Deep Dive in the Cloudformation Magic. (Following the Cloudformation Execution)

In the past lecture we set up the cloudformation service, and started the building process. Lets now follow the cloudformation building process...

In the stacks screen we are able to see the whole process. First it start with the Creating in progress status for our main architecture template And the first module cloudformation is going to create is our VPC stack. We can check everything happening in the event tab, selecting the stack we want to explore.

For each stack we can check the parameters, the template they are using, the events, and the outputs.

Lets start checking the resources already created.

First, the VPC, go to the Network Section and click the VPC service.

We can verify the VPC name in the CIDR block we specify in our template file, The we can check the subnets associated:

This ones are easier to find if we sort the subnets by name:

There is to important things to notice here, first is the 2 subnets under the VPC cidr block and the amount of ips available, the next one is the creation of this subnets each one different availability zone.

us east 1a and us east 1b

We can also check our route tables with the local and the internet gateway routes. And of course the subnet associate to this route table.

Then we check our internet gateway attached And finally the Default security group: lets remember this number vpc-

As you can see there is only one security group allowing incoming traffic only from the same Security Group.

Going back to our cloudformation stack screen we can check the outputs created from the VPC stack.

We see that our loadbalancer is already created, lets go to the Compute Section and click EC2 to check the Loadbalancer configuration.

Lets check now our Load balancer. The LoadBalancer is inside the EC2 service.

Inside the cicd loadbalancer details we are able to see:
1.- The vpc associated is the one we just created before
2.- The Availability zones for this load balancer are related to the subnets we created in the last step.
3.- This load balancer is an internet facing Load Balancer
4.- Lets check the Security Group associated with this load balancer, its the number... we can see in the details section Inbound Rules that only traffic from port 80 is allowed
5.- In the Listener tab, we can verify our port 80 listener and of course the tag StackName we just created during the Stack setup.

Going Back to the stack screen, lets check the ECS cluster details.

ECS cluster was created, you can verify that there is 1 service running and 2 instances instances, Going into the service detail we can see in the ECS instances tab both instances running, the CPU available, the memory available and of course the Docker version on this instances.

Its nice to notice that our instances are running one in the AZ1 and the other in the AZ2.

We wont see anything in the task tab, because we are not running any docker image on this instances

Lets go to the Task definition link, there we will the task definition we defined in our template file, lets click it to see the family name and the family revision.

If we click it again we can start checking all the task definition detail we defined, CPU units, memory, port, etc.

The last thing to check here is the ECR repository created, remember this is the one with the Deletion policy attribute, if we delete our stack this repository is not going to be deleted.

The last Module to check is the Database Cluster:

The first thing to check is the Aurora Engine in 2 instances. the db instance class type that we set up, the Multi Availability zone and the replication role, as we explained before .

One important data we will neee from here is the Cluster endpoint, this will be our url to connect from our Docker containers.

Inside we can also check more details, like the parameters groups we set up, the subnet attached, the Availability zone, the port and more.

Going back to our cloudformation stack screen, we see that all our architecture is ready, and with this we finish the first part of the course. In the next part we are going to explore and implement our deployment pipeline, tweak a little bit our architecture to verify the change set actions and of course, see our application and deployment pipeline in action.