Java jdk – Java-Home
Java\jdk\jib – Java

PAGE No.
DATE / /

Experiment ① & ②

Step1: open Cloud 9 IDE
Step2: Create Environment
Step3: Select the environment & click on open
Step4, new window will be open & click
Step5, create py file & run from top
& index.html & click on preview

Experiment No. ③ & ④
follow steps of Exp ①
Step2, aws-cloud 9 top right click on share
& create new user
- fill username
check provide access options
- check I want to create new user
- autogenerate pass
- attach policies
- & add cloud9Environment member
create user
- download csv file
- open incognito tab & open aws console
sign as IAM, provide ID & pass from
csv file & set new pass
Check the region (N.Virginia) must be
same for both envir

Now go to first created environment cloud
page
type new username & send invite
now go to newly created user & open
(Safari) in cloud9 IDE

Experiment No. 7

Install Terraform on windows machine & Build apply & destroy aws EC2 using Terraform

step 1: Install Terraform from Terraform.io
2. Extract it in C drive & copy the path in System variables.

C:\terraform_1.6.2_windows_amd64

go to C C:\terraform_1.6.2_windows_amd64 & open cmd terminal & check version by

Terraform -version

Now create Iam user with password & admin access & create group with any name with default permission & create user & download (csv file, create access key, checked (CLI)
, copy the access key & secret access key

create a .tf file in C:\terraform_1.6.2_windows_1.6.2 & write following

now create new instance with key value to be keypair name(key10) & PSA .pem & create & copy ami of new instance

create ts file in terraform.exe dir

provider "aws" {

provider "aws" {
access_key = " "
secret_key = " "
region = " "
}
resource "aws_instance" " terraform.tf"
ami = " "
instance_type = "t2.micro"
terraform init, plan, apply, destroy

Sonarqube

copy sonarcube windows path and set path in system variables - (jdk 17)
& open sonarcube visit
localhost (9000)

create project manually
Give name
use global setting
Locally
Generate
download sonar scanner for windows
visit official website
Extract in C drive
set sonar scanner bin & conf file in system variable
create a folder in sonar scanner bin file as (.py) file
open conf file & type

sonar.projectKey = Piyush More
sonar.projectName = Piyush
sonar.token = "
sonar.login = admin
sonar.password = Pm.mane@2652
sonar.projectVersion = 1.0
sonar.sources = .

Save & close

open cmd in sonar-scanner-bin file
& write sonar-scanner.bat

Exp 5: create deployment environment (Elastic Beanstalk)
Give name piyushapp

Step 2: Go to codepipeline
keep default setting

Step 1: create IAM role with Lamda function (Exp 4)
Step 2: Add AWSLambdaBasicExecution permission
Add CloudWatch FullAccess permis.
" Amazon S3 Full access
Step 3: Role details (Exp 12 Role)
Step 4: open S3 bucket in another window
create bucket & give name
enable bucket versioning
Step 5: create Lamda function with python 3.7
with existing role
Step 6: Delete the existing code & write the code
with your bucket name as S3 link

PAGE No.
DATE / /

~~go to test & clic~~
~~save test~~
save file & click on deploy
click on test & give event name
make ~~(?)~~ check shareable option

— create test & click again on test
& generate report of function

← go to cloudwatch → logs → log group
& check latest output

→ ~~(?)~~ go to S3 bucket & ~~(?)~~ object
will be created

→ Delete lambda fn, cloudwatch

(signature)

Expt 22.21

~~i~~ Launch instance
select ubuntu
create ppk key pair
launch instance
connect to instance

sudo apt-get update
sudo ~~apt~~ install docker.io
sudo systemctl enable docker
sudo systemctl status docker
docker --version

Exp 9.

- Get a copy imoishoma/aws-codepipe-s3-coded
- create bucket with bucket version enable .21 D
  & upload the downloaded Github zip file & download it & design ~~Explord~~ up
  codepipelin ~~add~~ Source stage using search gate
  provider Amazon S3
  object copy
  S3://piyexp4// ~~awspipeline-s3-code-dir~~
  ↓ bucket
  ↓ get this from Github repo → dist →
  skip build stage
- → Deploy ~~provider~~ amazon Bean stall
  piyapp-env

create Role with EC2

add permission
AWS Elastic Beanstalk WebTier

choose ~~[...]~~ platform
PHP

choose existing role

choose VPC

choose us-east-1a

on

both

click ~~next~~
next
~~[...]~~ Submit

go to environment
&
wait to launch
click on link below domain