AWS Fargate in practice.

How to run containers directly, without managing EC2 instances.

By Max Borysov

Amazon ECS

Amazon Elastic Container Service

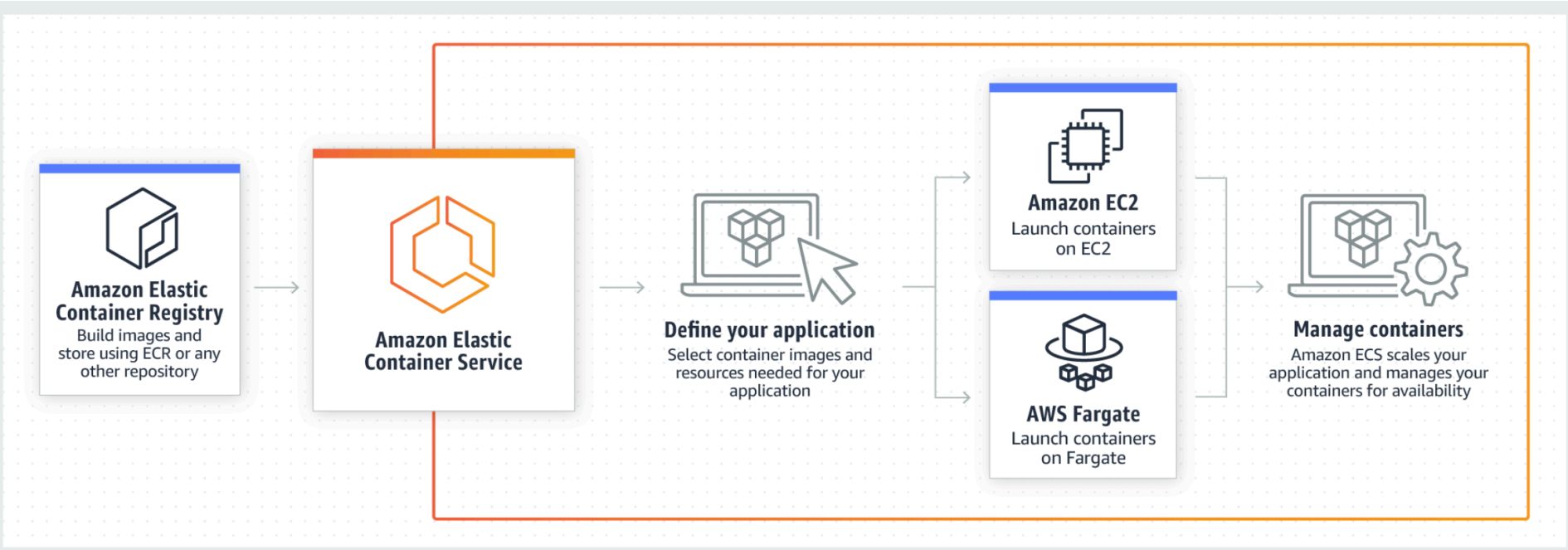
is a highly scalable, high-performance container orchestration service

that supports Docker containers

and bla-bla-bla.

Just helps to run containerized applications in production.

How Amazon ECS works



AWS Fargate

Technology is available with Amazon ECS.

With AWS Fargate, you no longer have to select Amazon EC2 instance types, provision and scale clusters, or patch and update each server.

Key terms

Repositories: place where docker images are stored and versionized.

Key terms

Cluster: place for services, tasks, task definitions. Some sort of a namespace.

An Amazon ECS cluster is a regional grouping of one or more container instances on which you can run task requests.

Each account receives a default cluster the first time you use the Amazon ECS service.

Clusters may contain more than one Amazon EC2 instance type.

Key terms

Task Definitions: configuration for container(based on docker image from AWS repositories).You can specify container resources(CPU, RAM), ENV variables, volumes, etc.

Task definitions specify the container information for your application, such as how many containers are part of your task, what resources they will use, how they are linked together, and which host ports they will use.

Key terms

Scheduled Tasks: represent scheduled task definitions with additional parameters.

Types of schedule: periodic(eg. hourly), fixed time(eg. 12:30PM)

Pricing

Fargate Launch Type Model: you pay for the amount of vCPU and memory resources that your containerized application requests.

vCPU and memory resources are calculated from the time your container images are pulled until the Amazon ECS Task* terminates, rounded up to the nearest second.

EC2 Launch Type Model: there is no additional charge for EC2 launch type. You pay for AWS resources (e.g. EC2 instances or EBS volumes) you create to store and run your application. You only pay for what you use, as you use it; there are no minimum fees and no upfront commitments.

ECS Costs examples

Fargate task

2vCPU, 4GB RAM, 8h/d, 30d

vCPU = 1 * 2 * $0.04048 * 8 * 30 = $19.43

RAM = 1 * 4 * $0.004445 * 8 * 30 = $4,26

Total: $23,69

EC2 task

t2.medium

2 vCPU, 4 GB RAM, 8h/d, 30d

8 * $0.0464 * 30 + 20GB * $0,10

Total: $13,13

Pricing comparison

Given: **500GB** of reserved db storage with **300GB** of data

300GB on RDS ~= **20GB** of raw compressed data

RDS snapshots	**AWS S3**	**AWS Glacier**
$0.095 per GB / M	$0.0125 per GB / M	$0.004 per GB / M
300GB * 30d = 9T	20GB * 30d = 600GB	20GB * 30d = 600GB
$855 per M	**$7.5** per M	**$2.4** per M
$5130 per 6M	**$45** per 6M	**$14.4** per 6M

AWS Glacier

Secure, durable, and extremely low-cost cloud storage service for data archiving and long-term backup

S3 Lifecycle policy: Standard-IA -> Glacier

S3 bucket events:

Restore from Glacier initiated,Restore from Glacier completed

Setup process

1. Build a docker image and deploy to AWS ECR(elastic container registry).
2. Create a Task Definition: specify docker image, task size(CPU, RAM), logs aggregator, ENV variables, etc.
3. Create a cluster.
4. Create a Scheduled Task. Specify Task Definition and schedule type.

DB backup process

Custom backups script utilising **aws cli**

1. Restore db instance from the snapshot (`aws rds restore-db-instance-to-point-in-time`)
2. Wait until it is ready (`aws rds wait db-instance-available`)
3. Generate backup file (`pg_dump/mysqldump`)
4. Compress
5. Encrypt backup
6. Copy to S3 (`aws s3 cp backup.dump.gpg $S3PATH –storage-class=STANDARD_IA`)
7. Delete backup
8. Delete instance (`aws rds delete-db-instance`)

Monitoring

- AWS CloudWatch
- Custom solution (eg. monitoring script to check if a backup for the current day is available)

Misc

Link to files:

https://github.com/paladinsoftware/fargate-tasks

In case you want to give a feedback, beer, hugs or some good vibes

max.borisov@paladinsoftware.com

THANK YOU

@PaladinSoftware

facebook.com/Paladin-Software

PaladinSoftware.com