BEFORE WE START

- **This session is being recorded. You have been warned !**
- **Slide deck and video recording will be made available.**
- **You can ask questions in chat throughout the session.**

rackspace®
the #1 managed cloud company

EC2 CONTAINER SERVICE (ECS)

amazon web services

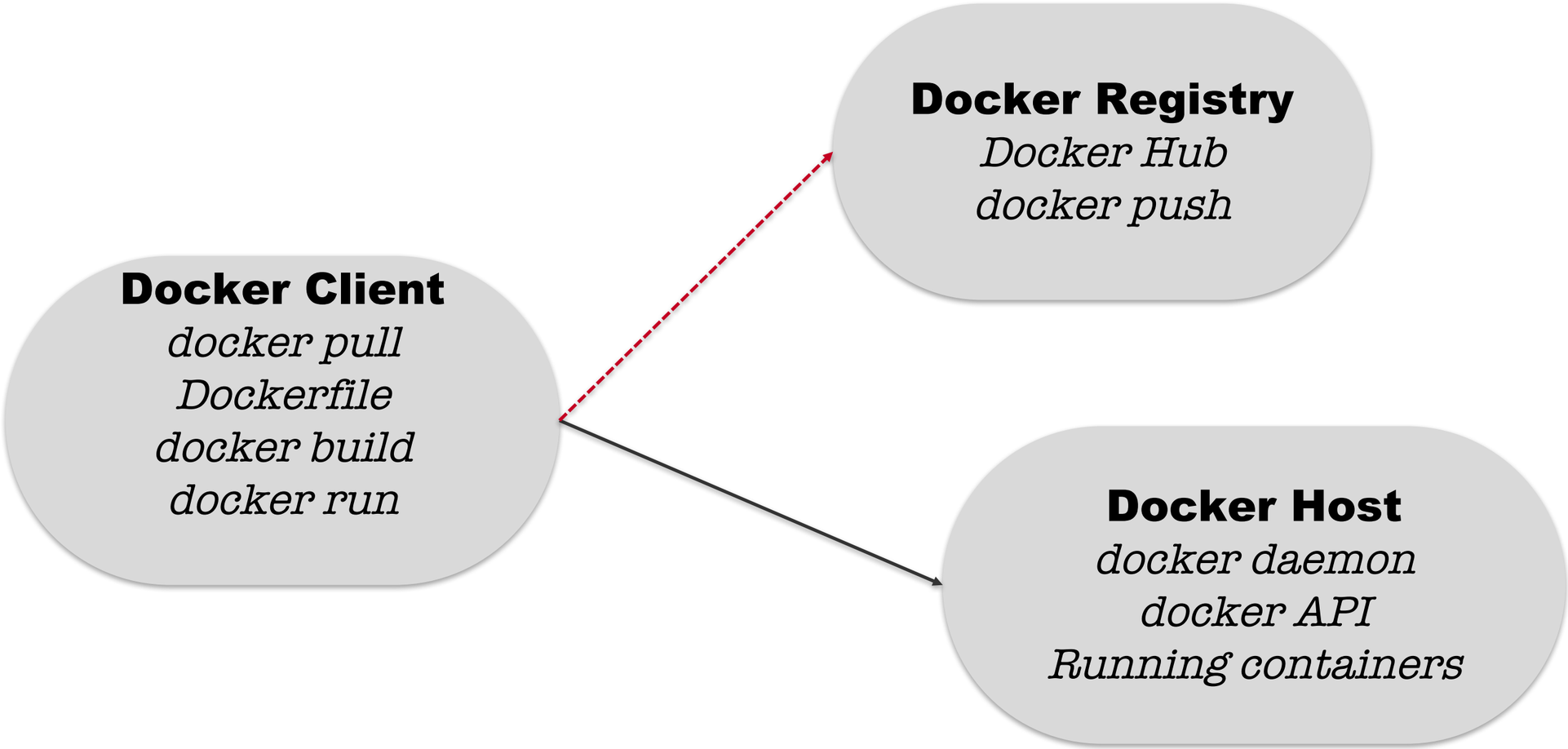
- Introduction
- Docker Recap
- ECS Concepts
- ECS Tutorial
- ECS Operational Considerations
- ECR
- Q & A

WHO AM I

DOCKER - RECAP

- Started as an internal project for DotCloud
- Open sourced in March 2013
- Written in Go and libcontainer (its own container code)
- Winning the Container war

DOCKER BASICS



DOCKER – FURTHER READING

- https://docs.docker.com/engine/getstarted/
- https://github.com/docker/labs

AWS ECS

- The AWS way to manage & run Docker in production. Part of the AWS ecosystem;
  - Autoscale models, IAM, etc.
  - No additional cost for the ECS layer

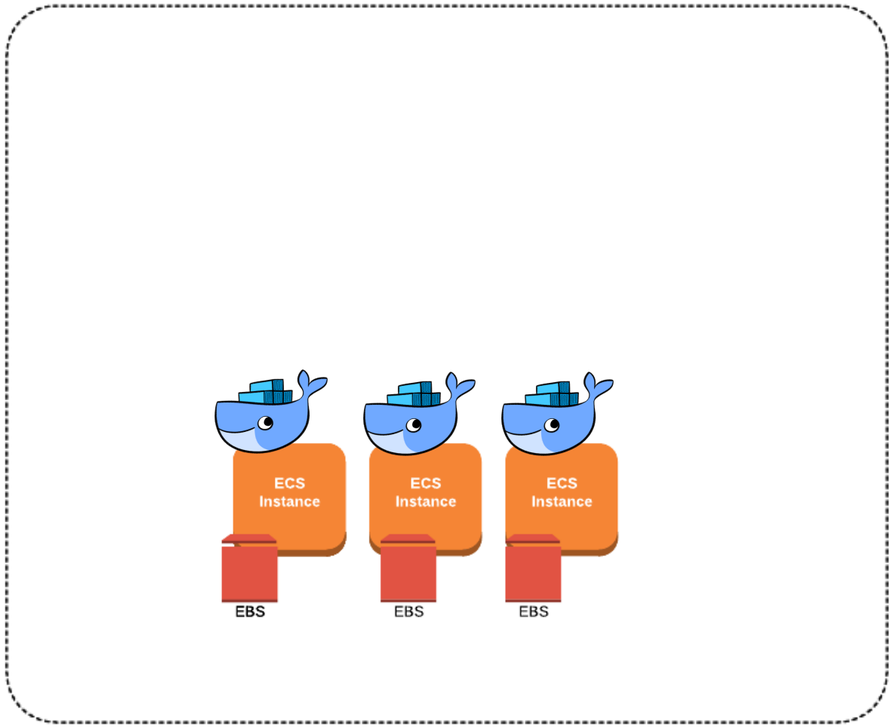
- Container management ecosystem
  - Docker Swarm
  - Kubernetes
  - Mesos/Marathon

ECS CONCEPTS

- *Cluster* : The cluster is a skeleton structure around which you build and operate workloads. EC2 container instances logically belong to it

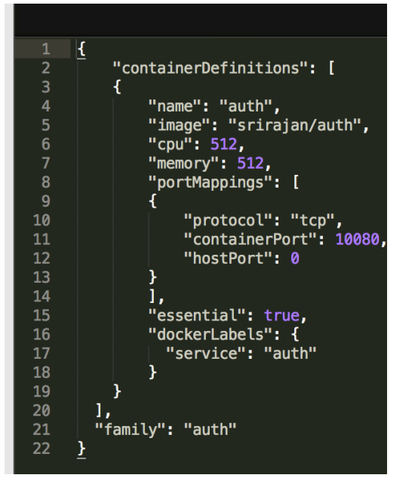
- *Container Instance/s* : This is actually an EC2 instance running Docker daemon

- *Container agent*: ECS agent that runs on the instance and talks to the management layers



ECS CONCEPTS

- *Task Definition* : A template the describes a task/application containing one or more containers. This is where you provide the Docker images, how much CPU/Memory to use, ports, etc.



```
{
    "containerDefinitions": [
    {
        "name": "auth",
        "image": "srirajan/auth",
        "cpu": 512,
        "memory": 512,
        "portMappings": [
        {
            "protocol": "tcp",
            "containerPort": 10080,
            "hostPort": 0
        }
        ],
        "essential": true,
        "dockerLabels": {
          "service": "auth"
        }
    }
  ],
  "family": "auth"
}
```

ECS CONCEPTS

- *Service* : A service allows you to run and maintain a specified number of instances of a task definition. Services ensure that the desired number running tasks is achieved and maintained.

Load balancer*

IAM Role

No of Tasks

Task Definition

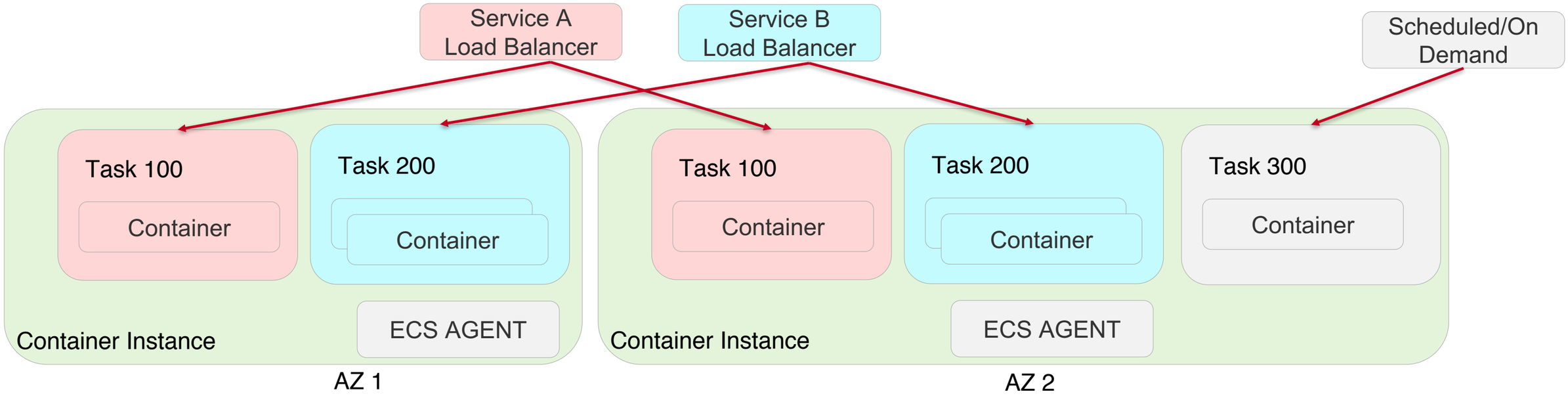
Cluster

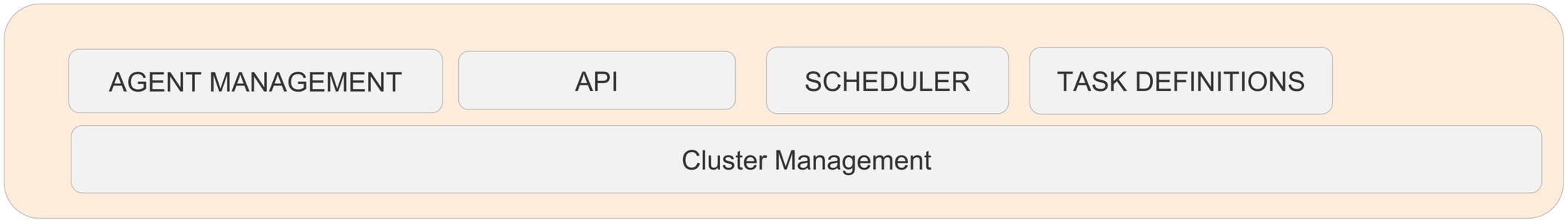
ECS CONCEPTS

- *Task* : An instance of a task definition running on a container instance.
- *Container* : A Docker container that is executed as part of a task.

Task	Task Definition	Container Instance	Last status	Desired status
04b352e1-1361-43ef-9391-9fc0b5c93e62	calvin:3	0e143f9c-18d0-4d0c-9...	RUNNING	RUNNING
3bb4d997-1aa9-48c4-b007-1b7cc6ca6d67	fin:2	0e143f9c-18d0-4d0c-9...	RUNNING	RUNNING
4d3e235b-6436-490e-bac3-af10e2dfd7ff	main:1	bfec6cd7-c76a-4367-9...	RUNNING	RUNNING
5ea010d5-d692-4eeb-80d1-81897bdcdd80	calvin:3	bfec6cd7-c76a-4367-9...	RUNNING	RUNNING

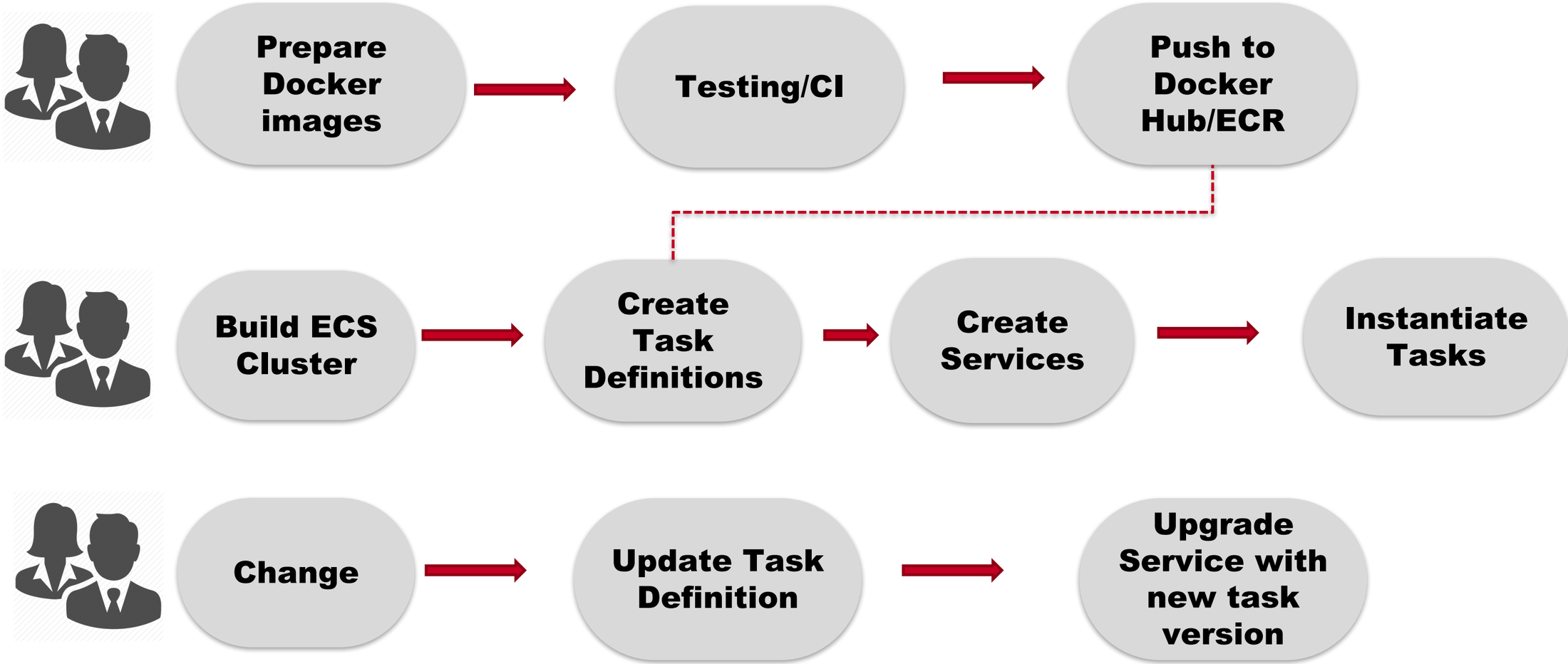
ECS CONCEPTS





rackspace
the #1 managed cloud company

ECS - Flow



ECS – TUTORIAL

https://github.com/rackerlabs/ecs-playground

ECS – SCHEDULING

- Long running services
  - Common job type or use case (e.g. a Web service).
  - ECS service scheduler ensures that the specified number of tasks are constantly running and reschedules tasks when they fail (for example, if the underlying container instance fails)
  - The service scheduler, optionally, makes sure that tasks are registered against an Elastic Load Balancer, which can do additional health checks

ECS – SCHEDULING

- Batch/Workers
  - On-demand or a scheduled job that executes in reaction to an event
  - RunTask action is ideally suited and you could have a process call RunTask when work comes into a queue
  - Cron-like fashion using CloudWatch Events rule that runs one or more tasks in your cluster at specified times.

ECS – SCHEDULING

- Custom
  - Blox – Open source scheduling framework
  - https://blox.github.io/

ECS – PLACEMENT

- Strategies (decide which container instance to run containers)
  - binpack: Place tasks based on the least available amount of CPU or memory. This minimizes the number of instances in use
  - random : Place tasks randomly
  - spread: Place tasks based on the specified key:value pairs. E.g.

```
"placementStrategy": [
{
   "field": "attribute:ecs.availability-zone",
   "type": "spread"
} ]
```

ECS – PLACEMENT

- Constraints
  - distinctInstance: Place each task on a different container instance
  - memberOf : Place tasks on container instances that satisfy an expression. This could be instance type, AMI ID, AZ, etc.

"placementConstraints": [
{ "expression": "attribute:ecs.instance-type =~ m4.", "type": "memberOf" }]*

"placementConstraints": [
{ "expression": "task:group == nginx", "type": "memberOf" }]

ECS – AUTOSCALING

- Scaling Clusters
  - Same as EC2 autoscaling. You can configure autoscale policies and launch configurations as you would do with any EC2 setup (e.g. using CPU or RAM)

- Scaling Services
  - Auto-scale services to adjust its desired count up or down in response to CloudWatch alarms
  - You can also use CloudWatch metrics published by other services. For example, a web service could increase the number of tasks based on Elastic Load Balancing metrics

ECS – CREDENTIAL STORAGE

- Environment variables (avoid)
- Encrypted S3 buckets with IAM
- EC2 Parameter storage is a good fit (https://blog.rackspace.com/securing-application-secrets-with-ec2-parameter-store)
- External service like Vault or Credstash

ECS – IAM ROLES AND TASKS

- Assign IAM roles to ECS tasks. Similar to IAM roles for EC2 but at the task level
- Credential Isolation: A container can only retrieve credentials for the IAM role that is defined in the task definition to which it belongs. Different tasks can have different privilege and security levels
- Auditability: Task credentials have a context of taskArn that is attached to the session, so CloudTrail logs show which task is using which role

ECS – CLOUD NATIVE NETWORKING (Released Nov 14th)

- Attach an ENI (Elastic Network Interface) directly to a task
- This
  - Allows you to treat each container just like an EC2 instance with full networking features
  - Provides better segmentation and security controls (security groups at the task level)
  - Will improve networking performance
  - Allows you to run multiple copies of the container on the same instance using the same container port

ECS –AGENT CONFIGURATION

- Loads of environment variables
  - */etc/ecs/ecs.config*
- Image cleanup
  - *ECS_ENGINE_TASK_CLEANUP_WAIT_DURATION (default 3 hours)*
  - *ECS_IMAGE_CLEANUP_INTERVAL (default : 30 mins)*
- Docker Hub Authentication
  - *ECS_ENGINE_AUTH_TYPE=docker*
  - *ECS_ENGINE_AUTH_DATA={"https://index.docker.io/v1/"{"username":"my_name","password":" my_password","email":"email@example.com"}}""*

ECS – AGENT CONFIGURATION

- Privileged containers
  - *ECS_DISABLE_PRIVILEGED (default false)*
- Use S3 to store configuration
  - *#!/bin/bash*
    yum install -y aws-cli
    aws s3 cp s3://your_bucket_name/ecs.config /etc/ecs/ecs.config

ECS – Monitoring

- CPU & Memory
  - EC2 Instance
  - ECS Service
  - ECS Cluster
- Memory
  - ECS Service
  - ECS Cluster
- Number of tasks in a service

ECS – Logging

- ECS Logs
  ECS_LOGLEVEL=debug
  ECS_LOGFILE=/somefile.log

- Cloudwatch Logs
  - Use cloud-init to install awslogs and consolidate logs to a central place

- https://github.com/awslabs/ecs-logs-collector
  (collects all logs including the container level)

ECS – BEST PRACTICES

- 12-factor app; https://12factor.net/ Best practices for micro-services

- Use ALB (ELB v2 a.k.a Application Load balancer)
  - Lots of features for use with containers
  - Dynamic ports
  - Content/Host/URL path based routing

ECS – BEST PRACTICES

- Have a logging strategy; i.e. consolidate logs from start

- Keeping AMIs updated
  - Important to do; Agent changes, Docker updates , Kernel updates
  - Subscribe to Amazon SNS topic when a new AMI is available. http://docs.aws.amazon.com/AmazonECS/latest/developerguide/ECS-AMI-SubscribeTopic.html

ECS – WINDOWS

- Windows containers are now available in Beta
- Launched with the Microsoft Windows Server 2016 Base
- Windows containers cannot run on Linux instances and vice-versa
- Still early days !
  - Windows on ECS is not at feature parity with Linux
  - Container instances take about 15-20 minutes(first time only) to start (Windows image size)

ECS - Limitations

- You are still running a container on a VM.
  - AWS has some instances where you can get close to bare-metal (user controlled c-states and p-states)

- Service discovery (key in micro-services world), is limited inside ECS.
  - You can run something like Consul or Weave and use it for service discovery.

- There is no central Docker endpoint as far as the ECS cluster is concerned. As a developer, you cannot point your local Docker client to an ECS endpoint.
  - ECS purpose is targeted at running Docker instances from an Operations perspective.

ECS - Pricing

- No cost for the ECS layer

- You pay for
  - EC2 (instance sizes, reserved instances)
  - EBS (Elastic Block Storage)
  - Load balancer
  - Bandwidth

ECR (ELASTIC CONTAINER REGISTRY)

- Very simple service
- Same as Docker Hub
- Highly available Docker repository
- Integrated into IAM
- Lifecycle policies

ECR (ELASTIC CONTAINER REGISTRY)

- Only pay for data stored and the network transfer
- Can be used with on-premise Docker setups (with AWS cli)
- No public images (needs an IAM user)

THAT's it for ECS today. Let's continue the conversation...

https://github.com/rackerlabs/ecs-playground
(https://goo.gl/Su9n1q)
Speak to your TAM

rackspace
the #1 managed cloud company

Over to our product team

FAWS ROADMAP

NOW
Compass: Report Performance Enhancements Compass provides, among many other things, advanced capabilities that allow you to get in-depth cost and inventory reporting over time. We'll be enhancing performance for many report types and pages within Compass so that you can get the specific information you need faster. Target Launch: November 2017
Deeper Monitoring and Alerting Capabilities Today, we manage AWS CloudWatch for our Aviator customers, which provides limited capabilities for monitoring and alerting on EC2 and other platforms. The purpose of this enhancement is to provide deeper capabilities, including full CDM (CPU, Disk, and Memory), intelligent alerting, and more. We're considering a third party tool (to be managed by Rackspace). Target Launch: Q1 2018
Customer Education Webinars AWS Certified Rackers will host live monthly webinar sessions to educate and inform customers about AWS services. These sessions will include Q&A, and be recorded and made available for viewing after the date. Target Launch: November 2017

WHAT'S NEXT IN THIS SERIES?

RECAP

Dec 7th , 9.00 AM US CENTRAL

Questions- Please use the chat option

You can also reach out to your TAM for follow up

Slides & GitHub links goo.gl/Su9n1q

rackspace | YOUR CLOUDS. OUR EXPERTISE.