

1

Capstone Project

✓ Git → my-aws-capstone-project (oluştur)

✓ VPC oluşturuyoruz 90.90.0.0/16 (django-vpc)

- DNS hostname enable yapınız

* Enable yaparsanız VPC altındaki instance'lar birbiri ile haberleşir

✓ * igw create igw ve VPC'ye attach et

Subnets → 2 Public, 2 Private

1A Public

90.90.10.0

1B Public

90.90.20.0

1A Private

90.90.11.0

1B Private

90.90.21.0

✓ * Public Subnetleri: IPV4 Enable

✓ * ~~VPC'ye igw oluştur~~

✓ * Route table → Public Route Table → igw ekle (local route)
Private Route Table oluştur → Private'leri associate et,

✓ * Subnetleri Route Table'a associate et.

✓ Endpoint → Create - AWS services - S3 getung - VPC sec
Route Table sec (Private route sec) → Policy full access (create)
(Endpoint'ın private Route table otomatik ekledi)

VPC sec ↓

✓ Security Group ① ELB Sec Grp

inbound	source	outbound	source
HTTP 80	anywhere	All http	0.0.0.0/0
HTTPS 443			

② EC2 Sec Grp

inbound	source	outbound	source
ssh 22	0.0.0.0/0	All http	0.0.0.0/0
http 80 → ELB sec			
https 443 → ELB sec			

③ RDS Sec Grp

inbound	source	outbound	source
MySQL/Aurora 3306 → EC2 sec		All http	0.0.0.0/0

④ Nat instance Sec Grp

inbound	source	outbound	source
HTTP 80	anywhere	All http	0.0.0.0/0
HTTPS 443			
SSH 22			

②

RDS

* Once subnet group terminierung (Jede once top default VPC de gruppen)
Create → VPC sec → AZ sec (1a, 1b) → Subnet 1a sec, Private 1a sec
* Create Database

MySQL → source sec 8.0.28

Free tier → DB 1sm, 1y, storage 20, ASB 40

VPC sec, Subnet group

Public Access NO → Confer secure design 1st or work beenden ulassen 12.

Sec group - RDS sec grg

Port 3306

- Database Name → Klausur

Backup abstammen

Maintenance abstammen, Gatignation.

(Enable deletion protection sec me X)

(Create)

S3

1) Create Bucket → vlog (name), N. Virginia

→ Object Ownership → ACLs enabled / Bucket Owner preferred

Public access ☐ Acknowledge, versioning disable

(Create)

2) Create Bucke → copete it be (name), Albany

ACLs enabled / Bucket Owner

Public Access ☐ (Create)

objekte lesen

Properties → Static website Edit, enable, index.html, save

Obj g ile (index.html + resim)

* permissions obje g lesen

→ Choose from predefined ACLs

Grant public-read access

☒ I understand

(upload)

Yoda zuerst

bedeuten

Public

Policy

Static

web

site

holder

g

zuerst

mit

site

using

Yoda

edit

bucket

holder

③

Dasgezi Github'a pushuna hazirlanar?

*

Userdata Mad page ✓

(her satırı repo ismini gancele)

#!/bin/bash

apt-get update -y

6 TOKEN="token_yaz"

7 git clone https://\$Token@github.com:/reposw/my-awesome-project.git

8 cd /home/ubuntu/my-ubuntu-aws-project

12 cd /home/ubuntu/my-aws-apt-project/src

*

→ src/cblog/settings

✓ 88 → database name

✓ 89 → user → admin

✓ 90 → Password → → (.env → 2. satır Password = Claustr1234)

✓ 91 → Database endpoint

✓ 146 → storage bucket name (Vlog), regi

✓ 148 → Region name - us-east-1

* Github'a push ediyoruz *

{ userdata da token'i xxx gibi push yap
{ token i userdata da kullanacağız }

Create instance
2018.03.01

NAT yazıktan AMI yi seç

Network → VPC, subnet Public IA

(Auto assign IPV4 ile size enable yapdik)

Tag → Name AWS - capstone - Nat - instance

Sec Grp - Nat Sec Grp (80,22,443 hejurdur)

(create)

↳ Nat instance olusturduktan sonra

Source / Destination check ☐ kaldirmali (stop)

Gateway / subnet tabisi oldugu yeri kaldirmali...

→ Private Route table a gel

bu ec2 ke disari cikmesi icin

Nat instance route table a ekliyoruz

Add route → destination Anywhere Target → Nat instance

Nat instance

2. unke 50.14

(Nat instance'i bastiran host gibi kullanmaliyiz, EC2 tan saglanmisk gorkuse disaridan)

(4)

Launch Template

* Once Role oluşturulmuş Launch Template için
EC2 ile S3 e data atabilirim diye

Create role → AWS source → EC2 → ^{S3} AmazonS3FullAccess → ^{İsim} AmazonEC2ReadOnlyAccess

* Create Launch Template

- Name → Django-project-Template
- ☒ ASG guidance to help me (security)
- AMI seç Ubuntu mi 18.04 amd64-secure-0723e439-56769d8ab
- Subnet datacenter
- Sec Group → django-EC2-secgrp

Advantage detail

- Role seç
- user data etile

[user data
django ile ilgili python code kurulum islemleri]

[create launch Template]

Target

- Instance
- isim → django-Target
- VPC django
- Health checks → 5/2/5/20

— Instance atama → (ASG atama sonra, Auto Scaling oluşturulmuş)

[Create Target Group]

ELB

Create - LB → Application → Internet facing

VPC → djangoVPC

* Mapping → 2 AZ de de Public'leri seçiyorsanız subnet olarak!

sec Grp → ALB django sec grp (22-80-443)

* Listeners → HTTP 80 → Forward → django Target

+ Add listener HTTPS 443 → Forward → django Target

* SSL etile (Create)

* oluşturulan sonra → ELB sec → Add listener
HTTP 80 sec

Redirect → 443
to
URL

same changes



ASG

name

Launch Template

vPC sec

A2 sec → Web servers Private to internet isyın.
Private 1A, Private 1B

Load balancer → Sec

Target Group → Sec

Health Check → ELB 300

Desired 2-2-4

Policy → Target Policy → %70

(Cluster nokta için ekleyebilirsiniz) (proje de pass geçtik)

Tag eklenir bilis (Launch Template da eklediyseniz gerek yok)

Not instance ile local e bağlanıyor deniyoruz
önce not instance'a ssh ile bağlan.

```
eval $(ssh-agent)
```

```
ssh-add /usr/.ssh/id_rsa
```

```
ssh -A ec2-user@notEC2IP
```

```
ssh ubuntu@privateIP
```

```
- websayfayı → is de projeye gir
```

CloudFront

Create Distribution

→ ELB copstone load balancer (Path yok / name yok)

Additional → Protocol (Matchless)

→ Default Cache Policy (Default) *compress yes!

When HTTP → HTTPS

Allowed HTTP Method → ALL

Options ☒

Cache Policy on Origin request

Legacy cache settings

Header → include polling header

Additional → Accept

Cloudfront front to Proto

Query All

Cookie All

Tom edge location

→ Alternate domain name | Select certificate

→ sitem nelerini cache yapar

ALL

6

Role S3 with failover → S3 → ELB → S3

Create Health Check Name / Endpoint / Domain Name
http
cloudfront DomainName

(create)

Create Record Failover Policy

name → www, discipline (Route 53)

endpoint → cloudfront

primary → healthcheck

secondary → S3

DynamoDB

Lambda → ^{create role} ① In role ^{create role} ~~cloudfront~~ role → Lambda to S3 and DynamoDB

② Create permission → Name, Python, select role

Advance Selty → VPC dmp → select all subnets, select secgrp

Trigger → To create object

Trigger for lambda

To create object