2026-01-12-aws-cloud-quest-task-1-a

What is AWS Cloud Quest?

So AWS Cloud Quest is basically a gamified learning platform by Amazon to teach you AWS services. instead of sitting through boring lectures or reading documentation that puts you to sleep, you're playing an actual game where you solve problems using AWS.

yeah i gotta learn AWS now cos.. well that's what the recruiters want and if my job requires it.. i am gonna learn it.

might as well make it fun, right?

First impressions

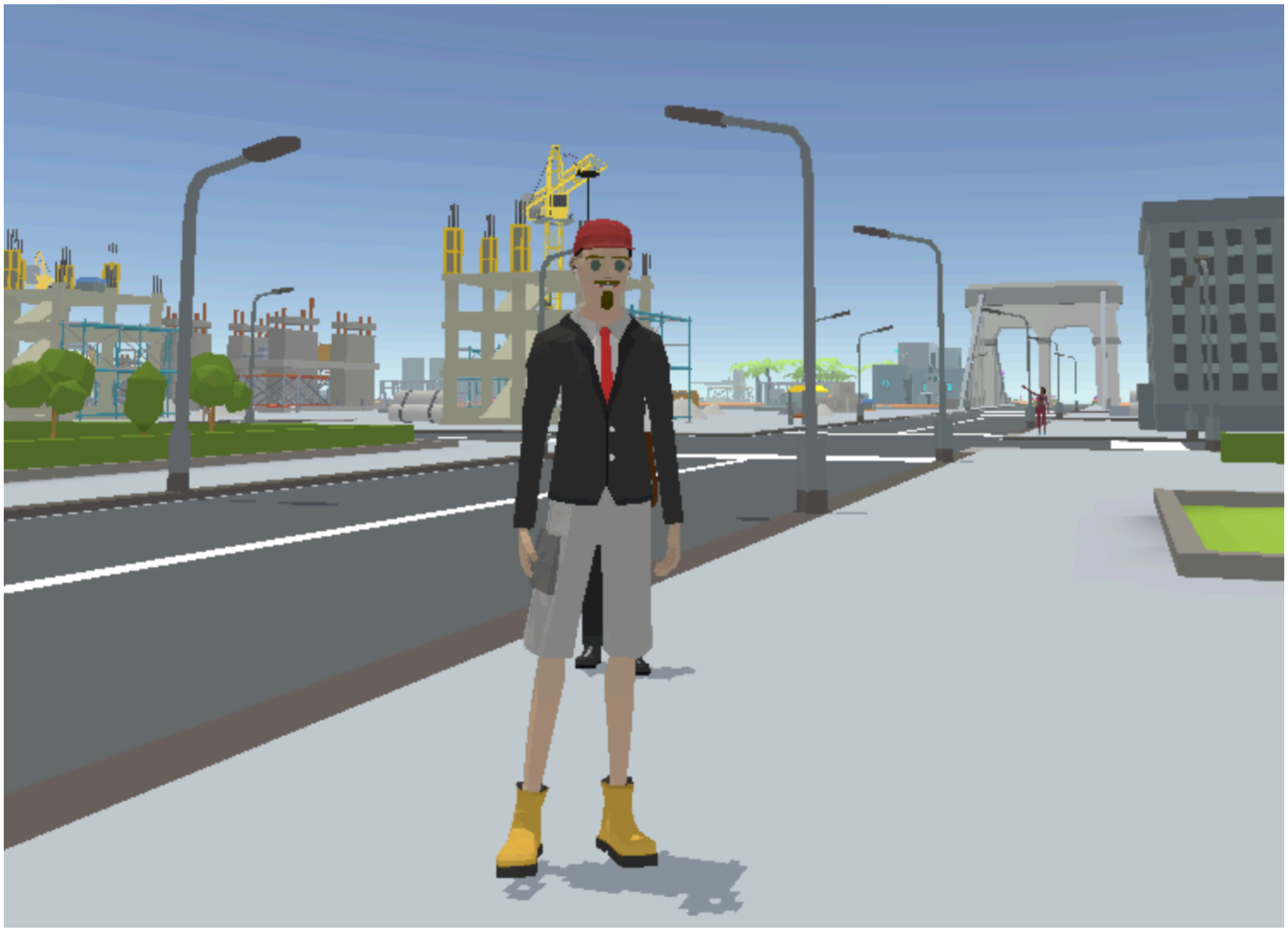
first impressions, "did i accidentally open some online game?". The welcome screen looks like some town building game and fun music, immediately impressed. turns out, it is a city building game where i have to fix "city's problem" using aws.. so freaking cool..

man the music is catchy af

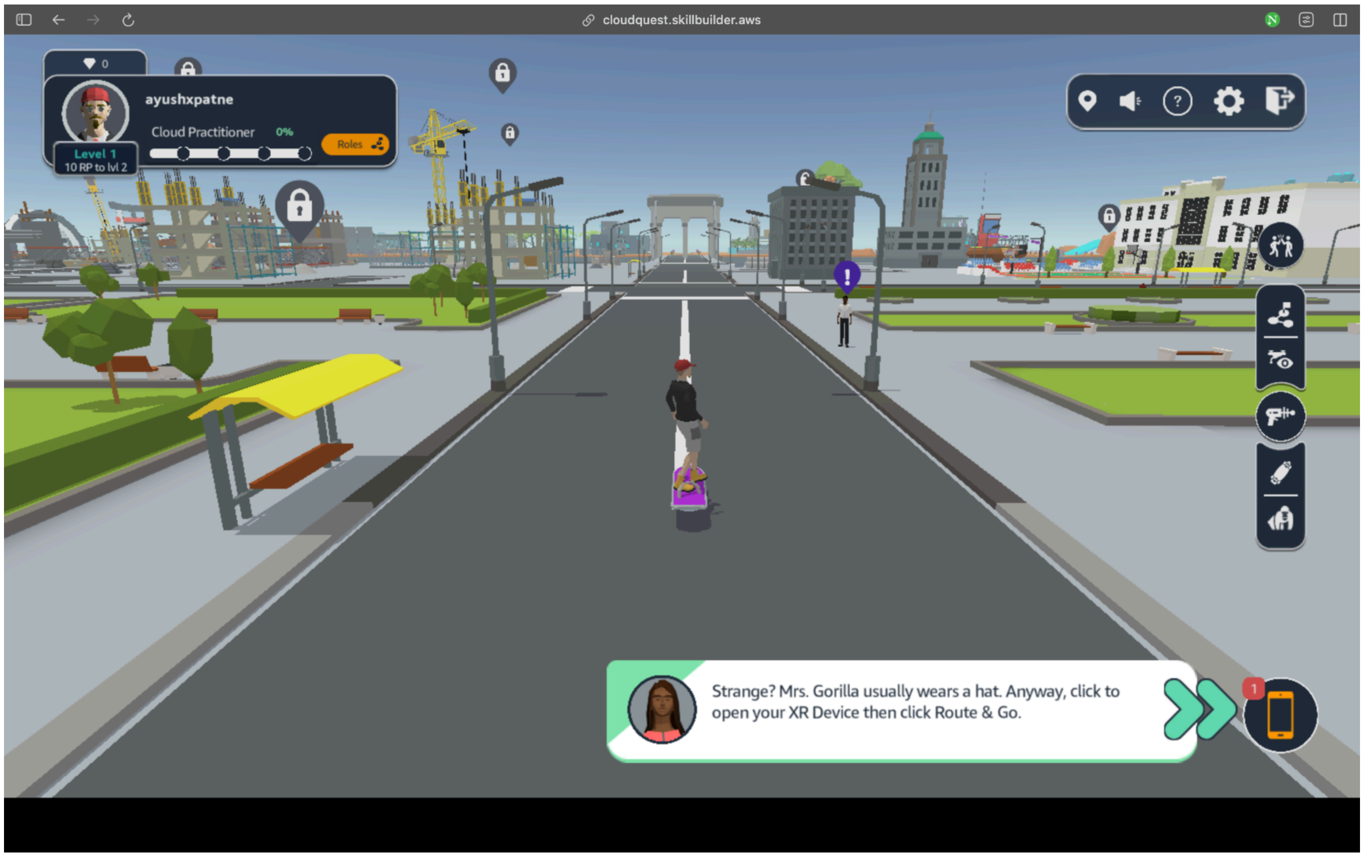
anyways, we are given a map of city and buildings– and each building have some problems they are facing / task assigned to us for each problem.

so, we are gonna start with the obvious task, "Cloud Computing Essentials".

but first, we had to create an avatar and you know, obviously, i went with.. that's right.. Suit with shorts. *PEAK FORMAL ATTIRE INNIT*



anyways.. as soon as the tutorial starts– yo why tf was there a gorilla? and why he started running behind this dude? wtf? and guess what... this is not the strangest part apparently.



Strange? Mrs. Gorilla usually wears a hat. Anyway, click to open your XR Device then click Route & Go.

Mrs. Gorilla.. is not wearing a hat.. yeah thats totally strange dude... her attacking the dude is completely normal.

The Problem

anyways moving forward.. for the first task we meet with the city's mayor and he describes the problem: **it takes months to setup new resources and infra for the data centres. The residents they use some website to get wave forecasts, but it keeps breaking down**

so we recommend them to migrate to AWS and suggest if the website is static we can use S3 to host it.

then we, the player, ie, me, are told why S3? (what is S3 will come later)

Why S3?

Amazon S3 can store and host static web pages. it's fully managed, so S3 handles all the server work for you. it also automatically creates copies of data across multiple data centers to keep it safe.

Learning objectives for the task:

1. Use bucket policy to secure an S3 bucket.
2. Enable static website hosting on S3.

3. Test access to then hosted webpage.

The Big Question : What is S3?

- S3 is used for storage and retrieval of data.
- The data is stored as objects (including the metadeta) and the container in which they are stored are called buckets.
- When configured for website hosting, the s3 will have a dedicated URL. Requests to this URL, will serve the bucket's root object, in this case html file.
- Access to bucket controlled through permissions, defined in Bucket Policy, which is written in JSON file.

More on S3:

- There are **multiple storage classes of S3 depending on data profile and use cases.**
- AWS also provides a management tool for granular data control, called **S3 Storage Class Analysis** in which based on access patterns it will tell adequate Storage class, help with data transfer and S3 lifecycle policy (ie when to transfer to another storage class?, when does objects expire?, etc). This enables us to be cost efficient.
- **S3 supports batch operations**, where with a single api request we can perform operations like copy, object tag sets, modify access controls, retrieve archived objects from S3 glacier (which is another storage class used to keep objects which dont need frequent or immediate access) on billions of objects.
- **S3 works with AWS Lambda**, which enables us to log activities, set up alerts and **automations**.
- We also have Query in place services which allow us to do big data analytics without the need to copy it separately.
- S3 also **has CRR ie cross region replication and SRR that ie Same region replication**.
- S3 is also compatible with AWS's analytical services like Amazon Athena & Amazon Redshift.
- **Athena is used to execute SQL expressions to query data,** but **Redshift is more suitable for complex queries** against data at rest and for larger data bases upto exabytes.
- S3 Select is used to retrieve object subsets instead of whole objects (which can go upto 5 TB). This increases query performance by 400% and reduce query costs by 80%.
- **Versioning allows to keep multiple variants for objects. *The default is Unversioned.*** Versioning-Suspended will stop creation of version IDs for new objects, but will keep existing object versions. This is done via Bucket Key & Version ID.
- Once versioning is turned on, it cannot be set back to unversioned. One can implement "Version-suspending"

- Versioning allows us to recover accidentally deletion or overwrites. if versioning is enabled, incase of deletion it wont delete the object, but just put delete marker. Incase of overwrite, new version of object is created. Allowing us to preserve, retrieve and restore every version of object stored in S3.

S3 Buckets Access Management

- by default, the S3 buckets are privates. Only resource owner can access the records.
- Owner can give access to others with the help of access policies.
- There are two types,
    - Resource based policies: Access policies attached resources. Ex: Bucket Policies, Access control lists (ACLs). Time limited access can be given with the help of query string authentication
    - User policies: Access policies attached to user. Ex: AWS IAM policies.
- AWS recommends using bucket policies and IAM policies.

Resource Based Policies

1. ACL - Each bucket and object has ACL. It uses XML sequence which defines accounts and groups which have access to the attached resource and what kind of access. By Default, ACL gives full control to owner.
2. Bucket Policy - Written in JSON. Defines access for all objects in said bucket.

User Policies

User policies are also written in json. It defines who has access to S3 resources. You can use AWS IAM to control access and permissions, create groups and users, etc.

Performing The Tasks

After learning all that, time to practice. got taken to the AWS dashboard to set up the website. This was step 1 of the task where we learned the above, now its time to practice. We are taken to AWS dashboard. Where we will now setup the website.

I did the following steps to host the website, as guided by the tutorial:

1. Search for S3. Open S3 Page.
2. There was a bucket existing already, with the website files in it.
3. After few exploratory steps, check for permissions. Make sure public access is turned on.

4. By default, AWS encrypts stuff and decrypts, but one can toy around with it i think for proper production database.
5. Go to properties section and turn on static website hosting, an URL is generated.
6. Thats it! If you now visit that URL you can access website.

> I am not attaching the URL, cos i think its temporary URL so yeah it will be deleted later ig.

DIY Section.

After this we had a DIY section, where I had to rename index.html to waves.html and validate it.

Final Thoughts

Overall It was Very Cool! 10/10 Really enjoyed the experience.

The gamification makes learning AWS actually bearable instead of just reading dry docs. I never thought i'd be learning about S3 buckets while dealing with gorilla attacks but here we are.

Key Takeaway: S3 isn't just storage.. it's a fully managed solution for hosting static content with versioning, redundancy, analytics, and a ton of other features.

Aight See you in next Task! :)