

Starbucks API on AWS Docker

PART 1 - EC2 CONFIG & LAUNCH - AWS CLI

<https://aws.amazon.com/cli/> (Links to an external site.)Links to an external site.
<http://docs.aws.amazon.com/cli/latest/userguide/cli-chap-welcome.html> (Links to an external site.)Links to an external site.
<http://docs.aws.amazon.com/cli/latest/userguide/installing.html> (Links to an external site.)Links to an external site.

- Install (if needed) AWS CLI and run the following command:

aws --version

Sample Output:

aws-cli/1.11.66 Python/2.7.13 Darwin/16.4.0 botocore/1.5.29

```
[Vijays-MacBook-Air:~ Vijay$  
[Vijays-MacBook-Air:~ Vijay$  
[Vijays-MacBook-Air:~ Vijay$ aws --version  
aws-cli/1.11.158 Python/3.6.2 Darwin/16.7.0 botocore/1.7.16  
[Vijays-MacBook-Air:~ Vijay$  
[Vijays-MacBook-Air:~ Vijay$
```

PART 1 - EC2 CONFIG & LAUNCH - BUILD STARBUCKS API

- Build the Java Project JAR and Test a Local Run of the App using CURL

```
[Vijays-MacBook-Air:starbucks_v3 Vijay$  
[Vijays-MacBook-Air:starbucks_v3 Vijay$ curl http://localhost:9090  
[{"status":"OK","message":"Starbucks API Service: Version 3"}]  
[Vijays-MacBook-Air:starbucks_v3 Vijay$  
[Vijays-MacBook-Air:starbucks_v3 Vijay$ java -jar target/starbucks_v3.jar  
at sun.net.httpserver.HttpServerImpl.<init>(HttpServerImpl.java:59)  
at sun.net.httpserver.DefaultHttpServerProvider.createHttpServer(DefaultHttpServerProvider.java:35)  
at com.sun.net.httpserver.HttpServer.create(HttpServer.java:138)
```

PART 1 - EC2 CONFIG & LAUNCH - DEPLOY CONTAINERS (Using First Run Wizard)

- http://docs.aws.amazon.com/AmazonECS/latest/developerguide/ECS_GetStarted.html (Links to an external site.)Links to an external site.
- <https://us-west-1.console.aws.amazon.com/ecs/home?region=us-west-1#/firstRun> (Links to an external site.)

Step 1: Click the "Get Started" Button - Make sure both options on Wizard is Selected then Click "Continue" and Proceed.

[x] Deploy a sample application onto an Amazon ECS Cluster

[x] Store container images securely with Amazon ECR

Step 2: Select Repository Name: cmpe281

Sample URI: 060340690398.dkr.ecr.us-west-1.amazonaws.com/cmpe281

Step 3: Build, Tag and Push Docker Image

- Retrieve the docker login command that you can use to authenticate your Docker client to your registry:

```
aws ecr get-login --region us-west-1
```

- Run the docker login command that was returned in the previous step.
- Build your Docker image using the following command. You can skip this step if your image is already built:

```
docker build -t cmpe281 .
```

- After the build completes, tag your image so you can push the image to this repository: (Note - Use your Repository)

```
docker tag cmpe281:latest 060340690398.dkr.ecr.us-west-1.amazonaws.com/cmpe281:latest
```

- Run the following command to push this image to your newly created AWS repository (Note - Use your Repository):

```
docker push 060340690398.dkr.ecr.us-west-1.amazonaws.com/cmpe281:latest
```

Step 4: Create a task definition (Note - Use your Repository and Image)

Task Definition Name:	starbucks-api
Container Name:	starbucks-api
Image:	060340690398.dkr.ecr.us-west-1.amazonaws.com/cmpe281:latest
Memory Limits (Hard):	300 (MBs)
Port Mappings:	
Host Port	9090
Container Port	9090
Protocol	TCP

Step 5: Configure Service

Service Name:	starbucks-api
Desired number of tasks:	2
Container name (host port):	starbucks-api:9090
ELB listener protocol:	http
ELB listener port:	9090
ELB health check:	http:9090/
Service IAM role:	ecsServiceRole

Step 6: Configure Cluster

Cluster Name: starbucks-api
EC2 instance type: t2.micro
Number of instances: 2
Key Pair: cmpe281-us-west-1
Security Group: Allowed ingress source: Select "Anywhere"
Container instance IAM role: Create New Role (ecsServiceRole)

The screenshot shows the AWS Management Console interface. The top navigation bar includes the AWS logo, 'Services', 'Resource Groups', and user information. The left sidebar contains a navigation menu with categories like INSTANCES, IMAGES, ELASTIC BLOCK STORE, and NETWORK & SECURITY. The main content area displays the 'Instances' page, which includes a table of instances and a detailed view for the selected instance 'i-003546dedfad36db9'.

Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS (IPv4)
ECS Instance - EC2Container...	i-003546dedfad36db9	t2.micro	us-west-1a	running	2/2 checks ...	None	ec2-13-56-197-229.us-...
ECS Instance - EC2Container...	i-053dab8cf51e5ea0e	t2.micro	us-west-1c	running	2/2 checks ...	None	ec2-54-193-103-153.us-...
Default	i-07a64e2392e9568...	t2.micro	us-west-1a	stopped		None	ec2-52-53-118-212.us-...
c04ff57e-7d3a-4b67-ac8-4ac...	i-0b2b20cc3e80be4fa	t2.nano	us-west-1c	running	2/2 checks ...	None	ec2-54-193-117-131.us-...
Lab1	i-0e0b82e292e7ba54c	t2.micro	us-west-1a	running	2/2 checks ...	None	ec2-13-57-8-59.us-wes...

Instance: i-003546dedfad36db9 (ECS Instance - EC2ContainerService-starbucks-api) Public DNS: ec2-13-56-197-229.us-west-1.compute.amazonaws.com

Description	
Instance ID	i-003546dedfad36db9
Instance state	running
Instance type	t2.micro
Elastic IPs	
Availability zone	us-west-1a
Security groups	EC2ContainerService-starbucks-api-
Public DNS (IPv4)	ec2-13-56-197-229.us-west-1.compute.amazonaws.com
IPv4 Public IP	13.56.197.229
IPv6 IPs	-
Private DNS	ip-10-0-0-74.us-west-1.compute.internal
Private IPs	10.0.0.74
Secondary private IPs	

ChromeFileEditViewHistoryBookmarksPeopleWindowHelp

Quiz: CMPE 281 - LAB #5 - S...Amazon EC2 Container ServiceEC2 Management Console

Securehttps://us-west-1.console.aws.amazon.com/ecs/home?region=us-west-1#/clusters/starbucks-api/containerInstances

ServicesResource GroupsVijay YadavN. CaliforniaSupport

Amazon ECS

ClustersTask DefinitionsRepositories

Clusters > starbucks-api

Cluster : starbucks-api

Get a detailed view of the resources on your cluster.

Delete Cluster

Status ACTIVE

Registered container instances 2

Pending tasks count 0

Running tasks count 2

ServicesTasksECS InstancesMetricsScheduled Tasks

Scale ECS InstancesActions

Last updated on September 23, 2017 1:34:48 AM (0m ago)

Status: ALL ACTIVE DRAINING

Filter by attributes (click or press down arrow to view filter options)

	Container Instance	EC2 Instance	Availability Zo...	Agent Connec...	Status	Running tasks...	CPU available	Memory availa
<input type="checkbox"/>	bbca2e51-809b-46e3-be4...	i-003546dedfad...	us-west-1a	true	ACTIVE	1	1014	693
<input type="checkbox"/>	c2e22428-dbe7-4383-812...	i-053dab8cf51e...	us-west-1c	true	ACTIVE	1	1014	693

FeedbackEnglish (US)© 2008 - 2017, Amazon Web Services, Inc. or its affiliates. All rights reserved.Privacy PolicyTerms of Use

Import+BuilderTeam Library

IN SYNCvijay8608

No Environment

Collection

Console

Filter Messages

Clear

GET http://ECS-first-run-alb-1706088315.us-west-1.elb.amazonaws.com:909001:23:03.786

Request Headers:
cache-control: "no-cache"
postman-token: "0f44a7c7-83d9-4581-a147-b2cd3a5c3209"
user-agent: "PostmanRuntime/6.3.2"
accept: "*//*"
host: "ECS-first-run-alb-1706088315.us-west-1.elb.amazonaws.com:9090"
accept-encoding: "gzip, deflate"

Response Headers:
date: "Sat, 23 Sep 2017 20:23:03 GMT"
content-type: "application/json"
transfer-encoding: "chunked"
connection: "keep-alive"
server: "Restlet-Framework/2.3.7"
vary: "Accept-Charset, Accept-Encoding, Accept-Language, Accept"
accept-ranges: "bytes"

Response Body:
status: "OK"
message: "Starbucks API Service: Version 3"

20015 ms

icks Order (V1)
ucks Order (V1)
bucks Order (V1)
cs Order (V1)
bucks Order (V1)
cs Orders (V1)
icks Order (V2)
ucks Order (V2)
bucks Order (V2)
cs Order (V2)
cs Order - ETAG (V2)
bucks Order (V2)
cs Orders (V2)
cks Service (V3)