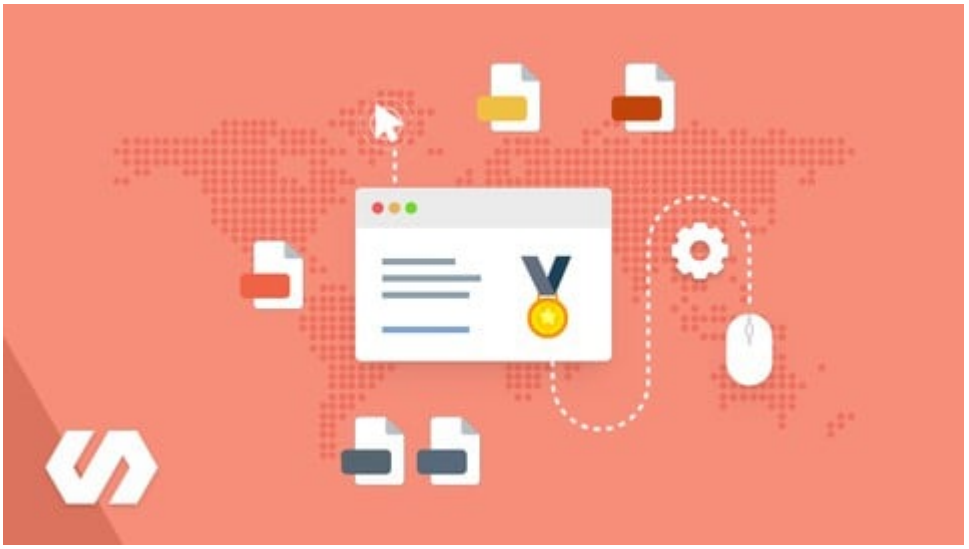
# Docker and Kubernetes: The Complete Guide

udemy.com/course/docker-and-kubernetes-the-complete-guide/learn/lecture/20695748



AWS Configuration Cheat Sheet

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.

**Elastic Beanstalk Application Creation**

1. Make sure you have followed the guidance in this note.
2. Go to AWS Management Console and use Find Services to search for Elastic Beanstalk
3. Click “Create Application”
4. Set Application Name to 'multi-docker'
5. Scroll down to Platform and select Docker
6. Verify that "Single Instance (free tier eligible)" has been selected
7. Click the "Next" button.
8. In the "Service Role" section, verify that "Use an Existing service role" is selected.
9. Verify that aws-elasticbeanstalk-service-role has been auto-selected for the service role.
10. Verify that aws-elasticbeanstalk-ec2-role has been auto-selected for the instance profile.

11. Click "Skip to review" button.
12. Click the "Submit" button.
13. You may need to refresh, but eventually, you should see a green checkmark underneath Health.

***RDS Database Creation***

1. Go to AWS Management Console and use Find Services to search for RDS
2. Click Create database button
3. Select PostgreSQL
4. In Templates, check the Free tier box.
5. Scroll down to Settings.
6. Set DB Instance identifier to **multi-docker-postgres**
7. Set Master Username to **postgres**
8. Set Master Password to **postgrespassword** and confirm.
9. Scroll down to Connectivity. Make sure VPC is set to Default VPC
10. Scroll down to Additional Configuration and click to unhide.
11. Set Initial database name to **fibvalues**
12. Scroll down and click Create Database button

***ElastiCache Redis Creation***

1. Go to AWS Management Console and use Find Services to search for ElastiCache
2. In the sidebar under Resources, click **Redis OSS caches**
3. Click the **Create Redis OSS caches** button
4. Select **Design your own cache** and C**luster cache**
5. **Make sure Cluster Mode is DISABLED.**
6. Scroll down to Cluster info and set Name to **multi-docker-redis**
7. Scroll down to Cluster settings and change Node type to **cache.t3.micro**
8. Change Number of Replicas to **0** (Ignore the warning about Multi-AZ)

9. Scroll down to Subnet group. Select **Create a new subnet group** if not already selected.
10. Enter a name for the Subnet Group such as **redis**.
11. Scroll down and click the Next button
12. Scroll down and click the Next button again.
13. Scroll down and click the Create button.
14. After the cache has been fully created (green **Available** status), click the new **multi-docker-redis** cache name. Then, click **Modify**.
15. Scroll down to Security and locate the **Transit encryption mode** setting. Change this setting from **Required** to **Preferred**. This is very important, otherwise, you will not see your Calculated Values in the client form.
16. Scroll down and click **Preview changes**, then click **Modify**.
17. You will need to wait for the cache to apply the changes and restart (green **Available** status) which could take around 5 minutes or more.

***Creating a Custom Security Group***

1. Go to AWS Management Console and use Find Services to search for VPC
2. Find the Security section in the left sidebar and click Security Groups
3. Click Create Security Group button
4. Set Security group name to multi-docker
5. Set Description to multi-docker
6. Make sure VPC is set to your default VPC
7. Scroll down and click the Create Security Group button.
8. After the security group has been created, find the Edit inbound rules button.
9. Click Add Rule
10. Set Port Range to **5432-6379**
11. Click in the box next to Source and start typing 'sg' into the box. Select the Security Group you just created.
12. Click the Save rules button

***Applying Security Groups to ElastiCache***

1. Go to AWS Management Console and use Find Services to search for ElastiCache
2. Under Resources, click Redis clusters in Sidebar
3. Check the box next to your Redis cluster
4. Click Actions and click Modify
5. Scroll down to find Selected security groups and click Manage
6. Tick the box next to the new multi-docker group and click Choose
7. Scroll down and click Preview Changes
8. Click the Modify button.

***Applying Security Groups to RDS***

1. Go to AWS Management Console and use Find Services to search for RDS
2. Click Databases in Sidebar and check the box next to your instance
3. Click Modify button
4. Scroll down to Connectivity and add select the new multi-docker security group
5. Scroll down and click the Continue button
6. Click Modify DB instance button

***Applying Security Groups to Elastic Beanstalk***

1. Go to AWS Management Console and use Find Services to search for Elastic Beanstalk
2. Click Environments in the left sidebar.
3. Click MultiDocker-env
4. Click Configuration
5. In the Instances row, click the Edit button.
6. Scroll down to EC2 Security Groups and tick the box next to multi-docker
7. Click Apply and Click Confirm
8. After all the instances restart and go from No Data to Severe, you should see a green checkmark under Health.

**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 Elastic Beanstalk Application Name
   eg: 'multi-docker'

3. *env* should be set to your Elastic Beanstalk Environment name.
   eg: 'MultiDocker-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.

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

6. Set the *bucket_path* to 'docker-multi'

7. Set *access_key_id* to $AWS_ACCESS_KEY

8. Set *secret_access_key* to $AWS_SECRET_KEY

***Setting Environment Variables***

1. Go to AWS Management Console and use Find Services to search for Elastic Beanstalk

2. Click Environments in the left sidebar.

3. Click MultiDocker-env

4. In the left sidebar click Configuration

5. Scroll down to the Updates, monitoring, and logging section and click Edit.

6. Scroll down to the Environment Properties section. Click Add environment property.

7. In another tab Open up ElastiCache, click Redis and check the box next to your cluster. Find the Primary Endpoint and copy that value but omit the :6379

8. Set REDIS_HOST key to the primary endpoint listed above, remember to omit :6379

9. Set REDIS_PORT to 6379

10. Set PGUSER to postgres

11. Set PGPASSWORD to postgrespassword

12. In another tab, open up the RDS dashboard, click databases in the sidebar, click your instance and scroll to Connectivity and Security. Copy the endpoint.

13. Set the PGHOST key to the endpoint value listed above.
14. Set PGDATABASE to fibvalues
15. Set PGPORT to 5432
16. Click Apply button
17. After all instances restart and go from No Data, to Severe, you should see a green checkmark under Health.

***IAM Keys for Deployment***

You can use the same IAM User's access and secret keys from the single container app we created earlier, or, you can create a new IAM user for this application:

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-multi-travis-ci

5. Click "Next"

6. Click "Attach Policies Directly"

7. Search for "beanstalk"

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

9. Click "Next"

10. Click "Create user"

11. Select the IAM user that was just created from the list of users

12. Click "Security Credentials"

13. Scroll down to find "Access Keys"

14. Click "Create access key"

15. Select "Command Line Interface (CLI)"

16. Scroll down and tick the "I understand..." check box and click "Next"

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

**AWS Keys in Travis**

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
4. Create an *AWS_SECRET_KEY* variable and paste your IAM secret key

**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 Elastic Beanstalk 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.