Gracias por permitirme realizar una prueba técnica con ustedes.  
  
Comparto los logros que obtuve, pero finalmente no logre completar lo solicitado.  
  
Estoy adjuntando 3 proyectos en los cuales estuve trabajando para poder lograr mi meta.

**Proyecto1** - Pensé en llevar un api sencillo en flask se creó y la probé con Insomnia Similar a postman.

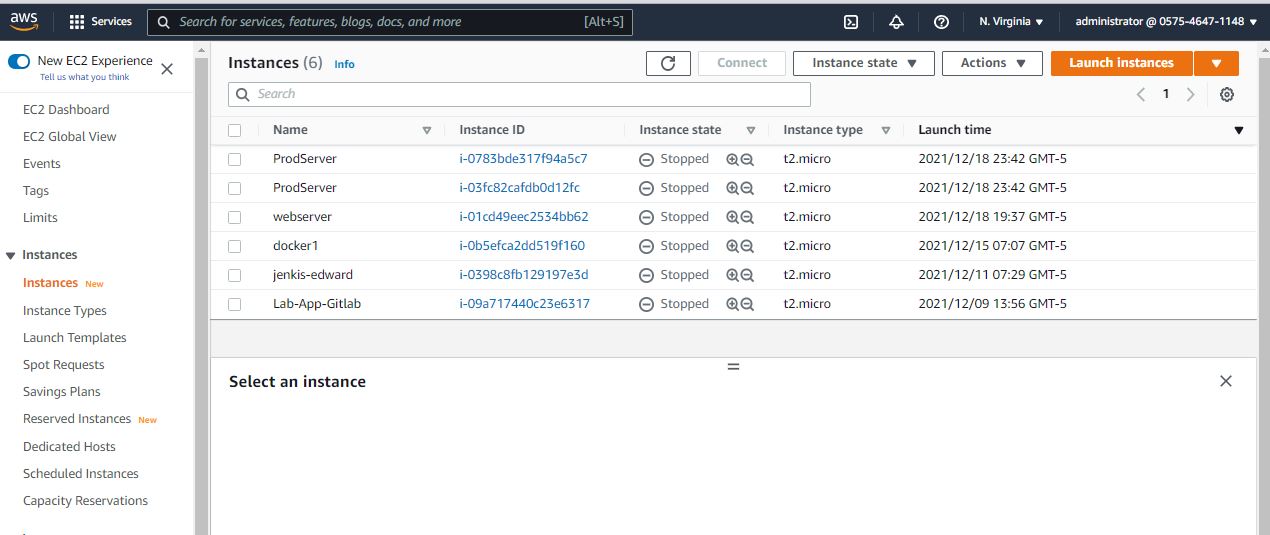
**Resultado** Solo se creó la Api

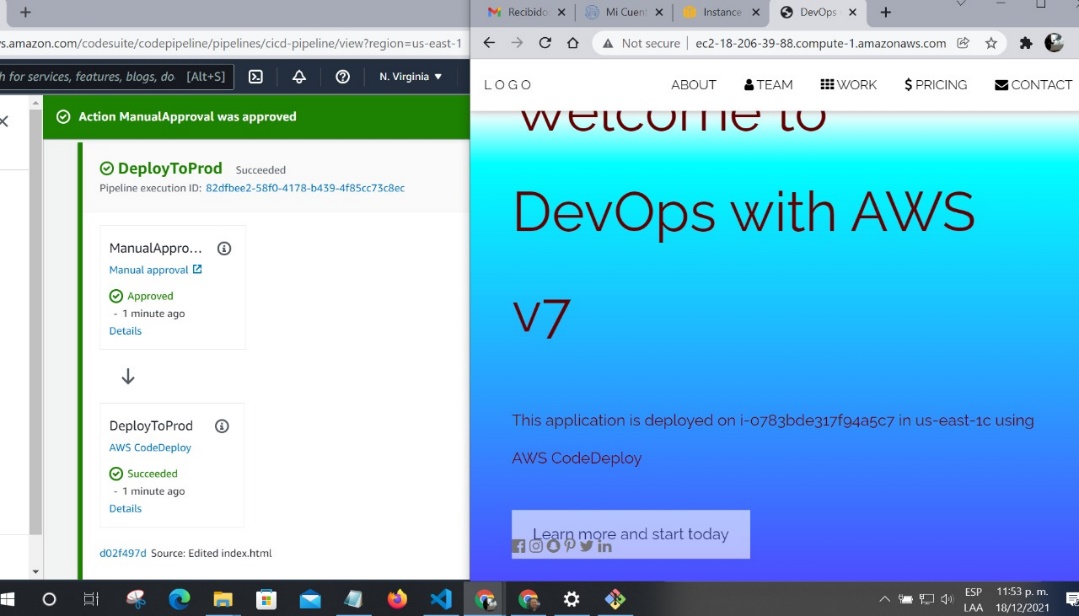
**Proyecto 2**- Por mi experiencia en montar sitios en WordPress, siempre había querido montar una aplicación en angular y aprender a dockerizarla. Y probar si funciona en mi máquina, funciona en la tuya y pude descargarla en otro equipo correrla y mi app angular funciono en otro equipo. **El docker File se encuentra dentro de la carpeta proyecto-angular>proyectoangula**

**Resultado** se logró crear la imagen, y luego subirla a Docker Hub <https://hub.docker.com/repository/docker/carloscloud/angularapp:v1> y también se subió a AWS ECR, donde desde consola cree un cluster de ECS y siguiendo un tutorial de youtube logre completar todos los pasos, pero no logre verla en producción, pienso que me falto exponerle correctamente el puerto 80. Si le abrí el puerto 80 en el security group pero no logre ver la app corriendo en el ECS.

**Proyecto3**- Tome un entrenamiento en Linkedin <https://www.linkedin.com/learning/devops-with-aws> , donde se crea completamente un AWS Pipeline, usando todas las herramientas, codecommit, code build, code deploy e integrarlos y correr pruebas en aws codepipeline de una aplicación web. Usando este repo: <https://github.com/bedoyacloud/devops-with-aws-2886059>

**Se crearon 2 ambientes uno de test WebServer y otro ambiente de ProdServer y las pruebas de despliegue fueron exitosas tanto en dev como en prod.**





Donde más me cuesta es en crear el buildspec.yml para llevar mi app de angular, dockerizarla y llevarla al ECR.   
  
La creación del Pipeline lo veo viable, entiendo que se requieren ciertos roles que le permitirán poder hacer las etapas de manera correcta y fluida. Y sé que es algo que con la práctica lograre dominarlo y cada vez uno se ira sintiendo más cómodo

Y donde realmente me siento con muy poca experiencia es de poder crear el ECS con Terraform o CloudFormation. Intente ver algunos training que me permitieran lograrlo, pero lo que más me detuvo fue lograr alimentar el pipeline con la imagen dockerizada.

**How you envision all the environments**

Por una experiencia laboral, entendí que es Buena práctica, tener una cuenta de AWS para cada ambiente, y la autenticación realizarla mediante AWS Single Sing-On.

**How you envision the workflow of the project :**

Yo lo veo:

-Clonar repo de AWS CodeCommit en mi máquina, para poder hacerle las actualizaciones al código.

Eso dispara el codebuild para que pruebe y genere una nueva imagen de docker, que sera entregada a code deploy para que la pruebe en ambiente de Dev, y si la pueba es correcta, pueda ser de manera manual o automática enviada al ECR que alimenta al ECS en la creación de los container.

**What alternative you propose for env variables management**

In AWS we have AWS SSM parameter store which is an good and economy solution, or AWS Secrets manager in case you need rotation for security credential, and will cost more than SSM.

**How would you improve availability in a production environment.**

Adding a load balancer, adding CloudFront too. And configure for containers be in different AZ, so the application can be resilient.

**If you ran out of time, how did you envision your finished project?**

I will deliver the project on ECS using ECR, and later, will do the implementation in Terraform

Pido excusas en la demora de la entrega, estaba buscando varias Fuentes para tratar de lograr el yaml de configuración para el codebuild, colocar a funcionar el pipeline, y asi luego poder generar la infra para ECS mediante Terraform.