:iTechArt

Software
Engineering
Services

JUN 16, 2018

Deployment:
Fast, Fine, Affordable

Alex Burym

Hi there.
I’m Alex Burym

Department: D2 G1

- 9+ years of diverse experience in IT
- CI/CD
- DevOps
- Cloud Computing
- Networking
- Process Automation

Contact Information:

Email: alex.burym@itechart-group.com
Skype: live:eng.zubr

Plan

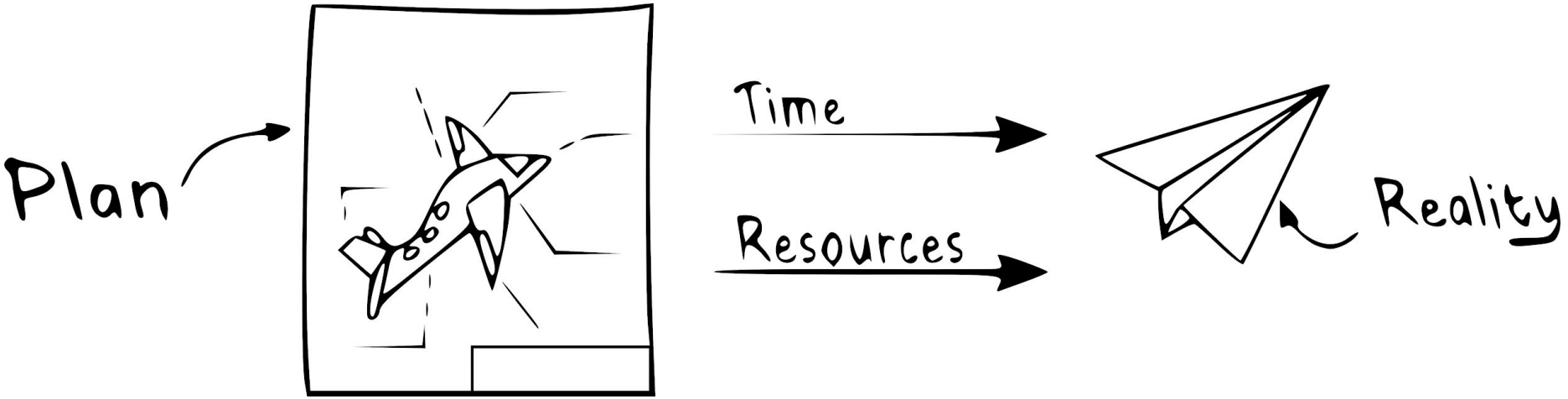
- What about this Infra, anyhow?
- Cloud basics!
- Infrastructure as a Code?
- Docker Docker Docker?
- Continuous Integration/Deployment/Delivery?
- Why ECS?
- Let’s dive!

What about this Infra, anyhow?

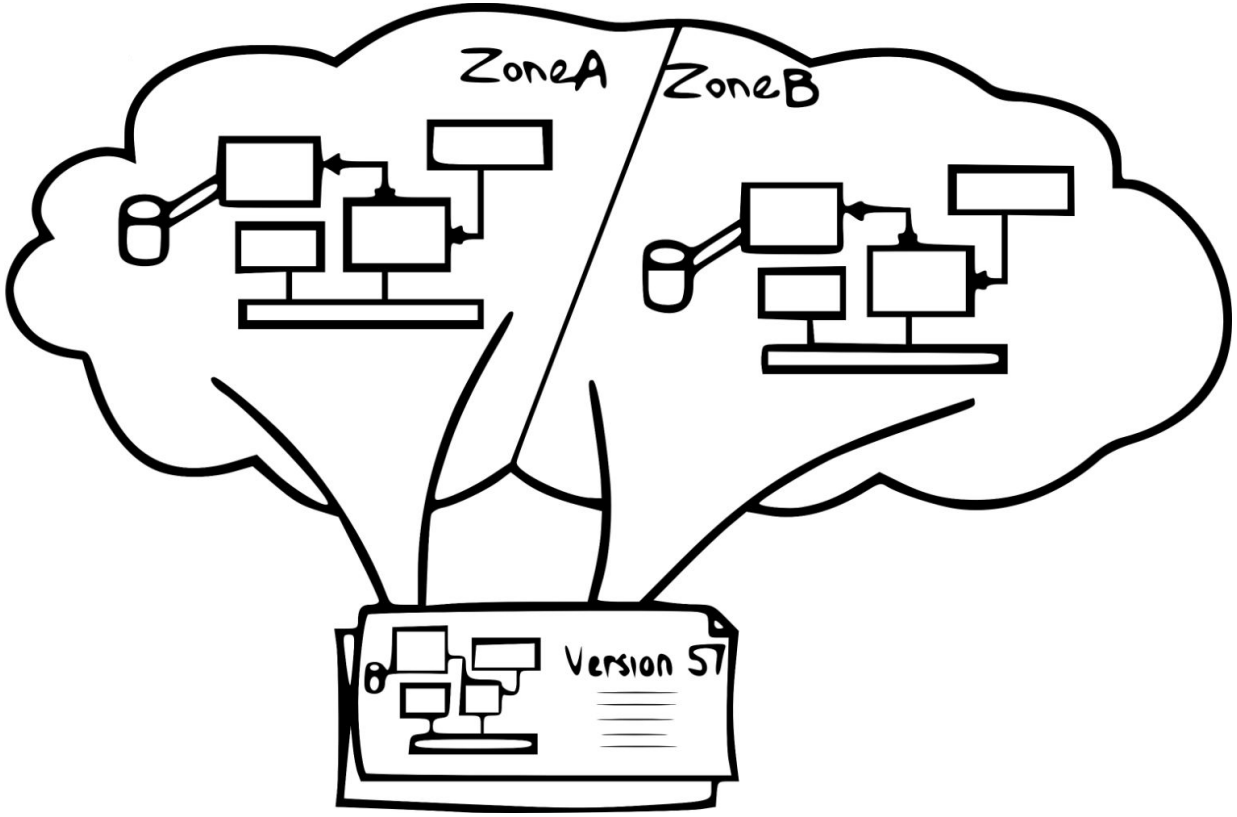
What about this Infra, anyhow?

What about this Infra, anyhow?

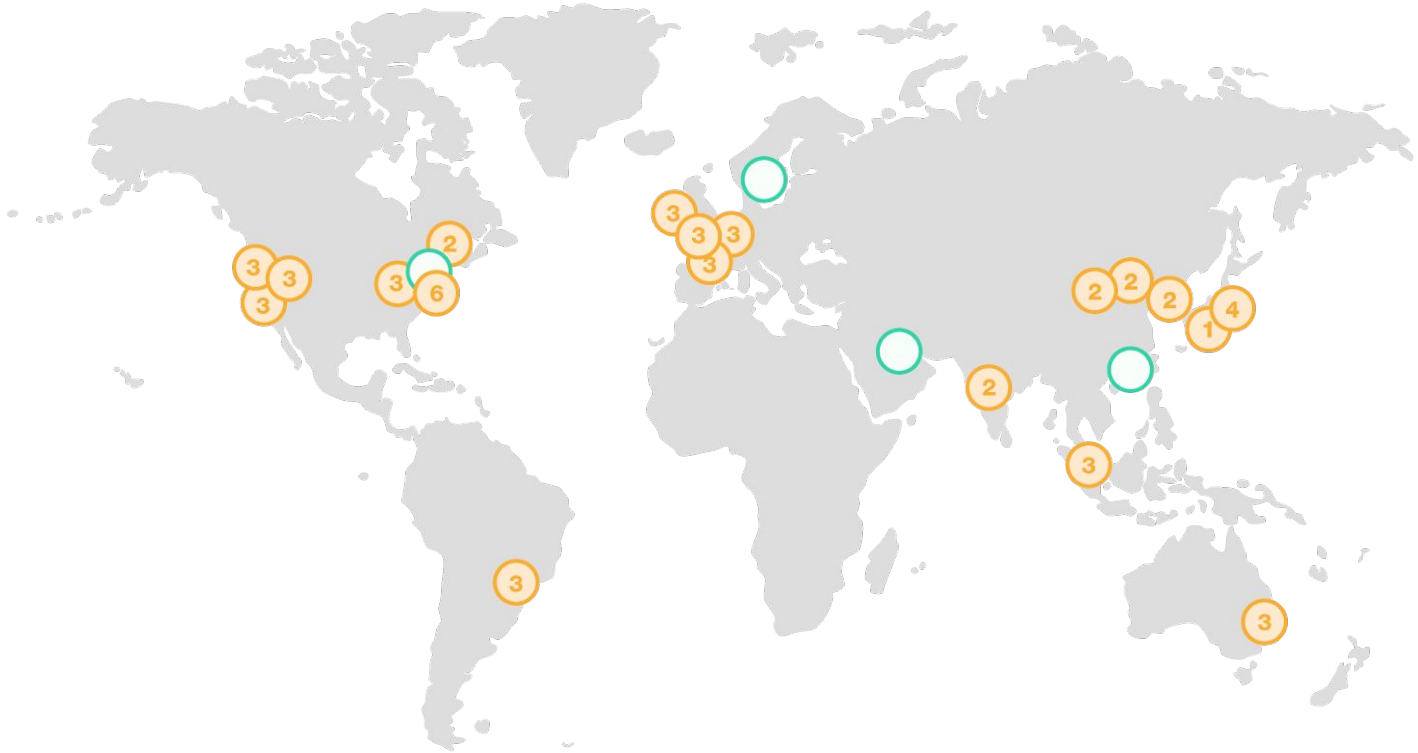
What about this Infra, anyhow?



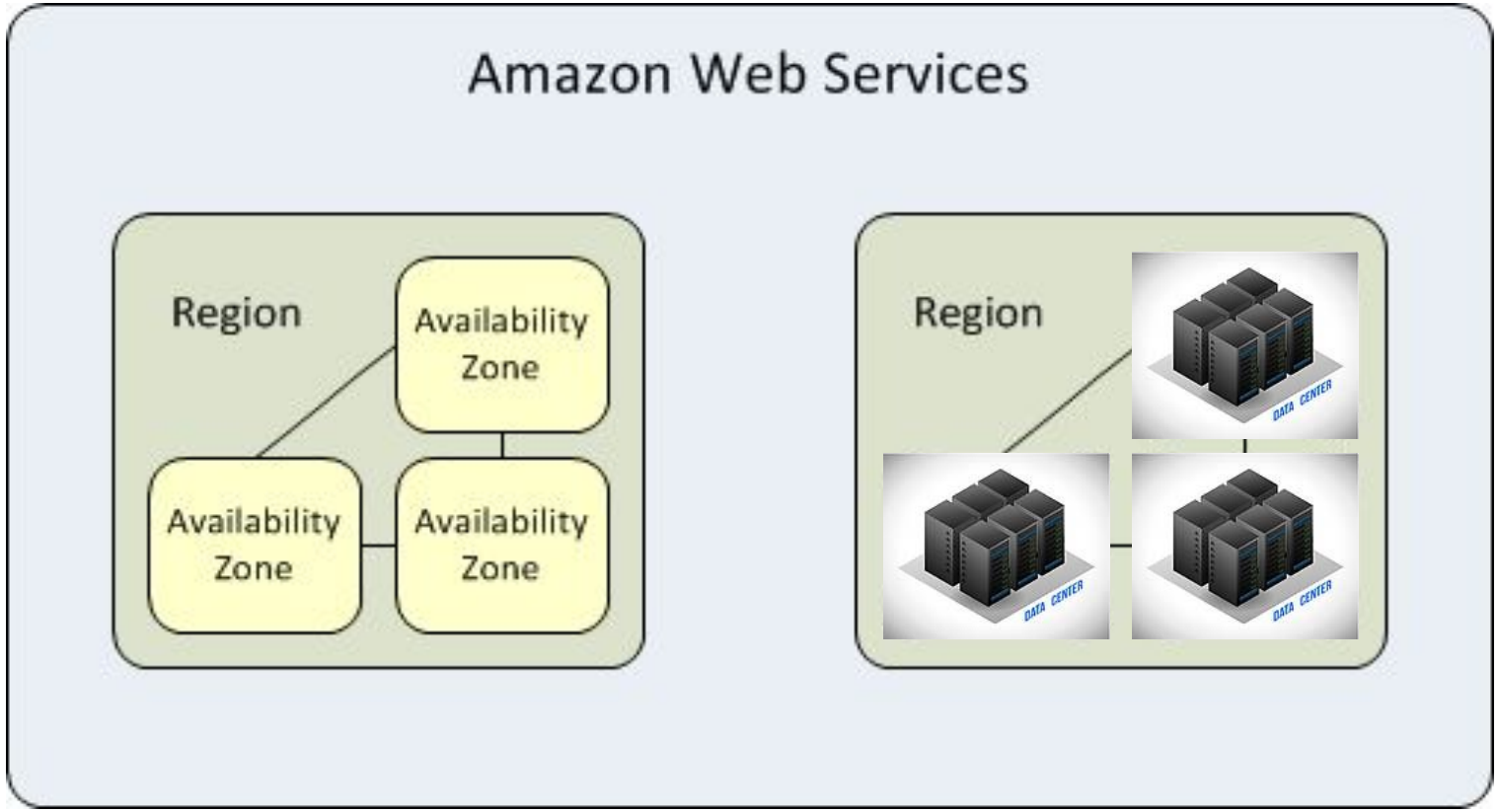
What about this Infra, anyhow?



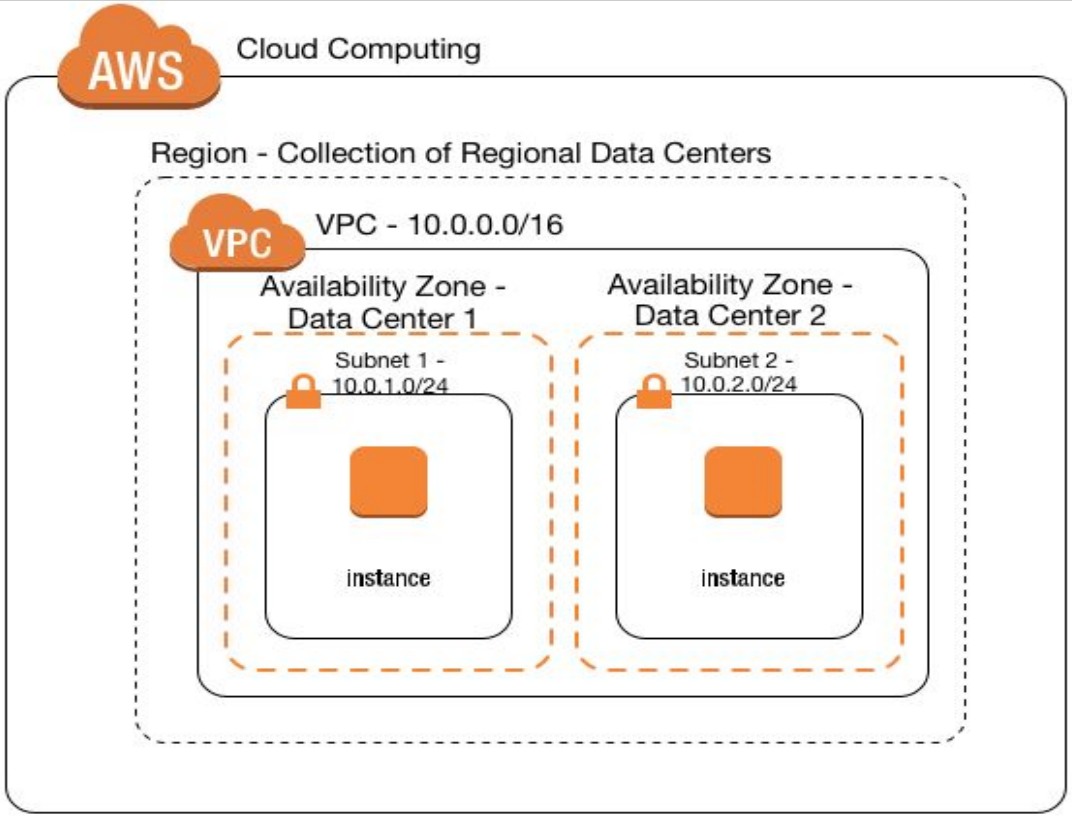
Cloud basics!



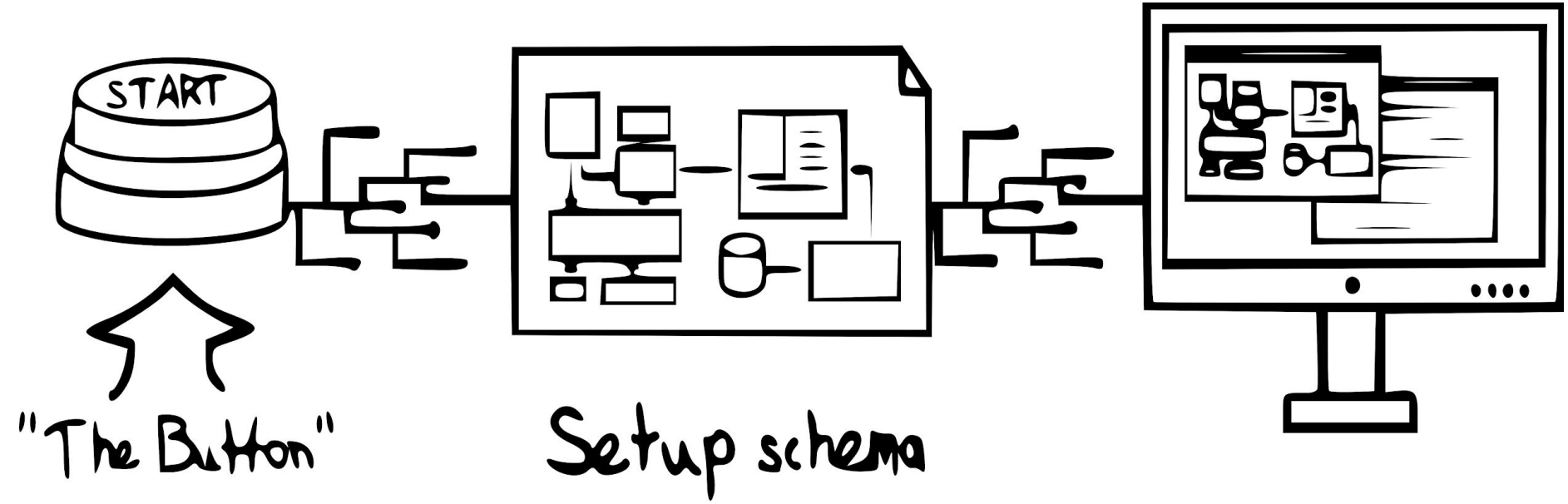
Cloud basics!



Cloud basics!



What about this Infra, anyhow?



IaC?

AWS CloudFormation

Infrastructure Build Tools

See AWS CloudFormation alternatives

Terraform

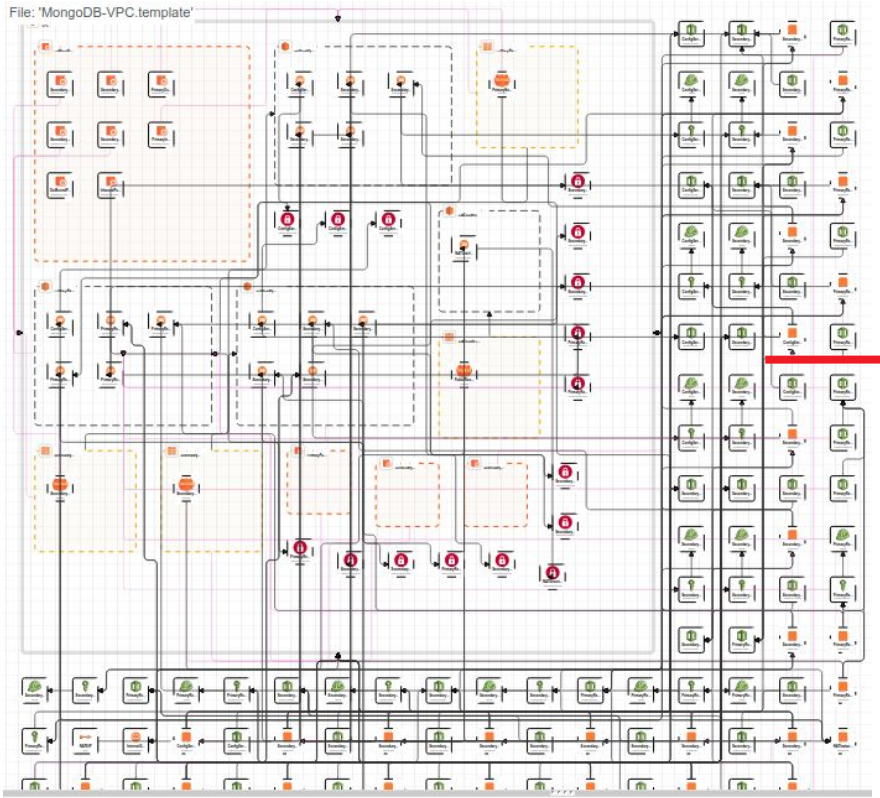
Infrastructure Build Tools

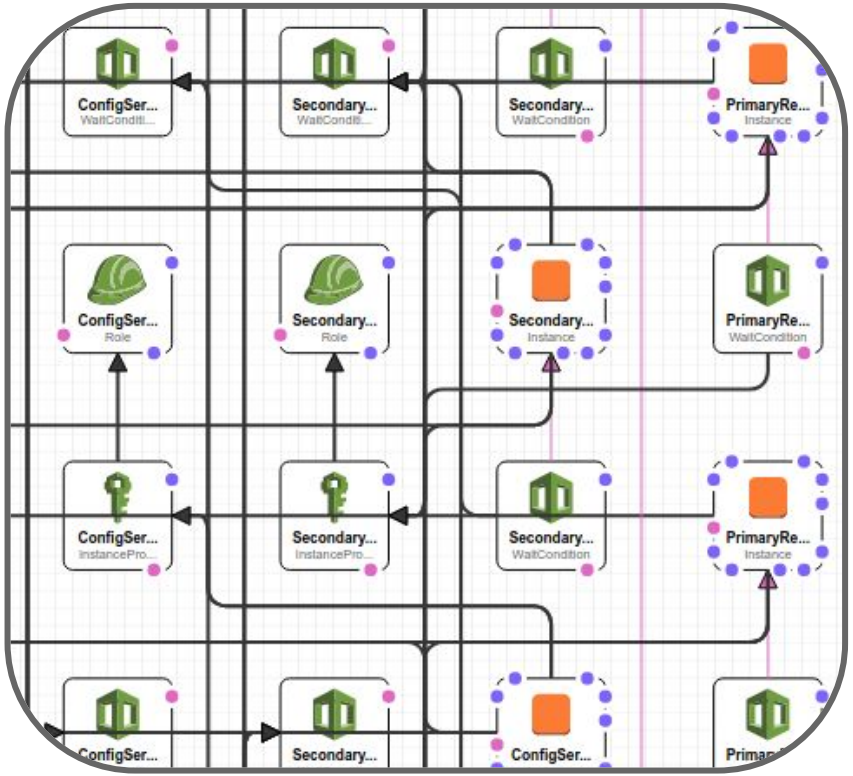
See Terraform alternatives

CloudFormation?

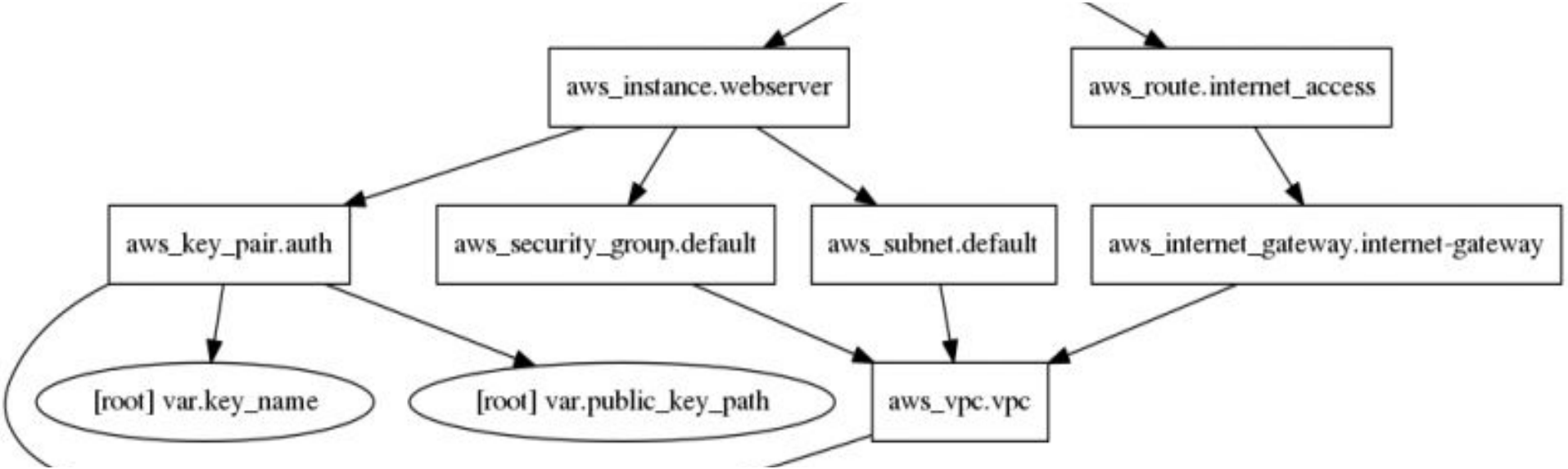

```
8965                                    "Ref": "VPC"
8966                                }
8967                            ]
8968                        ]
8969                    },
8970                    " >> config.sh\n",
8971                    "echo ",
8972                    {
8973                        "Fn::Join": [
8974                            "",
8975                            [
8976                                "export MongoDBVersion=",
8977                                {
8978                                    "Ref": "MongoDBVersion"
8979                                }
8980                            ]
8981                        ]
8982                    },
8983                    " >> config.sh\n",
8984                    "./init.sh > install.log 2>&1 \n",
8985                    "#  Cleanup \n",
8986                    "#rm -rf *\n",
8987                    "# All is well so signal success\n",
8988                    "/opt/aws/bin/cfn-signal -e 0 -r \"MongoDB
8989                    {
8990                        "Ref": "ConfigServer2WaitForNodeInstall
8991                    },
8992                    "'\n"
8993                ]
8994            ]
8995        }
8996    },
8997    "InstanceType": {
8998        "Ref": "ConfigServerInstanceType"
8999    }
9000  },
9001  "Condition": "CreateMinOneShard"
9002 }
9003 }
```

CloudFormation?





Terraform?


aws_instance.webserver
aws_route.internet_access
aws_key_pair.auth
aws_security_group.default
aws_subnet.default
aws_internet_gateway.internet-gateway
[root] var.key_name
[root] var.public_key_path
aws_vpc.vpc

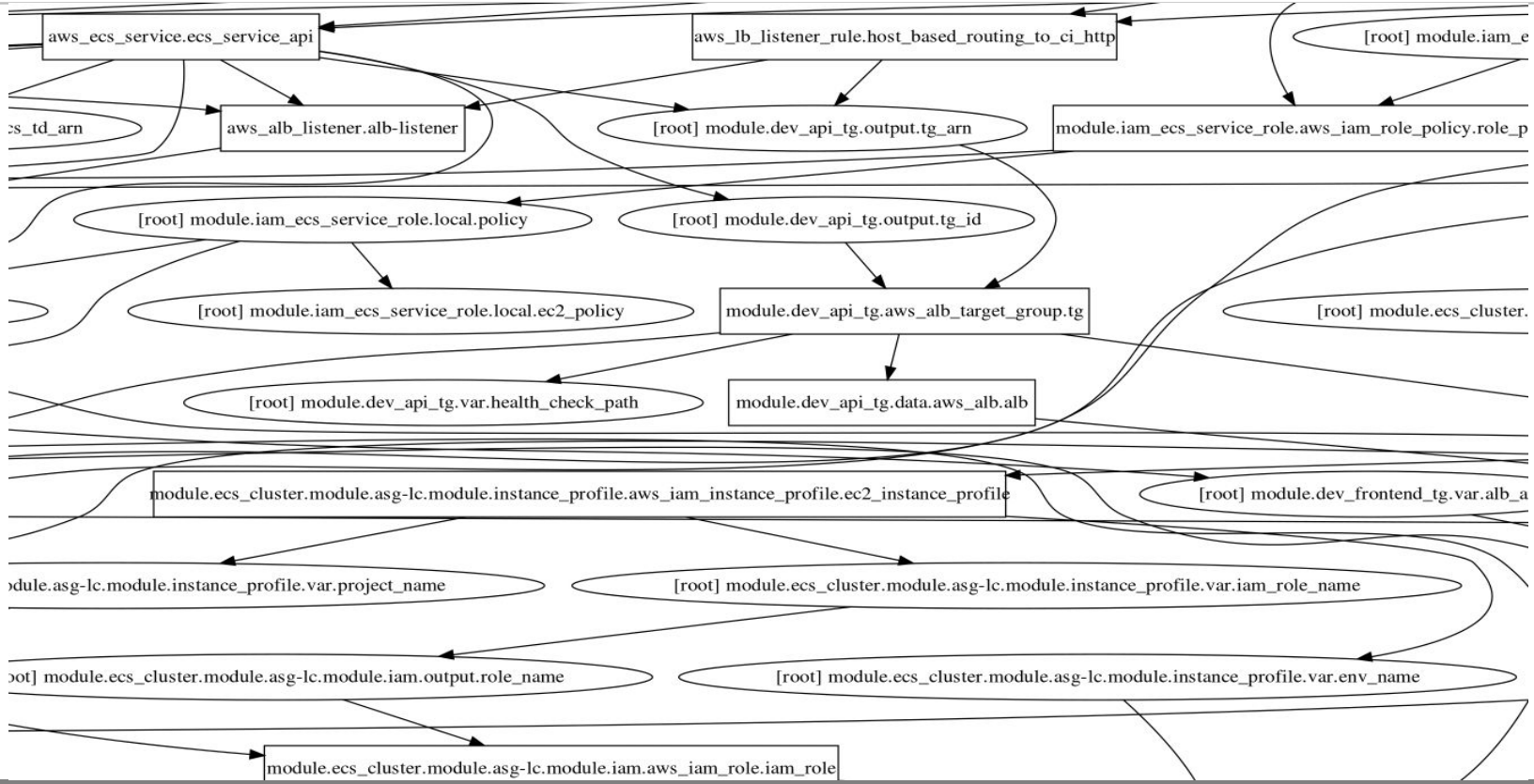
Terraform?


```
provider "aws" {
  region = "${var.region}"
}

resource "aws_instance" "webserver" {
    ami = "ami-405f7226"
    instance_type = "t2.nano"
}

resource "aws_security_group" "default" {
  name = "terraform_securitygroup"
  description = "Used for public instances"
  vpc_id = "${var.vpc_id}"
```

Terraform?


```
  ~ aws_instance.firstec2
      tags.%:    "0" => "1"
      tags.Name: "" => "FirstEC2TF!"

Plan: 0 to add, 1 to change, 0 to destroy.

+ azurerm_resource_group.production
    location: "" => "westus"
    name:     "" => "production"

+ azurerm_virtual_network.network

Plan: 2 to add, 0 to change, 0 to destroy.
```

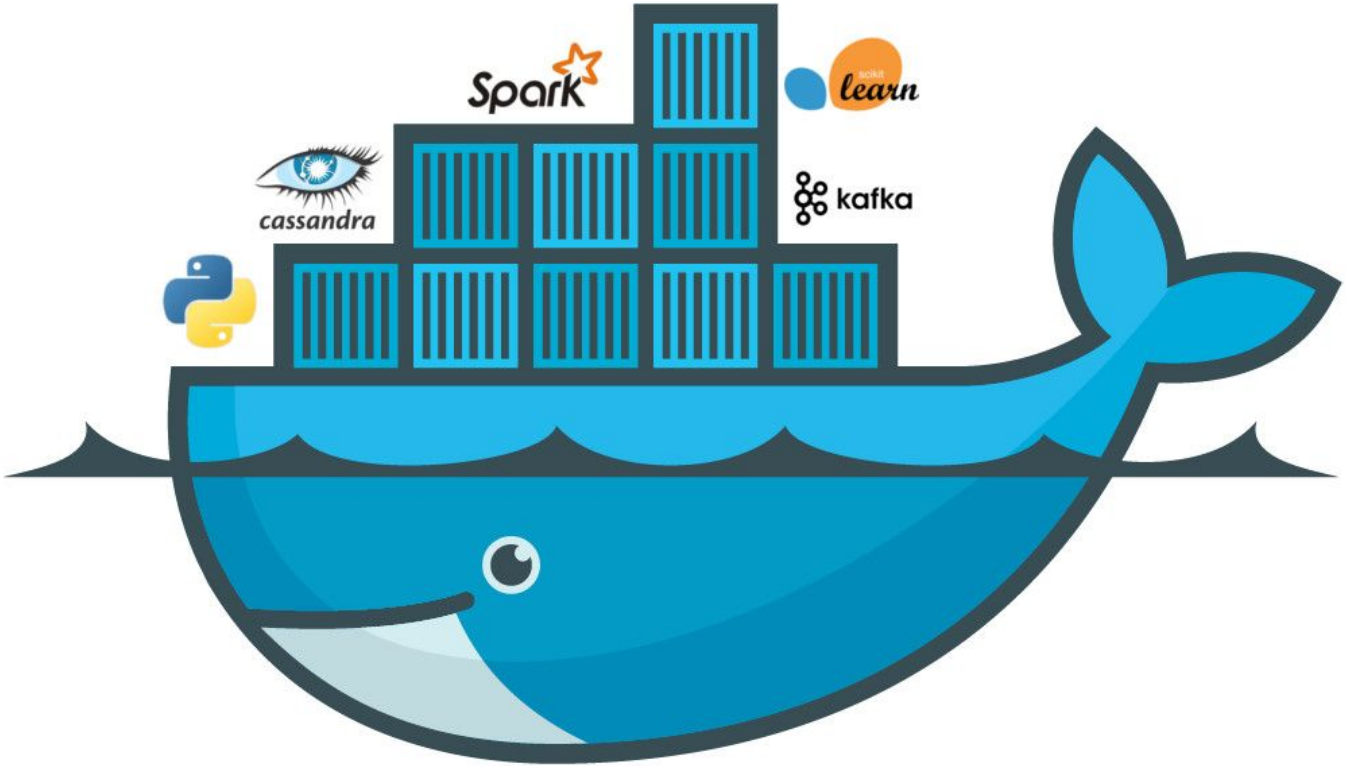
Terraform?


aws_ecs_service.ecs_service_api
aws_lb_listener_rule.host_based_routing_to_ci_http
[root] module.iam_e
aws_alb_listener.alb-listener
[root] module.dev_api_tg.output.tg_arn
module.iam_ecs_service_role.aws_iam_role_policy.role_p
[root] module.iam_ecs_service_role.local.policy
[root] module.dev_api_tg.output.tg_id
[root] module.iam_ecs_service_role.local.ec2_policy
module.dev_api_tg.aws_alb_target_group.tg
[root] module.ecs_cluster.
[root] module.dev_api_tg.var.health_check_path
module.dev_api_tg.data.aws_alb.alb
module.ecs_cluster.module.asg-lc.module.instance_profile.aws_iam_instance_profile.ec2_instance_profile
[root] module.dev_frontend_tg.var.alb_a
[root] module.ecs_cluster.module.asg-lc.module.instance_profile.var.iam_role_name
[root] module.ecs_cluster.module.asg-lc.module.instance_profile.var.env_name
module.ecs_cluster.module.asg-lc.module.iam.aws_iam_role.iam_role

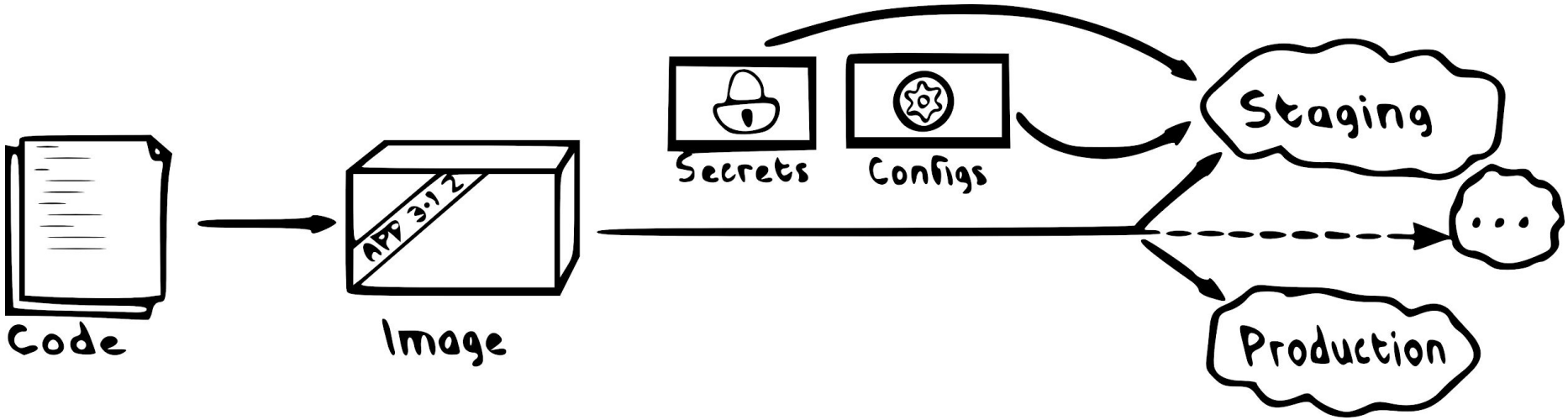
Demo Time

➔ Terraform ...

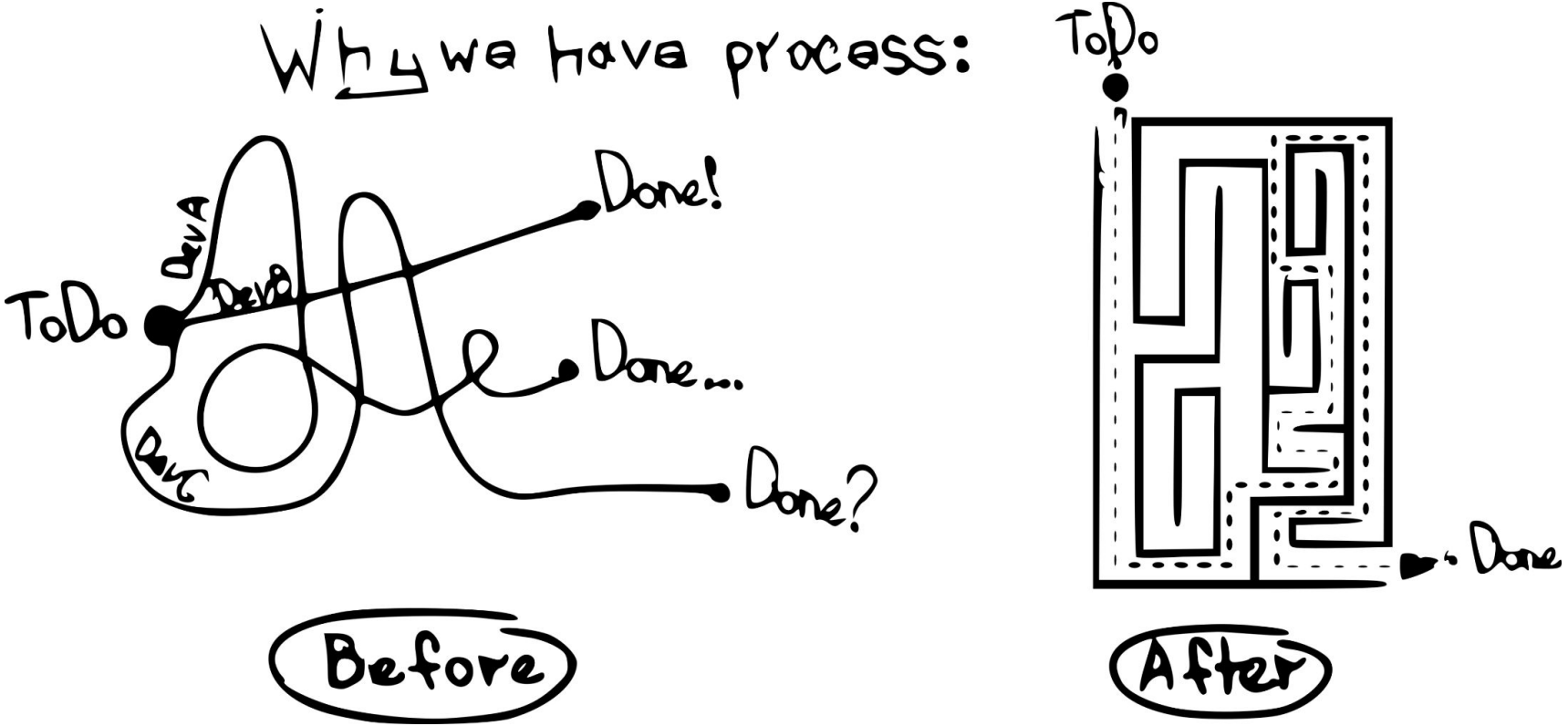
Docker?



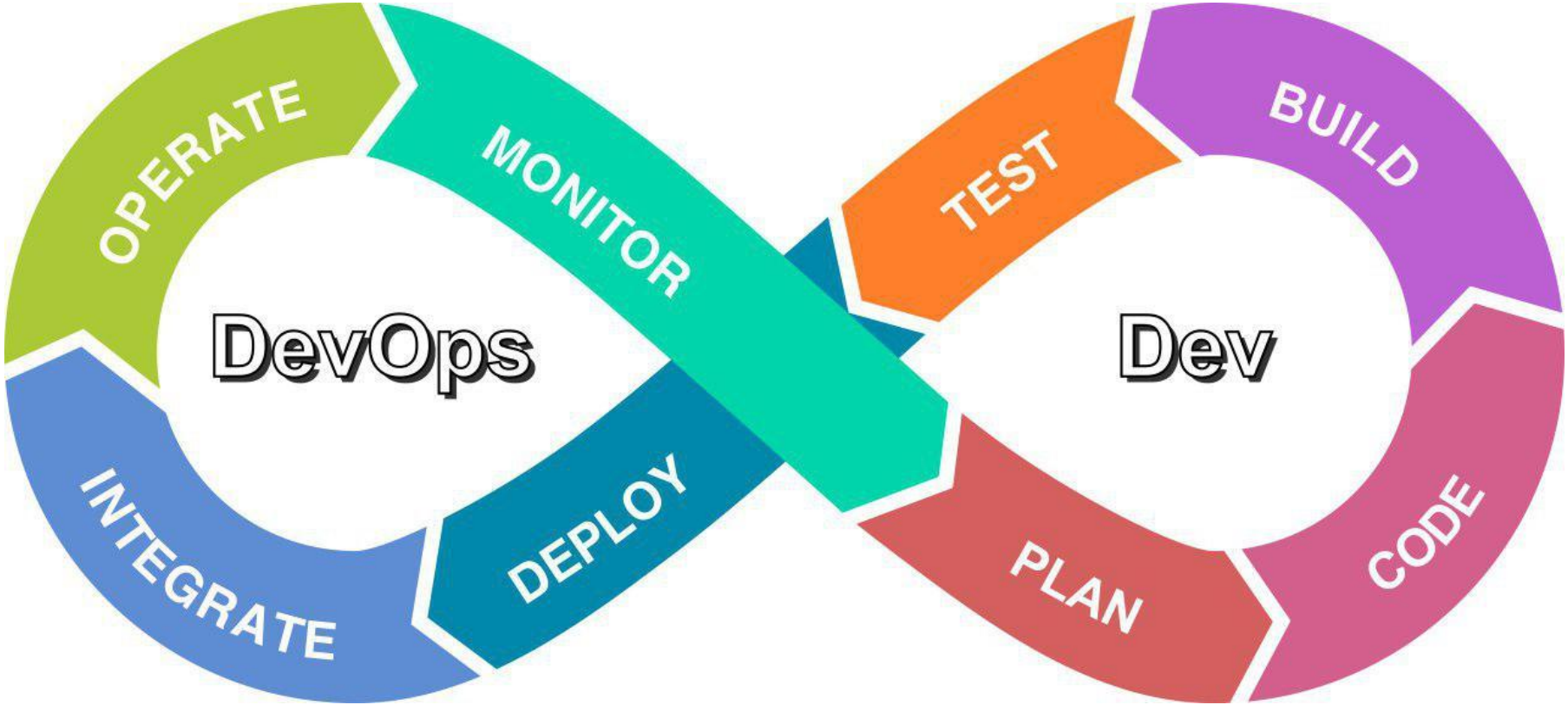
Docker?



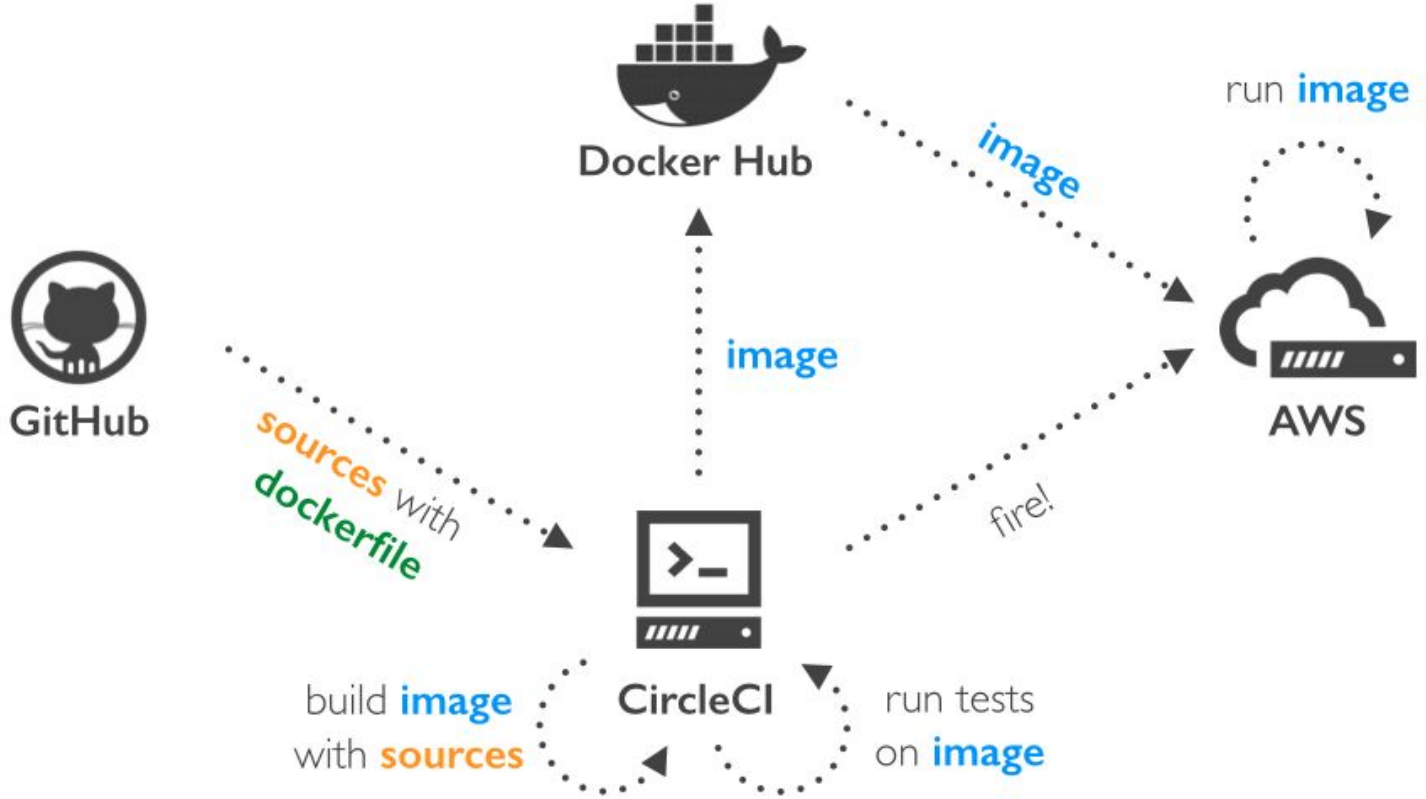
How?


Why we have process:
ToDo
Dev A
Dev B
Dev C
Done!
Done...
Done?
Before
ToDo
Done
After

How?


OPERATE
MONITOR
TEST
BUILD
DevOps
Dev
INTEGRATE
DEPLOY
PLAN
CODE

CircleCI?

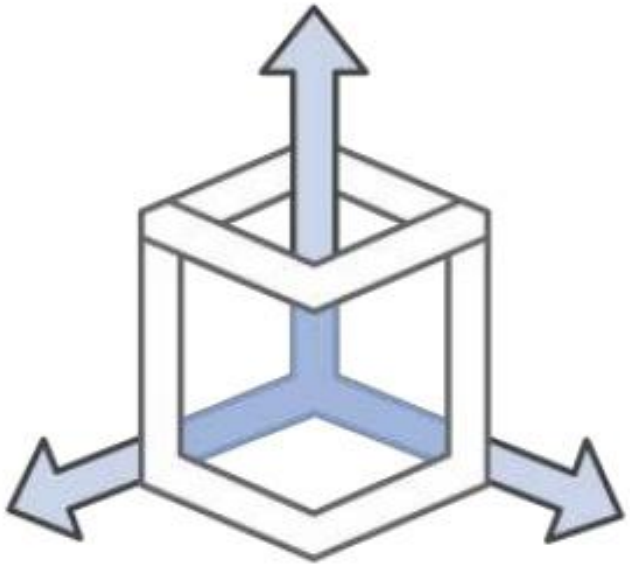

Docker Hub
run image
image
image
GitHub
sources with
dockerfile
AWS
fire!
build image
with sources
CircleCI
run tests
on image

CircleCI?

Following is a sample 2.0 `.circleci/config.yml` file.

```
version: 2
jobs:
  build:
    docker:
      - image: circleci/<language>:<version TAG>
    steps:
      - checkout
      - run: <command>
```

Why ECS?



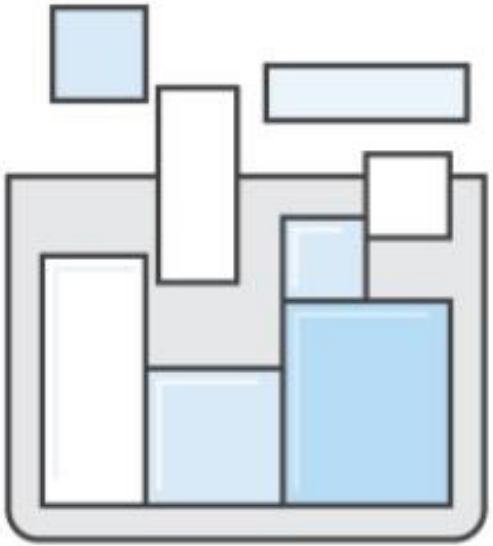
Nothing to run

Complete state

Control and monitoring

Scale

Why ECS?

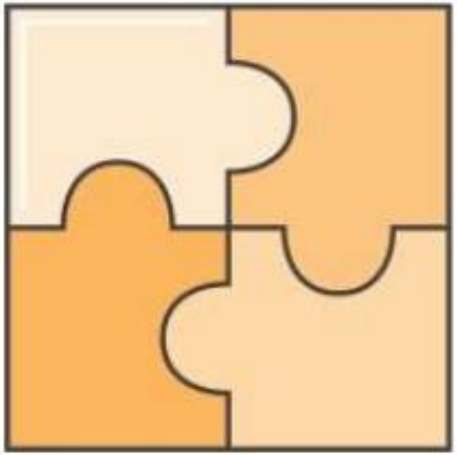


Applications

Batch jobs

Multiple schedulers

Why ECS?



Elastic Load Balancing

Amazon Elastic Block Store

Amazon Virtual Private Cloud

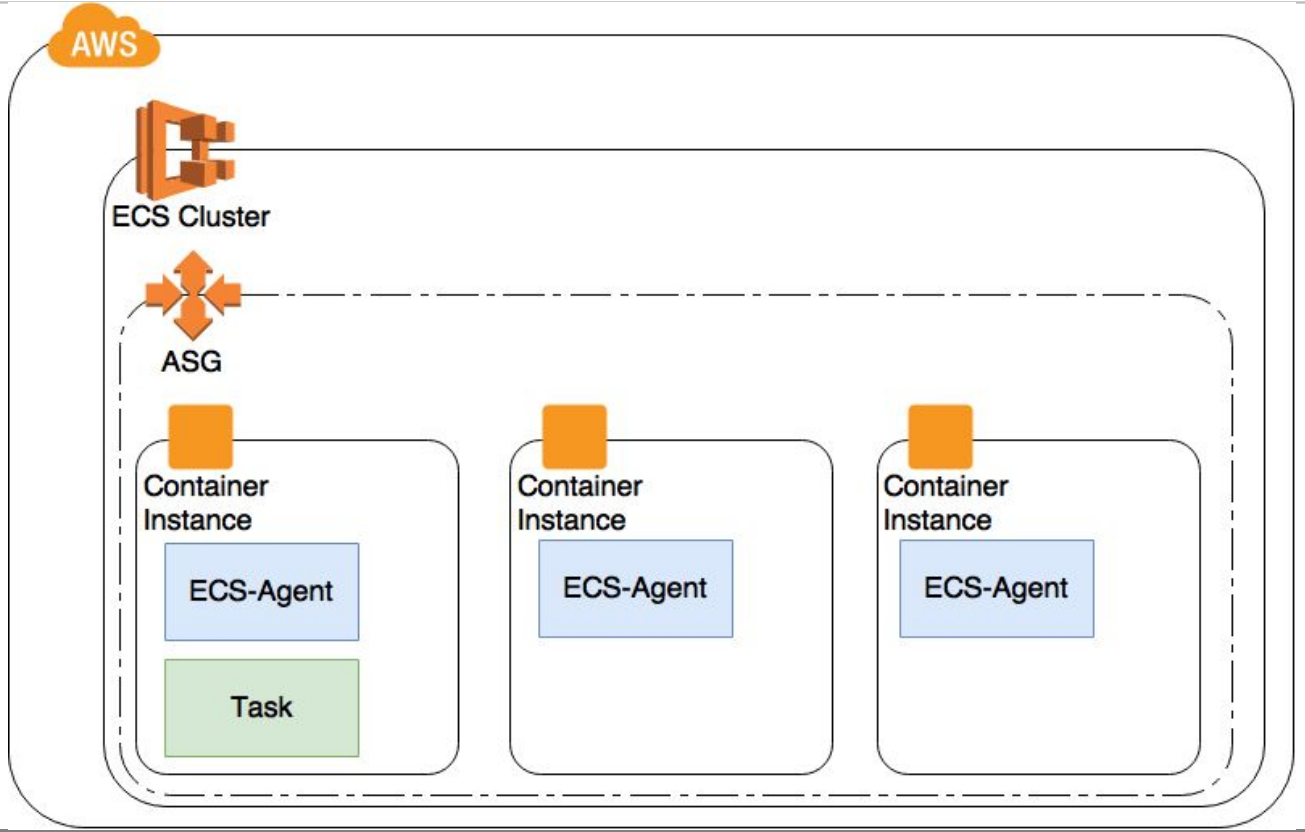
Amazon CloudWatch

AWS Identity and Access Management

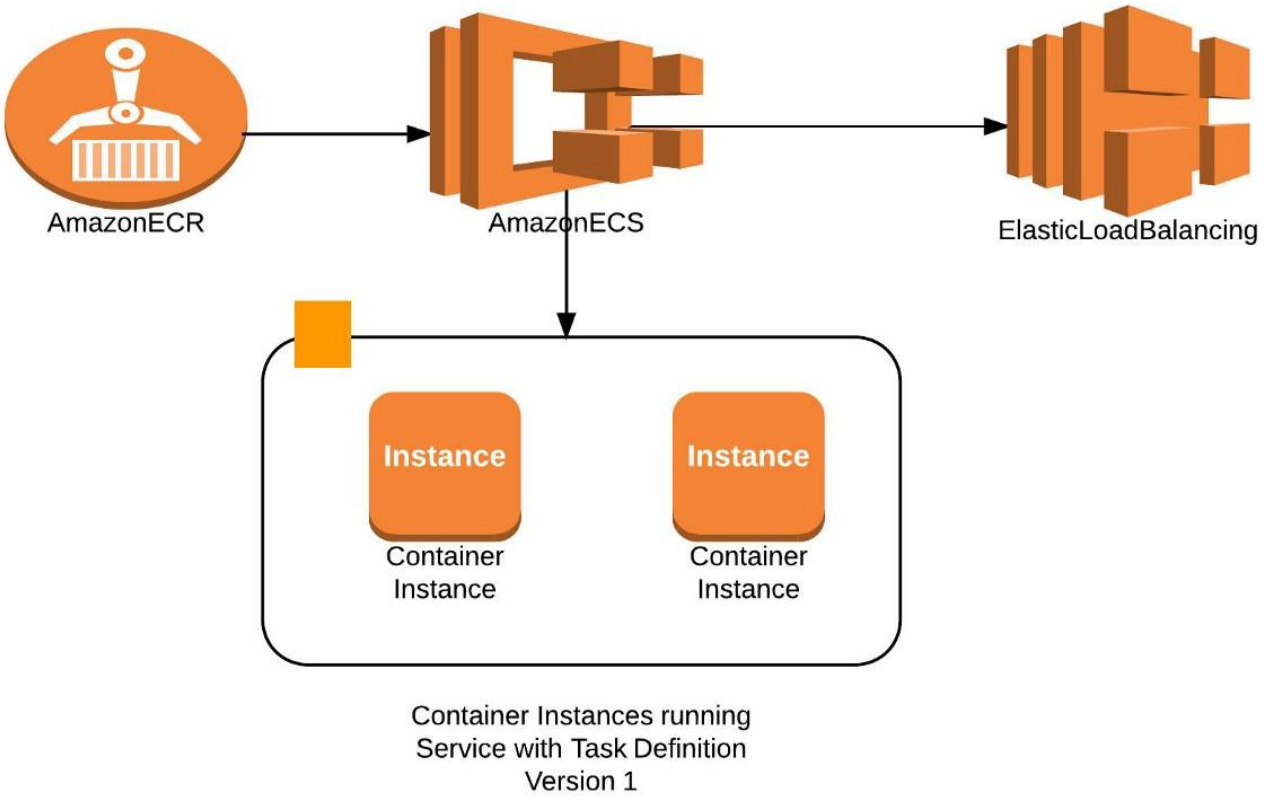
AWS CloudTrail

Let’s dive!

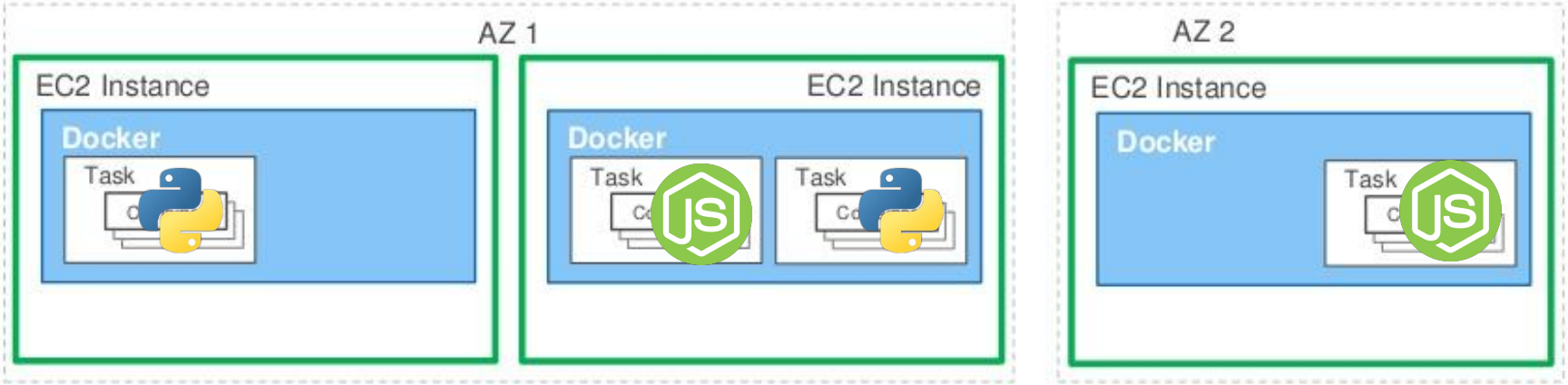
Let’s dive!


AWS
ECS Cluster
ASG
Container Instance
ECS-Agent
Task
Container Instance
ECS-Agent
Container Instance
ECS-Agent

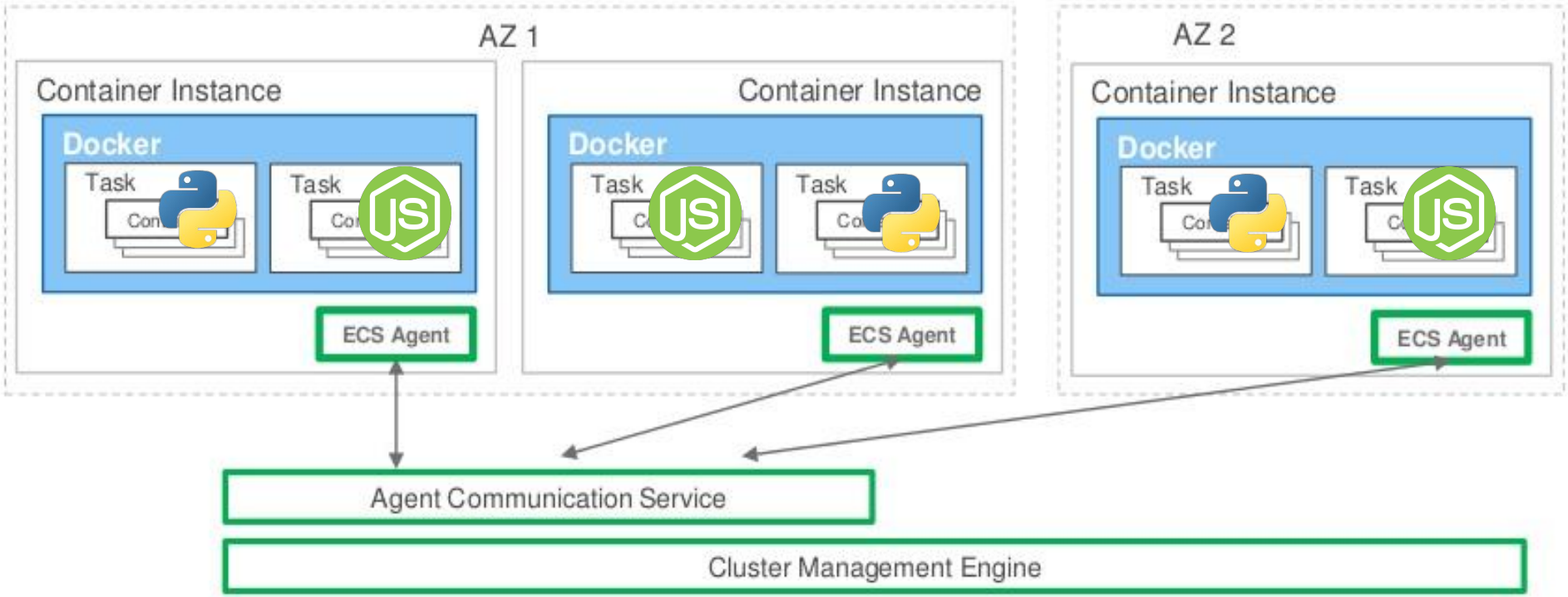
Let’s dive!


AmazonECR
AmazonECS
ElasticLoadBalancing
Instance
Container
Instance
Instance
Container
Instance
Container Instances running
Service with Task Definition
Version 1

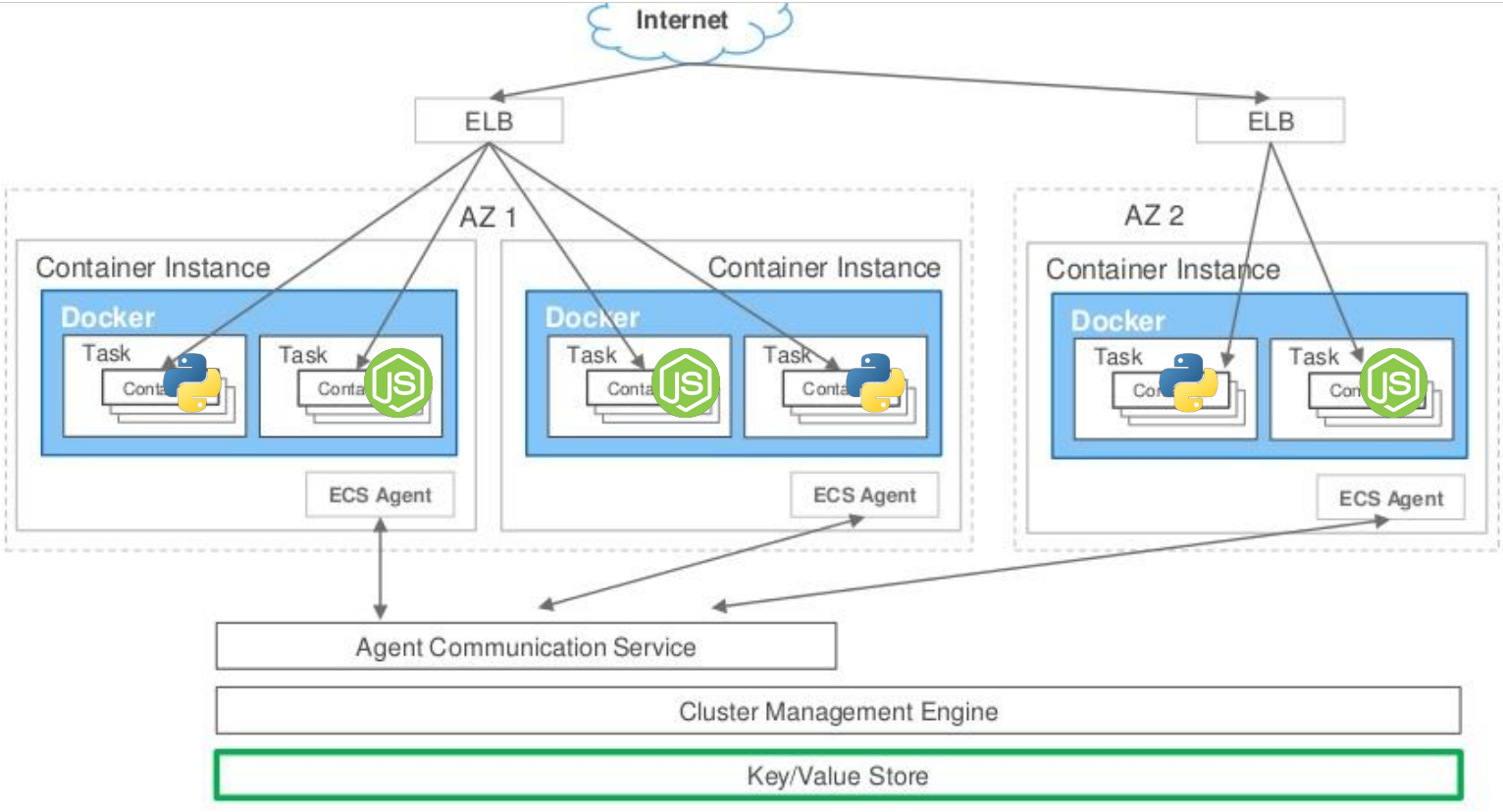
Let’s dive!


AZ 1
EC2 Instance
Docker
Task
EC2 Instance
Docker
Task
Task
AZ 2
EC2 Instance
Docker
Task

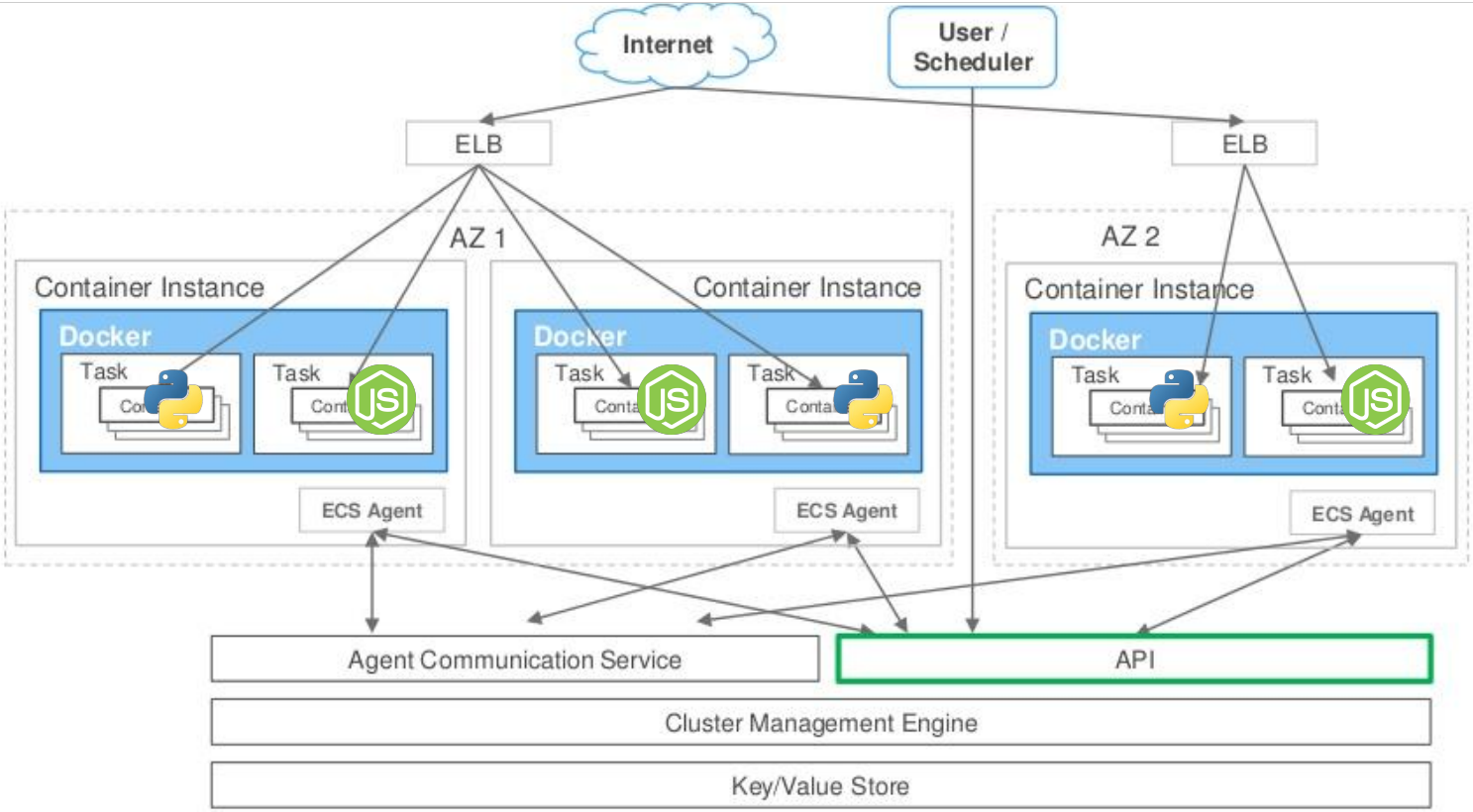
Let’s dive!


AZ 1
AZ 2
Container Instance
Container Instance
Container Instance
Docker
Docker
Docker
Task
Task
Task
Task
Task
Task
ECS Agent
ECS Agent
ECS Agent
Agent Communication Service
Cluster Management Engine

Let’s dive!


Internet
ELB
ELB
AZ 1
AZ 2
Container Instance
Container Instance
Container Instance
Docker
Docker
Docker
Task
Task
Task
Task
Task
Task
ECS Agent
ECS Agent
ECS Agent
Agent Communication Service
Cluster Management Engine
Key/Value Store

Let’s dive!


Internet
User / Scheduler
ELB
ELB
AZ 1
AZ 2
Container Instance
Container Instance
Container Instance
Docker
Docker
Docker
Task
Task
Task
Task
Task
Task
ECS Agent
ECS Agent
ECS Agent
Agent Communication Service
API
Cluster Management Engine
Key/Value Store

Demo Time

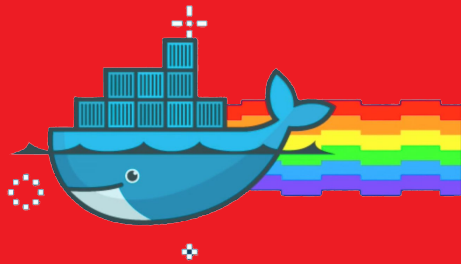
➔ Terraform ...

Useful links

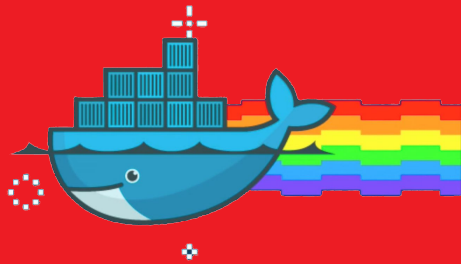
Show notes: https://be34.me/show-notes

1. https://habr.com/post/321810
2. https://cloudonaut.io/aws-security-primer
3. https://github.com/antonbabenko/terraform-best-practices
4. https://stackshare.io/stackups/terraform-vs-aws-cloudformation
5. https://github.com/eng-Zubr/launchpad

Conclusion



May the Cloud be with you!



Thank you!

→ Questions?