REVATURE

Project 1

Noah Greher

The Poke-Cloud

I first began by setting up the basic network architecture of the V.P.C.--similar to what was done in class. I initially tried to to do only one private instance, but later decided for convenience to create two private instances, allocated with their private and public subnets inside their respective AZ’s. I used the second instance as a sort of dummy. Once the Elastic Load Balancer was working, and capable of hosting my http servers, I paused my “dummy” instance after setting up an EFS between it and the “master” instance. What might distinguish the master from the dummy aside from naming conventions? Well, the master has a crontab that no other instances have running. So I snapshotted the non-cron instance and used it for a launch configuration which was then used for my autoscaling group. The autoscaled instances are launched with EFS mounted and http servers ready. This crontab updates the the master bucket, while the other instances (dummy instances) are then updated from a premounted EFS.

I chose this particular lay out because it did not seem wise to have a scaling number of servers calling upon a bucket at the same time, especially if the frequency and/or number of instances were to be high in number. I’d rather have one master and other slaves (so to speak). Ideally I would’ve liked to emulate the master slave architecture of RDS instances (likely implemented through lambda and Cloud Watch), but I did not manage to do so. Perhaps in the future.

With this all set up, the rest of the architecture was easy. I had to make sure that content synched from the s3 to my instances, and after verifying I moved onward. My API routes requests from the front end to the backend (lambda functions which help access databases and other resources). The last step I had was the Cloud Front. I deployed this with the ELB (Application) in all edge locations.