Creating a LAMBDA FUNCTION:

Under Lambda -> Functions -> Create function.

There are three options:

Author from scratch, Use a blueprint, Container image.

Use a blueprint means using the inbuild system. For example: triggering the lambda function when a S3 bucket is created or creating a microservice which interacts with DDB table.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

Or use can create the image from the DOCKER using your personal laptop using the option of “Container image”.

Creating using the scratch:

A screenshot of a computer

AI-generated content may be incorrect.

Following are the languages supported for this:

A screenshot of a computer

AI-generated content may be incorrect.

Under Advanced Settings-> Enable function URL, if you enable this one, we’ll get the public IP address to access the application which has been written in the lambda function.

If we don’t enable this, still we can run this inside AWS, but cannot access this outside.

A screenshot of a computer

AI-generated content may be incorrect.

People with IAM access or anyone.

A screenshot of a computer

AI-generated content may be incorrect.

Then click on Create function. And lambda function would be created.

A screenshot of a computer

AI-generated content may be incorrect.

Trigger and destination are the main parts of the lambda.

Under Code, there is sample code for the lambda function.

A screenshot of a computer

AI-generated content may be incorrect.

lambda\_handler is the first function that has been called, inside that function, we can call any function, specify by any name.

We can also upload file for the lambda function that would be written inside the VS code.

A screenshot of a computer

AI-generated content may be incorrect.

We can also use the environment variables in this:

Under Configuration -> Environment variables. Click on the Edit and we can add any environmental variables.

In the same way, triggers can also be applied to.

Under the permissions section, the AWS by default adds a role for the lambda functions.

And if you want to assign another role which is already there, then we need to do this while creating the lambda function as follows:

A screenshot of a computer

AI-generated content may be incorrect.

And you can add any role based on the service which you want to access for the lambda functions.

Function URL is to access the lambda functions from the internet:

A screenshot of a computer

AI-generated content may be incorrect.

Also we can add this lambda function inside a specific VPC:

A screenshot of a computer

AI-generated content may be incorrect.

COST OPTIMIZATION:

This is the major responsibility of the DEVOPS engineer.

A Developer has the IAM access and has the right to create the EC2 instance and attach a volume to that EC2 instance. And the volume is filled with the organization which is very sensitive to the organization. And they have taken backup of the volume everyday which is also known as SNAPSHOTS.

And after some time, user deleted the EC2 instance, and let’s assume it is using the externalize volume, which the user was forgot to delete. So, the volume is not deleted, and the snapshots are not deleted. So, AWS would charge you for these things.

In the similar way, user has created the S3 buckets and forgets to delete the buckets and there is content in those buckets. So, AWS starts charging those buckets.

**STALE RESOURCES:** If there are resources which are created and not used and user forgets to delete those resources, those are stale resources.

COST OPTIMIZATION USING DELETION:

We’ll be using lambda function and write the code in the python for the lambda functions using a module called BOTO3, which would be talking to the AWS APIs, which would get the information of the EBS snapshots which is discussed above. And we need to delete the snapshots using the lambda functions which are not in use and to trigger the LAMBDA functions, we need to use the CLOUDWATCH.

A diagram of a computer code

AI-generated content may be incorrect.

**Scenario:**

User has the EC2 instance with a volume and multiple snapshots.

Case 1: Deleting the EC2 instance only. In this case, volume and multiple snapshots would be there.

Case 2: Deleting the EC2 instance along with volume. In this case, multiple snapshots would be there.

**The process would be to fetch all the EBS snapshots and other one is to filter out the snapshots which are stale.**

**QUESTION: If we intentionally keep the snapshots, and delete the volume and EC2 instance attach to that, then is there an issue…? We can keep that snapshots but we need to give the buffer time, like if that snapshot is not getting used within 6 months or for other span of time, then we need to delete that snapshot.**

Creating an EBS volume snapshot (COPY OF THE IMAGE) using the EC2 instance:

Simply create an EC2 instance with default volume.

And, after that we need to check the EC2 dashboard and check the volume, which would be the volume of the currently created instance.

Under EC2 instance -> Storage -> We can check the volume.

A screenshot of a computer

AI-generated content may be incorrect.

After that create the snapshot and simply create the snapshot using the volume recently created.

A screenshot of a computer

AI-generated content may be incorrect.

The snapshot would be created then:

A screenshot of a computer

AI-generated content may be incorrect.

Now, the person deletes the EC2 instance, but forget to delete the snapshot and volume, so we need to use the LAMBDA function to delete those stale resources.

Creating a lambda function:

A screenshot of a computer

AI-generated content may be incorrect.

Paste the code in the lambda function:

A screenshot of a computer

AI-generated content may be incorrect.

And then click on the deploy button. And then click on the Test button and give the event name as test and then save.

A screenshot of a computer screen

AI-generated content may be incorrect.

Then click on the Test, this is the manual invocation, if this needs to be done using the cloud watch, then we don’t need to do this manually.

A screenshot of a computer

AI-generated content may be incorrect.

This would fail due to certain permissions. And under configuration-> Edit and change the default invocation time to 10 seconds. **But default execution time for the LAMBDA is 3 seconds.**

A screenshot of a computer

AI-generated content may be incorrect.

**AND LAMBDA PARAMETER is also a criterion for charging, so we need to make this as low as possible due to charge issue.**

So, for the above issue, we need to provide the permission, so we need to go to that role and add the permissions, which is describe snapshots and delete snapshots.

A screenshot of a computer

AI-generated content may be incorrect.

And then add the permissions and attach policies. Also snapshots policy is not there, so we need to create by ourselves that policy.

A screenshot of a computer

AI-generated content may be incorrect.

Select policy editor as EC2 and then:

A screenshot of a computer

AI-generated content may be incorrect.

Search for the snapshots and add that and then click next.

A screenshot of a computer

AI-generated content may be incorrect.

Resources needs to be All.

A screenshot of a computer

AI-generated content may be incorrect.

Then give the name of the policy and then click on the “Create Policy” button.

A screenshot of a computer

AI-generated content may be incorrect.

Attach same policy to the LAMBDA.

A screenshot of a computer

AI-generated content may be incorrect.

After that, running the lambda, there is again an issue, so need to give the other permissions as well or need to create a new policy with “DescribeVolumes” and “DescribeInstances” pemissions.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

Attach the same policy to the Lambda.

A screenshot of a computer

AI-generated content may be incorrect.

Again, click on the Test button and check.

A screenshot of a computer

AI-generated content may be incorrect.

Delete the EC2 instance and volume gets deleted by default.

A screenshot of a computer

AI-generated content may be incorrect.

After Instance deletion, snapshot is still there.

A screenshot of a computer

AI-generated content may be incorrect.

Again click on the “Test” button and check whether snapshot is deleted or not.

A screenshot of a computer

AI-generated content may be incorrect.

And the snapshot got deleted:

A screenshot of a computer

AI-generated content may be incorrect.

ANOTHER EXAMPLE:

Create a volume with 1 GB.

A screenshot of a computer

AI-generated content may be incorrect.

Create the snapshot from the same volume:

A screenshot of a computer

AI-generated content may be incorrect.

After executing the program again, the snapshot would be deleted, but volume won’t get deleted as the program is designed in such a way that would delete the snapshot whose volume is not associated with any EC2 instance.

A screenshot of a computer

AI-generated content may be incorrect.

**CODE EXPLANATION:**

Boto3:

A screenshot of a computer

AI-generated content may be incorrect.

Using the cloud watch to run the lambda function:

Create a rule: Cloudwatch -> Events -> Rules.

A screenshot of a computer

AI-generated content may be incorrect.

Instead of running the rule on an event pattern, run this on a schedule.

A screenshot of a computer

AI-generated content may be incorrect.

Click on the “Continue in EventBridge Scheduler” button which means this rule is creating the bridge between the lambda function and the cloud watch.

A screenshot of a computer

AI-generated content may be incorrect.

Click on the Next and that bridge would be created.

A screenshot of a computer

AI-generated content may be incorrect.

But need to create this event bridge when necessary, as this would also add the cost to the AWS, as this would be executing every day at the same time.

**CLOUD FRONT DEMO AND INTEGRATION WITH S3 BUCKET:**

CDN stands for content delivery network.

CDN…?

A person from Australia uploaded an image on the Instagram and Instagram copies the image on the central database.

Now if the CDN is not there so, and the central database is at the US location, so the person from India need to pass from too many routers to access that image and then again through number of routers to get that image which would be creating the Latency.

What CDN does is, it creates the multiple local copies of the image placed in the central database and then return the same to the person, who tries to access that which makes the process faster.

A diagram of a company

AI-generated content may be incorrect.

S3 bucket would be containing the images, videos and other large files. Also, we can host the website on the S3 bucket, but that won’t be a good practice, as user has directly access to the S3 bucket, then that won’t be good, security concerns would be there and latency to load the audio video files and cost would also be there.

So, the concept of CDN is there for this thing which caches the content in the nearest edge locations. And CDN service for the AWS is cloud front.

If one user is trying to access the data from the S3 bucket from NEW DELHI, then for the first time, the data would be returned using the CDN, and for the next time, that data is getting cached at the edge location(nearest location of the user) from where user is trying to access, which makes the response much faster and lower the latency. Same is for the user which is accessing sitting in the Canada.

*CDN service for the AWS is CLOUD FRONT.*

Example: We’ll be hosting the website related stuff in the S3 bucket using the Cloud front and host the website somewhere else in the EKS or Kubernetes cluster or any EC2 instance.

Creating a S3 bucket:

A screenshot of a computer

AI-generated content may be incorrect.

Make sure that the following option is checked so that user is not directly able to access it, only CLOUD FRONT would be able to access that.

A screenshot of a computer

AI-generated content may be incorrect.

Also need to enable the bucket versioning as well.

A computer screen with a white box

AI-generated content may be incorrect.

Bucket would be created.

A screenshot of a computer

AI-generated content may be incorrect.

Under bucket -> properties -> enable static website hosting.

A screenshot of a computer

AI-generated content may be incorrect.

“Redirect requests for an object” redirects the request to another bucket or domain. But we won’t be choosing that option.

A screenshot of a computer

AI-generated content may be incorrect.

Index and error page and save the changes.

A screenshot of a computer

AI-generated content may be incorrect.

After that upload the index.html and other resources as well.

Copy the URL from the static website hosting and open that which would be forbidden as we have disabled the public access which needs to be accessed from the CLOUD FRONT.

A screenshot of a computer

AI-generated content may be incorrect.

A computer screen with a white background

AI-generated content may be incorrect.

Creating the CLOUD FRONT:

Search for the Cloud Front -> “Create distribution” button.

Origin domain can be S3 bucket or use the API gateway or elastic load balancer.

A computer screen shot of a computer

AI-generated content may be incorrect.

Use S3 bucket:

A screenshot of a computer

AI-generated content may be incorrect.

Origin access needs to be selected: Legal access identities means consider a user that has an access to the bucket.

A screenshot of a computer

AI-generated content may be incorrect.

This would be a virtual user which would be able to access that. Click “Create new OAI”.

A screenshot of a computer

AI-generated content may be incorrect.

Click Create. And we need to use yes option to update the bucket policy. Which allows only that OAI user to access that bucket only.

A screenshot of a computer

AI-generated content may be incorrect.

Other settings would be same, and you need to select based on customer present in preferred countries. And, also you can add the domain name to this site as well.

A screenshot of a computer

AI-generated content may be incorrect.

When user enters the website name, then the following page would be rendered like clicking “/” with website name.

A screenshot of a computer

AI-generated content may be incorrect.

After that, just click on the “Create distribution” button.

A screenshot of a computer

AI-generated content may be incorrect.

Link to access the S3 bucket.

A screenshot of a computer

AI-generated content may be incorrect.

Now that cloud front access would be added to the bucket policy under the S3 bucket -> Permissions

A screenshot of a computer

AI-generated content may be incorrect.

Meaning, Principle is cloud front which has actions of “GETOBJECT” to access the resource named S3 bucket.

CLOUD FRONT can be able to access the S3 bucket, but user still won’t able to access that bucket publicly.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

**AWS ECR:**

ECR is a service in AWS used to store and managed containers.

ECR is divided into three parts:

E -> Elastic

C -> Container

R -> Registry

ECR is container registry is like DOCKER HUB, QUAY.IO, GCR (Google cloud service)

Container registry purpose is to store the docker images.

Like you have code in your personal laptop, that you want to share with someone else, you can share that using the container registry and the other user can pull the same image from the container registry, or you can share that image with anyone in this world.

Elastic means this service is highly scalable and available in nature.

Scaleable simply means you can increase the capacity of AWS services to accommodate any number of resources. Which means you can store any number of container images and pay as you use and that would be available all the times which is manages by AWS.

A whiteboard with writing on it

AI-generated content may be incorrect.

ECR vs DOCKER HUB:

Repository created in the Docker hub is public in nature while ECR are private ones.

In the AWS, you can use the IAM users and roles with the ECR, as this has been managed by AWS, but in the DOCKER HUB, each individual needs to create the separate accounts for the access. Like for more than 10,000 users, you need to make the account and need to manage the access as well.

Search for the ECR:

A screenshot of a computer

AI-generated content may be incorrect.

Create repository:

A screenshot of a computer

AI-generated content may be incorrect.

Private access is managed by IAM and repository policy permissions.

Public access is when anyone can be able to access that repository.

Tag immutability is to prevent the images from being overwritten by subsequent images using the same tag. And it needs to be disabled to allow image tags to be overwritten.

A screenshot of a computer

AI-generated content may be incorrect.

Scanning images at the push, that can be enabled or disabled in the ECR as well.

A screenshot of a computer

AI-generated content may be incorrect.

After that click on the “Create repository”.

A screenshot of a computer

AI-generated content may be incorrect.

Check the option of the “View push commands” to check how to push or pull the images under the created repository.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

Firstly, we need to install the AWS CLI, which we have already done. We’ll get the following output, if the AWS CLI is configured.

A computer screen with a blue background

AI-generated content may be incorrect.

By default, “docker login” command work for the docker.io.

And for the ECR, we are using the same command to login for the ECR using the username and password.

First command before pipe “|” is fetching the password from the ECR and is acting as input and the other part is submitting the output to the docker login.

*So, first part of the pipe “|” is input and second part of the pipe is taking input from the first part of the pipe.*



If we are using the root account, then the access is there, otherwise if we are not using the root account, then we need to explicitly gives the access push the image using the ECR.

For the IAM user:

Adding the permissions:

A screenshot of a computer

AI-generated content may be incorrect.

Attach policies directly:

A screenshot of a computer

AI-generated content may be incorrect.

Search for the ECR and then assign this permission to the IAM user.

A screenshot of a computer

AI-generated content may be incorrect.

After that create a new Dockerfile and paste the code:

A computer screen with green text

AI-generated content may be incorrect.

A screen shot of a computer

AI-generated content may be incorrect.

After that, execute the 3rd command to build the docker image.

A screenshot of a computer

AI-generated content may be incorrect.

Type “docker images” and check your image is created or not.

A screen shot of a computer

AI-generated content may be incorrect.

Need to add the tag to the created image:



After push command image would be there:

A computer screen with green text

AI-generated content may be incorrect.

Image would appear at the AWS.

A screenshot of a computer

AI-generated content may be incorrect.

**AWS ELASTIC CONTAINER SERVICE:**

ECS vs EKS.

How ECS is different from Kubernetes based environments like EKS, OpenShift.