[Instructions] Introduction - AWS Getting Started

Before getting into AWS Analytics services, let us take care of the pre-requisites such as users, roles and policies.

* We will not be covering all the policies that are required for a typical project at this time.
* We will start with an s3 bucket, group as well as user, role etc using AWS Web Console.
* Once we create the users and roles, we will setup AWS CLI.
* We will configure AWS CLI using a profile and then validate whether we are able to read from relevant s3 Bucket or not.
* Before going further, you need to ensure that you have a valid AWS Account. I would recommend using a personal account.

[Instructions] Create s3 Bucket

Let us create an s3 Bucket using AWS Web Console. It will be used to store GitHub Activity Data.

* Go to AWS Web Console and go to s3.
* Create a new bucket by name **itv-github**.
* We can also create folders for landing and raw zones for our data.
* The **landing zone** will be used to ingest data from external sources.
* We will store data in the landing zone using JSON. Typically data in the landing zone will be deleted. It will act as a scratch pad and we can delete data that is older than 30 days or as per the SLAs.
* The **raw zone** will be used to store data from sources following our data lake standards. In our case we will use parquet as target file format and partition all the data on a daily basis.
* We will have the data in the raw zone up to 7 to 10 years in most of the cases as part of data lakes.

[Instructions] Create IAM Group and User

Let us create an IAM Group as well as User to have access to GitHub Activity data and associated jobs. Typically, this step is taken care of by the AWS Admin or DevOps team.

* Go to IAM Console and add a group by the name **ITVGitHubGroup**.
* Go to IAM Console and add the user by name **ITVGitHubUser**. Let us add the user to the group **ITVGitHubGroup**. We will give both AWS web console as well as programmatic access to the user.
* We will only use this user using programmatic access. Make sure to download credentials so that we can configure AWS CLI on our desktop or laptop later.
* All the permissions that are attached to the group via policies will be inherited by all the users in the group.
* Typically we will have multiple users associated with a project and hence we create a group and add all the other users associated with the project.
* We might even have different groups with different permissions in a large scale project. For example, developers, testers, business analysts, devops engineers or administrators etc.

[Instructions] Overview of Roles

Let us also get an overview of AWS IAM Roles. They are typically associated with services such as ec2, emr, glue etc to get permissions on other services.

* For a Data Engineering or Data Lake project on AWS, we might have to use multiple services such as Glue, Kinesis, EMR, Athena etc.
* The services typically interact with other services and the standard way to grant required permissions is by creating IAM Role.
* We can create roles using IAM Web Console. Similar to Users or Groups, roles also inherit permissions via policies.

[Instructions and Code] Create and Attach Custom Policy

Let us create a custom policy and provide required permissions on the s3 bucket created earlier.

* We grant permissions to roles or to users via groups by attaching policies with them.
* We can either use AWS Predefined Policies or create custom ones.
* In this case we will create a custom policy by name **ITVGitHubS3FullPolicy** by adding a custom definition.

1. {
2. "Version": "2012-10-17",
3. "Statement": [
4. {
5. "Sid": "ListObjectsInBucket",
6. "Effect": "Allow",
7. "Action": [
8. "s3:ListBucket"
9. ],
10. "Resource": [
11. "arn:aws:s3:::itv-github"
12. ]
13. },
14. {
15. "Sid": "AllObjectActions",
16. "Effect": "Allow",
17. "Action": "s3:\*Object",
18. "Resource": [
19. "arn:aws:s3:::itv-github/\*"
20. ]
21. }
22. ]
23. }

Let us attach policy to users via group.

* Go to the **ITVGitHubGroup** and attach the policy **ITVGitHubS3FullPolicy**.
* All the users which are already added as well as users that will be added in future will automatically inherit the permissions to manage objects in the bucket.

[Instructions and Code] Configure and Validate AWS CLI

Let us configure AWS CLI to access the services and resources related to our GitHub Activity Project.

* We can use aws configure command to configure the credentials. We will configure AWS CLI for GitHub Activity project under **itvgithub** profile.
* Run the below command copy paste Access Key as well as Secret Key when prompted. Optionally you can also configure the default region.

aws configure --profile itvgithub

* The credentials will be saved under **.aws/credentials** in our home directory.
* Here is the command to list files under our s3 bucket using the profile.

1. # aws s3 ls <bucket\_name>
2. aws s3 ls s3://itv-github --profile itvgithub