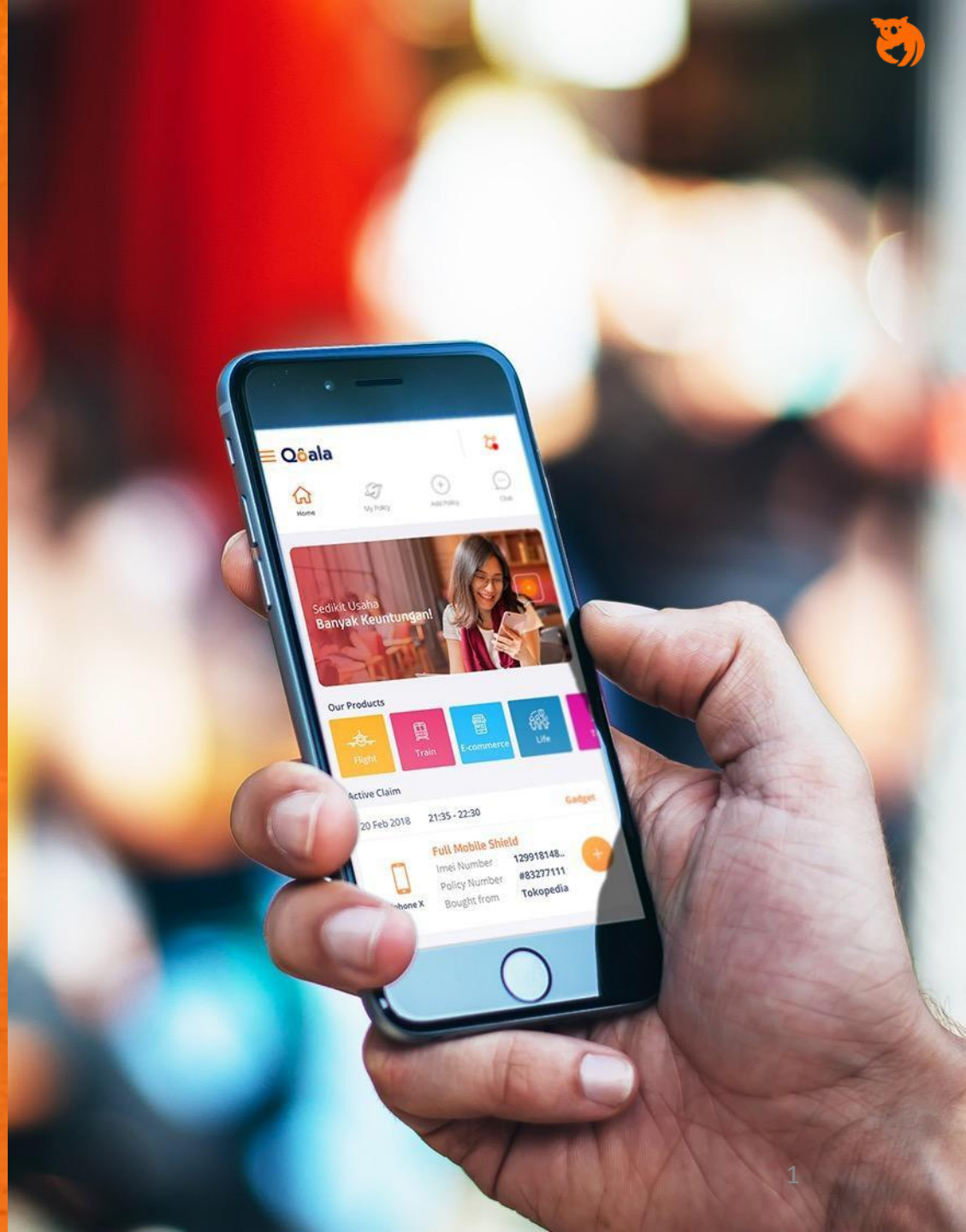
Running Containerized Applications in AWS ECS

Muhammad Sami

DevOps Community in Indonesia

Jakarta, 11 Desember 2019

Qoala x DevOpsDays Jakarta



Qoala

Qoala team and investors

Harshet Lunani
CEO

5-year experience in Insurtech industry (Co-Founder and CEO of BIMA Indonesia)

BIMA · k · BCG · Merrill Lynch

Tommy Martin
COO

Former Head of Marketing for Traveloka Malaysia and Singapore

traveloka · accenture · EY · NOKIA · EXCELLERATE

Martin Hong
CTO

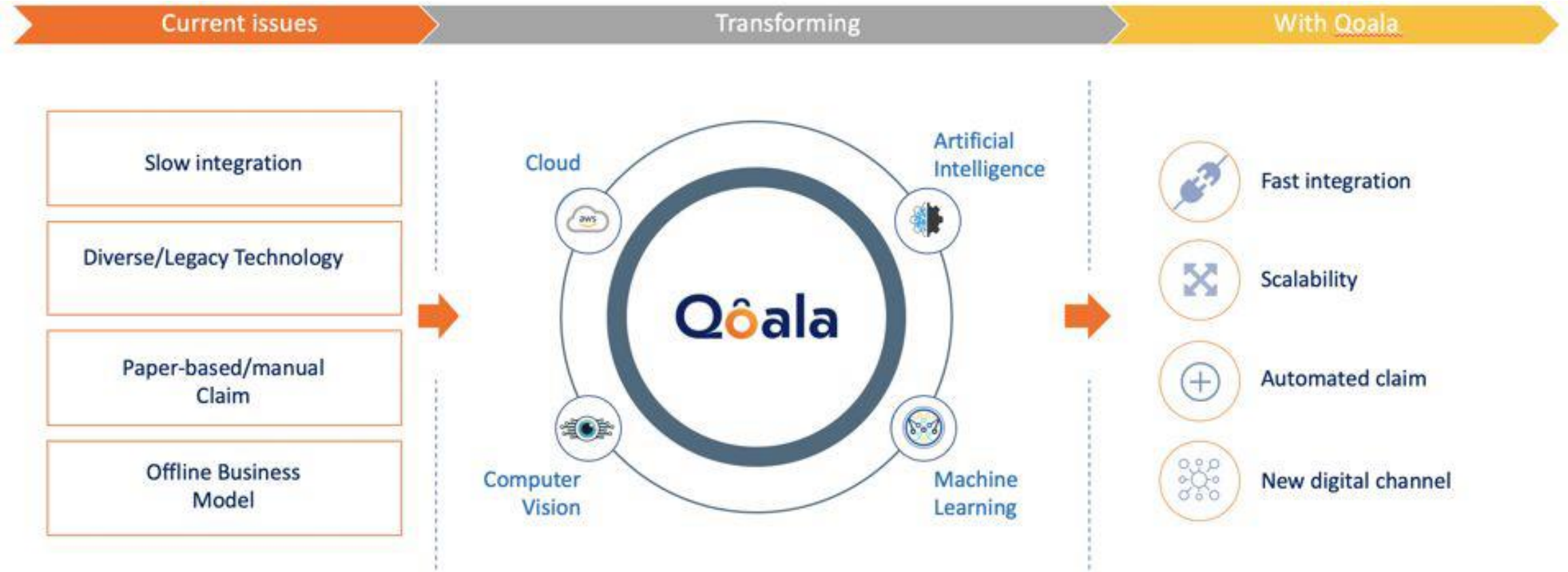
12 years in mobile & tech industry in UK, CTO for 2 previous startups

accenture · NOKIA · MFM · symbian OS

Investors

SEQUOIA · MassMutual · MDI METRA DIGITAL INOVASI by Telkom Indonesia · Golden Gate Ventures · central capital ventura · SEEDPLUS · Genesia Ventures

What are we solving..



Partners of Qoala

Travel

5-7 travel agents signed after launch

redBus, pegipegi, GOLDEN NUSA TRAVEL SERVICES, Aerotravel, MNC TRAVEL, HIS, Shopee

Ecommerce

10+ providers signed after launch

tokopedia, Grab, BUTIKDUKOMSEL.com, JD.ID Online Shopping, Sentra Ponsel, Firstindo, RCL Red Carpet Logistics, Finmas, TokoPDA.com, kargo, LOGISLY

Fintech

5 platforms signed after launch

investree, Akseleran, UangMe, MEKAR, GRADANA

Insurance Partners

AXA, TOKIO MARINE, ZURICH, Great Eastern, asuransi simasnet, ACA ASURANSI, BRIlife Proteksi Masa Depan Terpercaya, intra Asia insurance

www.qoala.id/karir-qoala/

Qoala

WE'RE **HIRING!**

- Software **Engineer Backend**
- Software **Engineer Frontend**
- Business **Development Manager**
- Sr Product **Manager**
- Recruiter & **People Partner**
- Business **Analyst**
- Associate Product **Manager**
- UI/UX Designer
- Content Lead

If you think the role is perfect for you, send your resume to ratu.laila@qoala.id.

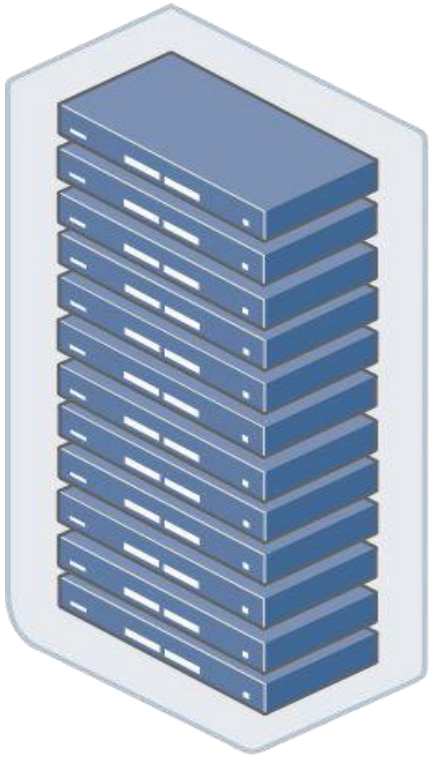
Speaker Profile - Muhammad Sami

Experiences:

Current:
Qoala, Senior Software Engineer (Nov 2018 - Present)

Past:
Activy, Mobile Application Developer (2017-2018)
Mediatrac/Dattabot, Software Engineer (2015-2017)

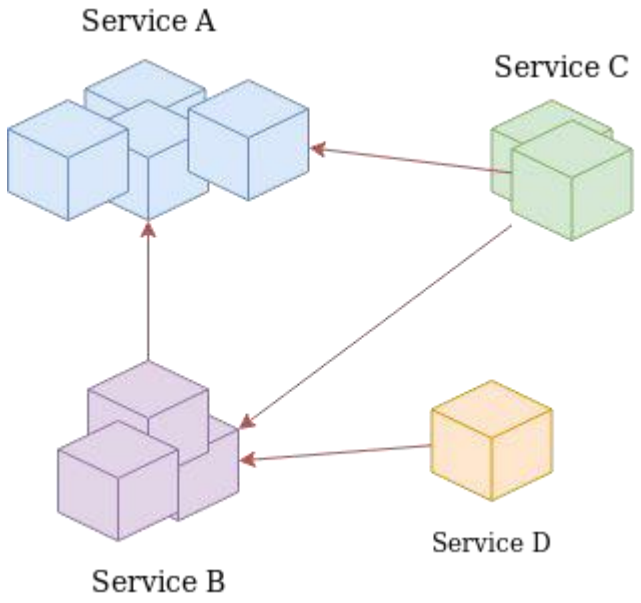
Applications are transforming



Monolith

Slow changes

Homogenous

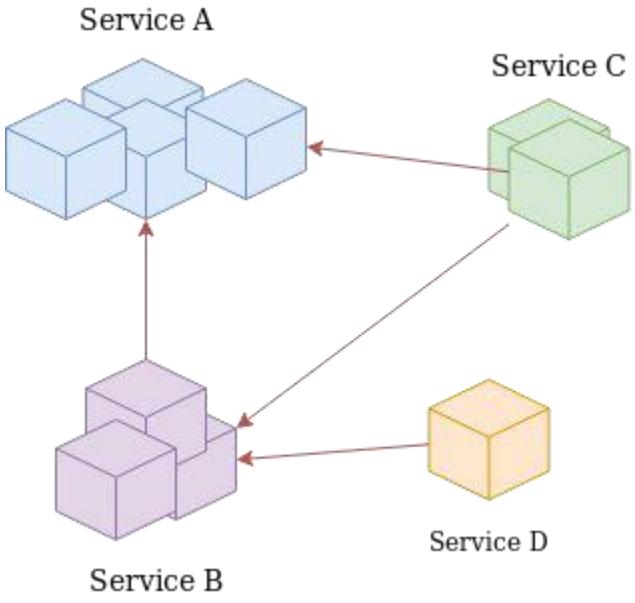


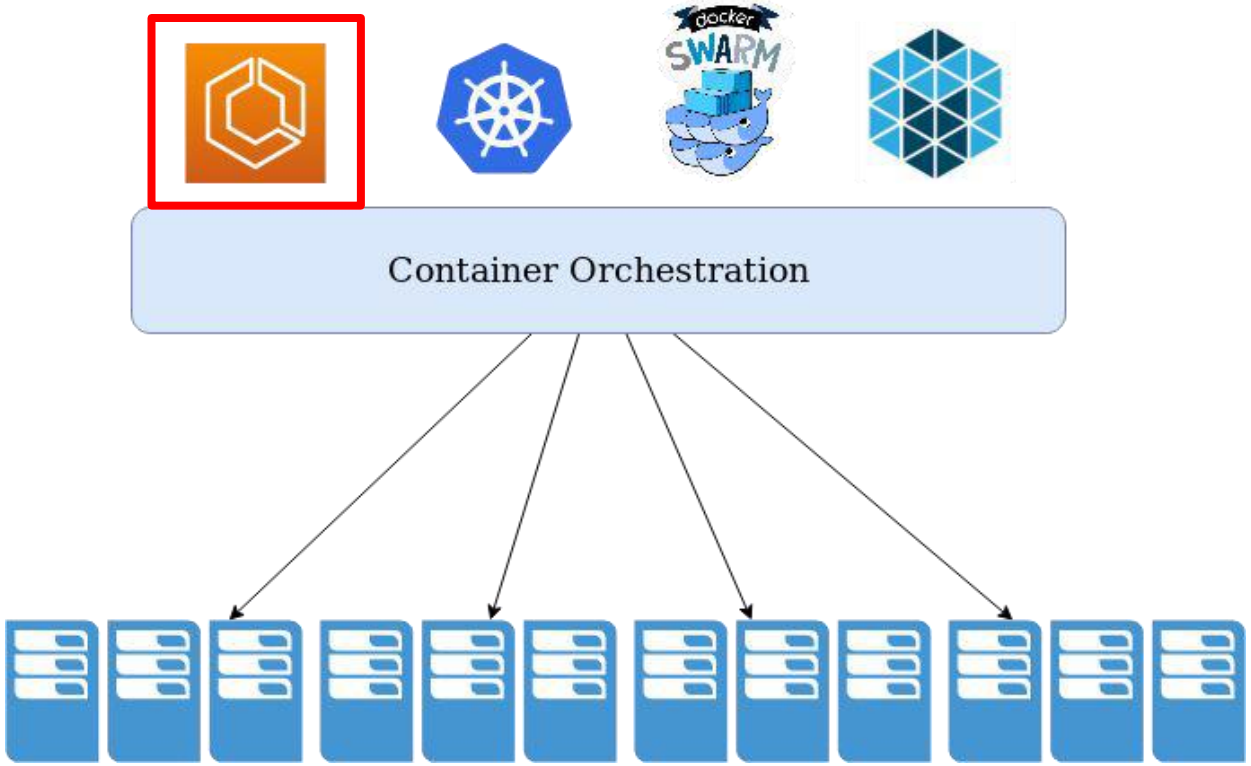
Loosely coupled

Rapid changes

Heterogenous

We have to manage this somehow



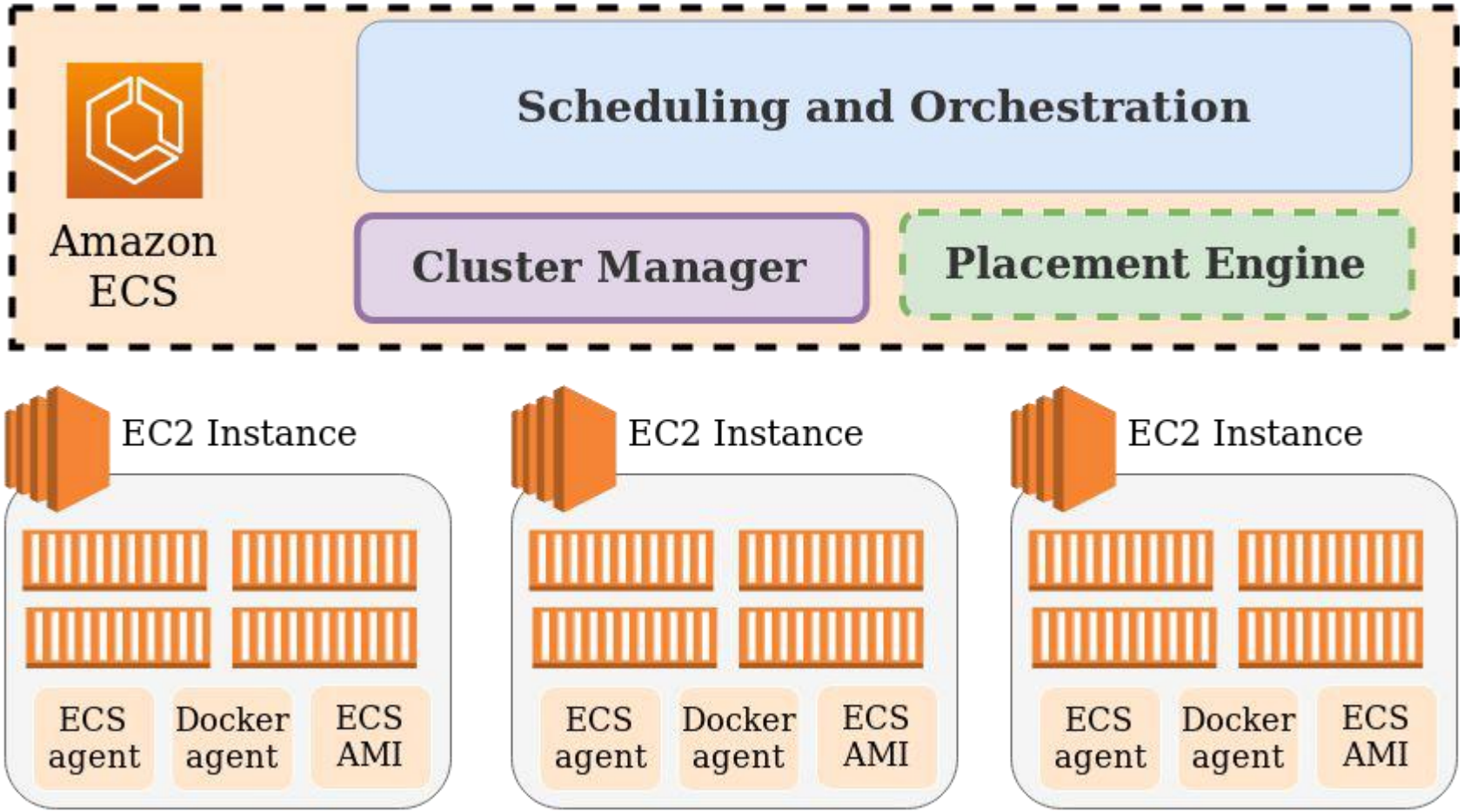


Amazon ECS

Amazon Elastic Container Service (Amazon ECS) is a fully managed container orchestration service

ECS supports *Fargate* to provide serverless compute for containers. Fargate removes the need to provision and manage servers.

Scales to support clusters of any size, and pay for what you use


Amazon ECS
Scheduling and Orchestration
Cluster Manager
Placement Engine
EC2 Instance
ECS agent
Docker agent
ECS AMI
EC2 Instance
ECS agent
Docker agent
ECS AMI
EC2 Instance
ECS agent
Docker agent
ECS AMI

Comparison on Kubernetes terminology

Task definitions

Like a yaml file that describes a **Deployment**.

Task

Like a **Pod**

Container instance

Like a **Node**

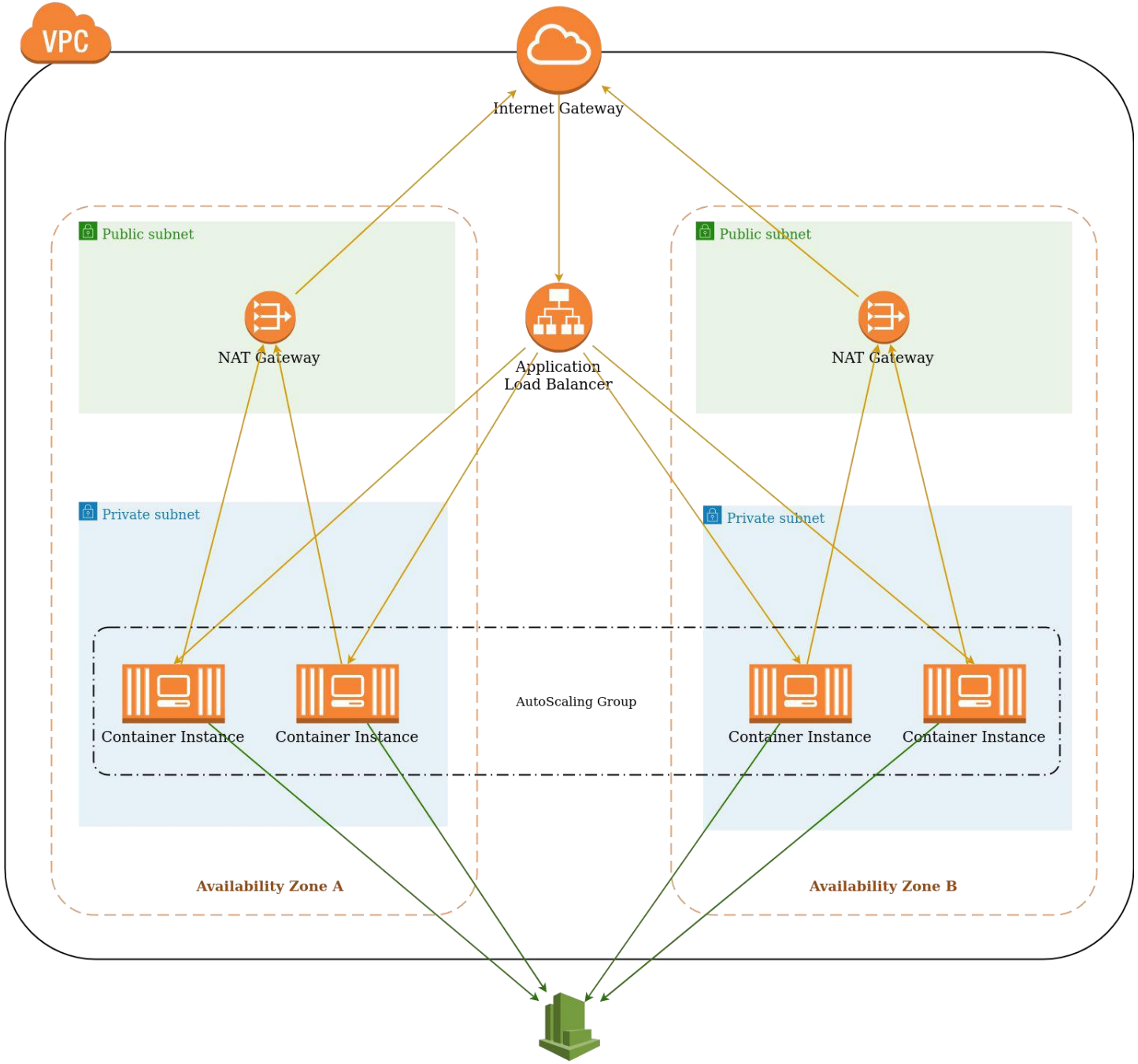
Container agent

Like a **kubelet**

Service

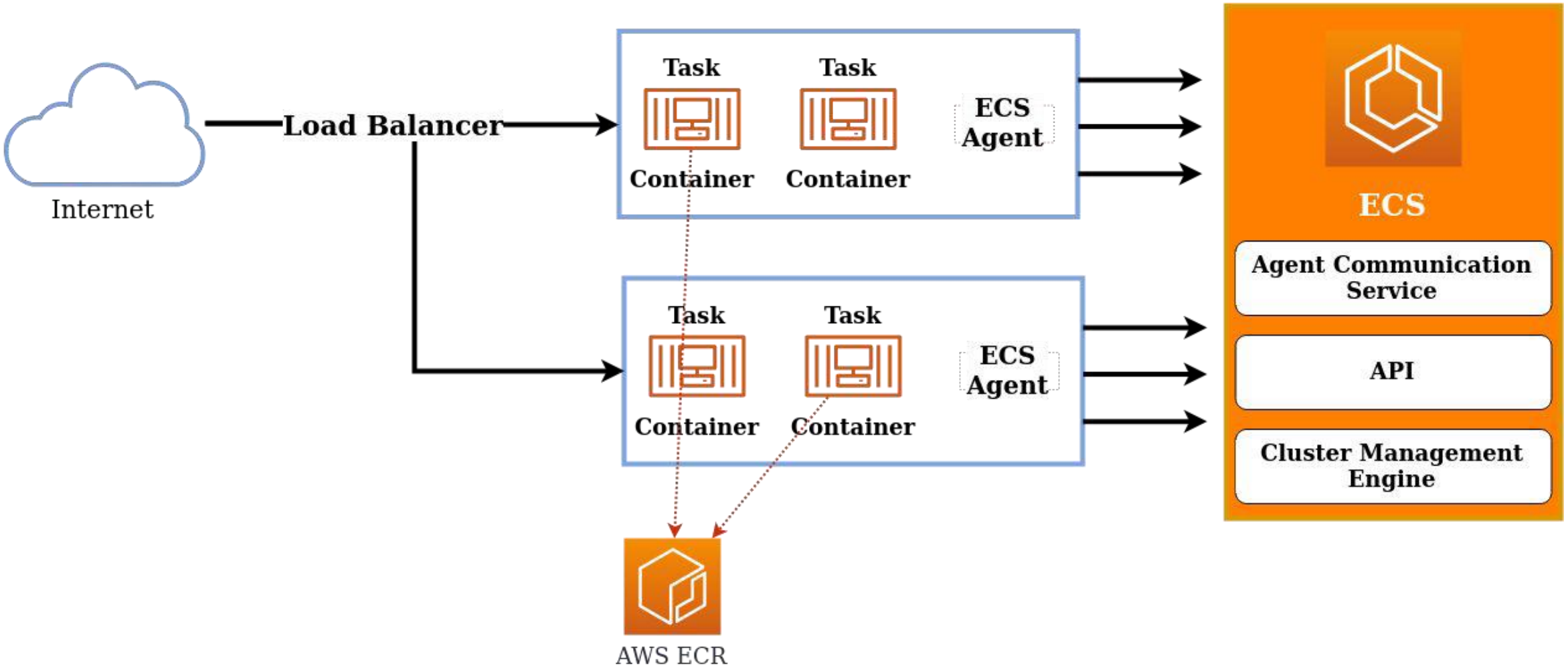
Like a **Deployment** with a **Controller**

credit:
https://medium.com/@eightlimbed/ecs-terminology-for-kubernetes-users-6f3ec91b4f47

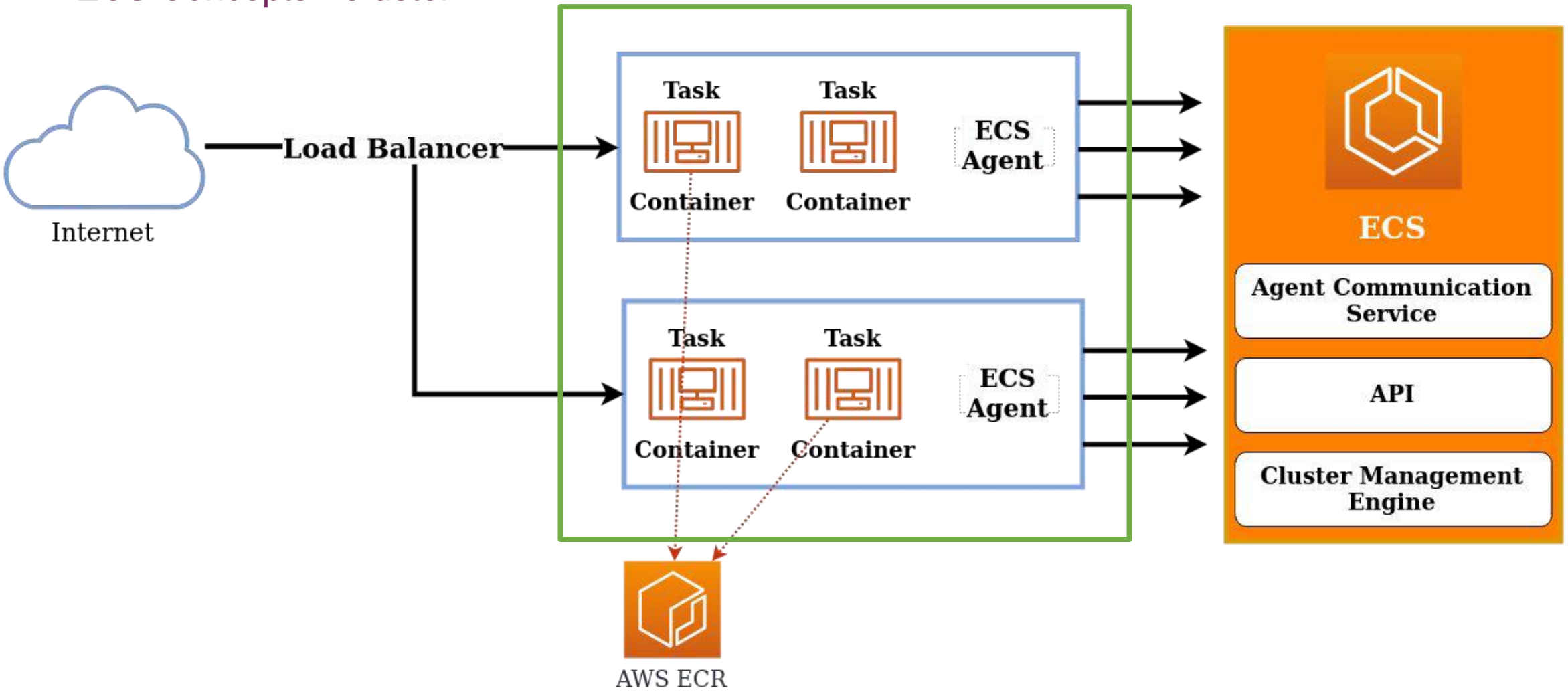


Sample ECS pattern

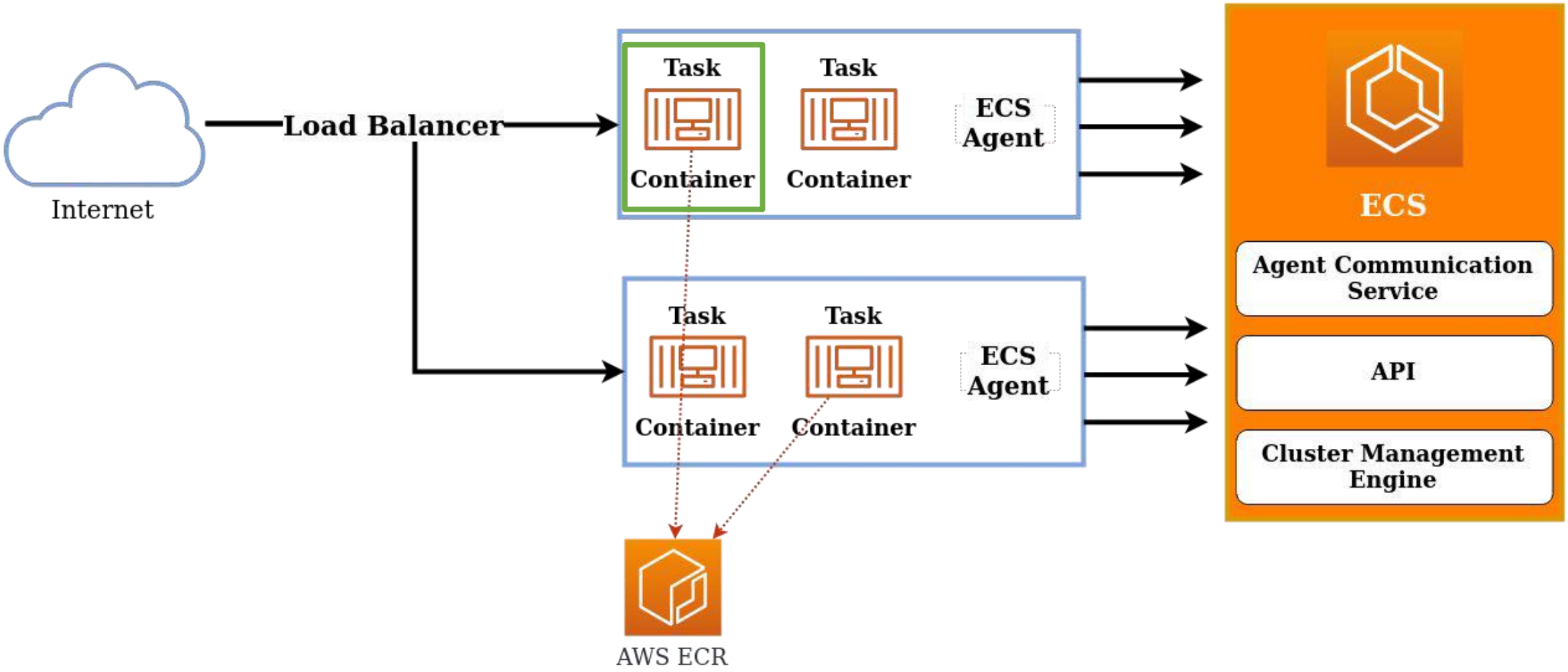
ECS Concepts



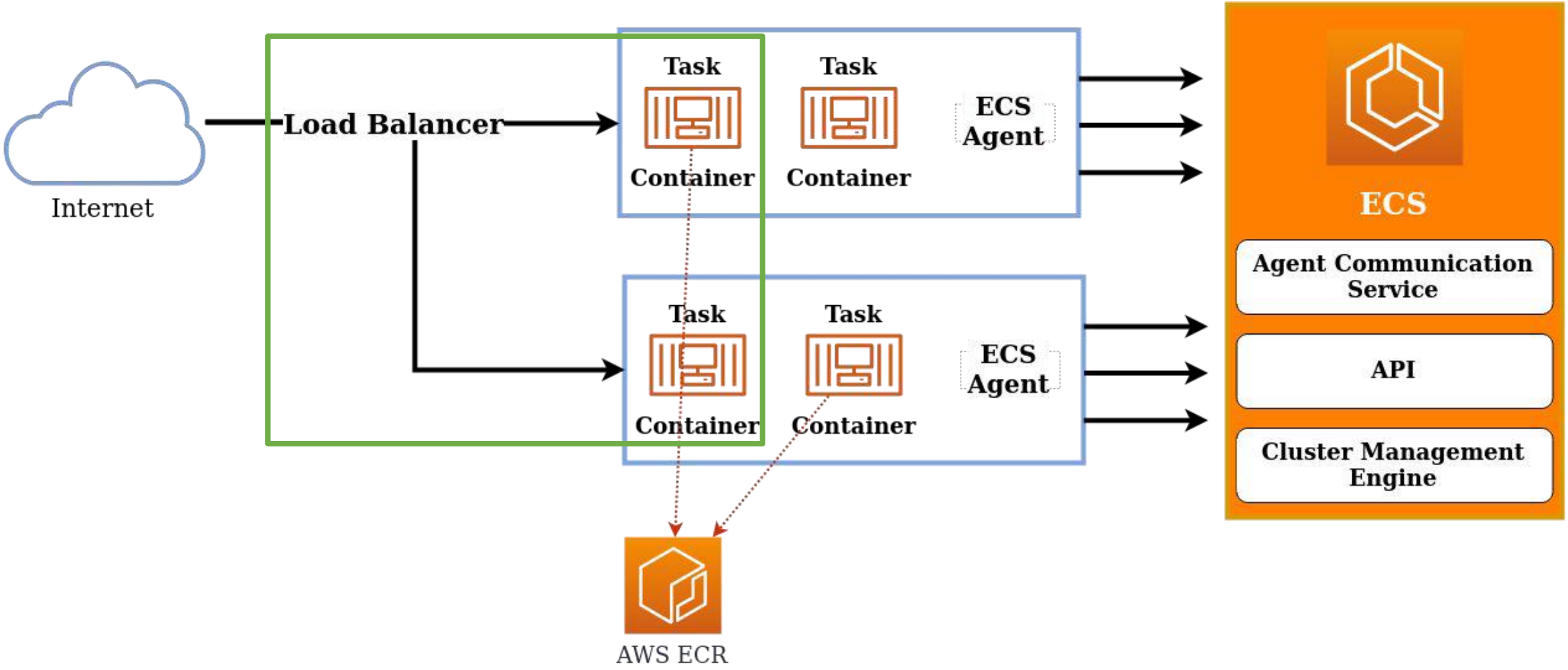
ECS Concepts - Cluster



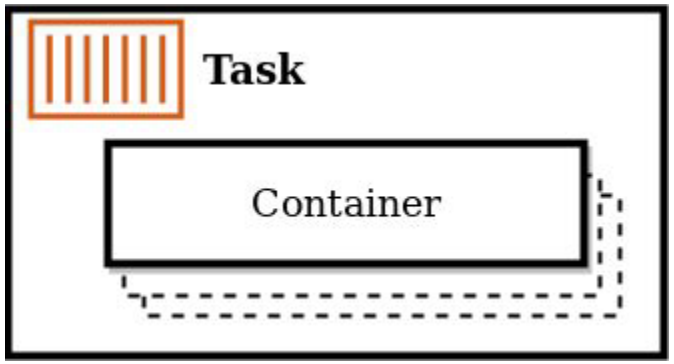
ECS Concepts - Task



ECS Concepts - Service



ECS - Task Definition



- Can contains up to 10 container definitions
- All containers are co-located on the same host

```
{
    "family": "devopsdays",
    "containerDefinitions": [
        {
            "name": "devopsdays-frontend",
            "image": "hub.docker.com/devopsdays-frontend"
        },
        {
            "name": "devopsdays-frontend",
            "image": "hub.docker.com/devopsdays-frontend"
        }
    ]
}
```

CPU and memory specification

```
{
    "family": "devopsdays",
    "cpu": "1024",
    "memory": "2048",
    "containerDefinitions": [
        {
            "name": "devopsdays-frontend",
            "image": "hub.docker.com/devopsdays-frontend",
            "cpu": "256",
            "memoryReservation": "512"
        },
        {
            "name": "devopsdays-frontend",
            "image": "hub.docker.com/devopsdays-frontend",
            "cpu": "768",
            "memoryReservation": "512"
        }
    ]
}
```

Task level

Container level

Simple walkthrough...

Create Cluster

Step 1: Select cluster template

Step 2: Configure cluster

Select cluster template

The following cluster templates are available to simplify cluster creation. Additional configuration and integrations can be added later.

Networking only

Resources to be created:

Cluster

VPC (optional)

Subnets (optional)

Powered by AWS Fargate

EC2 Linux + Networking

Resources to be created:

Cluster

VPC

Subnets

Auto Scaling group with Linux AMI

EC2 Windows + Networking

Resources to be created:

Cluster

VPC

Subnets

Auto Scaling group with Windows AMI

***Required**

Cancel

Next step

Step 1: Select cluster template

Step 2: Configure cluster

Configure cluster

Cluster name* ec2-spot-instance-cluster

☐ Create an empty cluster

Instance configuration

Provisioning Model

○ On-Demand Instance

With On-Demand Instances, you pay for compute capacity by the hour, with no long-term commitments or upfront payments.

● Spot

Amazon EC2 Spot Instances allow you to bid on spare Amazon EC2 computing capacity for up to 90% off the On-Demand price. Learn more

Spot Instance allocation strategy

● Diversified

Balance Spot Instances across selected Availability Zones and instance types

○ Lowest price

EC2 instance types*

- t3.medium
- t3.small
- t3a.small
- t3a.micro
- m5.large
- m1.medium

A maximum of six instance types can be selected.

Amazon ECS
- Clusters
- Task Definitions
- Account Settings

Amazon EKS
- Clusters

Amazon ECR
- Repositories

AWS Marketplace
- Discover software
- Subscriptions

Clusters > ec2-spot-instance-cluster

Cluster : ec2-spot-instance-cluster

Update Cluster | Delete Cluster

Get a detailed view of the resources on your cluster.

Status	ACTIVE
Registered container instances	1
Pending tasks count	0 Fargate, 0 EC2
Running tasks count	0 Fargate, 0 EC2
Active service count	0 Fargate, 0 EC2
Draining service count	0 Fargate, 0 EC2

Initial EC2 Spot instance as ECS instance

Services | Tasks | ECS Instances | Metrics | Scheduled Tasks | Tags | Capacity Providers

Add additional ECS Instances using Auto Scaling or Amazon EC2.

Actions

Last updated on December 9, 2019 1:01:56 AM (0m ago)

Status: ALL ACTIVE DRAINING

1-1 Page size 50

Filter by attributes (click or press down arrow to view filter options)

Container Instance	EC2 Instance	Availability Zone	Agent Connected	Status	Running tasks c...	CPU available	Memory availab...	Agent version ...	Docker version
05079df4-21b5-446d-be42-9...	i-0d565acbc590...	ap-southeast-2b	true	ACTIVE	0	2048	1955	1.33.0	18.06.1-ce

Create new Task Definition

Step 1: Select launch type compatibility

Step 2: Configure task and container definitions

Create task definition with EC2 launch type compatibility

Configure task and container definitions

A task definition specifies which containers are included in your task and how they interact with each other. You can also specify data volumes for your containers to use. Learn more

Task Definition Name* ecs-ec2-task

Requires Compatibilities* EC2

Task Role ecsServiceRole

Optional IAM role that tasks can use to make API requests to authorized AWS services. Create an Amazon Elastic Container Service Task Role in the IAM Console

Network Mode awsvpc

If you choose <default>, ECS will start your container using Docker's default networking mode, which is Bridge on Linux and NAT on Windows. <default> is the only supported mode on Windows.

Network Mode : awsvpc

Your containers in the task will share an ENI using a common network stack. Port mappings can only specify container ports (any existing host port specifications will be removed).

Task execution IAM role

This role is required by tasks to pull container images and publish container logs to Amazon CloudWatch on your behalf. If you do not have the ecsTaskExecutionRole already, we can create one for you.

Task execution role ecsTaskExecutionRole

Task size

The task size allows you to specify a fixed size for your task. Task size is required for tasks using the Fargate launch type and is optional for the EC2 launch type. Container level memory settings are optional when task size is set. Task size is not supported for Windows containers.

Pressing “Add container” button to configure

aws Services Resource Groups EC2 ECS

Create new Task Definition

Step 1: Select launch type compatibility

Step 2: Configure task and container definitions

Configure task and con

A task definition specifies which co
volumes for your containers to use.

Task Definitio

Requires Compat

Netwo

Network Mode : av
Your containers in the ta
ports (any existing host

Task execution IAM role

This role is required by tasks to pull
have the ecsTaskExecutionRole alre

Task execu

Task size

Add container

Standard

Container name* nginx

Image* nginx:latest

Private repository authentication*

Memory Limits (MiB)* Soft limit 512

Add Hard limit

Define hard and/or soft memory limits in MiB for your container. Hard and soft limits correspond to the `memory` and `memoryReservation` parameters, respectively, in task definitions.
ECS recommends 300-500 MiB as a starting point for web applications.

Port mappings

Container port	Protocol
80	tcp

Add port mapping

Host port mappings are not valid when the network mode for a task definition is host or awsvpc. To specify different host and container port mappings, choose the Bridge network mode.

Advanced container configuration

HEALTHCHECK

Command CMD-SHELL, curl -f http://localhost/ || exit 1

* Required

Cancel Add

Create Service

- **Step 1: Configure service**
- Step 2: Configure network
- Step 3: Set Auto Scaling (optional)
- Step 4: Review

Create a service based on previous task definition

Configure service

A service lets you specify how many copies of your task definition to run and maintain in a cluster. You can optionally use an Elastic Load Balancing load balancer to distribute incoming traffic to containers in your service. Amazon ECS maintains that number of tasks and coordinates task scheduling with the load balancer. You can also optionally use Service Auto Scaling to adjust the number of tasks in your service.

Launch type ○ FARGATE ● EC2

Task Definition
Family: ecs-ec2-task [Enter a value]
Revision: 1 (latest)

Cluster ec2-spot-instance-cluster

Service name nginx-service-1

Service type* ● REPLICA ○ DAEMON

Number of tasks 1

Minimum healthy percent 100

Maximum percent 200

Deployments

Choose a deployment option for the service.

Deployment type* ● Rolling update

○ Blue/green deployment (powered by AWS CodeDeploy)

This sets AWS CodeDeploy as the deployment controller for the service. A CodeDeploy application and deployment group are created automatically with default settings for the service. To change to the rolling update deployment type after the service has been created, you must re-create the service and select the "rolling update" deployment type.

Setup a load balancer in service configuration

Load balancer type*

- None
  Your service will not use a load balancer.
- Application Load Balancer
  Allows containers to use dynamic host port mapping (multiple tasks allowed per container instance). Multiple services can use the same listener port on a single load balancer with rule-based routing and paths.
- Network Load Balancer
  A Network Load Balancer functions at the fourth layer of the Open Systems Interconnection (OSI) model. After the load balancer receives a request, it selects a target from the target group for the default rule using a flow hash routing algorithm.
- Classic Load Balancer
  Requires static host port mappings (only one task allowed per container instance); rule-based routing and paths are not supported.

Service IAM role Task definitions that use the awsvpc network mode use the AWSServiceRoleForECS service-linked role, which is created for you automatically. Learn more.

Load balancer name devopsdays-lb

Container to load balance

nginx : 80 Remove

Production listener port* 80:HTTP

Production listener protocol* HTTP

Target group name tg-devopsdays

Target group protocol HTTP

Target type ip

Path pattern / **Evaluation order** default

Health check path /

Additional health check options can be configured in the ELB console after you create your service.

Clusters > ec2-spot-instance-cluster

Cluster : ec2-spot-instance-cluster

Update Cluster | Delete Cluster

Get a detailed view of the resources on your cluster.

Status	ACTIVE
Registered container instances	1
Pending tasks count	0 Fargate, 0 EC2
Running tasks count	0 Fargate, 1 EC2
Active service count	0 Fargate, 1 EC2
Draining service count	0 Fargate, 0 EC2

Task is running as desired

Services | Tasks | ECS Instances | Metrics | Scheduled Tasks | Tags | Capacity Providers

Create | Update | Delete | Actions

Last updated on December 9, 2019 3:10:18 AM (0m ago)

Filter in this page | Launch type ALL | Service type ALL | 1-1

Service Name	Status	Service type	Task Definition	Desired tasks	Running tasks ...	Launch type	Platform version
ecs-ec2-task-1	ACTIVE	REPLICA	ecs-ec2-task:1	1	1	EC2	--

ECS container networking

none

Tasks do not have external connectivity and port mappings can't be specified in the container definition

bridge

Task utilizes Docker's built-in virtual network which runs inside each container instance

host

Bypasses Docker's built-in virtual network and maps container ports directly to the EC2 instance's network

awsvpc

Task is allocated an elastic network interface, and you must specify a NetworkConfiguration when you create a service

Clusters > ec2-spot-instance-cluster

Cluster : ec2-spot-instance-cluster

Update Cluster | Delete Cluster

Get a detailed view of the resources on your cluster.

Status	ACTIVE
Registered container instances	1
Pending tasks count	0 Fargate, 0 EC2
Running tasks count	0 Fargate, 2 EC2
Active service count	0 Fargate, 1 EC2
Draining service count	0 Fargate, 0 EC2

ECS Task with EC2 launch type with awvpc issue

Services | Tasks | ECS Instances | Metrics | Scheduled Tasks | Tags | Capacity Providers

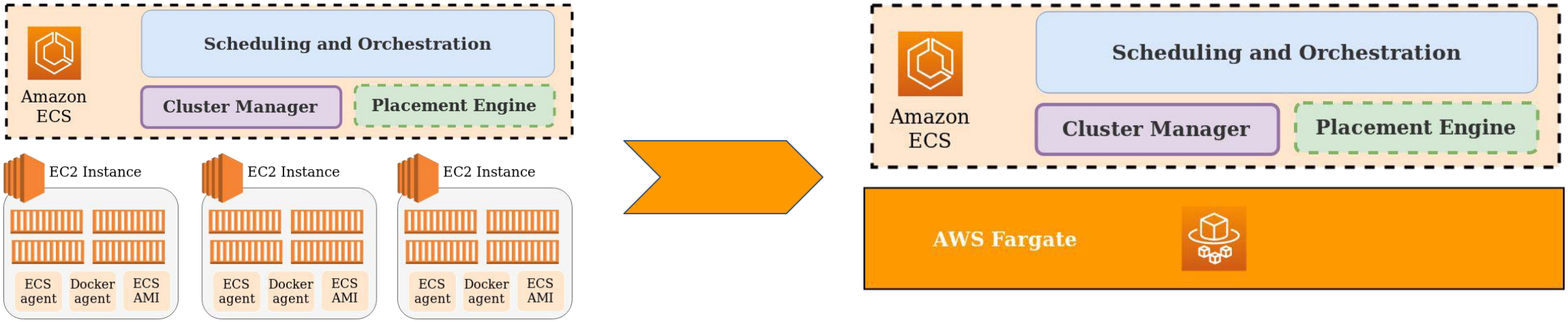
Create | Update | Delete | Actions

Last updated on December 9, 2019 3:23:51 AM (0m ago)

Filter in this page | Launch type ALL | Service type ALL | 1-1

Service Name	Status	Service type	Task Definiti...	Desired task...	Running task...	Launch type	Platform ver...
ecs-ec2-task-1	ACTIVE	REPLICA	ecs-ec2-task:1	5	2	EC2	–

ECS & Fargate


Amazon ECS
Scheduling and Orchestration
Cluster Manager
Placement Engine
EC2 Instance
ECS agent
Docker agent
ECS AMI
EC2 Instance
ECS agent
Docker agent
ECS AMI
EC2 Instance
ECS agent
Docker agent
ECS AMI
Amazon ECS
Scheduling and Orchestration
Cluster Manager
Placement Engine
AWS Fargate

AWS Fargate

Managed by AWS

No EC2 instances to provision, scale or manage

Elastic

Scale up & down seamlessly. Pay only for what you use

Integrated

VPC Networking, ELB, IAM, CloudWatch, etc.

Gotchas:

Default ECS service limit on concurrent Fargate tasks is 50 per region

credit: aws-ecs-workshop-a-journey-to-modern-applications

Task Definitions > ecs-ec2-task > 2

Task Definition: ecs-ec2-task:2

View detailed information for your task definition. To modify the task definition, you need to create a new revision and then make the required changes to the task definition

Create new revision | Actions

Builder | JSON | Tags

After updating our task definition to include fargate compatibility

Task Definition Name	ecs-ec2-task
Task Role	ecsServiceRole
Network Mode	awsvpc If you choose <default>, ECS will start your container using Docker's default networking mode, which is Bridge on Linux and NAT on Windows. <default> is the only supported mode on Windows.
Compatibilities	EC2, FARGATE
Requires compatibilities	EC2, FARGATE

Task execution IAM role

This role is required by tasks to pull container images and publish container logs to Amazon CloudWatch on your behalf. If you do not have the ecsTaskExecutionRole already, we can create one for you.

Task execution role	ecsTaskExecutionRole

Clusters > ec2-spot-instance-cluster

Cluster : ec2-spot-instance-cluster

Update Cluster | Delete Cluster

Get a detailed view of the resources on your cluster.

Status	ACTIVE
Registered container instances	2
Pending tasks count	5 Fargate, 0 EC2
Running tasks count	0 Fargate, 4 EC2
Active service count	1 Fargate, 1 EC2
Draining service count	0 Fargate, 0 EC2

Created a new service after using updated task definition and launch type

Services | Tasks | ECS Instances | Metrics | Scheduled Tasks | Tags | Capacity Providers

Create | Update | Delete | Actions

Last updated on December 9, 2019 3:37:19 AM (0m ago)

Filter in this page | Launch type ALL | Service type ALL | 1-2

Service Name	Status	Service type	Task Definiti...	Desired task...	Running task...	Launch type	Platform ver...
ecs-fargate-task-1	ACTIVE	REPLICA	ecs-ec2-task:2	5	5	FARGATE	LATEST(1.3.0)
ecs-ec2-task-1	ACTIVE	REPLICA	ecs-ec2-task:2	5	4	EC2	--

What’s to get from all of these

AWS ECS can simplify many of patterns in modern applications such as batch jobs, long live, etc.

AWS Fargate serverless compute provides almost *unlimited* flexibility and resources on compute

Vendor lock-in problem

Less documentations, case studies, open source tools compared to the alternatives

Extra: What's new on ECS

Fargate Spot

AWS EKS (Kubernetes) on Fargate

Extra: Resources on ECS

https://github.com/awslabs?q=ecs

https://github.com/nathanpeck/awesome-ecs

https://github.com/aws/amazon-ecs-cli

https://aws.amazon.com/blogs/devops/build-a-continuous-delivery-pipeline-for-your-container-images-with-amazon-ecr-as-source/

https://aws.amazon.com/blogs/compute/set-up-a-continuous-delivery-pipeline-for-containers-using-aws-codepipeline-and-amazon-ecs/

Credits:

https://www.slideshare.net/Docker/introduction-to-docker-2017

https://www.slideshare.net/AmazonWebServices/aws-ecs-workshop-a-journey-to-modern-applications

https://aws.amazon.com/blogs/compute/powering-your-amazon-ecs-cluster-with-amazon-ec2-spot-instances

https://aws.amazon.com/blogs/compute/building-deploying-and-operating-containerized-applications-with-aws-fargate/

https://aws.amazon.com/blogs/compute/building-blocks-of-amazon-ecs/

Stay Connected

@IDDevOps

http://www.devopsindonesia.com

@IDDevOps

DevOps Indonesia

@devopsindonesia

Alone We are smart, together We are brilliant

THANK YOU !

DEVOPS INDONESIA

Quote by Steve Anderson