Condo Daily Report 9-9-21

Thursday

* CloudFormation
  + App Stack
    - Started off this morning with the problem of not being able to ssh
      * Looked into the problem to find that there was no public IP
        + Looked on forums to solve this issue
        + Told me to turn on public dns

Did not work

* + - * + Looked on more forums to see about EIP

I noticed there was no option directly in instance setup in cloud formation for an EIP slot

Found out that an outside EIP must be configured and attached to an EC2

* + - * + Created EIP and locally referenced it to the EC2 being made in the stack
    - Was able to SSH into the instance
      * Checked the instance to see if the attached volume could access ruby
        + This was a test that I had done to see if volume attaching would be a feasible idea for our server instances
      * It did so this made me think that an attaching scenario would be the best way to go
    - I was wrong about the attaching scenario
      * Unfortunately during stack creation, you cannot remove the root volume and put another volume in root
    - I Tried using blockstore section to delete the original root then add the volume in the root stack
      * This didn’t work
    - I tried launching the instance without an AMI as an empty instance
      * This wasn’t possible
    - I was confused so I went on forums
      * A lot of forums said to manually touch the volumes in the ec2 console
        + This is bad because it causes stack drift
        + Stack drift is when the services in the stack look different than they do on the template that they were launched from
    - I found nothing on forums but had an idea with AMI’s
      * I learned that you can create a root ami from a snapshot of another systems root
      * I took a snapshot of the other systems root
      * Created an AMI from it
      * Attached that AMI to the ec2 instance
      * Bingo ruby popped up with version 3.0 like I installed
        + This confirms that we can take the root of the other system, make an AMI from it, and attach it to the new ec2 in the stack
      * This also means we can attach extra volumes that are already available to the stack as well as long as its not the root
    - Started creating parameters section of the stack
      * Created the ami insert parameter
      * Created the amount of volumes to attach parameter
      * Attached the parameters for each volume
    - In order for these parameters to work and not conflict conditions were needed to enforce it
      * Created conditions depending on how many volumes their were
      * Created a condition for if there were no volumes
    - Made the stack modular based so we will take this stack and make it a nested stack to the app stack and call it the ec2 stack
    - Ran into a bug and it doesn’t like the structure of the resource in the hasVolServer resource
      * Haven’t debugged completely
      * The editor is not very straight forward, and other code checkers say my yaml is valid
    - Met with kayce and talked about ELB implementation for scalability and security implementations