Öncelikle AWS sistemine login olmak için bir gmail hesabı oluşturulur.

**Yeni Servis Oluşturma**

1. **İlk olarak sağ üst köşeden sanal makine bölgesi Frankfurt seçilir.**

**Graphical user interface, application

Description automatically generated**

1. **Services bölümünden EC2 servisi seçilir. EC2 servisi sanal makine oluşturmak için kullanılır.**

**Graphical user interface, application

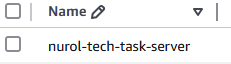
Description automatically generated**

1. **Launch Instance denilir.**

**Graphical user interface, application

Description automatically generated**

1. **Sanal makineye isim verilir.**

****

1. **İşletim sistemi olarak Amazon Linux seçilir.**

**Graphical user interface, text, application

Description automatically generated with medium confidence**

1. **Create key pair denilir.**

**Graphical user interface, application, Teams

Description automatically generated**

1. **Key pair ismi verilir**

Graphical user interface, text, application

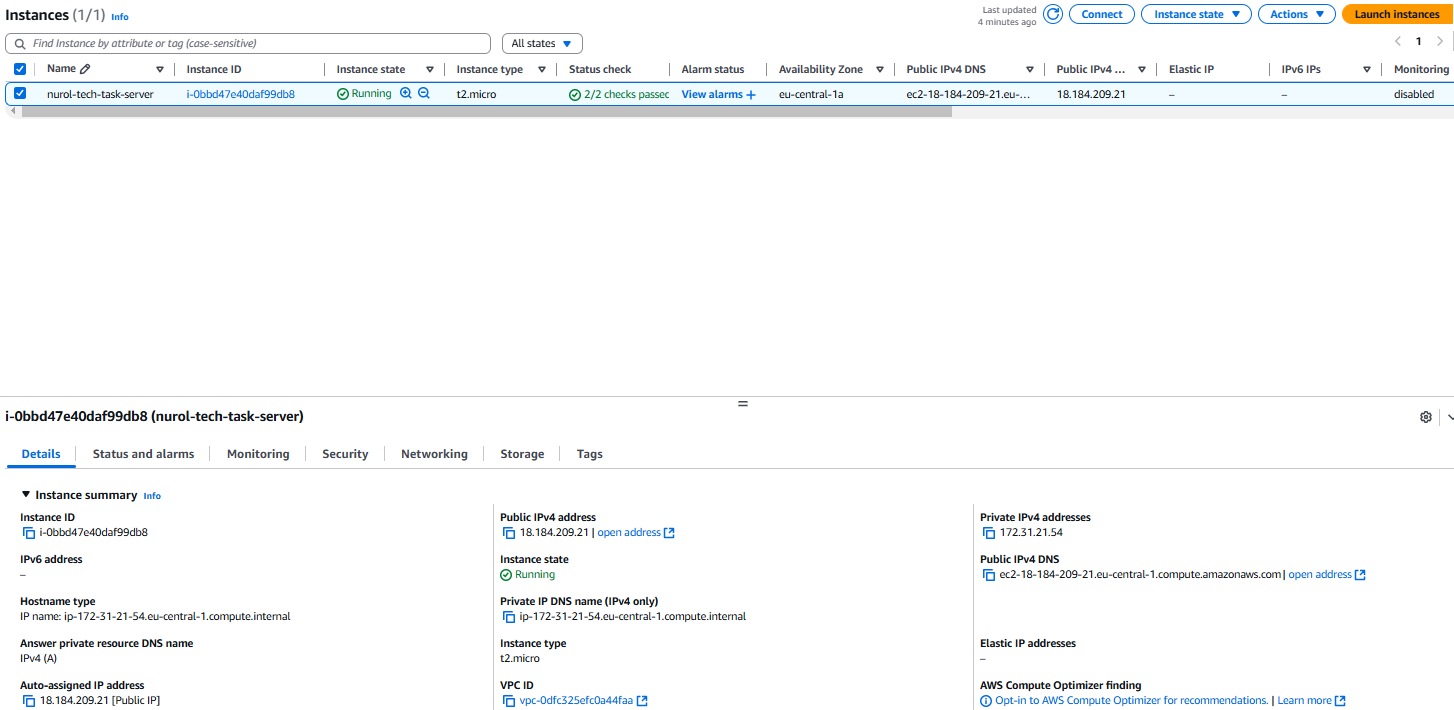
Description automatically generated

**.pem dosyası pc’ye iner.**

**.pem dosyası sunucuya pc üzerinden terminal ile bağlanmak için kullanılacaktır. Fakat sanal sunucuya browser üzerinden de bağlanılabilir.**

1. **Diğer seçenekler olduğu gibi bırakılır ve Launch Instance denilir.**
2. **Arama kutusuna EC2 yazılarak çalışan sanal sunucuya erişilebilir.**Graphical user interface, text, application, chat or text message

   Description automatically generated
3. **Instance-Id’ye tıklanarak sanal sunucu detayına ulaşılır.**

****

1. **Connect’e tıklayacak browser üzerinden sanal sunucuya ssh ile bağlanılabilir.**

**Sunucuyu dışarıya açma**

1. **EC2 servisi sayfasında sol taraftaki menüden Security groups’a tıklanır.**

**Text

Description automatically generated**

1. **Launch-wizard-1 ve Inbound rules seçilir.**

**Graphical user interface, text, application, email

Description automatically generated**

1. **Edit Inboud Rules butonuya gerekli değişiklikler yapılır.**

**Graphical user interface, application, Teams

Description automatically generated**

1. **Aşağıdaki seçenekler eklenir:**
2. **IP4 SSH TCP 22 0.0.0.0/0**
3. **IP4 HTTP TCP 80 0.0.0.0/0**
4. **IP4 HTTPS TCP 443 0.0.0.0/0**
5. **IP6 HTTPS TCP 443 ::/0**
6. **IP6 HTTP TCP 80 ::/0**

Graphical user interface

Description automatically generated with medium confidence

**Sunucuya Docker Servisi Kurulumu**

AWS tarafından free olarak sağlanan sunucu 1 CPU bulunmakta olup, kubernetes için minimum 2 CPU gerekmektedir. O nedenle docker container olarak servisleri ayağa kaldırıp devam ettim.

Docker kurulumu için aşağıdaki adımlar takip edilmelidir.

Sunucuya browser üzerinden ssh ile bağlandıktan sonra aşağıdaki komutlar sırayla çalıştırılır:

**sudo yum update -y**

**sudo amazon-linux-extras install docker**

**sudo yum install docker**

**sudo service docker start**

**sudo usermod -a -G docker ec2-user**

Çıkış yapılıp tekrar giriş yapılır:

**docker info**

ile kontrol edilir.

**Servislerin çalıştırılması**

Servislerin öncelikle AWS üzerinden pull edilebilmesi amacıyla projelerin localden docker hub repo’ya gönderilmesi için aşağıdaki adımlar takip edilir.

**docker login**

*-Docker hub authentication bilgileri girilir.*

**docker build -t <myusername>/tcp-producer-service-v0 .**

**docker push <your-dockerhub-username>/ tcp-producer-service-v0:latest**

Bu komutlar **tcp-consumer-service** için de tekrar edilir.

Daha sonrasında sunucuya tekrar dönüp aşağıdaki komut çalıştırılır.

**docker pull myusername/tcp-producer-service-v0:latest**

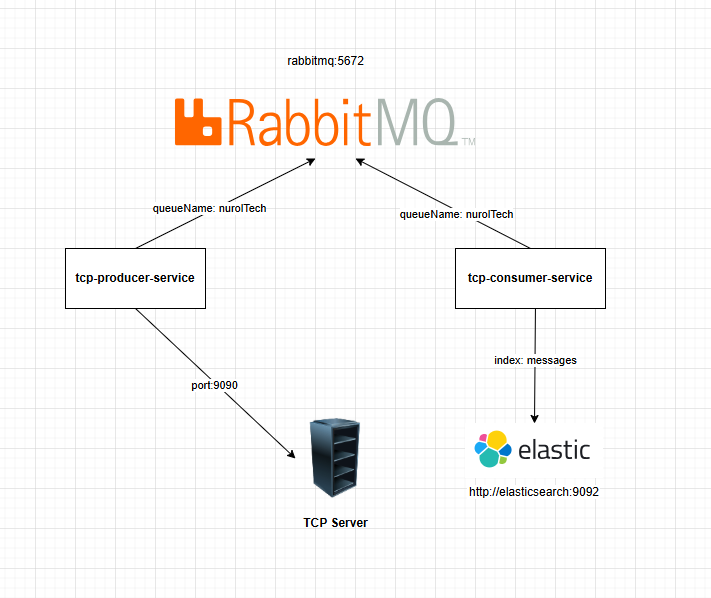
**docker pull myusername/tcp-consumer-service-v0:latest**

son adım olarak docker compose file içeriğinde bulunan servislerin çalıştırılması için sunucuya browser üzerinden ssh ile bağlandıktan sonra aşağıdaki komut çalıştırılır:

**docker compose up -d**

**Servis Topolojisi**

Aşağıdaki topolojide yer alan servisler ve görevleri şu şekilde özetlenmektedir.

****

**TCP Server:** Produce edilecek mesajların bulunduğu kaynak sunucu olarak görev yapmaktadır.

**tcp-producer-service:** TCP Server ile 9090 TCP portu üzerinden connection kurarak mesajları okumaktadır. Okuduğu mesajları bekletmeksizin rabbitmq üzerindeki “nurolTech” queue adresine publish etmektedir.

**tcp-consumer-service:** Rabbitmq’da yer alan “nurolTech” isimli queuedan okuduğu mesajları Elasticsearch servisindeki “messages” adlı index üzerine yazmaktadır.

**Rabbitmq:** TCP serverdan gelen mesajların gecikmeksizin sisteme aktarılmasını sağlayan message-queue yapısında çalışan bir sistem öğesidir.

**Elasticsearch:** Mesajların üzerinde kalıcı olarak saklandığı ve text sorgulama gibi işlemlerde oldukça başarılı olan güçlü indexleme algoritmalarına sahip bir sistemdir.