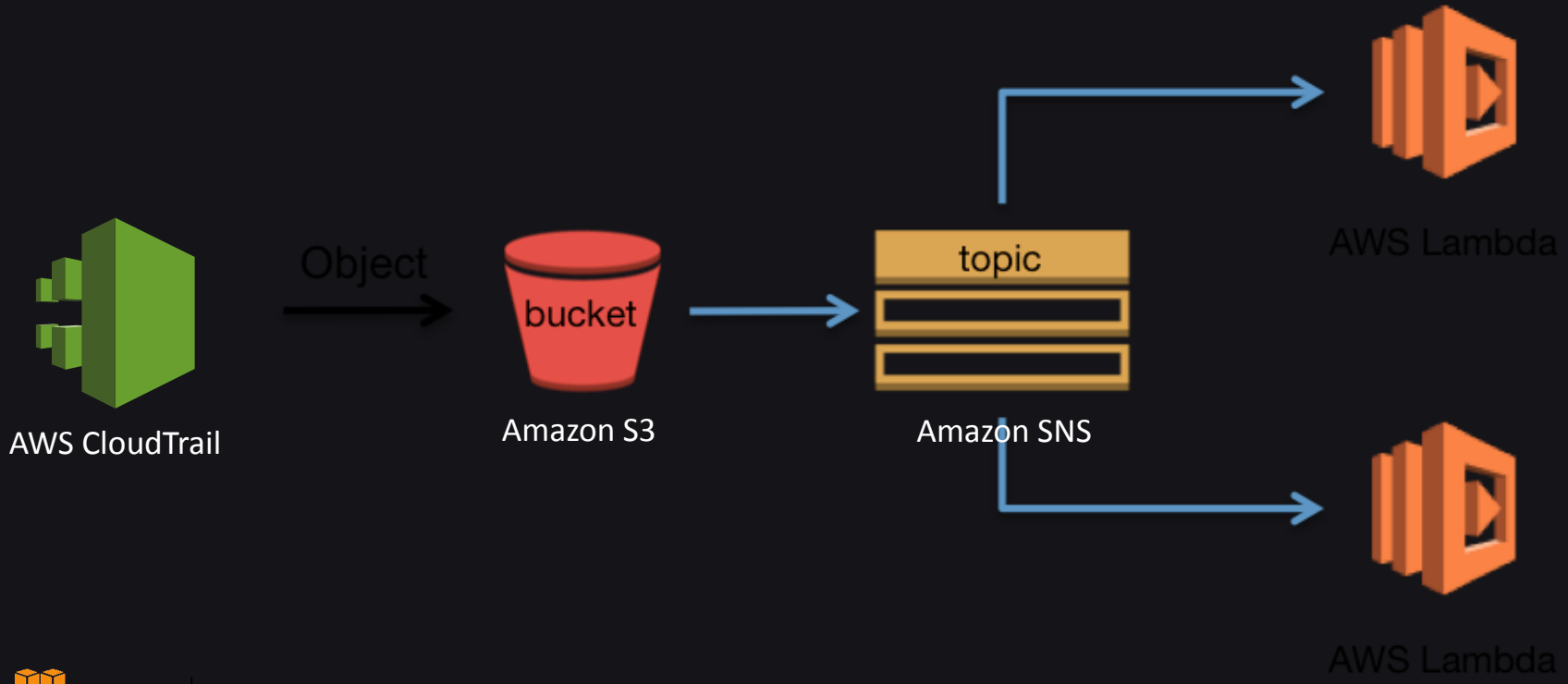
Story: As a security operations team member I want to take action on AWS CloudWatch alarms so that we respond responsibly

Sprint 2: Integrate alerting into security workflow & ticketing



Use Cases & Examples

Building a “Lambda Responder”



Reading events in Lambda


```
exports.handler = function(event,context) {

   var snsMsgString = JSON.stringify(event.Records[0].Sns.Message);

   var snsMsgObject = getSNSMessageObject(snsMsgString);

   var srcBucket = snsMsgObject.Records[0].s3.bucket.name;

   var srcKey = snsMsgObject.Records[0].s3.object.key;

   ...


function getSNSMessageObject(msgString) {

   var x = msgString.replace(/\\/g,'');

   var y = x.substring(1,x.length-1);

   var z = JSON.parse(y);

      return z;
```

Detecting events in Lambda

```
…
var EVENT_SOURCE_TO_TRACK = /cloudtrail.amazonaws.com/;
var EVENT_NAME_TO_TRACK   = /StopLogging/;


var matchingRecords = records
    .Records
    .filter(function(record) {
        return record.eventSource.match(EVENT_SOURCE_TO_TRACK)
            && record.eventName.match(EVENT_NAME_TO_TRACK);
    });
…
```

Responding to events in Lambda


```
...
if (matchingRecords.length >= 1) {
    console.log('StopLogging detected! Reverting...');
    cloudtrail.startLogging(cloudtrailParams, function(err, data) {
        ...
```

Responding to events in Lambda

▶	2015-09-23, 05:20:50 PM	awslambda_944_20150923...	StartLogging	Trail
▶	2015-09-23, 05:17:49 PM	reinvent-sec308	StopLogging	Trail

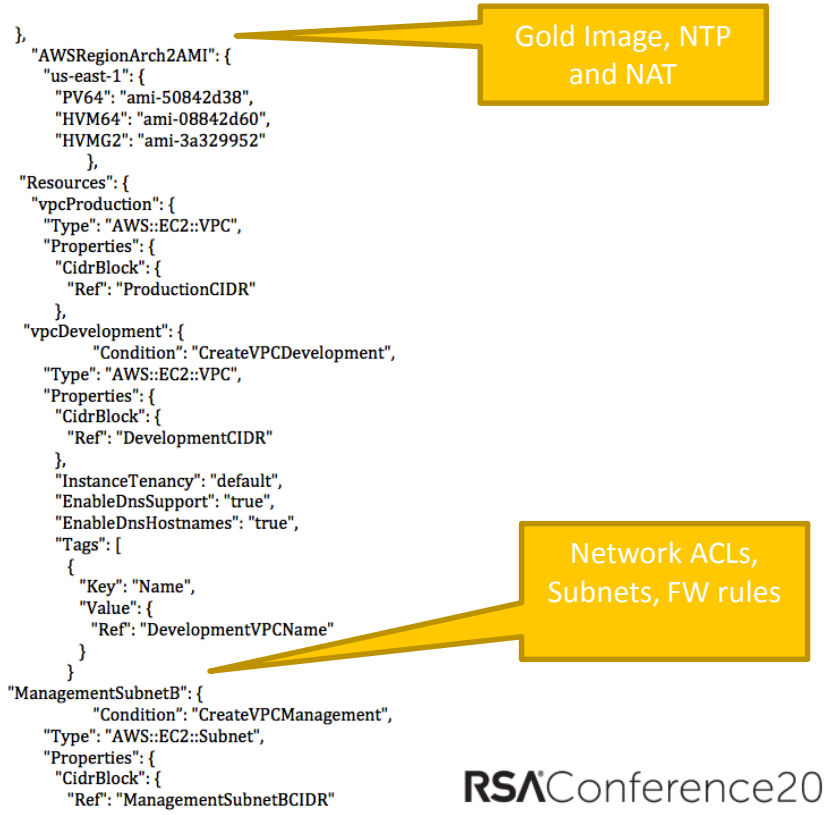
Golden Code: Security Translation to AWS

What you do in any IT Environment

- Firewall rules
- Network ACLs
- Network time pointers
- Internal and external subnets
- NAT rules
- Gold OS images
- Encryption algorithms for data in transit and at rest

http://docs.aws.amazon.com/quickstart/latest/accelerator-nist/welcome.html

AWS JSON translation



amazon web services | Professional Services

Security As Code: Using AWS CodeDeploy

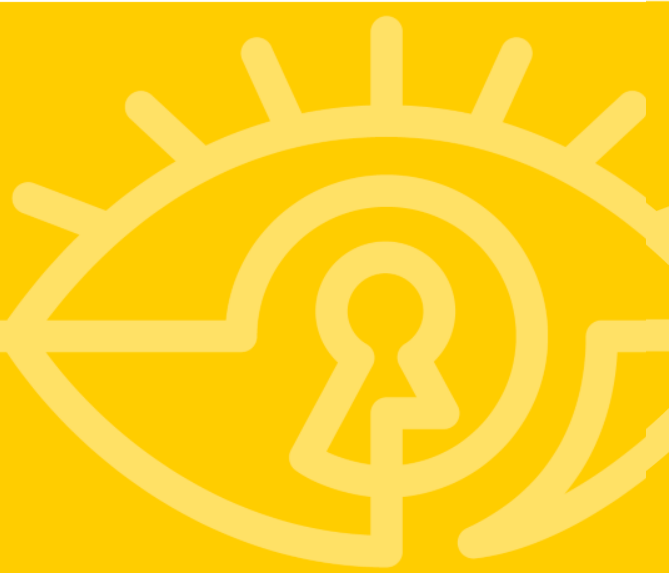
Imaging instance memory:

LiME - https://github.com/504ensicslabs/lime

AWS CodeDeploy:

```
a45e60bce0a3 responder $ ls
LIME            appspec.yml     scripts
a45e60bce0a3 responder $ cat appspec.yml
version: 0.0
os: linux
files:
  - source: LIME
    destination: /root/LIME
#   section may cause associated deployments to fail.
hooks:
 AfterInstall:
  - location: scripts/doit.sh
    timeout: 360


a45e60bce0a3 responder $ aws deploy push --application-name WebFinanceFE --descr
iption "Forensic tooling" --ignore-hidden-files --s3-location s3://deploysource/
limeDone.zip --source lime
```



Now What?

“Apply” Slide

- Make DevOps the security team’s job
- Harden your toolchain
- Plan your Security Epics
- Write your first Security User Story
- Sprint!