Project-004-Phonebook-Aplication

Cloud formation ile bir stack oluşturacağız ve bu stack de telefon rehberi uygulaması olacak. Stack oluştuğunda karşımıza 9 resource çıkacak. Bu süreçte kullanacağımız aws araçları **ALB** **(Application Load Balancer), ASG(Auto Scaling groups)** [birden fazla instance olacağı ve ilave instancelar devreye gireceği için ASG ALB gerekli] ve **MyDatabase**.

ALB, yükü alır, listener aracılığı ile trafiği dinler ve **target gruba** dağıtır.

Target grup olarak, instance ları işaret edeceğiz bir diğer değişle Target gruba, Auto Scaling grubun oluşturacağı instance ları tanımlayacağız. Instance lar ASG tarafından oluşturulacak. ASG, instance ları **Launch Template** de izin vereceğimiz yönergelere göre oluşturur (bu yönergeler keypair, userdata, Instance tipi, A.Zone belirleyeceğiz).

ALB oluşturacağız ve ALB nin listener bölümünü oluşturacağız bir diğer değişle neyi dinleyecek ve dinlediği trafiğide nereye aktaracak onu tanımlayacağız (80 portunu dinlemesini isteyeceğiz dinlediği trafiğide target gruba iletmesini isteyeceğiz). ASG ye bazı talimatlar vereceğiz desired, min. ve max. kapasite bu sayılara göre instanceları ayağa kaldıracak. Örneğin desired kapasite 2 instance olacak ve ASG iki instance ayağa kaldıracak ve onları Load balancerın target grubuna aktaracak. Target gruptan **healthy** aldığımız zaman uygulamamızın çalıştığını göreceğiz.

Bilgilerin tutulduğu bir DataBase olacak. DataBase i, cloud formatiana atacağımız yaml dosyasında yazacağız. DataBase Security grubu, özel oluşturacağımız RDS sec grubunu kullanacak. Bu uygulama için 3 security grup oluşturacağız ec2 lar, load balancer ve RDS DataBase için olacak. 3 tane oluşturma sebebi template açıklanacak...Stack oluştuğunda Outputs çıktı kısmında load balancerın adresi olacak (sitenin web adresi oluşacak). Birden fazla instance ile uygulama ayağa kaldırdığımızda instance ın public dns name ile uygulamaya **erişemeyiz**, load balancerın public dns name ile **ulaşabiliriz**.



**-GİTHUB-**

1-) Project-004-Phonebook-Aplication adında Readme seçili Private bir repo oluşturalım.

2-) Code dan http klonlayalım

**-LOCAL-**

1-) Terminale gelelim masaüstü MyProjects klasörümüzün içine gelelim ve “ git clone http...” klonlayalım.

2-) CLW 13 -> aws -> Project-> Project-004-Phonebook-Aplication a geliyorum ve kopyalayıp Myprojects içindeki phone adında ki klonladığım repounun (klasörün) içine yapıştırıyorum.

3-) Klonladığım reponun içinde ki Project-004-Phonebook-Aplication u VSC ile açıyorum.

4-) Phonebook-app.py geliyorum burada bazı değişiklikler yapıyorum (devlprların uygulama için yazdığı dosya).

-vsc de 15.satırdaki database ‘in şifresini düzenleyelim ‘abcd1234’ 8 karakter olmalı

-Vsc de Ctrl f ile serdar ismini kendi isminiz yapalım (replace ile)

-VSC de ctrl s yapıp kaydedelim ve Githuba gönderelim (git add . git commit –m”.....” git push)

-Phonebook-app.py de 9. Satırda db\_endpoint adında bir değişken tanımlamış devlprlar. Bu komut şu anlama gelmektedir: Read (r) olduğunda bir dosya oluştur (dbserver.entpoint) ve aç (open) içindeki değeri oku ve db\_endpointe ata demek.. burada dikkat edilmesi gereken konu **user data** oluştururken bu bölüme dikkat etmeli (**dbserver.entpoint** e).

5-) cfn-template.yaml (phonebook-cfn.yaml) da bazı değişiklikler yapacağız (bu tamplate yazabiliyor olmamız gerek. Neyin ne olduğunu neden yaptığımızı biliyor olmalıyız).

-VSC de 49. satırda ki (satırlar değişkenlik gösterebilir) ami adresini güncelleyelim ami-0b5eea76982371e91

-Vsc de userdata kısmında 77. satır token için token alalım varsa 77. Satıra yapıştıralım (token nasıl alırım ? profilden settings, devoloper settings, personal Access token, token clasic, generate new token, isim ve süre verelim, hepsni seçelim tokenı oluşturalım)

-78.satır için, githubda Phonebook-app.py gelelim, raw adresi için raw a basmalıyız, http linkinde Application a kadar kopyalayalım ve 78. Satıra yapıştıralım $TOKEN@ ekleyelim (https://**$TOKEN@**raw.githubusercontent.com/Rose94t/private-my-projects/main/Project-004-Phonebook-Application")

-142. satır 20 den sonrasını kabul ediyor

-147. Satırda kendimize göre bir isim girdik Havva-db

-154. Satırda Phonebook-app.py de şifreyi ne belirledi isek aynısını yazmalıyız abcd1234

-VSC de ctrl s dedikten sonra istersek kaydedip githuba gönderebiliriz...

Phonebook-cfn.yaml dosyasını nasıl dizayn ettik ?

-Cfn-litle seçtik

AWSTemplateFormatVersion: 2010-09-09

-***Description*** kısmında, template in telefon uygulaması için olduğuna değinilmiştir ve aws uyguamalarının konfigürasyonun dan bahsedilmiştir.

***-Parameters*** kısmında,

-*MyVPC parametresinde*, stack oluşurken kutucuk içinde vpc ler çıkacak ve stack o vpc lere göre ayağa kalkacak.

*-KeyName parametresinde*, N.virginia daki key name seçeceğiz.

*-Subnets parametresinde,* default ya da vpc ne ise ona göre subnetleri seçebiliriz. Public ve private 6 subneti de seçiyoruz. Subnetler liste şeklinde karşımıza çıkacak.

***-Resources,*** kısmında 9 tane Resources ımız var demiştik.

- Öcelikle **ALB için sec grup oluşturmalıyız**. Uygulamada dış dünya ile irtibat kuran ilk yer ALB dir. ALB bunu listener aracılığı ile yapar. Ve dinlediği trafik uygun ise mimarinin içine gönderecektir. Gelelim ***ALB için sec. grubu***nu oluşturmaya ve talimatları vermeye... ALB için http portunu enable hale getiriyoruz. Inbound role tanımlıyoruz. 80 portuna her yerden izin veriyoruz. vpc olarak da referens veriyoruz **MyVPC** ye Ref ile (Ref dosyadaki resource a referans verirken kullanılır). Outbound otomatik geliyor ayrıca tanımla**ma**dık.

- ***EC2 lar için sec grubu oluşturacağız***. Flask web serverı için ssh ve http enable ediyoruz. 22 portuna her yerden izin veriyoruz. 80 portunda her yerden izin ver**mi**yoruz! Peki ne yapıyoruz bir tane sec grubu kaynak(source) olarak gösteriyorum. **Getatt** ile ALB nin alt özelliğini çekiyorum ALB nin sec grup ıd sini çekiyorum (Getatt bir resources ın alt özelliğini çekerken kullanılır) özetle 80 portu ALB den gelen trafiğe izin verecek. EC2 lar ancak bu şekilde çalışacak (satır 51 oluyor herkesde farklılık gösterebilir). Ref ile MyVPC referans gösteriyor ve seçiyoruz.

-***LaunchTemplate*** ne işe yarıyordu? Ayağa kalkacak instancelara emir ve yönergeleri vermektedir. ASG kafasına göre kaldırmayacak ınstance ları yönergelere göre kaldıracak.

Güncel Ami , t2.micro, Ref ile keyname çekiyorum, GetAtt ile sec. grubun alt özelliğini seçiyorum. Tag liyorum oluşan sitenin adını alacak . Geldik *userdata* bölümüne , işletim sistemi güncelliyoruz.Token tanımlıyoruz, Folder değişkenini tanımlıyoruz bunu github daki Phonebook-app.py dosyamıza gelerek raw basarak yapıyoruz (78.satır). Sonraki satırlar folder değişkenini ne olarak tanımladıysak folder ın içeriğini curl komutu ile çekip komutları çalıştıracak. Ardından python3 komutu ile phonebook-app.py çalıştıracak (83. satır). MyDBURI database imizin URI demek GetAtt ile entpoint.addresini çekecek (84.satırda) ve bu değeri 76. Satırda bulunan komut ile sıfırdan bir dosya oluşturup yazdıracak (ec2-user altında bulunan dbserver.entpoint e). Hatırlıyorsak dvlprlar –app.py dosyası 9.satırda bunu tanımlamışlardı.

***-TargetGrup*** oluşturalım, port 80 olup instance tipinde olmalı, bir instance dan 3 kere unhealthy alırsak onu unhealthy ilan edecek eğerki 2 kere healthy alırsa healthy ilan edecek. Vpcıd Ref ile çekiyorum.

-***ApplicationLoadBalancer*** oluşturalım, ipv4 ve internet-facing seçiyoruz. Sec grup olarak GetAtt ile ALB sec grubunu seçiyorum. 80 portundan dış dünyayı dinle diyorum. Diğer önemli bir konu subnetler ref ile çekiyorum. Tiplerini seçerken http HTTPS lerde Application tipi seçilir diğerlerinide bilmek önemli tipler sertifika sınavlarında çıkar.

***-ALBListener*** oluşturalım, en önemli kısmı default olarak hangi actionu yapacak (80 portundan dinliyor). Tipi forward yani ilet diyorum ve target gruba iletiyor. Listener hangi ALB ait ise Ref ile çekiyorum. Port 80 dinliyor yönergeler bu şekilde.

***-ASG*** oluşturalım, ALB de subnetler vardı ASG de ise AZ oluşturacağız. 120. Satırın anlamı tüm AZ seç demek. İstersen parametre kısmında AZ leri tek tek belirtip burada Ref ile çekebilirsin. Benim istediğim instance Kapasite 2 , min 1 , max 3 bu değerleri değiştirebilirsin ihtiyaç ve maliyete göre. Healtcheck grace period 300 sn bekleme süresi. Healtcheck tipi ELB. Kafasına göre instance kaldırmayacağı için yukarıda oluşturduğum LaunchTemplati referans veriyorum. LaunchTemplate lerde version vardı değişikliklerde version1 version2 oluyordu ve burada 126. Satırda en son ki versiyonunu GetAtt ile çek diyorum. En son olarak 130. Satırda ASG oluşturduğu instance ları TargetGRUBA gönderiyor (Ref ile çekiyoruz targetı).

-***DataBase sec grubunu*** oluşturalım, tipi RDS in dbsec grubu olacak***.*** Database ile iletişim kurmasını istediğim Ec2 secgrup ıd kullanacağım GetAtt ile çekiyorum. Burada aslında bize denilen şey şu, DataBase diyorum ki bu secgruba sahip ec2 lardan sana bir istek geldiğinde kabul et. 137. Satırda back–end database, arka planda çalışması adına.

***-MyDatabase server*** ı oluşturalım, 142. Satırda 20Gb a izin veriyoruz. **145. Satır sınav sorusu** backup almasını istemediğimiz cloudformationun BackupRetetionPeriod değerine sıfır (0) yazarsak backup almaz. 147. Satırda kendimize göre bir isim verebiliriz bir Identifier giriyoruz. Engine MySQL olacak şekilde ayarlayalım. 154. Satırda Phonebook-app.py Şifreyi ne belirledi isek aynısını girelim abcd1234. MySQL 3306 portundan çalışıyor. 156. Satıra true dedik **Workbench** den ulaşabilmek adına.

**-Outputs kısmı,** site olarak ALB nin dns name kullanacağız GetAtt ile çekiyor bunu.

CTRL S ie kayıt edelim.

**-Aws cloud formation-**

**1-) N.Virginia cloud formation** gelelim, Stacks, creat stack, upload ve phonebook-cfn.yaml seçelim, next diyelim. Stack name girelim. Key name seçelim, default VPC seçelim, 6 tane default subneti seçiyorum ve stack i oluşturalım. Outputa gelip web adresini kopyalayalım...

2-)Browsera yapıştıralım, Browserda linkin sonuna /add ekleyip isimleri ve numaraları kaydedelim ardından /add silip istediğimiz kişilerin ismini girebiliriz.. /delete ile silme işlemini /update ile numaraları güncelleyebilirim.

STACKLERİ SİLMEYİ UNUTMAYALIM

 