

## Token Oluřturma

- github.com sayfamıza giriyoruz
- 
- Clarusway repo adresini kopyalıyoruz
- git clone Clarusway repo web adresini terminale yazıyoruz ama enter'a basmadan
- Settings developer settings'den token oluřturuyoruz.
- Token bir kere g z k yor o y zden kopyalay p bir yere kaydediyoruz
- B t n checkleri tik koyarak iřaretliyoruz.

## Token Kullanarak Clone Yapma

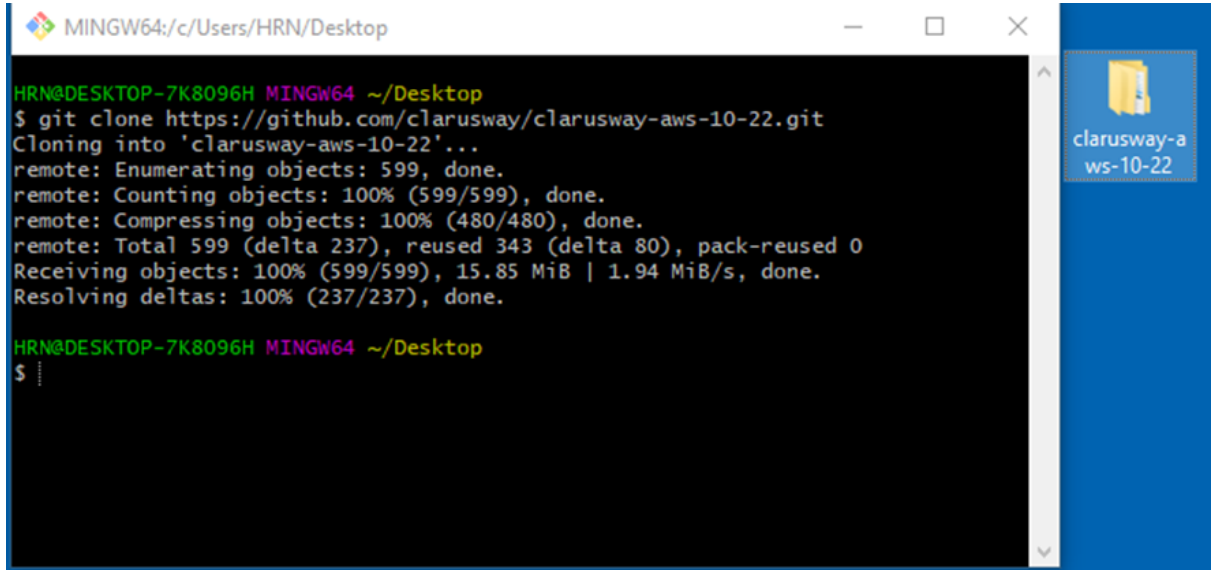
 lk defa repo a arken Github.com  zerinden repo a ılır.

Masa  st nde bulunmayan bir repoyu, pcimize kopyalamak i in;

Kaydetmek istedi imiz dosya konumuna gelinir ve sa  t klayıp “gitbash here” t klanır.

( rnek olarak: Clarusway/Clarusway-10-22 dosyasının masa st nde oluřturulması)

cd desktop (istenilen konuma(desktop a gelinir)

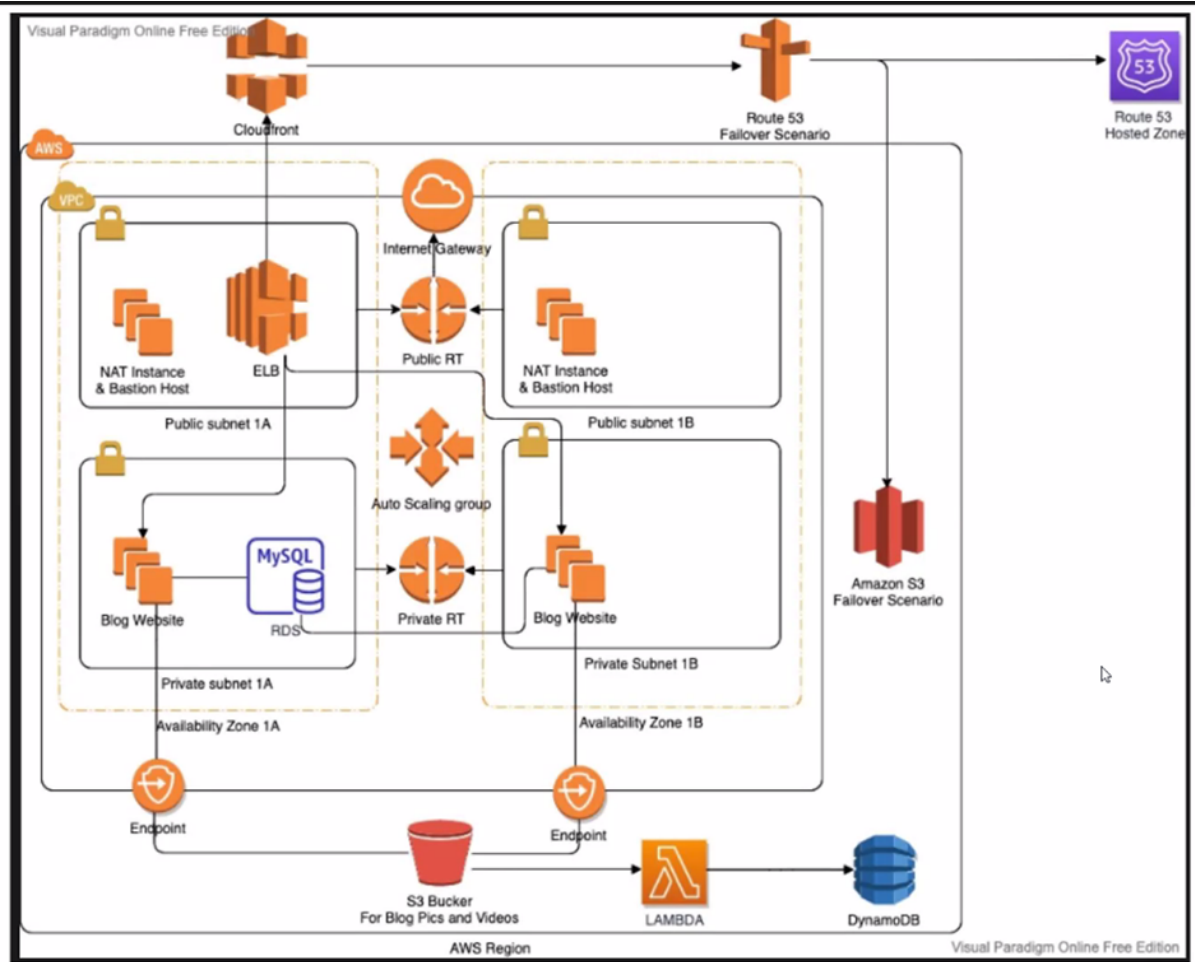


```
MINGW64/c/Users/HRN/Desktop
HRN@DESKTOP-7K8096H MINGW64 ~/Desktop
$ git clone https://github.com/clarusway/clarusway-aws-10-22.git
Cloning into 'clarusway-aws-10-22'...
remote: Enumerating objects: 599, done.
remote: Counting objects: 100% (599/599), done.
remote: Compressing objects: 100% (480/480), done.
remote: Total 599 (delta 237), reused 343 (delta 80), pack-reused 0
Receiving objects: 100% (599/599), 15.85 MiB | 1.94 MiB/s, done.
Resolving deltas: 100% (237/237), done.
HRN@DESKTOP-7K8096H MINGW64 ~/Desktop
$
```

The screenshot shows a terminal window titled "MINGW64/c/Users/HRN/Desktop". The user has executed the command `git clone https://github.com/clarusway/clarusway-aws-10-22.git`. The output shows the progress of cloning the repository, including enumerating objects, counting objects, compressing objects, and receiving objects. The process is completed successfully. On the right side of the terminal window, there is a blue sidebar with a folder icon and the text "clarusway-aws-10-22", indicating that the repository has been cloned to the desktop.

 řlem sonucunda Masa  zerinde bu klas r oluřtu.

## PhoneBook



Yukarıdaki şema Capstone da yapacağımız proje.

Dış dünya ile bağlantı load balancer üzerinden olacak. 80 porttan dış dünyadan gelecekleri listen edecek.

.....

Phonebookapp.py gireyoruz.

Sedar\_1 şifresini değiştirip , developer lardan serdar ismi yerine kendi ismimizi değiştiriyoruz.

Sonra phone .yaml isimli dosya oluşturuyoruz.

.yaml da komutlarla sonra bir boşluğa dikkat etmek gerekir.

LB düşman ile ilk karşılaşan birliktir.

Ec2 için sec group oluşturduk.

LB dan EC2 ya bir trafik var. Talimatla sadece 80 portundan gelene izin ver diyeceğim.

Bir kaynağın alt özelliğini kullanırken !GetAtt kullanılır.

LB ve EC' lar için SEC grupları oluşturduk. 80 portundan herkes görsün istemediğim için LB dan gelenin groupID sini al ve sadece ona izin ver.

Trafiği ayarlayan, dış dünyadan gelenleri bu dinliyor ve 80 portundan alıp target group üzerinden dağıtan yine Load Balancer. KeyPair oluşturulduğu için sadece sizin belirlediğiniz girecek.

-ASG neye göre kaldıracak bunları? O yüzden bir "LUNCH Template" oluşturuyorum.

-User datayı oluşturduk. Token ı dahi ettik. Sonra github dan url yi çektik.

-Burdan sonra Load Balancer ı yazıyoruz.

-Listener dinliyor ve target gruba dağıtıyor.

-Şimdi Auto Scaling Groubu halletmek gerekiyor. HealthcheckGarce Period motorun ısınma süresidir.

LB lerin version numaraları oluyor.

Son versiyonu çekmesi için !GetAtt WebServerLT.LatestVersionNumber yazılır.

AWS ARCHİTECT de önemli. Soru gelir. Life Cycle ile ilgili .

Target Group oluşturuyoruz.

DB sec Grubu oluşturuyoruz.

DB server ve WEB server olarak iki tane oluşturacağız. ??? sayı doğru mu?

Cloud Formation da stack oluşturup, çalıştırıyoruz.

Complete olduktan sonra ;

ASG oluşmuş,

LB DNS Name üzerinden WEB sayfasına ulaşırız. 2 tane instance ayağa kalkmış, DataBase oluşmuş. Onunda end point'ini alıyoruz.