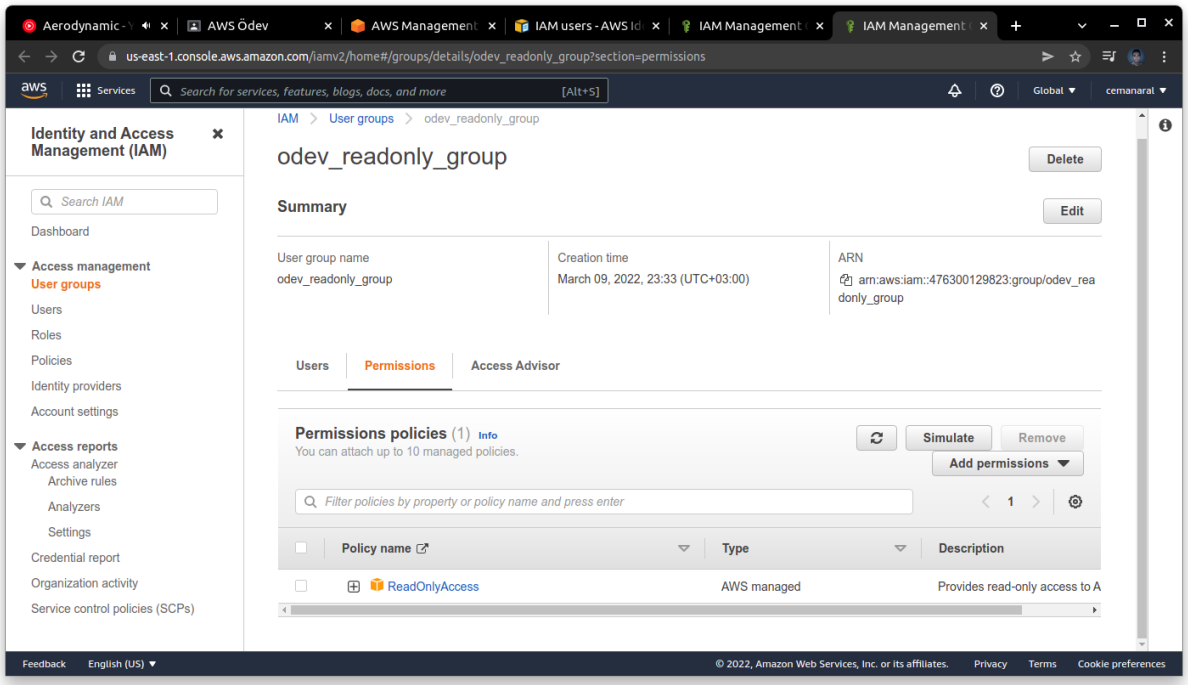
Cem Anaral
+90 553 346 90 42
cemanaral425@gmail.com

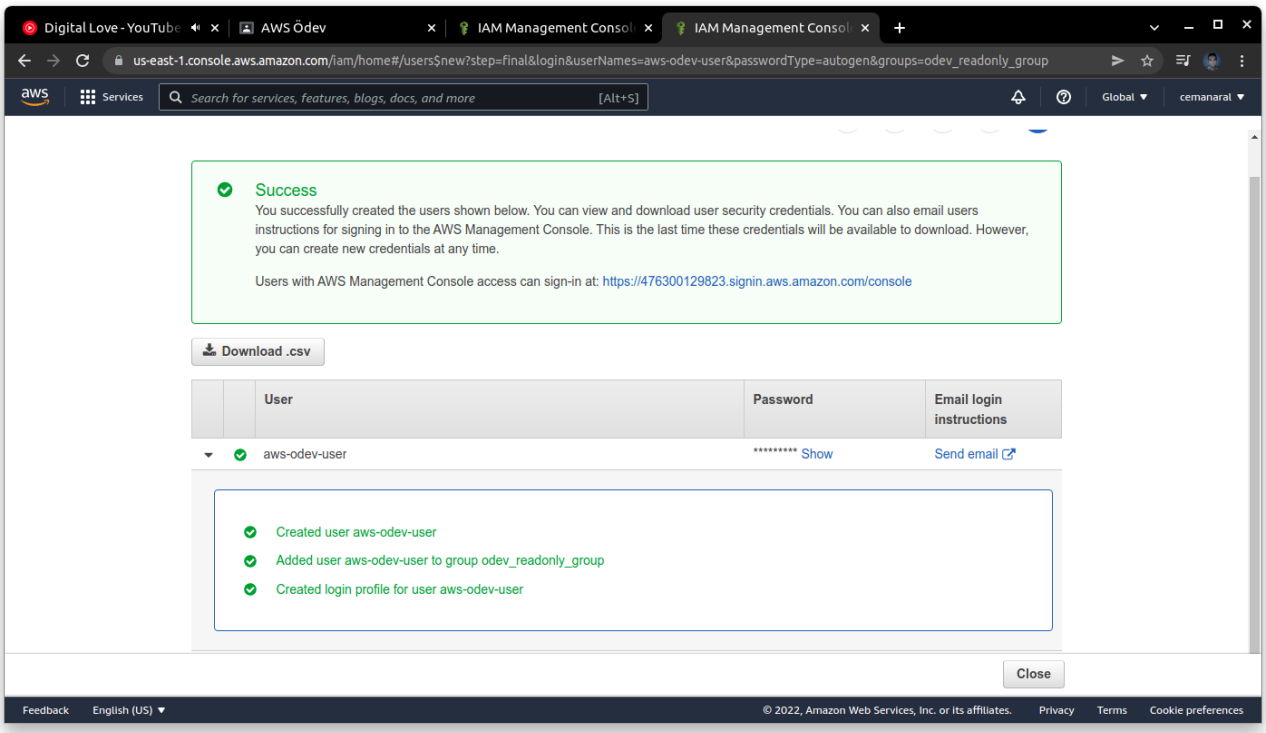
IAM User Credentials
region: us-east-1 (N. Virginia)
link: https://476300129823.signin.aws.amazon.com/console
username: aws-odev-user
password: {MQ{34VrZqSQzl-

IAM User Oluşturma

odev_readonly_group adında yeni bir grup oluşturuldu. Policy olarak sadece ReadOnlyAccess policy atandı.



Oluşturulan bu gruba bir user oluşturulup eklendi. Her yeni loginde yeni password zorunluluğunu kaldırdım.



VPC Kurulumu

RDS ya da EC2 kaldırmadan önce VPC kurulumu yapmak daha mantıklı geldi. Böylelikle güvenlik için önlemi de en baştan almaya çalışacağım.

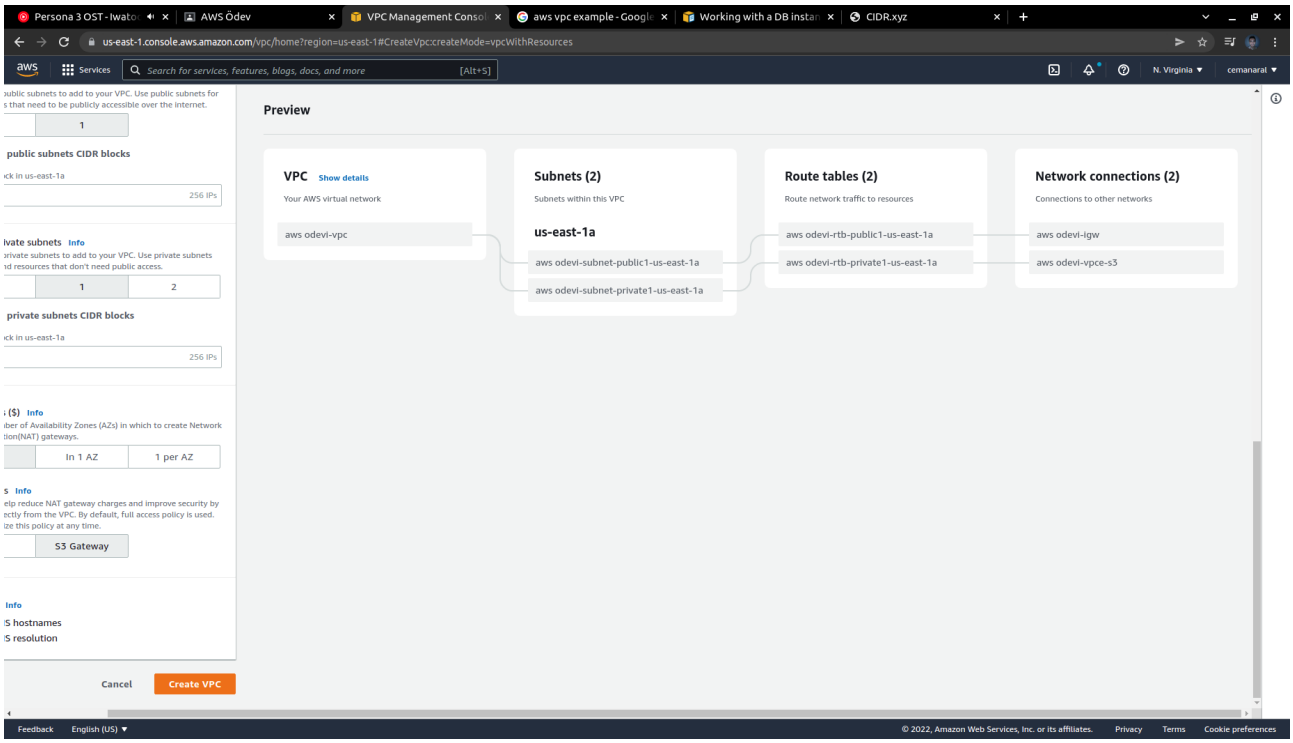
RDS ve EC2yu iki farkli subnete ayıracagim. RDS'in oldugu subneti private network, EC2nun oldugu subneti ise public network olarak ayarlayacağım.

Tek bir availability zone seçtim. Çünkü iki tane seçince nedense her bir zone için bir public bir de private subnet ayarlamam gerekiyordu. Yani toplamda iki tane private iki tane de public subnet seçtirmeye zorluyordu (ya da 0 tane seçebiliyordum).

Aslında amacım private subneti farklı bir availability zone, public subneti farklı bir availability zone'a koymaktı (Daha güvenli olsun diye. Ne kadar uzak olurlarsa o kadar iyi olur diye düşünmüştüm). Sanırım bu mantık yanlıştı. Tek bir availability zone seçip aşağıdaki şekilde ayarladım.

Vpc name: aws odevi
IPv4 CIDR : 10.0.0.0/16
Public subnet CIDR: 10.0.1.0/24
Private subnet CIDR: 10.0.2.0/24

NAT gateway'in yanında $ işareti olduğu için ve private subnetlerle alakalı oldugunu düşündüğümden dolayı seçmedim (aslında para öderim diye korktum). İleride ihtiyacım olursa diye buraya not ediyorum.



RDS Servis Kurulumu

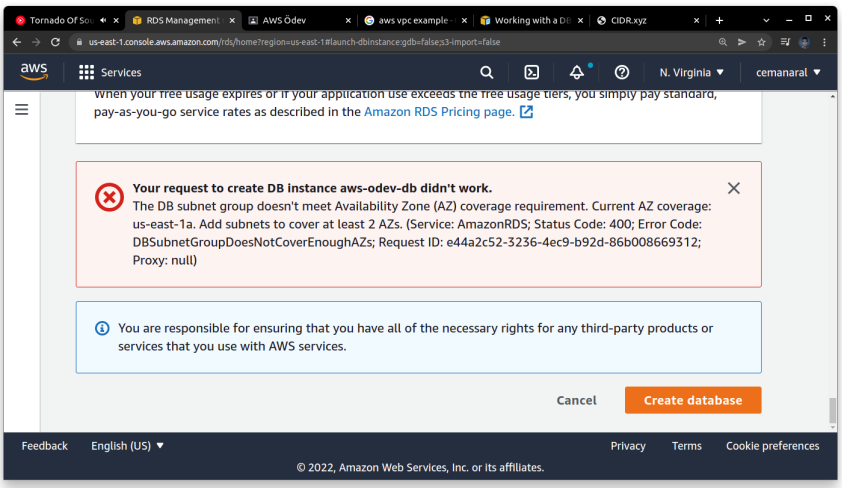
Create database tuşuna basıp başlıyorum. Template kısmında free tier seçeneğini görünce dayanamayıp bastım. Ne anlama geldiğini bilmiyorum.

Database username default olan “admin” olarak bıraktım. Auto generate a password seçeneğini tikledim. db.t2.micro instance’ı seçili geldi. Muhtemelen free tier template’ini seçtiğim için.

VPC tarafında daha önce oluşturduğum aws odevi-vpc yi seçtim. Subnet group tarafında nedense önceden ayarladığım private subneti seçtirtmiyor, ille de yeni bir db subnet group yaratırım diyor. Kendi haline bıraktım.

Public access kapalı. VPC security group yaratmamışım, default olarak bırakıyorum. Belki ileride VPC security group yaratabilirim.

İlk hatamla karşılaştım.



Önceden yarattığım VPC’de iki tane availability zone olmadığı için RDSi kurmama izin vermedi. Sanırım iki tane zone istemesinin sebebi database’imizin bir yedeğinin de olmasını istemesinden kaynaklı.

Cancel’a basıp VPC tarafına geri dönüyorum. Eski VPC’yi sildim. “aws odevi yeni-vpc”yi oluşturmaya başlıyorum. Bu sefer default olarak 2 gelen availability zone’u ellemiyorum.

Bu sefer default olarak gelen CIDR bloklarını da ellemiyorum. cidr.xyz sitesinden anlayabildiğim kadarıyla bir çakışma/problem yok gibiydi. Aşağıdaki gibi yeni VPC ve subnetlerimi oluşturdum

Ipv4 cidr 10.0.0.0/16

Public subnetler:
Subnet CIDR block in us-east-1a
10.0.0.0/20

Subnet CIDR block in us-east-1b
10.0.16.0/20

Private subnetler

Subnet CIDR block in us-east-1a
10.0.128.0/20

Subnet CIDR block in us-east-1b
10.0.144.0/20

RDS tarafına geri dönüyorum. Önceki ayarların aynısını yaptım. Bir sorun yaşamadan aws-odev-db’yi yarattım.

RDS passwordunu not etmeyi unutmuşum. “aws-odev-db” olarak resetliyorum. Password resetini iki gün beklememek için apply immediately’yi işaretliyorum.

EC2 ve Wordpress

Ubuntu Server 20.04 LTS AMI seçiyorum. VPC olarak oluşturduğum VPCyi seçiyorum. Subneti us-east-1a’daki public olarak ayarlıyorum.

Wordpressi docker ile kurmaya karar verdim. Böylelikle ileride swarm yapmak istersem daha kolay olabilir diye düşündüm. Tabii öncesinde docker kurmam gerekiyor.

Aşağıdaki scripti user data olarak veriyorum (en son hali yer almakta):

```
#!/bin/bash
curl -fsSL https://get.docker.com | sh
sudo usermod -aG docker $USER
sudo chmod 777 /var/run/docker.sock
docker pull wordpress
docker run -p 8080:80 --name odev-wordpressi -d -e WORDPRESS_DB_HOST=aws-odev-db.c7kz1oqay3qw.us-east-1.rds.amazonaws.com:3306 -e WORDPRESS_DB_USER=admin -e WORDPRESS_DB_PASSWORD=aws-odev-db -e WORDPRESS_DEBUG=1 -e WORDPRESS_DB_NAME=wordpress_db wordpress
echo "oldu" > /tmp/oldu_mu.txt
```

EC2 instance'ımın public ipsinin olmadığını fark ediyorum. Instance'ı silip tekrar oluştururken Auto-assign Public IP seçeneğini etkinleştiriyorum.

Maalesef instance'a bağlandığımda dockerın yüklenmediğini fark ettim.

sudo usermod -aG docker $USER
sudo chmod 777 /var/run/docker.sock

satırlarını user dataya ekleyip manuel olarak da calistigini teyit ediyorum.

Bir sekilde EC2 instance'ım RDS databaseini göremiyordu. Security grouplarla alakalı olacağını düşünüp wordpress odevi security group adında yeni bir security group açtım ve bu EC2 instance'ımı ve RDS databaseimi bu security group'un içine koydum.

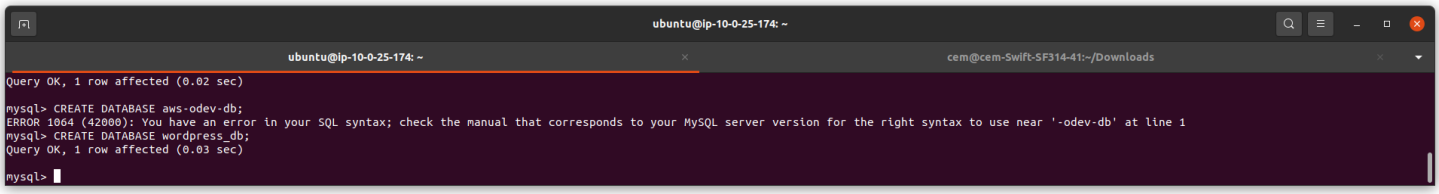
Security groupta bağlantı sorunu yaşamamak için bütün portları ve bütün ipleri inbound ve outbound için açtım. Güvenli olmadığını biliyorum ama daha fazla bağlantı sorunlarıyla uğraşmak istemedim. İleride bu kısmı çözeceğim. Gerçi RDS ayarlarında hala publicly visible olmadığını gösteriyor. Bu duruma tam olarak anlam veremedim. Assign ettiğim security group her şeye izin veriyordu.

Yukarıdaki gibi bağlantı sorunlarını çözdüğümde ipmi curllediğimde şöyle bir hata mesajı beni karşıladı.

```
<p>We were able to connect to the database server (which means your username and password is okay) but not able to select the <code>aws-odev-db</code> database.</p>
<ul>
<li>Are you sure it exists?</li>
<li>Does the user <code>admin</code> have permission to use the <code>aws-odev-db</code> database?</li>
<li>On some systems the name of your database is prefixed with your username, so it would be like <code>username_aws-odev-db</code>. Could that be the problem?</li>
</ul>
```

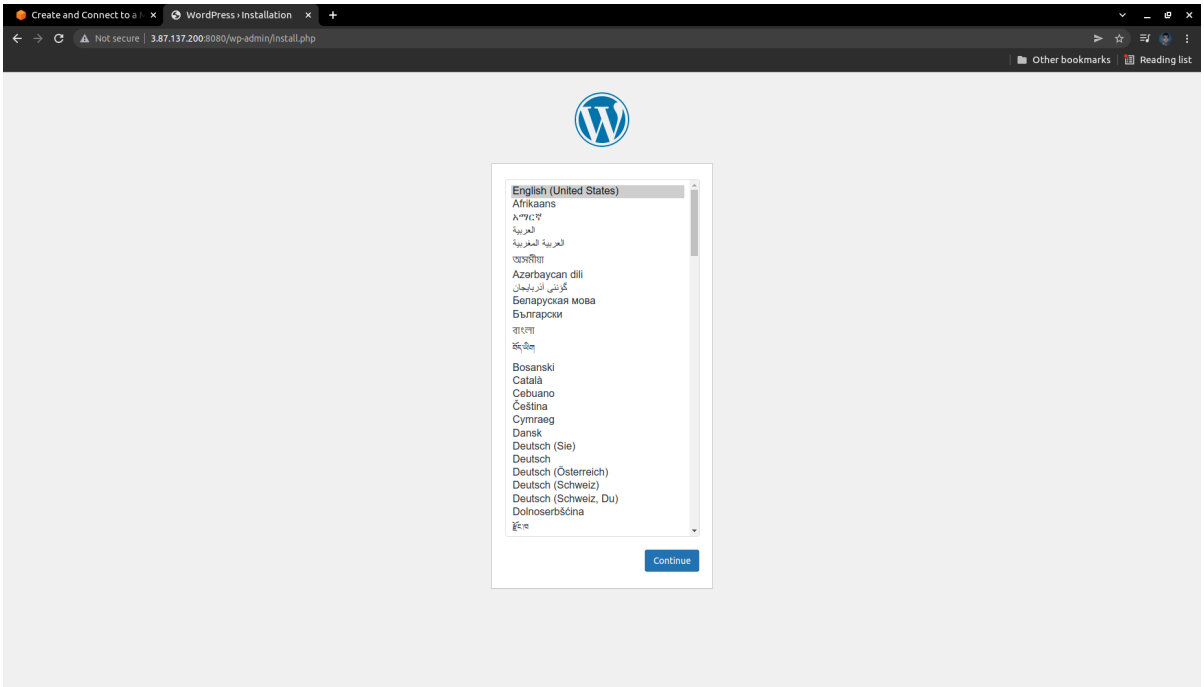
If you don’t know how to set up a database you should contact your host. If all else fails you may find help at the WordPress Support Forums.

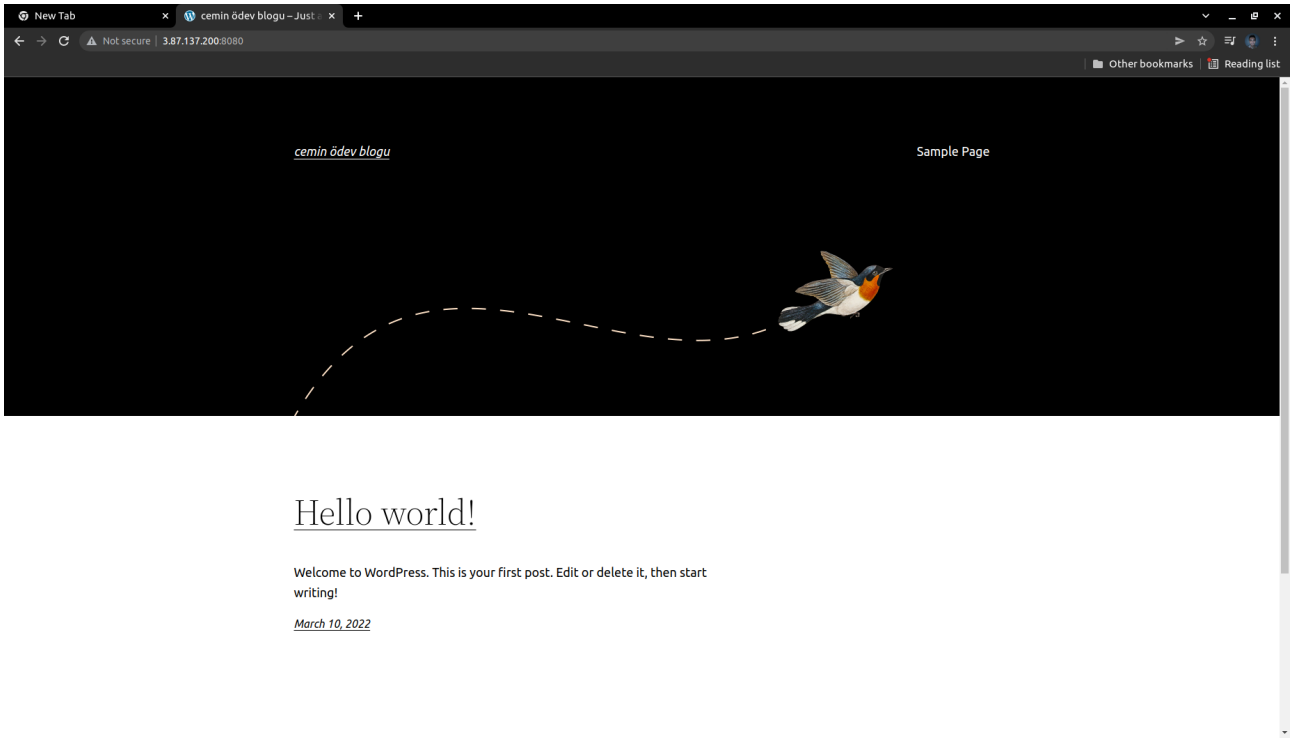
Yukarıdaki HTMLde anladığım kadarıyla aws-odev-db adında bir database'in olmamasına kızıyordu. Docker image'ı run ederken database ismi olarak bu adı vermiştim. Muhtemelen bu database'in daha önceden oluşturulması gerekiyor ya da RDSdeki adının aynısı olmalı.



Hazır EC2 makinem RDSi görebiliyorken aws-odev-db databaseini oluşturmaya koyuldum. Mysql clientini yukleyip baglandigimda CREATE DATABASE aws-odev-db; yazınca syntax hatası aldım. Meğersem mysql database isimlerinde '-' karakterini kabul etmiyormuş. Database adını wordpress_db olarak değiştirmeye karar verdim. (Yukarıdaki user dataya da bu değişikliği ekliyorum. Yani en son çalışır user data olacak dökümanda)

Sonunda çalıştı.





Şimdiye kadar yazdığım user data scriptini ssh ile servera bağlanıp çalıştırıyordum. Nedense user data olarak verince kesinlikle çalışmıyor (en sondaki tmp klasorune yazdırma işlemi de dahil). Ubuntu ve Amazon Linux 2 AMI'larını denedim. Neden olmadığına dair hiçbir fikrim yok.

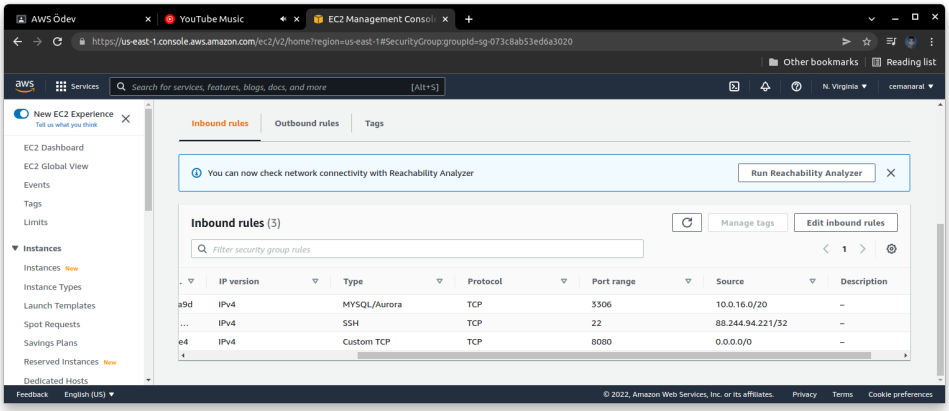
Sorunu çözdüm. User data scriptinin başına #!/bin/bash direktifini koymayı unutmuşum. Şu anda çalışıyor. Tek sorun response time'ın aşırı uzun olması.

Güvenlik için aldığım önlemler

Bu ödev için açtığım "wordpress odevi security group"un inbound ayarlarını görüntüdeki gibi değiştirdim.

SSH bağlantısı için olan 22 portunu sadece kendi IP'me açtım.

Database için 3306 portu açık. Ancak sadece EC2 instance'ımın olduğu subnet bağlanabiliyor. (RDS de aynı security grubunun içinde)



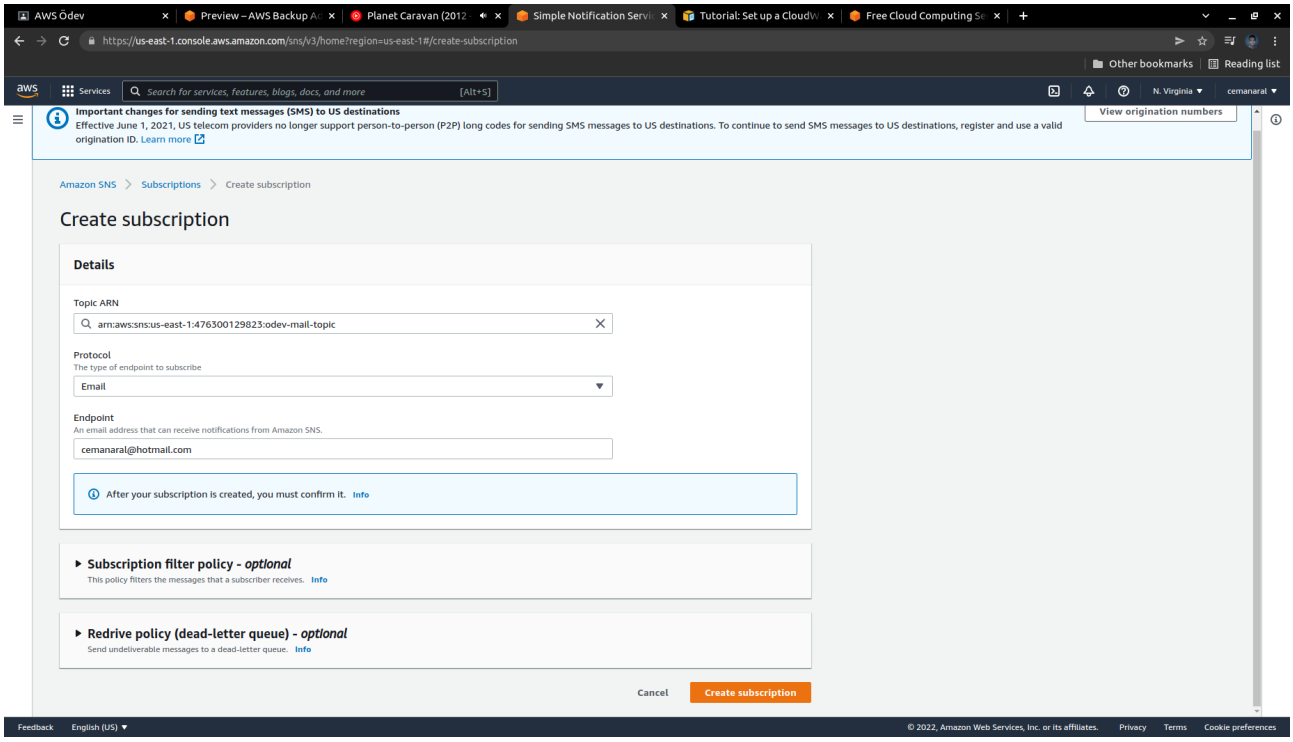
Wordpressi 8080 portundan yayına aldığım için 8080 portu herkese açık.

Outboundda bütün portlar ve ip adresleri açık.

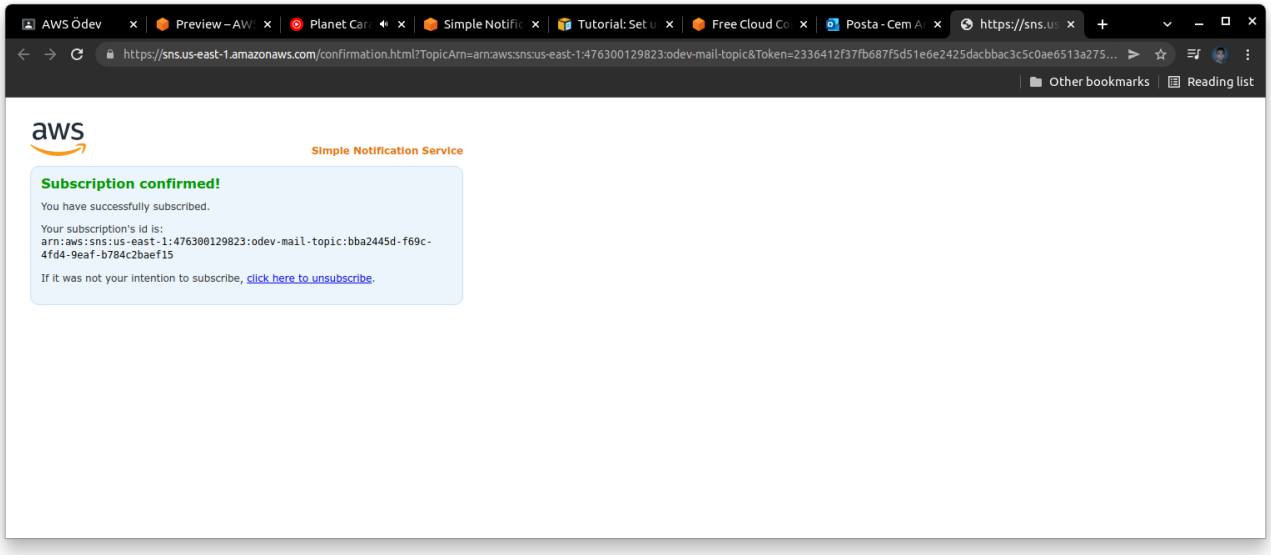
Ayrıca RDSin “Public accessibility” ayarı kapalı.

CloudWatch kurulumu

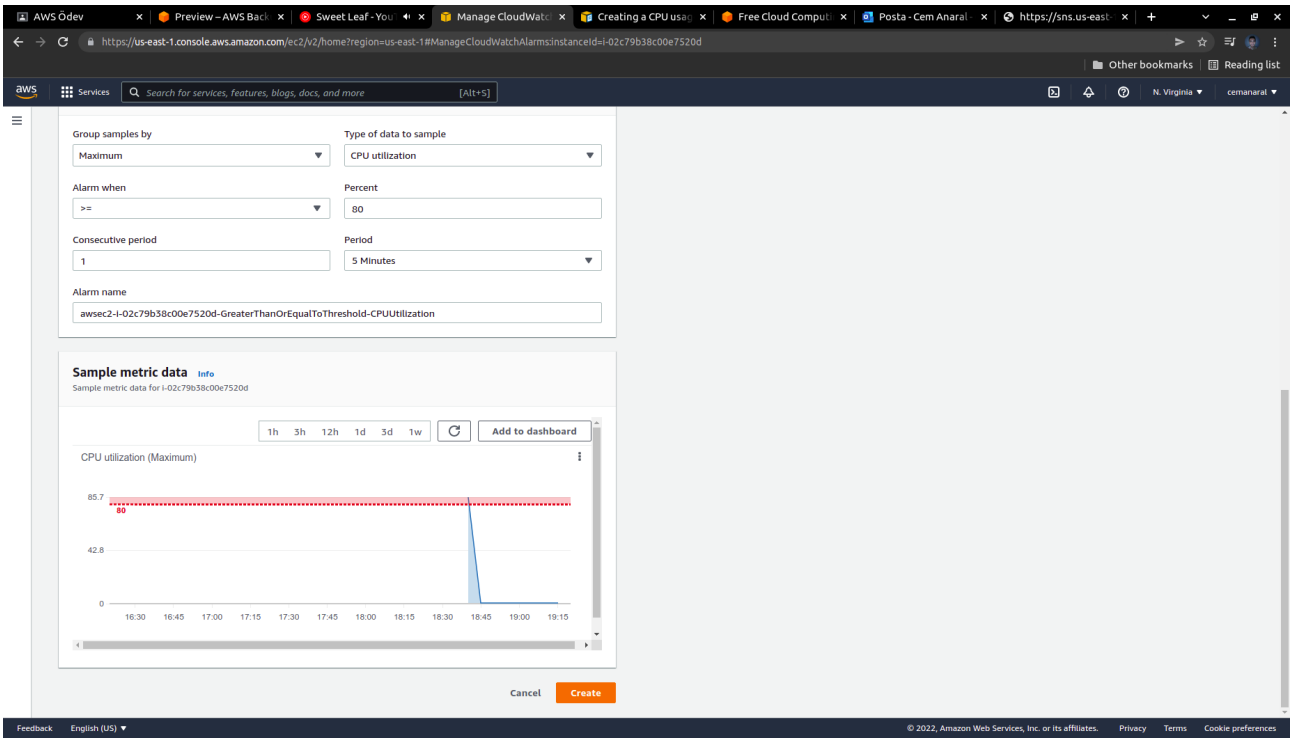
Mail yollamak için öncelikle SNS tarafına bakmaya başlıyorum. odev-mail-topic isimli bir SNS topic oluşturuyorum. Daha sonra bu topici kullanarak bir subscription oluşturuyorum. Mail adresi olarak cemanaral@hotmail.com’ı seçtim.



Mailime gelen subscription linkine tıklayıp onayladım.



Daha sonra EC2 dashboard kısmından alarm ekleme sekmesine tıkladım. Topic olarak odev-mail-topic seçtim. Maximum cpu utilization 80e eşit ya da daha fazla olduğunda mail atacak şekilde ayarladım.



Mail gelecek mi merak ettiğimden ec2 instance'ıma bağlanıp CPU utilization'ı artırmaya çalıştım.

stress paketini EC2 instance’ıma kurdum ve stress --cpu 12 --timeout 120s komutuyla C’deki sqrt() fonksiyonunu çağıran 12 tane worker thread oluşturdum. Kısa bir süre sonra mail geldi.