Customizing ECS Container Instances

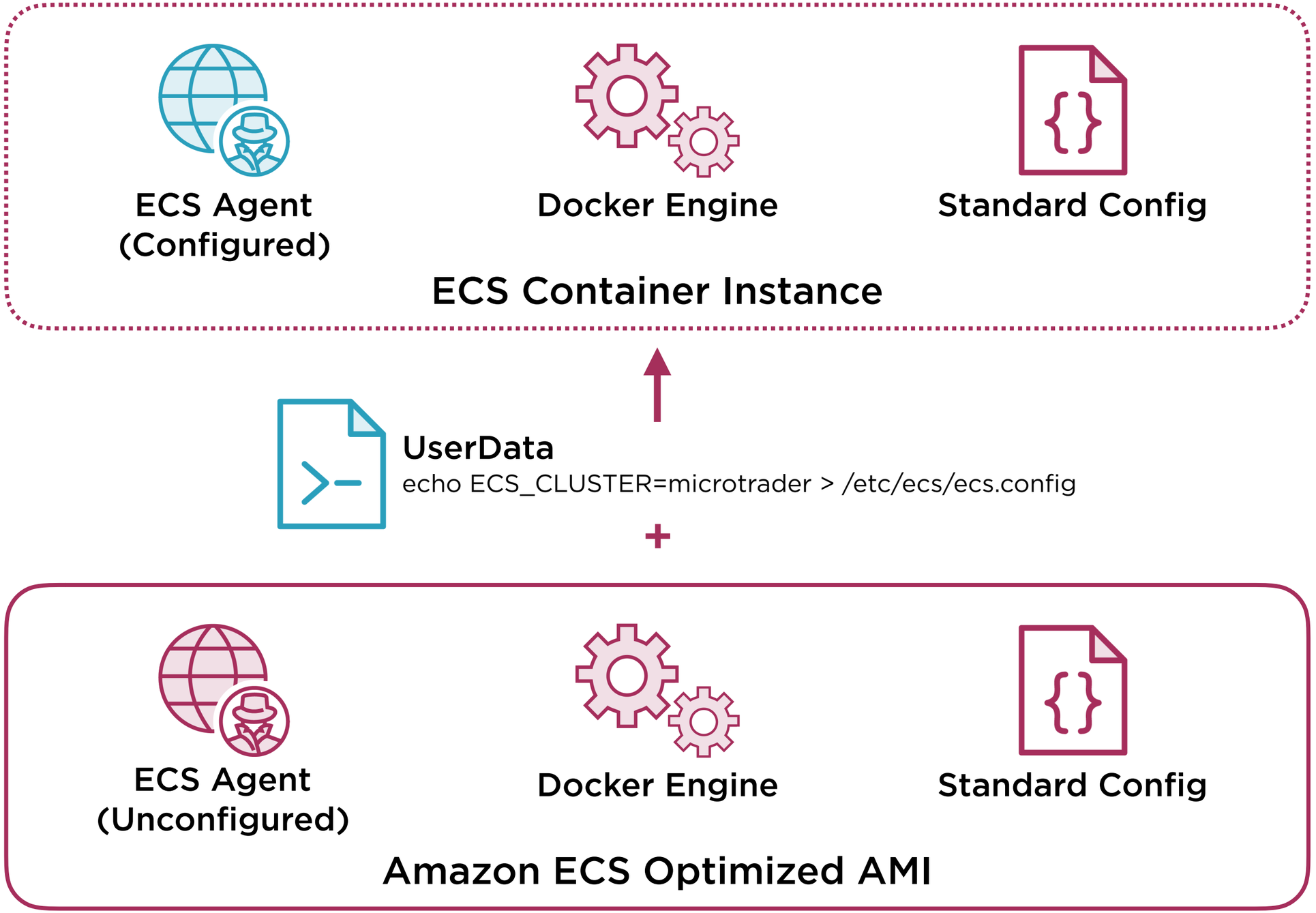
Justin Menga
FULL STACK TECHNOLOGIST
@jmenga pseudo.co.de

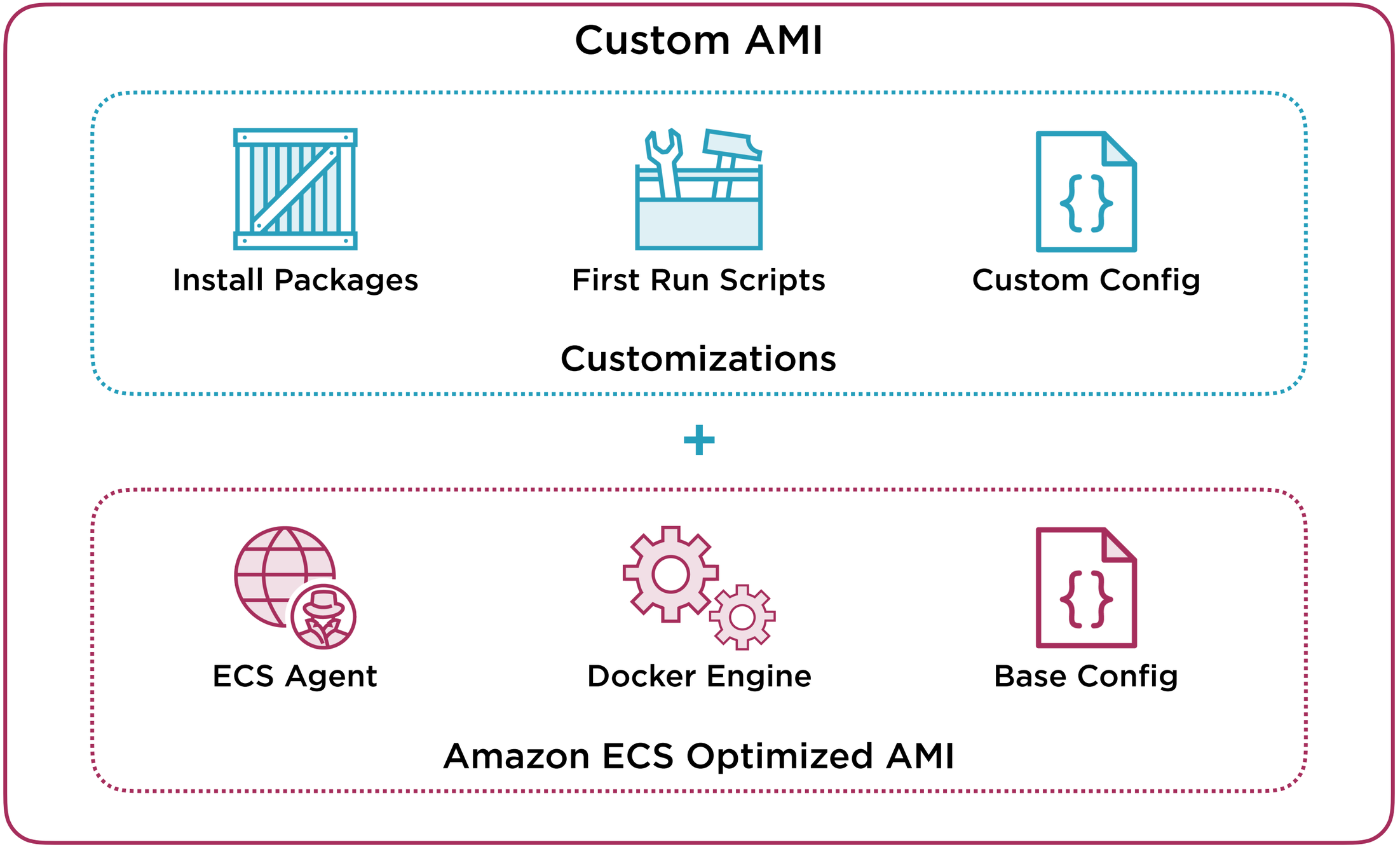
Introduction

Customizing ECS Container Instances

- Custom AMI Design
- Understanding EC2 instance initialization
- Using Packer to build Amazon Machine Images
- Customizing Docker
- CloudWatch Logs Integration
- HTTP Proxy Support
- ECS Container Instance Health Checks
- Building and Publishing the Image

Custom Amazon Machine Image Design


ECS Agent
(Configured)
Docker Engine
Standard Config
ECS Container Instance
UserData
echo ECS_CLUSTER=microtrader > /etc/ecs/ecs.config
+
ECS Agent
(Unconfigured)
Docker Engine
Standard Config
Amazon ECS Optimized AMI


Custom AMI
Install Packages
First Run Scripts
Custom Config
Customizations
+
ECS Agent
Docker Engine
Base Config
Amazon ECS Optimized AMI

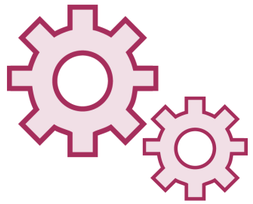
Custom AMI

CloudWatch Logs Agent

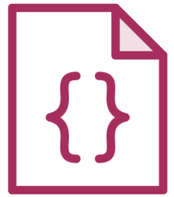
First Run Script
- HTTP Proxy Support
- ECS Agent Config
- CloudWatch Logs Config
- Health Check

Custom Config
- Timezone
- Enable NTP
- Custom Docker Config

ECS Agent



Docker Engine



Base Config

Amazon ECS Optimized AMI

Understanding EC2 Instance Initialization

Example UserData Script

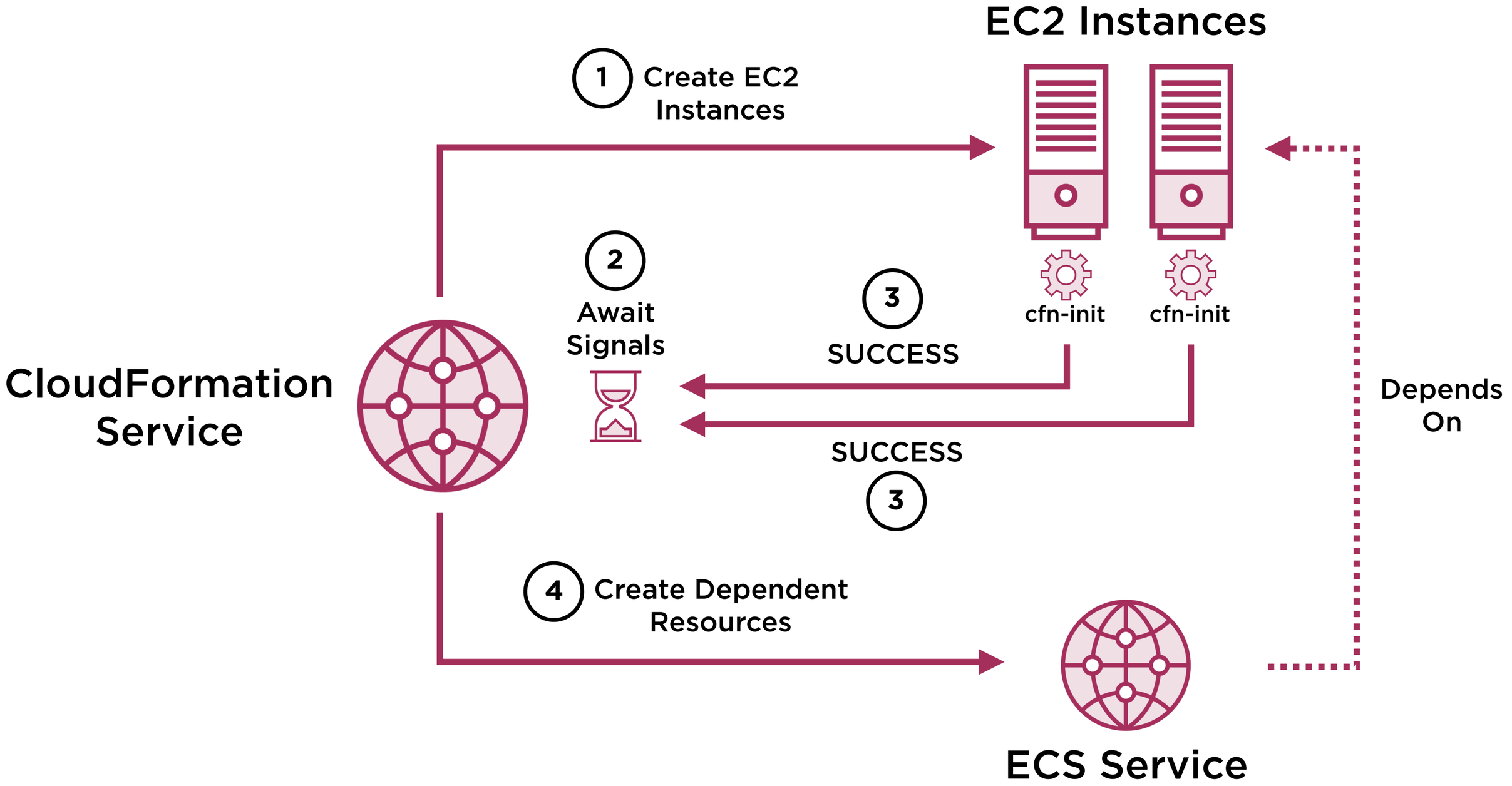
```
#!/usr/bin/env bash

yum install awslogs -y

echo “ENABLED=true” > /etc/awslogs.conf

service awslogs start

...
...
```

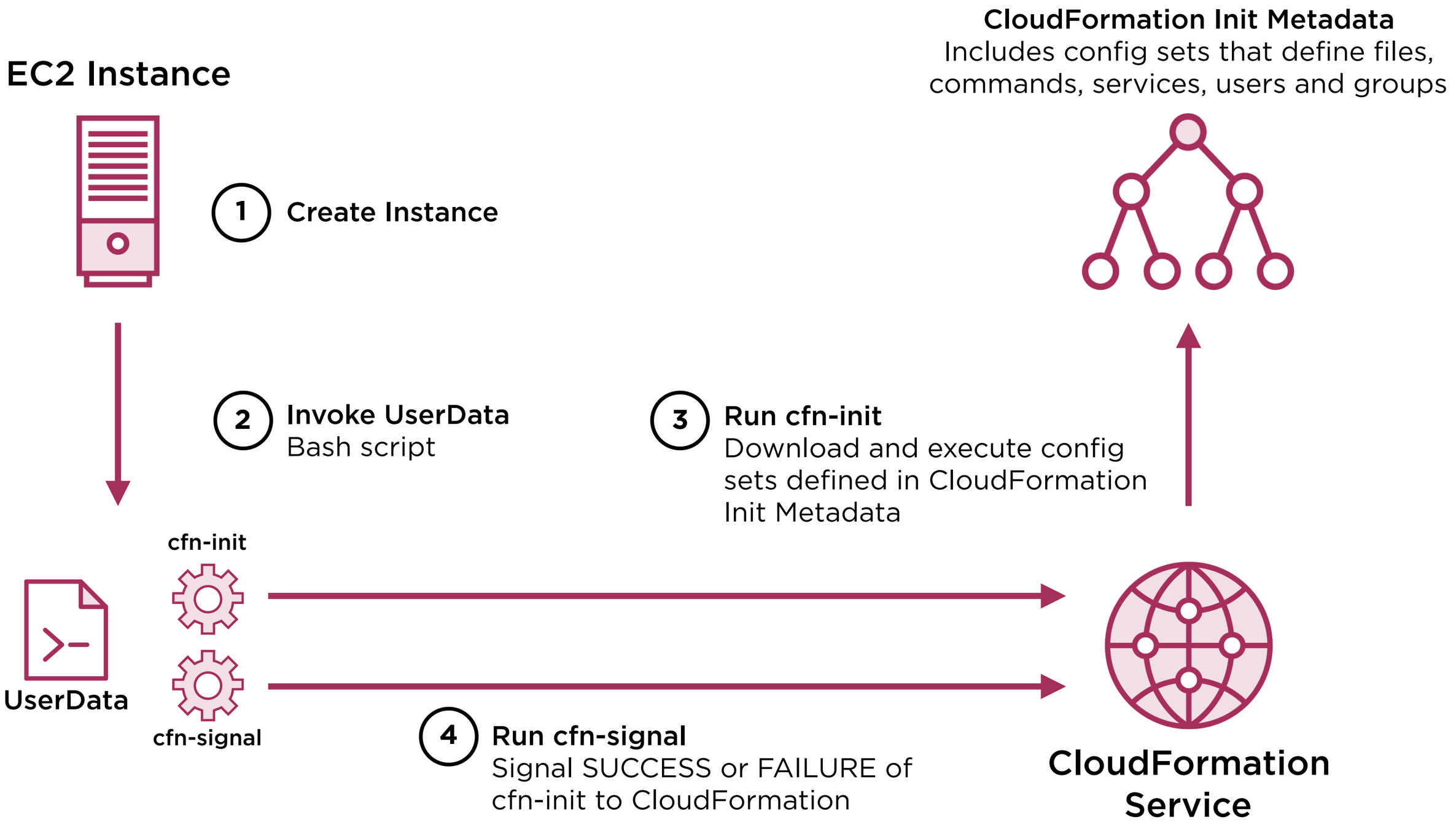
- Declare shell script
- Install packages
- Configure packages
- Start services
- Run additional commands

```
config:
 commands:
  01_install_awslogs:
   command: “yum install awslogs -y”
   env:
    MY_ENV: “true”
   cwd: “/home/ec2-user”
 files:
  /etc/awslogs.conf:
    content: “ENABLED=true”
 services:
  sysvinit:
   awslogs:
    enabled: “true”
    ensureRunning: “true”
```

CloudFormation Init Metadata

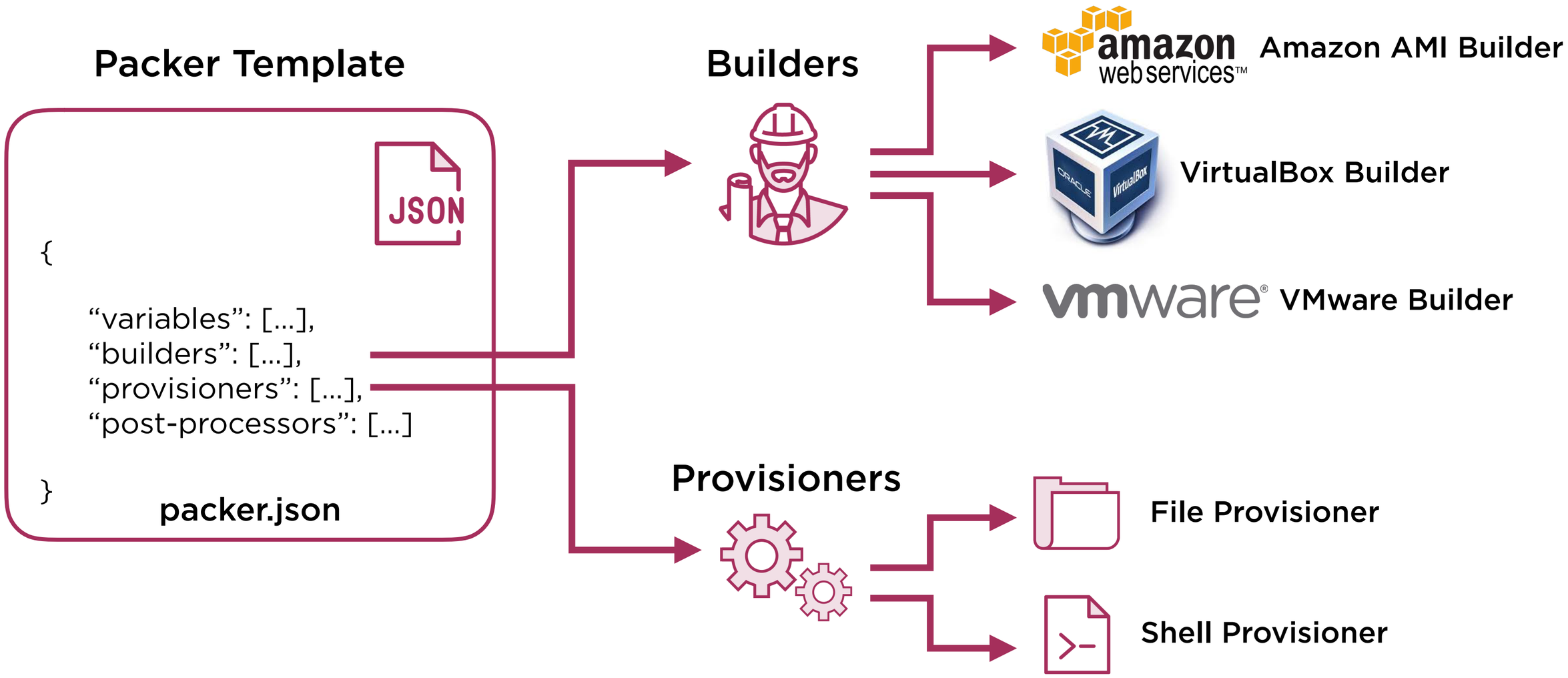
- Config Set
- Commands to run
- Environment settings for commands
- Current working directory for commands
- Files to create
- Services to configure and start


EC2 Instances
1
Create EC2 Instances
2
Await Signals
3
SUCCESS
cfn-init
cfn-init
SUCCESS
3
CloudFormation Service
Depends On
4
Create Dependent Resources
ECS Service

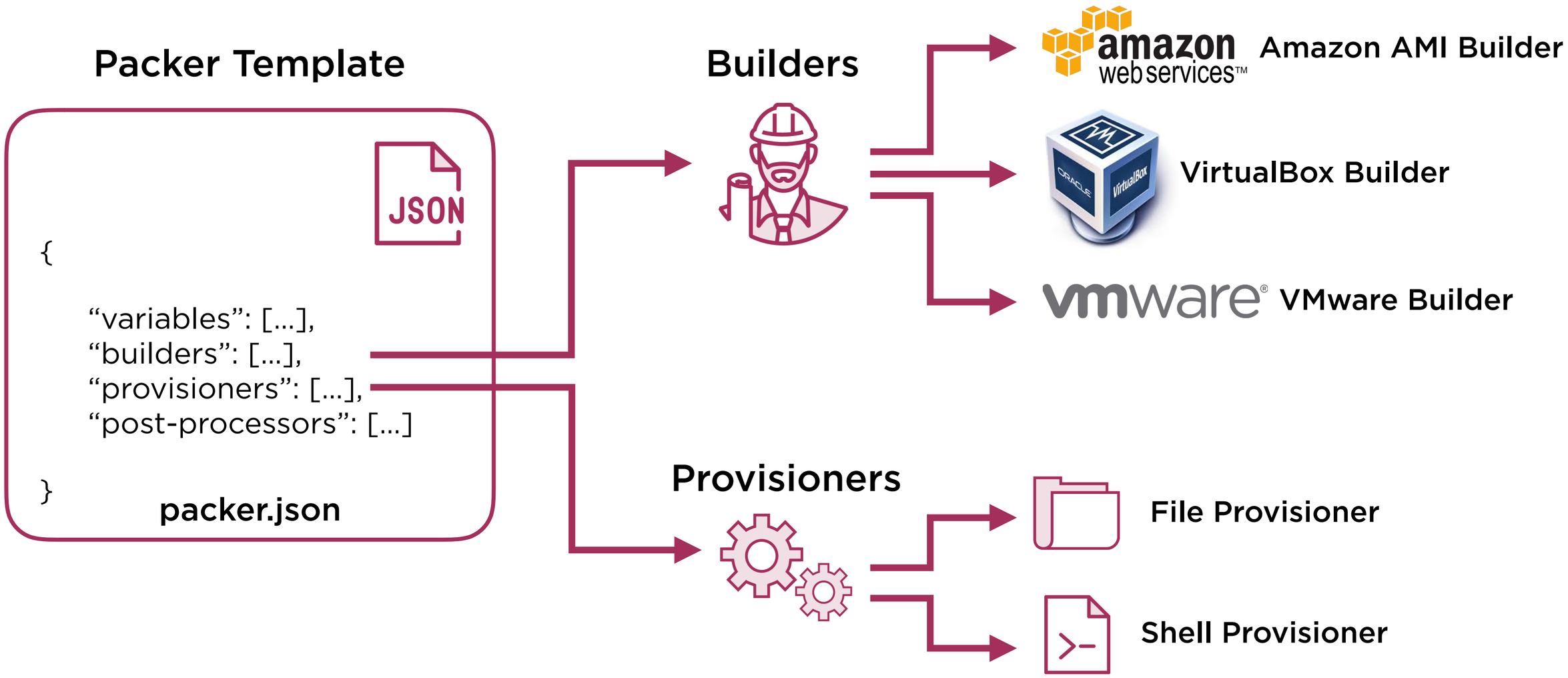

EC2 Instance
CloudFormation Init Metadata
Includes config sets that define files, commands, services, users and groups
1 Create Instance
2 Invoke UserData
Bash script
3 Run cfn-init
Download and execute config sets defined in CloudFormation Init Metadata
cfn-init
UserData
cfn-signal
4 Run cfn-signal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

Using Packer to build Amazon Machine Images

Packer Architecture



Packer Architecture

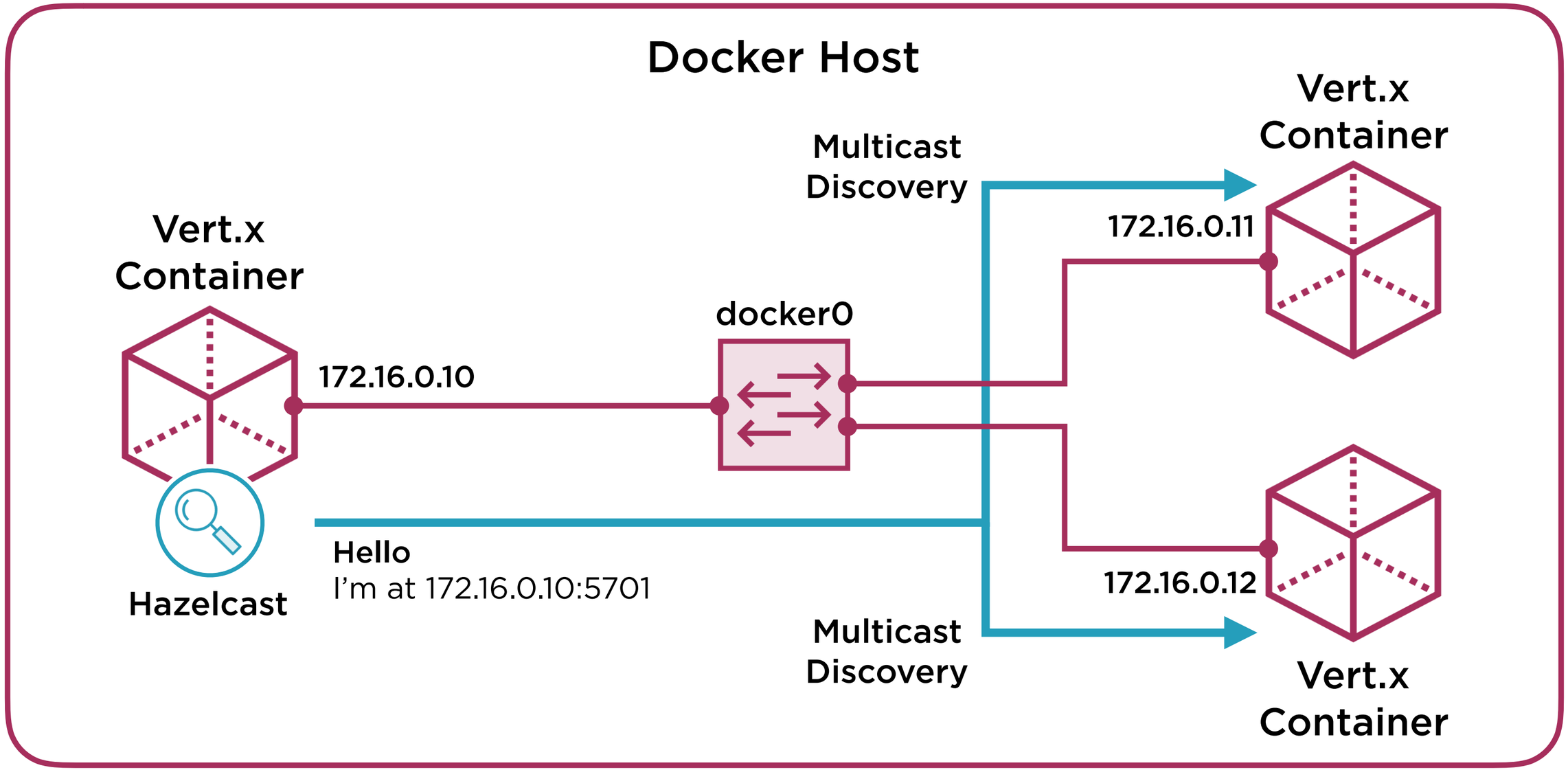


Creating a Packer Template

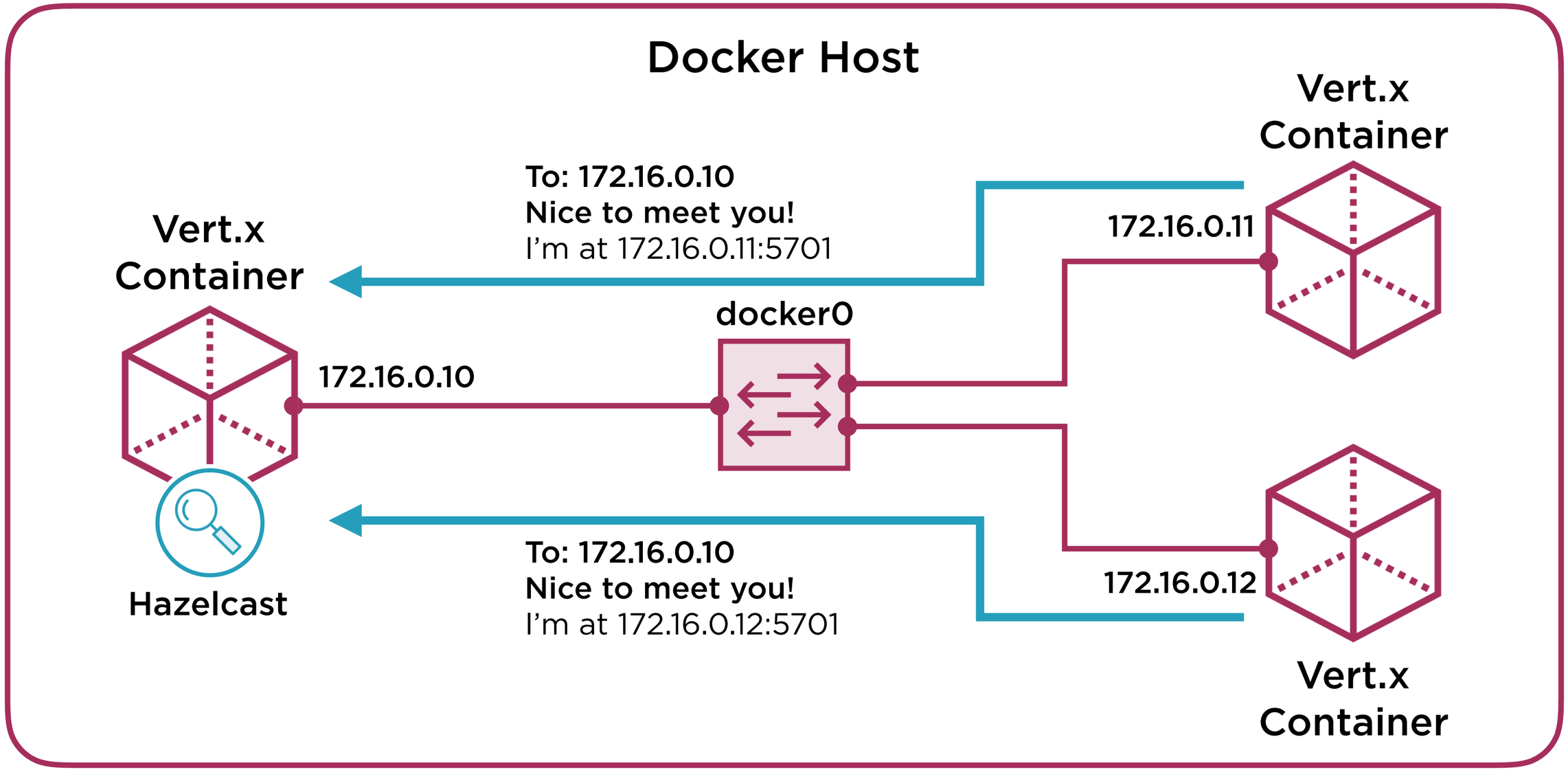
Adding Packer Provisioning Tasks

Customizing Docker

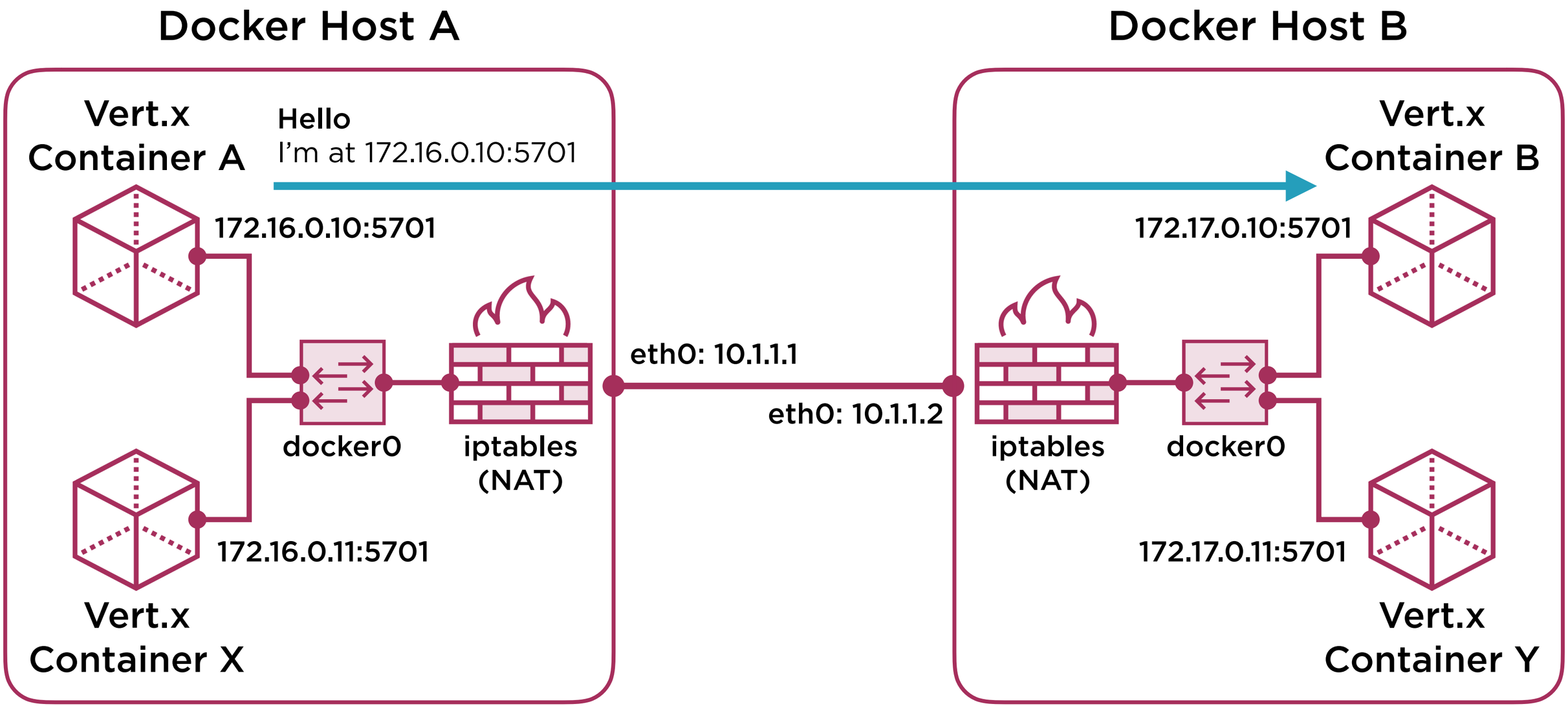
Vert.x Cluster Discovery



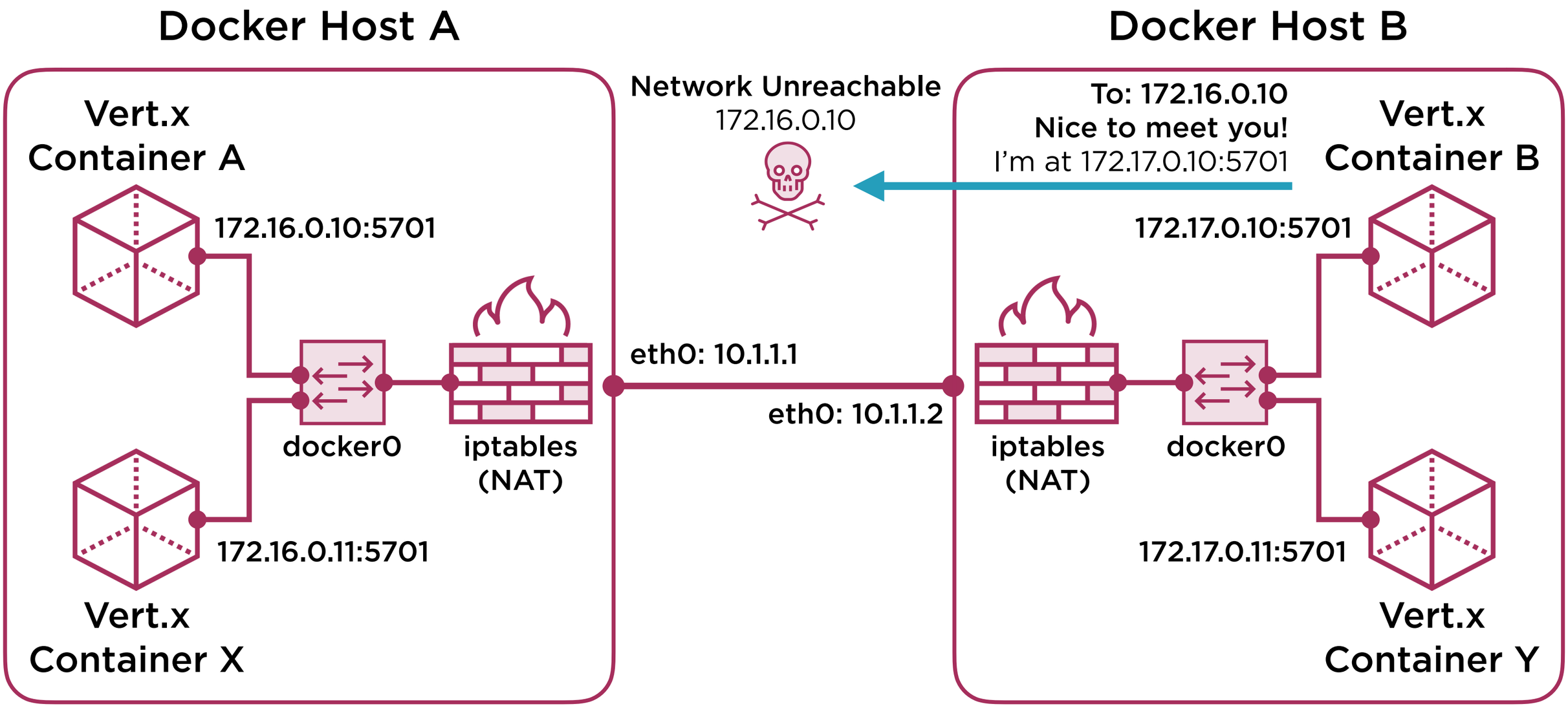
Vert.x Cluster Discovery



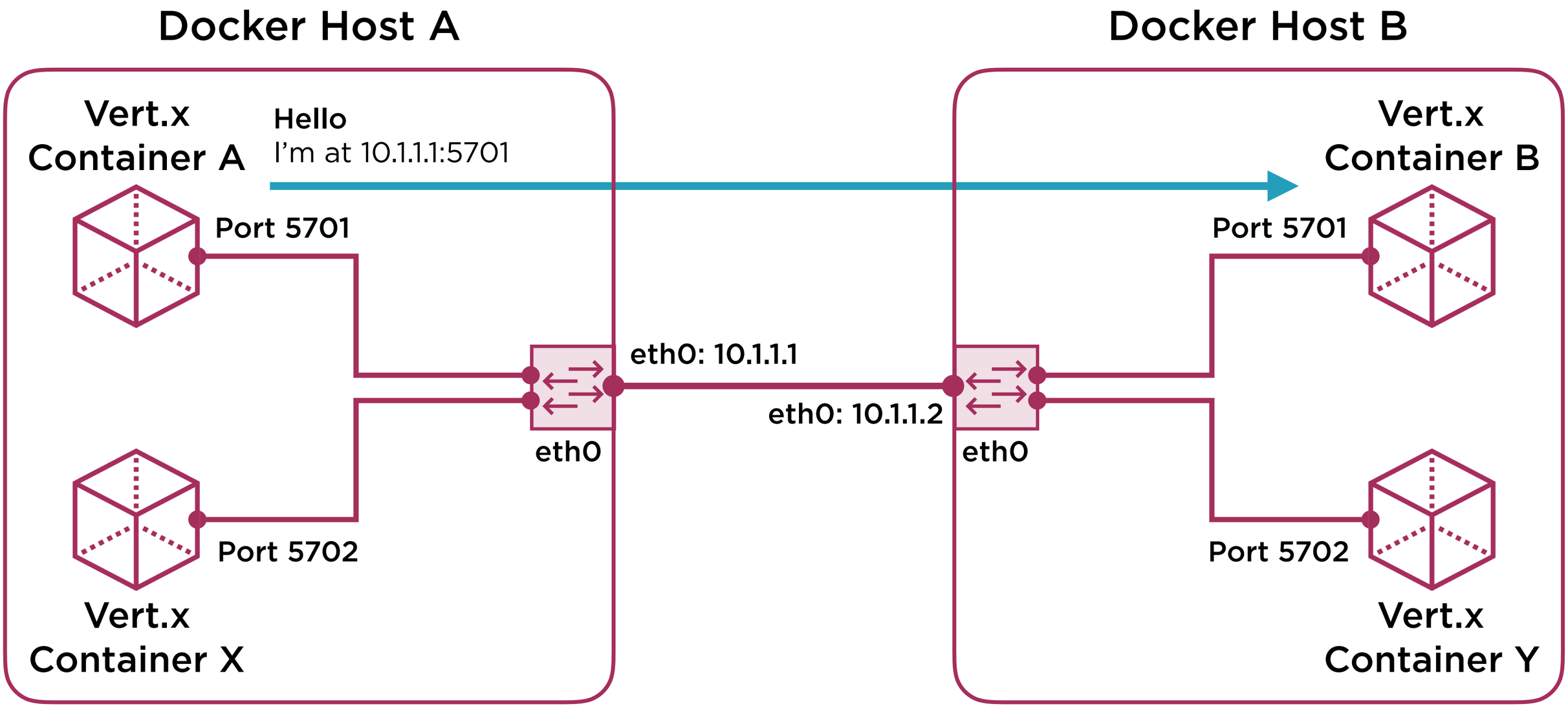
Network Address Translation Challenges



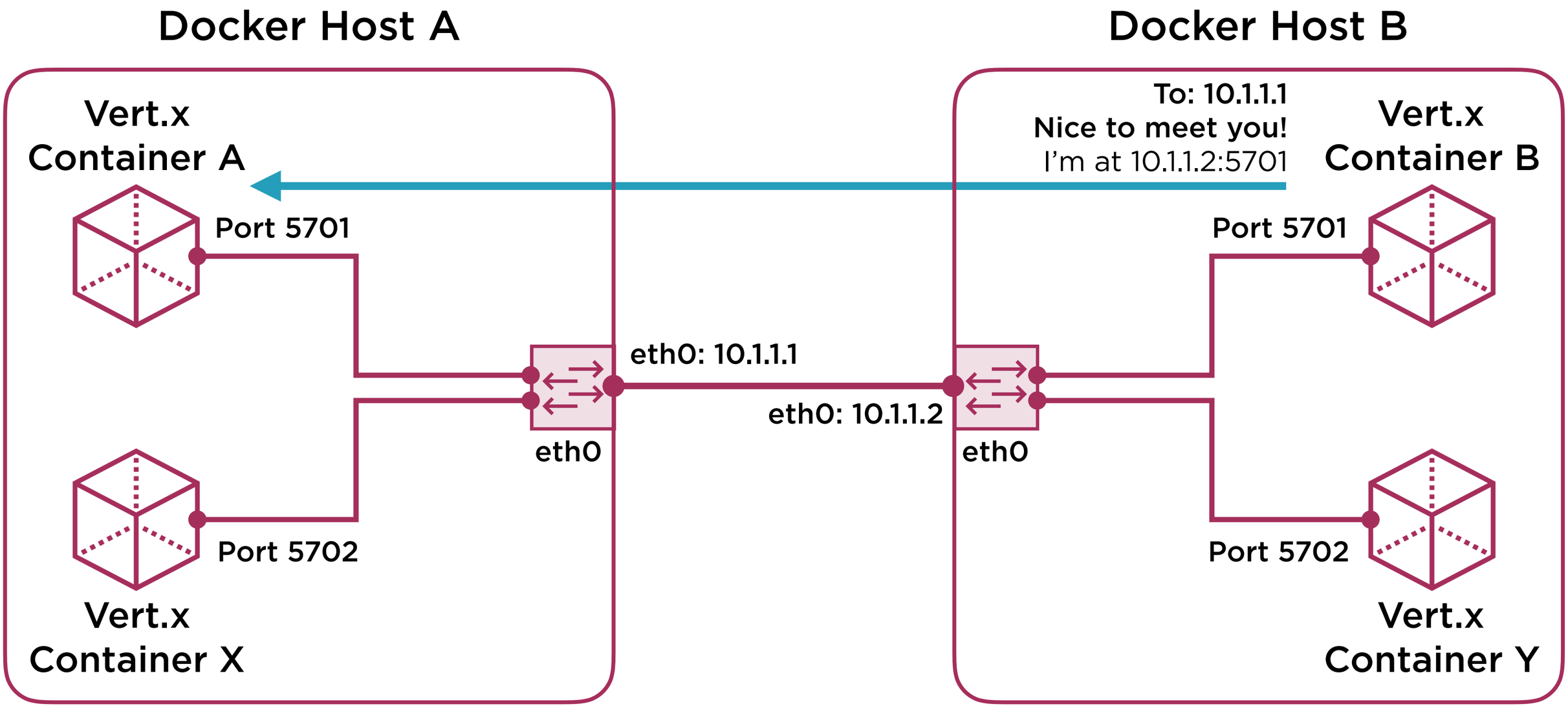
Network Address Translation Challenges



Docker Host Networking



Docker Host Networking



Docker Host Networking

Pros

No network address translation issues

Better network performance

Works well with dedicated Docker hosts per application

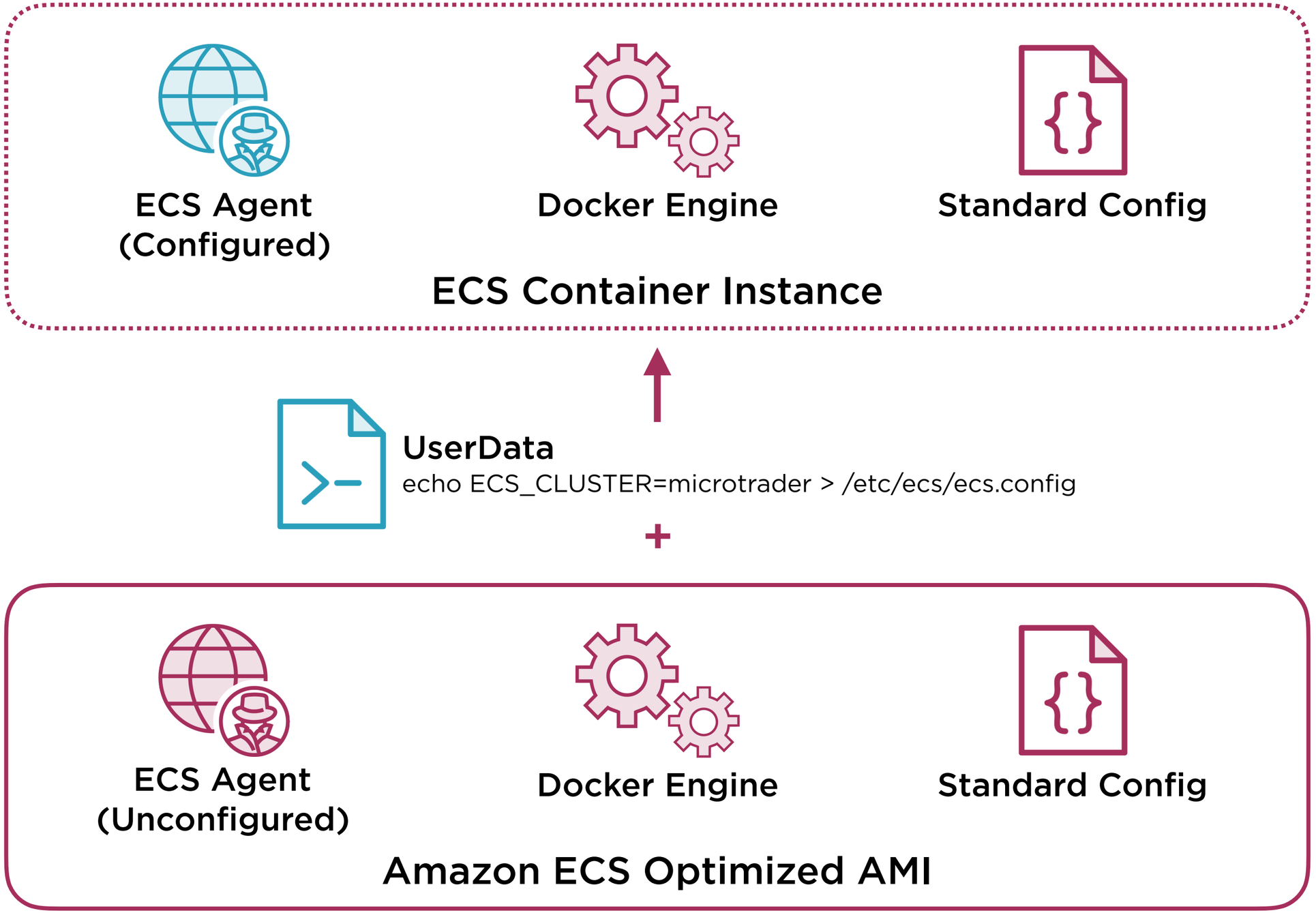
Cons

No support for user namespaces

Shared address space for TCP/UDP ports

Does not work well with shared Docker hosts running lots of different applications

Configuring the ECS Agent


ECS Agent
(Configured)
Docker Engine
Standard Config
ECS Container Instance
UserData
echo ECS_CLUSTER=microtrader > /etc/ecs/ecs.config
+
ECS Agent
(Unconfigured)
Docker Engine
Standard Config
Amazon ECS Optimized AMI

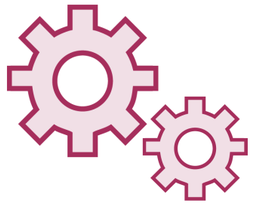
Custom AMI

CloudWatch Logs Agent

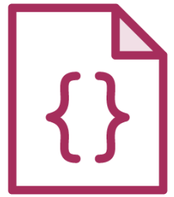
First Run Script
- HTTP Proxy Support
- ECS Agent Config
- CloudWatch Logs Config
- Health Check

Custom Config
- Timezone
- Enable NTP
- Custom Docker Config

ECS Agent



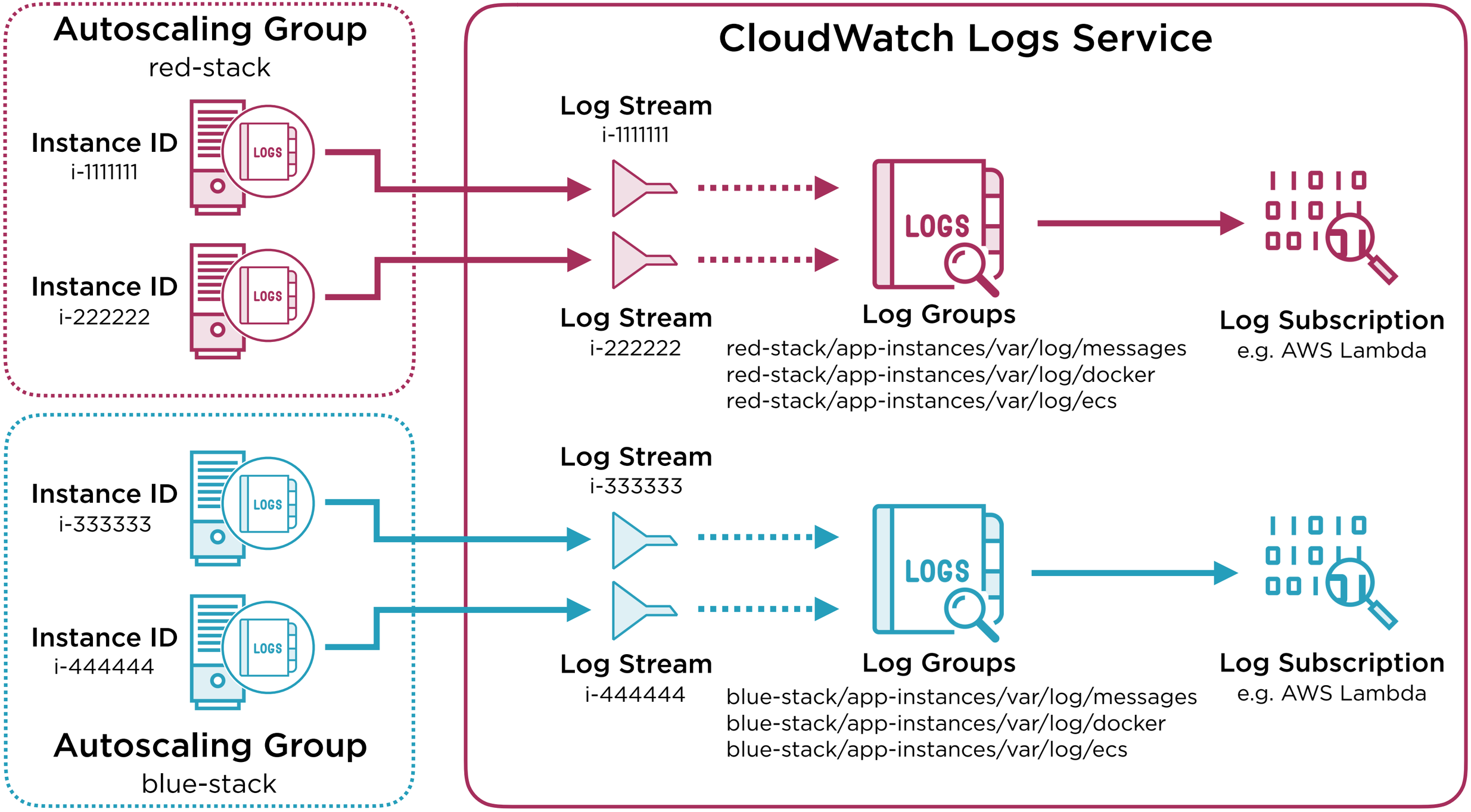
Docker Engine



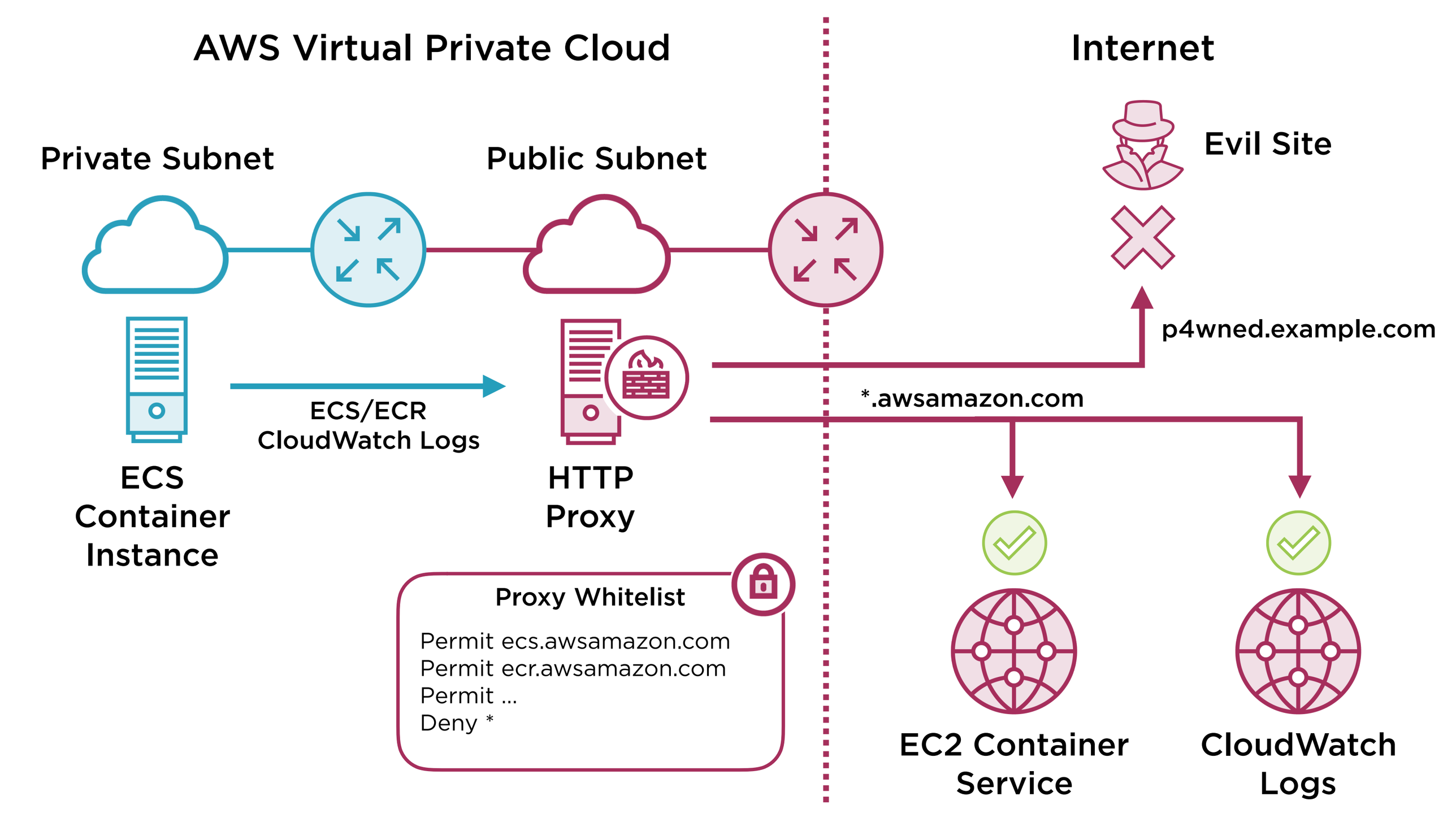
Base Config

Amazon ECS Optimized AMI

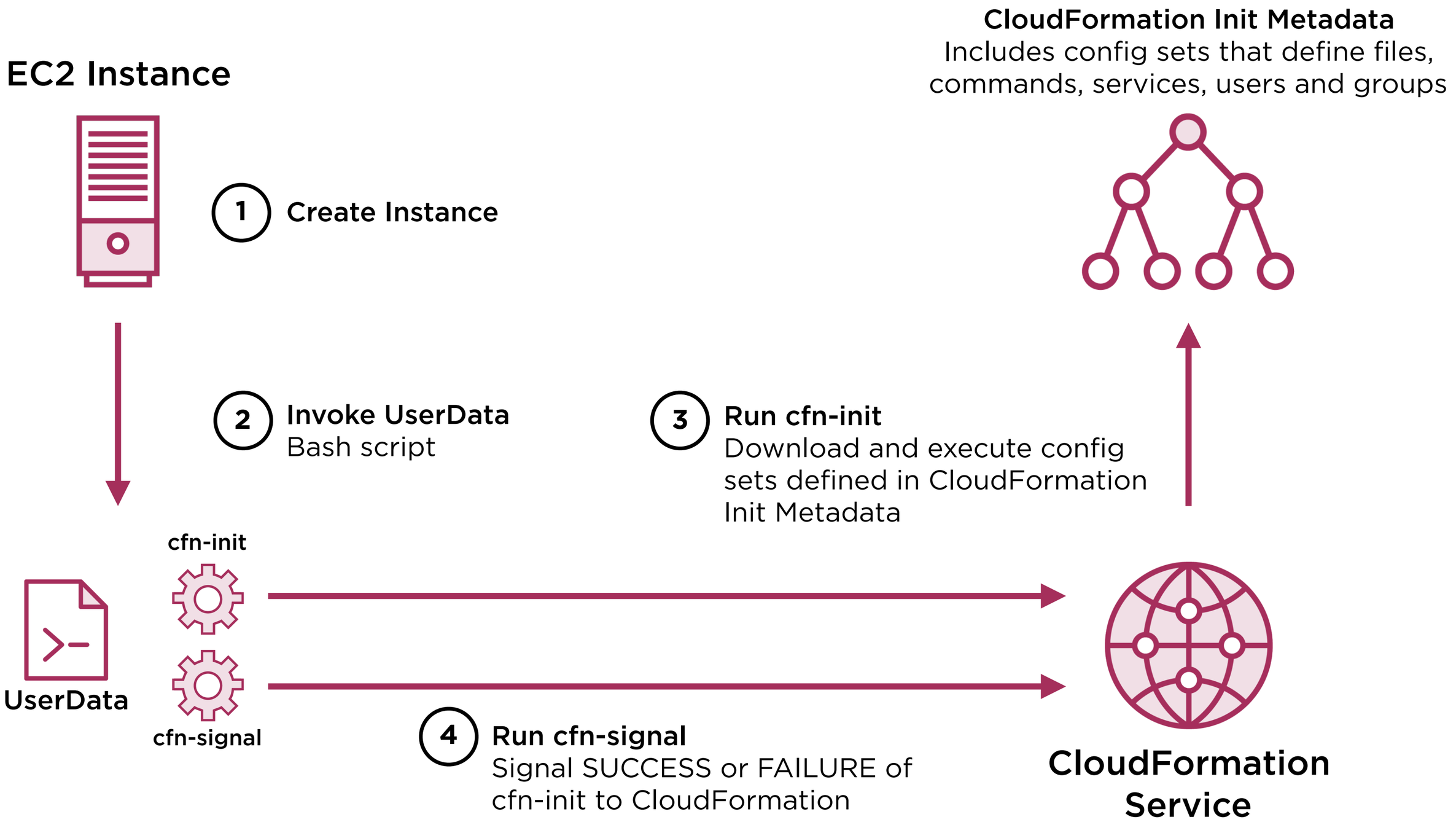
CloudWatch Logs Integration


Autoscaling Group
red-stack
Instance ID
i-1111111
Instance ID
i-222222
LOGS
LOGS
CloudWatch Logs Service
Log Stream
i-1111111
Log Stream
i-222222
LOGS
Log Groups
red-stack/app-instances/var/log/messages
red-stack/app-instances/var/log/docker
red-stack/app-instances/var/log/ecs
Log Subscription
e.g. AWS Lambda
Log Stream
i-333333
Log Stream
i-444444
LOGS
Log Groups
blue-stack/app-instances/var/log/messages
blue-stack/app-instances/var/log/docker
blue-stack/app-instances/var/log/ecs
Log Subscription
e.g. AWS Lambda
Instance ID
i-333333
Instance ID
i-444444
LOGS
LOGS
Autoscaling Group
blue-stack

HTTP Proxy Support


AWS Virtual Private Cloud
Internet
Private Subnet
Public Subnet
Evil Site
p4wned.example.com
*.awsamazon.com
ECS/ECR
CloudWatch Logs
ECS
Container
Instance
HTTP
Proxy
Proxy Whitelist
Permit ecs.awsamazon.com
Permit ecr.awsamazon.com
Permit ...
Deny *
EC2 Container
Service
CloudWatch
Logs

ECS Container Instance Health Checks


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

Post Build Cleanup

Building and Publishing the Image

Summary

Customizing ECS Container Instances

- Packer
- Build time changes
- First run script
- Docker customizations
- CloudWatch Logs
- HTTP Proxy support
- Health checks
- Building and publishing the AMI