Integrating security testing into your container build pipeline

Laurent Domb
Principal Solutions Architect
AWS

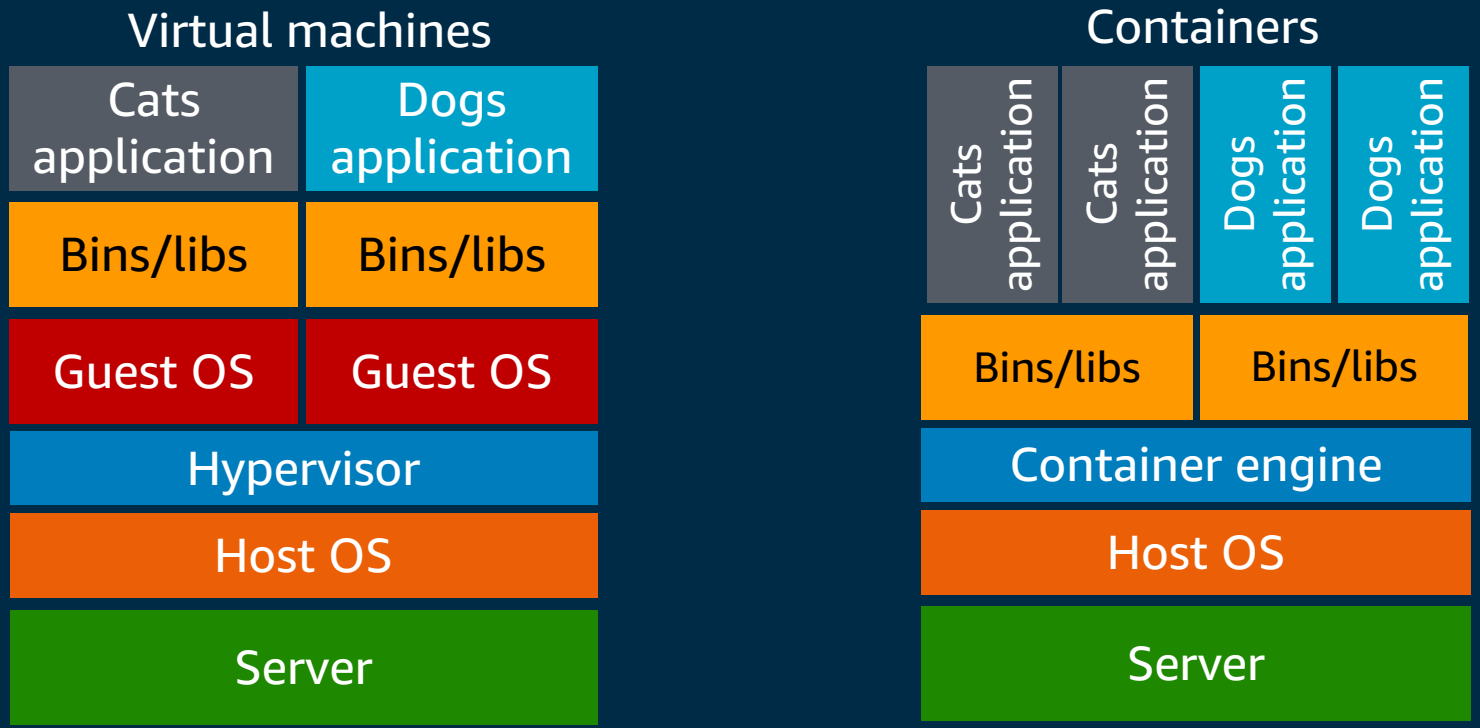
© 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Goals

- Learn about container security using DevSecOps
- Learn about open-source container security tools and standards
- Learn about AWS development tools and DevOps services

© 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved.

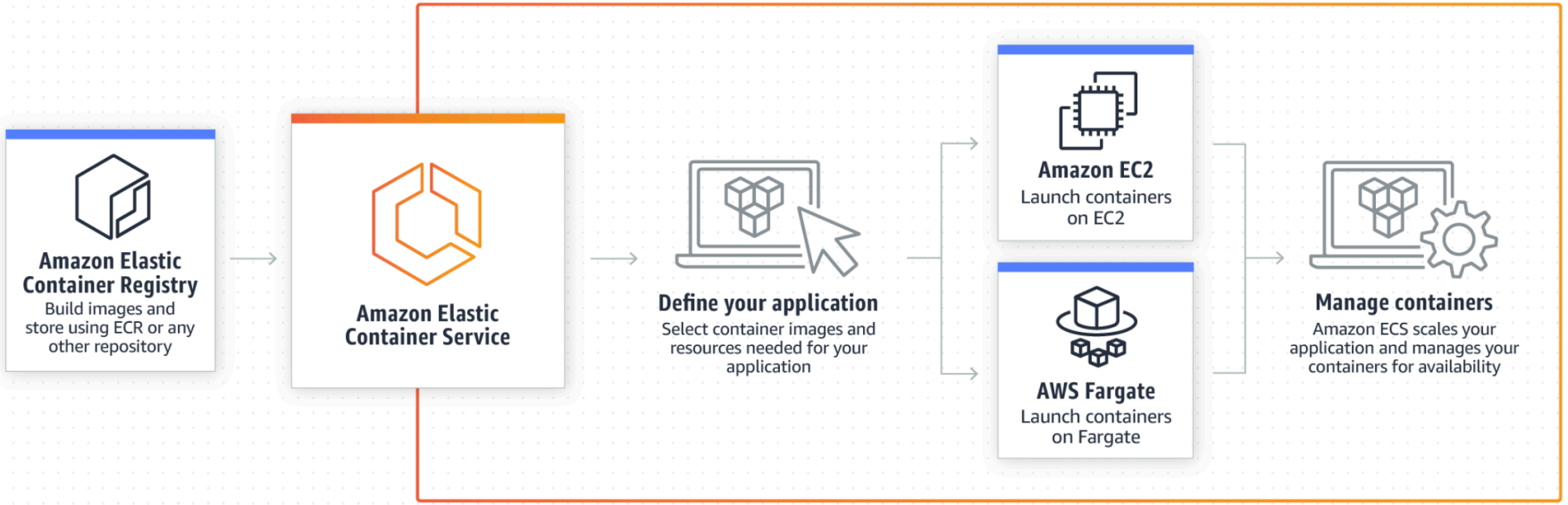
Why is container security different?



© 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved.

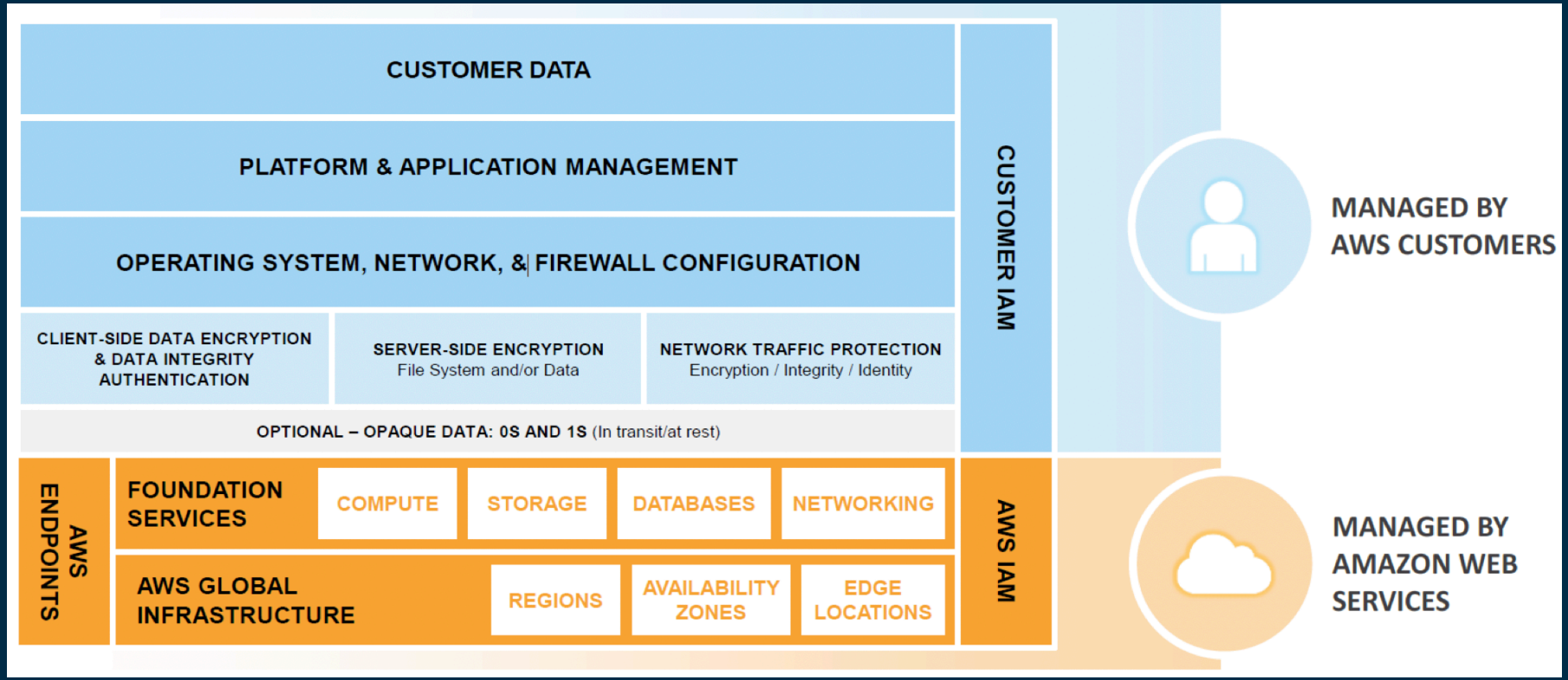
aws

Containers on AWS



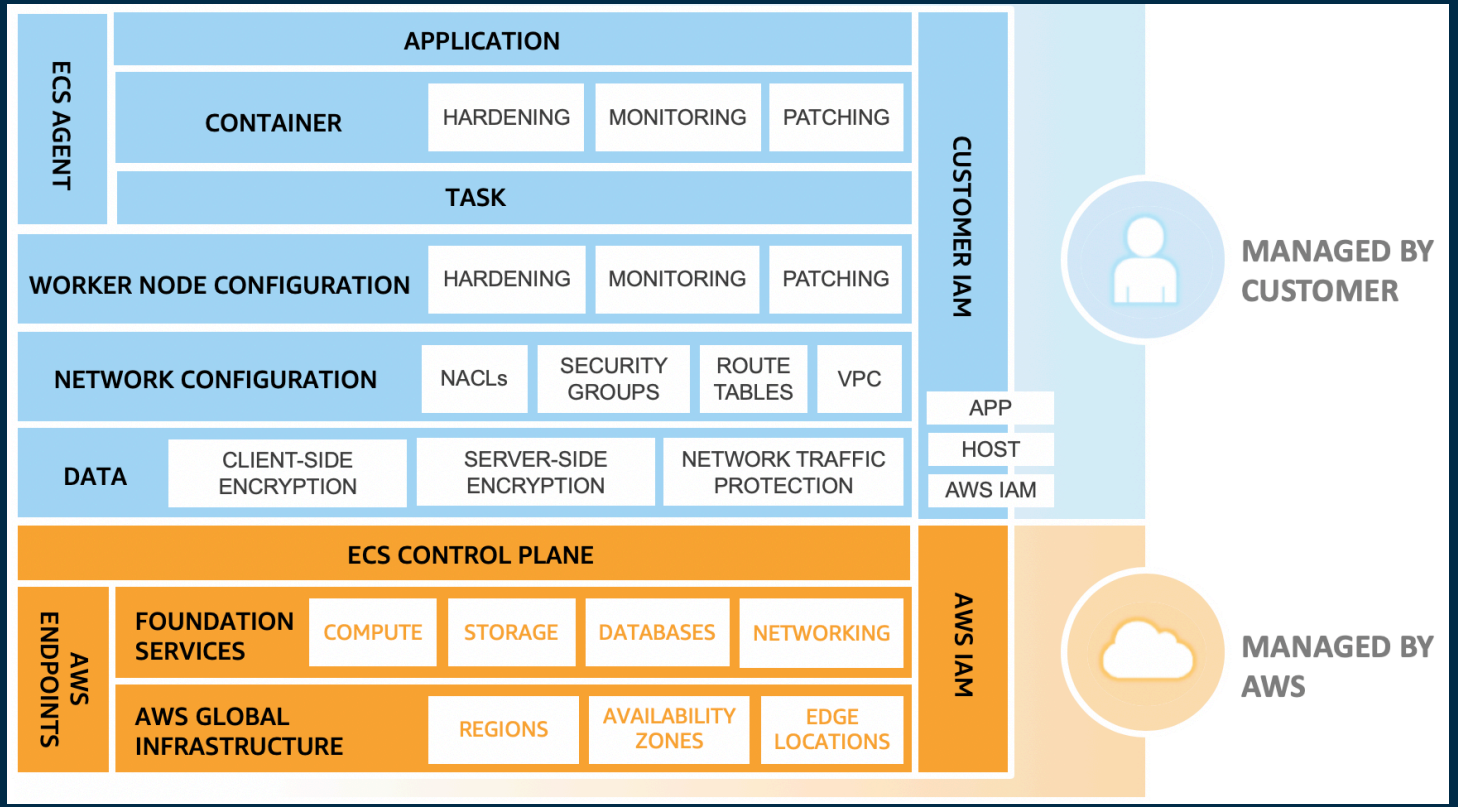
© 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved.

AWS shared responsibility model



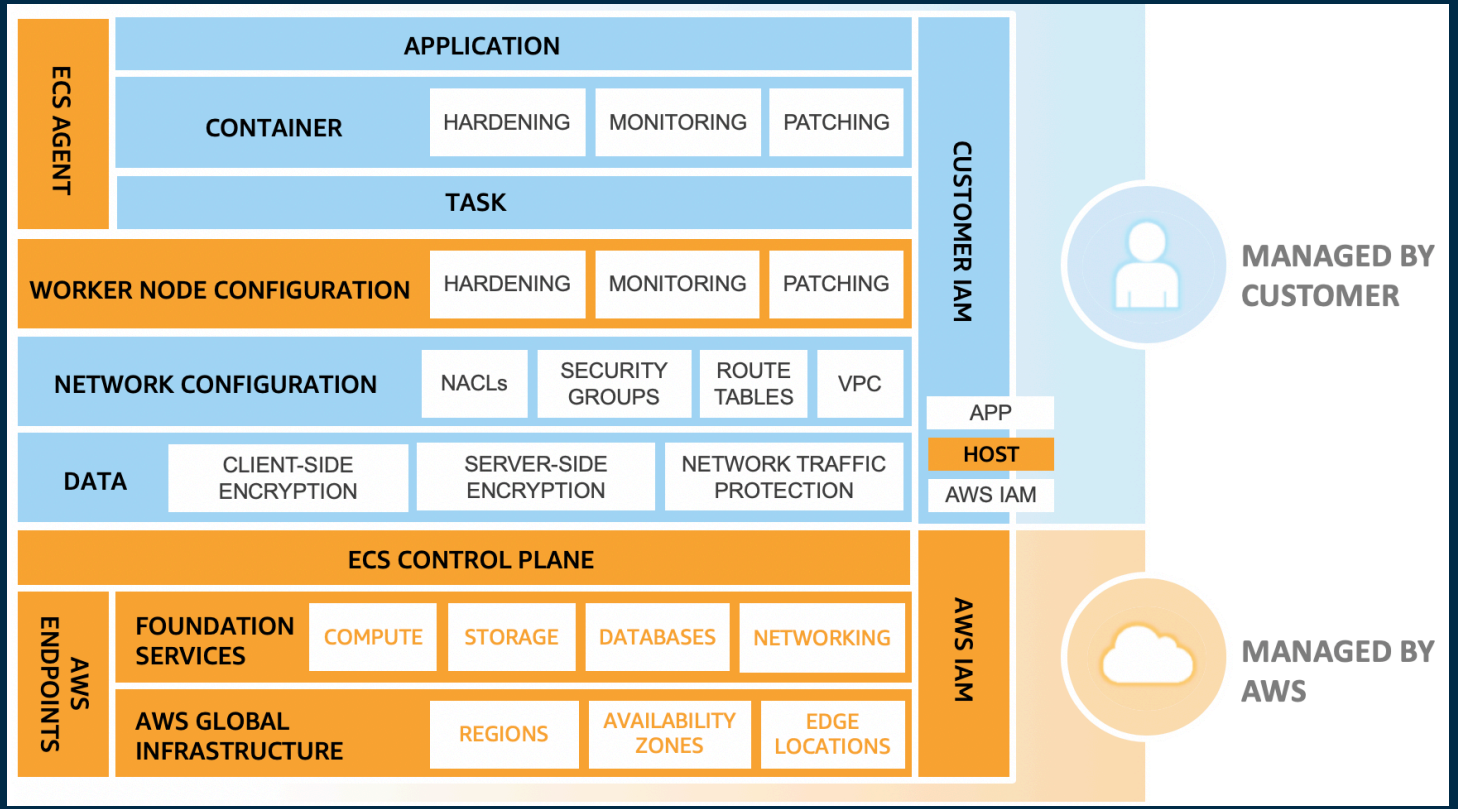
© 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Amazon ECS: AWS shared responsibility model



© 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved.

AWS Fargate: AWS shared responsibility model



© 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Automated pipelines: DevSecOps

Speaking of automation, you should automate everything, including

- Code and container builds
- Infrastructure via infrastructure as code patterns
- Deployments
- Process of making things self-healing
- Security!

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

© 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved.

aws

Container security threats

- Host security
- Image security
- Denial of service
- Credentials and secrets
- Container breakouts
- Runtime security

© 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Container security threats

- **Host security**
- **Image security**
- Denial of service
- **Credentials and secrets**
- Container breakouts
- Runtime security

© 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved.

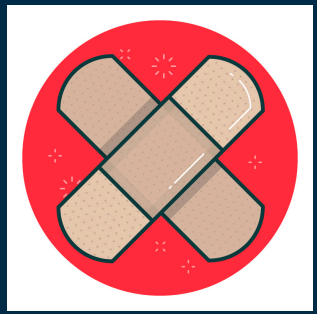
Container security threats

- **Host security**
- **Image security**
- Denial of service
- **Credentials and secrets**
- Container breakouts
- Runtime security

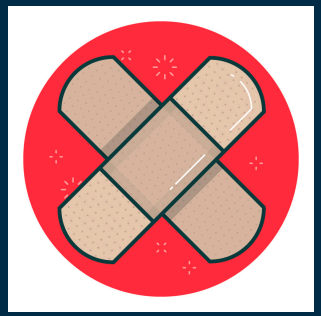
© 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Full blown OS or Container Optimized

ECS-optimized AMI



Debian/SuSe/RHEL



CoreOS

© 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved.

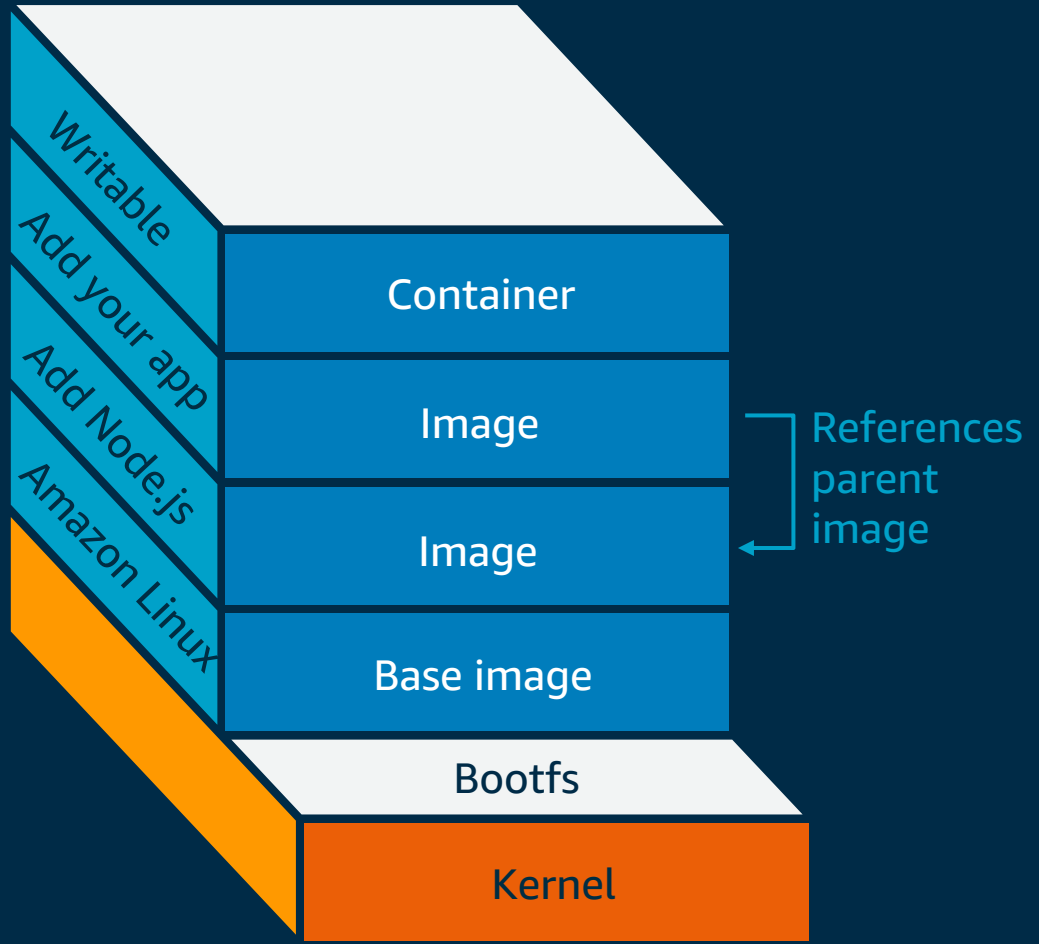
Container security threats

- **Host security**
- **Image security**
- Denial of service
- **Credentials and secrets**
- Container breakouts
- Runtime security

© 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Security best practices for container images

- **Less is more (secure)**
- No secrets in them
- One service per container
- Minimize container footprint
- Include only what is **needed** at runtime

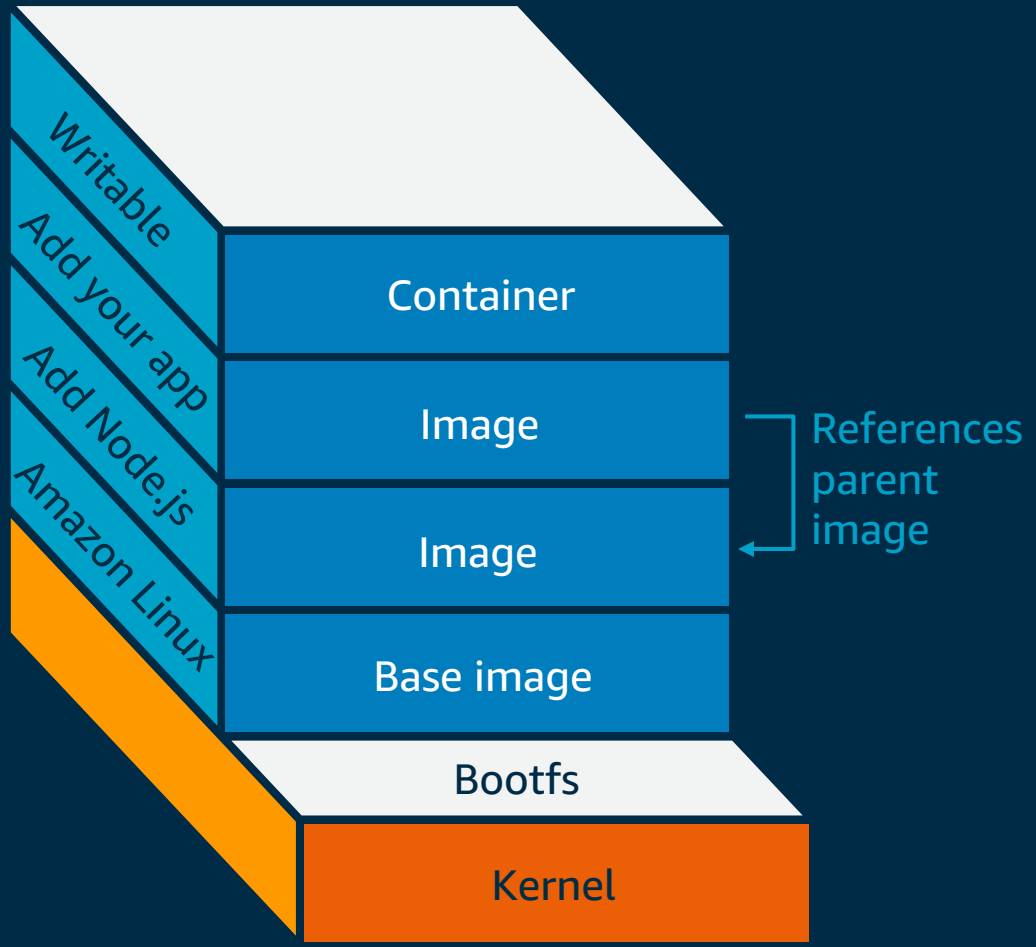


© 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved.

aws

Security best practices for container images

- Use known and trusted base images
- Scan the image for CVEs
- Specify USER in Dockerfile (otherwise it's a root)
- Use unique and informative image tags



© 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved.

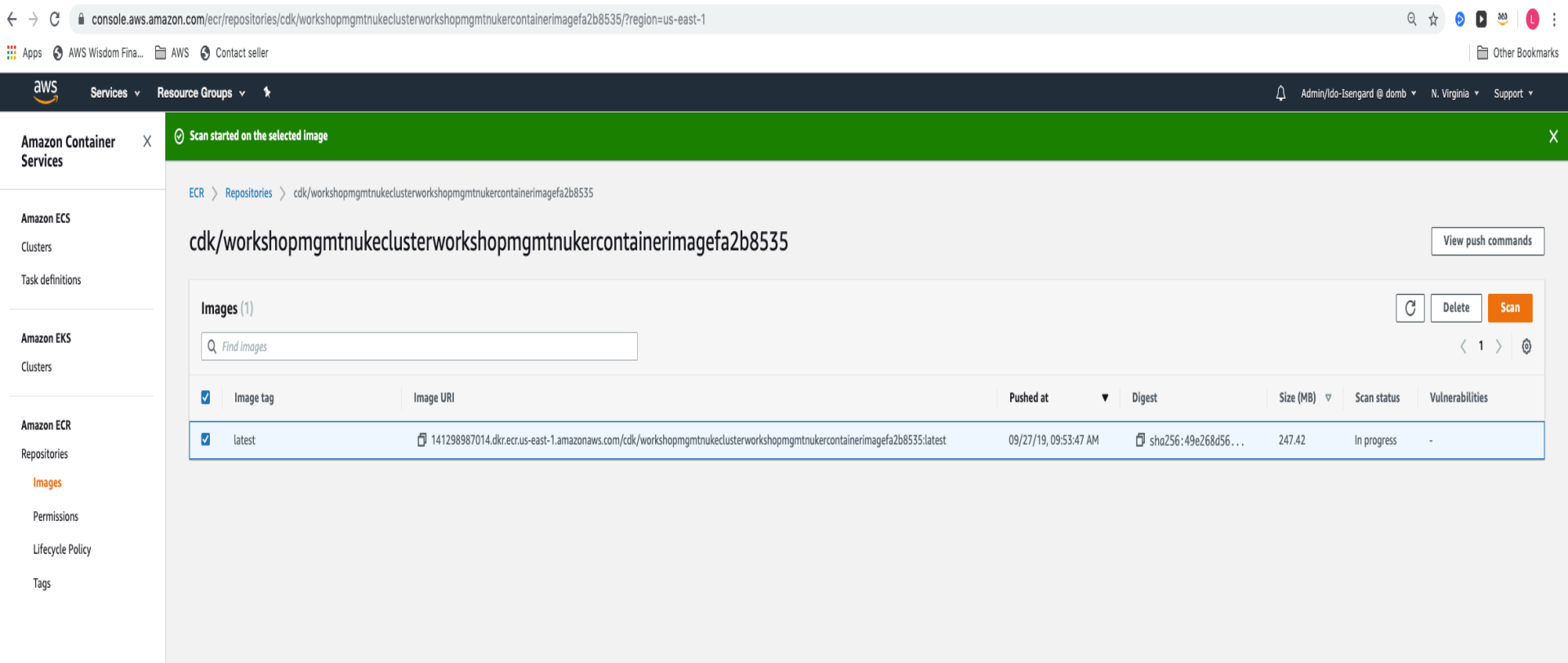
aws

Image security

- Docker linting: Validation of Docker configuration (PCI DSS v3.2.1 Req 2.2)
  - hadolint
  - dockerfile_lint
- Secrets scanning in images (PCI DSS v3.2.1 Req 6.3.1)
  - truffleHog
  - git-secrets
- Vulnerability scanning of images in your build pipeline (PCI DSS v3.2.1 Req 6.1)
  - Anchore Open-Source Engine
  - CoreOS Clair
  - Amazon ECR

© 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved.

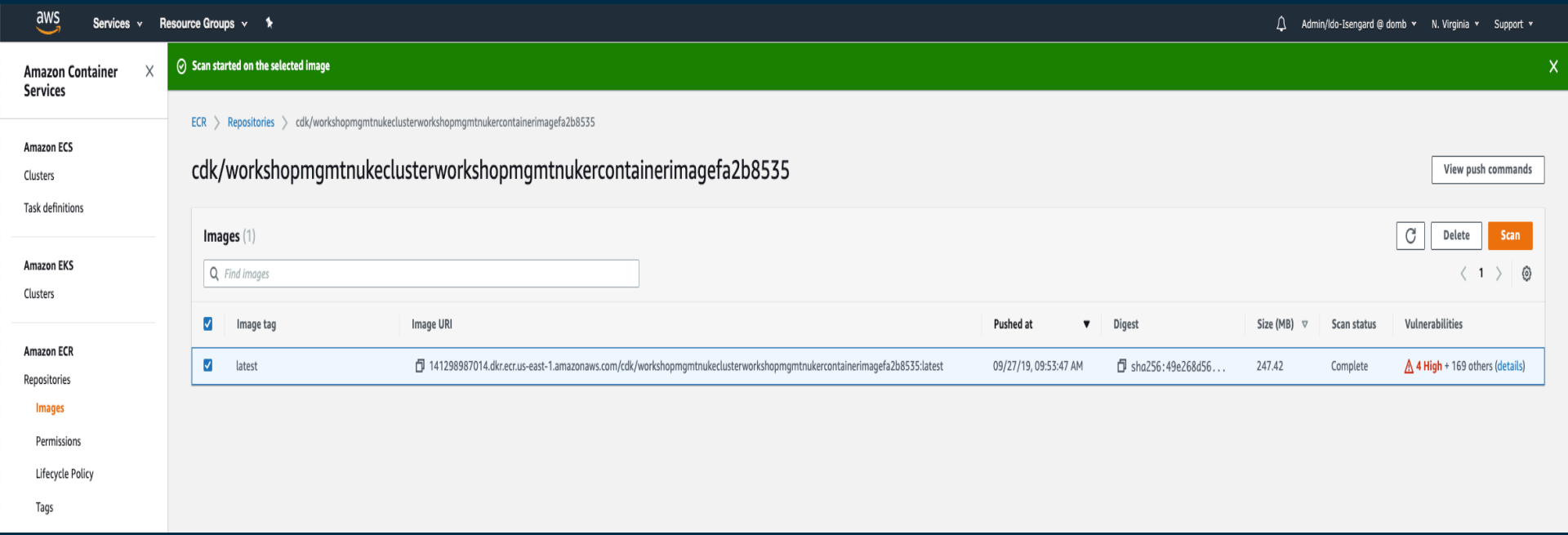
Image Scanning in AWS



© 2019, Amazon Web Services, Inc. or its Affiliates.

aws

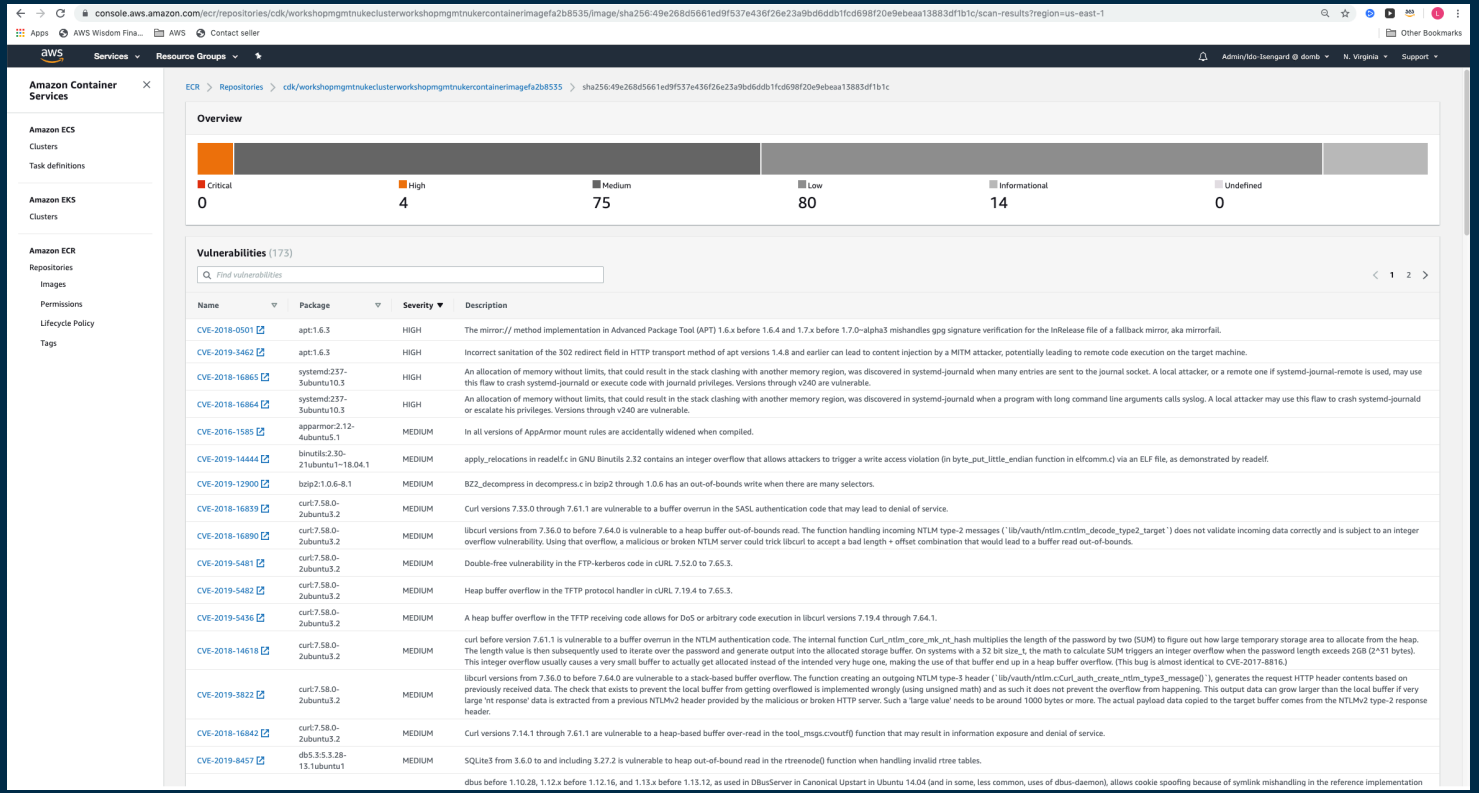
Image Scanning in AWS



© 2019, Amazon Web Services, Inc. or its Affiliates.

aws

Image Scanning in AWS

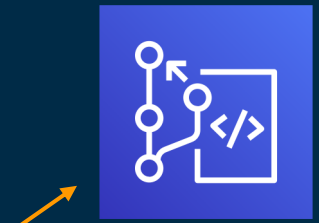


© 2019, Amazon Web Services, Inc. or its Affiliates.

DevSecOps container pipeline

```
{
  "memory": 128,
  "portMappings": [
    {
      "hostPort": 443,
      "containerPort": 443,
      "protocol": "tcp"
    }
  ],
  "image": "nginx",
}
```

Task definition



AWS CodeCommit

```
FROM centos:centos7
MAINTAINER cb@demo.com
RUN yum -y update
RUN yum -y install openssh-
server
EXPOSE 5432
ENTRYPOINT sshd
```

Dockerfile

AWS CodeBuild

© 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved.

aws

DevSecOps container pipeline

Developers **Sec**urity engineers **Ops** engineers

AWS CodeCommit

```
{
  "memory": 128,
  "portMappings": [
    {
      "hostPort": 443,
      "containerPort": 443,
      "protocol": "tcp"
    }
  ],
  "image": "nginx",
}
```

Task definition

```
FROM centos:centos7
MAINTAINER cb@demo.com
RUN yum -y update
RUN yum -y install openssh-server
EXPOSE 5432
ENTRYPOINT sshd
```

Dockerfile

AWS CodeBuild

© 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved.

aws

DevSecOps container pipeline

Developers **Sec**urity engineers **Ops** engineers

AWS CodeCommit

```
➢ python   ./check_dockerfile.py
           ./examples/Dockerfile-demo
           |jq ".warnings.warnings[].message"
"yum clean all is not used"
"installing SSH in a container is not recommended"
"No 'USER' instruction"
```

```
{
  "memory": 128,
  "portMappings": [
    {
      "hostPort": 443,
      "containerPort": 443,
      "protocol": "tcp"
    }
  ],
  "image": "nginx",
}
```

Task definition

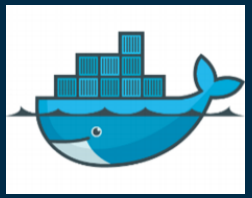
```
FROM centos:centos7
MAINTAINER cb@demo.com
RUN yum -y update
RUN yum -y install openssh-server
EXPOSE 5432
ENTRYPOINT sshd
```

Dockerfile

AWS CodeBuild

Validate configuration > Merge > Scan for secrets > Merge >

Docker image



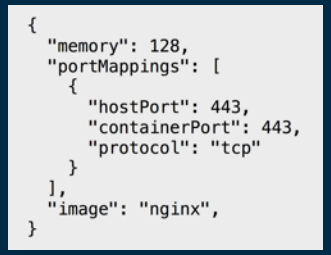
© 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved.

aws

DevSecOps container pipeline

Developers **Sec**urity engineers **Ops** engineers

Amazon EC2 container registry



Task definition

AWS CodeCommit

```
➢ python ./check_dockerfile.py
         ./examples/Dockerfile-demo
         |jq ".warnings.warnings[].message"
"yum clean all is not used"
"installing SSH in a container is not recommended"
"No 'USER' instruction"
```

Docker image

```
FROM centos:centos7
MAINTAINER cb@demo.com
RUN yum -y update
RUN yum -y install openssh-server
EXPOSE 5432
ENTRYPOINT sshd
```

Dockerfile

AWS CodeBuild

Validate configuration > Merge > Scan for secrets > Merge >

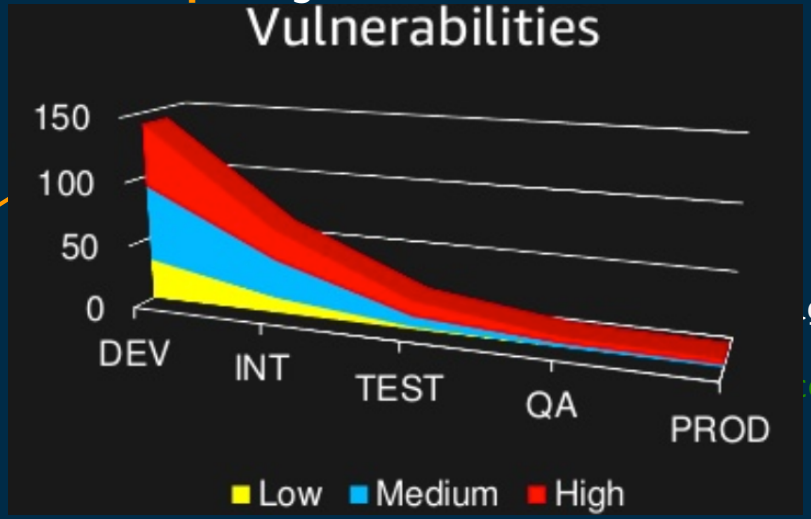
Scan Docker image > Publish >

© 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved.

aws

DevSecOps container pipeline

Developers **Sec**urity engineers **Ops** engineers



Amazon EC2 container registry

```
{
  "memory": 128,
  "portMappings": [
    {
      "hostPort": 443,
      "containerPort": 443,
      "protocol": "tcp"
    }
  ],
  "image": "nginx",
}
```

Task definition

AWS CodeCommit

Docker image

```
FROM centos:centos7
MAINTAINER cb@demo.com
RUN yum -y update
RUN yum -y install openssh-
server U
SER sshduser
EXPOSE 5432
ENTRYPOINT sshd
```

Dockerfile

AWS CodeBuild

Validate configuration > Merge > Scan for secrets > Merge >

Scan Docker image > Publish >

© 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Container security threats

- **Host security**
- **Image security**
- Denial of service
- **Credentials and secrets**
- Container breakouts
- Runtime security

© 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Credentials and secrets

AWS has Parameter Store and AWS Secrets Manager to store your secrets

They are integrated into Amazon ECS, but you need to call them within the pod on Kubernetes via AWS CLI or SDK

Assigning an IAM role to an instance, task, or function means that the right AWS access key and secret to call the AWS CLI or SDK are transparently obtained and rotated

© 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Benefits of Using IAM Roles for Tasks

Credential Isolation: A container can only retrieve credentials for the IAM role that is defined in the task definition to which it belongs; a container never has access to credentials that are intended for another container that belongs to another task.

Authorization: Unauthorized containers cannot access IAM role credentials defined for other tasks.

Auditability: Access and event logging is available through CloudTrail to ensure retrospective auditing. Task credentials have a context of taskArn that is attached to the session, so CloudTrail logs show which task is using which role.

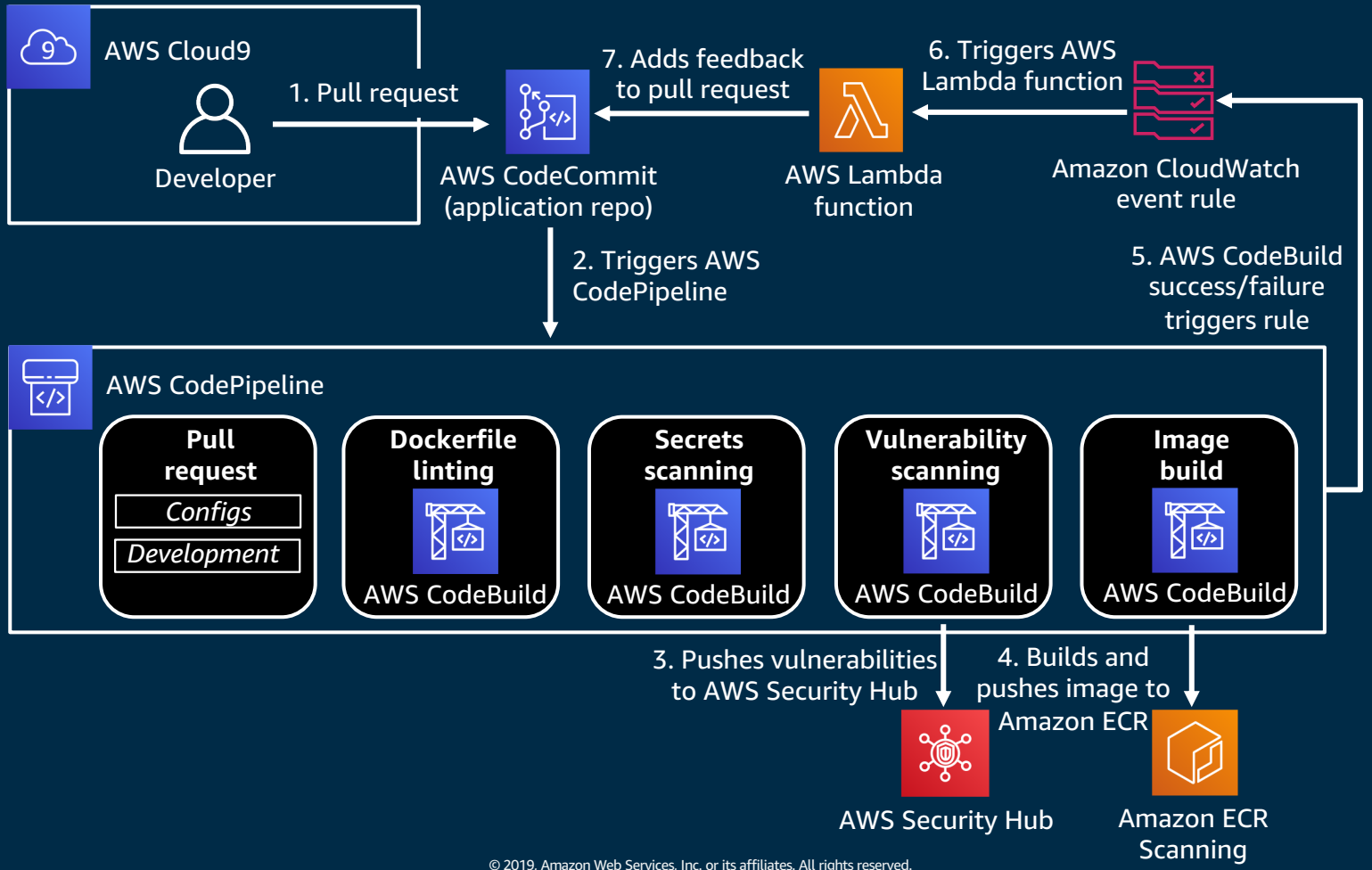
© 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved.

aws

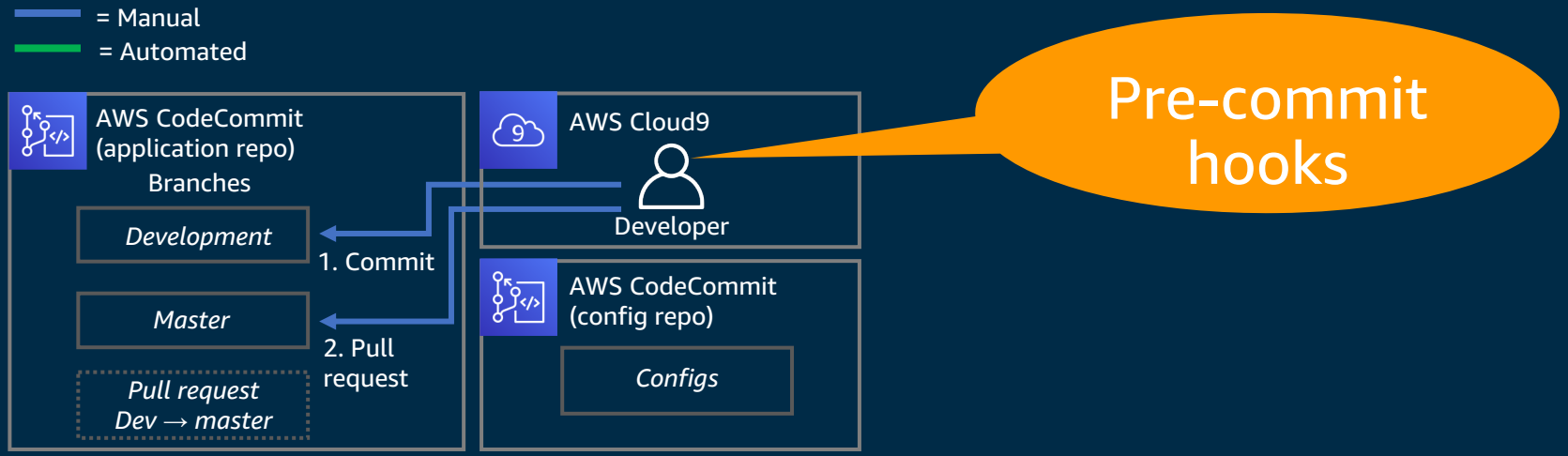
Pipeline high level overview

aws RE:INFORCE

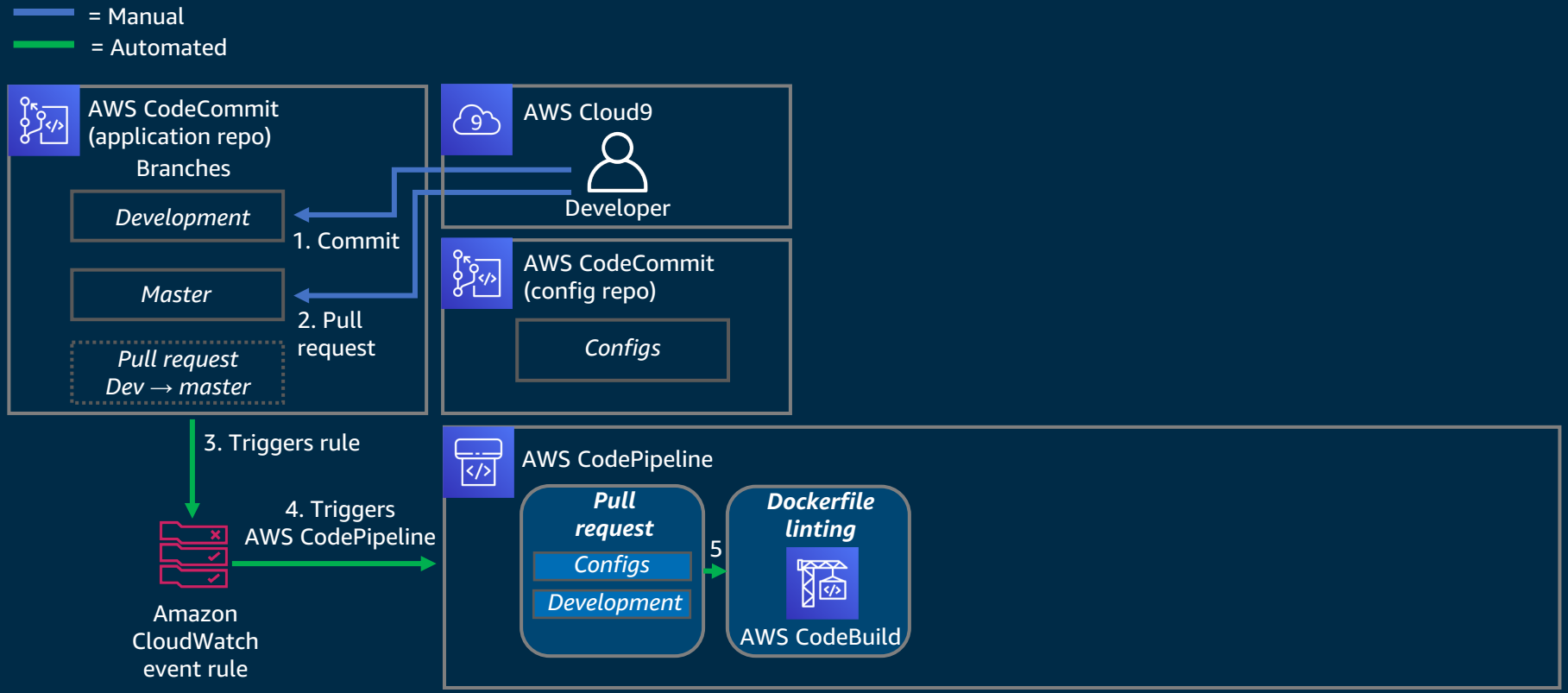
© 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved.



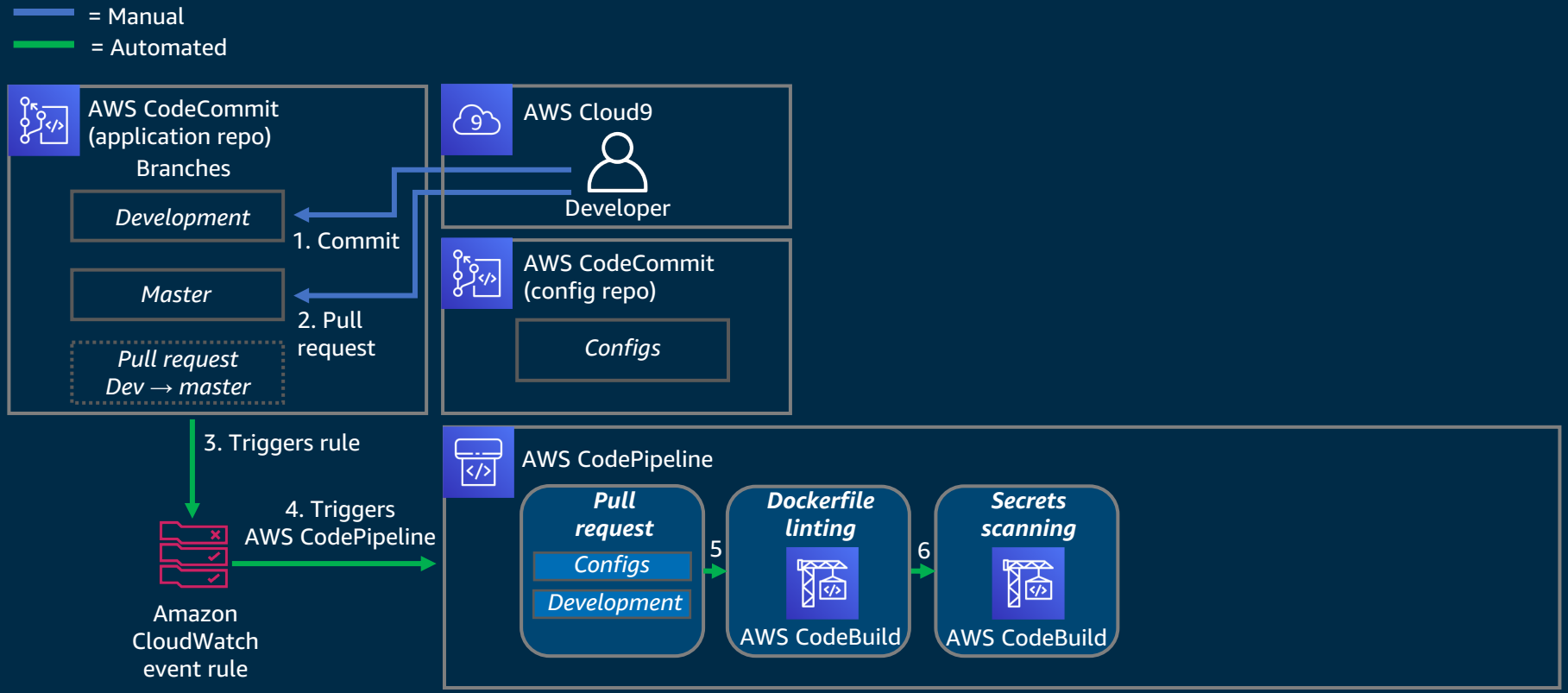
© 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved.



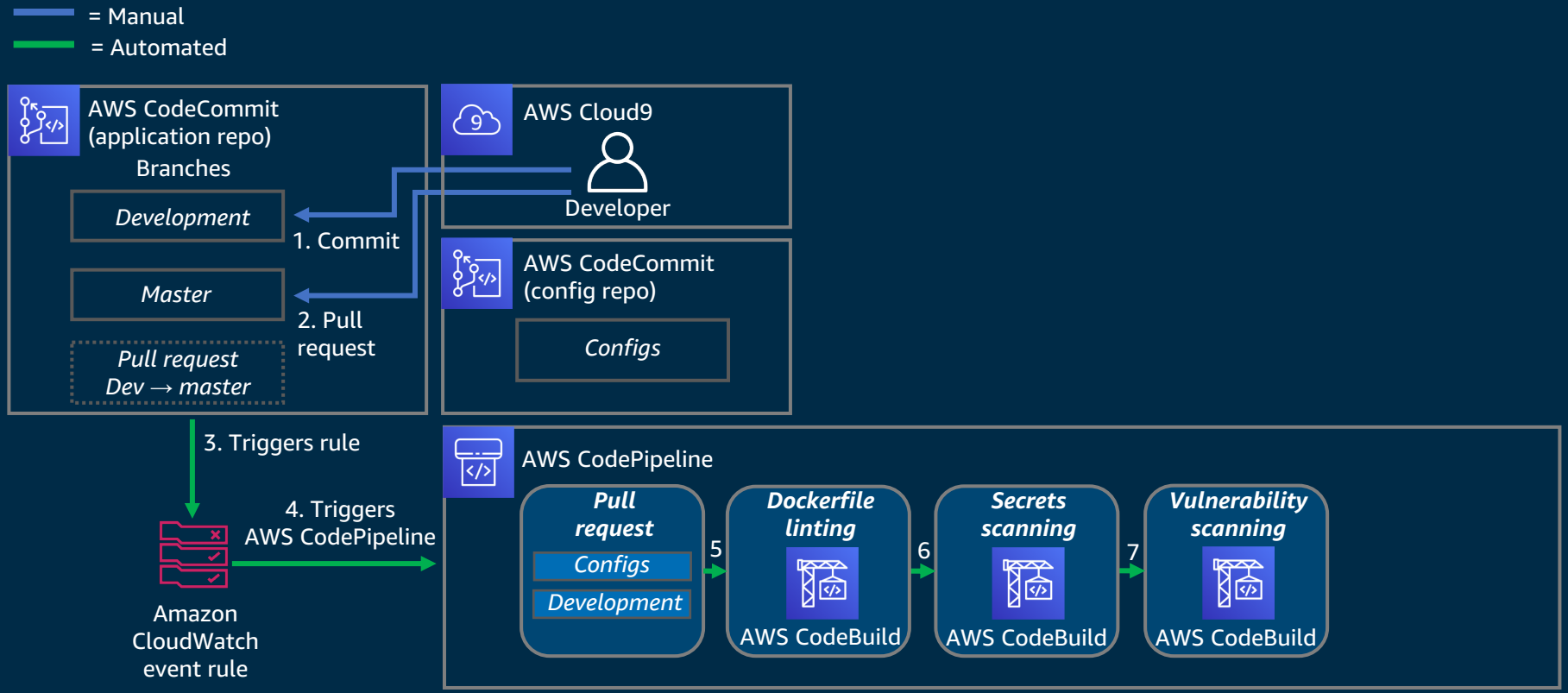
© 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved.



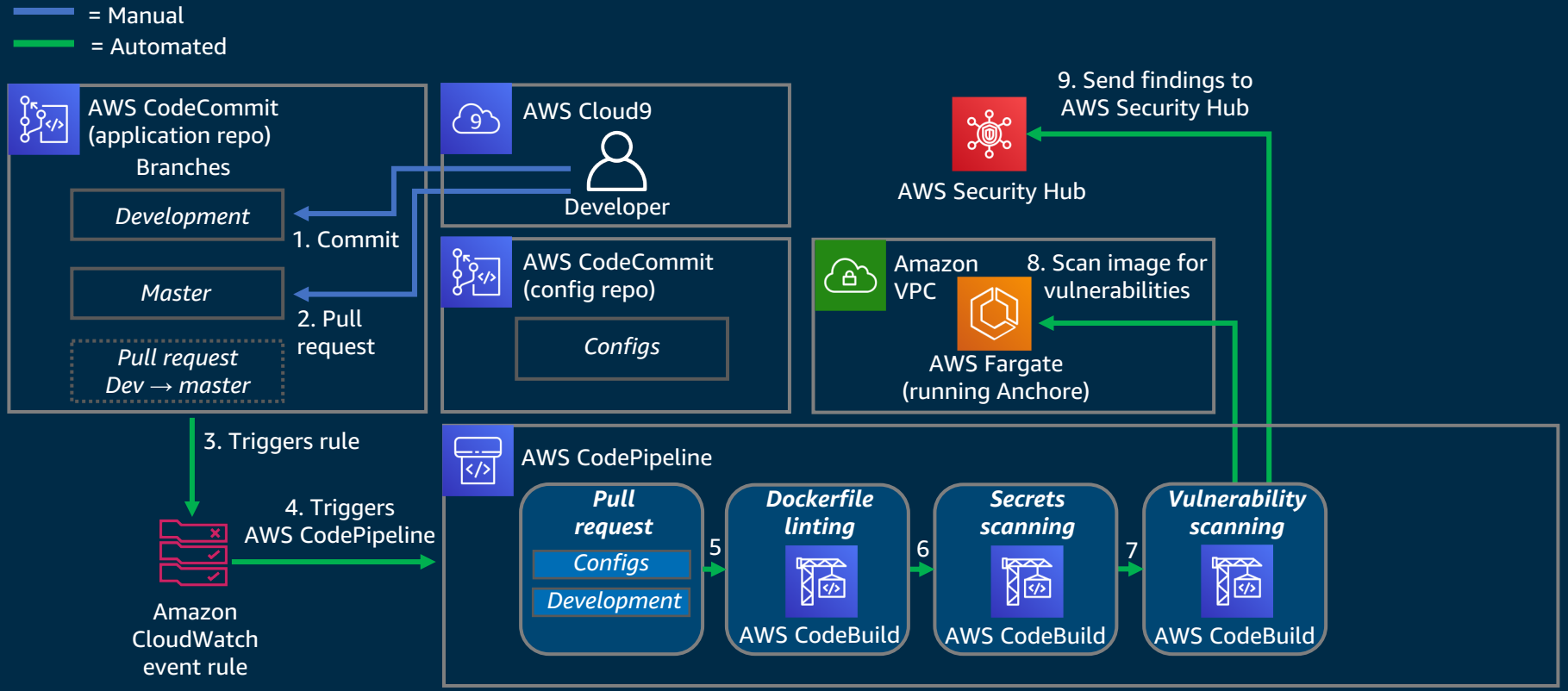
© 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved.



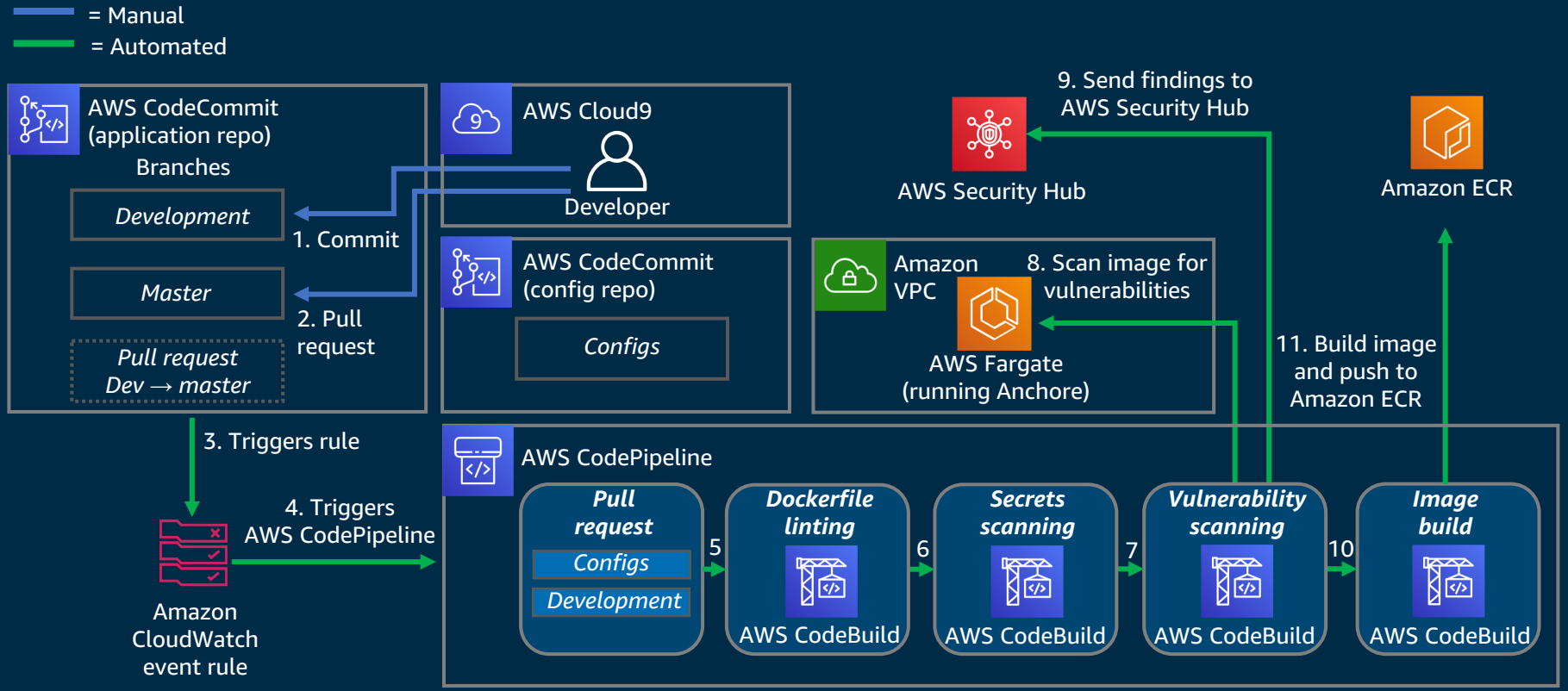
© 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved.



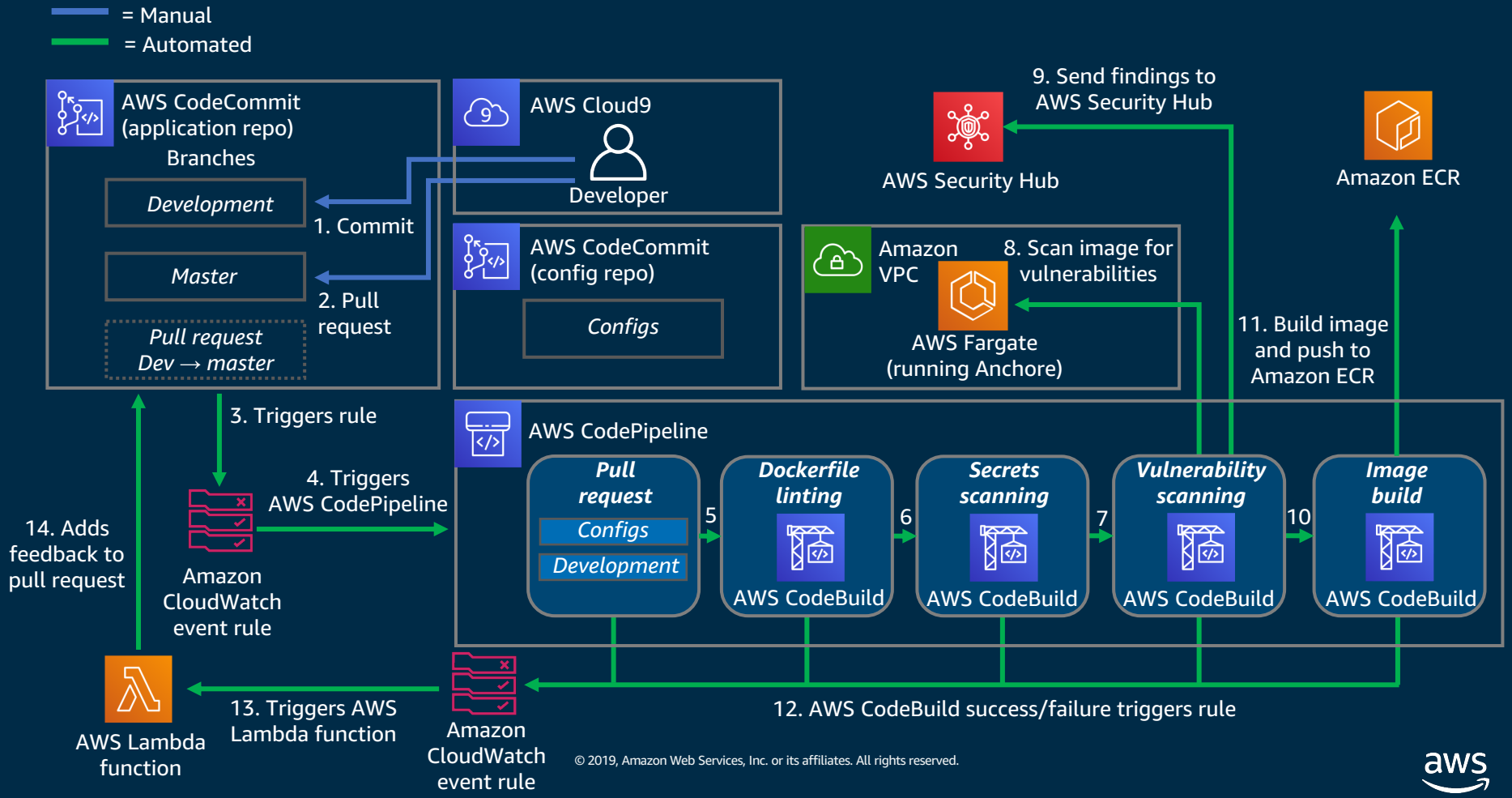
© 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved.



© 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved.



© 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved.



© 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Thank you!

aws RE:INFORCE

© 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved.