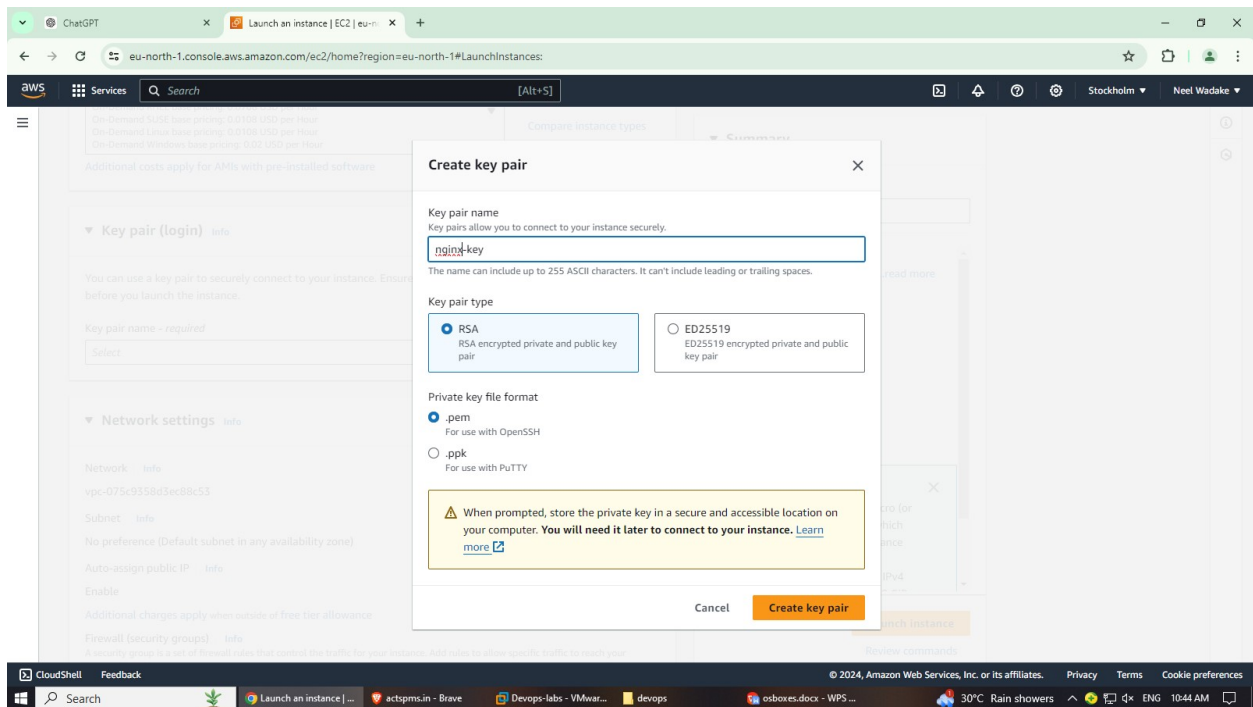


# AWS

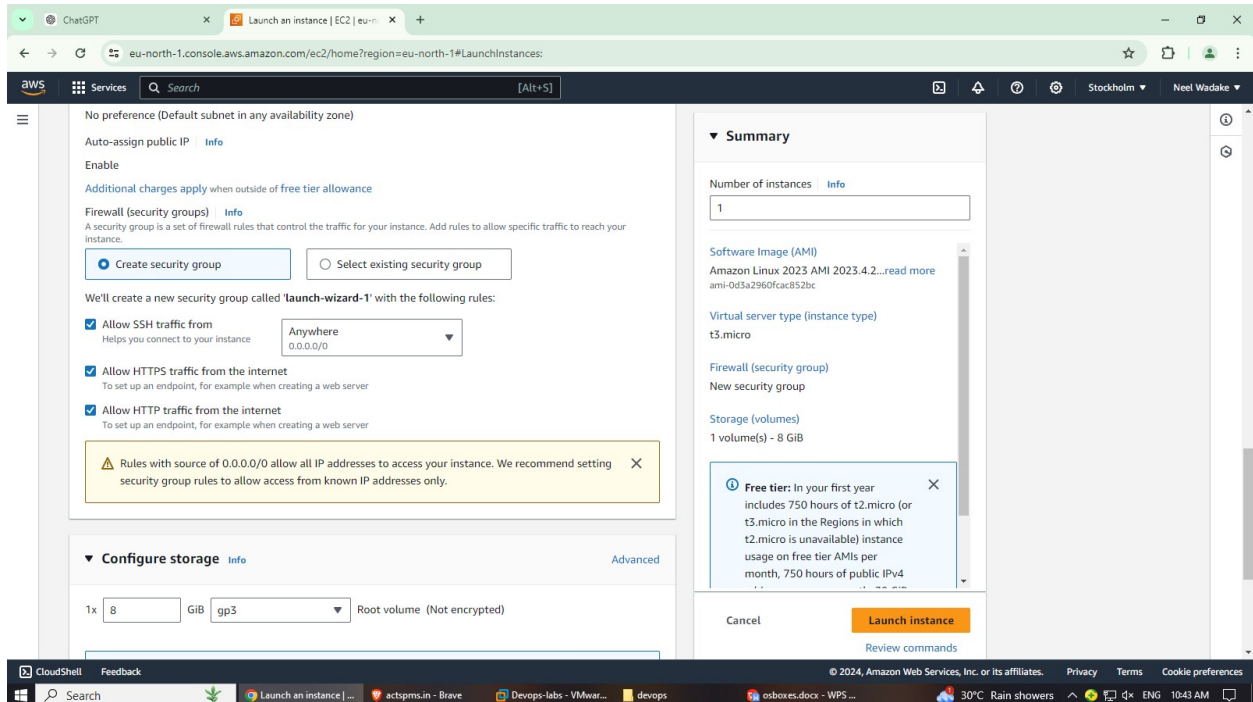
## 1. EC2

This screenshot shows the AWS Management Console for the eu-north-1 region. The left sidebar contains navigation links for EC2 Dashboard, EC2 Global View, Events, Instances, Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations, Images, AMIs, AMI Catalog, Elastic Block Store, Volumes, Snapshots, Lifecycle Manager, Network & Security, and Security Groups. The main content area is titled 'Resources' and shows a summary of EC2 resources in the Europe (Stockholm) Region. The summary includes: Instances (running) 0, Auto Scaling Groups 0, Dedicated Hosts 0, Elastic IPs 0, Instances 0, Key pairs 0, Load balancers 0, Placement groups 0, Security groups 1, Snapshots 0, and Volumes 0. Below the summary, there are sections for 'Launch instance' (with a 'Launch instance' button and a 'Migrate a server' button), 'Service health' (showing AWS Health Dashboard and a status of 'This service is operating normally'), 'Instance alarms' (with a 'View in CloudWatch' button), and 'Zones' (showing Zone name and Zone ID). On the right, there is a 'EC2 Free Tier' section showing 0 free tier offers in use, and an 'Account attributes' section showing the Default VPC and Settings.

This screenshot shows the 'Launch an instance' page in the AWS Management Console. The page is titled 'Launch an instance' and provides a summary of the instance configuration. The 'Summary' section includes: Number of instances 1, Software Image (AMI) Amazon Linux 2023 AMI 2023.4.2..., Virtual server type (instance type) t3.micro, Firewall (security group) New security group, and Storage (volumes) 1 volume(s) - 8 GiB. A 'Free tier' notification is displayed, stating: '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'. The 'Name and tags' section shows the instance name 'nginx'. The 'Application and OS Images (Amazon Machine Image)' section provides a search bar and a 'Quick Start' section with buttons for Amazon Linux, macOS, Ubuntu, Windows, Red Hat, and SUSE Linux. The bottom of the page features a 'Launch instance' button and a 'Review commands' link.



The key is download



## Now launch instance

The screenshot shows the 'Launch an instance' page in the AWS Management Console. A green success banner at the top states: 'Successfully initiated launch of instance (i-0937fdf1ca51d8dc7)'. Below this is a 'Launch log' section. The 'Next Steps' section contains a search bar and several recommended actions:

- Create billing and free tier usage alerts**: To manage costs and avoid surprise bills, set up email notifications for billing and free tier usage thresholds. [Create billing alerts](#)
- Connect to your instance**: Once your instance is running, log into it from your local computer. [Connect to instance](#) [Learn more](#)
- Connect an RDS database**: Configure the connection between an EC2 instance and a database to allow traffic flow between them. [Connect an RDS database](#) [Create a new RDS database](#) [Learn more](#)
- Create EBS snapshot policy**: Create a policy that automates the creation, retention, and deletion of EBS snapshots. [Create EBS snapshot policy](#)
- Manage detailed monitoring**
- Create Load Balancer**
- Create AWS budget**
- Manage CloudWatch alarms**

The bottom of the console shows the taskbar with various applications like CloudShell, DevOps-labs, and a weather widget indicating 30°C and rain showers.

The screenshot shows the 'Instances' page in the AWS Management Console. The left sidebar contains navigation links for EC2 Dashboard, EC2 Global View, Events, Instances, Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations, Images, AMIs, AMI Catalog, Elastic Block Store, Volumes, Snapshots, Lifecycle Manager, Network & Security, Security Groups, and Elastic IP Addresses.

The main content area displays a table of instances:

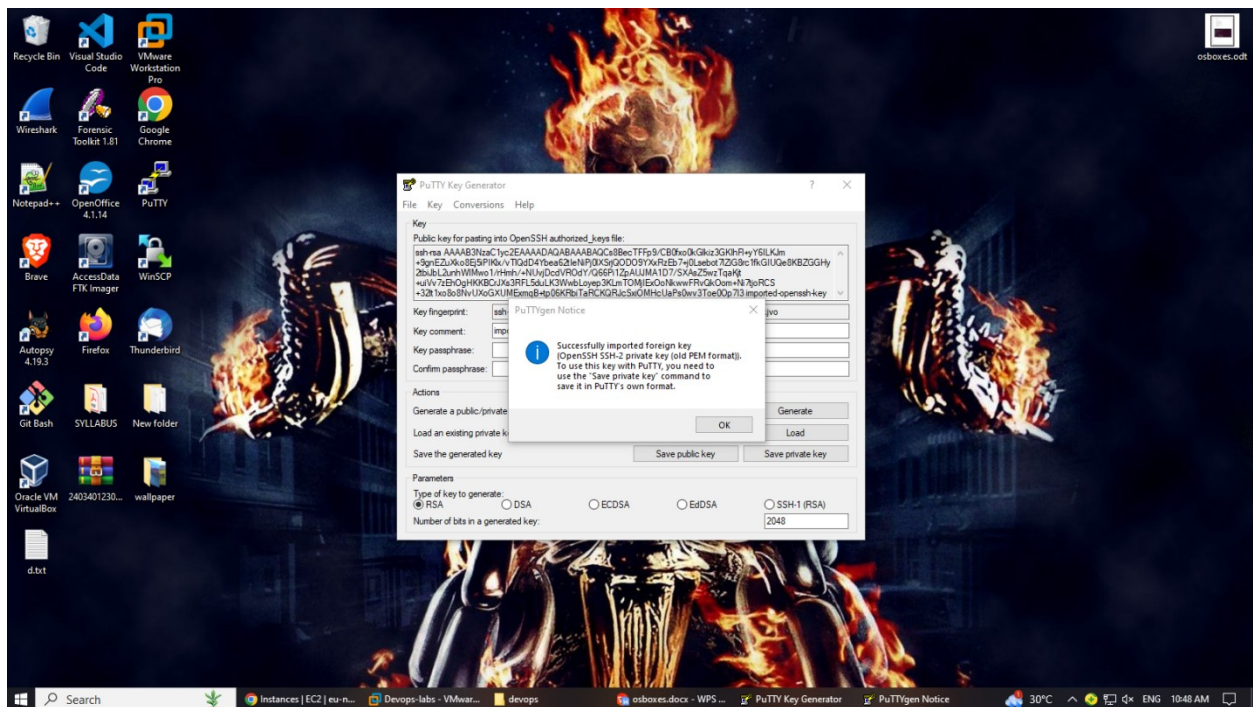
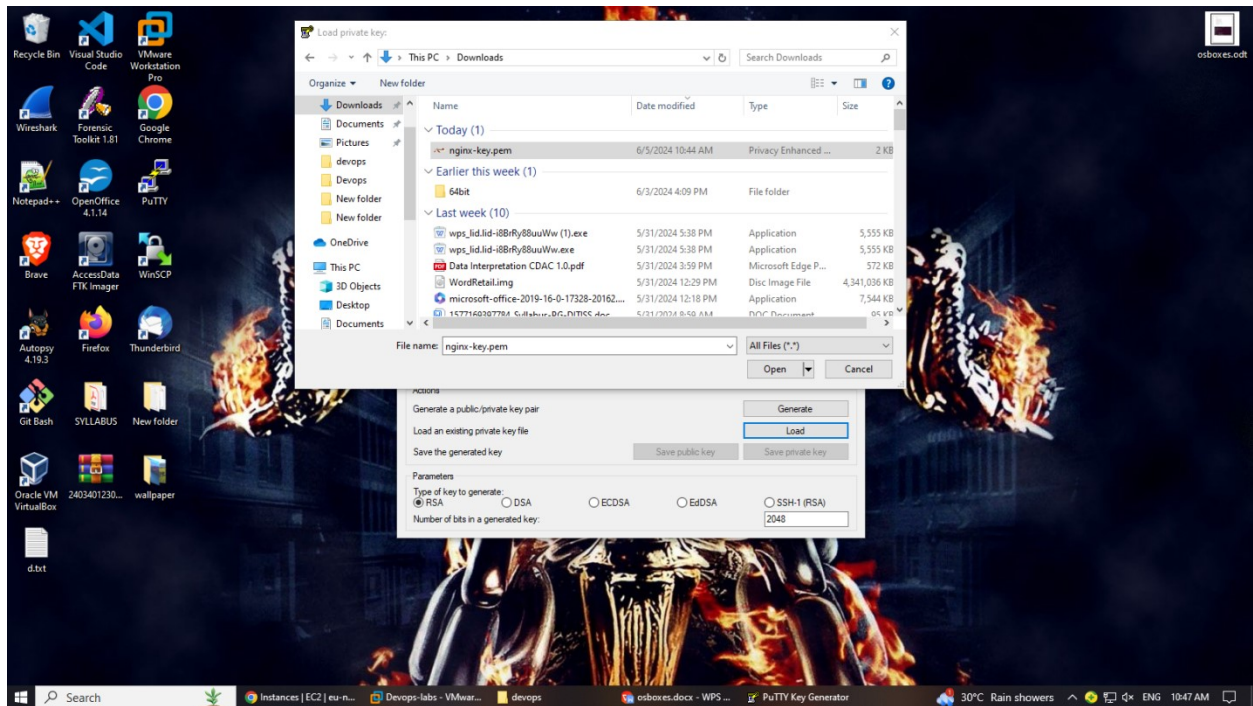
Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS
nginx	i-0937fdf1ca51d8dc7	Running	t3.micro	Initializing	View alarms	eu-north-1b	ec2-13-51-198-134...

Below the table, the details for instance **i-0937fdf1ca51d8dc7 (nginx)** are shown:

- Instance summary**
  - Instance ID: i-0937fdf1ca51d8dc7 (nginx)
  - IPv6 address: -
  - Hostname type: IP name: ip-172-31-38-45.eu-north-1.compute.internal
- Public IPv4 address**: 13.51.198.134 | [open address](#)
- Instance state**: Running
- Private IPv4 addresses**: 172.31.38.45
- Public IPv4 DNS**: ec2-13-51-198-134.eu-north-1.compute.amazonaws.com | [open address](#)
- Private IP DNS name (IPv4 only)**: ip-172-31-38-45.eu-north-1.compute.internal

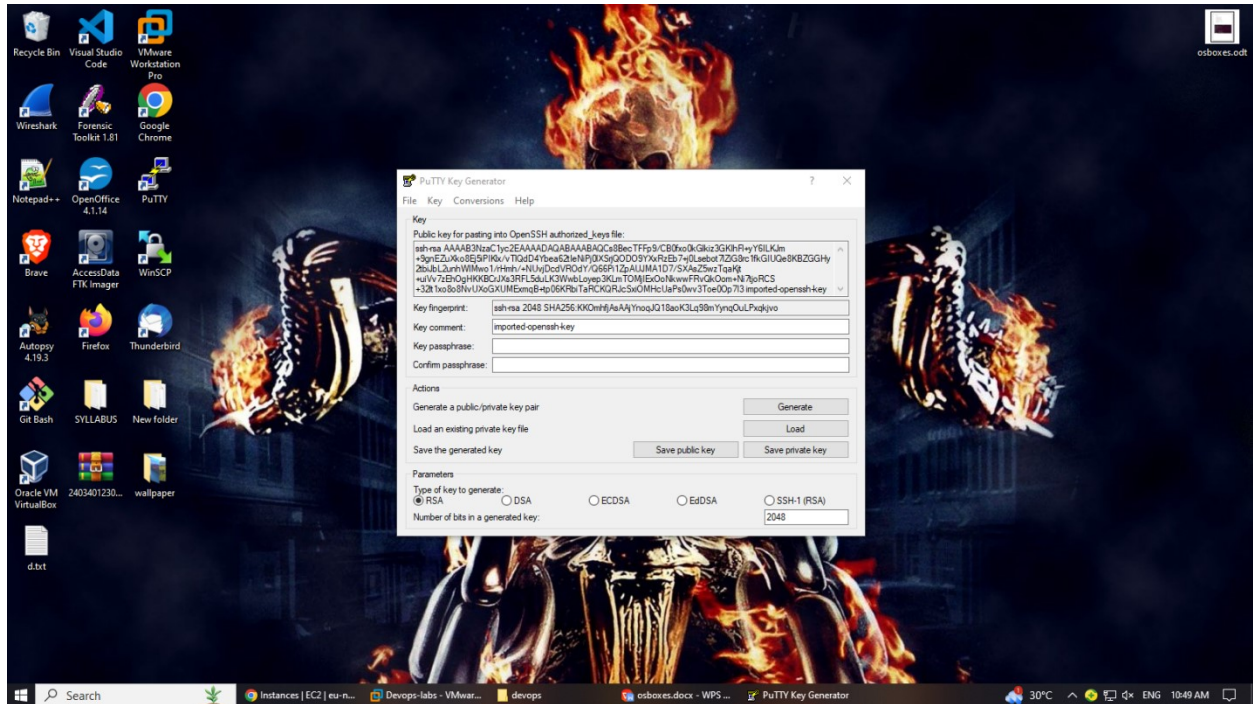
The bottom of the console shows the taskbar with various applications like CloudShell, DevOps-labs, and a weather widget indicating 30°C and rain showers.

Now go to the putty gen and load and select the key which downloaded

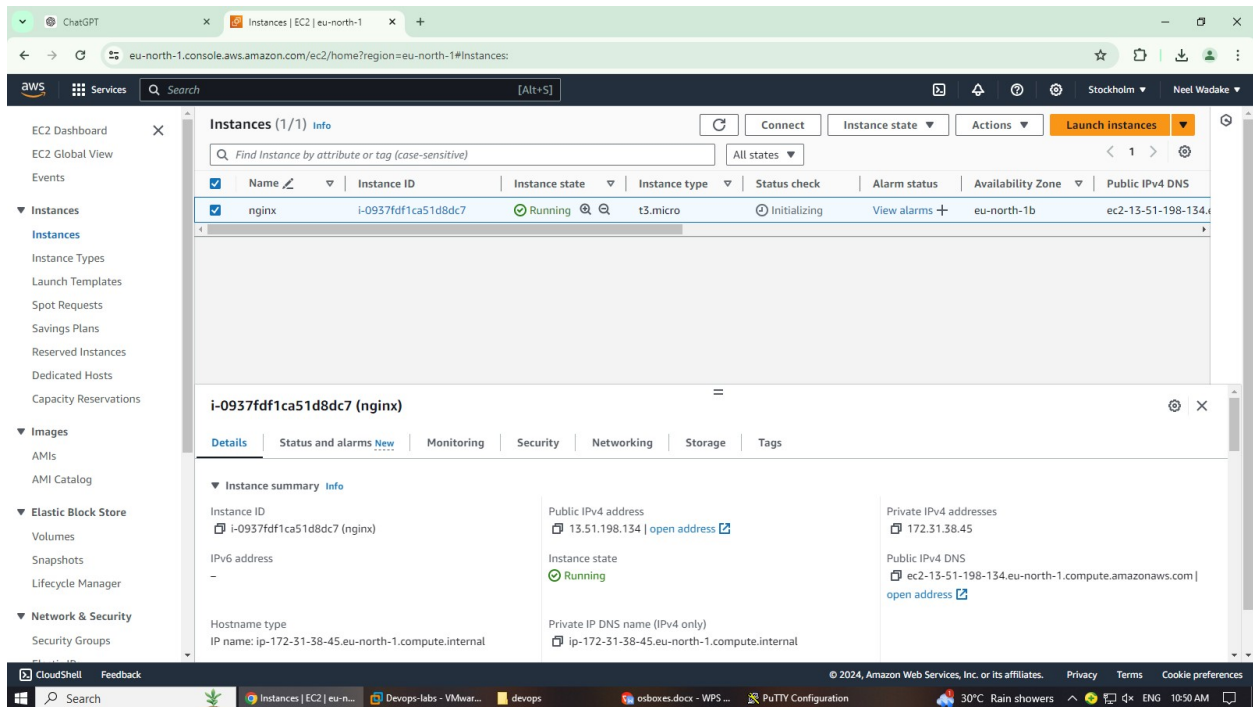




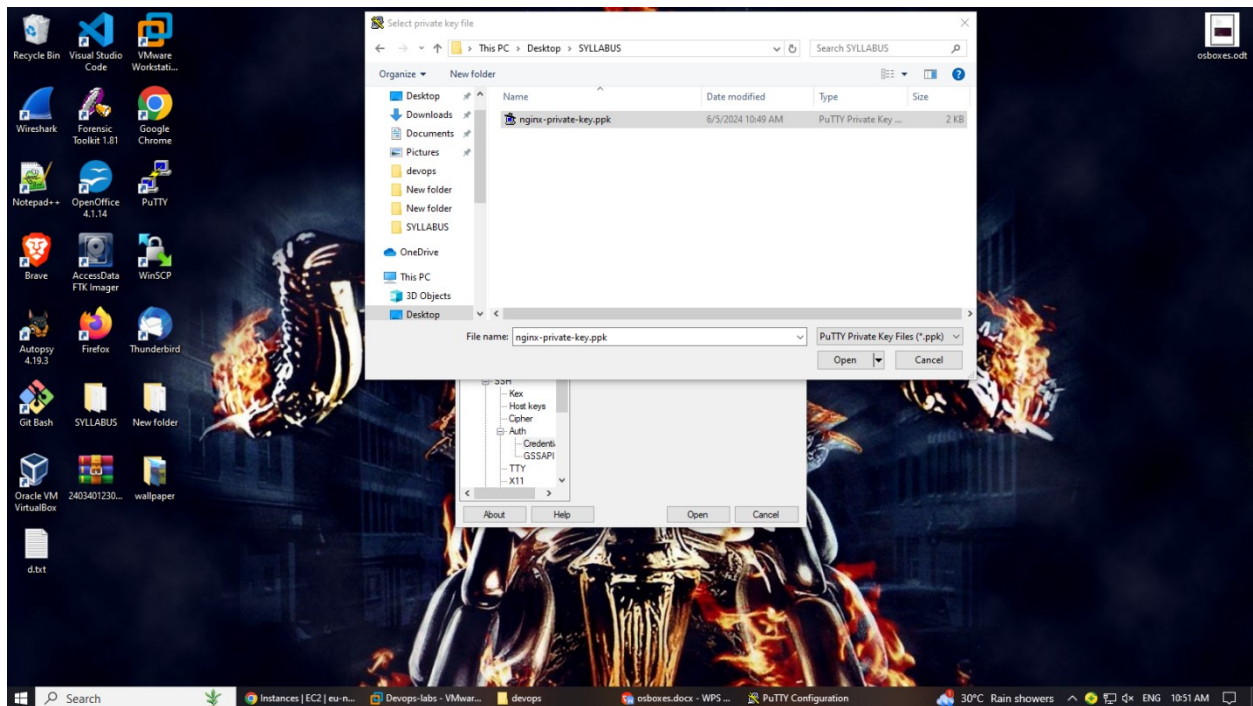
Select the private key and save it

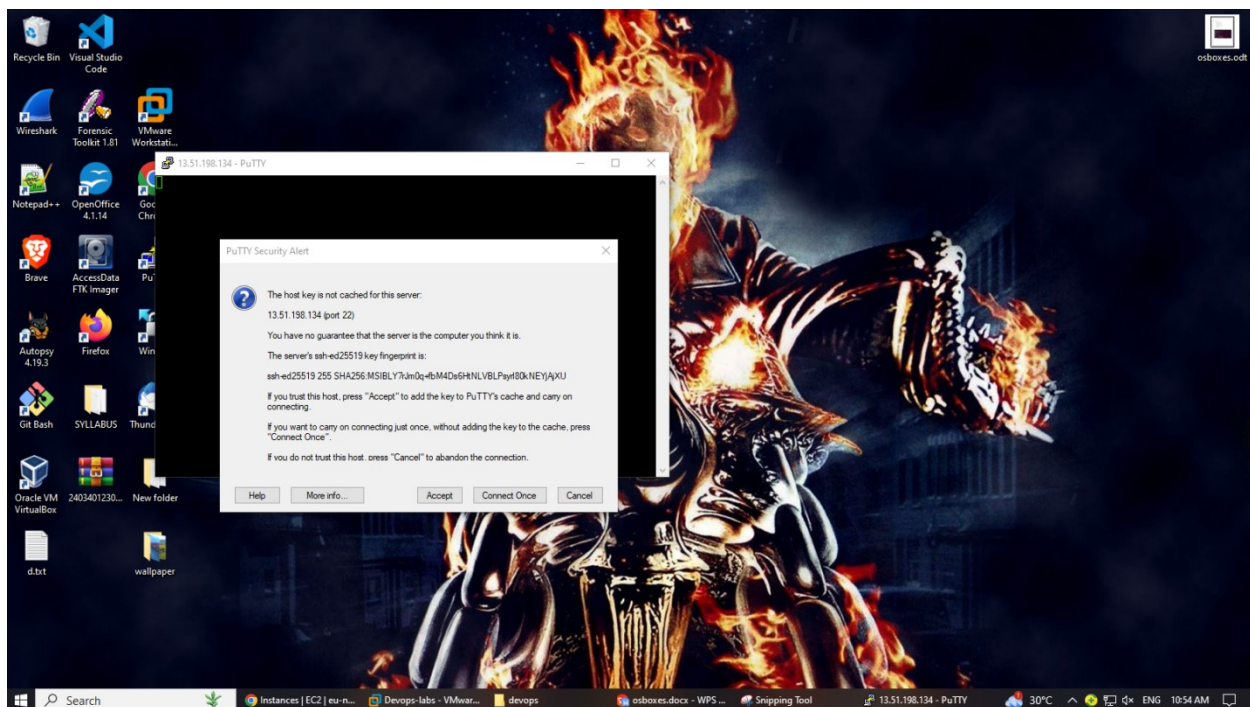
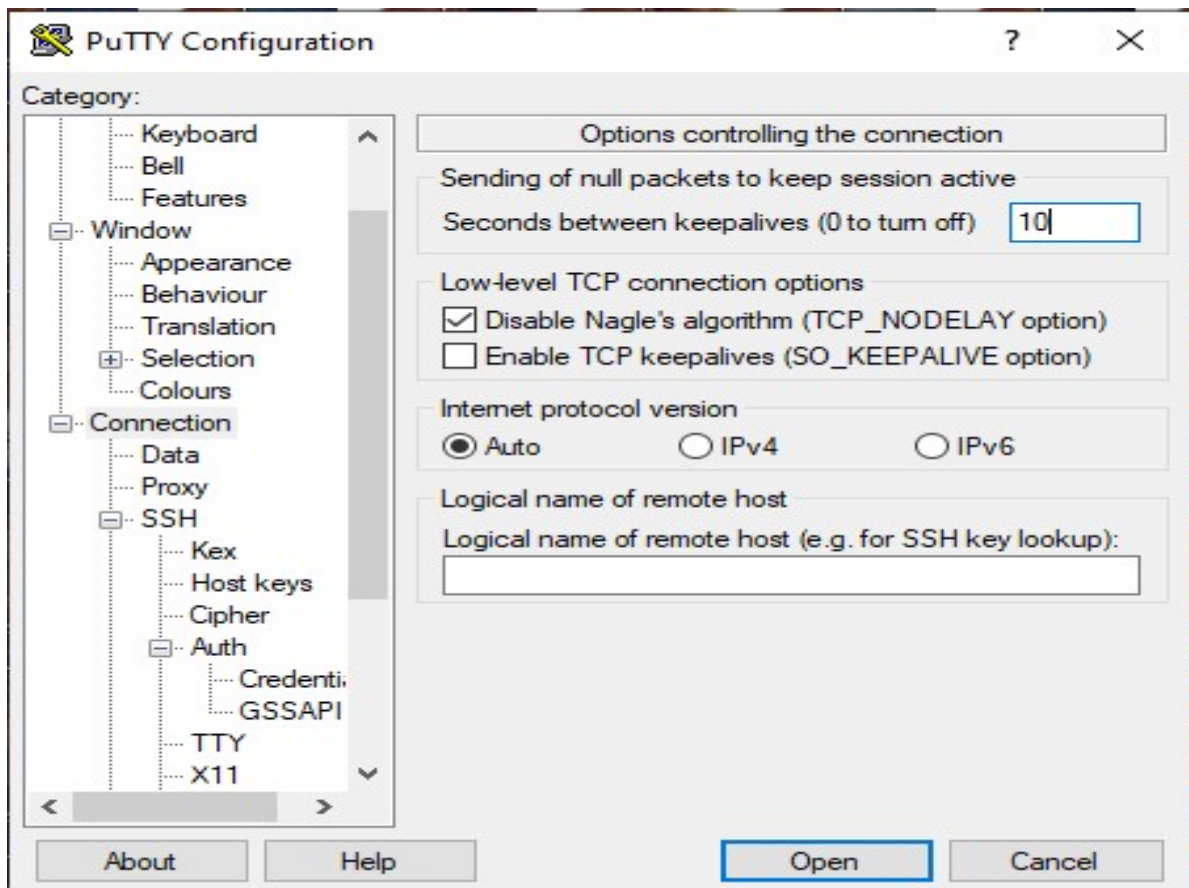


Now go to the putty



Now go to the SSh--> Auth-->private key and select it







1. Sudo yum update -y
2. Sudo yum install docker -y
3. Sudo systemctl start docker
4. Sudo docker pull nginx:latest
5. Sudo nano welcome.html
6. sudo docker run -d -p 80:80 --name nginx-webpage -v \$(pwd)/message.html:/usr/share/nginx/html/index.html nginx:latest

```
ec2-user@ip-172-31-38-45:~$ sudo yum update -y
Installing : libgroup-3.0-1.amzn2023.0.1.x86_64 8/10
Running scriptlet: docker-25.0.3-1.amzn2023.0.1.x86_64 10/10
Installing : docker-25.0.3-1.amzn2023.0.1.x86_64 10/10
Running scriptlet: iptables-libs-1.8.8-3.amzn2023.0.2.x86_64 10/10
Created symlink /etc/systemd/system/socket.target.wants/docker.socket -> /usr/lib/systemd/system/docker.socket.
Verifying : containerd-1.7.11-1.amzn2023.0.1.x86_64 1/10
Verifying : docker-25.0.3-1.amzn2023.0.1.x86_64 2/10
Verifying : iptables-libs-1.8.8-3.amzn2023.0.2.x86_64 3/10
Verifying : iptables-nft-1.0.8-3.amzn2023.0.2.x86_64 4/10
Verifying : libgroup-3.0-1.amzn2023.0.1.x86_64 5/10
Verifying : libnetfilter_conntrack-1.0.8-2.amzn2023.0.2.x86_64 6/10
Verifying : libnetfilter_log-1.0.1-19.amzn2023.0.2.x86_64 7/10
Verifying : libnftnl-1.2.2-2.amzn2023.0.2.x86_64 8/10
Verifying : pigz-2.5-1.amzn2023.0.3.x86_64 9/10
Verifying : runc-1.1.11-1.amzn2023.0.1.x86_64 10/10

Installed:
containerd-1.7.11-1.amzn2023.0.1.x86_64 docker-25.0.3-1.amzn2023.0.1.x86_64
libnetfilter_conntrack-1.0.8-2.amzn2023.0.2.x86_64 iptables-libs-1.8.8-3.amzn2023.0.2.x86_64
libnftnl-1.2.2-2.amzn2023.0.2.x86_64 libnftnl-1.2.2-2.amzn2023.0.2.x86_64
libnetfilter_log-1.0.1-19.amzn2023.0.2.x86_64 pigz-2.5-1.amzn2023.0.3.x86_64
runc-1.1.11-1.amzn2023.0.1.x86_64

Complete!
[ec2-user@ip-172-31-38-45 ~]$ sudo systemctl start docker
[ec2-user@ip-172-31-38-45 ~]$ sudo systemctl enable docker
Created symlink /etc/systemd/system/multi-user.target.wants/docker.service -> /usr/lib/systemd/system/docker.service.
[ec2-user@ip-172-31-38-45 ~]$ sudo docker pull nginx:latest
latest: Pulling from library/nginx
069366bb150: Pull complete
528e0722481: Pull complete
069366bb150: Pull complete
b7910a8c4316: Pull complete
069366bb150: Pull complete
b7910a8c4316: Pull complete
Digest: sha256:10f04e4f6e4a314bf31d8b08d8e5b0c9b1f0c4258b0d6e45f6c4d18a0c0a30d
Status: Downloaded newer image for nginx:latest
docker.io/library/nginx:latest
[ec2-user@ip-172-31-38-45 ~]$ sudo nano welcome.html
[ec2-user@ip-172-31-38-45 ~]$ sudo docker run -d -p 80:80 --name my-nginx -v $(PWD)/welcome.html:/usr/share/nginx/html/index.html nginx:latest
sha1: PWD: command not found
c81a17b1f0c0a419b81b0c7f2f2f710b551c9a413b19314e5173e0dded131ad
docker: Error response from daemon: failed to create task for container: failed to create shim task: OCI runtime create failed: runc create failed: unable to start container process: error during container init: error mounting "/welcome.html" to rootfs at "/usr/share/nginx/html/index.html": mount: /welcome.html:/usr/share/nginx/html/index.html (via /proc/self/fd/6): flags 0x5000: not a dir or file, unknown: Are you trying to mount a directory onto a file (or vice-versa)? Check if the specified host path exists and is the expected type.
[ec2-user@ip-172-31-38-45 ~]$ sudo docker run -d -p 80:80 --name my-nginx -v $(PWD)/welcome.html:/usr/share/nginx/html/index.html nginx:latest
docker: Error response from daemon: Conflict. The container name "my-nginx" is already in use by container "c81a17b1f0c0a419b81b0c7f2f2f710b551c9a413b19314e5173e0dded131ad". You have to remove (or rename) that container to be able to reuse that name.
See 'docker run'
[ec2-user@ip-172-31-38-45 ~]$ sudo docker run -d -p 80:80 --name my-nginx-1 -v $(PWD)/welcome.html:/usr/share/nginx/html/index.html nginx:latest
sha256:060d0c221791b0b9afce037d121a9310401a4ca07110b0c2
[ec2-user@ip-172-31-38-45 ~]$
```

```
ec2-user@ip-172-31-38-45:~$ login as: ec2-user
Authenticating with public key "imported-openssh-key"

Amazon Linux 2023
https://aws.amazon.com/linux/amazon-linux-2023

[ec2-user@ip-172-31-38-45 ~]$ sudo yum update -y
Last metadata expiration check: 0:09:12 ago on Wed Jun 5 05:17:25 2024.
Dependencies resolved.
Nothing to do.
Complete!
[ec2-user@ip-172-31-38-45 ~]$ sudo yum install docker
Last metadata expiration check: 0:09:14 ago on Wed Jun 5 05:17:25 2024.
Dependencies resolved.

Package      Architecture Version                                Repository Size
--
Installing:
docker       x86_64      25.0.3-1.amzn2023.0.1                amazonlinux 44 M
Installing dependencies:
containerd   x86_64      1.7.11-1.amzn2023.0.1                amazonlinux 35 M
iptables-libs x86_64      1.8.8-3.amzn2023.0.2                amazonlinux 401 k
iptables-nft x86_64      1.8.8-3.amzn2023.0.2                amazonlinux 183 k
libgroup     x86_64      3.0-1.amzn2023.0.1                  amazonlinux 75 k
libnetfilter_conntrack x86_64      1.0.8-2.amzn2023.0.2                amazonlinux 58 k
libnetfilter_log x86_64      1.0.1-19.amzn2023.0.2              amazonlinux 30 k
libnftnl     x86_64      1.2.2-2.amzn2023.0.2              amazonlinux 84 k
pigz         x86_64      2.5-1.amzn2023.0.3                  amazonlinux 83 k
runc         x86_64      1.1.11-1.amzn2023.0.1              amazonlinux 3.0 M

Transaction Summary
--
Install 10 Packages

Total download size: 83 M
Is this ok [y/N]: y
Downloading Packages:
(1/10): iptables-libs-1.8.8-3.amzn2023.0.2.x86_64.rpm 3.5 MB/s | 401 kB 00:00
(2/10): iptables-nft-1.8.8-3.amzn2023.0.2.x86_64.rpm 6.1 MB/s | 183 kB 00:00
(3/10): libgroup-3.0-1.amzn2023.0.1.x86_64.rpm 2.4 MB/s | 75 kB 00:00
(4/10): libnetfilter_conntrack-1.0.8-2.amzn2023.0.2.x86_64.rpm 1.7 MB/s | 58 kB 00:00
(5/10): libnetfilter_log-1.0.1-19.amzn2023.0.2.x86_64.rpm 1.5 MB/s | 30 kB 00:00
(6/10): libnftnl-1.2.2-2.amzn2023.0.2.x86_64.rpm 2.3 MB/s | 84 kB 00:00
(7/10): pigz-2.5-1.amzn2023.0.3.x86_64.rpm 2.5 MB/s | 83 kB 00:00
(8/10): runc-1.1.11-1.amzn2023.0.1.x86_64.rpm 30 MB/s | 3.0 MB 00:00
```





**This is page created on putty**

