

AWS
re:Invent

C O N 3 3 4

Operations for Containerized Applications

Tiffany Jernigan
[@tiffanyfayj](#)

Developer Advocate
Amazon Web Services

Nathan Peck
[@nathankpeck](#)

Developer Advocate
Amazon Web Services

Session Times

Monday, November 26

Operations for Containerized Applications

1:00 PM | Bellagio, Level 1, Grand Ballroom 1

Tuesday, November 27

Operations for Containerized Applications

3:15 PM | Mirage, St. Thomas B

Agenda

Automation: Deployments

Security

Observability

Automation: Scaling

Minimizing operational overhead

Example architecture

AWS native container stack



Amazon ECR

IMAGE REGISTRY

Stores your docker container right there in the datacenter where you will run it



Amazon ECS



MANAGEMENT

The API interface you use to launch applications
Tracks application state and connects application to other resources like load balancers



AWS Fargate

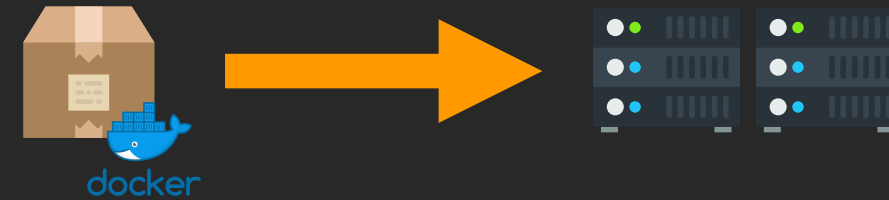
HOSTING

Containers run on demand
No capacity planning needed
Automatically updated and patched infrastructure

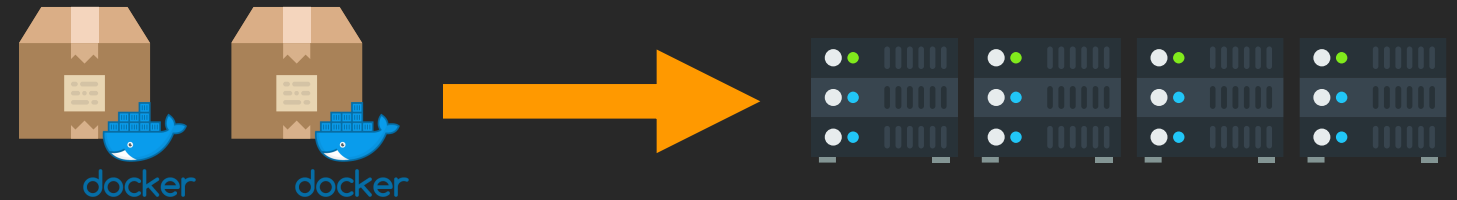
Automation: Deployments

Where are you on path of container adoption?

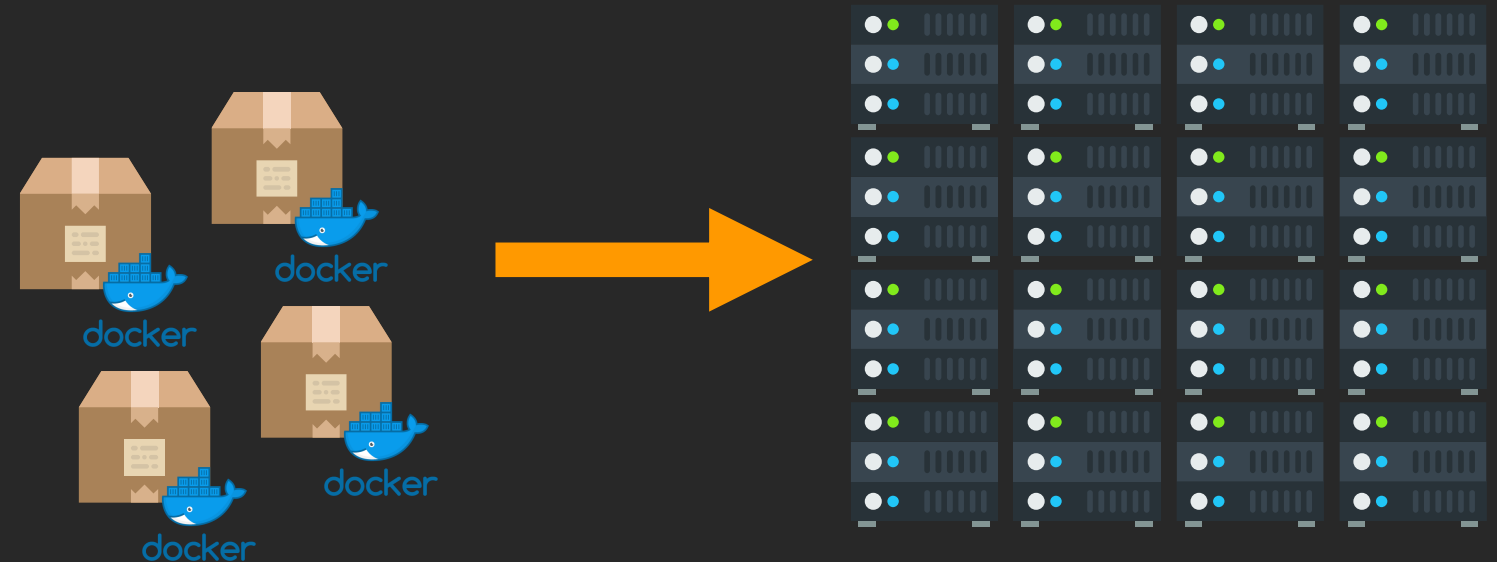
One app on a couple instances



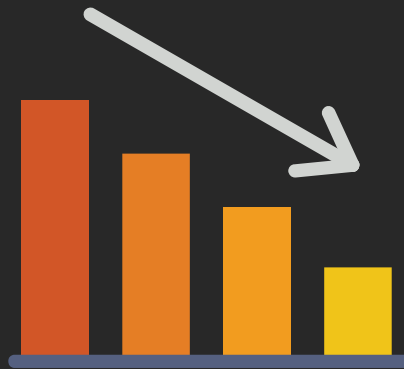
A couple apps on a few instances



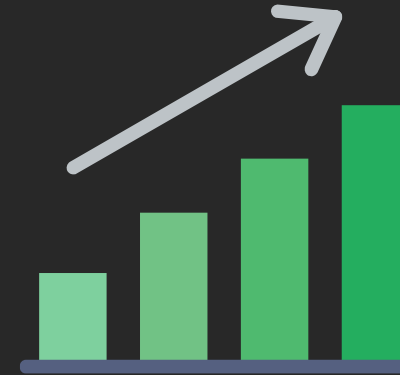
Many apps in a large cluster



Two paths... two results



Manual setup, hand rolled deploys
Ever growing burden of overhead
That engineer who knew how
everything worked just left the
company and we don't know how to
do a deploy

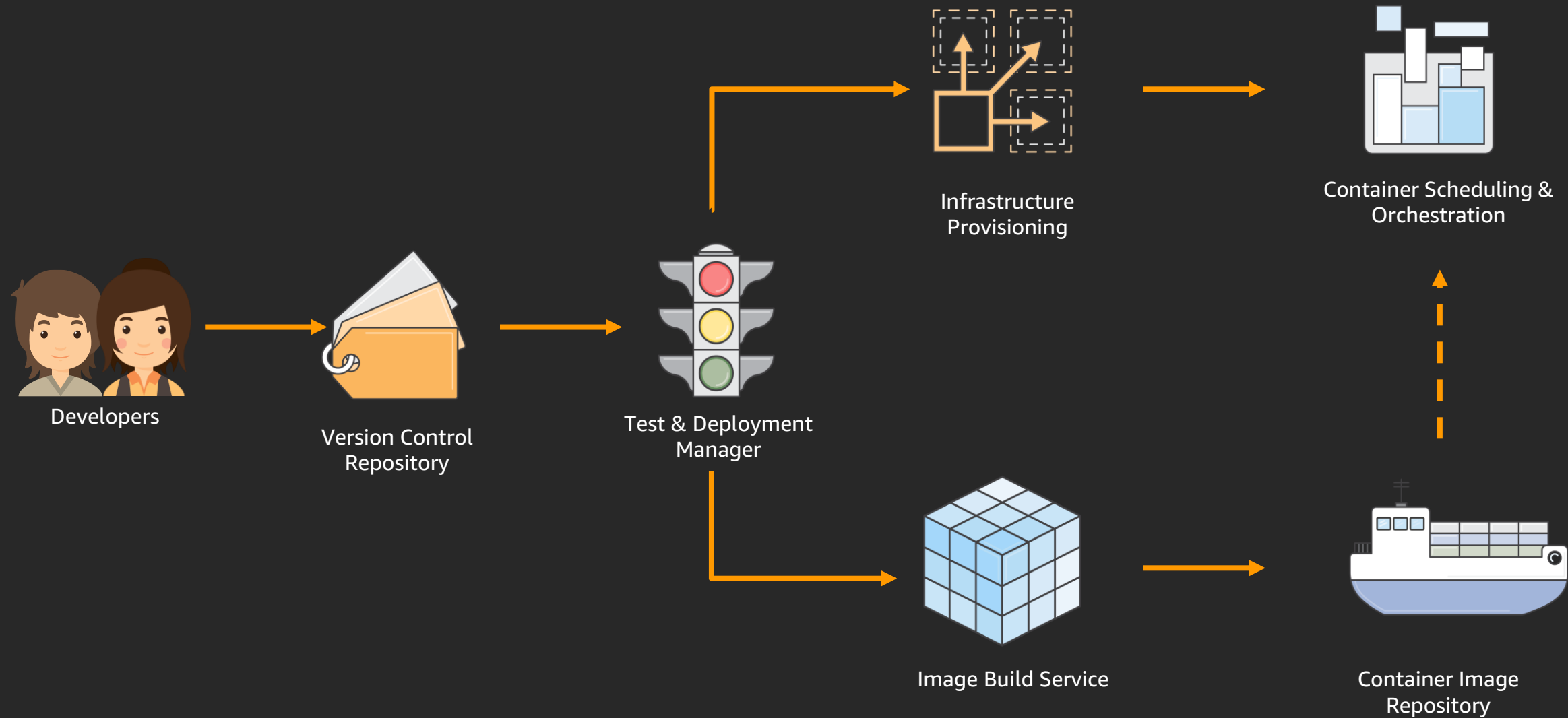


Automate all the things
Each piece automated increases
velocity
All operation processes clearly
defined by automation code and
infrastructure as code templates

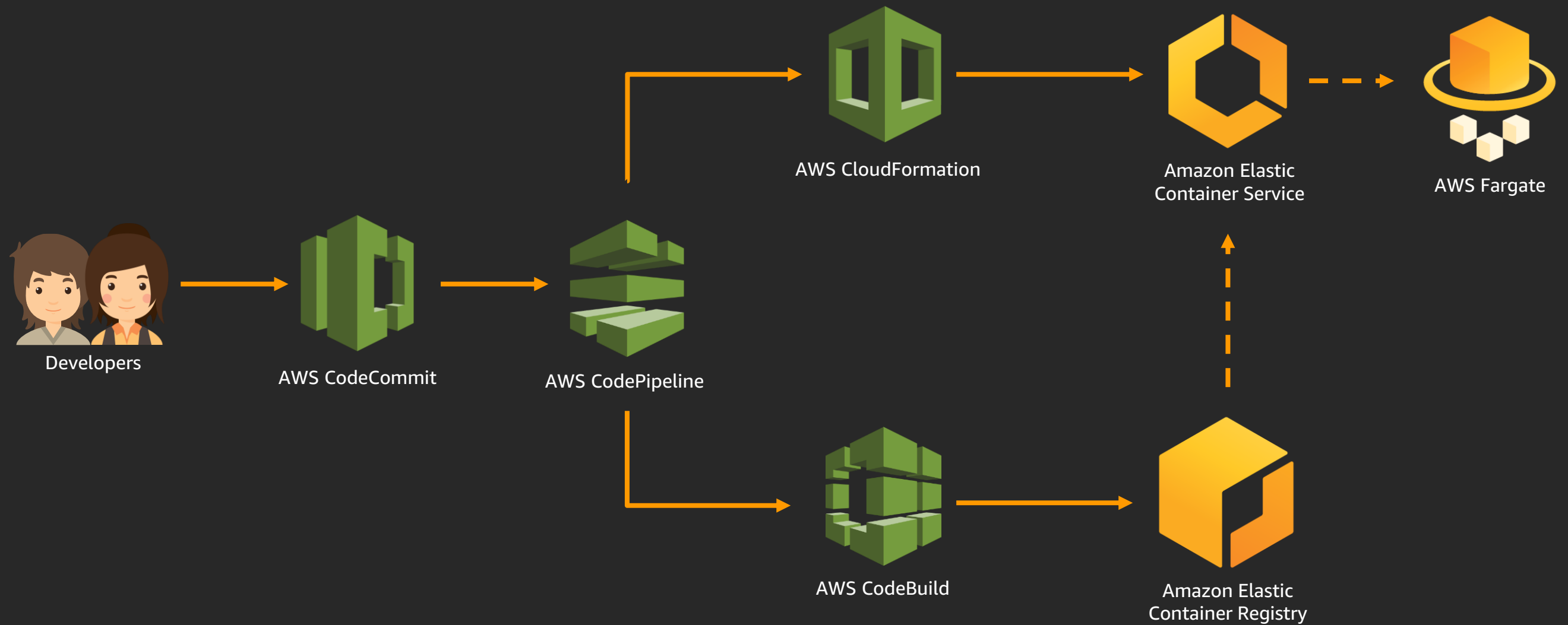
Effective engineering teams use deployment automation tooling



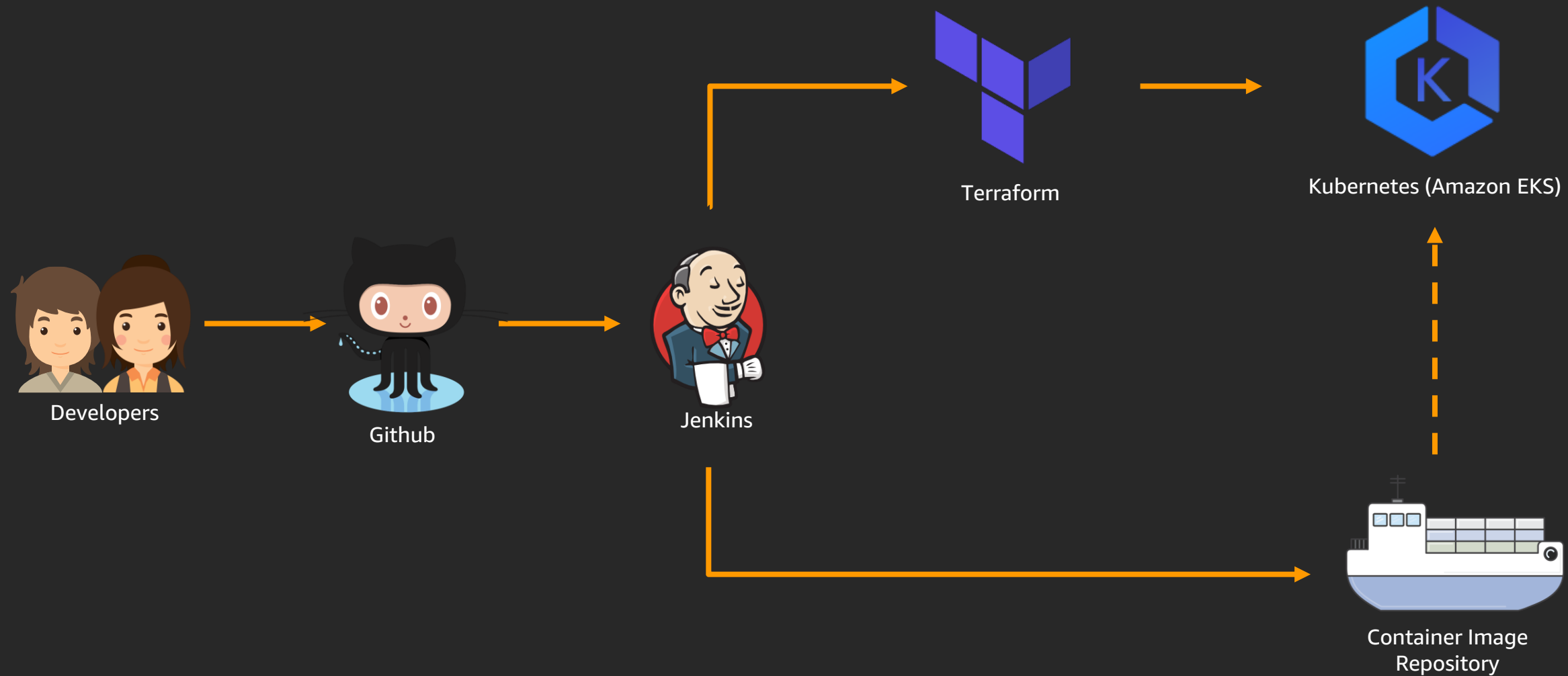
Components of effective container operations



The AWS native stack



An open source stack



Security

Networking

VPC

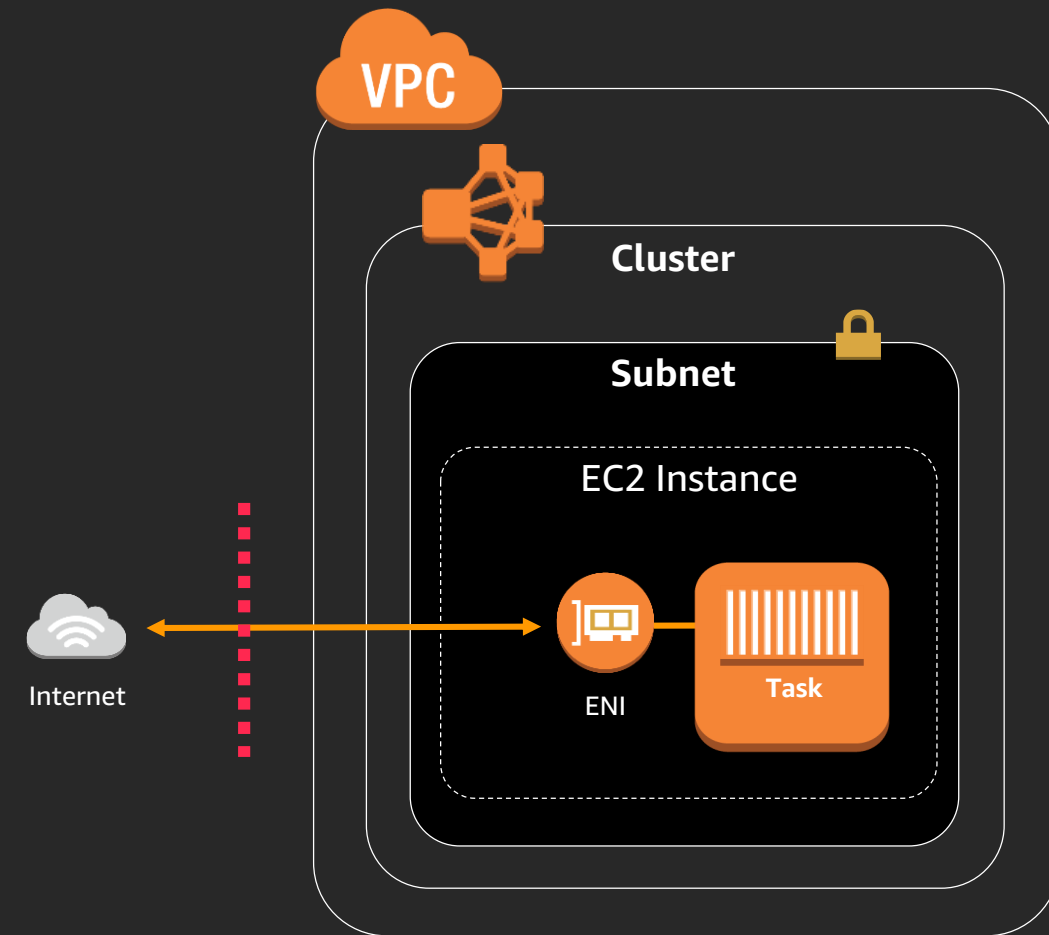
Subnets

Networking mode

Amazon Virtual Private Cloud (Amazon VPC): Each task gets its own interface

Security groups

Control inbound & outbound traffic



IAM

Instance (Amazon Elastic Compute Cloud (Amazon EC2 launch type))

Cluster

Control who can launch/describe tasks in your cluster

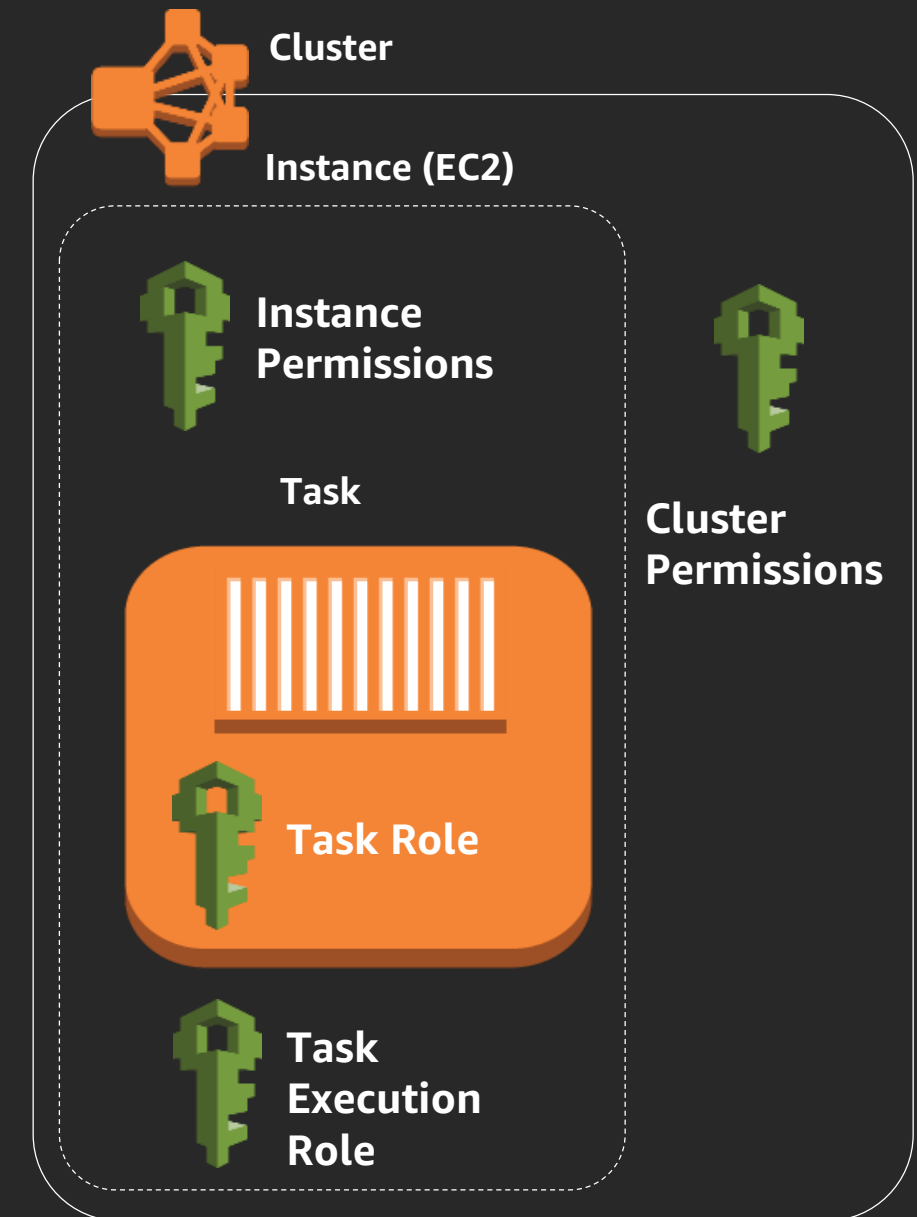
Application: Task Role

Allows your application containers to access AWS resources securely

Housekeeping: Task Execution Role

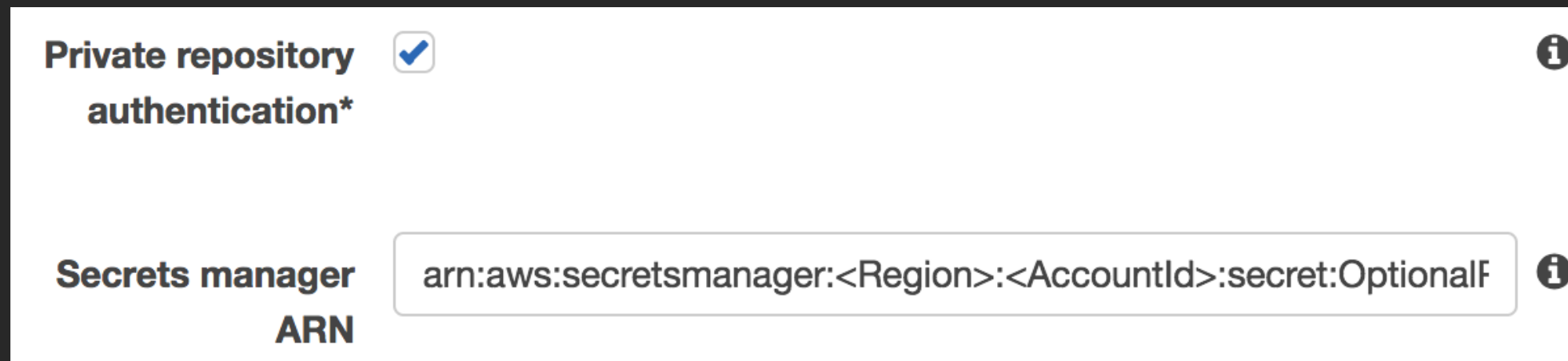
Allows ECS to perform housekeeping activities around your task:

- Private registry image pull
- Amazon CloudWatch Logs pushing (Fargate launch type)
- ENI creation (AWSVPC mode)
- Register/Deregister targets into Elastic Load Balancing (Fargate launch type)



Private Registry Authentication

- Used for 3rd party private registries
- Takes a secret in AWS Secrets Manager with registry username and password
- Task needs a task execution AWS Identity and Access Management (IAM) role with permissions to get the secret value



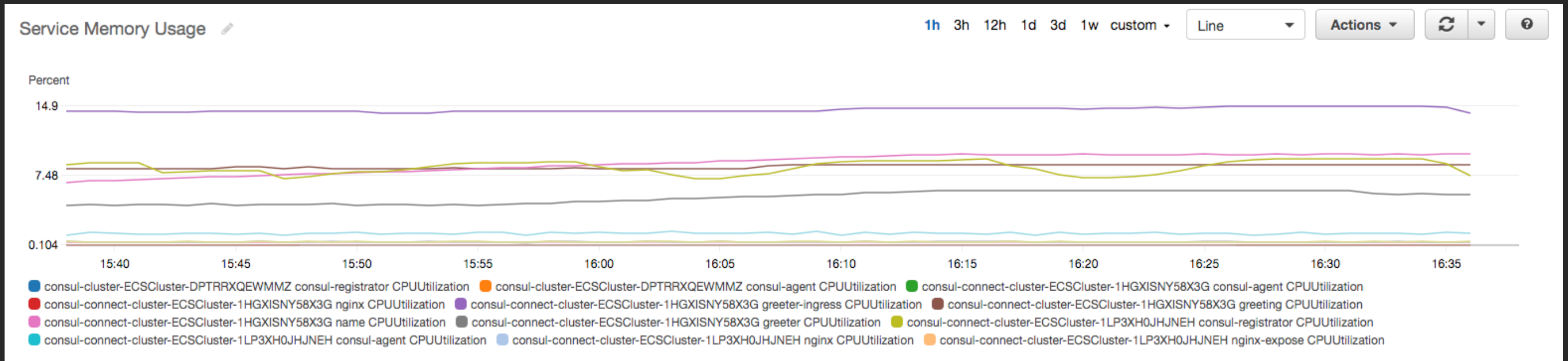
The screenshot shows a configuration panel for 'Private repository authentication*'. It includes a checked checkbox, an information icon, and a text input field for the 'Secrets manager ARN' containing the placeholder 'arn:aws:secretsmanager:<Region>:<AccountId>:secret:OptionalF'.

Private repository authentication* ☒ ⓘ

Secrets manager ARN ⓘ

Observability

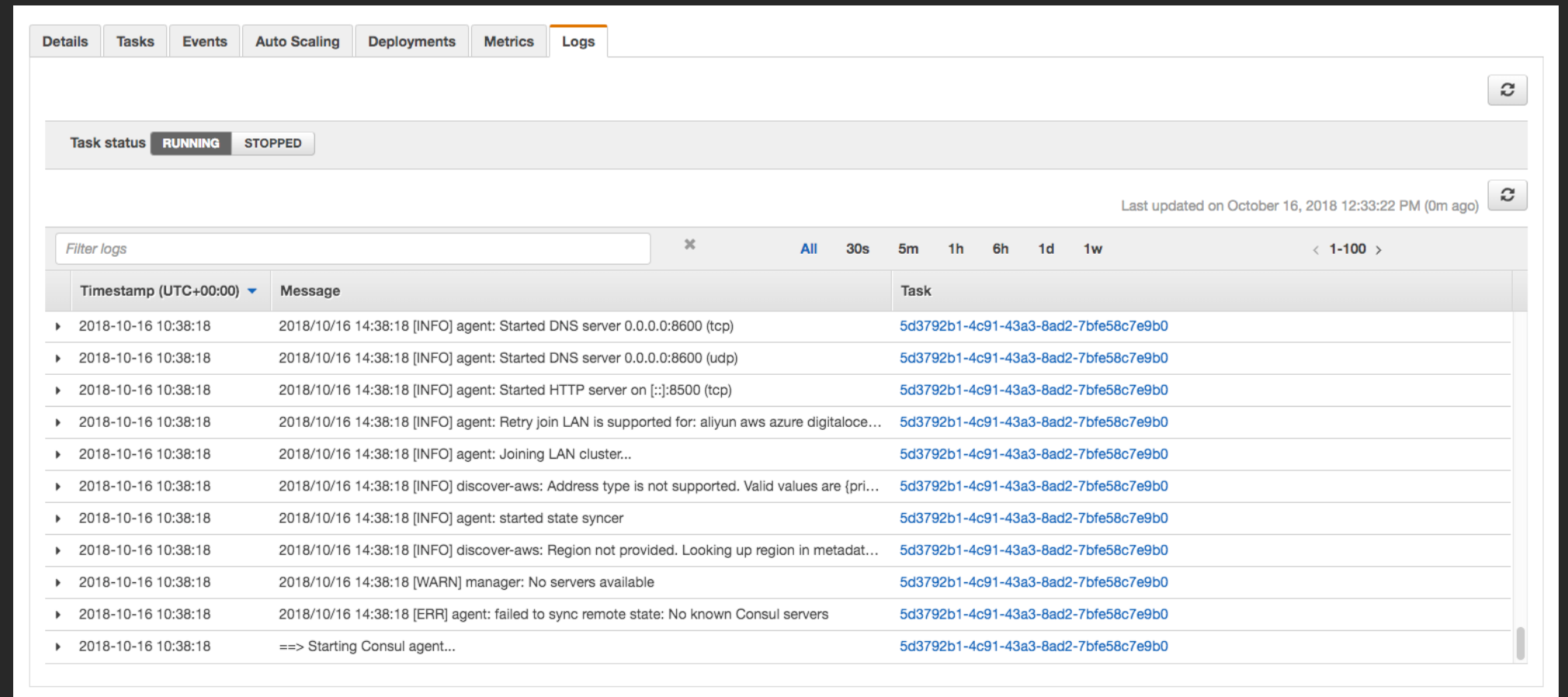
Metrics



Logs

Log integration is built in via the awslogs Docker log driver.

Logs automatically visible in the ECS console, and in Amazon CloudWatch logs



The screenshot shows the Amazon ECS console with the 'Logs' tab selected. The task status is 'RUNNING'. The logs are filtered by 'All' and show a list of messages with timestamps and task IDs. The messages include information about starting DNS and HTTP servers, joining a LAN cluster, and starting the Consul agent.

Timestamp (UTC+00:00)	Message	Task
2018-10-16 10:38:18	2018/10/16 14:38:18 [INFO] agent: Started DNS server 0.0.0.0:8600 (tcp)	5d3792b1-4c91-43a3-8ad2-7bfe58c7e9b0
2018-10-16 10:38:18	2018/10/16 14:38:18 [INFO] agent: Started DNS server 0.0.0.0:8600 (udp)	5d3792b1-4c91-43a3-8ad2-7bfe58c7e9b0
2018-10-16 10:38:18	2018/10/16 14:38:18 [INFO] agent: Started HTTP server on [::]:8500 (tcp)	5d3792b1-4c91-43a3-8ad2-7bfe58c7e9b0
2018-10-16 10:38:18	2018/10/16 14:38:18 [INFO] agent: Retry join LAN is supported for: aliyun aws azure digitaloce...	5d3792b1-4c91-43a3-8ad2-7bfe58c7e9b0
2018-10-16 10:38:18	2018/10/16 14:38:18 [INFO] agent: Joining LAN cluster...	5d3792b1-4c91-43a3-8ad2-7bfe58c7e9b0
2018-10-16 10:38:18	2018/10/16 14:38:18 [INFO] discover-aws: Address type is not supported. Valid values are {pri...	5d3792b1-4c91-43a3-8ad2-7bfe58c7e9b0
2018-10-16 10:38:18	2018/10/16 14:38:18 [INFO] agent: started state syncer	5d3792b1-4c91-43a3-8ad2-7bfe58c7e9b0
2018-10-16 10:38:18	2018/10/16 14:38:18 [INFO] discover-aws: Region not provided. Looking up region in metadat...	5d3792b1-4c91-43a3-8ad2-7bfe58c7e9b0
2018-10-16 10:38:18	2018/10/16 14:38:18 [WARN] manager: No servers available	5d3792b1-4c91-43a3-8ad2-7bfe58c7e9b0
2018-10-16 10:38:18	2018/10/16 14:38:18 [ERR] agent: failed to sync remote state: No known Consul servers	5d3792b1-4c91-43a3-8ad2-7bfe58c7e9b0
2018-10-16 10:38:18	==> Starting Consul agent...	5d3792b1-4c91-43a3-8ad2-7bfe58c7e9b0



Audit Trail

Audit capability is built in with AWS CloudTrail

CloudTrail Events show who made what API calls, when.

View Event

```
"eventTime": "2018-10-04T18:39:14Z",
"eventSource": "elasticloadbalancing.amazonaws.com",
"eventName": "DescribeTargetHealth",
"awsRegion": "us-west-2",
"sourceIPAddress": "ecs.amazonaws.com",
"userAgent": "ecs.amazonaws.com",
"requestParameters": {
  "targetGroupArn": "arn:aws:elasticloadbalancing:us-west-2:012345678987:targetgroup/eko"
},
"responseElements": {
  "targetHealthDescriptions": [
    {
      "targetHealth": {
        "state": "healthy"
      },
      "target": {
        "id": "10.0.0.127",
        "port": 80,
        "availabilityZone": "us-west-2a"
      },
      "healthCheckPort": "80"
    }
  ]
}
}
```

Event time	User name	Event name
▶ 2018-10-04, 11:39:14 AM	ecs-service-scheduler	DescribeTargetHealth
▶ 2018-10-04, 11:39:14 AM	tiffany	DescribeNetworkInterfaces
▶ 2018-10-04, 11:39:14 AM	ecs-service-scheduler	GetInstancesHealthStatus
▶ 2018-10-04, 11:39:14 AM	tiffany	DescribeTasks
▶ 2018-10-04, 11:39:14 AM	tiffany	DescribeTaskDefinition
▶ 2018-10-04, 11:39:12 AM	tiffany	DescribeServices
▶ 2018-10-04, 11:39:12 AM	tiffany	DescribeTasks
▶ 2018-10-04, 11:39:12 AM	tiffany	ListTasks

Event time	User name	Event name	Resource type	Resource name
▼ 2018-10-04, 11:39:14 AM	ecs-service-scheduler	DescribeTargetHealth		

AWS access key

ABCDEFGHIJKLMNQRST

AWS region

us-west-2

Error code

Event ID

184e1f93-880c-425c-a5e7-1e74ac4a8866

Event name

DescribeTargetHealth

Event source

elasticloadbalancing.amazonaws.com

Event time

2018-10-04, 11:39:14 AM

Request ID

caadeb9b-c804-11e8-8b42-4f52727b2706

Source IP address

ecs.amazonaws.com

User name

ecs-service-scheduler

Resources Referenced (0)

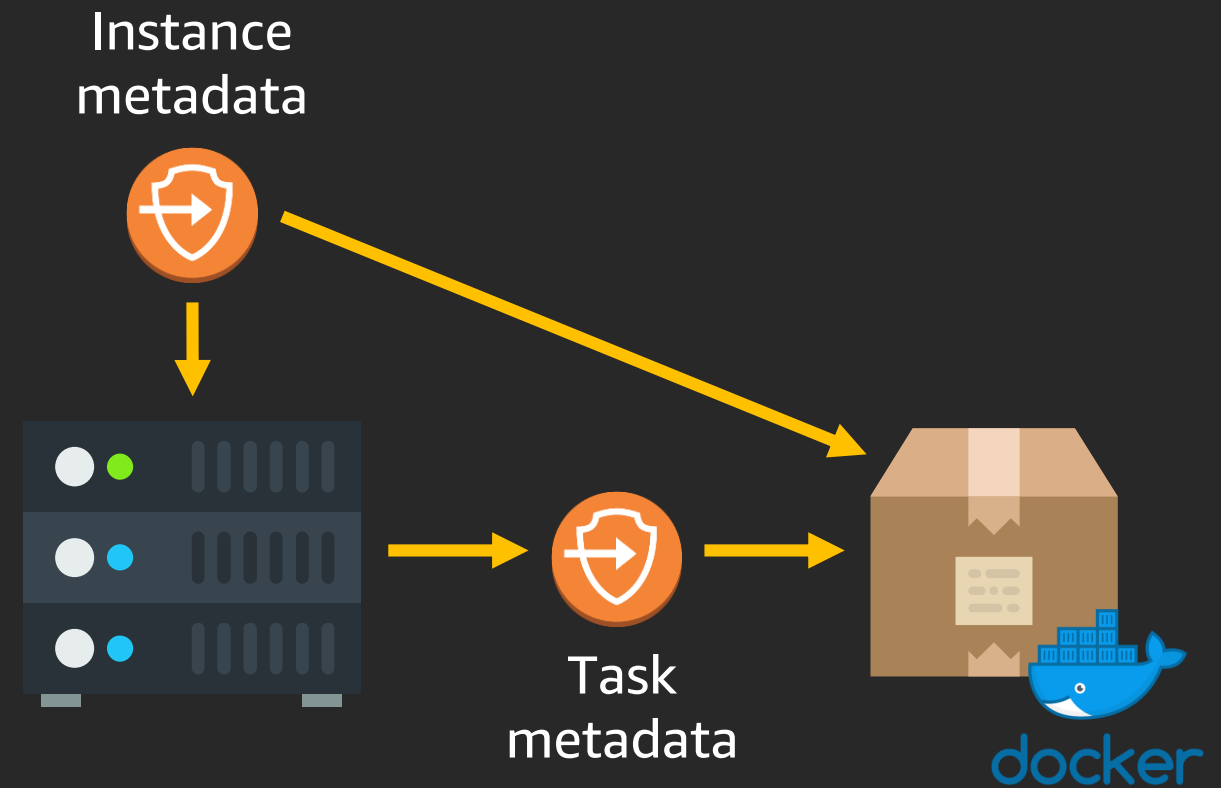
View event



Endpoints

Instance metadata endpoint gives your containers information about what's running on the instance.

Task metadata endpoint gives a container visibility into its own settings



Automation: Scaling

Automate service scaling



Service CPU Utilization

↑
Less than 20%
Decrease container
desired count by 1

You can define your own
custom rules and thresholds
for how to automatically scale
your service based on its
metrics. Custom metric
dimensions also supported.

↑
Greater than 85%
Increase container
desired count by 2

↑
Greater than 95%
Increase container
desired count by 3

Automate cluster scaling

Autoscaling group of EC2 instances



Scales according to metric

Cluster CPU

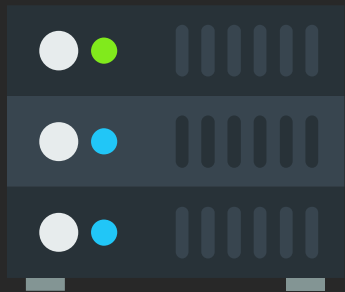
Custom metric



Minimizing Operational Overhead



AWS Fargate



No instances
to manage



docker

Task native API



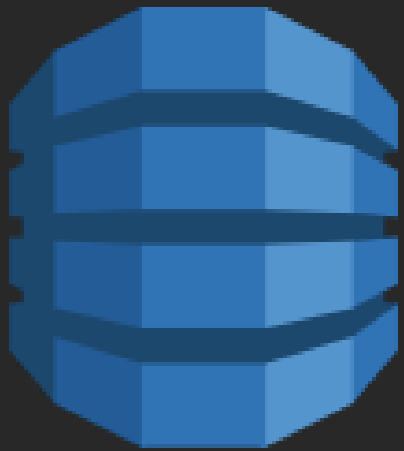
Resource
based pricing



Simple, easy to use,
powerful – and new
consumption model

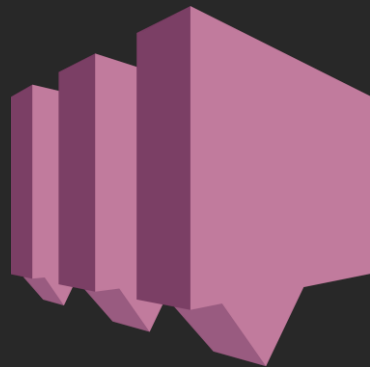
Cloud services "on tap" minimize overhead

Database



Amazon RDS
Amazon Aurora
Amazon DynamoDB

Messaging



Amazon Simple Queue
Service (Amazon SQS)



Amazon Simple
Notification Service
(Amazon SNS)

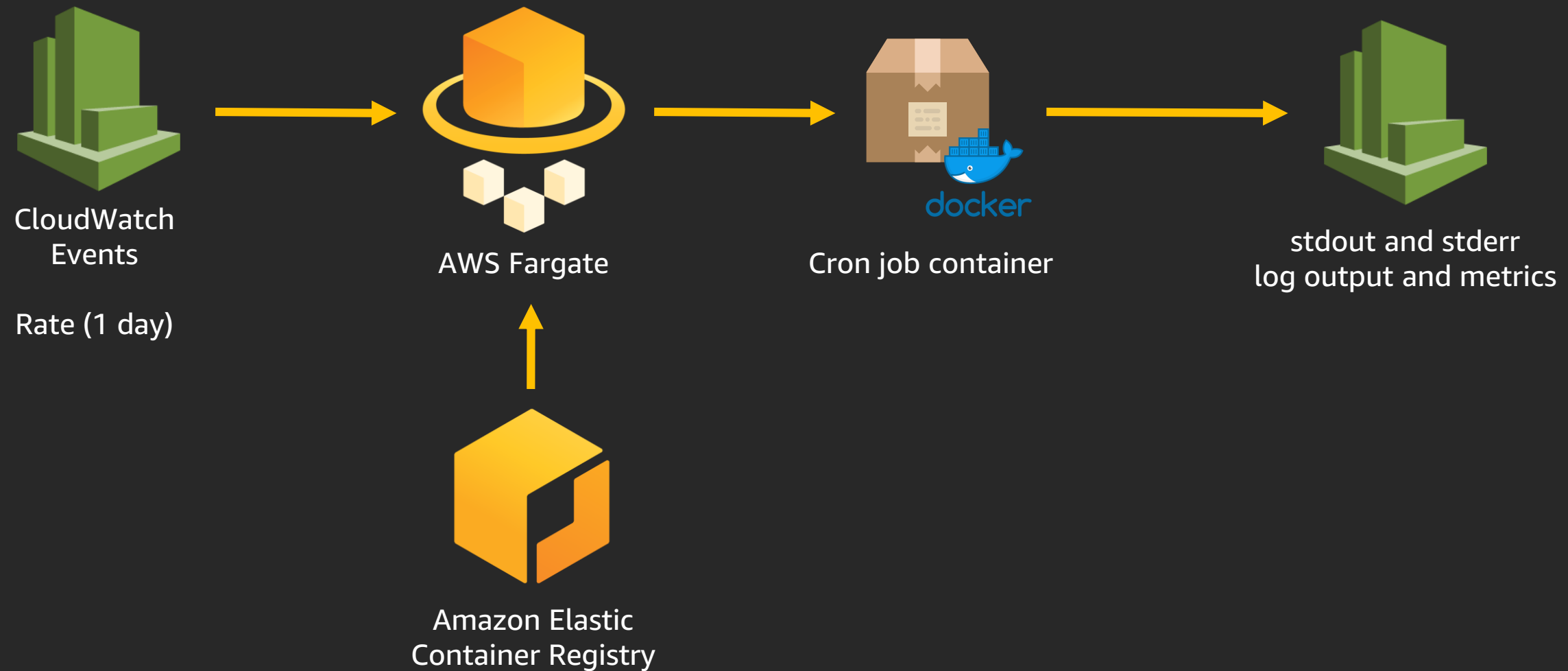
Storage



Amazon Simple
Storage Service
(Amazon S3)

Example Architecture

Serverless containerized cron job



Thank you!

@nathankpeck
@tiffanyfayj



Please complete the session
survey in the mobile app.