



AWS Elastic Beanstalk In Depth

Vancouver Amazon Web Services User Group

Yaroslav Tkachenko
@sap1ens
Director of Engineering, Platform at Bench Accounting



Bench





AWS Elastic Beanstalk

3

Goals for today:

- ✓ You've been using EC2, ELBs, etc., but you don't want to setup anything manually anymore and CF is too complex*
- ✓ You want to run Docker in AWS*
- ✓ You've been using Elastic Beanstalk, but you have some issues (who doesn't?) you want to discuss



AWS Elastic Beanstalk

4

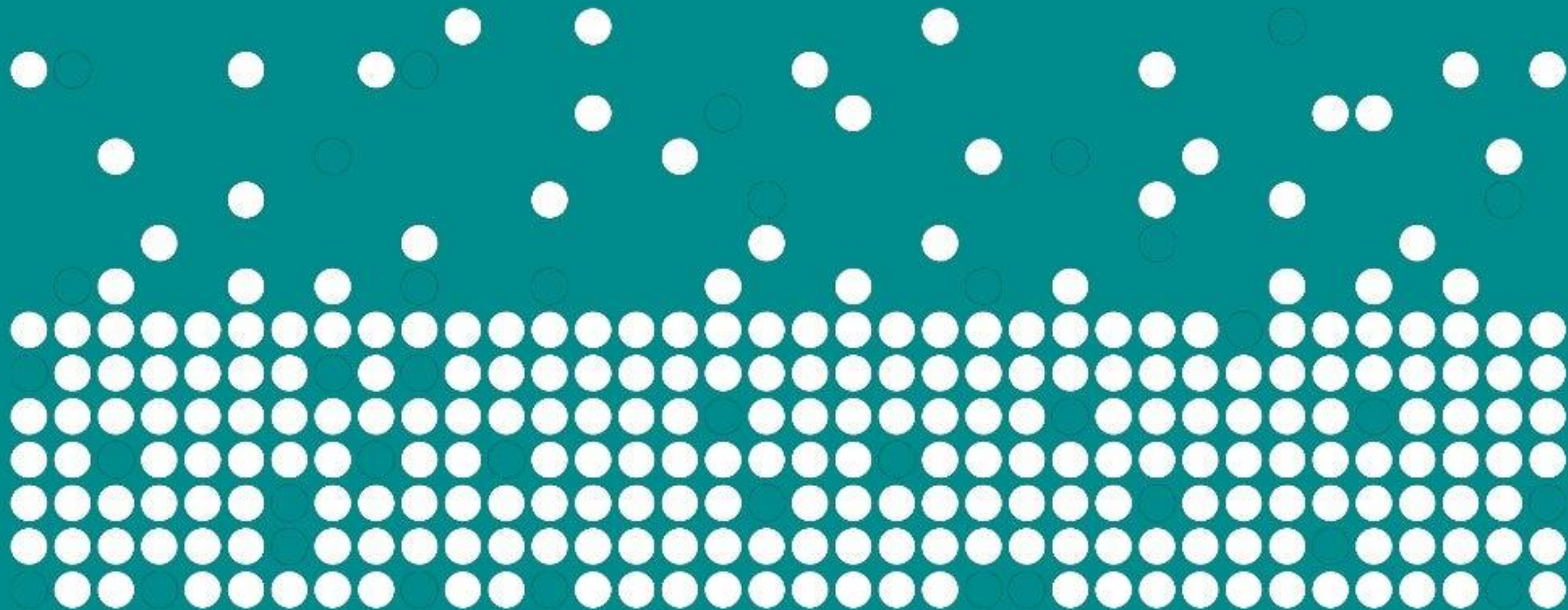
Agenda:

- ✓ Intro
- ✓ Details
- ✓ Our tooling



AWS Elastic Beanstalk - Intro

5





AWS Elastic Beanstalk - Intro

6

Environment = Application(X, Y) | X <- Config, Y <- Version



AWS Elastic Beanstalk - Intro

7

[All Applications](#) > [eventing](#) > [prod-eventing](#) (Environment ID: [e-bk-123456789012](#) URL: [http://elasticbeanstalk.com](#)) [Actions](#)

[Dashboard](#)
[Configuration](#)
[Logs](#)
[Health](#)
[Monitoring](#)
[Alarms](#)
[Managed Updates](#) NEW
[Events](#)
[Tags](#)

Overview [Refresh](#)



Health

Ok

[Causes](#)

Running Version

2016-09-27T21-01-03ZB-d1fb76f

[Upload and Deploy](#)



Configuration

64bit Amazon Linux 2016.03
v2.1.6 running Multi-container
Docker 1.11.2 (Generic)
Newer version available

[Change](#)

Recent Events [Show All](#)

Time	Type	Details
2016-09-30 19:48:37 UTC-0700	INFO	Environment health has transitioned from Warning to Ok.
2016-09-30 19:48:37 UTC-0700	INFO	Removed instance [i-03c4723877a542261] from your environment.
2016-09-30 19:47:37 UTC-0700	WARN	Environment health has transitioned from Ok to Warning. No data received from 1 out of 3 instances.
2016-09-30 19:45:38 UTC-0700	INFO	Environment health has transitioned from Info to Ok.
2016-09-30 19:42:38 UTC-0700	INFO	Environment health has transitioned from Ok to Info. Command is executing on 1 out of 3 instances.



AWS Elastic Beanstalk - Intro

8

Web Tier

Scaling

Environment type: Load balanced, auto scaling

Number instances: 2 - 4

Scale based on: Average network out

Add instance when: > 6000000

Remove instance when: < 2000000

Instances

Instance type: m3.large

Root volume type: General Purpose (SSD)

Availability Zones: Any

Key pair: [redacted]

Notifications

Notifications: On

Send notifications to: [redacted]@example.com

Software Configuration

Environment variables: CONFIG_FILE, EXTRA_OPTS,
CONSUL_SERVERS, EB_MICROSERVICE

Log publication: Off

Updates and Deployments

Deployment batch size: 1

Rolling updates: are enabled

Rolling update type: Health

Max batch size: 1

Minimum instances in service: 2

Health

Application health check URL: /api/heartbeat

Health reporting: Enhanced

Managed Updates

Managed updates: are disabled



AWS Elastic Beanstalk - Intro

9

Network Tier

Load Balancing



Port: HTTP on port 80

Port: 80

Secure port: Off

Health: Requests /api/heartbeat over HTTP on port 80

Cross zone load balancing is enabled

Connection draining is enabled with 20 second timeout

Data Tier

You do not have a database. You can [create a new RDS database](#) or [use an existing database](#).



AWS Elastic Beanstalk - Intro

10

Logs

Request Logs ▾

Refresh

Click Request Logs to retrieve the last 100 lines of logs or the entire set of logs from each EC2 instance. [Learn more](#)

Log file	Time	EC2 instance	Type
Download	2016-10-10 18:44:05 UTC-0700	i-000000000000000000	Last 100 Lines
Download	2016-10-10 18:44:05 UTC-0700	i-000000000000000000	Last 100 Lines
Download	2016-10-10 18:44:15 UTC-0700	i-000000000000000000	Full Logs
Download	2016-10-10 18:44:16 UTC-0700	i-000000000000000000	Full Logs



AWS Elastic Beanstalk - Intro

11

Enhanced Health Overview

Filter By ▾

Instance Actions ▾

Hide details



Auto refresh (5s)

Server					Requests					Latency					Load Average		CPU Utilization			
	Instance ID	Status	Running ▴	Dep. ID	R/sec	2xx	3xx	4xx	5xx	P99	P90	P75	P50	P10	Load1	Load5	User%	Sys%	Idle%	I/O wait%
▼	Overall	Ok	N/A	N/A	-	-	-	-	-	-	-	-	-	-	N/A	N/A	N/A	N/A	N/A	N/A
Total 2Ok 2Pending 0Info 0Unknown 0No data 0Warning 0Degraded 0Severe 0																				
	...	Ok	3 days	111	-	-	-	-	-	-	-	-	-	-	0.97	0.62	3.7	2.8	93.5	0.0
	...	Ok	6 days	111	-	-	-	-	-	-	-	-	-	-	0.00	0.21	5.6	4.1	90.2	0.1



AWS Elastic Beanstalk - Intro

12

2.0

Healthy Host Count

9.8%

CPU Utilization

68.1

Average Latency
in Milliseconds

189.0

Sum Requests

2MB

Max Network In

2MB

Max Network Out

Monitoring

Time Range **3 hours** Period **1 minute** Edit





AWS Elastic Beanstalk - Intro

13

aws elasticbeanstalk describe-environments

```
{
  "Environments": [
    {
      "ApplicationName": "eventing",
      "EnvironmentName": "prod-eventing",
      "VersionLabel": "2016-10-06T01-47-06Z1-9c7d71b",
      "Status": "Ready",
      "Description": "prod-eventing",
      "EnvironmentLinks": [],
      "EnvironmentId": "e-xxxxxxxx",
      "EndpointURL": "awseb-e-t-AWSEBLoa-XXXXXXXXXXXX-11111111.us-east-1.elb.amazonaws.com",
      "SolutionStackName": "64bit Amazon Linux 2016.03 v2.1.6 running Multi-container Docker 1.11.2 (Generic)",
      "CNAME": "prod-xxxxxxxx.elasticbeanstalk.com",
      "Health": "Green",
      "AbortableOperationInProgress": false,
      "Tier": {
        "Version": "",
        "Type": "Standard",
        "Name": "WebServer"
      },
      "HealthStatus": "Ok",
      "DateUpdated": "2016-10-11T01:44:22.012Z",
      "DateCreated": "2016-01-23T23:28:46.050Z"
    }
  ]
}
```



AWS Elastic Beanstalk - Intro

14

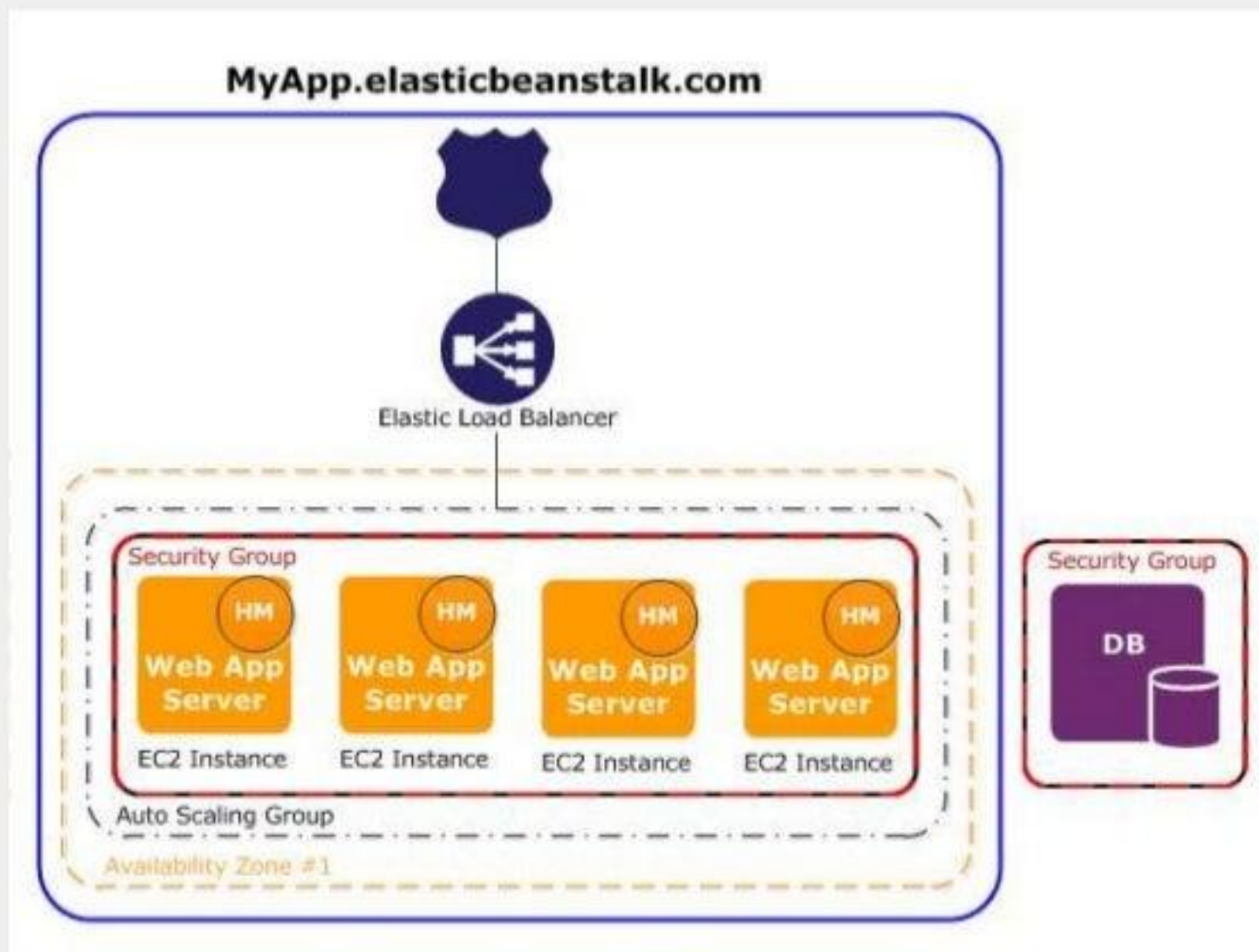
Every Elastic Beanstalk environment:

- ✓ EC2/ASG
- ✓ ELB
- ✓ CloudWatch
- ✓ S3
- ✓ ...



AWS Elastic Beanstalk - Intro

15





AWS Elastic Beanstalk - Intro

16

Supported platforms:

- | | |
|---------------------------|-----------------------------------|
| ✓ Single Container Docker | ✓ .NET on Windows Server with IIS |
| ✓ Multicontainer Docker | ✓ Node.js |
| ✓ Preconfigured Docker | ✓ PHP |
| ✓ Go | ✓ Python |
| ✓ Java SE | ✓ Ruby |
| ✓ Java with Tomcat | |

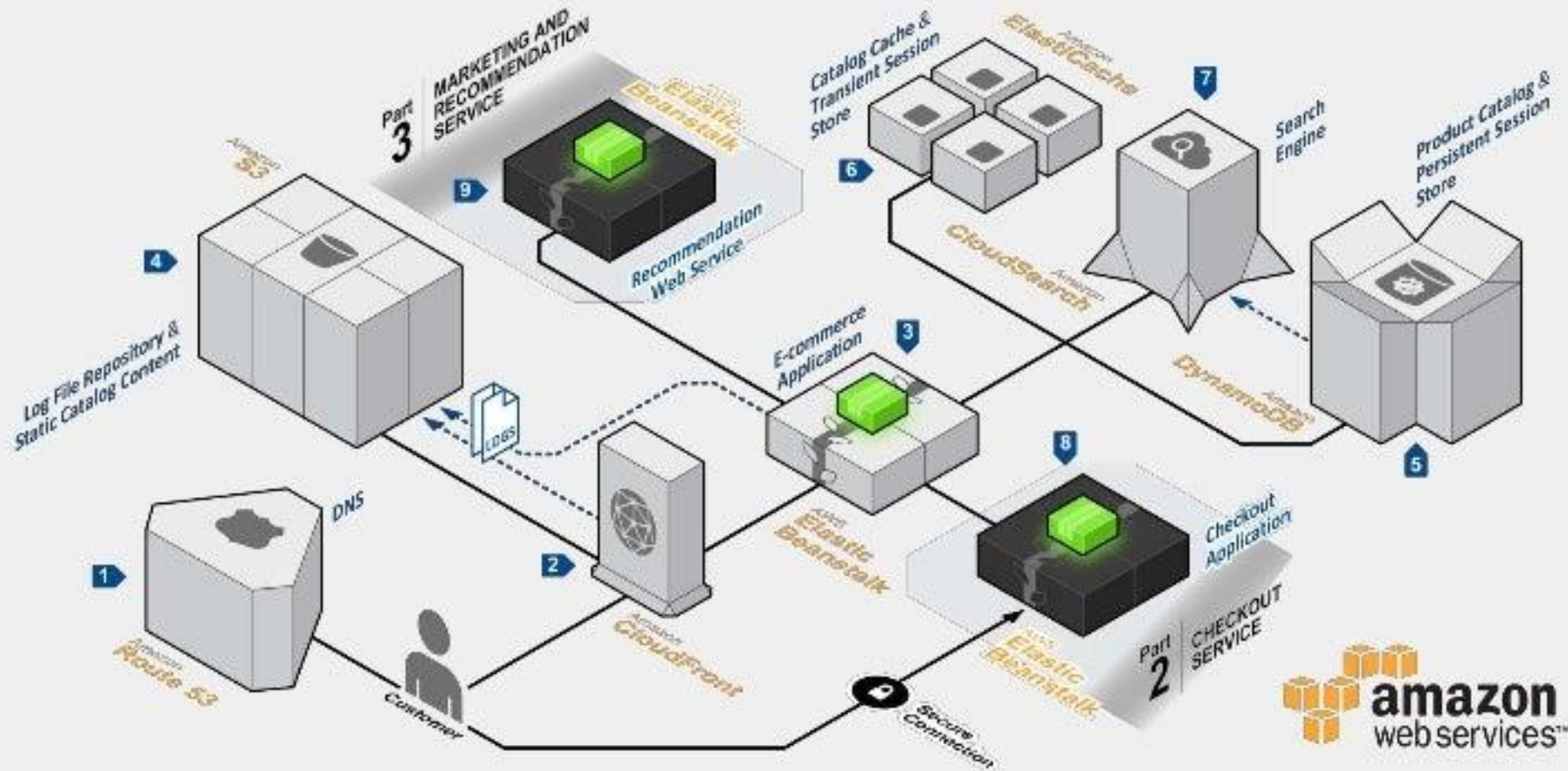


AWS Elastic Beanstalk - Intro

17

E-COMMERCE WEB SITE

PART 1: WEB FRONT-END



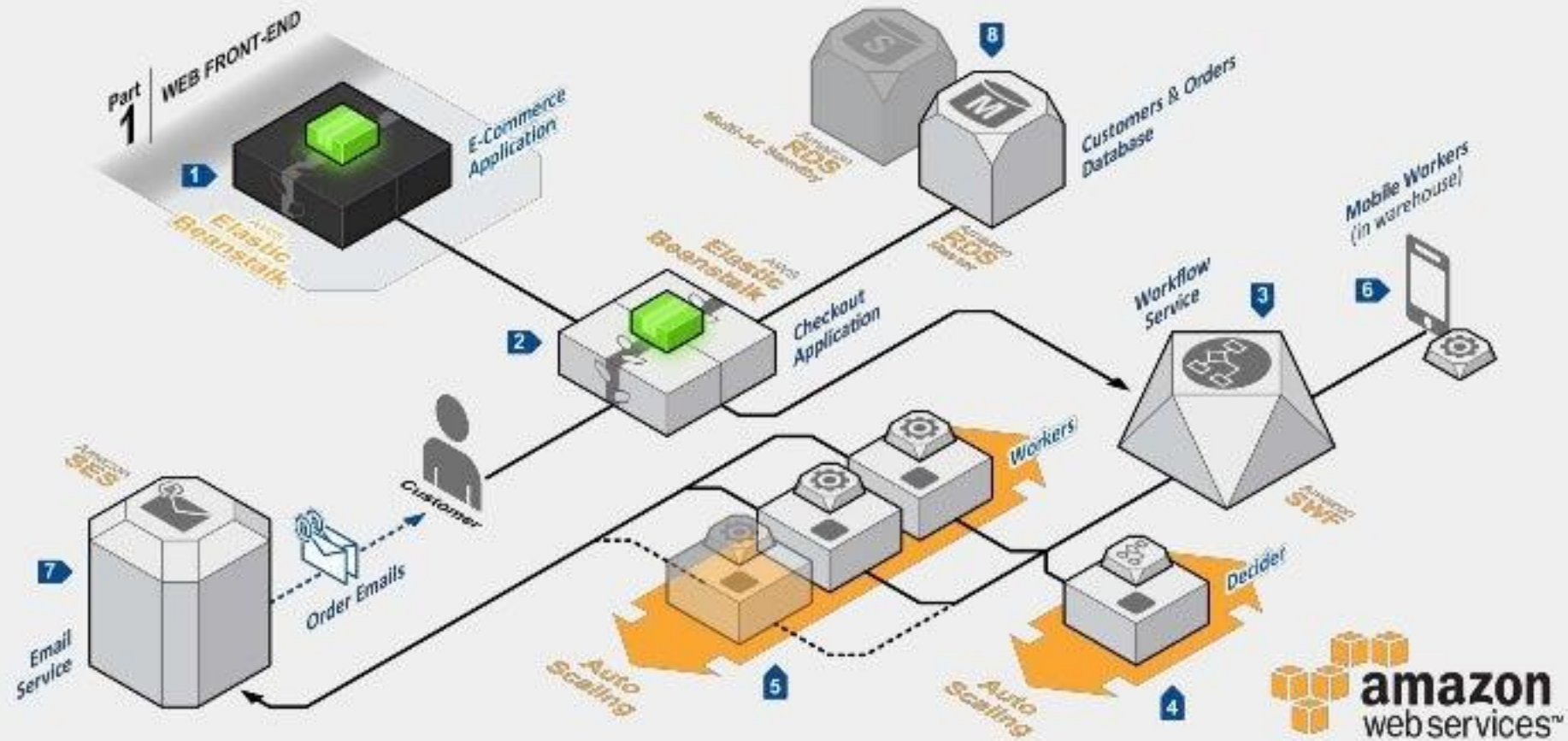


AWS Elastic Beanstalk - Intro

18

E-COMMERCE WEB SITE

PART 2: CHECKOUT SERVICE





AWS Elastic Beanstalk - Intro

19

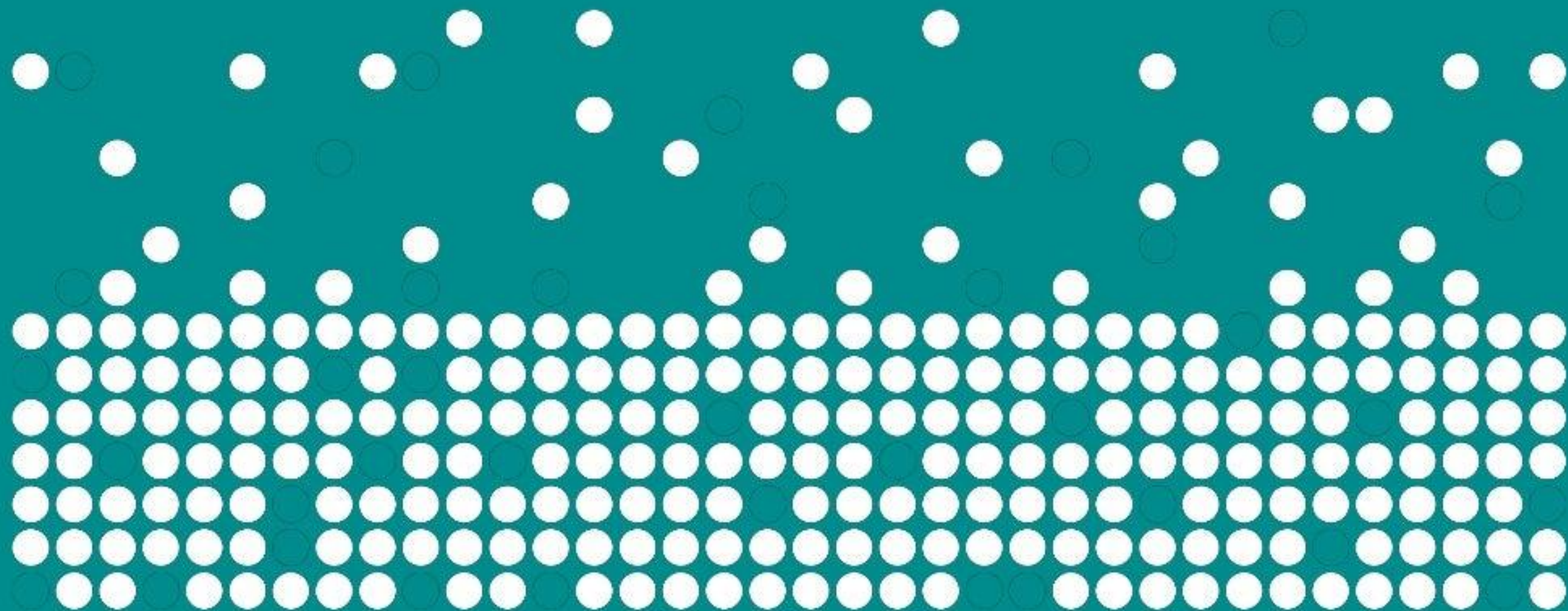
Deployment policies:

- ✓ **All at once** – Deploy the new version to all instances simultaneously. All instances in your environment are out of service for a short time while the deployment occurs.
- ✓ **Rolling** – Deploy the new version in batches. Each batch is taken out of service during the deployment phase, reducing your environment's capacity by the number of instances in a batch.
- ✓ **Rolling with additional batch** – Deploy the new version in batches, but first launch a new batch of instances to ensure full capacity during the deployment process.
- ✓ **Immutable** – Deploy the new version to a fresh group of instances by performing an immutable update.



AWS Elastic Beanstalk - Details

20





AWS Elastic Beanstalk - Details

21

We're going to take a look at:

- ✓ Web Server environment type
- ✓ Multicontainer Docker platform



AWS Elastic Beanstalk - Details

22

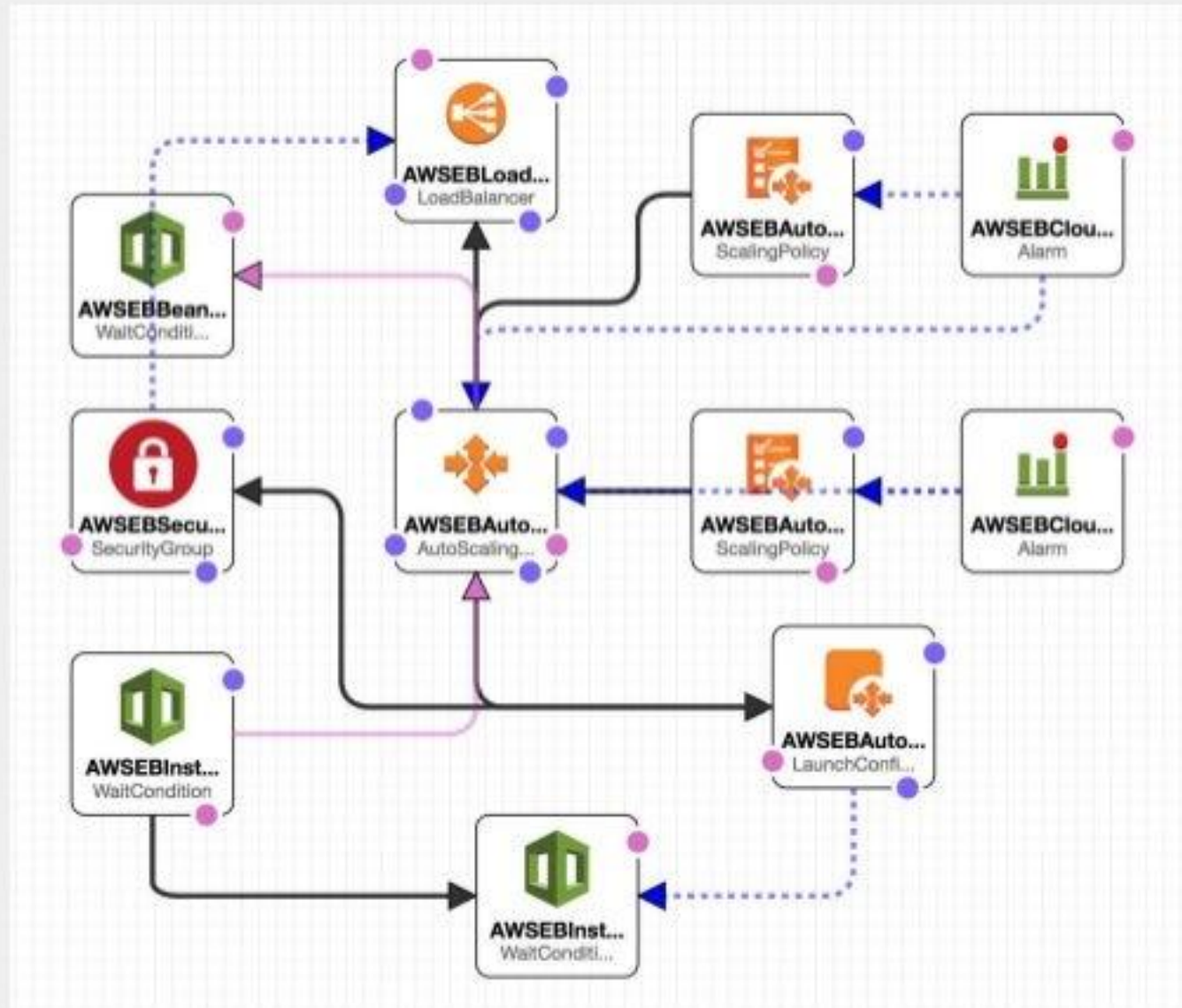
There is no magic:

- ✓ CloudFormation template to rule them all
- ✓ You can see and modify (?) all resources separately: EC2 instances, ELBs, etc.



AWS Elastic Beanstalk - Details

23

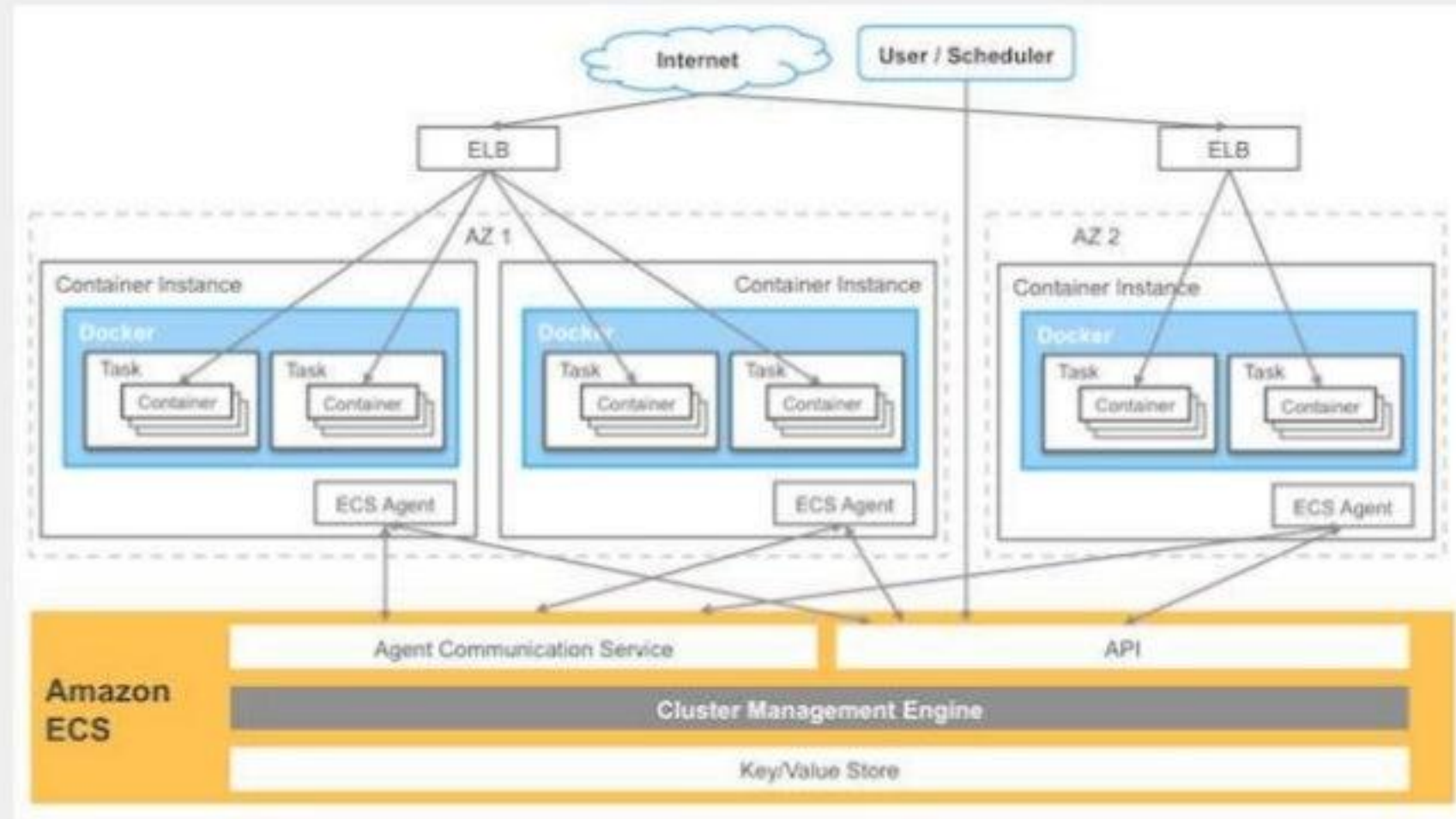




AWS Elastic Beanstalk - Details

24

ECS is used for Docker platform:

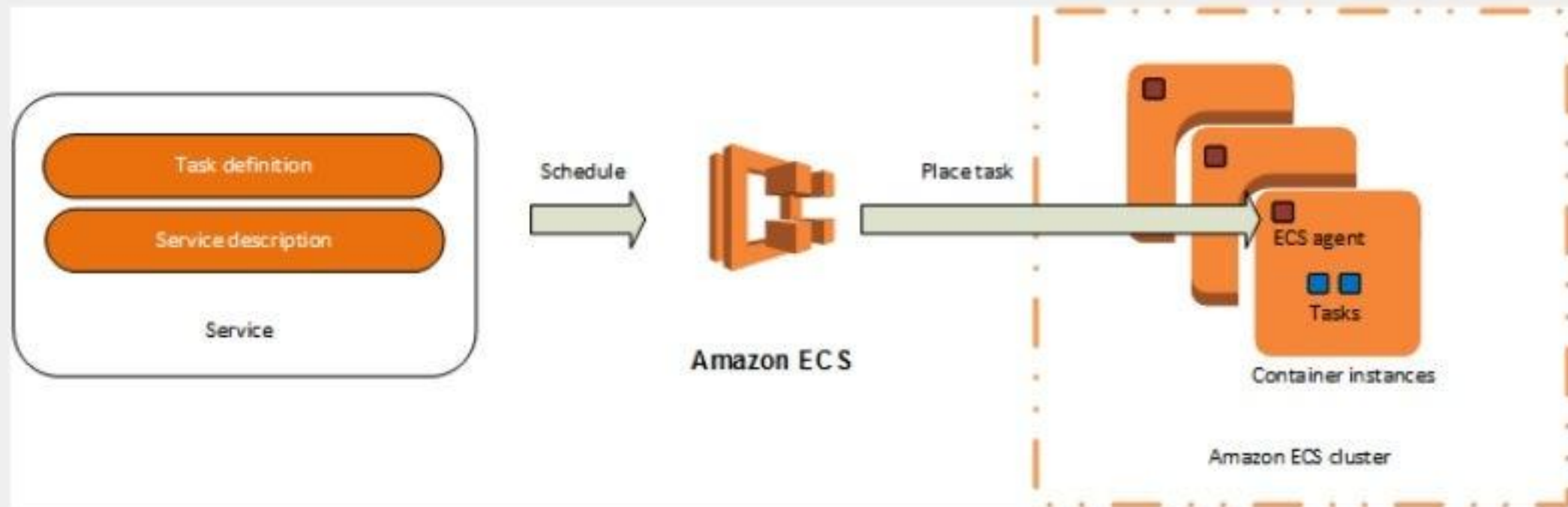




AWS Elastic Beanstalk - Details

25

ECS is used for Docker platform:





AWS Elastic Beanstalk - Details

26

But **EB** $\neq$ **ECS**:

- ✓ One EB environment always equals to one ECS cluster
- ✓ Every EC2 instance runs fixed configuration of Docker images, usually 1...
- ✓ Which means utilization is not great



AWS Elastic Beanstalk - Details

27

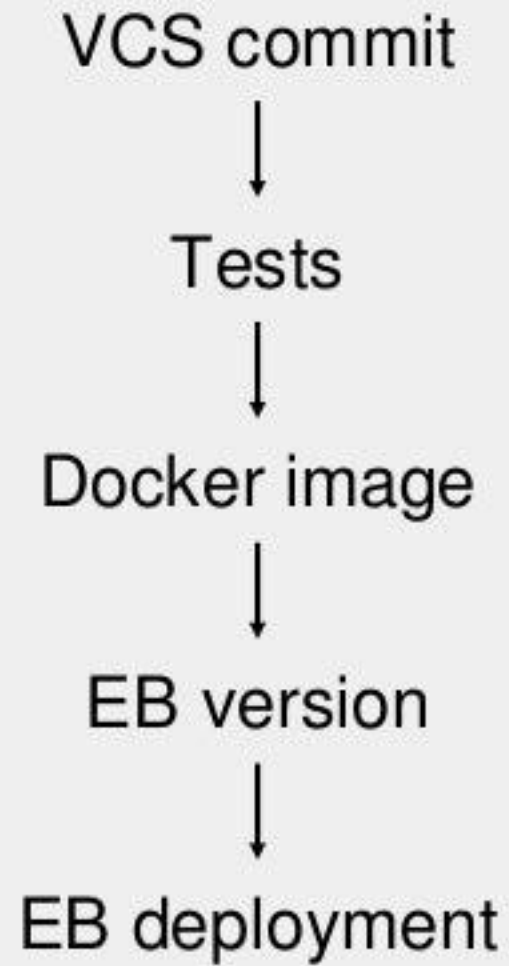
ECS is used for Docker platform (Dockerrun.aws.json):

```
{
  "AWSEBDockerrunVersion": 2,
  "containerDefinitions": [{
    "mountPoints": [],
    "name": "service-name",
    "image":
"DOCKER_REGISTRY/service/SERVICE_IMAGE:VE
RSION",
    "portMappings": [{
      "containerPort": 80,
      "hostPort": 80
    }],
    "memory": "7000",
    "essential": true
  }],
  "volumes": []
}
```



AWS Elastic Beanstalk - Details

28





AWS Elastic Beanstalk - Details

29

So, deployment pipeline looks like:

- ✓ Build new Docker image, tag and push it
- ✓ Create Dockerrun.aws.json, containing new Docker image
- ✓ `$ aws s3 cp artifact.zip s3://BUCKET/SERVICE/VERSION.zip`
- ✓ `$ aws elasticbeanstalk create-application-version --application-name SERVICE --version-label VERSION --source-bundle S3Bucket=BUCKET,S3Key=SERVICE/VERSION.zip`
- ✓ `$ aws elasticbeanstalk update-environment --environment-name SERVICE_ENV --version-label VERSION`



AWS Elastic Beanstalk - Details

30

Someone said “push Docker image”? Use **ECR** (EC2 Container Registry)!



AWS Elastic Beanstalk - Details

31

```
$ mkdir HelloWorld  
$ cd HelloWorld  
$ eb init -p PHP  
$ echo "Hello World" > index.html  
$ eb create dev-env  
$ eb open  
$ eb deploy
```



AWS Elastic Beanstalk - Details

32

.ebextensions is a secret weapon:

- ✓ Packages
- ✓ Groups
- ✓ Users
- ✓ Sources
- ✓ Files
- ✓ Commands
- ✓ Services
- ✓ Container Commands



AWS Elastic Beanstalk - Details

33

Files

Container commands

And Any AWS Resource!

files:

```
"/home/ec2-user/myfile" :  
  mode: "000755"  
  owner: root  
  group: root  
  source: http://foo.bar/myfile
```

```
"/home/ec2-user/myfile2" :  
  mode: "000755"  
  owner: root  
  group: root  
  content: |  
    # this is my file  
    # with content
```

container_commands:

```
collectstatic:  
  command: "django-admin.py collectstatic --noinput"  
01syncdb:  
  command: "django-admin.py syncdb --noinput"  
  leader_only: true  
02migrate:  
  command: "django-admin.py migrate"  
  leader_only: true  
99customize:  
  command: "scripts/customize.sh"
```

Resources:

AWSEBLoadBalancer:

Type:

"AWS::ElasticLoadBalancing::LoadBalancer"

Properties:

AccessLoggingPolicy:

S3BucketName: logs

S3BucketPrefix: elb-logs/SERVICE

Enabled: true

EmitInterval: 60



AWS Elastic Beanstalk - Details

34

```
"Resources": {
  "EventingService": {
    "Type": "AWS::ElasticBeanstalk::Environment",
    "Properties": {
      "ApplicationName": "eventing",
      "EnvironmentName": { "Fn::Join": [ "", [ { "Ref": "EnvPrefix" }, "-eventing" ] ] },
      "TemplateName": { "Ref": "EventingConfig" },
      "OptionSettings": [
        {
          "Namespace": "aws:elasticbeanstalk:application:environment",
          "OptionName": "EB_MICROSERVICE",
          "Value": "true"
        },
        {
          "Namespace": "aws:elasticbeanstalk:application:environment",
          "OptionName": "ENV_VAR",
          "Value": { "Ref": "EnvValue" }
        }
      ],
      "VersionLabel": { "Ref": "EventingVersion" },
      "Tier": {
        "Type": "Standard",
        "Name": "WebServer"
      }
    }
  }
}
```



AWS Elastic Beanstalk - Details

35

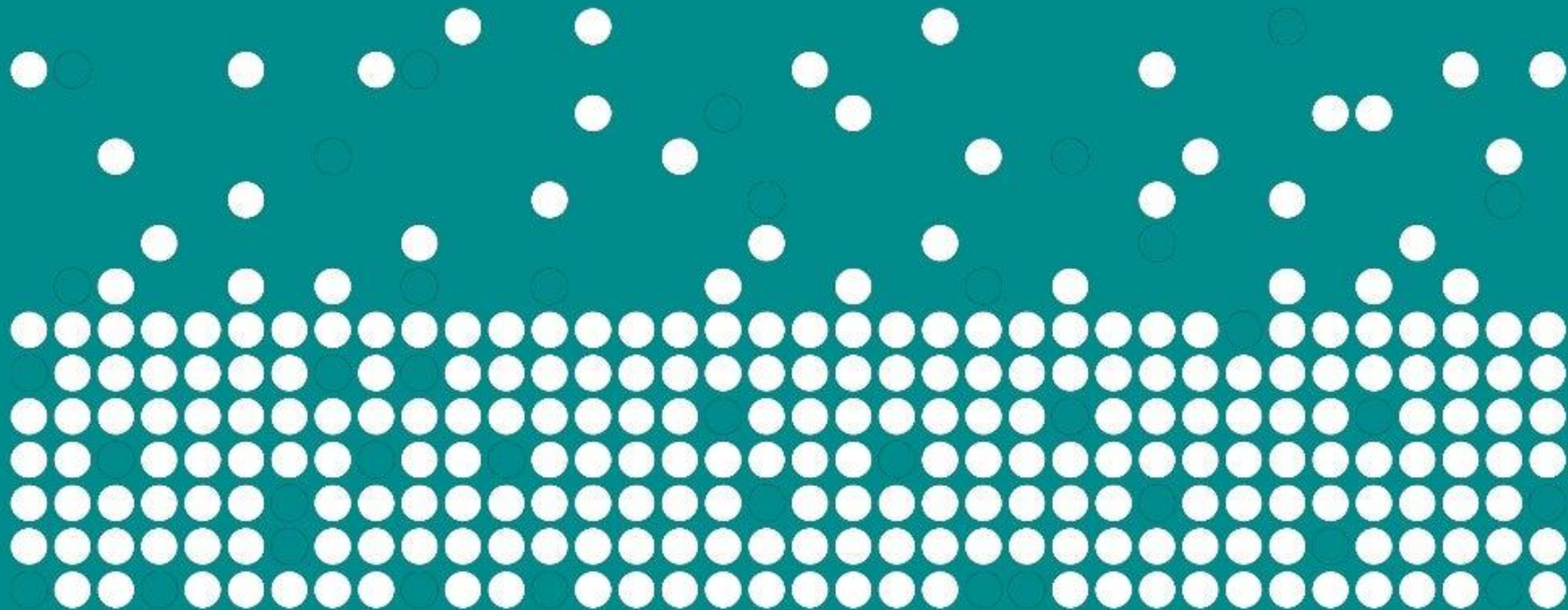
Best practices and things to notice:

- ✓ Use environment variables as much as you can
- ✓ Staging/prod environments should be as close as possible, obviously use the same Docker image everywhere
- ✓ Think about restricting access early
- ✓ Health status is not always correct and it's definitely not real-time
- ✓ Not all errors are recovered automatically. Sometimes you have to recreate environment from scratch*



AWS Elastic Beanstalk - Our tooling

36





AWS Elastic Beanstalk - Our tooling

37

<https://github.com/BenchLabs/eb-tools>

- ✓ eb-artifacts
- ✓ eb-envconf
- ✓ eb-environments
- ✓ eb-notifications



AWS Elastic Beanstalk - Our tooling

38

usage: eb-artifacts.py [-h] --name NAME --version VERSION

[-container-port CONTAINER_PORT]

[-port-mappings [PORT_MAPPINGS [PORT_MAPPINGS ...]]]

[-mount-points [MOUNT_POINTS [MOUNT_POINTS ...]]]

[-log-path LOG_PATH] [--memory MEMORY]

[-registry REGISTRY] [--templates TEMPLATES]

[-extensions-filter EXTENSIONS_FILTER]

[-output OUTPUT]



AWS Elastic Beanstalk - Our tooling

39

```
usage: eb-envconf.py [-h] [-p PREFIXES] [-t TEMPLATE] [-i INSTANCE_TYPE]
```

```
[-k KEY_NAME] [-e ENV_VARS]
```

```
applications [applications ...]
```



AWS Elastic Beanstalk - Our tooling

40

usage: eb-environments.sh [arg...]

You can pass in one of the following options:

- list List all applications and environments, their health and other status.
- events List all events for every applications and environments.
- status APPLICATION ENVIRONMENT Get environment status and health.
- ready-and-green APPLICATION ENVIRONMENT Wait for environment to be green and ready.
- is-online APPLICATION ENVIRONMENT Wait for environment's actual heartbeat endpoint to be alive.
- ready-and-online APPLICATION ENVIRONMENT Wait for environment to be green, ready and actual heartbeat endpoint to be alive.
- help Print this help message.



AWS Elastic Beanstalk - Our tooling

41

3:25 ☆

staging-cashcow

[WARN] in [staging-cashcow](#): Environment health has transitioned from Ok to Warning. 2 out of 2 instances are impacted. See instance health for details.

staging-stripe

[WARN] in [staging-stripe](#): Environment health has transitioned from Info to Warning. 1 out of 2 instances are impacted. See instance health for details.



eb BOT 3:39 PM

staging-eventing

[ERROR] in [staging-eventing](#): ECS task stopped due to: Essential container in task exited. (eventing-service: CannotPullContainerError: Error: image [\[REDACTED\]](#) not found)

staging-eventing

[ERROR] in [staging-eventing](#): Failed to start ECS task: arn:aws:ecs:us-east-1:[\[REDACTED\]](#) is STOPPED.



AWS Elastic Beanstalk - Conclusion

42

- ✓ EB gives you first-class Docker support
- ✓ EB is not similar to Kubernetes/Mesos despite of using ECS internally
- ✓ EB seamlessly integrates with other AWS tools
- ✓ EB is the easiest way to quickly get auto-scaled, load-balanced and monitored application with almost zero work in AWS. At the same time, every component is customizable when needed



Questions?

43



sap1ens.com
@sap1ens