**AWS Configuration Cheat Sheet**

*updated 8-12-2021*

This lecture note is not intended to be a replacement for the videos, but to serve as a cheat sheet for students who want to quickly run thru the AWS configuration steps or easily see if they missed a step. It will also help navigate through the changes to the AWS UI since the course was recorded.

**Docker Compose config Update**

Make sure to follow the steps in the earlier lecture note to rename your development docker compose file and create a new production compose file:

<https://www.udemy.com/course/docker-and-kubernetes-the-complete-guide/learn/lecture/27975358>

**Initial Setup**

1. Go to AWS Management Console

2. Search for Elastic Beanstalk in "Find Services"

3. Click the "Create Application" button

4. Enter "docker" for the Application Name

5. Scroll down to "Platform" and select "Docker" from the dropdown list.

6. Change "Platform Branch" to **Docker running on 64bit Amazon Linux 2**

7. Click "Create Application"

8. You should see a green checkmark after some time.

9. Click the link above the checkmark for your application. This should open the application in your browser and display a Congratulations message.

**Change from Micro to Small instance type:**

*Note that a t2.small is outside of the free tier. t2 micro has been known to timeout and fail during the build process on the old platform. However,****this may not be an issue on the new Docker running on 64bit Amazon Linux 2 platform. So, these steps may no longer be necessary.***

1. In the left sidebar under Docker-env click "Configuration"

2. Find "Capacity" and click "Edit"

3. Scroll down to find the "Instance Type" and change from *t2.micro* to *t2.small*

4. Click "Apply"

5. The message might say "No Data" or "Severe" in Health Overview before changing to "Ok"

**Add AWS configuration details to .travis.yml file's deploy script**

1. Set the *region*. The region code can be found by clicking the region in the toolbar next to your username.

eg: 'us-east-1'

2. *app* should be set to the Application Name (Step #4 in the Initial Setup above)

eg: 'docker'

3. *env* should be set to the lower case of your Beanstalk Environment name.

eg: 'docker-env'

4. Set the *bucket\_name*. This can be found by searching for the S3 Storage service. Click the link for the elasticbeanstalk bucket that matches your region code and copy the name.

eg: 'elasticbeanstalk-us-east-1-923445599289'

5. Set the *bucket\_path* to 'docker'

6. Set *access\_key\_id* to $AWS\_ACCESS\_KEY

7. Set *secret\_access\_key* to $AWS\_SECRET\_KEY

**Create an IAM User**

1. Search for the "IAM Security, Identity & Compliance Service"

2. Click "Create Individual IAM Users" and click "Manage Users"

3. Click "Add User"

4. Enter any name you’d like in the "User Name" field.

eg: docker-react-travis-ci

5. Tick the "Programmatic Access" checkbox

6. Click "Next:Permissions"

7. Click "Attach Existing Policies Directly"

8. Search for "beanstalk"

9. Tick the box next to "AdministratorAccess-AWSElasticBeanstalk"

10. Click "Next:Tags"

11. Click "Next:Review"

12. Click "Create user"

13. Copy and / or download the *Access Key ID* and *Secret Access Key* to use in the Travis Variable Setup.

**Travis Variable Setup**

1. Go to your Travis Dashboard and find the project repository for the application we are working on.

2. On the repository page, click "More Options" and then "Settings"

3. Create an *AWS\_ACCESS\_KEY* variable and paste your IAM access key from step #13 above.

4. Create an *AWS\_SECRET\_KEY* variable and paste your IAM secret key from step #13 above.

**Deploying App**

1. Make a small change to your src/App.js file in the greeting text.

2. In the project root, in your terminal run:

1. git add.
2. git commit -m “testing deployment"
3. git push origin main

3. Go to your Travis Dashboard and check the status of your build.

4. The status should eventually return with a green checkmark and show "build passing"

5. Go to your AWS Elasticbeanstalk application

6. It should say "Elastic Beanstalk is updating your environment"

7. It should eventually show a green checkmark under "Health". You will now be able to access your application at the external URL provided under the environment name.

**Course content**

Section 1: Dive Into Docker!

14 / 14 | 36min

Section 2: Manipulating Containers with the Docker Client

14 / 14 | 51min

Section 3: Building Custom Images Through Docker Server

11 / 11 | 48min

Section 4: Making Real Projects with Docker

11 / 11 | 54min

Section 5: Docker Compose with Multiple Local Containers

11 / 11 | 52min

Section 6: Creating a Production-Grade Workflow

24 / 24 | 1hr 34min

* Lecture completed

Play

62. Development Workflow

1min

* Lecture completed

Play

63. Flow Specifics

7min

* Lecture completed

Play

64. Docker's Purpose

2min

* Lecture completed

Play

65. Project Generation

3min

Resources

* Lecture completed

Start

66. Create React App Generation

1min

* Lecture completed

Play

67. More on Project Generation

2min

Resources

* Lecture completed

Play

68. Necessary Commands

5min

* Lecture completed

Play

69. Creating the Dev Dockerfile

4min

Resources

* Lecture completed

Play

70. Duplicating Dependencies

1min

* Lecture completed

Play

71. Starting the Container

3min

Resources

* Lecture completed

Play

72. Docker Volumes

7min

* Lecture completed

Start

73. WSL2 and Windows Users Must Read Before Next Lecture

2min

* Lecture completed

Play

74. Bookmarking Volumes

5min

* Lecture completed

Play

75. Shorthand with Docker Compose

4min

Resources

* Lecture completed

Play

76. Overriding Dockerfile Selection

2min

Resources

* Lecture completed

Play

77. Do We Need Copy?

3min

* Lecture completed

Play

78. Executing Tests

4min

* Lecture completed

Play

79. Live Updating Tests

5min

* Lecture completed

Play

80. Docker Compose for Running Tests

6min

Resources

* Lecture completed

Play

81. Shortcomings on Testing

9min

* Lecture completed

Play

82. Need for Nginx

3min

* Lecture completed

Play

83. Multi-Step Docker Builds

7min

* Lecture completed

Play

84. Implementing Multi-Step Builds

7min

Resources

* Lecture completed

Play

85. Running Nginx

2min

Section 7: Continuous Integration and Deployment with AWS

20 / 20 | 1hr 6min

* Lecture completed

Play

86. Services Overview

3min

* Lecture completed

Play

87. Github Setup

4min

* Lecture completed

Start

88. Important Info About Travis and Account Registration

1min

* Lecture completed

Play

89. Travis CI Setup

4min

* Lecture completed

Play

90. Travis YML File Configuration

8min

* Lecture completed

Start

91. Required Travis Script Updates

1min

* Lecture completed

Play

92. A Touch More Travis Setup

4min

Resources

* Lecture completed

Play

93. Automatic Build Creation

4min

* Lecture completed

Start

94. Required Updates for Amazon Linux 2 Platform - DO NOT SKIP

1min

* Lecture completed

Play

95. AWS Elastic Beanstalk

4min

Resources

* Lecture completed

Play

96. More on Elastic Beanstalk

2min

* Lecture completed

Play

97. Travis Config for Deployment

9min

Resources

* Lecture completed

Play

98. Automated Deployments

7min

* Lecture completed

Play

99. Exposing Ports Through the Dockerfile

4min

Resources

* Lecture completed

Play

100. Workflow With Github

4min

* Lecture completed

Play

101. Redeploy on Pull Request Merge

2min

* Lecture completed

Play

102. Deployment Wrapup

2min

* Lecture completed

Start

103. Environment Cleanup

1min

* Lecture completed

Start

104. AWS Configuration Cheat Sheet

3min

* Lecture completed

Start

105. Finished Project Code with Updates Applied

1min

Resources

Section 8: Building a Multi-Container Application

14 / 15 | 1hr 13min

Section 9: "Dockerizing" Multiple Services

15 / 19 | 1hr 18min

Section 10: A Continuous Integration Workflow for Multiple Images

9 / 11 | 46min

Section 11: Multi-Container Deployments to AWS

24 / 26 | 1hr 53min

Section 12: Onwards to Kubernetes!

13 / 17 | 1hr 46min

Section 13: Maintaining Sets of Containers with Deployments

11 / 18 | 1hr 32min

Section 14: A Multi-Container App with Kubernetes

27 / 30 | 2hr 22min

Section 15: Handling Traffic with Ingress Controllers

5 / 14 | 44min

Section 16: Kubernetes Production Deployment

0 / 43 | 2hr 41min

Section 17: HTTPS Setup with Kubernetes

0 / 18 | 49min

Section 18: Local Development with Skaffold

0 / 7 | 29min

Section 19: Extras

0 / 1 | 1min

Overview

Q&AQuestions and answers

Notes

Announcements

Reviews

Learning tools