

Effectiveness of Disaster Recovery

AWS Infrastructure Setup :-

1) Terraform Infrastructure :-

1) Create user from IAM

for
terraform

2) Create usergroup with admn access*

3) Create user

4) Create an access key — key should not be

5) Create an S3 bucket -- for storing terra

6) Create access point for bucket

7) Change the name of terraform folder
in infrastructure eg slimriver-aws-ap-south-1

copy & paste
from s3gate

8) edit terraform.tfvars change name =
"slimriver-test" in ".terraform" folder in

S3

refer
video

9) Create CI pipeline for new instance
in "github/workflow". Copy paste "s3gate-
aws.yaml" and change to VPC folder name
we created.

10) In this file change name, env-
paths in or., env - ~~accesskeyid~~
jcb & filters, infrastructer, if: working-di-
rectory, with: file-path.

11) Push changes in github in
SDGT/github/Infrastructure (link: https://
github.com/SDGT/infrastructure)

12) Create pull request main from feature/
VPC.

13) Build failed ⚡

14) Add secret key in settings ⚡

14) Build failed ⚡

15) Change bucket name in terragrunt.hcl in
VPC folder and commit to github - ⚡

16) Build failed ⚡

(Q) Explain

due not
hade down

M	T	W	T	F	S	S
Page No.:	2					
Date:	7/21/25	YOUVA				

- 16) Conforming aws accesskey and Secret key is working. (+)
- 17) Due to some change the bucketname terraform-hol and CI file which we create in that change in env: (+)
- 18) Commit in github infrastructure.
- 19) Build succeed (-) (+) (*) (*) We got moderation for VPC
- 20) Merge the pull request with force
- 21) Squash and merge bcz of many commits it will combine and show only one commit.
- 22) Now VPC is creating.
- 23) VPC created (-)
- 24) Check VPC in AWS VPC.
- 25) VPC was created in AWS.
- 26) Creating EKS:-
- 27) Change the eks folder "main.tf" file "vpc_id" and "private_subnets" (This folder was copied and edited from sealgate)
- 28) change "cluster_name" in eks folder in "terraform.tfvars" but here we are using same cluster of sealgate-SIT.
- 29) We can check vpcid in actions of model we build.
- 30) charge environment and clustername in eks in "terraform.tfvars" in eks folder
- 31) All changes related to eks are done
- 32) Commit all changes in github infra
- 33) In CI file we add eks in modules
- 34) All changes related to eks are done.
- 35) Commit changes in github infra
- 36) Create pull request "main from feat add-eks" in github.
- 37) Build success - Here we get model plan for eks
Here main step is Plan Terraform Module



built
not triggered
automatically
it's branch change
raise issue

M	T	W	T	F	S	S
Page No.:	3					

Date: 7/21/25
YOUVA

38) It will create a 4 node E cluster

39) Now do squash and merge.

Note:- Do squash and merge for releases and infrastructure.

40) Build succeed.

41) In EKS check whether environment cluster is created in AWS.

42) Download kube config from AWS S3 → Buckets → hapi-test terraform tf states → kuberconfigs → devops-admin-hapi-test kubeconfig.yaml. (Not this is admin)

43) Go to EKS 'kubeconfig'

44) Infrastructure is Done.

45) Releases Part :-

46) Copy Selgate-stage dir from SAGT Releases & rename to hapi-test

47) flux setup: - flux config in local system

48) Cmd: - Screenshot 1. img

49) Add ~~the~~ Deploy key in releases SAGT. we can get this github auth secret file in git repository folder.

50) apply it cmd: - Screenshot 2. img.

51) In git repository folder - change the primary & secondary yaml files ignore:

52) ~~Apply~~ change the primary & secondary file in kustomization folder here and change

Change the path:

53) Apply it.

54) Flux setup Done.

55) In ingress-nginx folder forment the nodeports in ingress-nginx-helmrelease file. and also to change subnets which we will get in AWS VPC subnets.

Create:

- 56) Add the storage class by creating new folder named ~~storage~~ "gp3-sc" in SDGT release.
- 57) Move the storage class files to ~~that~~ the gp3-sc folder from ~~in~~ ^{inside} velocity
- 58) Create aws-ebs-csi-driver-helm-release.yaml in gp3-sc folder.
- 59) The namespace and name in aws-ebs-csi-driver-helm release file was edited.
- 60) Change the filename in helm repository folder to aws-ebs-csi-driver-helmrelease.yaml by copy paste any helm repo and change the url which we get from artifact hub aws-ebs-csi-driver-helmrelease.yaml. Commit all in github and push.
- 61) Manually reconcile flux-Screenshot 3
- 62) Rename sit folder to test.
- 63) Change mongo-sec.yaml file ~~name~~ name of mongodb-root-password.
- 64) Create -namespace.yaml in test folder.
- 65) ~~Add~~, commit and push all changes.
- 66) ~~Add~~, commit and push all changes.
- 67) Take backup by cmd: Screen shot

Here we are editing cmd by adding mongoDB password.

Key for

cmd:- mongodump -u
 "mongodb://root:root@192.168.120.41:3030" --out ./dump.

We changed the password here and username is root only.
- 68) Now releases point started.
- 69) ~~Backup and restore~~ Perform DB Dump and restore. Which we did earlier.

6g) we copied changed mongoDB password
 common folder \rightarrow common see .yaml
 After dump restore

- 70) we added cdss service in clinicare
 folder \rightarrow clinicare - bucketed helm release
 .yaml
- 71) Add, Commit and push changes in git hub
 and do reconcile - in flux
- 72) Tr gd \rightarrow clinicare - clinicare - bucketed
 helm release .yaml \rightarrow appointment .copy
 tag and change ~~the~~ ^{copy} - to hapi - test
 change namespace in the appointment
- 73) cm .yaml file to test in hapi - test
- 74) Add, Commit push and reconcile
- 75)