1.5 or 2 hrs
friday

AWS Day 5

Agenda:
- aws-CLI-configuri, dry run, create ec2, profiles
- S3-lab-hosting a website, bucket versioning
- cloudfront-AWS

Lab: to upload files in S3 bucket via cmd not via AWS console

Google: download AWS CLI
AWS command line Interface
↓
Install it
Finish

search command prompt in personal laptop

~~See 4th 3:44~~

Notepad notes:

① AWS access is very confidential for a company/org to give individual access to each employee

② AWS provides 3 types of access: management console, CLI, CDK (Cloud development kit)

It is essential for knowing that which employee is playing what role (IAM: Identity & Access management)

Later, this IAM role will be added as a profile in AWS CLI, by using the access key & the secret key

Note: Ansible/Teraform these are the IAC (Infrastructure ~~Access~~ As Code) & these are the devops tool, we have to be very clear that AWS CLI is not a devops tool. As AWS CLI doesn't use yaml script

AWS CLI uses the AWS API for the step by step work

The person who will be creating the AWS account (email id / password), this will be known as root user. This root user will not be having administrative access (IAM). For using AWS CLI the best practice is creating the IAM user with task specific IAM role. for eg. if we want to create an ECR (Elastic Container Repository) Then we will be creating an IAM user with ECR permission, after that we will be configuring the user & performing the task.

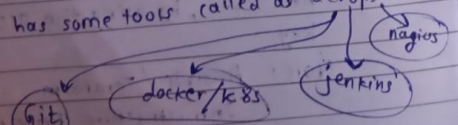
First create an IAM role. After that assign it to a user → Access key & secret key received.

One user can have multiple roles.

* Devops } not the tools
  SDLC } they are process

SDLC: plan → dev → test → build → prod → monitor

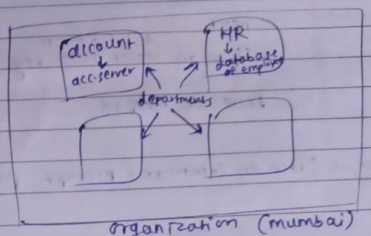
Devops has some tools called as devops tools



Version control system

IAC is one of the component of Devops

eg.



* We can create ec2 by using Ansible / Terraform / aws CLI

Lab

AWS CLI commands:

on command prompt:

type aws ec2 describe-instances enter

type aws configure enter

~~type~~

company laptop → go to aws management console → IAM search

IAM dashboard → Click users on left pane → Add users

name: megna-cli

☑ Access key

next

create group → name: group1-cli-mp

☑ Administrator Access

create group

next

next

Click on secret access key (Access key ID) copy it

Paste on notepad

personal laptop → Now on command prompt paste the Access key & Secret key.

Default region name: us-east-2 (see in aws console)

Default output format: json

C:\Users\meghana → M:\> नंतर aws ec2 describe-instances

ctrl+c दाबणे.

type aws help

Notes: Commands starting with minus symbol are called switch.

And, commands starting with double minus symbol are called variables.

Every AWS CLI commands start with the word 'AWS' as AWS is the Client.

eg. git add hello.py
↑
Client

las continue ~~Now on command prompt type aws create --bucket~~

Now on command prompt: type →

aws (space) s3api (space) create-bucket --bucket meghana-bucket-989282 space --region space us-east-1

Then type: space (S)

aws (S) s3 api (S) put-object (S) --bucket (S) meghana-bucket-989282 (S) --key (S) meghana-bucket-989282/file1.txt (S) file1.txt

file1 { Then type: aws (S) s3api (S) put-object (S) --bucket (S) meghana-bucket-989282 (S) --key (S) file1.txt

Then type: aws (S) s3 (S) sync (S) s3://meghana-bucket-989282 (S) file1.txt }

file2 { Then type: aws (S) s3api (S) put-object (S) --bucket (S) meghana-bucket-989282 (S) --key (S) file2.txt

Then type aws (S) s3 (S) sync (S) s3://meghana-bucket-989282 /file2.txt (S) paste here }

काय paste करायचं आहे →

Go to This pc → C disk → Users → Meghana → desktop → copy path by clicking on [v]

This doesn't cycm?

see next page from "continued"

Then type

aws s3 ls s3://meghana-bucket-989282

we get 2 files

→

Continued → (another way)

This PC → C → Users → Meghana → Desktop

create file notepad File type abcd in it & save it ← Create folder named data

Now type command:

~~aws~~ dir

cd DESKTOP

dir

aws s3 cp data s3://meghana-bucket-989282 --recursive

file gets uploaded ✓ in S3 bucket

Check it in AWS management console

* > 160 GB वाले files को upload करने के लिये S3 bucket created by CLI ✓

continue →

New

* Task: Create website using AWS CLI

steps: ① Create bucket with bucket name meghanawebsite

Cmd command → C:\Users\Meghna~Desktop यहां पर:

aws s3 api create-bucket --bucket meghanawebsite --region us-east-1 enter

② Putting our website as object inside bucket

aws s3 api put-object --bucket meghanawebsite --key website.html enter

③ List the object that is inside our bucket to confirm placement

aws s3 ls s3://meghanawebsite

④ Uploading code to blank website.html file with copy command

aws s3 cp website s3://meghanawebsite --recursive

(5) Set S3 bucket as website

aws (S) s3 (S) website (S) s3:// meghanawebsite/ (S) --
index-document (S) website.~~html~~ txt

(6) Presign URL of S3 object for temporary access

aws (S) s3 (S) presign (S) s3:// meghanawebsite/website.
txt ~~html~~

→ link generates

Copy & paste it in chrome tab

* Content Delivery Network (eg. cloudflare)

Dubai — wp voice call

North korea: — netflix, etc

CDN → Distributed Network, which provides services to end user/client.

- Geographical Region <> Services (website, DNS) will be
  - blaklist
  - whitelist

If we want to use CDN - at aws then
- Cloudfront is the option.

Lab

cloudfront search ~~create~~
5:00 to ~~5:40~~ ~~5:00~~
Public ✓

2/9/24 — Sarana

Sir

~~megaurs notepad aide~~

~~[illegible]~~

Lab :

S3 → create bucket → name → ACL enabled

↓

upload ← create ← Bucket owner preferred

↓

.png file select → specify individuals ACL permissions

↓

☑ ☑

☑

☑ ☑

☑

upload

Now search Cloudfront → create → name, select bucket created

↓

public → create → Geographic restrictions India, US

↓

In → General ↓ copy domain name

↓

paste in chrome/ http चाहिये

* Introduction : Devops:

most basic principle → CI/CD
continuous integration / continuous deployment

website : Flipkart.com
mobile App: Zomato.com /swiggy

will they use same deployment platform (EC2) or S3 ?

Deployment platform will be depending on Project

Best deployment platform for mobile app is aws amplify

~~Deployment~~ platform for php/node/react/javaspr
spring → EBS (ElasticBean stalk)

Static website → S3
Dynamic website → EC2

Deployment platform : Heroku
Firebase
Gitlab
Github
AWS
GCP
Azure

SDLC → Software Development life cycle

waterfall model : upstream → data going
downstream → data incoming

CI → CD = CI/CD pipeline by yaml, Terraform

dev | test | prod

Down stream ← → upstream

CI → falls in development phase

CI → continuous integration of code

For maintaining CI, we need VCS (version control system) eg. github

yaml script supports linux commands?
• cd ls ls-

Lab →

Github code deployment

github account login करना
Create new repository → name (AWS-training), public, ☑
create repository

Click on settings → Click on ~~Github~~ pages on left pane

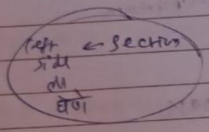
↓

Click on <> code upside tab

← click on "creating a new file" link

↓

name → any html code copy & paste here



Settings → pages → Deploy from a branch

↓

None ← main ← select it

save

↓

refresh page

Now in custom domain copy whatever is written after "other than"

eg. meghana.github.io

in chrome http://paste/index.html ← in capital

http://paste it/name of repository/index.html

Now, open index.html file created & make some changes in that code (delete some code lines)

commit changes

.

→ paste this link again in chrome

The changes occur