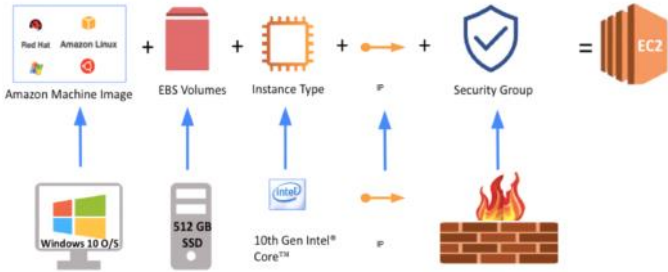


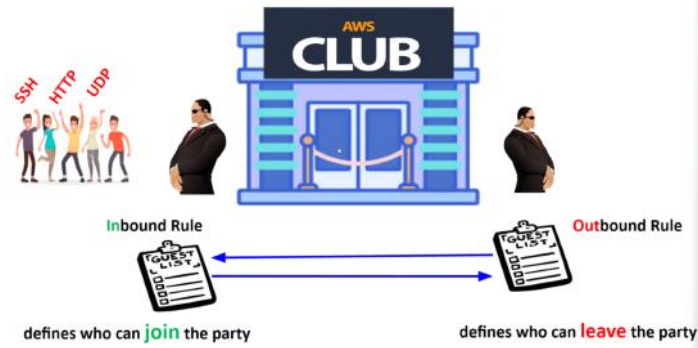
EC2 sanal bir bilgisayar.  
Server, EC2 sunucu, EC2, sanal sunucu gibi kelimelerle ifade edeceğiz.  
Instance ayaga kaldırmak diyince de ec2 dan bahsediyor olacağız.  
Ayaga kaldırmak, run etmek, launch etmek vs gibi kavramların hepsini burada kullanıyor olacağız.

# 1 EC2 Linux Hands-On Lab

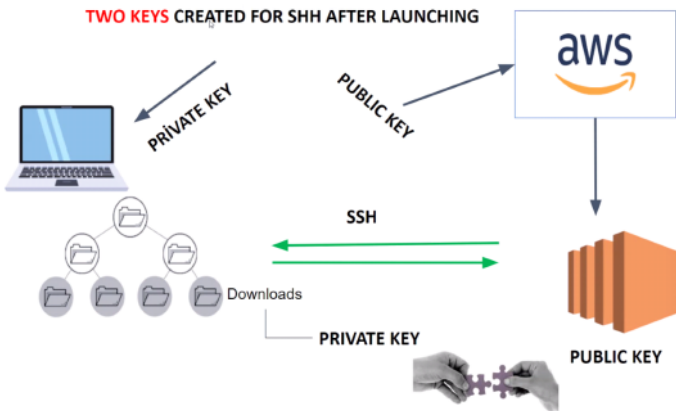
## Connect to your EC2 Instance



Elimizdeki bilgisayarla bunu hep kardiastiriyor olacağız.  
Isletim sistemi tum bilgisayarlar da vardır. Windows, linux vs bir tanesi mutlaka kullanilir oluyor.ec2 icinde de AMI da bunu gerceklestirmis olacağız. AMI = linux, windows,ubuntu vs Biz genel de amazon 23 AMI ni kullaniyor olacağız.  
Amazon 23 gectigimiz mart ayi itibariyle kullanima baslandı.  
AMI larin icine bir çok appler de kurulabiliyor ya da kurulmus halde hazır olarak customize edilmiş şekilde gelebilir.  
PC alırken normalde bazı bileşenler önemlidir. PC icinde disk mesele. Burada da bir harddiske ihtiyaç duyuyoruz. EC2lari attach edebilecegimiz EBS volumelerimiz olacak. Ucuz veya daha yenileyici SSD diskler de baglayabiliriz. Disardaki farklı markalardaki SSD ler gibi. Sectigimiz tipe gore fiyatlar artacak. Intel bir işlemciye sahip olmak genelde tercih edilen irseydir. Kac core işlemciye sahip olmak istiyorsak burada da bir işlemci secmis oluyoruz. Bununla birlikte RAM kac GB olsun kısmi ise burada instance tipidir. Instance tipi RAM ve CPU degerini kapsar.  
Network ihtiyaci burada da geçerli. Bir kimlik atamamız gerekir burada da public veya private. Virus duvarı olarakta burada da ec2 yu korumamız gerekir. EC2 nun bodyguardi diyebiliriz.



Ssh bağlantısı şart.  
Http portu seçmeye ihtiyacımız var. Internet tarafından herkesin katılabilmesi için bu portu da seçiyoruz.  
EC2ya bağlanabilmek için bazı inbound rulelara ihtiyacımız var. Bunu belirledigimiz yerlerde security gruplardir. EC2 lara security grup attach ederek guveligi arttiriyoruz  
Outbound rules için default olarak all traffic atanir normalde.  
Simdilik sadece inbound rule için SSH portunun acik olması gerektigini bilmeliyiz.



Aws public key atiyor ama biz kendimiz private key oluşturmamız ve saklamamız gerek. Kendi key pairimiz yoksa bu ec2 ye bağlanabilmemiz çok mümkün değil.  
Private ve public eslesmeli ki ec2 ya bağlanabillelim.  
Downloada atıp orada saklamış olacağız daha sonra kullanmak için.  
Open ssh için .pem kullanıyoruz

Go the directory of folder where your key.pem is located

## Connect

Linux/Mac

```
cd Downloads
ls
firstkey.pem
```

Git Bash  
Windows 10 Powershell

```
cd \Downloads\
ls
firstkey.pem
```

WSL

```
cd /mnt/c/Users/paul/Downloads
ls
firstkey.pem
cp firstkey.pem ~
cd (go to home directory)
ls
firstkey.pem
```

```
chmod 400 firstkey.pem
```

---No chmod 400---

```
chmod 400 firstkey.pem
```

```
ssh -i firstkey.pem ec2-user@publicIP of Instance
```

BAGLANMA KILAVUZU BU SEKILDE

## Launch instance

My AMIs Quick Start

Amazon Linux macOS Ubuntu Windows Red Hat SUSE Linux

Amazon Machine Image (AMI)

Amazon Linux 2023 AMI  
ami-08a0d1e16fc3f51ea (x86\_64, uefi-preferred) / ami-08a0d1e16fc3f51ea (x86\_64, uefi)  
Virtualization: hvm ENA enabled: true Root device type: ebs

Description  
Amazon Linux 2023 AMI 2023.4.20240611.0 x86\_64 HVM kernel-6.1

Architecture: 64-bit (x86) Boot mode: uefi-preferred AMI ID: ami-08a0d1e16fc3f51ea

Instance type: t2.micro

AMI ID kismini CloudFormation için kullanacağız.  
Bazı resouceları ayaga kaldırırken bizden Jason ya da Yamldosyası isteyecek. Bu durumda launch edeceğimiz EC2 nasıl bir AMI sahip olmasına karar verirken bu AMI ID yi kullanıyor olacağız.  
AMI IDler genelde unik. Amazon linuxun kendi bir Idsi, Ubuntunun kendi vs diye tek tek herbirisinin kendine özel bir AMI Idsi var.

Instance tipi için dikkat etmemiz gereken iki şey var. CPU ve MEMORY!!  
Free tier olması sebebiyle biz t2 micro ile devam edeceğiz.  
Markette en çok m5 kullanılır.

### ▼ Key pair (login) [Info](#)

You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.

Key pair name - required

Select

[Create new key pair](#)

Instancea secure bir sekilde connet olabilmemiz icin bir tane key paire ihtiyacimiz var. (public ve private)

Launchtan sonra ssh icin iki tane key pair/key olusturuluyor.

AWS ec2nun icine direkt bir public key koyuyor. Daha sonra biz ec2 launch ederken ssh baglantisi icin ihtiyac olan keyimizi olusturuyoruz.

Private keyimizin bizde olmasi gerekiyor. Private key yoksa ec2 ya baglanamzsiniz.

Pem uzantili bir key.

Bunu bizim localizde tutmamiz gerekiyor.

Bunlar yapboz parçasi gibilerdir public ve private birbirini tamamlamali.

Encrypt algoritmasi icin bu konu uzerine okuma yapmaliyiz.

Microsoft icindeli farkli uygulamalar icin ppk kullaniyoruz bazen.

### ▼ Network settings [Info](#)

[Edit](#)

Network [Info](#)

vpc-0735d80b056875a54

Subnet [Info](#)

No preference (Default subnet in any availability zone)

Auto-assign public IP [Info](#)

Enable

[Additional charges apply](#) when outside of [free tier allowance](#)

Firewall (security groups) [Info](#)

A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

☒ Create security group

☐ Select existing security group

We'll create a new security group called **launch-wizard-2** with the following rules:

☒ Allow SSH traffic from

Helps you connect to your instance

Anywhere

0.0.0.0/0

☐ Allow HTTPS traffic from the internet

To set up an endpoint, for example when creating a web server

☐ Allow HTTP traffic from the internet

To set up an endpoint, for example when creating a web server

**Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.**

Burada eger HTTPS ve HTTP tragine izin verirse AWS default

olarak launch-wizard-2 adinda bir security group attach

edeceginin teklifini yapmis. Fakat biz kendimiz bir security group

olusturacagiz. Cunku security gruplarimizin copluge donmesini

istemiyoruz, neyin ne oldugunu bilemiyoruz. Biz kendimiz 22 ve

80 portunun acik oldugu bir grup olusturup bu belirsizlikleri

onlemis olacagiz.

Security kismini ilerde daha detaylica isleyecgiz.

### ▼ Configure storage [Info](#)

[Advanced](#)

1x 8 GiB gp3 Root volume (Not encrypted)

**Free tier eligible customers can get up to 30 GiB of EBS General Purpose (SSD) or Magnetic storage**

[Add new volume](#)

[Click refresh to view backup information](#)

The tags that you assign determine whether the instance will be backed up by any Data Lifecycle Manager policies.

0 x File systems

[Edit](#)

Burayi da daha sonra detaylica isleyecgiz.

Burada bir EBS Volume seciyoruz.

Gp2 ve gp3 free tier destekli, secebiliriz. Fakat gp3 daha performansli.

### ► Advanced details [Info](#)

Bu kısmi detaylica daha sonra isleyecegiz

**▼ Summary**

Number of instances [Info](#)

1

Software Image (AMI)  
Amazon Linux 2023 AMI 2023.4.2...[read more](#)  
ami-08a0d1e16c3f61ea

Virtual server type (instance type)  
t2.micro

Firewall (security group)  
New security group

Storage (volumes)  
1 volume(s) - 8 GiB

**Free tier:** In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 750 hours of public IPv4 address usage per month, 30 GiB of EBS storage, 2 million I/Os, 1 GB of snapshots, and 100 GB of bandwidth to the internet.

Cancel **Launch instance** [Review commands](#)

Burada summary kismini check edip instancemizi launch ediyoruz artik.

```
MINGW64/c/Users/seyma/Dt x + v
seyma@Seymanurkara MINGW64 ~
$ pwd
/c/Users/seyma
seyma@Seymanurkara MINGW64 ~
$ cd Downloads/
seyma@Seymanurkara MINGW64 ~/Downloads
$ ll | grep pem
-rw-r--r-- 1 seyma 197609      1674 jun 22 10:15 firstkeypair.pem
-rw-r--r-- 1 seyma 197609      1678 des  5 2022 new.key.pem
seyma@Seymanurkara MINGW64 ~/Downloads
$
```

Masaustunde git bash acip oradan c/user ve sonra downloada gidip keyimizin orda olup olmadigina baktik burada ama eger downloadda key pairimizin oldugunu biliyorsak buna mecbur degiliz.

Instances (1/1) [Info](#)

[All states](#)

[Find](#) [Clear filters](#)

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4
paul-first-instance	i-06b0b979c6fc6b0a	Running	t2.micro	2/2 checks passed	<a href="#">View alarms</a>	us-east-1d	ec2-44-201-130-222.co...	44.201.130.222

Daha sonra consolea tekrar donup instancemizi secip sag ustteki connect kismina basiyoruz.

**EC2 Instance Connect** | Session Manager | SSH client | EC2 serial console

Instance ID  
i-0f37001eff203762e (seymanurkara)

Connection Type

☒ Connect using EC2 Instance Connect  
Connect using the EC2 Instance Connect browser-based client, with a public IPv4 address.

☐ Connect using EC2 Instance Connect Endpoint  
Connect using the EC2 Instance Connect browser-based client, with a private IPv4 address and a VPC endpoint.

Public IP address  
54.147.148.65

Username  
Enter the username defined in the AMI used to launch the instance. If you didn't define a custom username, use the default username, ec2-user.

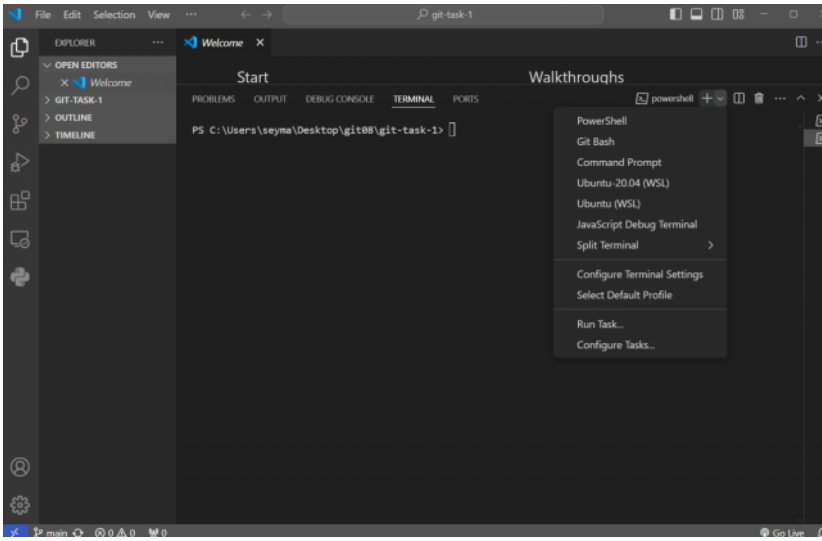
ec2-user

**Note:** In most cases, the default username, ec2-user, is correct. However, read your AMI usage instructions to check if the AMI owner has changed the default AMI username.

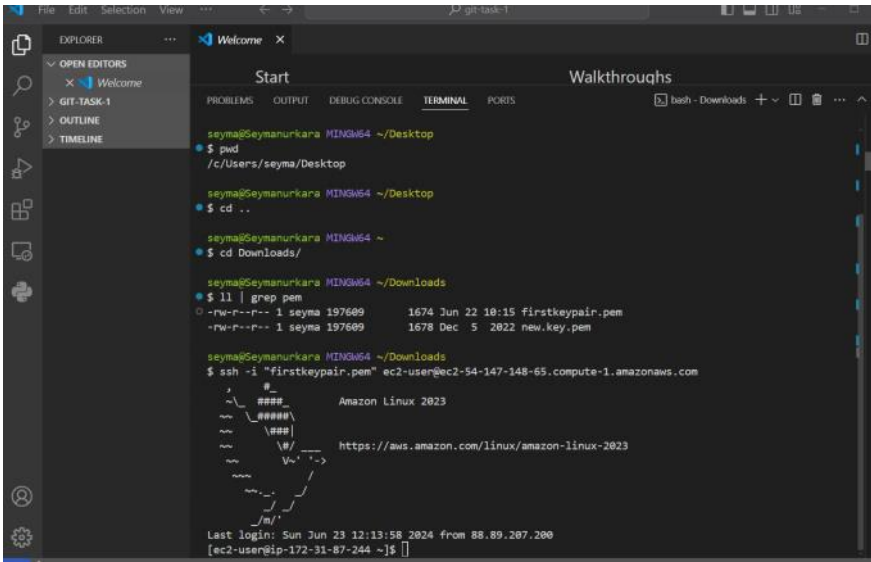
Cancel **Connect**

Buradan connect dedigimizde bizi kendi ec2 muzu yonlendirir. Yeni bir pencere acar. Ve kartal resmi karsimiza cikmis olur. Biz bunu bir de kendi BASH terminalimizde acacagiz.

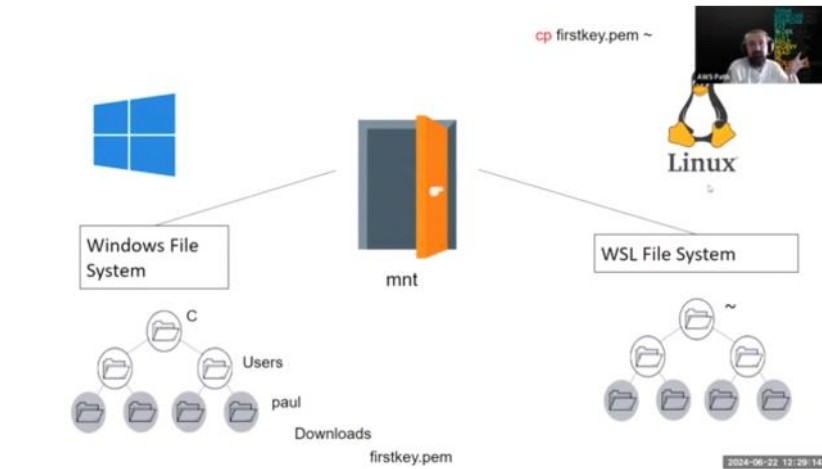




Simdi de vs code uzerinde bunu acacagiz.  
VS Code acina yukaridan terminale gidip new terminal diyoruz ve ekrandaki yere tiklayip Git Bash seciyoruz  
Burada da ayni deksupta baglandigimizda pemi kontrol etmek icin downloadsa gittigimiz gibi buradan da cd downloads yapip pemimizi kontrol ediyoruz.  
Ayni sekilde konsola gidip connect kisminda SSH Clientinden ssh -i baglantisini buraya kopyalayıp burada da baglanmis oluyoruz.



Sonuc Boyle Olmalı.



WSL tarafinin dosya sistemi linux dosya sistemi. Windowsta mesela c userinin altiinda download icindeydi. Ama WSL icin bu keyi alip WSL file systeminin icine almam lazim yoksa baglanti gerceklemez.

Last modified: Saturday, 22 June 2024, 11:22 PM





```
config x
C: > Users > seyma > .ssh > config
1 Host Seymanur_ubuntu
2   HostName ec2-54-147-148-65.compute-1.amazonaws.com
3   IdentityFile ~/.ssh/seymanur.pem
4   User ubuntu
5
6
7
8 Host Seymanur_linux
9   HostName ec2-54-147-148-65.compute-1.amazonaws.com
10  IdentityFile ~/.ssh/seymanur.pem
11  User ec2-user
12
13
14 Host *
15   ServerAliveInterval 30000
16   ServerAliveCountMax 2
```

Burada config kismini guncelliyoruz.  
Isim kismina istediginizi yazabilirsiniz.  
Hostname kismina ise konsolda ssh clientta public DNS kismini yapistiriyoruz.

```
Host Seymanur_ec2server
HostName ec2-54-147-148-65.compute-1.amazonaws.com
IdentityFile ~/Downloads/firstkeypair.pem
User ec2-user
```

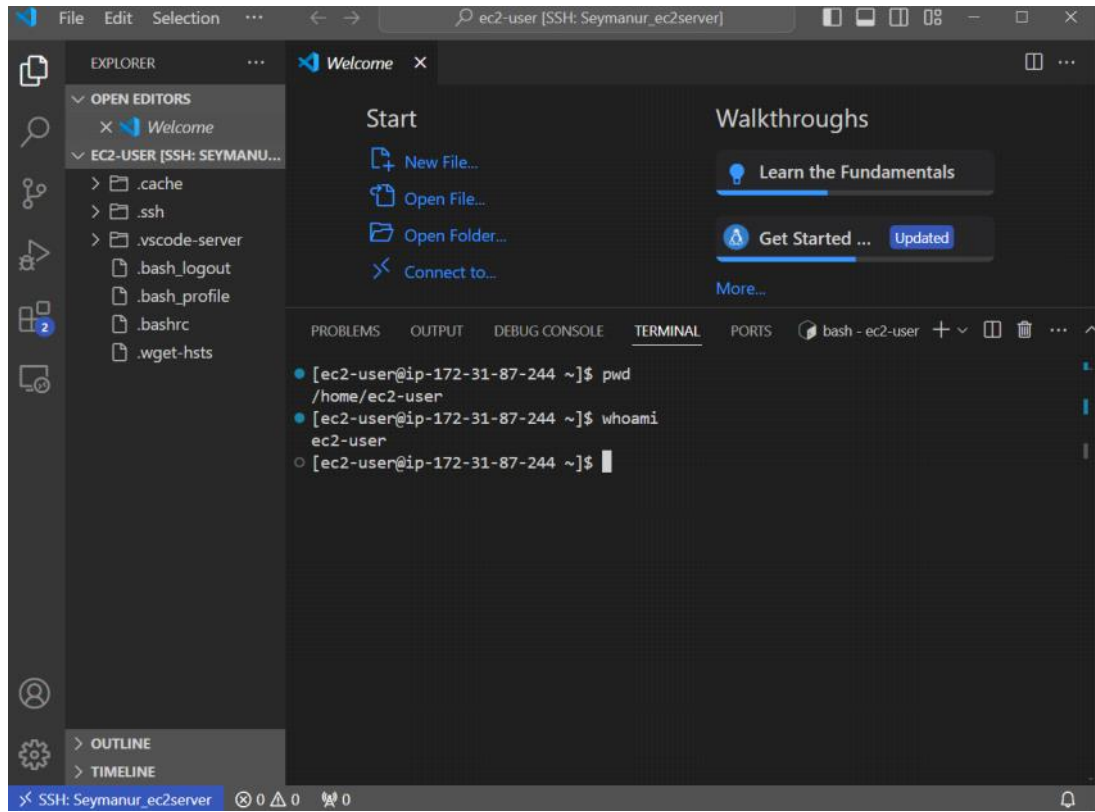
Burada tekrar bir guncelleme yapiyoruz pemimizin oldugu yer ve bilgiler dogrultusunda

```
File Edit Selection View
Select configured SSH host or enter user@host
config x
C: > Users > seyma > .ssh >
1 Host Seymanur_ubuntu
2   HostName ec2-54-147-148-65.compute-1.amazonaws.com
3   IdentityFile ~/.ssh/seymanur.pem
4   User ubuntu
5
6
7
8 Host Seymanur_ec2server
9   HostName ec2-54-147-148-65.compute-1.amazonaws.com
10  IdentityFile ~/Downloads/firstkeypair.pem
11  User ec2-user
12
13
14 Host *
15   ServerAliveInterval 30000
16   ServerAliveCountMax 2
```

Daha sonra tekrar sol alttaki mavi yere basil oradan connect to host diyip az once Seymanur\_ec2server diye adlandirdigimiz tyere baglaniyoruz. O da bizi yeni bir vs code a yonlendiriyor.

Bu islemi her yeni bir servver actikca hostname deki uzantiyi DNS yeni haliyle guncelleyip tekrar etmemiz gerek.





Baglanti sirasinda linux secenegini secip baglantiyi saglamis olduk.  
Terminalde artik bi daha bi daha konsola girip ssh clienta gitmemize gerek kalmadi buradan direkt baglnti yapiyoruz.  
En sik tercih edilen bu oluyor  
Cok daha flexible