Integrating Security Testing Into Your Container Build Pipeline

Workshop Prerequisites

https://bit.ly/2M7BzCd

OR

https://container-devsecops.awssecworkshops.com/

Use "us-east-2"

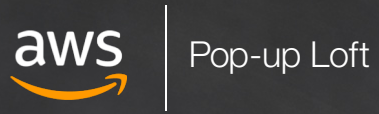
Directions (20 min):

- **Module 0: Environment Setup**
  - **Deploy the anchore service CFT**
  - **Deploy the pipeline CFT**

Learning SURVEY

https://bit.ly/2X70BlJ

Integrating Security Testing Into Your Container Build Pipeline



Aditya Patel
Security Consultant

Avik Mukherjee
Security Architect

Apoorva Kulkarni
DevOps Consultant

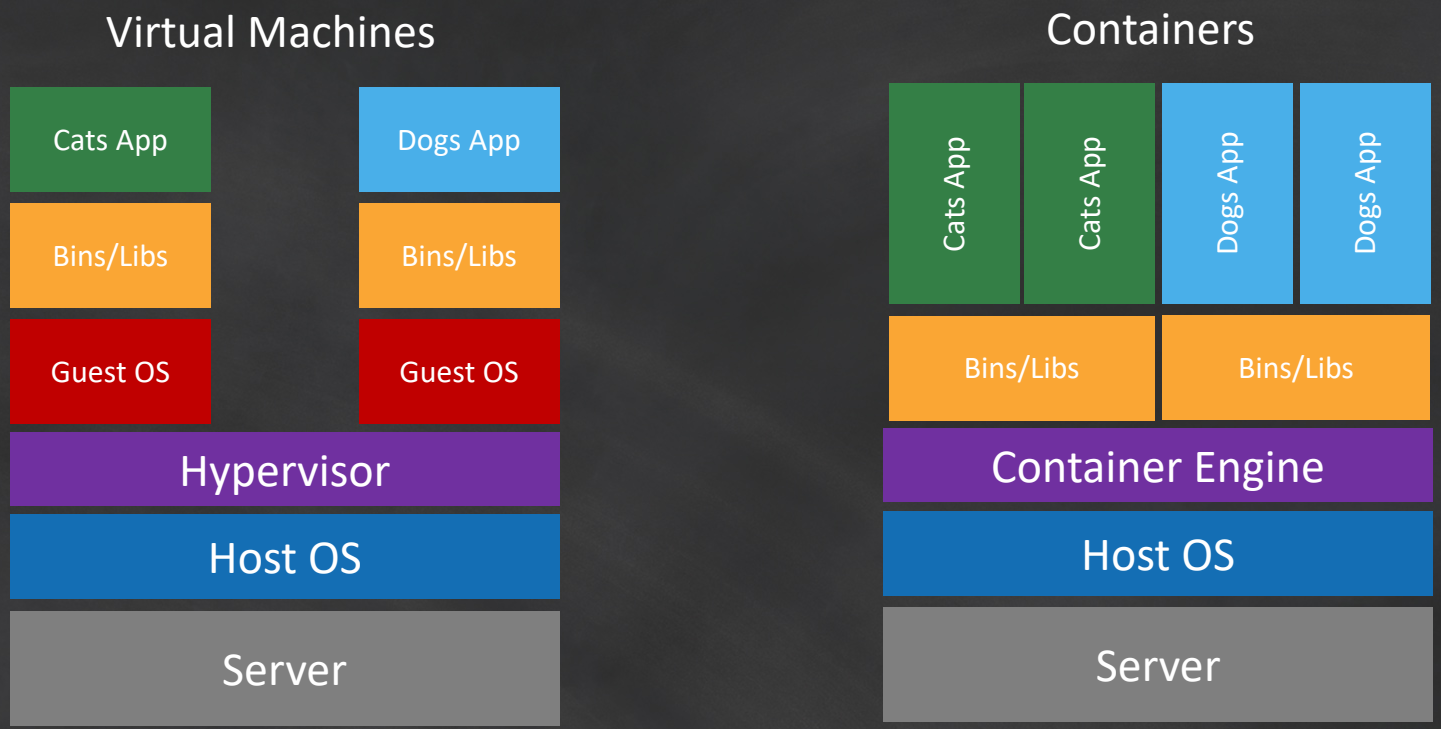
Jesse Fuchs
Sr Solutions Architect

Goals

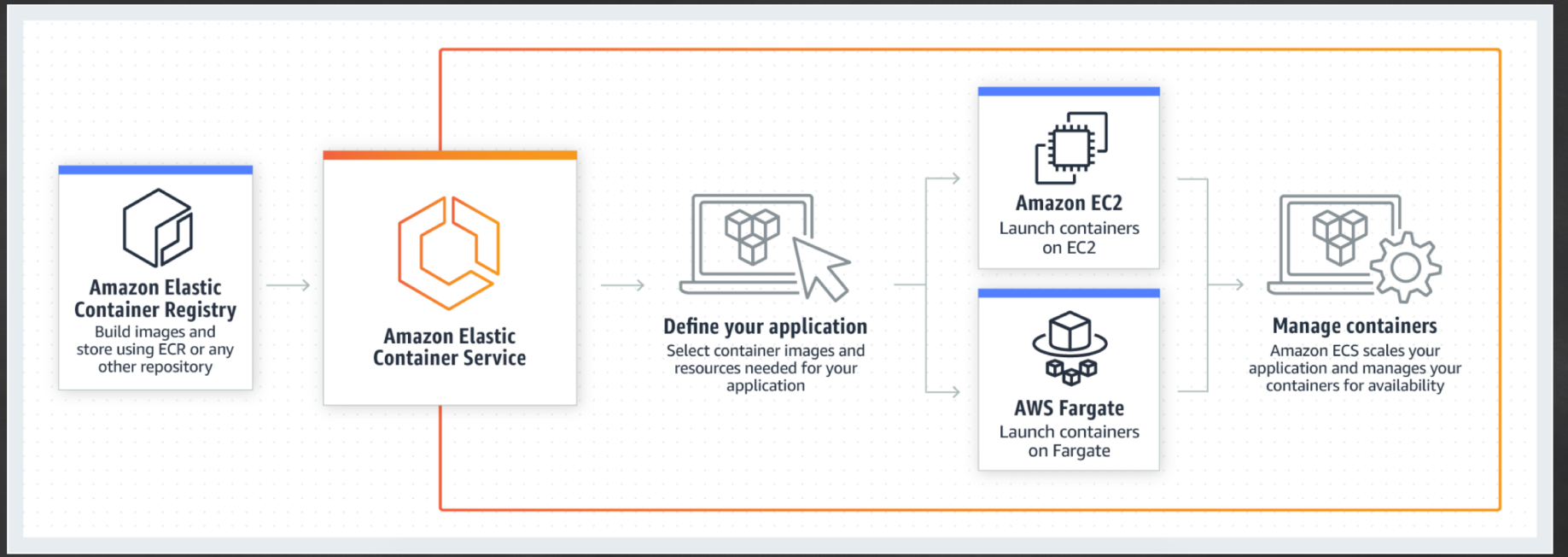
- Learn about Containers using DevOps on AWS
- Learn about container shared responsibility model
- Learn how to approach DevSecOps for Containers
- Learn about Container Security threats
- Learn about container image security
- Learn about AWS Development Tools and DevOps services
- *Have fun while you're at it!*

aws

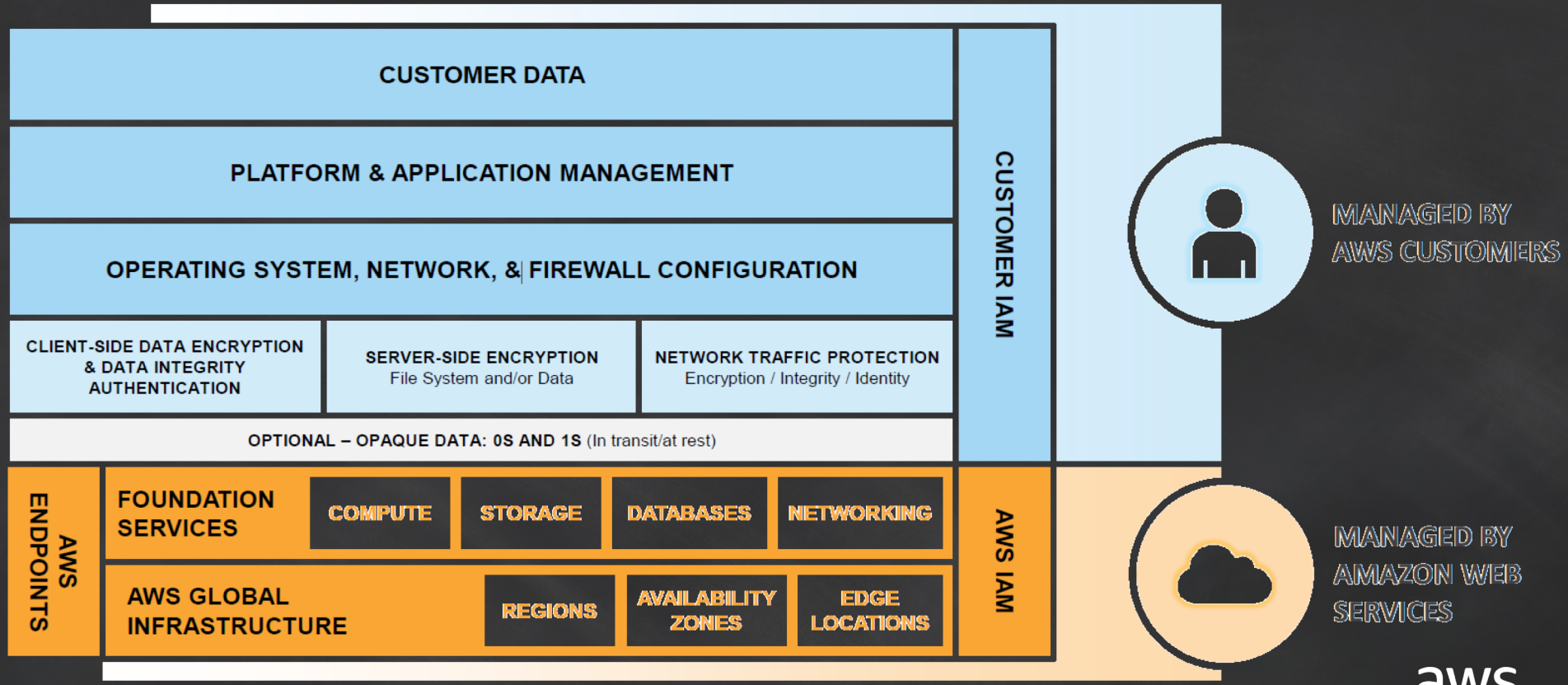
Why Container Security is Different?



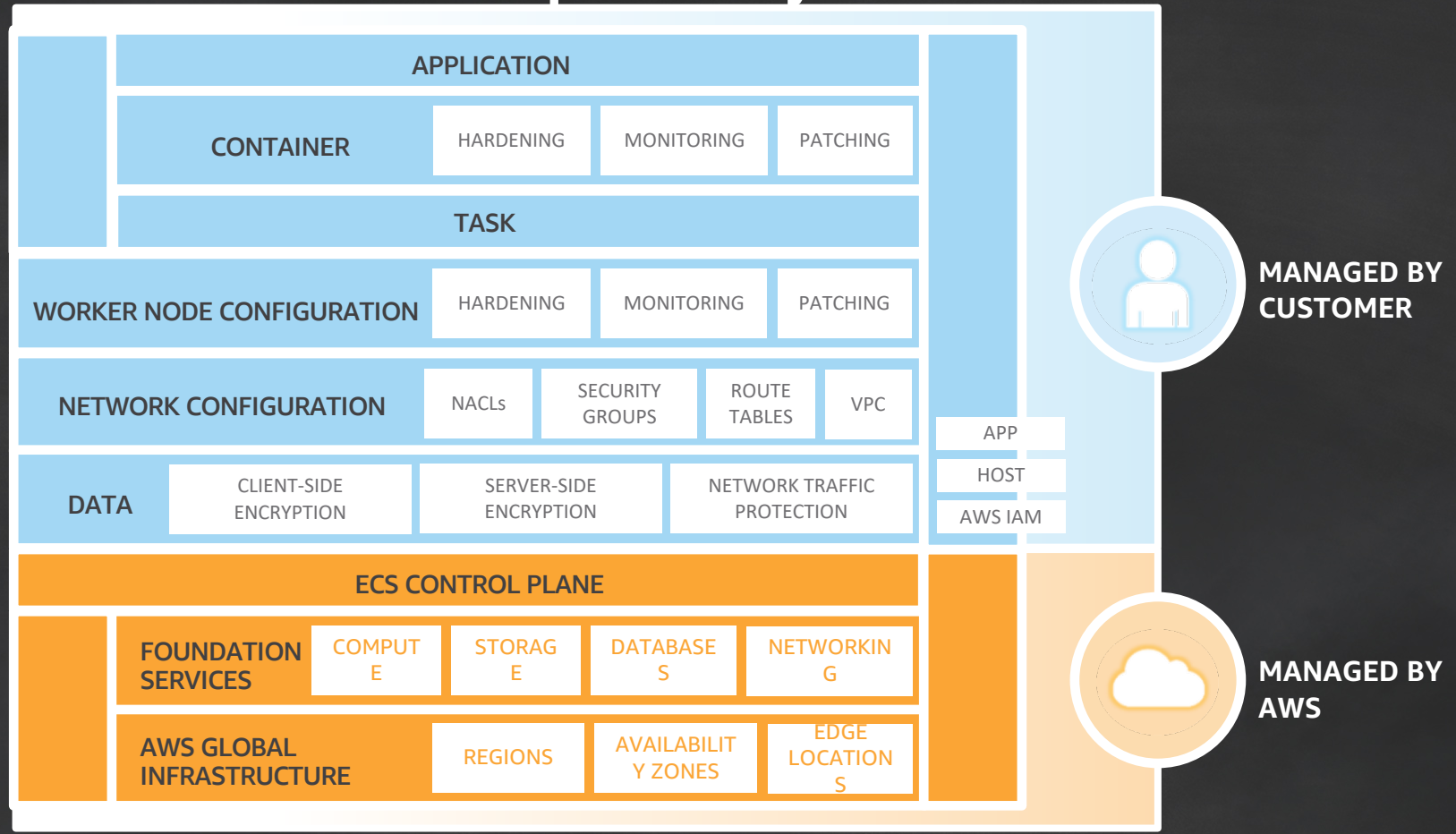
Containers @ AWS - Running workloads on Amazon ECS



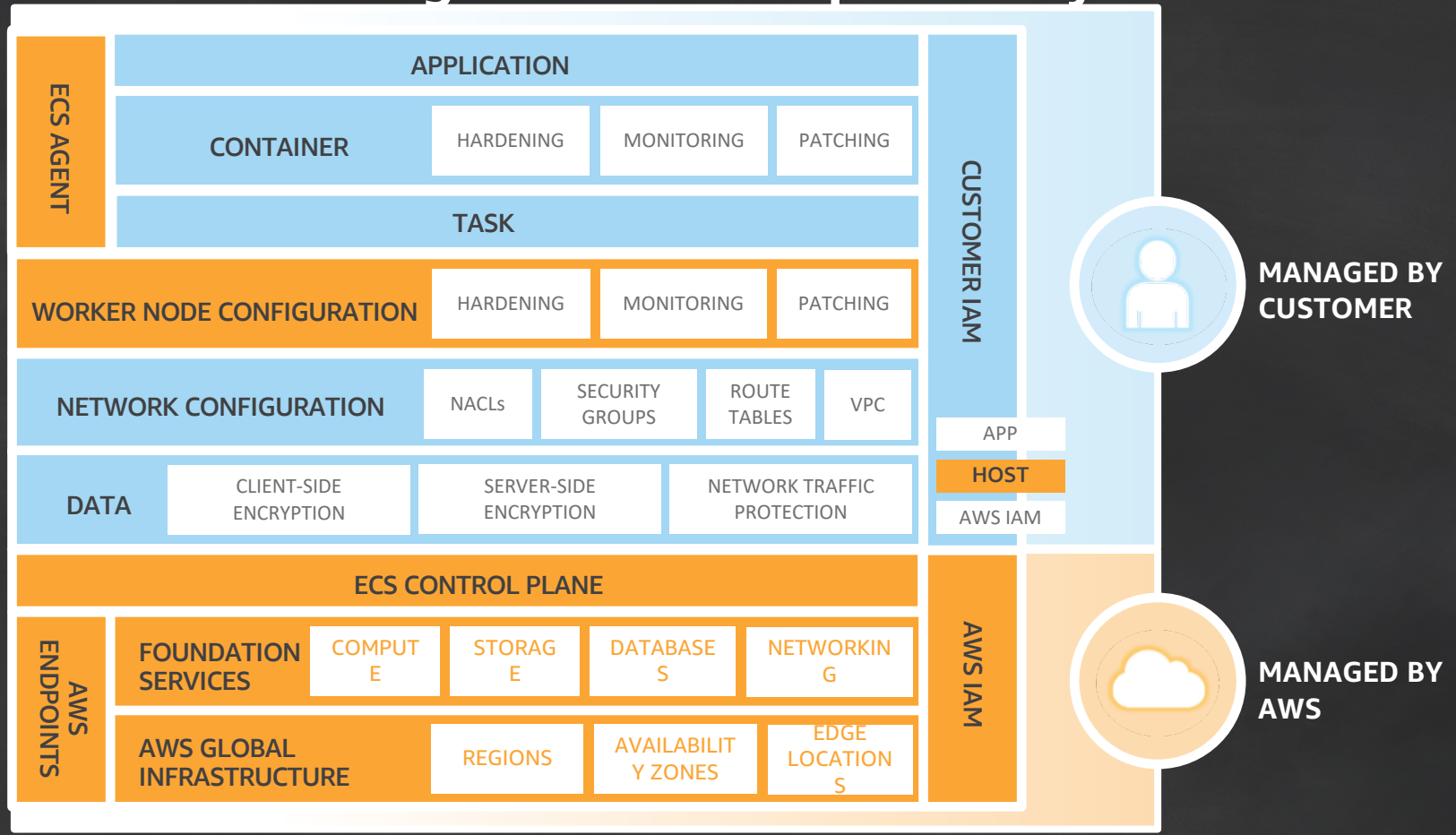
AWS Shared Responsibility Model



AWS ECS Shared Responsibility Model



AWS ECS with Fargate Shared Responsibility Model



Automated pipelines - DevSecOps

Speaking of Automation – you should automate everything including your:

- Code & Container Builds
- Your infrastructure via Infrastructure-as-Code patterns
- Your Deployments
- Making things Self-Healing
- **And Your Security**

Make it fast and easy for your team to do the right thing!

Container Security Threats

- Host Security
- Image Security
- Denial of Service
- Credentials and Secrets
- Container Breakouts
- Runtime Security

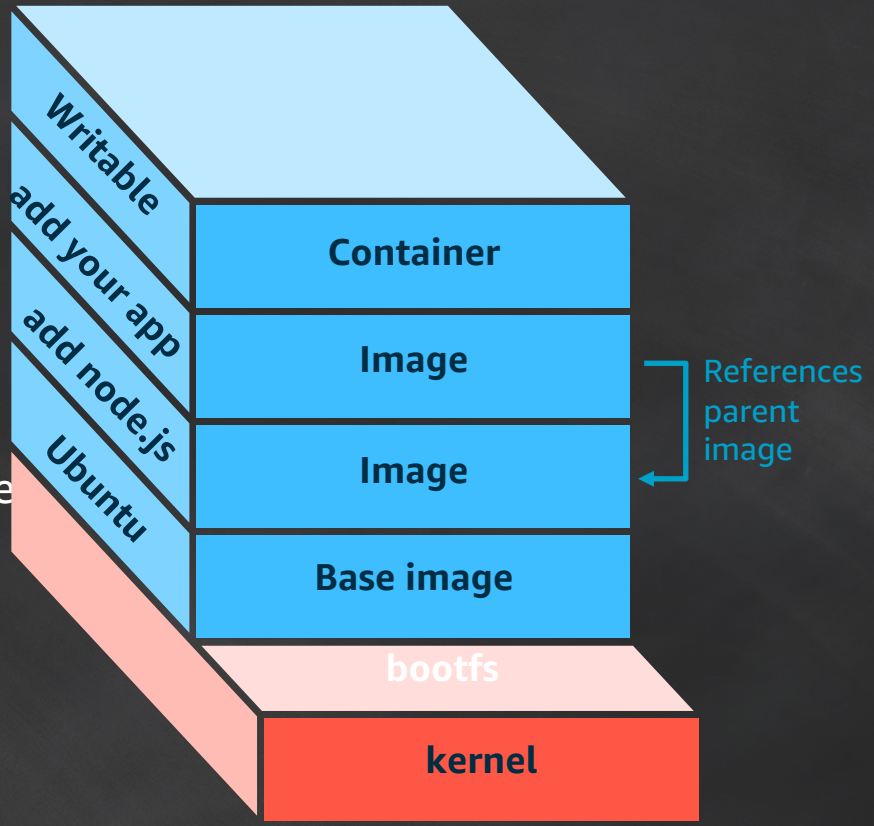
Container Security Threats

- Host Security
- Image Security
- Denial of Service
- Credentials and Secrets
- Container Breakouts
- Runtime Security

Security Best Practices for Container Images

Less is more (secure)

- No secrets in them
- One service per container
  - Use sidecars within Task / Pod
- Minimise container footprint
  - Include only what is **needed** at runtime



Security Best Practices for Container Images

- Use known and trusted base images
  - Official on Docker Hub
  - Read the Dockerfiles
  - Scan the image for CVEs
- Specify USER in Dockerfile (otherwise it's root)
- Unique and informative image tags
  - Be able to tell which commit at a glance

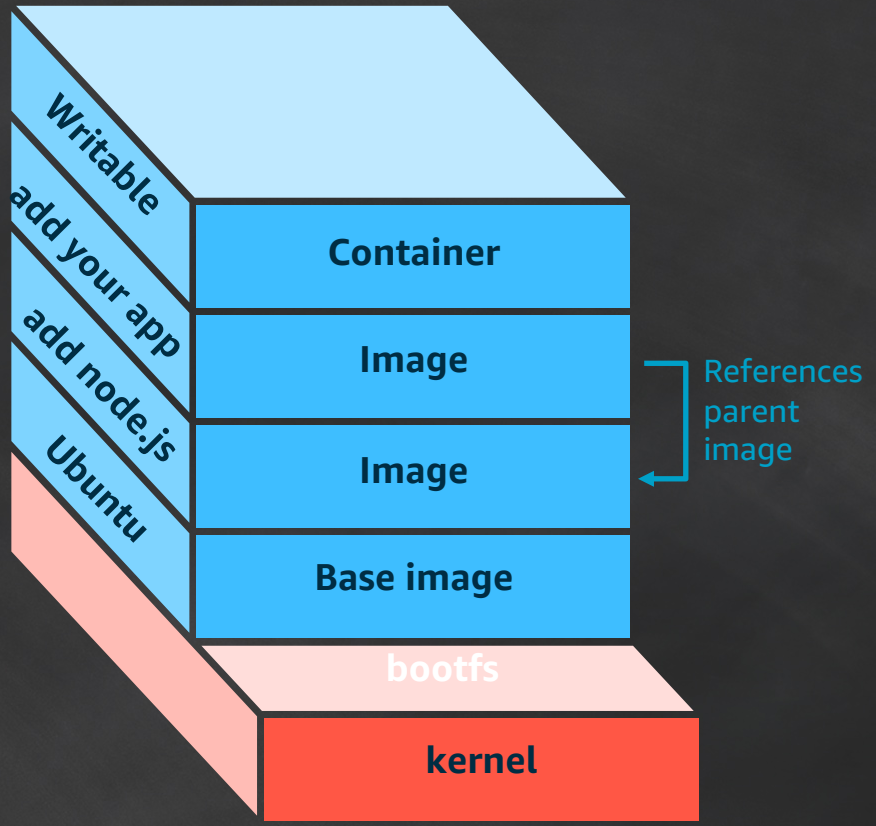


Image Security

- Docker Linting: Validation of docker configuration as per defined standards (PCI DSS v3.2.1 Req 2.2)
  - Hadolint
  - linterfordockerfile - https://access.redhat.com/labs/linterfordockerfile/
  - dockerfile_lint - https://github.com/projectatomic/dockerfile_lint

- Secrets scanning: Scanning repositories for secrets (sensitive information)
- (PCI DSS v3.2.1 Req 6.3.1)
  - Trufflehog
  - Git-secrets - https://github.com/awslabs/git-secrets/blob/master/README.rst

- Image scan: Vulnerability scanning of images in your build pipeline (PCI DSS v3.2.1 Req 6.1)
  - Anchore
  - Clair - https://github.com/coreos/clair

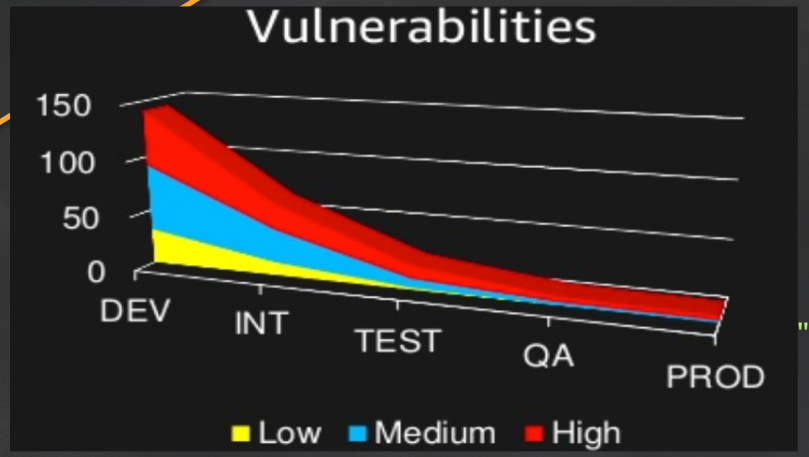
DevSecOps container pipeline

Developers Security Engineers Ops Engineers

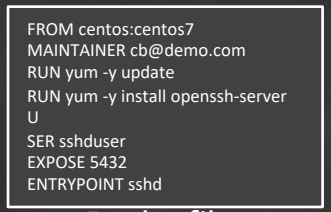
Amazon EC2 Container Registry

Task Definition

AWS CodeCommit



Docker image



Dockerfile

AWS CodeBuild

Validate Configuration > Merge >
Scan for secrets> Merge >

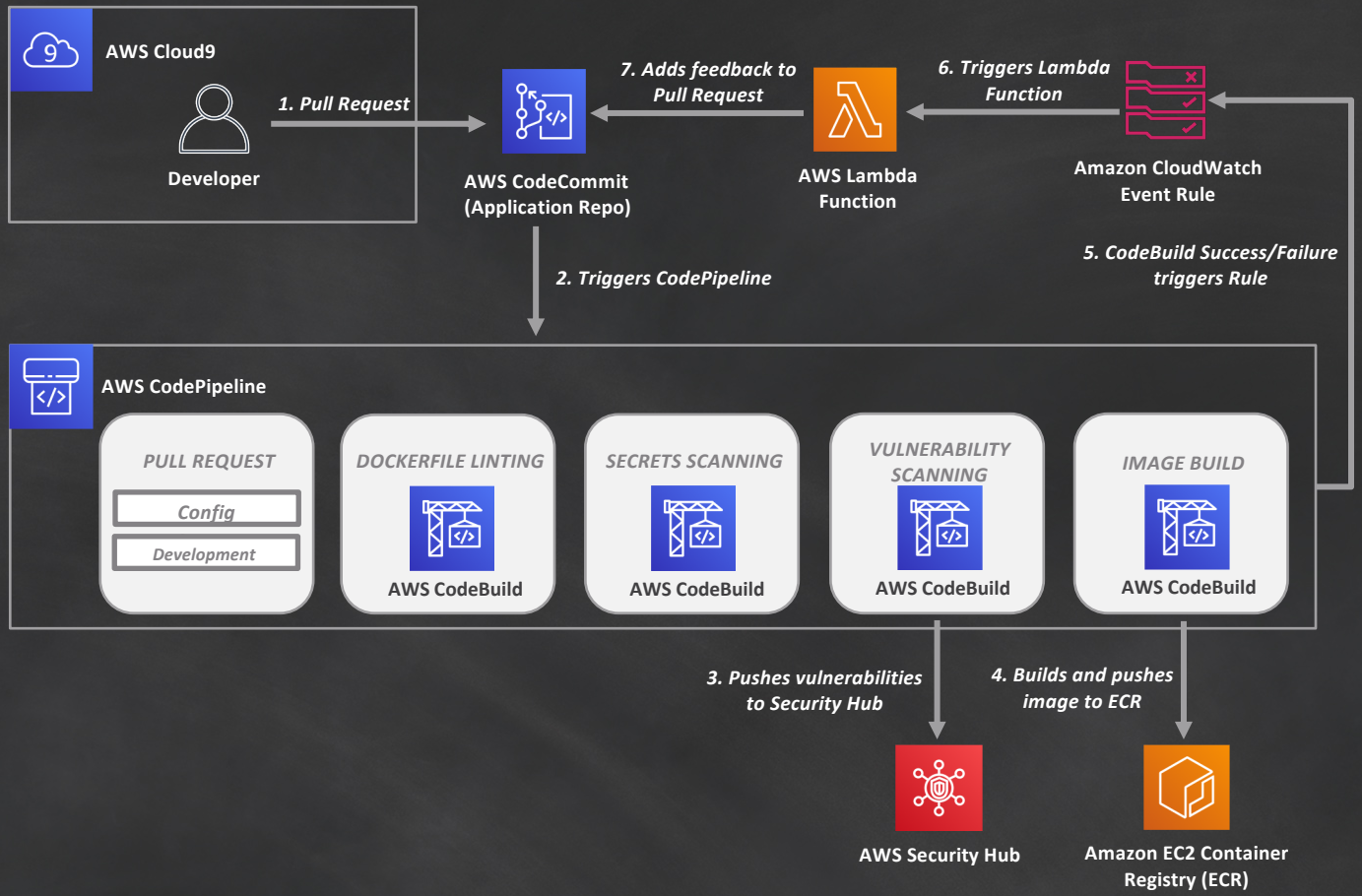
Scan Docker Image > Publish >

Credentials and Secrets

AWS has Parameter Store and Secrets Manager to store your Secrets. They are integrated into ECS but you'll need to call them within the Pod on Kubernetes via our CLI or SDK.

Assigning an IAM role to an Instance/Task/Function means the right AWS access key and secret to call the AWS CLI/SDK are transparently obtained and rotated.

Workshop Architecture – From 10,000 feet



Let’s build….and have fun…

Integrating Security Testing Into Your Container Build Pipeline

https://bit.ly/2M7BzCd

OR

https://container-devsecops.awssecworkshops.com/

Directions (15 min):

- **Module 1: Dockerfile Linting**
  - **Create buildspec file**
  - **Add Hadolint configuration**
- Module 2: Secrets Scanning
- Module 3: Vulnerability Scanning
- Module 4: Pipeline Testing

Use "us-east-2"

Integrating Security Testing Into Your Container Build Pipeline

https://bit.ly/2M7BzCd

OR

https://container-devsecops.awssecworkshops.com/

Directions (15 min):

- Module 1: Dockerfile Linting
- **Module 2: Secrets Scanning**
  - **Create buildspec file**
  - **Add trufflehog regex configuration**
- Module 3: Vulnerability Scanning
- Module 4: Pipeline Testing

Use "us-east-2"

Integrating Security Testing Into Your Container Build Pipeline

https://bit.ly/2M7BzCd

OR

https://container-devsecops.awssecworkshops.com/

Directions (15 min):

- Module 1: Dockerfile Linting
- Module 2: Secrets Scanning
- **Module 3: Vulnerability Scanning**
  - **Create buildspec file**
    - **Add command to deploy and run anchore**
- Module 4: Pipeline Testing

Use "us-east-2"

Integrating Security Testing Into Your Container Build Pipeline

https://bit.ly/2M7BzCd

OR

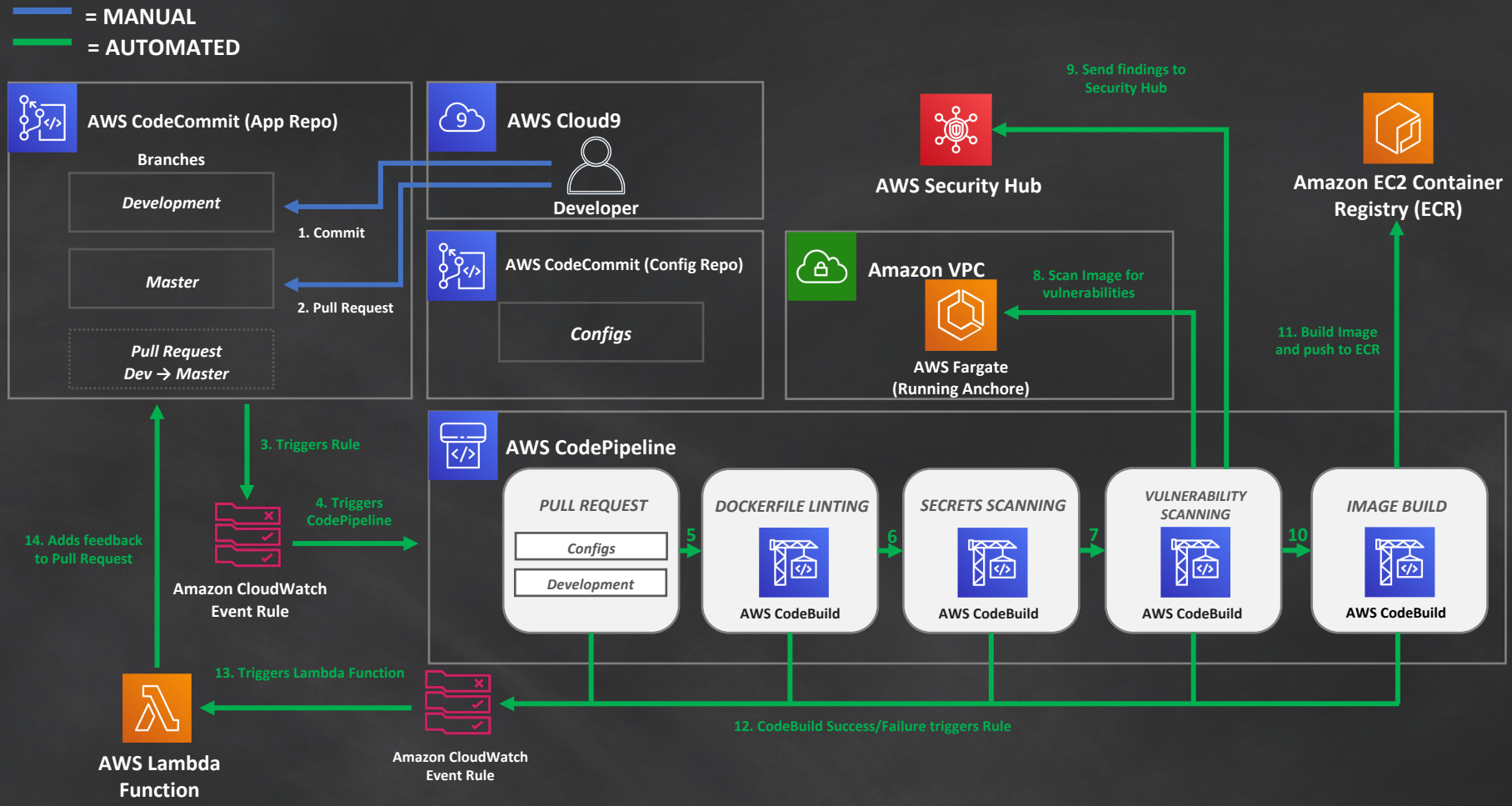
https://container-devsecops.awssecworkshops.com/

Directions (30 min):

- Module 1: Dockerfile Linting
- Module 2: Secrets Scanning
- Module 3: Vulnerability Scanning
- **Module 4: Pipeline Testing**
  - **Make a commit**
  - **Create pull request**
  - **View feedback loop**

Use
“us-east-2”

Let’s wrap up



Workshop Learning Survey – After

https://bit.ly/2YM8hdB

Thank you!