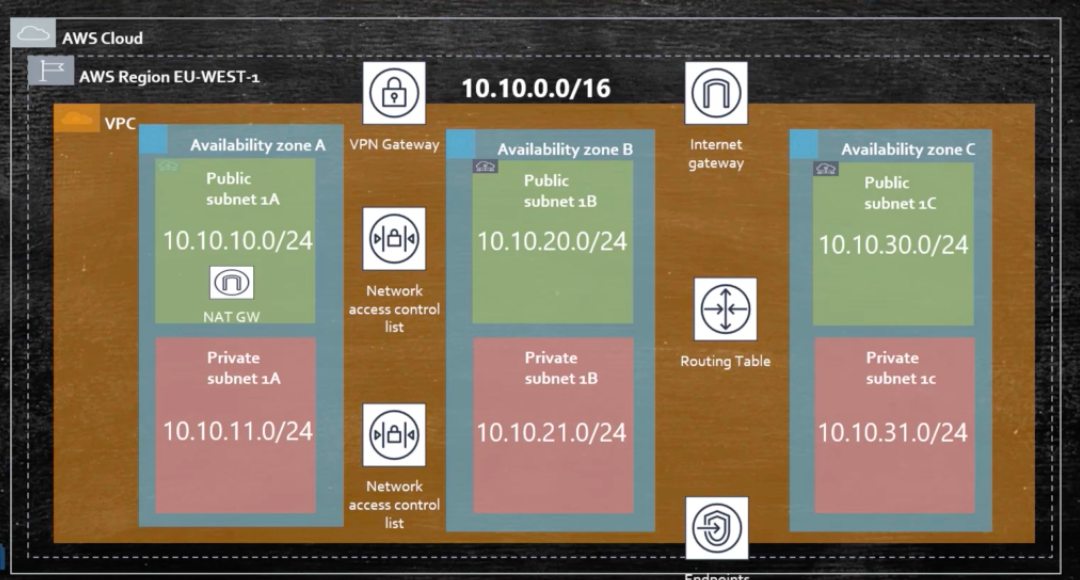
Özgür Scenario

Virtual Private Cloud (VPC)

# PART 1 - Setting Up VPC

# create a vpc with 3 public subnets and 3 private subnets and place them in 3 AZs, each az should include 1 public and 1 private subnet.



VPC mizi 10.10.0.0/6 CIDR ında kuracağız .Bu bize 65000 den fazla IP adresi verecek. AWS VPC atadığımız IP aralığını genişletmemize izin vermiyor, sadece ek ikincil bloklar ekleyebiliyoruz. VPC de 3 adet public 3 adet private toplam 6 adet subneti Region’umuzdaki 3 farklı AZ ye kuaracağız. Her sabnet için 10.10.0.0/16 network’ün altında olacak /24 lük networkler. Public ve private subnetler aşağıda gösterildiği gibidir.

Public olanlar

10.10.10.0/24 (251 tane IP adresi atayabiliriz ama biz sadece 251 tanesini kullanabileceğiz)

10.10.20.0/24

10.10.30.0/24

Private olanlar

10.10.11.0/24

10.10.21.0/24

10.10.31.0/24

bizim 10.10.10. 0/24'ün ilk kısmı bizim Kendi sanal ağımızda kullanmak istediğimiz ip adresin bloğunun nerede başladığını belirtir ikinci kısmı olan /24'te Bu adres kulunun ne büyüklükte olduğunu belirtiyor. her bir bloktaki ilk adres network adresi en son adres broadcast adresidir.

toplamda 254 IP’yi subnetteki makinelere atayabilirim. Fakat 10.10.10.0 network, 10.10.10.1 VPC router, 10.10.10.2 DNS hizmti için rezerve ediyır. 10.10.10.3 ü de ileride lazım olur diye rezerve ediyor. Böylece 251 IP adresi kullanılabilir.

Internet Gateway = Bu cihaz public subnetlerimizin internete bağlanmasına, Aynı zamanda da internetten de public subnetlerimize bağlanılması imkan verecek cihaz. Aslında cihaz diyoruz ama sanal bunların hepsi

NAT GW : Private subnetteki kurulan makinelerin internetten erişimlerinin olamayacağını da sözyledik ama belki bu cihazların belki de dışarıya çıkıp internete erişmeleri gerekecek. belki de sunuculara güncelleme almaları gerekecek. işte bunun için NAT GW denilen cihazlar koyuyoruz

Endpoint: tüm bu cihazların AWS servislerine daha hızlı erişsinler diye bunun için bir private link atayacağız. buna da end point diyecegiz

1) create a vpc named working-group-vpc-a' CIDR blok is ----> 10.10.0.0/16

# no ipv6 CIDR block

# tenancy: default (Aynen Ec2 daki dedicate mantığına göre hazırlanan bir servis. Kendimize dedice edebiliriz bu VPC yi)

1.1) VPC yi göster

2) İlk VPC oluşturulduktan sonra yapılacak ilk hareket actions kısmından enable DNS hostnames for the vpc working-group-vpc-a' Eğer bunu enable etmezsek bu VPC altındki makinelere bir DNS host name vermyecek ve bu makineler birbirleri ile haberleşemeyecekler. Bunun için bir DNS e ihtiyaç duyuyorlar.

3) Route table’a gel. Her VPC oluşturulduğunda o VPC ile beraber bir route table oluşturulur. Route table o VPC altında nereye ne şekilde erişileceğini gösteren kurallar tablosudur. Bizim oluşturduğumuz subnetler, ne şekilde bir yere erişsinler, nereden nereye gitsinler/gidemesinler, bunların tüm hepsi bu route table altında belirlenir.

rename the route table of the vpc 'working-group-vpc-a' as 'working-group-default-rt'

4)VPC oluşturlduktan sonra ilk yapmamız gereken şeylerden biri internet gateway i oluşturmaktır

5) Az önce VPC mizi oluşturduk, bir nevi data centre’mızı kurduk. Şşimdi bu data center’ımızın dışarı internete nereden çıkacağını ayarlayalım. Yani bir nevi buraya bir ADSL, bir fiber hat çekiyoruz.

create an internet gateway named 'working-group-igw'

attach the internet gateway 'working-group-igw' to the vpc 'working-group-vpc-a'

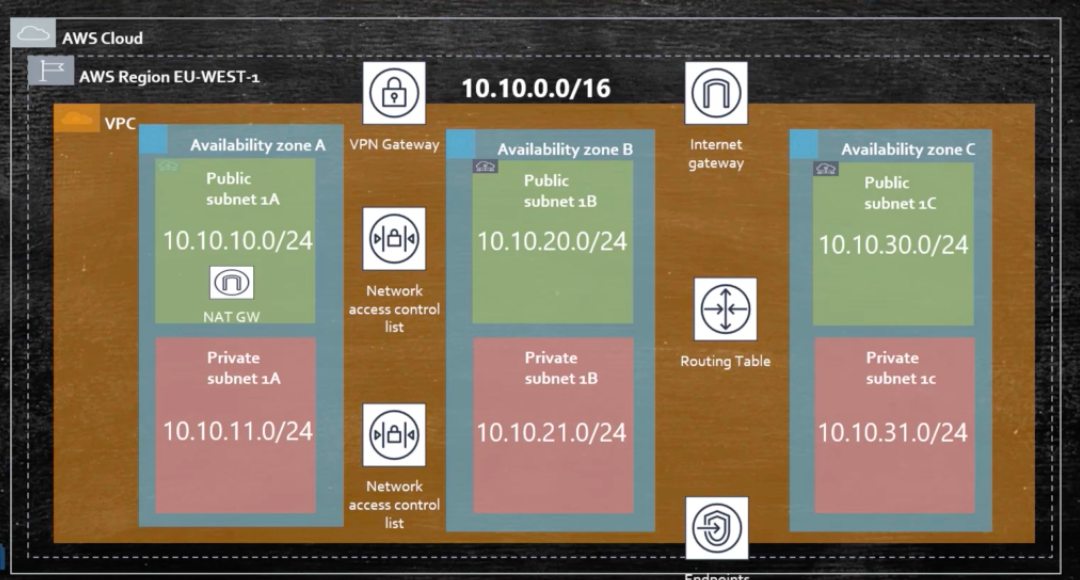
tıkla, actions dan attack to vpc de ve vpc yi listeden seç attach de.

diagrama geri dön

Bu internet Gateway i oluşturduk

VPC olarak Bu dış çeperi oluşturduk







İşte bu noktada internet Gateway oluştuktan sonra, bizim paket yönlendirme kurallarını düzenlememiz lazım. Bunu da route table dan yapıyoruz

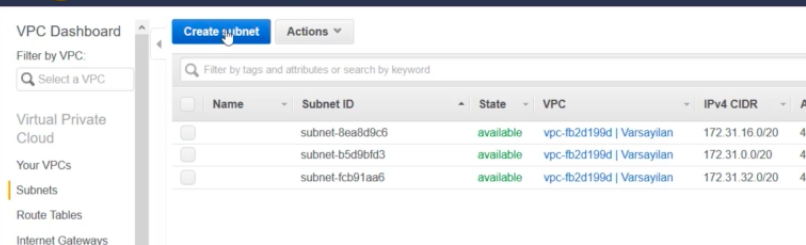
6)Gel route table’a, ilgili oluşturulan table’a tıkla, routs’a gel aşağıda, edit routes a tıkla

bir tane default route var. diyor ki eğer bu vpc nin içindeki paket destination olarak 10.10.0.0/16 dan gidiyorsa bunların hepsi localdir. Fakat bir tane daha eklemeye tıkla ve destination 0.0.0.0/0 yani dış dünyaya paket gönderiyor isen bunu oluşturulan target internate gateway e teslim et denilecek.

# add a route for destination 0.0.0.0/0 (any network, any host) to target the internet gateway 'working-group-igw' in order to allow access to the internet

# explain routes in the working-group-default-rt

7) şimdi subnetleri teker teker yaratacağız.



oluşturulan VPC seçilir, AZ seçilir buna göre aşağıdaki değerler girilerek subnetler oluşturulur

# create a public subnet named 'working-group-az1a-public-subnet' under the vpc working-group-vpc-a in AZ us-east-1a with 10.10.10.0/24

# create a private subnet named 'working-group-az1a-private-subnet' under the vpc working-group-vpc-a in AZ us-east-1a with 10.7.11.0/24

# create a public subnet named 'working-group-az1b-public-subnet' under the vpc working-group-vpc-a in AZ us-east-1b with 10.10.20.0/24

# create a private subnet named 'working-group-az1b-private-subnet' under the vpc working-group-vpc-a in AZ us-east-1b with 10.10.21.0/24

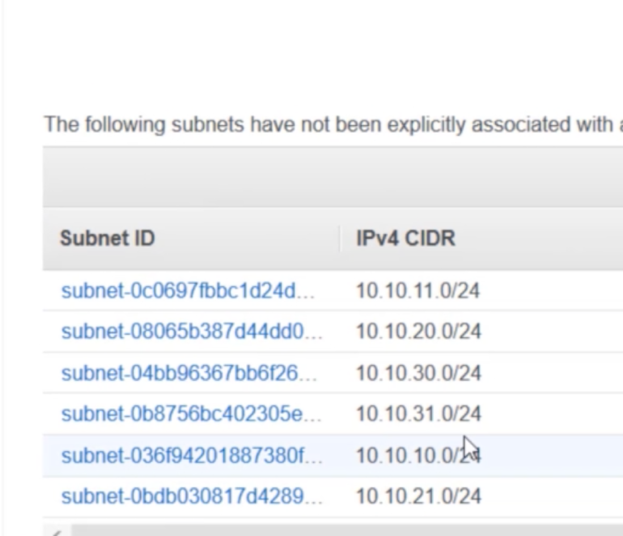
# create a public subnet named 'working-group-az1c-public-subnet' under the vpc working-group-vpc-a in AZ us-east-1c with 10.10.30.0/24

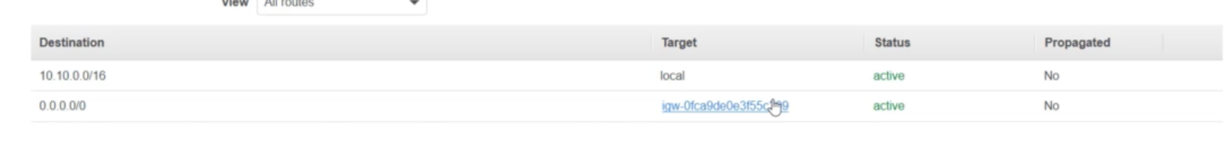
# create a private subnet named 'working-group-az1c-private-subnet' under the vpc working-group-vpc-a in AZ us-east-1c with 10.10.31.0/24

# explain the subnet descriptions and reserved ips (why 251 instead of 256)

# show the default subnet associations on the route table working-group-default-rt (internet access even on private subnets)

8) peki public ya da private olmasını ne belirliyor. Tüm ayarlar aynı şekilde oluşturuldu. Peki bunu nasıl belirleyeceğiz. şimdi bunu görmek için route table a gelinir. Subnet associations a bakılır. şu şekilde görülür. hepsi bir route table a atanmış görülür. Bi bir subnet yarattığımızda o subnet o VPC nin route table ı ile default olarak ilişki kurar. Route a tıkla şunu görürsün;





işte buburada private ve public ayrımını yeni bir route table yaratarak sağlarız. yeni oluşturlan route table a private olanları bağlayack ve ona kurallar gireceğiz.

# create a private route table (not allowing access to the internet) named 'working-group-private-rt' in the vpc working-group-vpc-a for private subnets

# show the routes in the route table working-group-private-rt

# associate the route table working-group-private-rt with private subnets

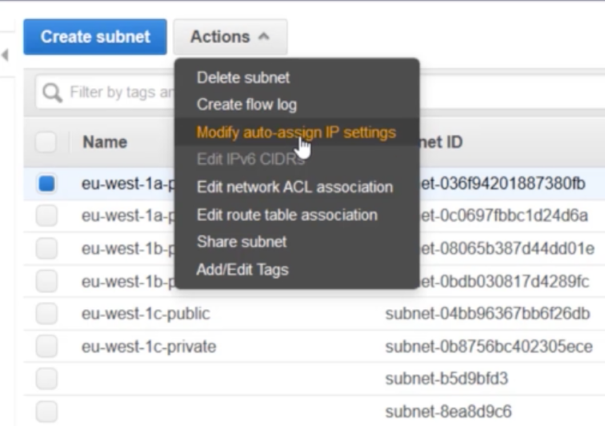
bu da yaratılan route table seçilir ve subnet associations seçilir. ve private olanlar seçilerek bu route table a atanır. Bu route table ın route larına bakarsak aşağıdakini görürüz



yani bu route table in herhangi bir dış dünyaya paket teslimi için bir yolu yok

9) Biz public subnetlere koyduğumuz makinelerin dışdünyaya erişmesini istediğimizi söylemiştik. bunun için buraya koyduğumuz makinelerin public subnetlere sahip olması gerekir. Bu varsayılan olarak kapalı gelir Bunu açmak için her subnet seçilir ve aşağıdaki işlem uygulanır.

# enable Auto-Assign Public IPv4 Address for public subnets



Bu sayede bu subnetler içerisine konulan tüm makinelere public IP adresiverilir.

10) Bastion Host(Jump Box)

Özgür 66.VPC 2 dk dan ititbaren senaryoyu izle

Şimdi private networkde kurulan EC2 lara dış dünyadaki internetten erişilebilmesi için Public networklerden bir tanesine jump box denilen bir sanal makine kuracağız. bu makinenin tüm amacı, dış dünyadan bu makineye bağlanıp daha sonra private networkdeki tüm makinelere bu makine üzerinden atlayarak ulaşmam sağlanacak.

AWS management consola gidelim ve 2 tane instance oluşturalım. Oluşturuken VPC de public subnetten biri seçilir ve eğer Autoassign Public IP atanırsa bu makineye otomatik olarak bir IP atanır ama biz zaten subneti oluşturuken içine konulan makinelere otomatik IP ata demiştik ondan dolayı bunu seçmemize gerek yok. (IAM role seçti özgür). Şimdi yeni bir tane security grup yaratacağız. Çünkü sec grouplar VPC ler ile ilişkilendirilmiş durumdaır. Önceki oluşturulan sec grıuplar burada görülmez. Yeni bir sec group oluşturulur.

SSH --> tüm dünyadan aç

http --> tüm dünyadan aç

https ---> tüm dünyadan aç

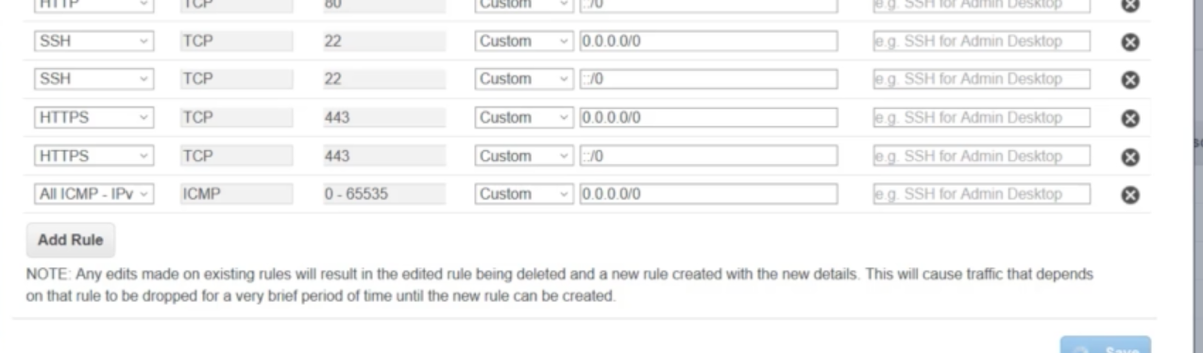
tekrardan bir EC2 instance yarat. ama VPC de oluşturulan VPC yi seç ama bu sefer private olanı seç ki alet private bölümde olsun. Hatta bu durumda auto assign IP adres in disable olduğu görülür. IAM rol olarak S3 ve EC2 ilişkilendirildi. az önce oluşturulan sec grup seçilebilir.

Bu iki sunucu arasındaki farklara bakarsan public olanda IP adresi olduğu görülür ve dış dünyadan erişilebilir. Ama private olanda IP adresi yoktur, private adresi var olduğu görülür, AMA NETWORK İÇİNDEDİR. Yani VPC içerisindeki sanal makineler bu makineye erişir ama dış dünyadan bir makine erişemez.

Tüm önemli makineler private networkde kurulur. public olanda ise bir makine atanarak dış dünyadan tüm yöneticiler bu jump box a bağlanır ve buradan içeridek makinelere bağlanırlar. şimdi bunu simüle edelim

Şimdi public sunucuya bağla terminalden ve ping [private instance IP adresi ] yaz ve erişilemediğini gör. Sonra bu erişime açılmasını göster .

Security grous a gel ve private’ye atadığın sec groups’u seç. inbounds a gel ve edit diyerek ping paketlerinin protokolü olan ALL ICMP-IPv



tekrardan public sunucuna dön ve tekrar aynı komutla ping at! Yani ben bağlandığım jump box dan içerideki makineye erişebildiğim görüldü.

Peki içerideki makineye nasıl bağlanacağım?

Bunun için key-pair dosyası olan pem dosyasına ihiyacım var. pem dosyasını txt formatında aç ve içeriğini kopyala, public instance ine gel ve aynı pem uzantılı boş dosyayı vi ile aç ve içerisine yapıştır, chmod 400 ile dosyaya hak ver ardından bağlanmayı ssh kullanılarak, Private sunucunun IP adresi ile bağlanmayı dene ve bağlanabildiğini gör. (private’ye ssh izni verdiğimiz için bu erişim sağlanabilir). exit dersen public instance ye bağlanılır.

Aşağıdaki Jump Box uygulaması Callahan ın uygulamasıdır.

Aşağıdaki bölümde Bastion Host (Jump box) için instance

# launch an instance (amazon linux 2) in az-1a public subnet using the public-custom-sg-for-working-group-vpc-a and name it 'ec2-in-az1a-public-sn'

# launch an instance (amazon linux 2) in az-1a private subnet using the private-custom-sg-for-working-group-vpc-a and name it 'ec2-in-az1a-private-sn'

# summarize what we have done so far

# show and explain the descriptions on ec2-in-az1a-public-sn instance

# show and explain the descriptions on ec2-in-az1a-private-sn instance

# connect to the ec2-in-az1a-public-sn instance with ssh

# show the interface configuration of the ec2-in-az1a-public-sn

# show the routes for the ec2-in-az1a-public-sn

# show that ec2-in-az1a-public-sn instance has internet connection (use ping and curl command)

# show that ec2-in-az1a-public-sn instance has connection with instances in private subnet(use ping)

# PART 2 - Bastion Host and NAT Gateway/Instance

# discuss about how to connect to the ec2-in-az1a-private-sn instance

# launch an instance (amazon linux 2) in az-1a private subnet using the private-custom-sg-for-working-group-vpc-a, by enabling public-ip-assign and name it 'ec2-w/public-ip-in-az1a-private-sn'

# show that even with public-ip, we cannot connect to it with ssh due to rules in routing table

# setup a jump box/bastion host using the currently running ec2-in-az1a-public-sn instance to connect the instances in private subnet

# add your private key to the ssh agent on your localhost

ssh-add .ssh/call-training.pem

# run the ssh agent if it is returning error like "Could not open a connection to your authentication agent"

eval "$(ssh-agent)"

# try again to add your private key to the ssh agent

ssh-add .ssh/call-training.pem

# connect to the ec2-in-az1a-public-sn instance in public subnet

ssh -A ec2-user@ec2-3-88-199-43.compute-1.amazonaws.com

# once logged into the ec2-in-az1a-public-sn (bastion host/jump box), connect to the ec2-in-az1a-private-sn instance in the private subnet

ssh ec2-user@10.7.2.20

# show that the ec2-in-az1a-private-sn has connection with the ec2-in-az1a-public-sn instance

ping -c 4 ec2-54-242-52-186.compute-1.amazonaws.com

ping -c 4 ip-10-7-1-121.ec2.internal

# show that the ec2-in-az1a-private-sn does not have connection with the internet

ping -c 4 google.com

curl google.com

# show the interface config of the ec2-in-az1a-private-sn

ifconfig

# show the route config of the ec2-in-az1a-private-sn

route

11) Network Access List NACL ve Security group konusuna bakalım.

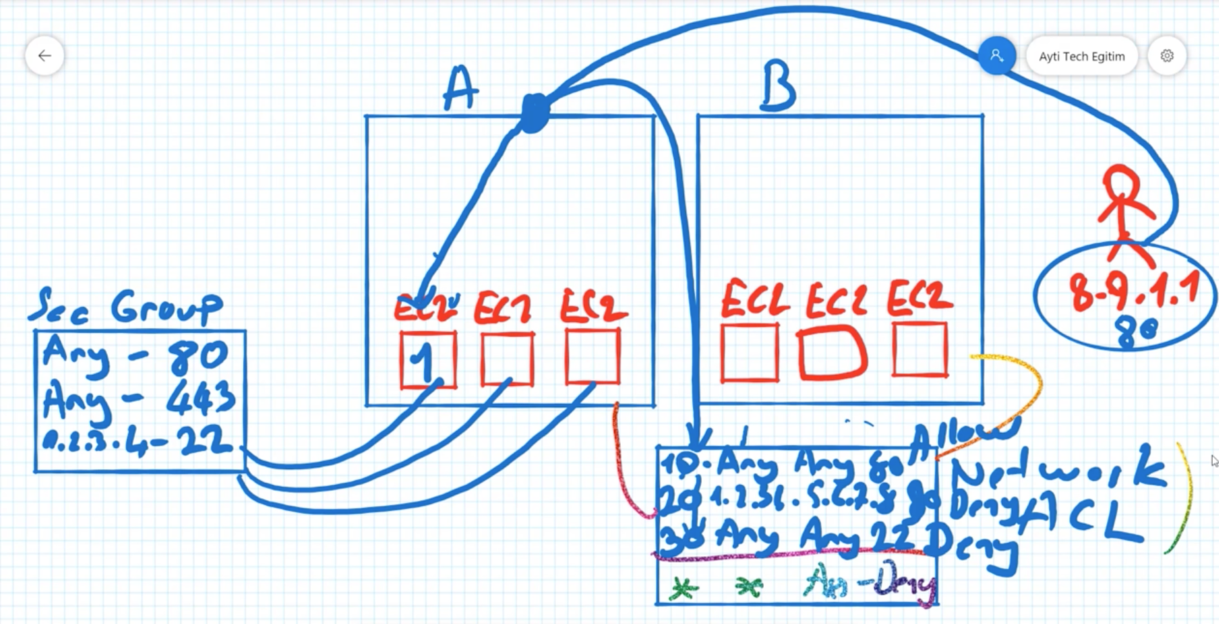
Network ACL atandığı subnetin hangi orijinden hangi türde trafiği kabul edeceğine ve hangi destinasona hangi türden trafik yönlendirileceğinin kuralarının yazıldığı yerdir.Her subnet en azından bir tane NACL e atanmak zorundadır, bu bir kural. her subnet için ayrı NACL yaratılabileceği gibi bir tane oluşturulup tüm subnetler bu NACL a atana da bilir. NACL ler subnet bazında security gruplardır. Bu ACL ler de tüm subnete atanıyor ve o subnet içerisindeki tüm kaynaklara erişimi denetliyor, bir subnete ancak bir tane nacl atayabilirsin, birden fazla olmaz. Ama buna karşı security group ise buna benze bir amaçla oluşturulan ve hangi orijinden hangi türde trafiği kabul edeceğine ve hangi destinasyona hangi türden trafik göndermesine izin vereceğine belirlediğimiz kabul bölümüdür. Fakat bu ikisi arasındaki en temel fark NACL subnetlere atanıp subnet bazında kurallar belirler. security grouplar ise makine bazında erişim kuralları belirlerler.

ÖZGÜR 67 uygulama 2:30 civarında başlayan örneği izle.

NACL nasıl çalışıyor. NACL bağlı olduğu subnetin tamamında geçerli olur. Defaultta NACL

x ------> x ------> Any -----> Deny

kuralı vardır. Bu kuralın üzerine yeni kurallar yazılır ve bir derecelendirme verilir. Yukarıdan aşağıya bu derecelendirmeye göre bakılır ve herhangi bir kurala uyuyor ise bu kural uygulanır ama uymuyor ise en altta az önce yazığımız kural uygulanır ve erişim reddedilir. Burada allow belirlendiği gibi deny lar da belirlenebilir. Ayrıca NACL ler subnet içerisindeki makinelerin erişim kurallarını da belirler. Yani bu şekildeki subnet içerisinden bir erişim isteğinde yine NACL tablosuna bakılır ve bu tabloya göre hareket edilir. Bu da security group dan ayrılan ikinci temel farklılığı

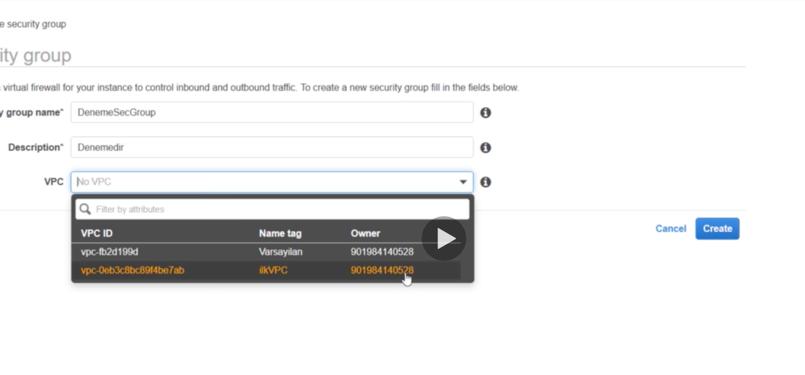


12)

# show the default security group for the vpc working-group-vpc-a and name it as 'default-sg-for-working-group-vpc-a'

security grouplar çok basit, default’da security grup içinde herşey kapalı. biz istediğimiz port’a izin verebiliriz.

# create a custom security group with default values for public subnets within the vpc working-group-vpc-a and name it as 'public-custom-sg-for-working-group-vpc-a'



# show and explain inbound and outbound rules of public-custom-sg-for-working-group-vpc-a

Inboun rules boş görülür. Yani bu security group herhangi bir sanal makineye atanırsa dış dünyadan bu sana makineye ulaşılamayacak. Outbound rules ise atanan sanal makinenin dış dünyaya erişim kurallarıdır ki, Outbound rules default olarak dış dünyadaki her türlü porttan her destinasyona tüm trafiğe izin verilmiş gelir. Yani esas kısıtlama yapılan yer inbound rule’lardır. Inbound rule da ise biz bu sanal makineye ne şekide erişilmesi isteniyorsa ona göre izin veririz. Bu da inbound rules a gelinir ve edit rule denir. Aşağıdaki kuralları girelim;

1. a. set inbound rule for public-custom-sg-for-working-group-vpc-a --> type:ssh, port:22, source:my ip, description: 'allow ssh from call localhost'

1. b. set inbound rule for public-custom-sg-for-working-group-vpc-a --> type:ssh, port:22, source:public-custom-sg-for-working-group-vpc-a, description: 'allow ssh from public-custom-sg-for-working-group-vpc-a'

1. c. set inbound rule for public-custom-sg-for-working-group-vpc-a --> type:all icmp - ipv4, port:all, source:my ip, description: 'allow icmp from call localhost'

1. d. set inbound rule for public-custom-sg-for-working-group-vpc-a --> type:all icmp - ipv4, port:all, source:public-custom-sg-for-working-group-vpc-a, description: 'allow icmp from public-custom-sg-for-working-group-vpc-a'

2. create a custom security group with default values for private subnets within the vpc working-group-vpc-a and name it as 'private-custom-sg-for-working-group-vpc-a'

2.a set inbound rule for private-custom-sg-for-working-group-vpc-a --> type:all, port:all, source:public-custom-sg-for-working-group-vpc-a, description: 'allow all from public-custom-sg-for-working-group-vpc-a'

2.b set inbound rule for private-custom-sg-for-working-group-vpc-a --> type:all, port:all, source:private-custom-sg-for-working-group-vpc-a, description: 'allow all from private-custom-sg-for-working-group-vpc-a'

13. Buna karşılık MACL ler biraz daha karmaşık. NACL e gelindiğinde yaratılan NetworkACL in associated with diye bir kısmı var. Yani bu ACL hangi subnet e atanmış. Bir NACL birden fazla subnete atanır ama her subnet’e saece bir NACL atanabilir. Aklında tut. Create NACL yapalım



#show the default network acl for the vpc working-group-vpc-a and name it as 'default-nacl-for-working-group-vpc-a'

# explain the working principles of rules -> the lower number: the higher priority, rules numbering btwn 1-32766, rules policies: allow/deny, default rule of \*.

Oluşturulan NACL yi aç. security gruplardaki gibi burada da inbound ve outbound rules var bunun yanında burada asssociation diye bir yer var. Burada bu NACL , hangi subnet e bağlanırsa o subnet o ACL in kurallarına göre çalışır.

Inbound u aç…. Herşey engellenmiş durumda default olarak. Security grouplar ile NACl lerin ikinci temel farklılığı budur. Security gruplarda biz izin veriyorduk. ama NACL lerde hem izin verebiliyor hem de engelleyebiliyoruz. Ayrıca NACL lerdeki kurallar bir sıralama ile çalışır.

A. create a custom network acl for public subnets on the vpc working-group-vpc-a and name it as 'public-custom-nacl-for-working-group-vpc-a'

# show default the inbound and outbound rules of public-custom-nacl-for-working-group-vpc-a

a. set inbound rule for public-custom-nacl-for-working-group-vpc-a; rule #: 80, type:ssh, port:22, source:0.0.0.0/0, policy: allow

b. set inbound rule for public-custom-nacl-for-working-group-vpc-a; rule #: 120, type:all icmp - ipv4, port:all, source:0.0.0.0/0, policy: allow

c. set outbound rule for public-custom-nacl-for-working-group-vpc-a; rule #: 80, type:ssh, port:22, destination:0.0.0.0/0, policy: allow

d. set outbound rule for public-custom-nacl-for-working-group-vpc-a; rule #: 120, type:all icmp - ipv4, port:all, destination:0.0.0.0/0, policy: allow

# explain the rules applied for public-custom-nacl-for-working-group-vpc-a

Artık bu NACL i atadığımız subnet’e gelen tüm paketler buradaki kurallara göre denetlenecek.

B. create a custom network acl for private subnets on the vpc working-group-vpc-a and name it as 'private-custom-nacl-for-working-group-vpc-a'

# set inbound rule for private-custom-nacl-for-working-group-vpc-a; rule #: 80, type:all, port:all, source:10.7.0.0/16, policy: allow

# set outbound rule for private-custom-nacl-for-working-group-vpc-a; rule #: 80, type:all, port:all, source:10.7.0.0/16, policy: allow

Bunlar ayarlandıktan sonra subnet association kısmından edit subnet association denilerek istenilen subnetler bu NACL e atanır.

14. Açık olan bir instance’ye bir bak IP adresini göster. sonra kapatıp tekrar aç. IP nin değiştiğini göster. Public sunucu yeniden başlatıldıktan sonra yeni bir IP adresi alıyor. Bu IP adresi bize dedike değil. Her açıp kapadığınızda Amazon kendi hazvuzundan hangi adres boşta ise onlardan bir tanesini bize atıyor. Bu bazı durumlar için sakıncalı. Mesela; IP adresinin sabit olamsı gerekebilir, bir lisansı bu IP adresi üzerine atamamız gerekebilir vs. İşte bu durumlarda Elastic IP denilen bir hizmet sunulur Amazon tarafından. Bu sabit IP adresi istenirse sanal makinelerde istenirse diğer kaynaklarda kullanabiliriz.

Konsoldan Elastik IP adresine gel ve allocate new adres de ve allocate e bas. Bize bu durumda Amazon bize bir IP adresi dedike edilir.

# allocate Elastic IP and name it 'working-group-eip-for-nat-gw'

Action a gel associate adres e tıkla ve hangi makine ile ilişkilendirmek istiyorsan ona attach et. Hangi snal makineye atadı ise onu göster ve artık Elastic IP nin atandığını görürsün. Instance’ı açıp kapat ve değimediği görülür.

15.

Arkadaşlar VPC yaratıp subnetleri yaratınca 3 public, 3 private subnet yarattık. Bunların public ya da private olmasını route tablosu belirliyor. Burada public subnetler için bir kural koyduk. Dedik ki dışarıya çıkış paketlerini internet Gateway’e teslim et…. Buradan da internete çıkış yapılıyor.

# discuss about how to connect to internet from the ec2-in-az1a-private-sn instance in private subnet

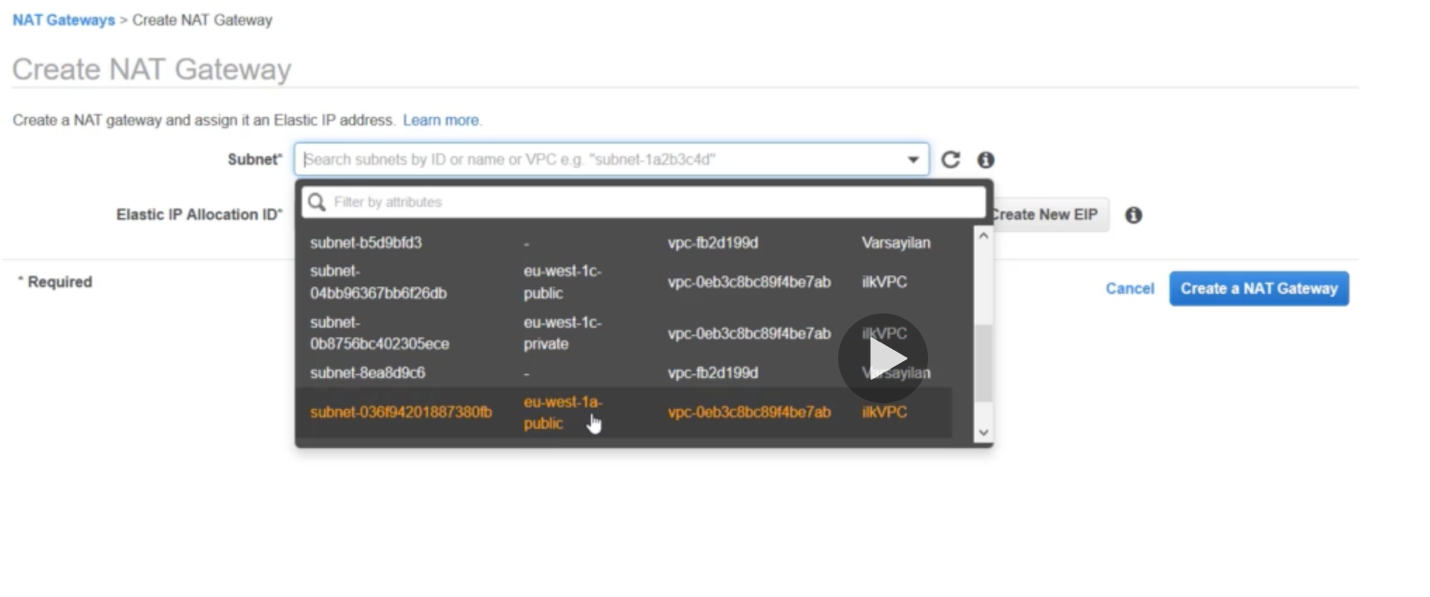
Ama private network de böyle bir dış dünya’da internete erişim yoktur. Ama kimi zaman belli güncellemeleri almak için dış dünyaya private network deki makinelerin erişmesi gerekebilir, ya da internetten bir paket çekmesi gerekebilir. Bu durumda kullanılacak şey NAT Gateway ya da NAT instance dır.

Bunun için önce bir NAT Gateway create edilir ve bu NAT gateway i public subnetlerden birine koyuyoruz. Sonra private subnetlerin olduğu route table a diyoruz ki, sen internete gidiyorsan paketleri bu NAT Gateway e teslim et. NAT Gateway bir internet Gateway olmadığı için dış dünyadan bu makinelere erişilemiyor

# set up internet connection with NAT Gateway

Sol konsoldan NAT Gateways e gel ve create NAT Gateways e tıkla

# create a NAT Gateway in the public subnet working-group-az1a-public-subnet using the working-group-eip-for-nat-gw and name it 'working-group-az1a-nat-gw'

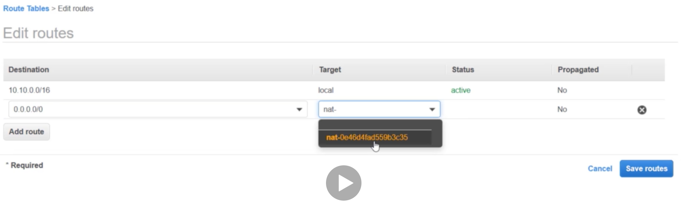


Buna bir Elastic IP adresi atamamız gerekiyor. Bunun için bir tane yaratılan Elastic IP yaratılıp atanır. Sonra da private subnetlerin bağlı olduğu route table a gelinir ve Routes kısmı tıklanır, edit rules denir.

# set a rule in working-group-private-rt to ensure requests to be routed to the NAT Gateway working-group-az1a-nat-gw

Destination -----> 0.0.0.0./0

Target --------> Oluşturulan NAT Gateway seçilir.



ÖZGÜR 68- 8.30 dk anlatımı inceleyebilirsin.

# connect to the ec2-in-az1a-private-sn instance in the private subnet with ssh via bastion host

# show that the ec2-in-az1a-private-sn now has connection with the internet

16. NAT Gateway in yaptığı işi yapmanın tek yolu bu değil. NAT Gateway herşeyi Amazon tarafondan yönetilen bir servis. Fakat bunun yerine biz kendimiz bu işi yönetmemiz gerekirse ve bunu da arka planda bir sanal makine ile yapmak istersen buna da NAT instance deniliyor.

NAT Gateway ------ AWS tarafından yönetilir, Artısı tamamen otomatik, fail over imkanı var ve 45 Mbit e kadar dandwith sağlıyor. Ama bazı durumlarda ihtiyacı karşılamıyor.

NAT Instance ------ Biz yönetiriz.

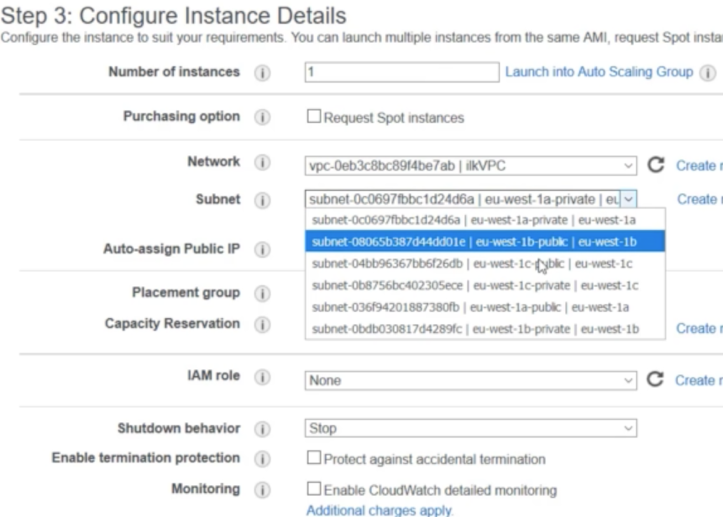
Konsola dönelim ve konsol üzerinden NAT Instance nasıl oluşturulur görelim;

# set up internet connection with NAT Instance

Bunun için özel nat instance seçilir.

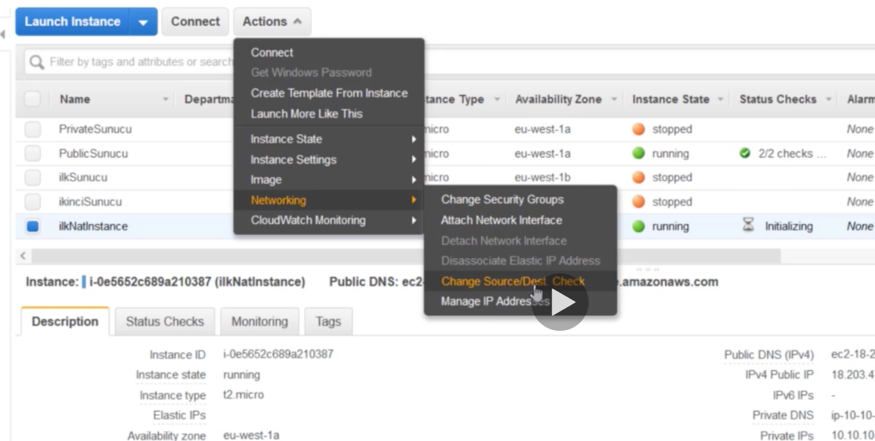
# launch an instance from amzn-ami-vpc-nat-hvm-2018.03.0.20181116-x86\_64-ebs (ami-00a9d4a05375b2763) in working-group-az1b-public-subnet using public-custom-sg-for-working-group-vpc-a and name it as 'NAT Instance'

ikinci ekranda NAT Instance kurarken kendi kapasitemize göre bir instance type seçmemiz gerekir. Fakat 200 sunucu varsa, t2.mikro karşılamaz. bu da buradan ayarlanır. Sunrasında da oluşturduğumuz VPC de public subnetlerden birine yerleştirilir makine



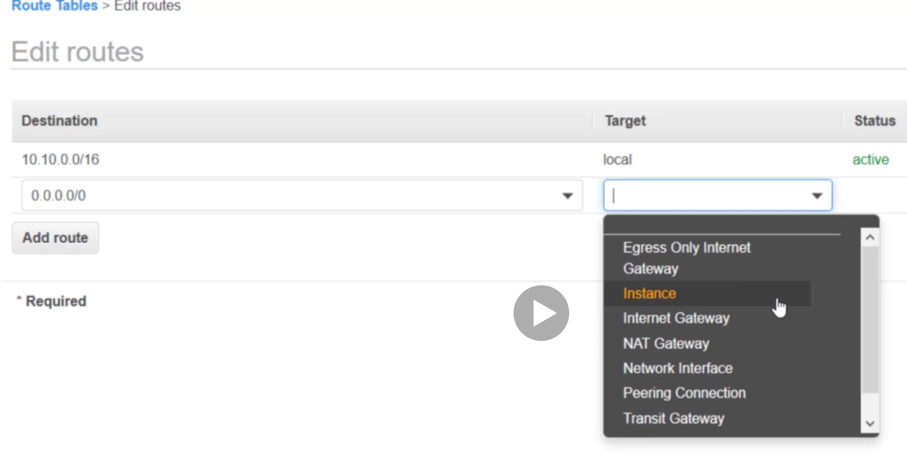
Bu AMI da magnetic seçilmiş ama SSD yapabilirsin. Security grup olarak ssh, http, https olan secgroup seçilebilir. launch edilir.

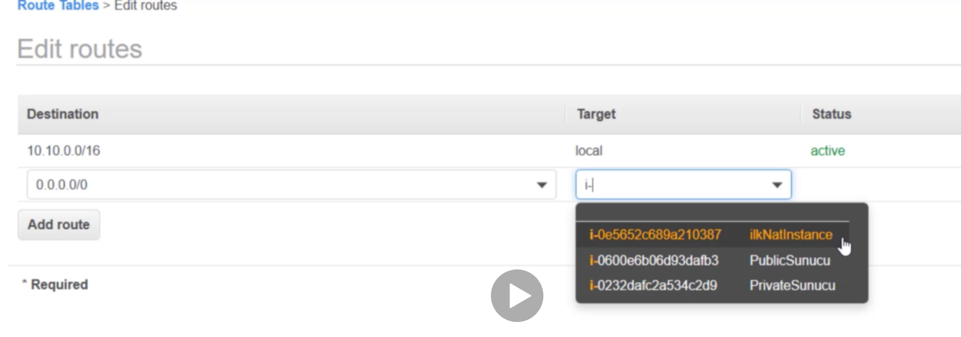
Ilk yapılması gereken şey (sınavda soru çıkılabiliyor) Bir nat instance yarattı isek ve nat instance olarak kullanılacaksa bu instance seçilir, actions a gelinir network den change sourse destination seçilir ve disable edilmesi gerekir



# disable source/destination check on NAT Instance

Normalde bu makineler hangi kaynaktan geliyor hangi kaynağa gidiyor diye kontol ederler ve paket kendilerine ulaşmıyor ise paketi düşürürler ve benimle alakalı değil derler. Ama NAT ınstance de bu sözkonusu değil. Biz bunu disable etmezsek bu durumda içerideki makine dışarı bu NAT Instance üzerinden ulaşmak istediğinde nat instance pakit benimle alakalı değil deyip paketi düşürecek. Bunun olmaması için disable edildi. NAT Instance oluşturulduktan sonratekrardan VPC ye geri dönülür. Private için olan Route Table a gelinir. ve orada ya eski oluşturulan private instance ler için olan route table’ın route kısmına gelinir ve Nat Gateway satırı silinir. Bunun yerine Nat instance yi gösteren bir güncelleme yapılır;





seçilir. ya da aşağıda olduğu gibi yeni bir private route table oluşturulurve ona Nat Instance ataması yapılır.

# create a private route table named 'working-group-private-rt-for-nat-ins' in the vpc working-group-vpc-a for working-group-az1b-private-subnet

# show the routes in the route table working-group-private-rt-for-nat-ins

# set a rule in working-group-private-rt-for-nat-ins to ensure requests to be routed to the NAT Instance

# associate the route table working-group-private-rt-for-nat-ins with private subnet working-group-az1b-private-subnet

# launch an instance (amazon linux 2) in az-1b private subnet using the private-custom-sg-for-working-group-vpc-a and name it 'ec2-in-az1b-private-sn'

# connect to the ec2-in-az1b-private-sn instance in the private subnet with ssh via bastion host

# show that the ec2-in-az1a-private-sn now has connection with the internet

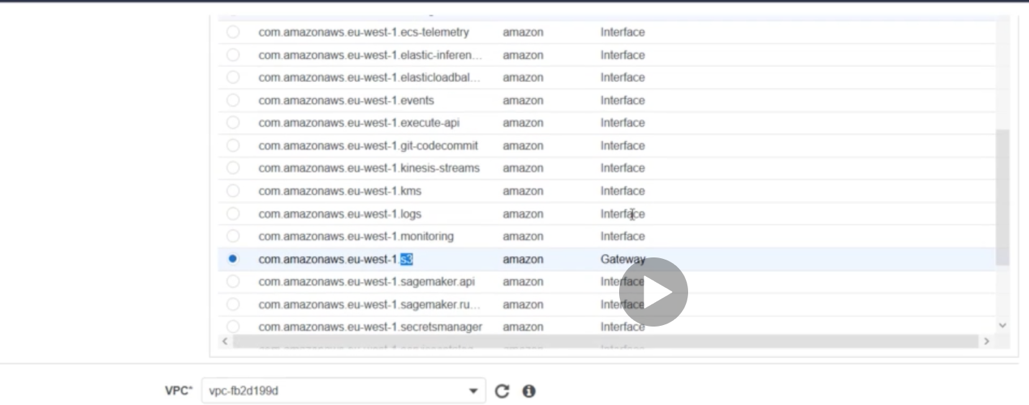
Nat Instance larda TCP temelli ayarlar yapılabiliyor.

17. End point

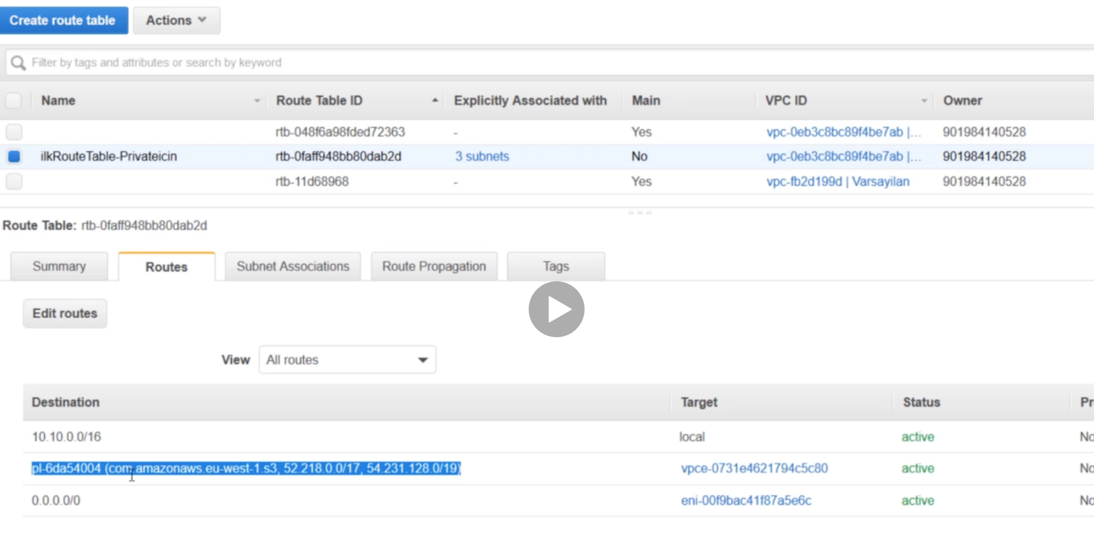
Şimdi benim oluşturduğum bu VPC içerisindeki public sanal makineler AWS’in farklı hizmetleri ile bağlantı kurmaları gerekiyor. Örneğin S3, RDS gibi hizmetlere erişmek istiyorlar. Bu tür bir erişim sözkonusu olduğunda bu makineler internet ile çıkış yapıp o servislere ulaşmaları gerekiyor. Ancak bunda bazı gecikmeler yaşanıyor. Halbuki çoğu zaman ulaşılacak olan servisler aynı AZ içerisinde birbirine çok yakın olabiliyor. Bu durumda AWS diyor ki; siz bu servislere internetten ulaşayım diye uğraşmayın. Ben size buz servislere daha hızlı ulaşabileceğiniz bir arka kapı oluşturayım, bunun adına da Endpoint diyeyim.

Konsolda Endpoint kısmına gelinir, create endpoint denilir

AWS services’i seçelim, s3 listelenir(ya da istediğin diğer başka servisler de olabilir. ) Aynı AZ içerisinde olan makinelerin yanında Gateway yazacaktır. İstenilen servis seçilir.

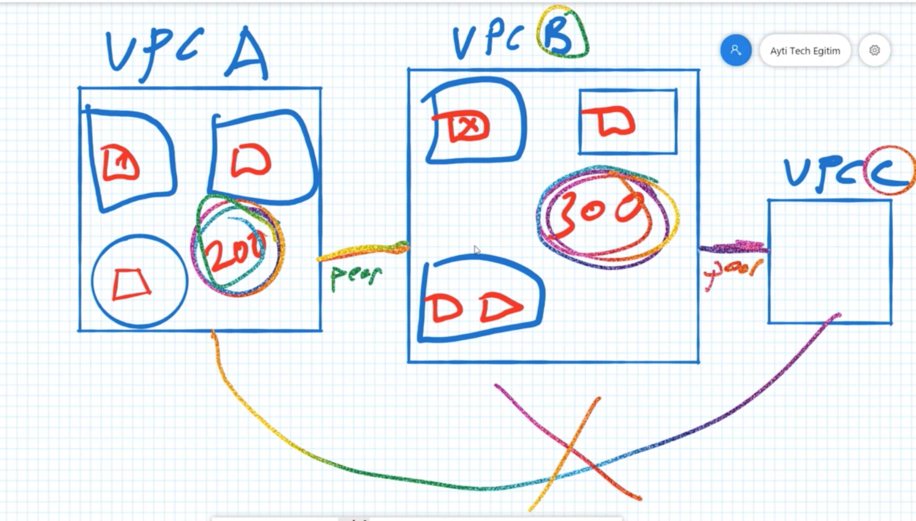


Hangi VPC ‘ye bağlanılmak istendiği sorulur, biz kendi oluşturduğumuz VPC ‘yi seçeriz. Peki hangi subnetlerin bağlı olduğu route table’ları güncelleyim diye sorar. Ben hem public pem private olanları güncelle diyorum. (Sadece public de seçilebilir) Policy yaratılabilir. Yarat dendikten sonra git route table ları kontrol et. Yeni bir madde eklendiğini göreceksin



VPC Peering

Benim VPC ‘m altında farklı subnetlerim var ve bu subnetlerimin içerisinde EC2 larım var. Ben şimdi istiyorum ki farklı subnetlerde bulunan makinelerim birbiri ile iletişim kurabilsin. Aslında bu public olanlar internet üzerinden birbiri ile konuşabiliyor. Ancak kabul edelim ki konuşlanan makine sayısı 200, 300 lerde. Bu durumda tek tek bu şuna gitsin türünden eşleştirmeler ya da engellemeler oldukça zahmetli olacaktır. Onun yerine bu iki network arasında peer yapıyorum. Buradan da A networkündeki kaynaklar B’ye, B’deki kaynaklar A VPC sindeki kaynaklara bu peering vasıtası ile erişebiliyorlar.



A, B’ye ve B de C’ye ulaşabiliyor ise A, C’ye ulaşamaz. Bu durumda A ve C arasında da bir peer kurmak gerekmektedir.

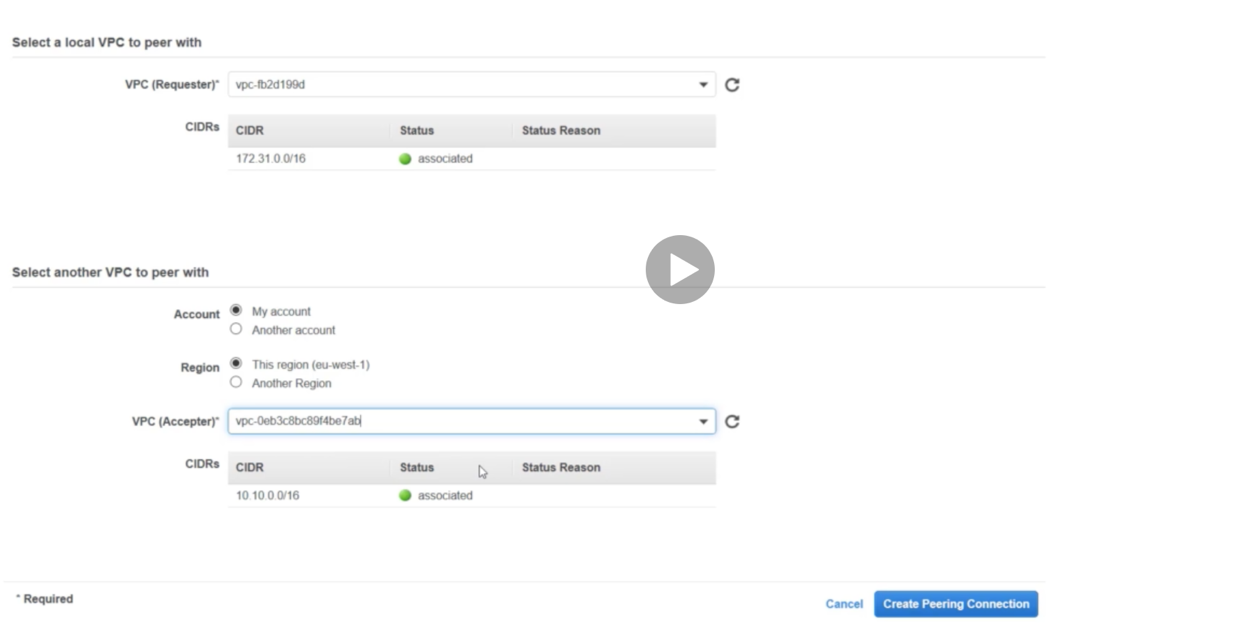
# connect to the ec2-in-az1a-public-sn instance in the public subnet with ssh

# show that the ec2-in-az1a-public-sn in working-group-vpc-a does not have connection with MySQL Database Server in default-vpc

şimdi peering oluşturalım…

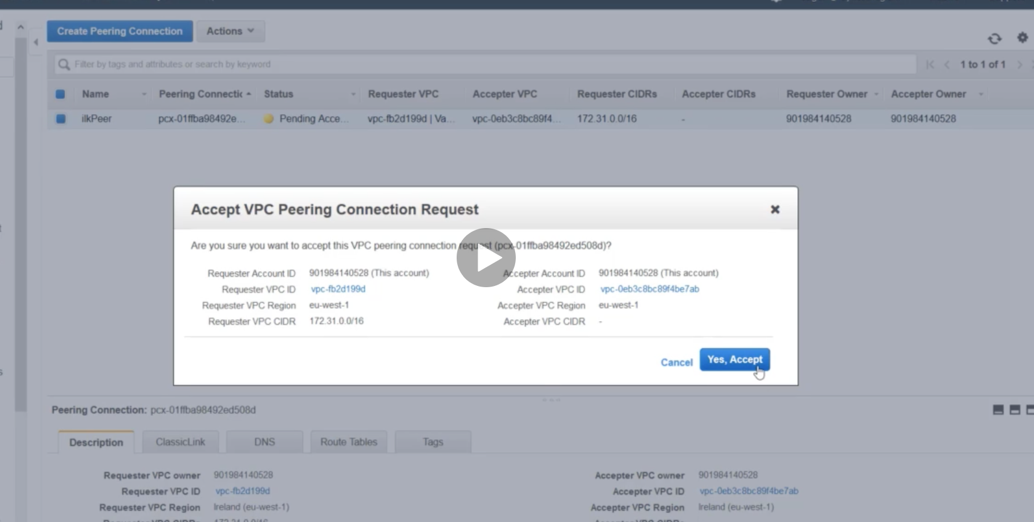
# create vpc peering between working-group-vpc-a (requester) and default-vpc (accepter) and name it 'working-group-vpc-a-peering-w/default-vpc'

Bunu oluşturmak için iki VPC olması gerekir. Şimdi bu iki VPC arasında bir peering kuralım. Sol konsolda VPC altında Peering Connections sekmesini tıkla, create peering connetion a tıkla



Not: Accpter ve Requester’ın IP bloklarının çakışmaması lazım DIKKAT.

Yukarıdaki request yapıldığında bunun onaylanması gerekir. Ben kendi hesaplarım arasında yaptığım için bu requesti yine bunu kendim onaylayacağım. Actions’a gelinir ve accept request e tıklanır ve aşağıdaki ekran karşımıza çıkar.



onay verilir.

# accept the request on peering dashboard

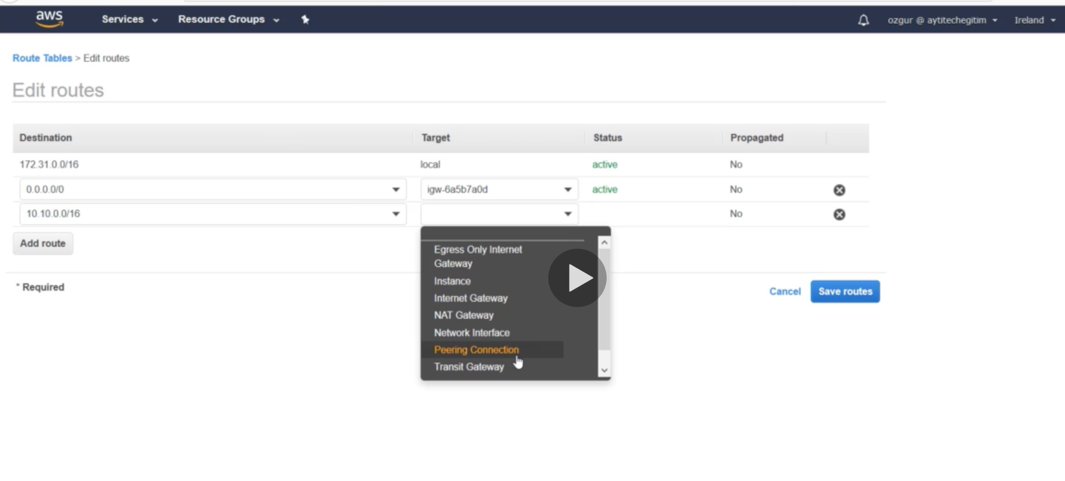
Onay verilince bana der ki; git route tablolarını güncelle. Modify route table a tıklanır. Route table’lar açılır ve edit routes tıklanır… Dikkat! Her ikisi de düzeltilmesi gerekmektedir.

# add route for peering connection from default-vpc in the working-group-default-rt

# add route for peering connection from default-vpc in the working-group-private-rt

# add route for peering connection from working-group-vpc-a in the default-rt





Sonra diğer route table’ da düzenlenir. Her biri yapılırken diğerinin IP adresleri yine kendisi üzerinden görülebilir.

Callahan’ın örnek devamı

# set inbound rule for db-servers-sg --> rule type:all icmp - ipv4, port:all, source:working-group-vpc-a, description: 'allow icmp for working-group-vpc-a'

# set inbound rule for public-custom-sg-for-working-group-vpc-a --> type:all icmp - ipv4, port:all, source:defaul-vpc, description: 'allow icmp for default-vpc'

# connect to the ec2-in-az1a-public-sn instance in the public subnet with ssh

# show that the ec2-in-az1a-public-sn in working-group-vpc-a now has connection with MySQL Database Server in default-vpc

# connect to the ec2-in-az1a-private-sn instance in the private subnet with ssh via bastion host

# show that the ec2-in-az1a-private-sn has connection with the internet

# connect to the MySQL Database Server in default-vpc with ssh

# show that MySQL Database Server in default-vpc has connection with the ec2-in-az1a-public-sn and ec2-in-az1a-private-sn instances in working-group-vpc-a