Defining ECS Applications using Ansible and CloudFormation

Justin Menga
FULL STACK TECHNOLOGIST
@jmenga pseudo.co.de

Introduction

Ansible Playbook

- AWS CloudFormation role
- Create a development environment

CloudFormation Stack

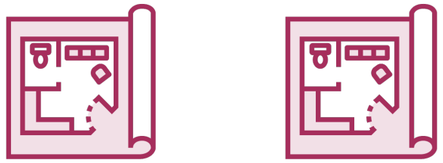
- EC2 Autoscaling Group
- Application Load Balancers
- DNS Records
- RDS Instance
- CloudWatch Logs
- Security Groups and IAM Roles

Creating the Microtrader Deployment Playbook

Configuring EC2 Autoscaling Groups

Microtrader Application Stack

ECS Task Definitions



Route 53 Private DNS

dev-microtrader.dockerproductionaws.org

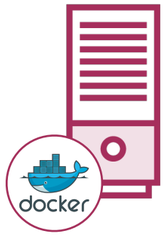
Public Load Balancer
(Internet Facing)

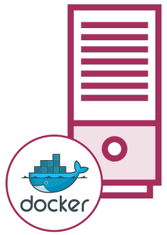
Dashboard Endpoint

Portfolio Service

Audit Service

Dashboard Service





Autoscaling Group

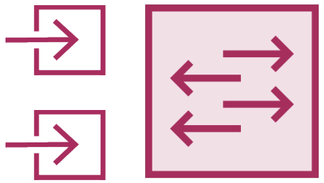
Quote Service

ECS Cluster

Application Load Balancer
(Internal)

Audit Endpoint

Quote Endpoint

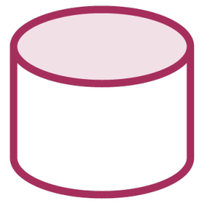


CloudWatch Log Groups

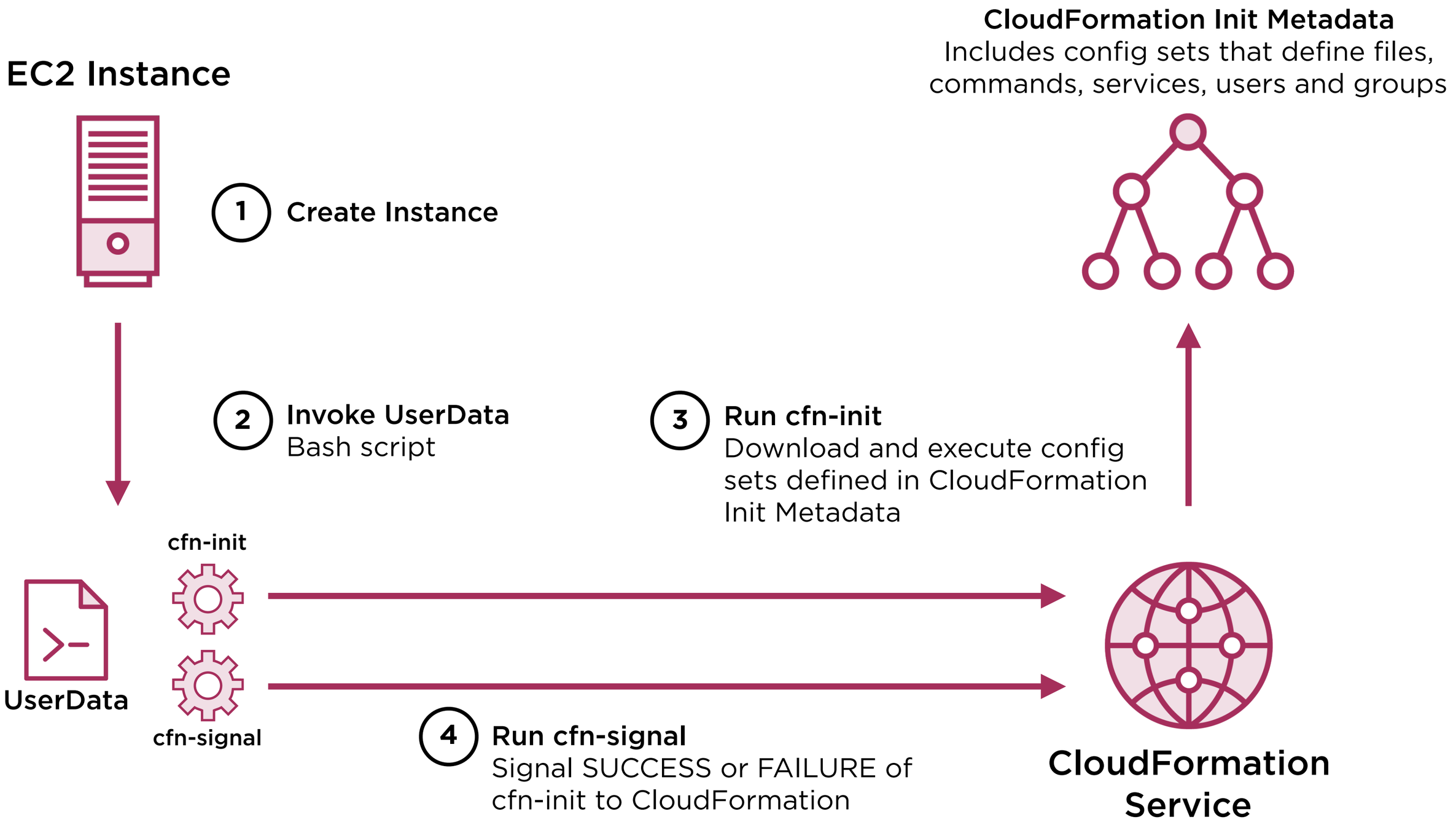
System Logs

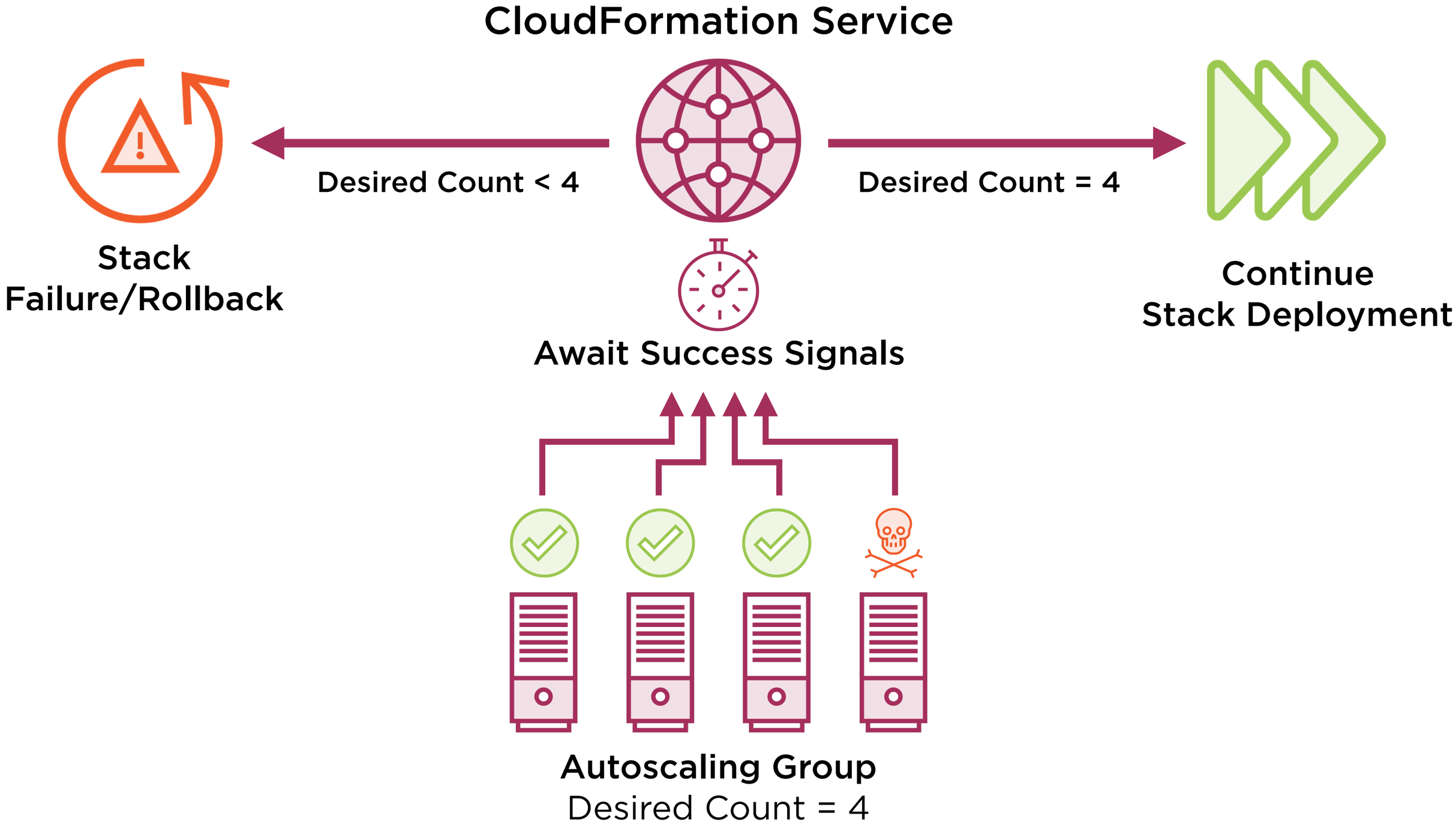
Container Logs

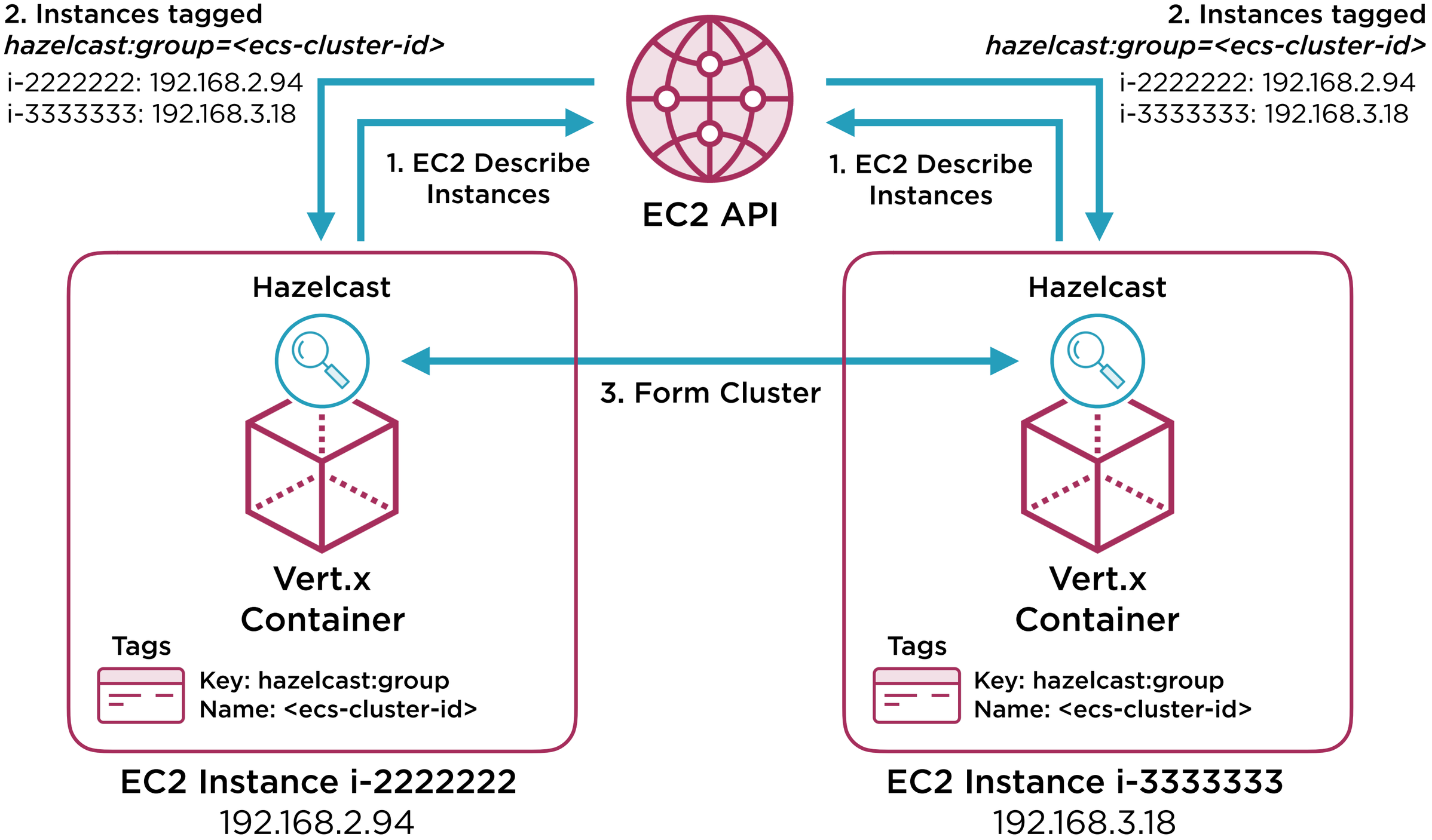
RDS Instance



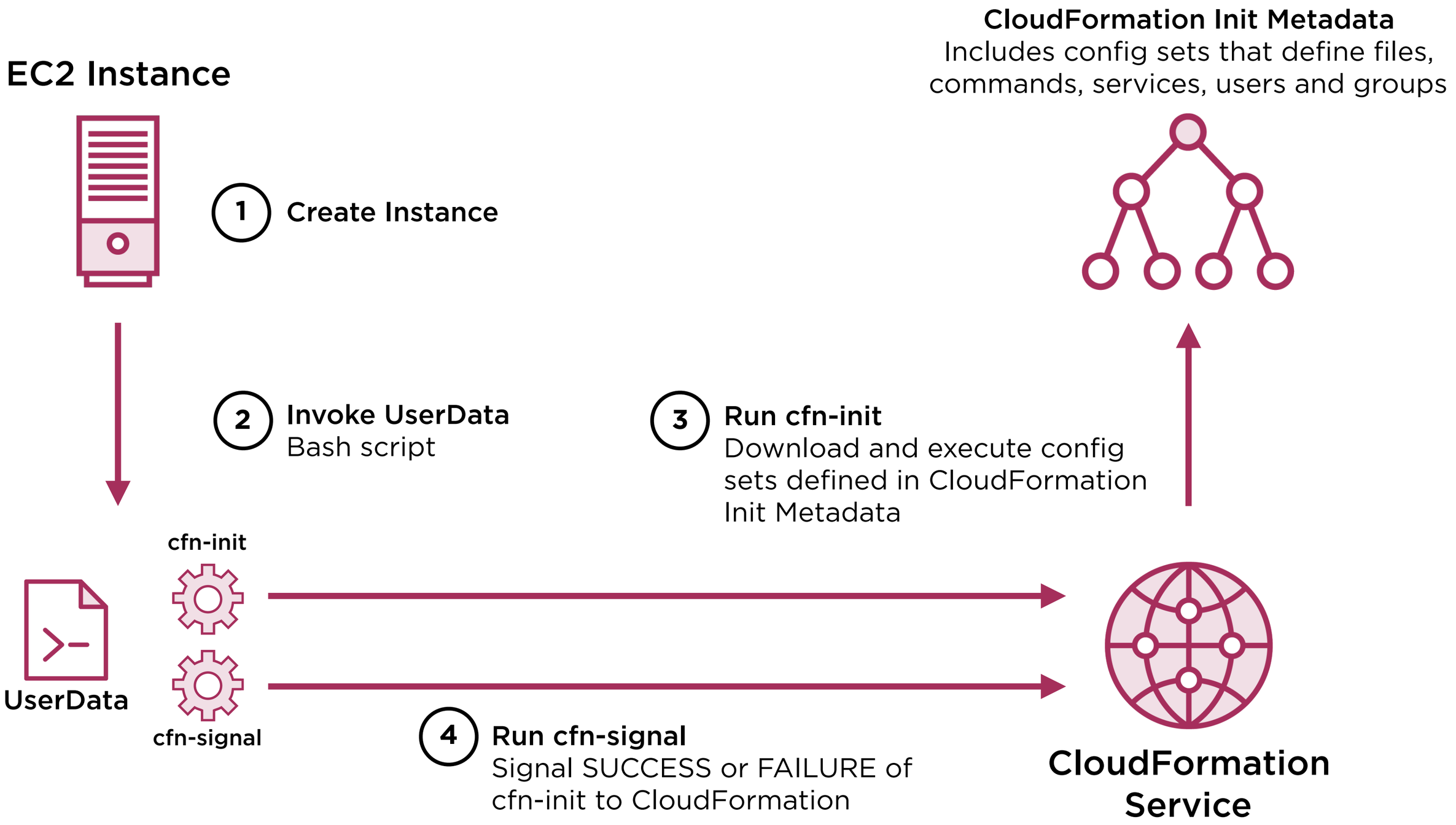
Audit Database


EC2 Instance
CloudFormation Init Metadata
Includes config sets that define files, commands, services, users and groups
1 Create Instance
2 Invoke UserData
Bash script
3 Run cfn-init
Download and execute config sets defined in CloudFormation Init Metadata
cfn-init
UserData
cfn-signal
4 Run cfn-signal
Signal SUCCESS or FAILURE of cfn-init to CloudFormation
CloudFormation Service


CloudFormation Service
Desired Count < 4
Desired Count = 4
Stack
Failure/Rollback
Continue
Stack Deployment
Await Success Signals
Autoscaling Group
Desired Count = 4

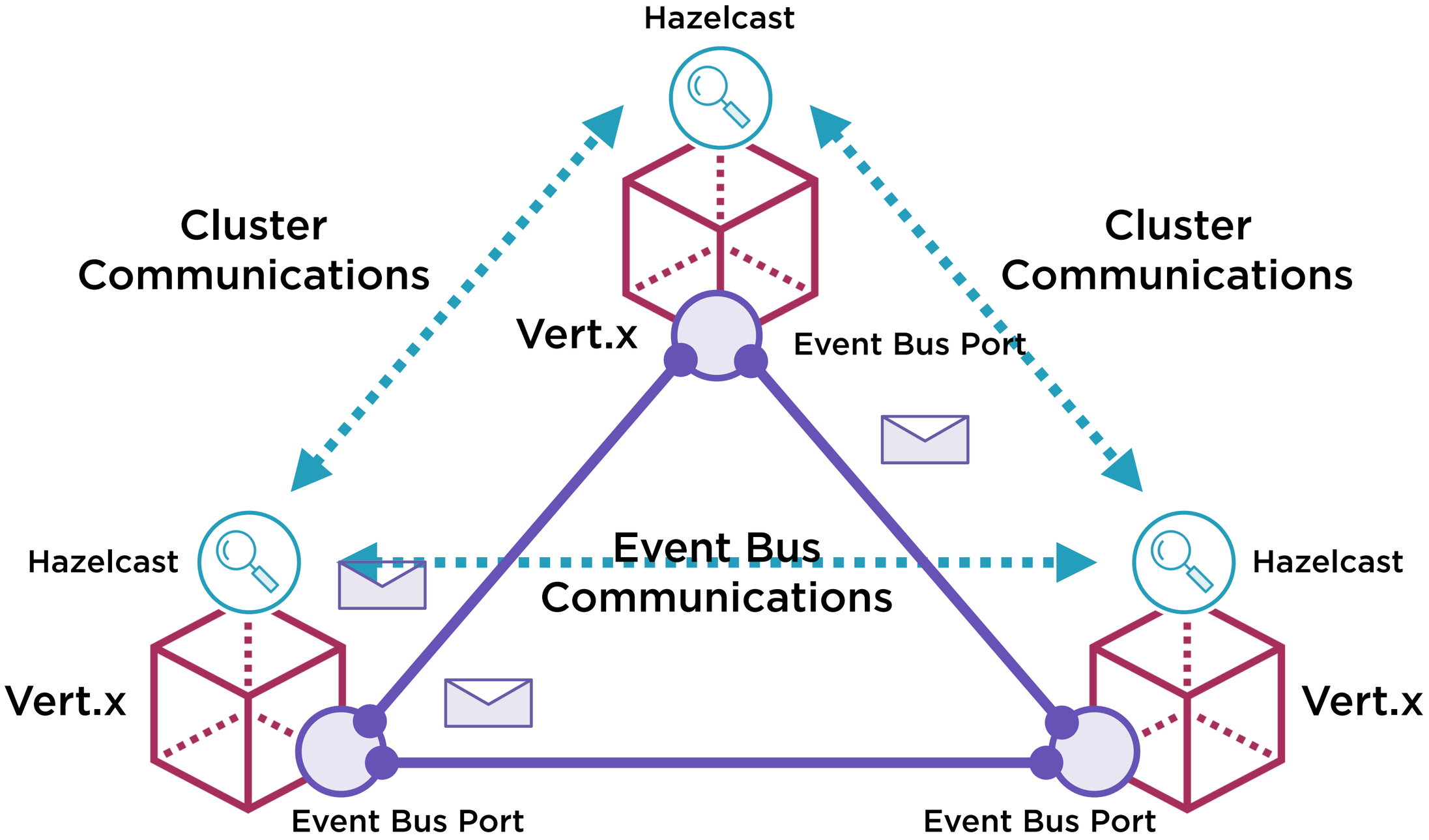

2. Instances tagged
hazelcast:group=<ecs-cluster-id>
i-2222222: 192.168.2.94
i-3333333: 192.168.3.18
1. EC2 Describe Instances
EC2 API
2. Instances tagged
hazelcast:group=<ecs-cluster-id>
i-2222222: 192.168.2.94
i-3333333: 192.168.3.18
1. EC2 Describe Instances
Hazelcast
Vert.x Container
Tags
Key: hazelcast:group
Name: <ecs-cluster-id>
3. Form Cluster
Hazelcast
Vert.x Container
Tags
Key: hazelcast:group
Name: <ecs-cluster-id>
EC2 Instance i-2222222
192.168.2.94
EC2 Instance i-3333333
192.168.3.18

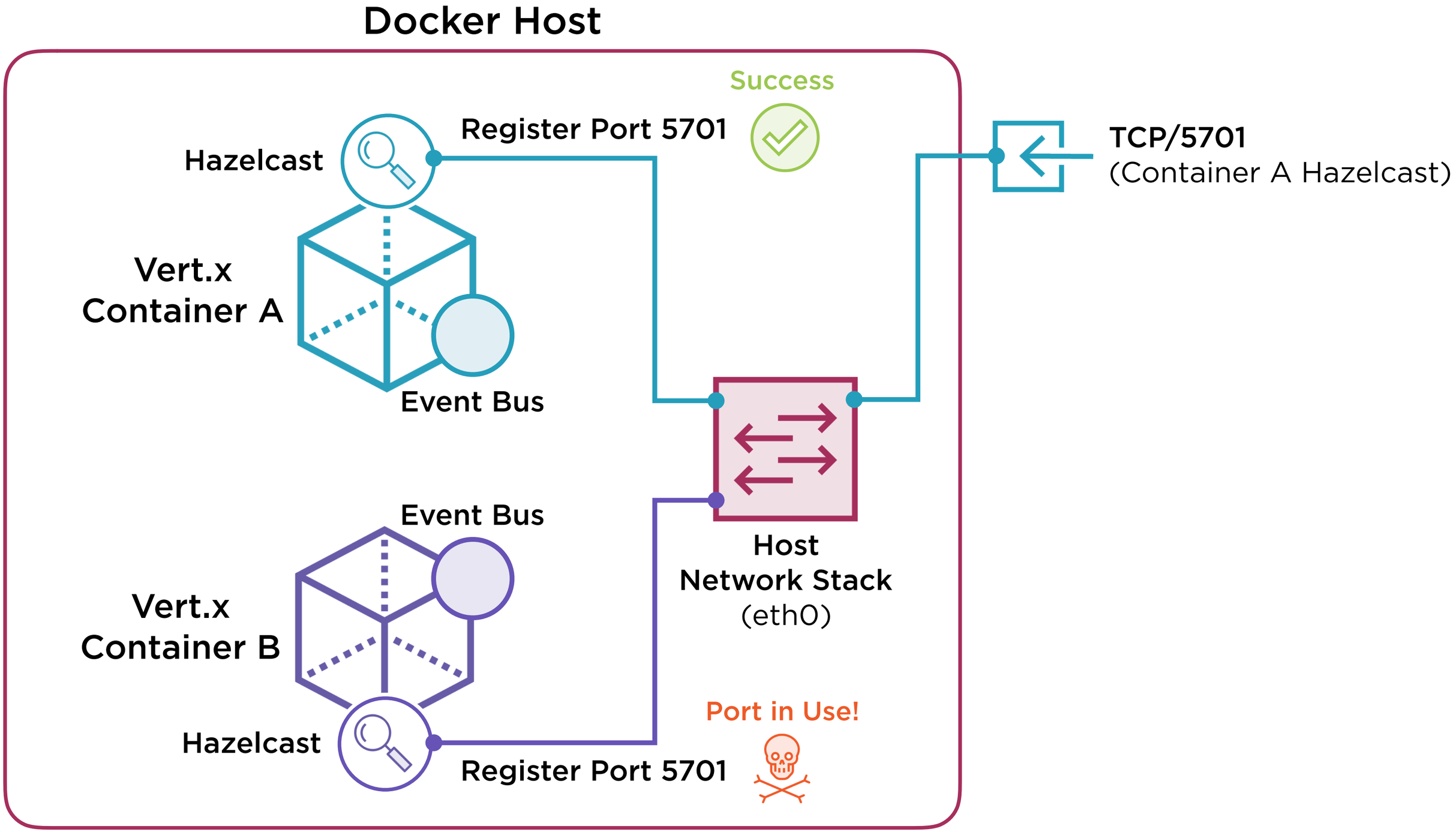
Configuring Autoscaling Launch Configurations

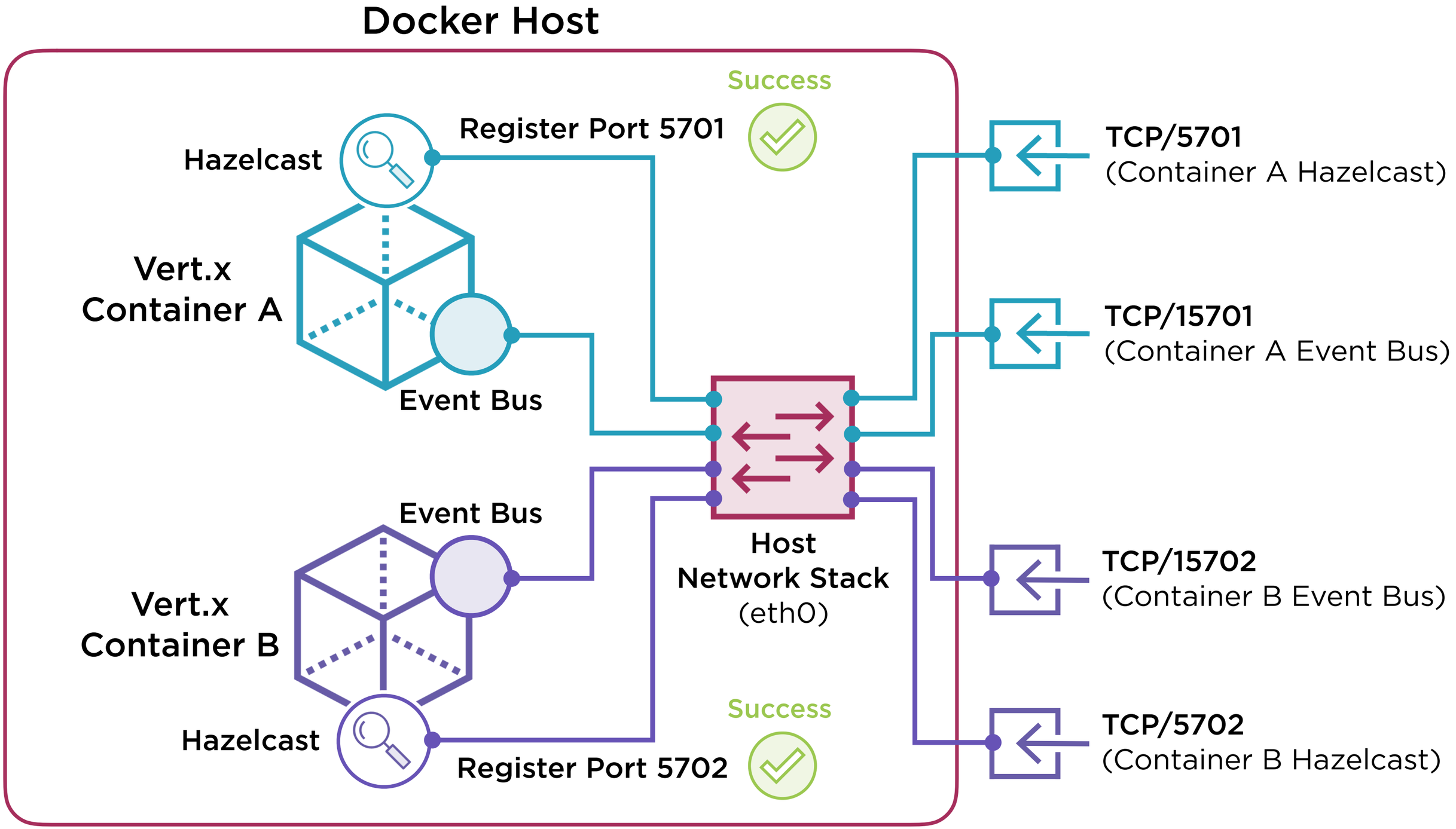

EC2 Instance
CloudFormation Init Metadata
Includes config sets that define files, commands, services, users and groups
1 Create Instance
2 Invoke UserData
Bash script
3 Run cfn-init
Download and execute config sets defined in CloudFormation Init Metadata
cfn-init
UserData
cfn-signal
4 Run cfn-signal
Signal SUCCESS or FAILURE of cfn-init to CloudFormation
CloudFormation Service

Configuring CloudFormation Init Metadata

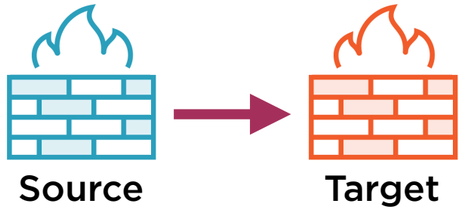
Configuring EC2 Autoscaling Security Groups


Hazelcast
Cluster
Communications
Cluster
Communications
Vert.x
Event Bus Port
Hazelcast
Event Bus
Communications
Hazelcast
Vert.x
Vert.x
Event Bus Port
Event Bus Port

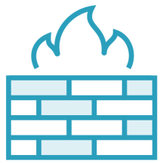

Docker Host
Success
Register Port 5701
Hazelcast
Vert.x
Container A
Event Bus
TCP/5701
(Container A Hazelcast)
Host
Network Stack
(eth0)
Event Bus
Vert.x
Container B
Hazelcast
Port in Use!
Register Port 5701


Docker Host
Success
Register Port 5701
Hazelcast
Vert.x
Container A
Event Bus
TCP/5701
(Container A Hazelcast)
TCP/15701
(Container A Event Bus)
Host
Network Stack
(eth0)
Event Bus
Vert.x
Container B
Hazelcast
Success
Register Port 5702
TCP/15702
(Container B Event Bus)
TCP/5702
(Container B Hazelcast)

AWS::EC2::SecurityGroupIngress



Permit: “tcp”
FromPort: 5701
ToPort: 5710
SourceSecurityGroupId: { “Ref”: “ApplicationSecurityGroup” }
GroupId: { “Ref”: “ApplicationSecurityGroup” }

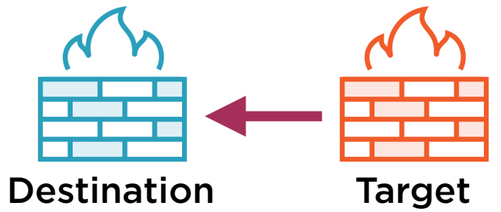


Source Security Group
ApplicationSecurityGroup
sg-11112222

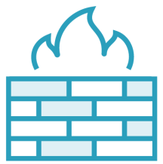
Target Security Group
ApplicationSecurityGroup
sg-11112222

Ingress Rule Added to Target Group
Permit: “tcp”
FromPort: 5701
ToPort: 5710
SourceSecurityGroupId: sg-11112222

AWS::EC2::SecurityGroupEgress



Permit: “tcp”
FromPort: 5701
ToPort: 5710
DestinationSecurityGroupId: { “Ref”: “ApplicationSecurityGroup” }
GroupId: { “Ref”: “ApplicationSecurityGroup” }



Destination Security Group
ApplicationSecurityGroup
sg-11112222

Target Security Group
ApplicationSecurityGroup
sg-11112222

Egress Rule Added to Target Group
Permit: “tcp”
FromPort: 5701
ToPort: 5710
DestinationSecurityGroupId: sg-11112222

```
ApplicationAutoscalingSecurityGroup:
  Type: AWS::EC2::SecurityGroup
    Properties:
      SecurityGroupIngress:
        - IpProtocol: tcp
          FromPort: 22
          ToPort: 22
          CidrIp:
            Fn::ImportValue:
              Fn::Sub: ${VpcName}VpcCidr
```

+

```
MicrotraderClusterDiscoveryIngress:
    Type: AWS::EC2::SecurityGroupIngress
    Properties:
      IpProtocol: tcp
      FromPort: 5701
      ToPort: 5710
      SourceSecurityGroupId:
        Ref: ApplicationAutoscalingSecurityGroup
      GroupId:
        Ref: ApplicationAutoscalingSecurityGroup
```

→

```
ApplicationAutoscalingSecurityGroup:
  Type: AWS::EC2::SecurityGroup
    Properties:
      SecurityGroupIngress:
        - IpProtocol: tcp
          FromPort: 22
          ToPort: 22
          CidrIp:
            Fn::ImportValue:
              Fn::Sub: ${VpcName}VpcCidr
        - IpProtocol: "tcp"
          FromPort: 5701
          ToPort: 5710
          SourceSecurityGroupId:
            Ref: ApplicationAutoscalingSecurityGroup
```

```
ApplicationAutoscalingSecurityGroup:
  Type: AWS::EC2::SecurityGroup
    Properties:
      SecurityGroupIngress:
        - IpProtocol: tcp
          FromPort: 22
          ToPort: 22
          CidrIp:
            Fn::ImportValue:
              Fn::Sub: ${VpcName}VpcCidr
```

+

```
MicrotraderClusterDiscoveryEgress:
    Type: AWS::EC2::SecurityGroupEgress
    Properties:
      IpProtocol: tcp
      FromPort: 5701
      ToPort: 5710
      DestinationSecurityGroupId:
        Ref: ApplicationAutoscalingSecurityGroup
      GroupId:
        Ref: ApplicationAutoscalingSecurityGroup
```

→

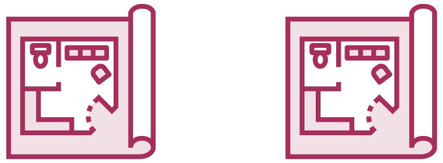
```
ApplicationAutoscalingSecurityGroup:
  Type: AWS::EC2::SecurityGroup
    Properties:
      SecurityGroupIngress:
        - IpProtocol: tcp
          FromPort: 22
          ToPort: 22
          CidrIp:
            Fn::ImportValue:
              Fn::Sub: ${VpcName}VpcCidr
        - IpProtocol: tcp
          FromPort: 5701
          ToPort: 5710
          SourceSecurityGroupId:
            Ref: ApplicationAutoscalingSecurityGroup
      SecurityGroupEgress:
        - IpProtocol: tcp
          FromPort: 5701
          ToPort: 5710
          SourceSecurityGroupId:
            Ref: ApplicationAutoscalingSecurityGroup
      ...
      ...
```

Configuring Autoscaling EC2 Instance Profiles

Configuring a Public Load Balancer

Microtrader Application Stack

ECS Task Definitions



Route 53 Private DNS

dev-microtrader.dockerproductionaws.org

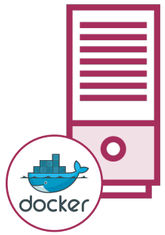
Public Load Balancer
(Internet Facing)

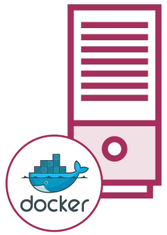
Dashboard Endpoint

Portfolio Service

Audit Service

Dashboard Service





Autoscaling Group

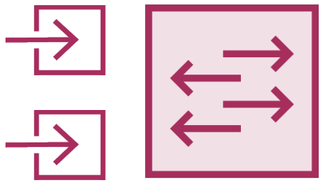
Quote Service

ECS Cluster

Application Load Balancer
(Internal)

Audit Endpoint

Quote Endpoint

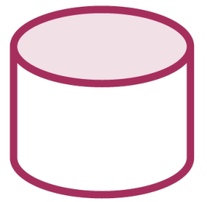


CloudWatch Log Groups

System Logs

Container Logs

RDS Instance



Audit Database

Classic vs Application Load Balancers

Classic Load Balancer

L4 Load Balancer with L7 Awareness

Support for non-HTTP TCP applications

Support for targets with multiple ports

Application Load Balancers

Layer 7 Load Balancer

Host-based routing

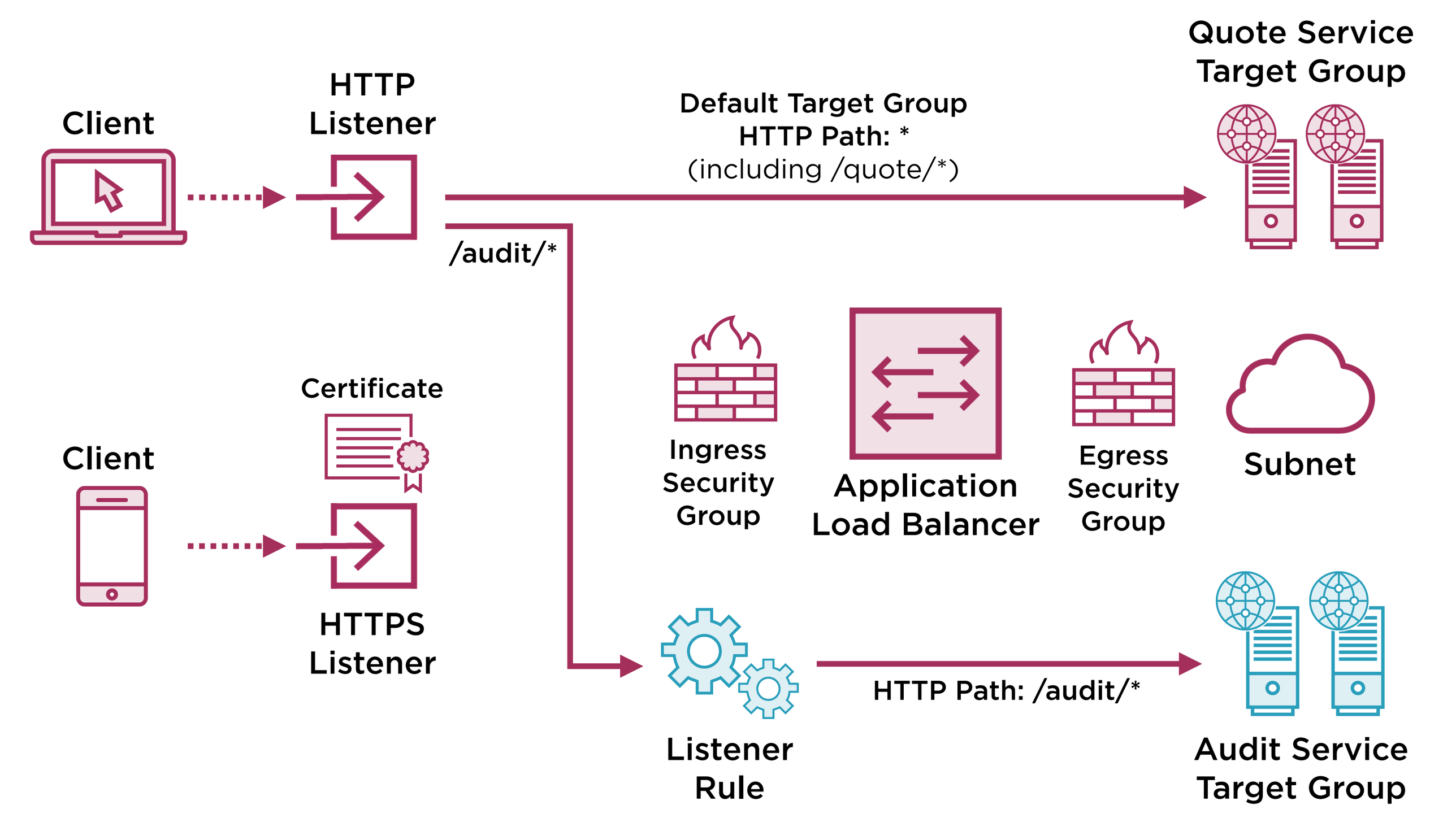
Path-based routing

ECS dynamic port mapping

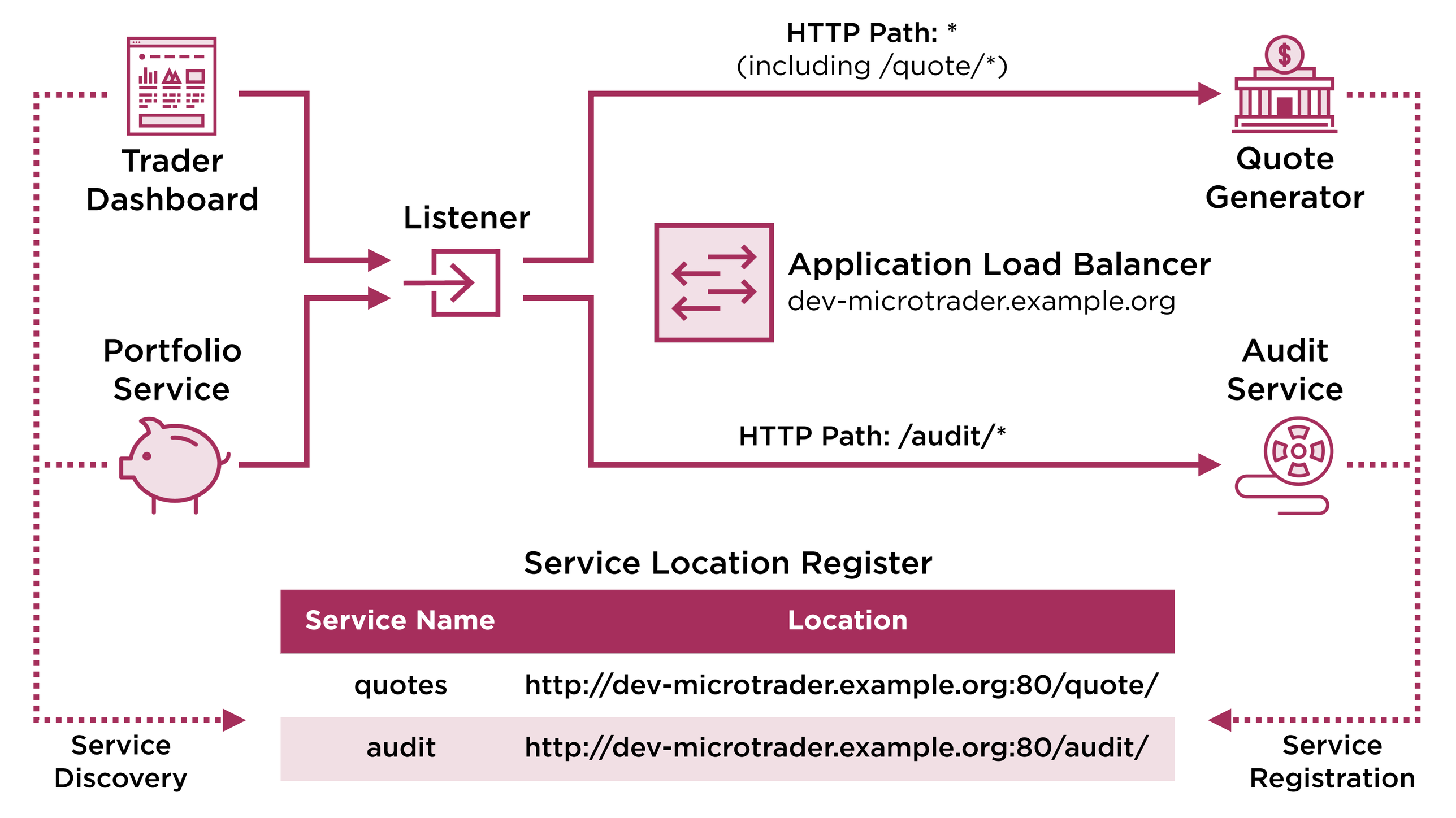
HTTP/2 support

WebSockets support

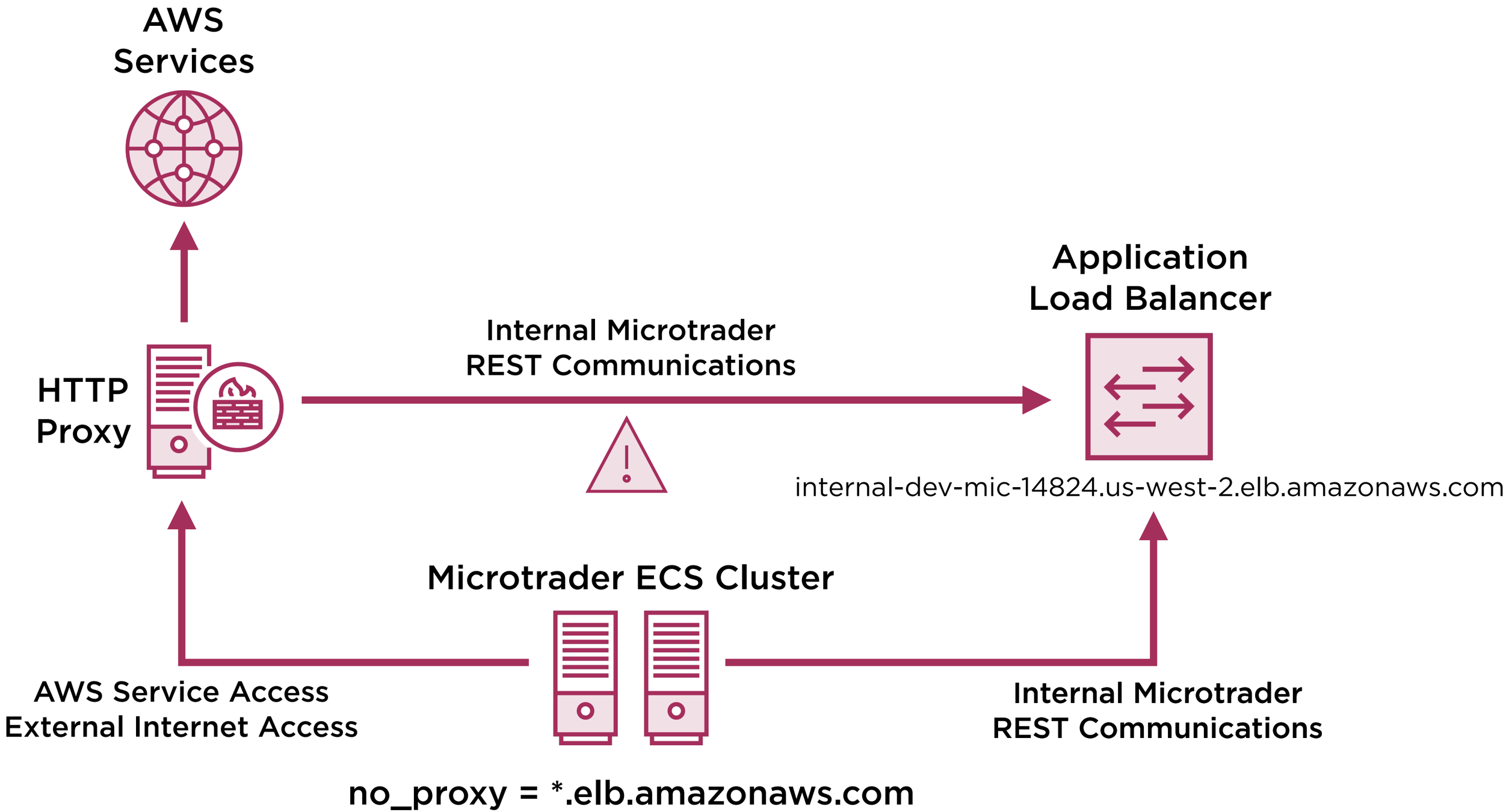
Native IPv6 support

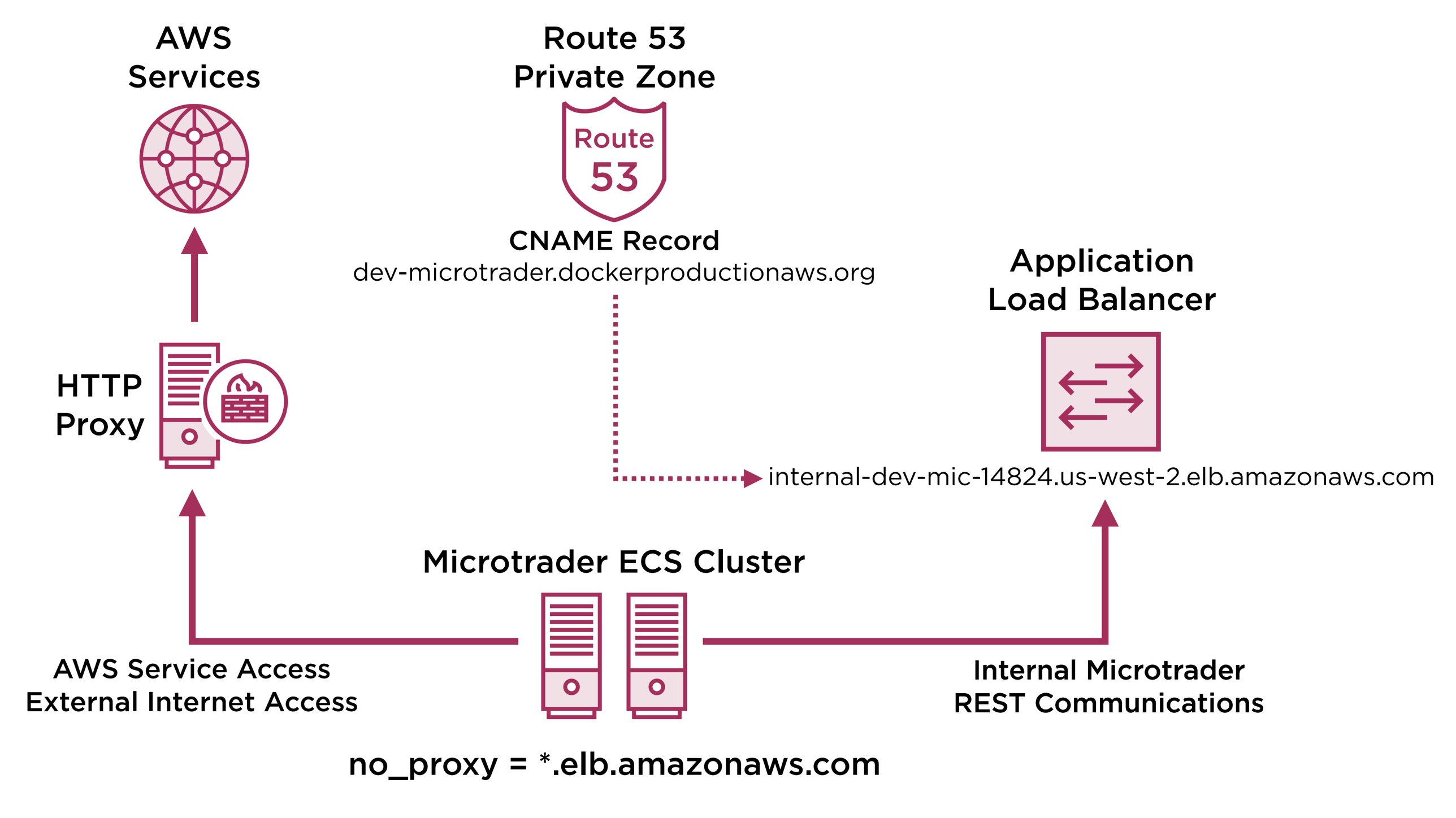

Quote Service
Target Group
Client
HTTP
Listener
Default Target Group
HTTP Path: *
(including /quote/*)
/audit/*
Certificate
Client
Ingress
Security
Group
Application
Load Balancer
Egress
Security
Group
Subnet
HTTPS
Listener
Listener
Rule
HTTP Path: /audit/*
Audit Service
Target Group

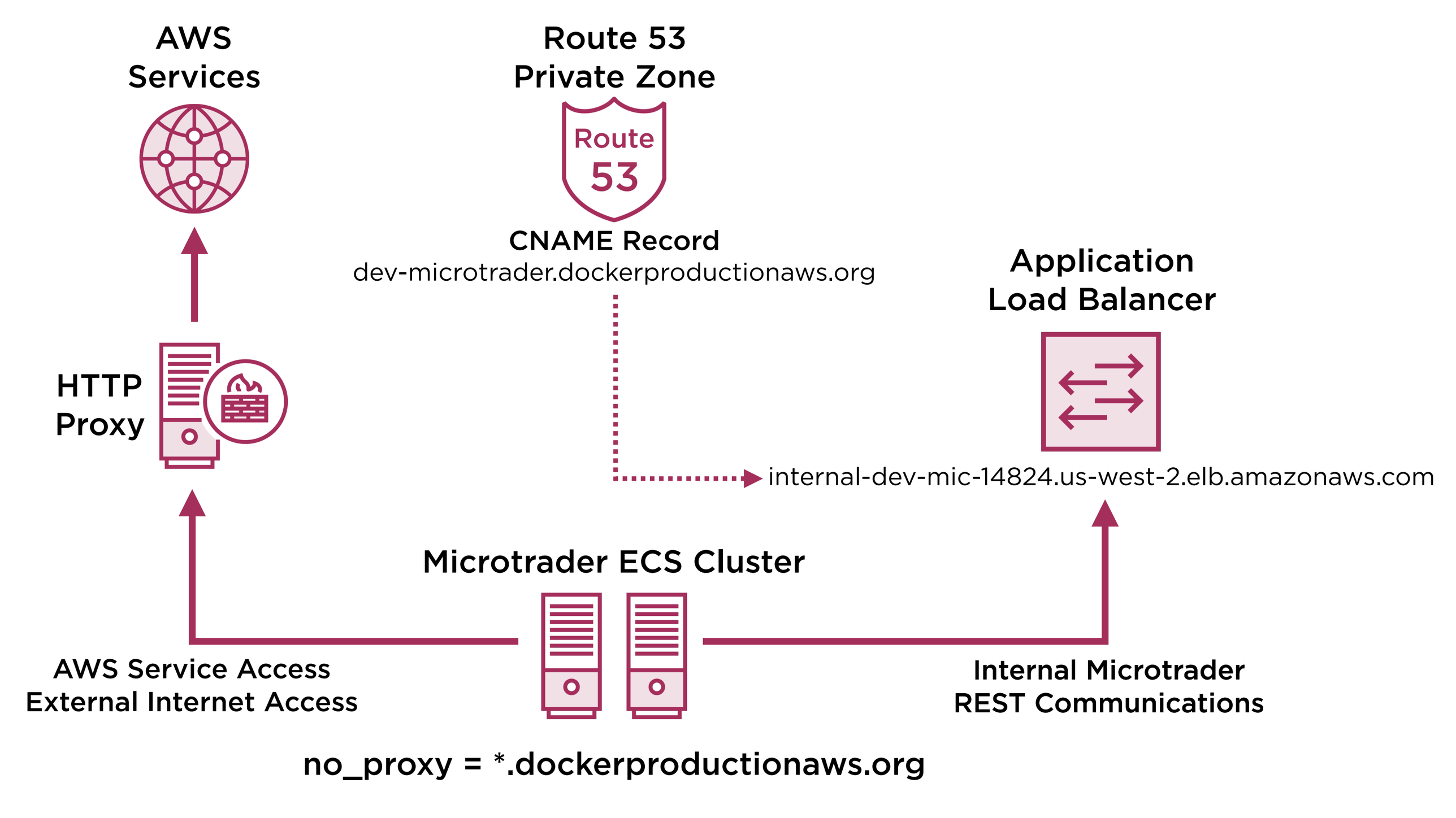
Configuring an Internal Load Balancer


HTTP Path: *
(including /quote/*)
Trader
Dashboard
Quote
Generator
Listener
Application Load Balancer
dev-microtrader.example.org
Portfolio
Service
Audit
Service
HTTP Path: /audit/*
Service Location Register
Service Name
Location
quotes
http://dev-microtrader.example.org:80/quote/
audit
http://dev-microtrader.example.org:80/audit/
Service
Discovery
Service
Registration

Configuring DNS Records


AWS
Services
HTTP
Proxy
Internal Microtrader
REST Communications
Application
Load Balancer
internal-dev-mic-14824.us-west-2.elb.amazonaws.com
Microtrader ECS Cluster
AWS Service Access
External Internet Access
Internal Microtrader
REST Communications
no_proxy = *.elb.amazonaws.com

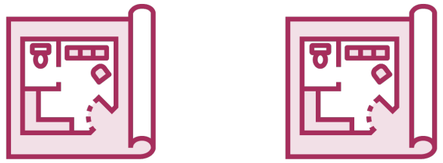

AWS
Services
Route 53
Private Zone
Route
53
CNAME Record
dev-microtrader.dockerproductionaws.org
Application
Load Balancer
HTTP
Proxy
internal-dev-mic-14824.us-west-2.elb.amazonaws.com
Microtrader ECS Cluster
AWS Service Access
External Internet Access
Internal Microtrader
REST Communications
no_proxy = *.elb.amazonaws.com


AWS
Services
Route 53
Private Zone
Route
53
CNAME Record
dev-microtrader.dockerproductionaws.org
Application
Load Balancer
HTTP
Proxy
internal-dev-mic-14824.us-west-2.elb.amazonaws.com
Microtrader ECS Cluster
AWS Service Access
External Internet Access
Internal Microtrader
REST Communications
no_proxy = *.dockerproductionaws.org

Configuring the Relational Database Service

Microtrader Application Stack

ECS Task Definitions



Route 53 Private DNS

dev-microtrader.dockerproductionaws.org

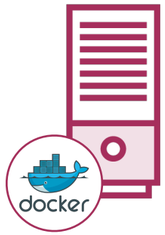
Public Load Balancer
(Internet Facing)

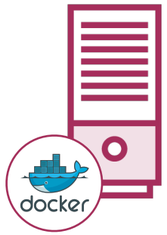
Dashboard Endpoint

Portfolio Service

Audit Service

Dashboard Service





Autoscaling Group

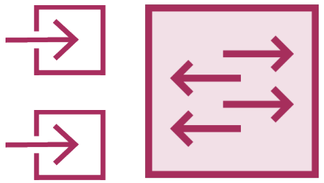
Quote Service

ECS Cluster

Application Load Balancer
(Internal)

Audit Endpoint

Quote Endpoint

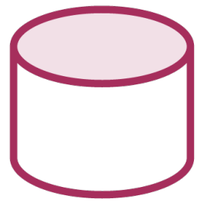


CloudWatch Log Groups

System Logs

Container Logs

RDS Instance



Audit Database

RDS Instance DeletionPolicy Options

Retain

Suitable for production environments

Retains RDS instance whilst other stack resources are deleted

You are responsible for charges and lifecycle of RDS instance

Snapshot

Suitable for production environments

Creates snapshot before deleting RDS instance

You are responsible for charges and lifecycle of RDS snapshot

Delete

Suitable for non-production environments

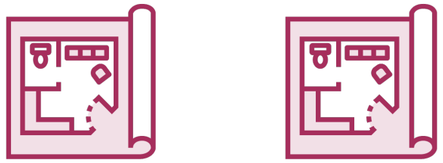
Deletes resources upon stack deletion

Default setting if deletion policy is not specified

Configuring CloudWatch Log Groups

Microtrader Application Stack

ECS Task Definitions



Route 53 Private DNS

dev-microtrader.dockerproductionaws.org

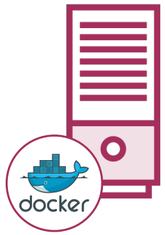
Public Load Balancer
(Internet Facing)

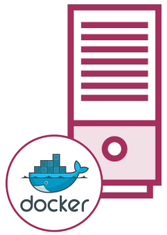
Dashboard Endpoint

Portfolio Service

Audit Service

Dashboard Service





Autoscaling Group

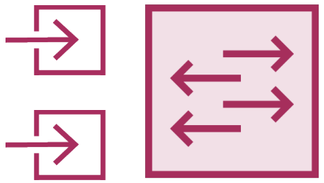
Quote Service

ECS Cluster

Application Load Balancer
(Internal)

Audit Endpoint

Quote Endpoint

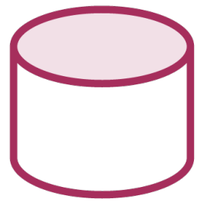


CloudWatch Log Groups

System Logs

Container Logs

RDS Instance



Audit Database

Summary

Defining AWS Infrastructure using Ansible and CloudFormation

- Autoscaling Groups
  - CloudFormation Init
- Application Load Balancers
  - Content-based routing
- RDS Instance
  - Multi-AZ failover
- CloudWatch Logs
- Security Groups
- IAM Roles