20/04/22

Auto scaling :-

* Scaling up went Load is Max.
* Scaling in when Load in Min

Launch config

creat Launch config.

Name :- My-LC

AMI :-

Instance type :- t2.micro.

Assign security group :-

New security group

✓ existing one

Key pair

Choose existing pair.

Existing pair :- Vamshi-devops

| create Launch Config |

NOW, create Launch Conf group.

Auto Scaling Group

Create Auto Scaling group.

Name :- my-ASG

Launch template :- re-usability

✓ Launch config :- NO re-usability

My-LC

Next

VPC

⇒ Select in all the Zones (Available Zones).

NO Load balancer.

Health checks

Additional settings

monitoring.

☑ Enable the cloudwatch.

Group size.

mini capacity	1
max capacity	5
desired capacity	1

Scaling policy :-

Target Tracking policy.

⇒ Metric type

Average cpu utilization } — it is only for scale up.

Target value

70

instance need

180 sec

↓

But in cloudwatch All alarms we can see that scale in is also created

⇒ Add notification

SNS Topic :- email

Create auto scaling Group

NOW to check auto scale up and in so, put the load on the server for that

Sudo amazon-linux-extras install epel -y

Sudo Yum install stress -y

Stress

stress --cpu 90 --timeout 420 &

"&" it will run in backed in linux 'd' in docker

"top" → to check the load on the server

on only Tracking policy but we also have

Step scaling, scaleout in dynamic scaling.

If we want to clean up auto scaling then go to auto scaling and delete it.

*RDS :- Data base. (Transcational data).

Relational data base server.

Data base

Create DB.

Standar create (or) Easy create.

**How many day's backup can be maintained on RDS :- 35 days.

⇒ Log exports :- Go with all

create data base.

Data base is ready.

now, to connect this RDS we have to have a Server and set-up mysql on the server and connect to the data-base

21/04/22

RDS

↓

connectivity and security

Endpoint and Port.

On server to connect to RDS

* mysql -h endpoint -P 3306 -u admin -p

Password : the Passw which set earlier while RDS set-up

⇒ for that install mysql on the server

* Yum install mysql

* mysql -h (database from aws (RDS)) -P 3306 -u admin -p

Password :

Now, this can be seen in 'cloudwatch' in logs

'log Groups'.

After the completion we can delete the RDS.

Relational database server

Cloud Formation service :- From (AWS) side.

cloud Formation templet :- create and Manage resource with Templets.

* AWS Template Format version
* Description
* Metadata
* Parameters.
* Mappings.
* conditions
* Resources :- Mandatotory section. (required section).
* Outputs :- This section is for outputs of cloud formation displayed on cloud formation console.

CloudFormation (AWS).

use a Simple Templet.

Lamp stack.

(code) → View in designer

Now, there is a GUI where we can create the Template

Here, we can View has a Component (or) as Template

Create template Designer . by UI .

Resource type

drag and drop this one
to the right side, what
-ever is needed

(or)

we can create with stack

in real time we will use Terraform (or) cloud formation

***

EFS :- Elastic File System :-(To manage the data .

To store dynamic data EFS is used.

Elastic file systems are automatically scalled.

Now,

Elastic file system

Create file system. (and fill in the details)

EFS will also have security Group.

with EFS two different servers can share the data.

b/w them

22/04/2022

S3 Buckets :-

To Manage the static data (images, video).

the date which can't change is static data.

Features :-

* Version control
* Antomatic Backup
* Storage classes
* Website hosting (ex:- Netflix).

⇒ Buckets are folders with unlimited storage but on the content level storage is there which is 5TB.

Global Serviceses :-

* IAM
* S3 Buckets
* Route 53.

Amazon S3

⇒ creati Bucket

Bucket Name :- "Name should be unique".

Aws Region :-

⇒ Object ownership

Block public access

Bucket Versioning :-

Default encryption :-

Then Go to the created Bucket and start uploading

the Bucket

create floder upload

Files and Folders Add files Add folder

Destination add files here.

permission

properties

→ upload then close

If we click on the file which we added then we can find a URL

⇒ Object URL

↓

Pass where we want to consume this item.

→ to access these we need to give the permissions.

So, go to Bucket first.

⇒ Permission.

→ Blocks public access.

and change the access and grant permissions.

and | Save changes |

Now, Allowing the access on content level. (or) item level.

⇒ Objects

→ Click on item.

→ make public using ACL

Now, after access is given then we can see the content using the URL

Properties:-

Versioning is present here. for that.

→ Bucket Versioning

☑ Enable.

To see the versions enable the ☑ Show versions in "objects"

If we upload a latest version then we have to give the permissions again.

→ Default encryption :-

☑ Enable

○ Amazon S3-managed keys (SSE-S3).

☑ AWS Key Management service key (SSE-KMS).

AWS KMS Key → To create our own key.

Key Managment service.

Key type.

Symmetric Asymmetric.

Key administration :- Who can manage the key (user)
(User)

Vamshi

Other AWS account :- Item level.

then [Finish]

Now, add this created key for the encryption.

Properties :-

Server access logs :-

Enable.

and then choose the path (Item).

→ To Deploy the Item (or) Launch the content using.

S3 bucket.

→ Static website hosting

☑ Enable

→ Hosting type

☑ Host website static.

→ Index document :- ex :- index.html.

↓

Launch page.

→ Error document :-

[Save changes]

Now, we can see website Domain Name. we can access the content with this.

Permissions :-

→ Bucket policy

Edit

Policy example

policy Generator.

↓

different policies can be created using this like, IAM, SNS, EBS, S3

Effect :- Allow.

principle :- *

Aws Service :- Amazon S3.

Actions :- (Give what actions you want).

Amazon resource name (ARN) :- (Bucket policy we can find 'ARN').

Add statement

Generate the policy → copy this policy

Done

Paste the policy in Bucket policy.

Object ownership :-

Access control list (ACL) :-

Grants the permission which are needed to access

then

Save changes

Metrics :- Items placed in buckets, Objects and size.

Mangement :- Manging the storage class (data).

life cycle Rules :-

Rule actions :- Current version, non current version (storing of this are not).

Storage class :- S-tandard - IA → 30 days → Intelligent Tiering → No limit days → One zone IA 30+30 = 60 days 30 days (or)

Replication Rules :-

Glacier instant Retrival → 90 days → Glacier flexible retrival → 180 days

only one zone.

60 days

Archived data

it will take hours to retrive data.

Glacier deep Archive

Hours of retrival

Automatic storage.

Name :- Keep this bucket in some other region.

in :- North Verginea

Status Rule :-

⊙ Enable

→ Choose a rule

- Apply to all

→ Destination

- Choose a bucket in this account (AWS)
- Specify a bucket in another account (AWS)

For that we have to have an account in another region Before that Disable KMS key if any is Enabled

Create bucket

Name :-

AWS Region :- Give target region

Bucket versioning :- whenever we go for replication both 'source' and 'destination bucket should be enabled with versioning.

- enable

Create bucket

Now, Buck Name | Browse

Choose the target path

IAM role

✓ choose from existing IAM role.

○ Enter IAM role ARN

IAM role.

[Create new role.]

Encryption

✓ Enable

[Save.]

→ Replicate existing objects

✓ Yes, replicate existing objects

[Submit]

Batch job

report

✓ All tasks.

path to completion report destination.

[Browse S3]

[Save]

Now,
the existing items are copied to the target bucket in another region.

Access point :-

Who can access the datasets in S3.