**Deploying Multi-Container Applications to AWS Elastic Beanstalk**

The easiest way to deploy containerised applications to AWS is to use [Elastic Beanstalk](https://docs.aws.amazon.com/elasticbeanstalk/latest/dg/concepts.html). Instead of creating and managing individual resources (such as EC2, security groups, load balancer, RDS and so on) to run applications, it creates and manage them as environments.

If you want to deploy containers to AWS quickly without learning too much about more granular services like ECS or load balancer, Elastic Beanstalk is the way to go.

To deploy application to Elastic Beanstalk, AWS has a wonderful CLI tool, [EB CLI](https://docs.aws.amazon.com/elasticbeanstalk/latest/dg/eb-cli3.html). With a few commands, we can easily deploy an application to Elastic Beanstalk.

As an example, we will use wiremock container. Wiremock is a open-source tools to create API mocks. For the details about how wiremock container works, you can refer to the github repo I created for this post [here](https://github.com/mydatahack/deploy-wiremock-container-aws-elastic-beanstalk).

First of all, we need to create a Dockerrun.aws.json file in the project folder. This file is similar to the docker-compose file. It is specifically used for Elastic Beanstalk. It is much simpler than the config files for ECS. When we do multi-container deployment, Beanstalk uses ECS and docker-intalled EC2 as an underlining infrastructure. But, the complexity to manage it is abstracted.

For the container application, the asset is pushed to /var/app/current folder. In the example below, we are mapping the relevant folders that contains data files to the container.

Elastic Beanstalk uses the health check on port 80 from the load balancer by default. So, the first container is mapped to port 80. If you need to use https, the best option is to use route 53 to map DNS to the load balancer created in the environment.

<script src="https://gist.github.com/mydatahack/26eef95d122f0da279e413f30243a169.js"></script>

Once we have Dockerrun.aws.json, the rest is easy. First, we initialise the application. Then follow the prompt to choose the options. Make sure to choose multi-container option.

<pre>

eb init

</pre>

Now it is ready to go. Just create the environment. Environment in Elastic Beanstalk means application plus all the associated resources. Once we set the name for the environment, the rest will be taken care by the EB CLI.

<pre>

eb create swapi-mock

</pre>

After a few minutes, you will have your containers running in AWS.

For updating the code, we can run deploy command.

<pre>

eb deploy awapi-mock

</pre>

When you finish it, you can delete the environment. This will terminate all the resources created for the environment.

<pre>

eb terminate

</pre>

That’s it. Pretty simple. For further information, you can check the git repo made for this post: [deploy-wiremock-container-aws-elastic-beanstalk](https://github.com/mydatahack/deploy-wiremock-container-aws-elastic-beanstalk).