AWS ECS & FARGATE

The good, the bad and the ugly

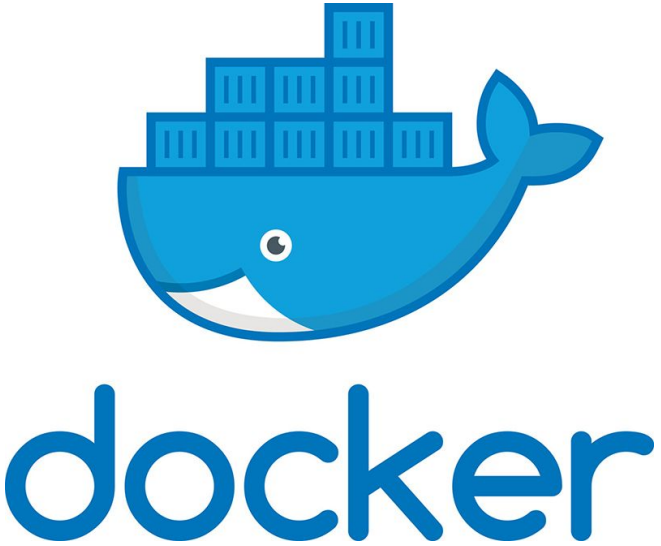
Jacob Verhoeks
Devops Engineer
Edrans

About me

- Dutch
- Background in Infrastructure and Medical IT
- 1 year fully working the “Cloud”
- AWS Architect & Devops Professional

AmazonECS

- Amazon Elastic Container Service
- Container Orchestration Service
- Managed by AWS
- API
- Docker only

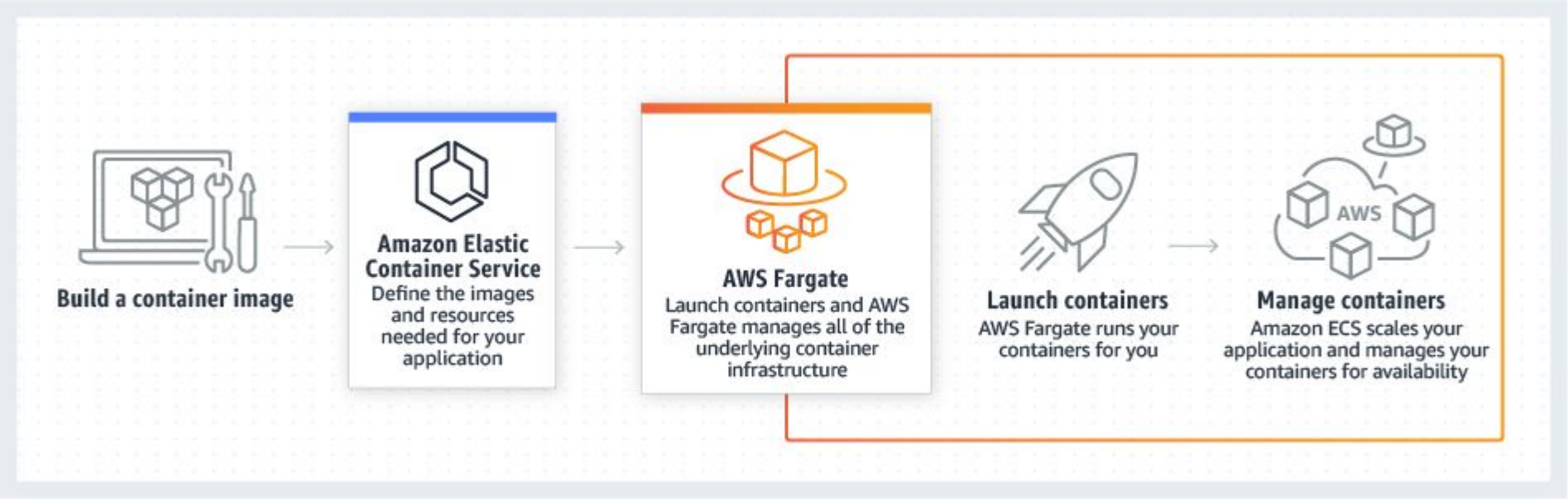


AmazonECS

- Need to provide ec2 nodes as workers
- “Supports” Autoscaling
- Install
  - Docker
  - ECS Agent
- AMI: amazon-optimized-ecs

AWS Fargate

- Run containers without managing servers or clusters
- Amazon provides all the resources and scaling



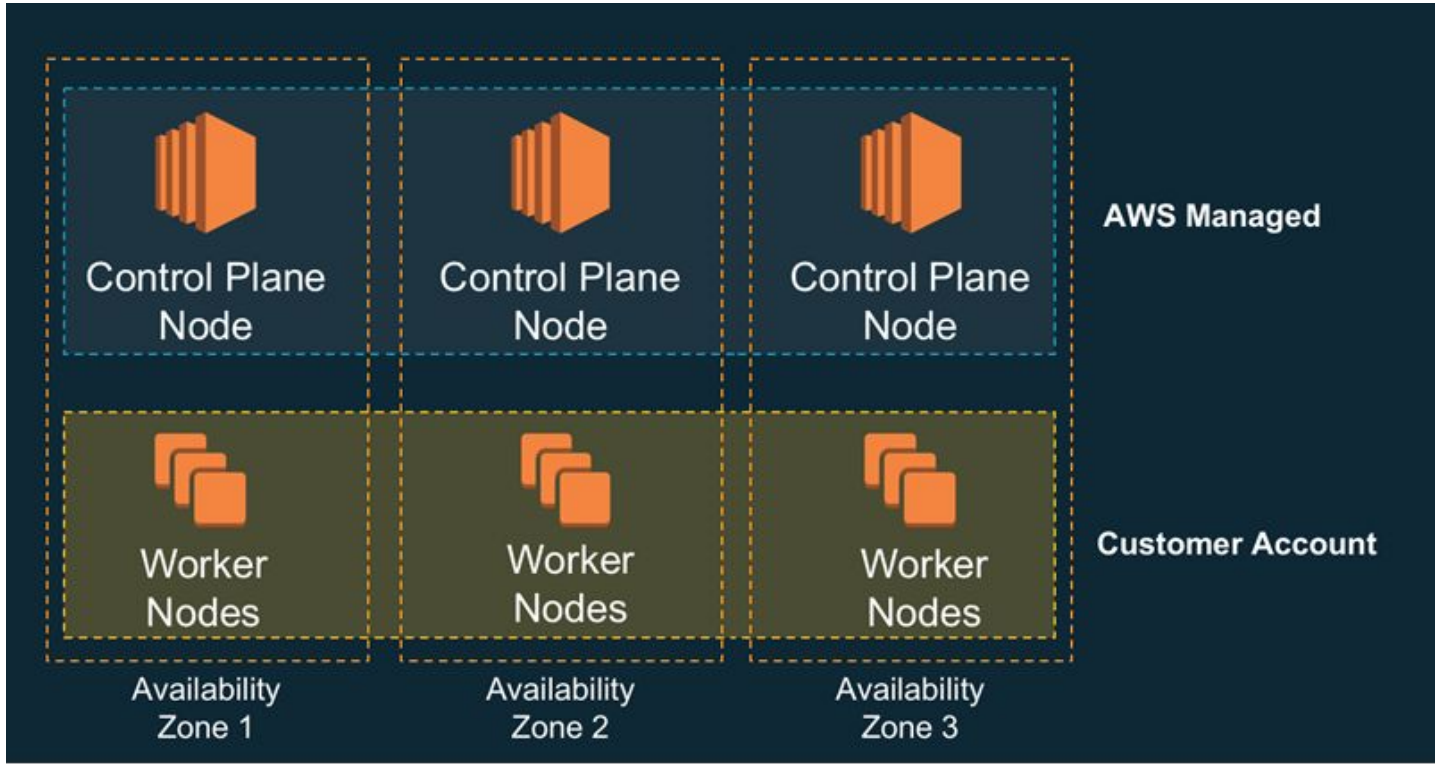
AWS Fargate

EKS: Kubernetes on AWS

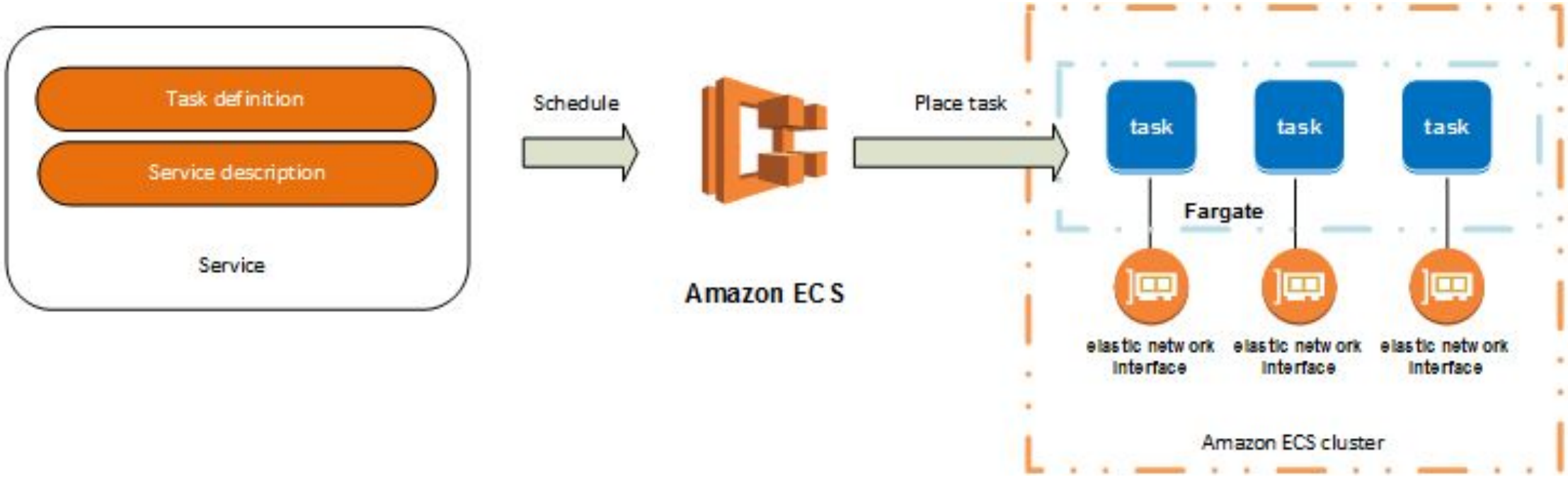
AWS Provided Control Plane where mgmt pods run.

Workers nodes are EC2 Customer Provided.

No Fargate like support (yet)

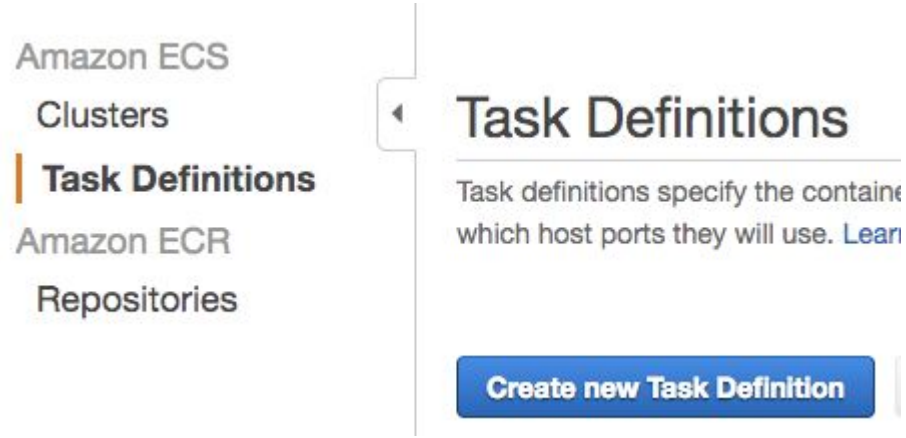


ECS


Task definition
Service description
Service
Schedule
Amazon ECS
Place task
task
task
task
Fargate
elastic network interface
elastic network interface
elastic network interface
Amazon ECS cluster

Basics: Task Definition

- Name
- Task Role
- Task Execution Role
- Network Mode
- Task Size (cpu/mem reservation+limits)
- Task Container
  - Port definition, logging (cloudwatch)
- Volumes



A Task definition can’t be edited, a change results in a new Revision. Like Launch Configuration

ECS Task Sizes

When using reservation, make sure there are enough resources. For example blue/green deployments.

Fargate has a fixed set of cpu-mem settings

CPU value	Memory value (MiB)
256 (.25 vCPU)	512 (0.5GB), 1024 (1GB), 2048 (2GB)
512 (.5 vCPU)	1024 (1GB), 2048 (2GB), 3072 (3GB), 4096 (4GB)
1024 (1 vCPU)	2048 (2GB), 3072 (3GB), 4096 (4GB), 5120 (5GB), 6144 (6GB), 7168 (7GB), 8192 (8GB)
2048 (2 vCPU)	Between 4096 (4GB) and 16384 (16GB) in increments of 1024 (1GB)
4096 (4 vCPU)	Between 8192 (8GB) and 30720 (30GB) in increments of 1024 (1GB)

https://docs.aws.amazon.com/AmazonECS/latest/developerguide/task-cpu-memory-error.html

Basics: Service

Description of how many Tasks you want to run

- Task Definition with revision
- Number of tasks
- Type: EC2/Fargate
- Placement strategy
  - How to spread over the servers

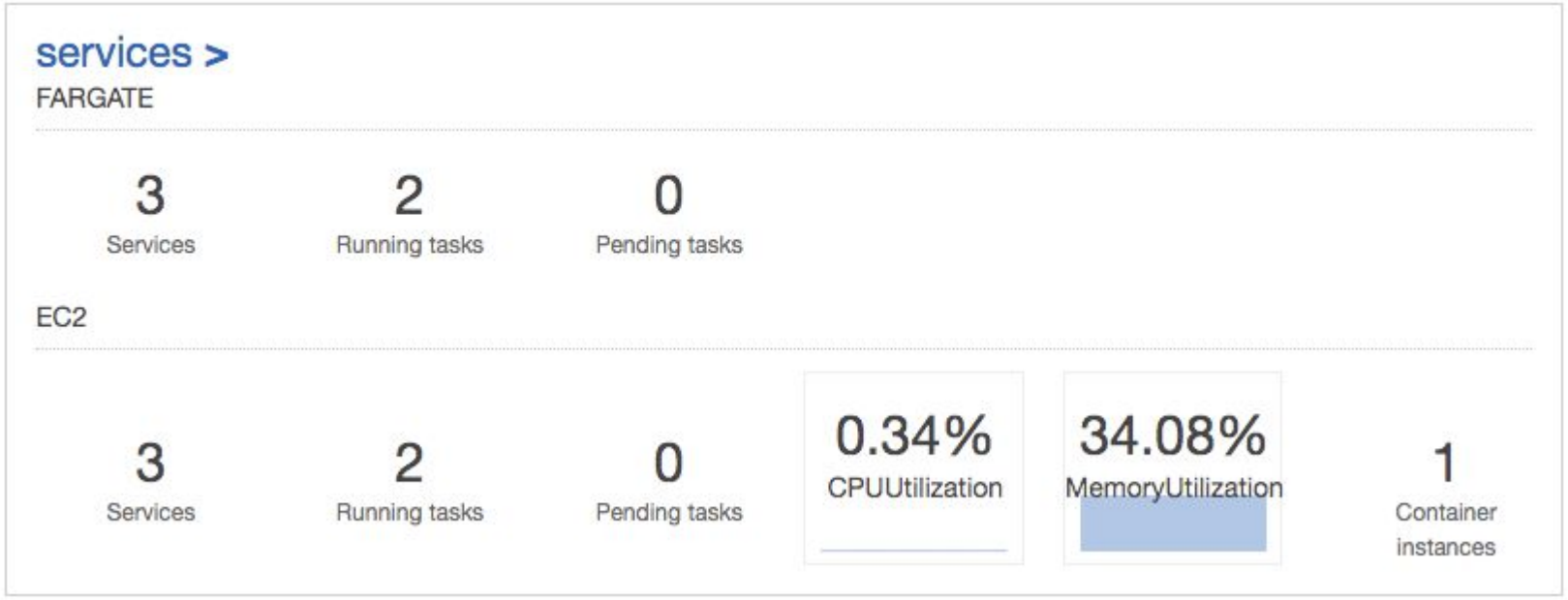
Basics: Task

A container instance

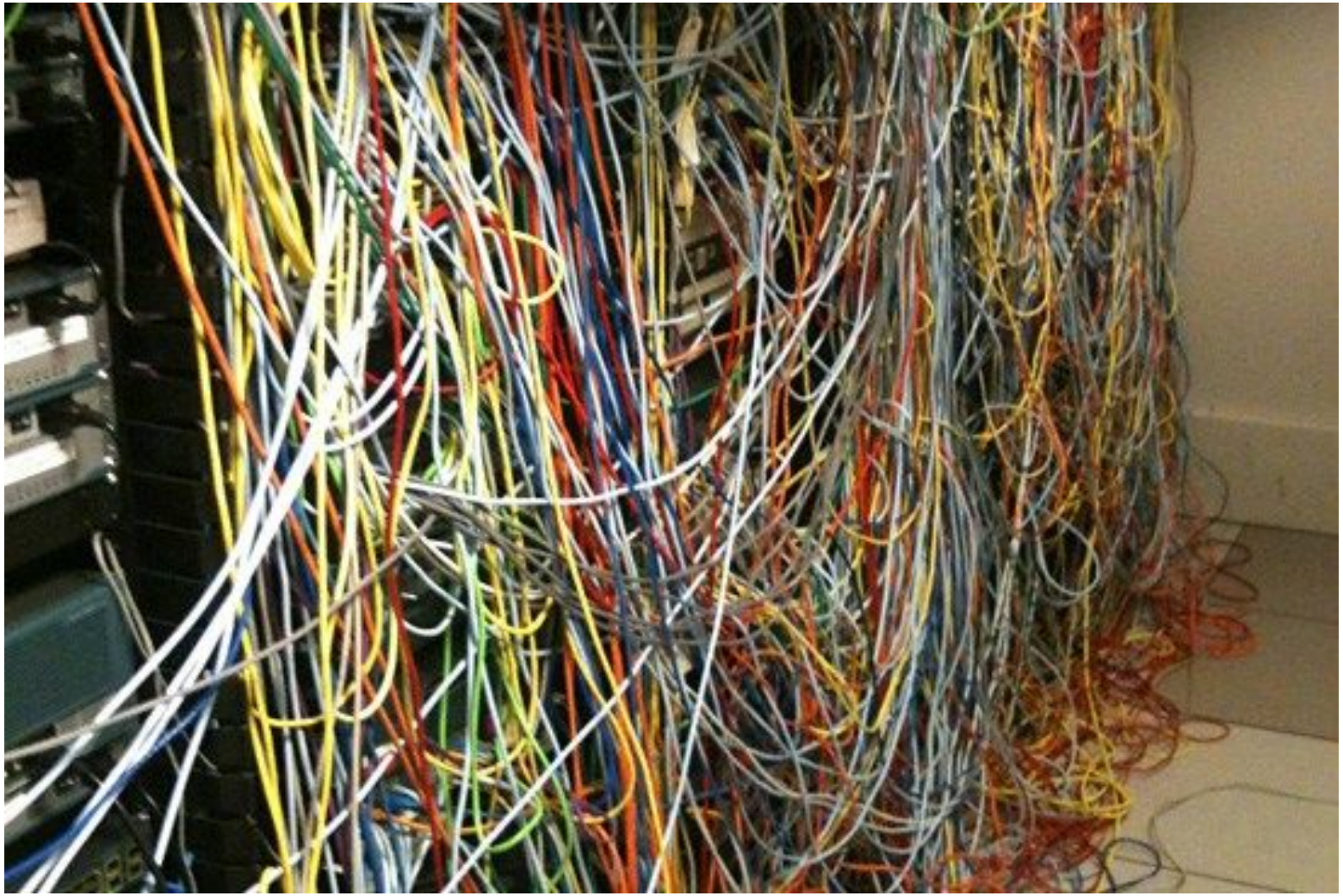
- Running with the task-definition settings
- Container Network information
- Logs (if using awslogs driver)

Basics: Cluster

- Services
- Tasks
- Ec2 Instance worker Nodes
- Scheduled Tasks
- Metrics



ECS Networking



ECS Networking

Bridge (ec2)

- Use same IP as host
- Ephemeral ports
- ALB TargetGroup Type = Instance

Host (ec2)

- All containers share same stack

AWSVPC (ec2/fargate)

- Use Dedicate ENI with it’s own IP and security Group
- Can have public IP
- Can’t have Elastic IP (yet)
- ALB TargetGroup Type = IP

ECS Networking ENI

AWSVPC on EC2 Based nodes has a limitation on the ENI per instance-type

https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/using-eni.html

t3.nano	2
t3.micro	2
t3.small	2
t3.medium	3
t3.large	3
t3.xlarge	4
t3.2xlarge	4

m5.large	3
m5.xlarge	4
m5.2xlarge	4
m5.4xlarge	8
m5.12xlarge	8
m5.24xlarge	15

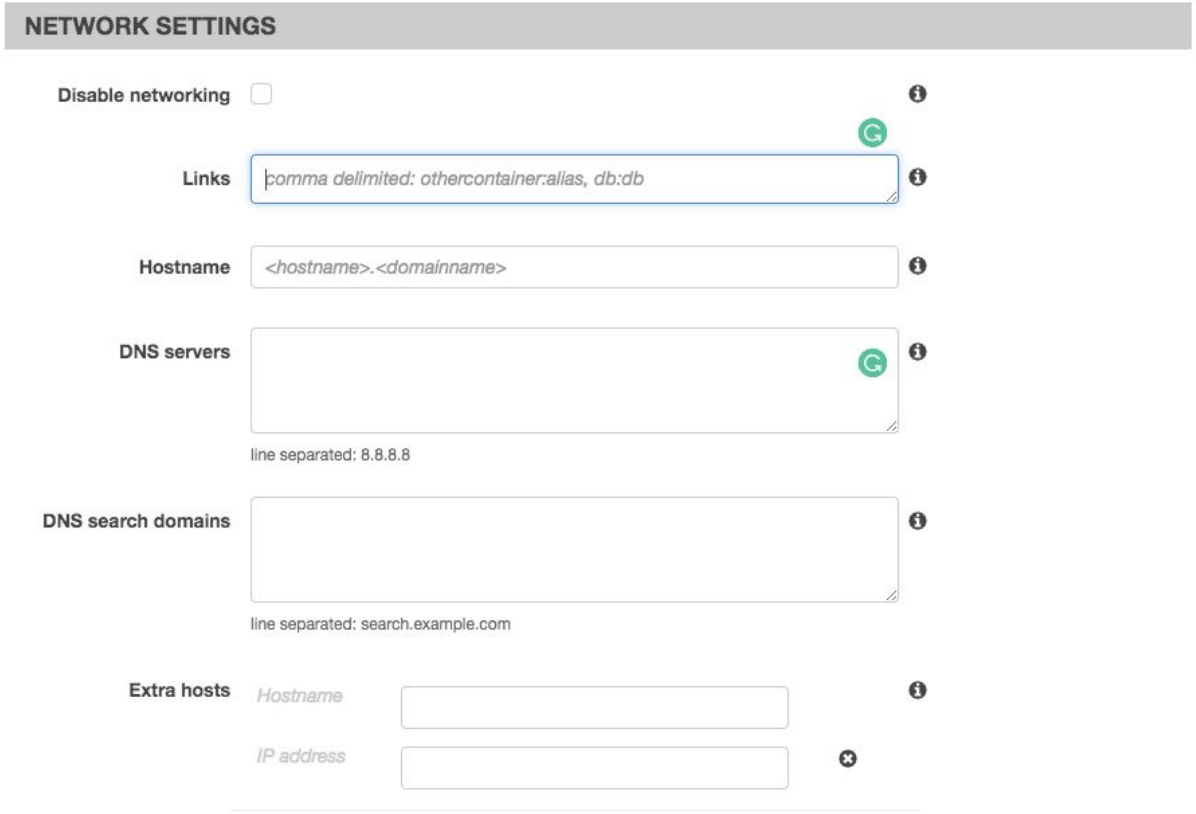
Amazon EKS

ECS Networking

There is no private Docker Network with hostnames

Allows Links between Containers

AWS Service Discovery via Route53 (awsvpc only)



ECS IAM Security

EC2 Based ecs uses by default the EC2 Instance Role

Task Roles

- Similar as ec2 but assume-role = ecs-tasks.amazonaws.com

Task Execution Role (fargate)

- AmazonECSTaskExecutionRolePolicy

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ecr:GetAuthorizationToken",
        "ecr:BatchCheckLayerAvailability",
        "ecr:GetDownloadUrlForLayer",
        "ecr:BatchGetImage",
        "logs:CreateLogStream",
        "logs:PutLogEvents"
      ],
      "Resource": "*"
    }
  ]
}
```

ECS Storage

EC2 Based

- Via volume options via under laying instance
- Using Docker Volume Plugins. Mount EFS,EBS,S3
  - Rexray
- Running as Privileged and use Fuse

Fargate

- …..

https://aws.amazon.com/about-aws/whats-new/2018/08/amazon-ecs-now-supports-docker-volume-and-volume-plugins/
https://github.com/rexray/rexray

Compare EC2/Fargate

EC2 Based

- Persistence Storage
- Privileged Mode
- Autoscaling limited
- Cheaper with a lot of small containers

Fargate

- Fully managed
- Auto Scale up and down
- More Expensive than small EC2

Container Deployments

Combination Terraform -> image retagging.

Tag: latest to the production container

- Jenkins
  - Script: https://github.com/silinternational/ecs-deploy
  - Standard Plugin
- CodePipeline
  - Code Deploy
  - Cloudformation
- Drone

Jenkins Build Cluster

Use the Amazon EC2 Container Service Plugin to create clean On-Demand Slaves on ECS

Caveats: Run Docker inside Docker is tricky!!

- Mount the /var/run/docker.sock from ec2 into ecs (in jenkins config)
- The docker tool talks to the daemon, volumes mapping are relative to the ec2 not the docker instance.
- Mount your workspace into the docker also.

/var/run/docker.sock -> /var/run/docker.sock
/home/jenkins/workspace/ -> /home/jenkins/workspace/
Now you can run `docker -v /home/jenkins/workspace/mydir:/mydir` because the directory is the same

https://wiki.jenkins.io/display/JENKINS/Amazon+EC2+Container+Service+Plugin

Demo

Demo

Terraform

A webserver on a public IP that writes the client and server IP to a dynamoDB database and shows the content of the database.

- ECR Repository
- ECS Cluster
- ECS Service + Task definition
- DynamoDB
- IAM Role to access DynamoDB
- Security Group

Terraform Variables

```
variable "ecs_task" {
  default = {
    service       = "demo"
    family        = "demo"
    cpu           = "256"
    memory        = "512"
    desired_count = "1"
    cluster       = "demo"
    host_port      = 8080
    container_port = 8080
  }
}
```

Thank you!

Questions?

¡We're hiring!

- OpenStack DevOps Engineer @ BCN
- Senior Devops Engineer @ BCN
- Senior PHP Front End Developer @ BCN
- Site Reliability Engineer (SRE) @ BCN

More info:
https://edrans.applytojob.com/apply