Day 6

Agend: AWS devops (CI/CD), ECR, ~~ECB~~ EBS, ECS

ECR: Elastic container Repository
EBS: Elastic Beanstalk (deployment purpose)
ECS: Elastic container service

49:00 Lab

Search EBS
↓
Elastic Beanstalk → Create application
↓
name → Platform → Java
↓
sample
↓
create application

Click on URL: "...beanstalk.com"

~~Go to first tab now of chrome~~ (configuration on left pane)
→ we can edit here in configuration.

Now go to applications versions → ☑ → Upload
↓
~~description~~ → select file (platform = java so java file needs to be uploaded) → upload → file gets downloaded
→ thus deployment नाही झाले

Imp: for deleting → go to Application versions → ✓ → delete

~~On chrome: type javacode for sample application~~

* S3 Bucket मध्ये file upload करणे, & attach S3 to EBS हे भी एक option है.

~~Now go to~~.

See next page

Theory Notes: Artifact: Deployment ready code
: code which has passed through build pipeline

Types of pipeline in aws:

Build Release

Test stage comes under build pipeline
There will be no build stage at EBS

Lab: Form a code named source code. And link it to EBS & if we make change in source code then it should reflect in EBS
i.e. Deploy code in EBS

Sir की github की link दिखी होगी → click it → fork
https://github.com/sidoncode/node-hello-aws-codepipeline

Now search code pipeline → Pipelines in left pane

create pipeline → name → New service role

Now go to github tab of chrome

click on fork (on meghana personal github account)
on sir's account link then it will create repo on↑

owner / repository name
meghana / node-hello-aws-codepipeline

Description देना

Create fork ⇒ This will create repository

Click on copy the chrome link above
https://github.com/.....

Now, after this do

click on next

Add source stage → Source = github version2

next

Now connection में connect to github

name देना (any name)

connect to github ← Authorize

install a new app → click on meghana

All repository → install → password डालना

Github Apps में number होगा → then click connect

After ready to connect

Repository name देना → (account-name / repo name) meghana/node-----2022

Branch : master

☑

Codepipelinedefault

↓

next

~~Now, go to github account → settings Actions~~

↓

Add build stage नहीं → skip it

↓

Add deploy stage :

Select AWS Elastic Beanstalk

Application name: ~~select it~~
Env name: ~~select it~~
~~Click on previous~~

for this, search EBS after duplicating tab

↓

create new application

↓

~~name देना~~

↓

Create

Create new one

web server env

↓

next

↓

Platform : Node.js

create environment

Now, Application name } select it now which is created
Env name

getting error?

↓

Previous

↓

Add Build stage : ~~next~~ skip

Previous

Add deploy Stage : select AWS Elastic beanstalk

Now next next → Applicatn, env name ✓

↓

next

↓

create pipeline

↓

copy link & paste in chrome tab → hello Node22 आएगा!
↓
EBS से

Now go in github,

index.js
↓
edit the code by cutting Node22
↓
commit directly to master branch
↓
commit changes

Now go to pipeline tab, → pipeline in left pane
↓
Deploy: in progress
↓
then ← success
↓
refresh the link of the code
↓
Hello! ✓

successful!

* ECS:
Elastic container service

Container:
Devops
↓
Blame Game

containers
tools
docker, kuberenetics

Architecture of container:

Lab: Docker:

In chrome tab type & search "dockerhub"

↳ click on Docker Hub Container Image library | App Containerization
↓
Register → username: Meghana
email: ✓
password: ✓

☑ I agree
Iam not robot
↳ Signup

email, password → Continue
↓
personal ✓
↓
verify email address

classmate
Date ______
Page ______

click on Explore (near search bar)

new tab वर chrome on google } docker playground

↑

open on the link →

https://labs.play-with-docker.com/

↓

login

↓

docker

↓

accept

↓

start

Add new instance वर click करणे

type: dockerⓈimages

dockerⓈpullⓈubuntu

dockerⓈpullⓈmysql

dockerⓈimages → we get number of images in node1 container

dockerⓈrunⓈ-itⓈ--nameⓈmyubuntuⓈubuntu

mkdirⓈmeghana

ls

→ blue मध्ये येतं meghana

apt-getⓈupdateⓈ-y

apt-getⓈinstallⓈfiglet

figletⓈmeghana

figlet is software for graphical visualization

classmate
Date ______
Page ______

exit

dockerⓈimages

dockerⓈpsⓈ-a

dockerⓈcommitⓈ write container ID eg. fdfe60deb5f9..

Now go to https://hub.docker.com/

click on repositories

↓

create

↓

name = mp

↓

Public → create

eg. meghana/mp असं येईल काहीतरी

Now type in prompt:

→ docker commitⓈfdfe..fⓈmeghana/mp:commit

dockerⓈimages

dockerⓈpushⓈmeghana/mp:commit1

denied ✓

dockerⓈlogin

Username } password } टाकणे

docker Ⓢ push Ⓢ meghana/mp:commit1

refresh the docker hub tab

see all

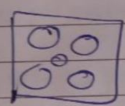
Now come on black screen again → close session
↓
again start
↓
Add new instances
↓
type docker Ⓢ pull Ⓢ meghana/mp:commit1
docker Ⓢ images
docker Ⓢ run Ⓢ -it Ⓢ --name Ⓢ pizza Ⓢ meghana/mp: commit1

now we come under pizza
ls
→
figlet Ⓢ meghana

* ECS : Elastic container services.
↳ Deployment platform
Docker Hub → store
docker playground → code/cmd

AWS:
Docker hub → ECR (Elastic container Repository)
docker playground → ECS (Elastic container service)

Create cluster :
group of container/node/server/ec2 instance, etc

ECS → cluster of instances 

Fargate is automated technology by AWS runs behind ECS
fargate is same as auto scaling.

Lab
search ~~docker~~ ecs in aws console → EC2 linux
↓
t2.micro ← cluster name ← next step
देणे
↓
1 → create

Lab
Cluster create केलेले click करणे → get started
↓
← nginx

code commit lab: } we add index.html file to repository using code-commit of AWS

→ go in command prompt → type git (enter)

Now in management console → code commit
↓
create
↓
name (meghana-repos)
↓
create
↓
clone URL
(clone HTTPS)

in command prompt : git clone paste the URL

Now in aws console, create file वर click ~~करो~~

असं येणार : Developer tools > codecommit > Repositories > meghana-repos > File

Copy code (html hello word code) in this blank repos.

File name : index.html

Author name:
email :

commit changes

Now ~~search~~ duplicate tab → search
elas search → elastic Beanstalk → create → webserver → create
~~codepipeline2 search करो → create~~

name → meghna-app

name → meghna-env

managed Platform → PHP → ~~upload your code~~

↓ sample applica ~~latest local file~~

↓ create env

Now code pipeline search → piplines → create

↓ name

✓ Allow ← new service ← name

↓

Default → next → AWS code commit

↓

main ← meghana-repo ← next

↓

AWS codepipeline → next → skip build stage

↓

next ← names ✓ ← select AWS ElasticBeanstalk

↓

Create pipeline → edit → edit deploy के बाद add stage → name → add

↓

mannual approval ← name दो ← add action group

↓

Now to go beanstalk → click on created application →

↓

copy URL → paste in URL for review → Done → save → save → Release change

↳ Approve it (by clicking on review)