

KURULUM

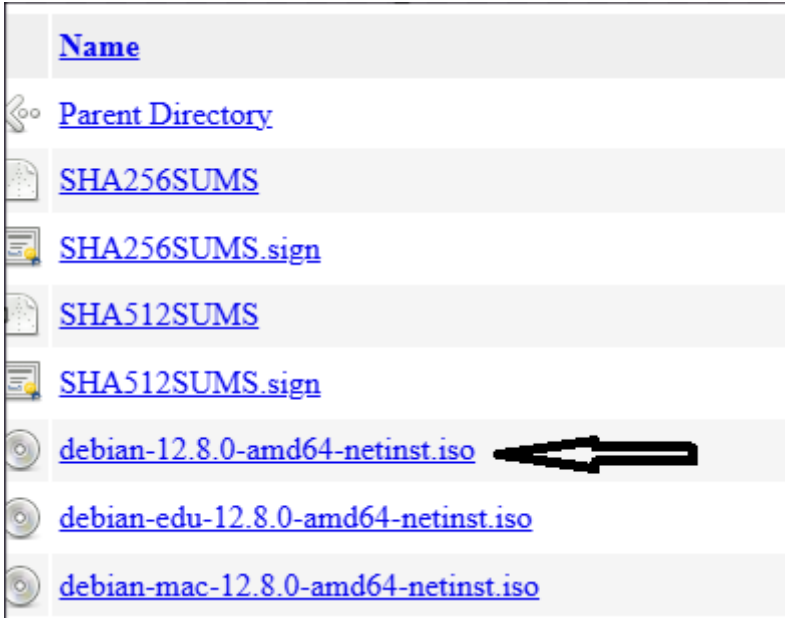
Öncelikle iyi çalışmalar. Sizlere bu pdfte born2beroot adlı projede adım adım yapmanız gerekenleri anlatacağım. Kurulumu Windows bilgisayarımdan yapacağım fakat yapacaklarımın okul ortamında yapılanlardan bir farkı yok. Ek olarak bonusa göre kurulum yapacağım.

Öyleyse başlayalım:

İlk olarak iso dosyamızı indirmemiz lazım.

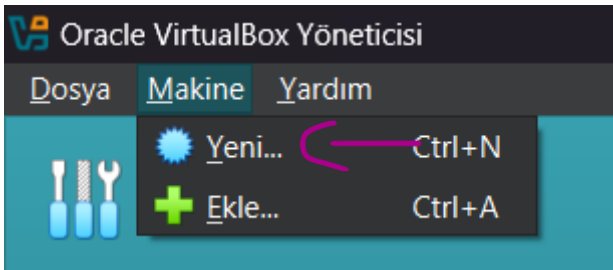
Şu linkten : <https://cdimage.debian.org/debian-cd/current/amd64/iso-cd/>

Altta okla gösterdiğim dosyayı indiriyoruz:



Oracle Virtual Box'u açıyoruz. Ve sırasıyla alttaki işlemleri yapıyoruz:

Makine > Yeni



Bu adımda sanal makinamızın adına, türüne, belleğine vs. müdahale edebiliyoruz.

Ad: Eğer ismin herhangi bir yerinde “deb” bulunursa makina türü otomatik olarak Debian olur.

Örnek : borndeb, deb, debproject vs.

Bu tamamen sizin isteğinize kalmış. Ben “born2beroot” olarak devam edeceğim.

Ad: born2beroot

Klasör: Bu kısım çalıştırılacak dosyalarımızın bulunacağı kısım olacak. Windows üzerinden yaptığım için bende olmasa da sizin bu dizini “goinfre” olarak ayarlamanız lazım. Bunun içinde sağdaki aşağı gösteren oka basıp “Diğer...” seçeneğinden “goinfre” klasörünü seçin.

Windows’ta olduğumdan varsayılandan devam ediyorum.

Klasör: C:\Users\ggolg\VirtualBox VMs

ISO Kalıbı: Bu kısmı burada değil kurulum esnasında ayarlayacağız. Şu anlık dokunmadan devam ediyoruz.

Tür: Eğer sanal makine ismini “deb” içeren bir isim yaptıysanız bu ve alttaki iki adımı geçebilirsiniz. Yapmadıysanız bu kısmı “Linux” olarak seçin.

Tür: Linux

Alt Tür: Bu kısmı “Debian” olarak seçin.

Alt tür: Debian

Sürüm: Bu kısım üstteki iki adımdan sonra otomatik değişecektir fakat değişmezse “Debian (64-bit)” yapın. Bilgisayarınız 32 bit ise “32-bit” olan seçeneği seçin.

Sürüm: Debian (64-bit)

Bu kısımda yapacağımız son şey sanal makinanın bellekte ne kadar alan kaplayacağını seçmek.


Bunun için de öncelikle terminalinize girip “df -h” komutunu çalıştırın. Bu size klasörlerinizde kalan kullanılabilir alanı gösterecektir. goinfre klasörünün kullanılabilir alanı “Avail” sütunu hizasında olacaktır. Şimdi tekrar Oracle Virtual Box’a dönün ve en alttaki “Sabit Disk” e tıklayın.

> Sabit Disk

Aşağıdaki gibi bir kısım açılacaktır:

☒ Şimdi Sanal Bir Sabit Disk Oluştur

Sabit Disk Dosya Konumu ve Boyutu


C:\Users\ggolg\VirtualBox VMs\born2beroot\born2beroot.vdi ✓ 

4,00 MB 20,00 GB 2,00 TB

Sabit Disk Dosya Türü ve Çeşidi


VDI (VirtualBox Disk Kalıbı) ▼ ☐ Tam Boyutu Önceden Ayır ☐ 2GB'lık Parçalara Böl

☐ Varolan Sanal Bir Sabit Disk Dosyası Kullan

Boş ▼ 

☐ Sanal Bir Sabit Disk Eklemek

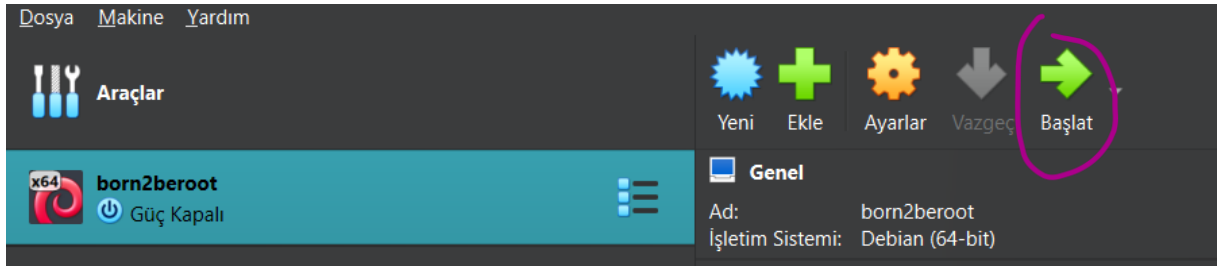
Otomatik olarak “Şimdi Sanal Bir Sabit Disk Oluştur” kısmı seçili olacaktır. Eğer olmazsa onu seçin ve alttaki boyut kısmını goinfre klasörünün boyutuna göre ayarlayın. Örneğin goinfre klasöründe 15 GB alan kaldıysa bu kısmı 15 yapın.

C:\Users\ggolg\VirtualBox VMs\born2beroot\born2beroot.vdi ✓ 

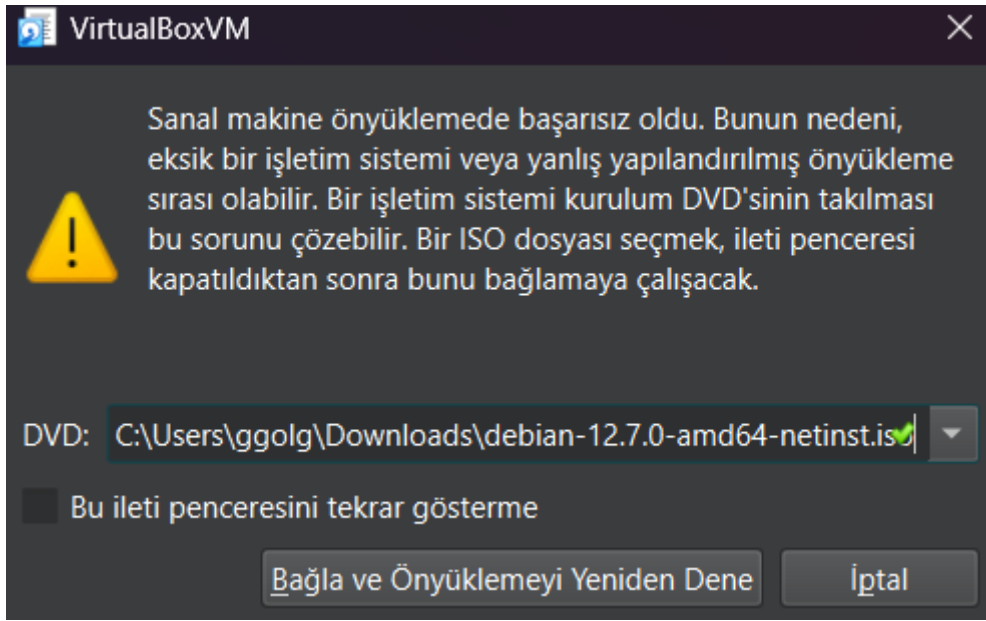
4,00 MB 15,00 GB 2,00 TB

Ben de 15 olarak devam edeceğim. Bu kısımda yapacağımız başka bir şey kalmadı. “Bitir” diyoruz.

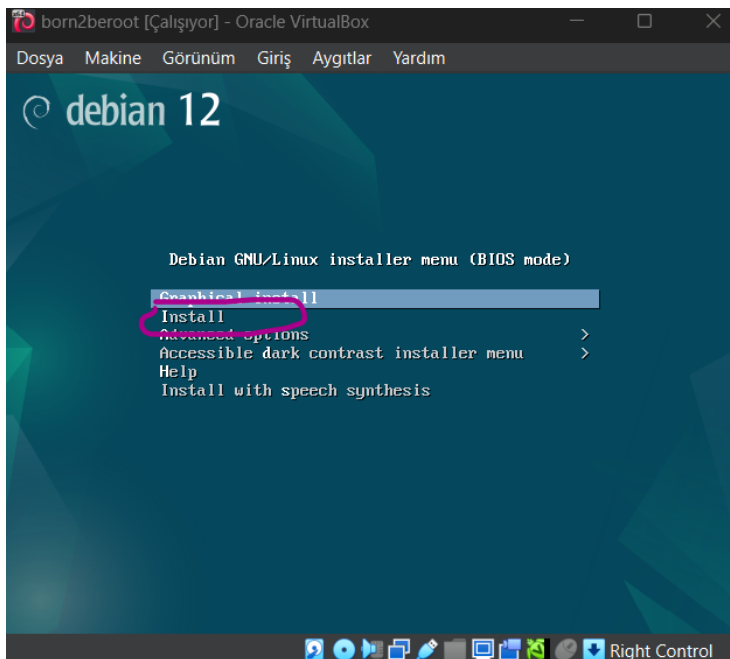
Üst kısımdan “Başlat” a basıp ikinci kısma geçiyoruz.



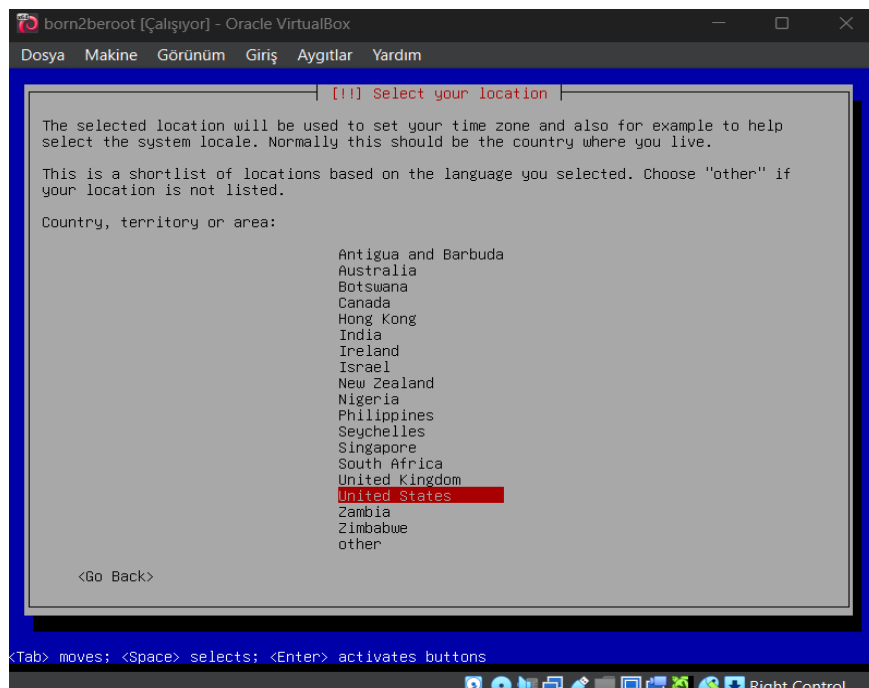
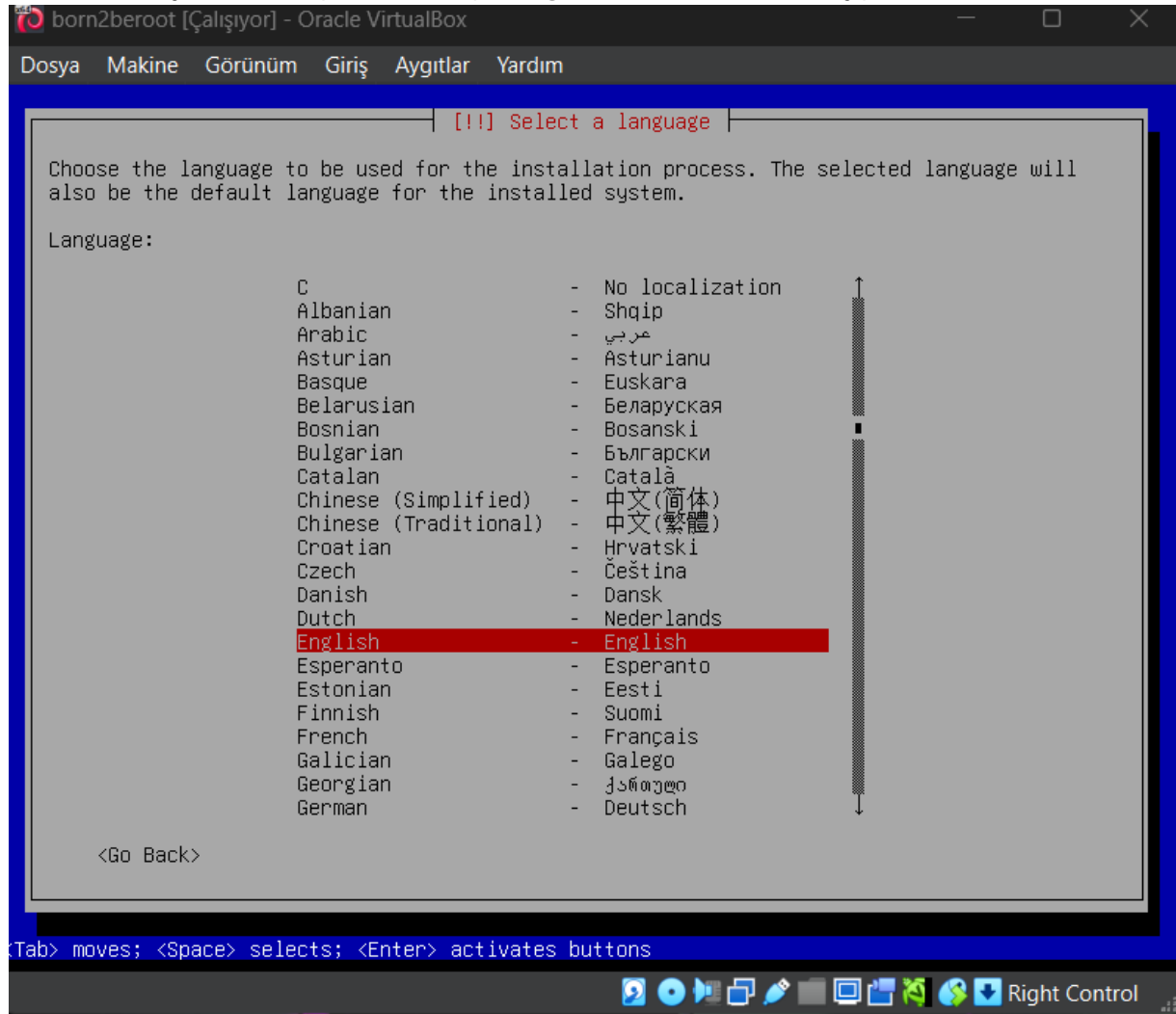
İlk açıldığında bize bir uyarı mesajı verip ISO dosyası seçmemizi isteyecek. ISO dosyamızı seçip “Bağla ve Önyüklemeyi Yeniden Dene” butonuna basıyoruz.



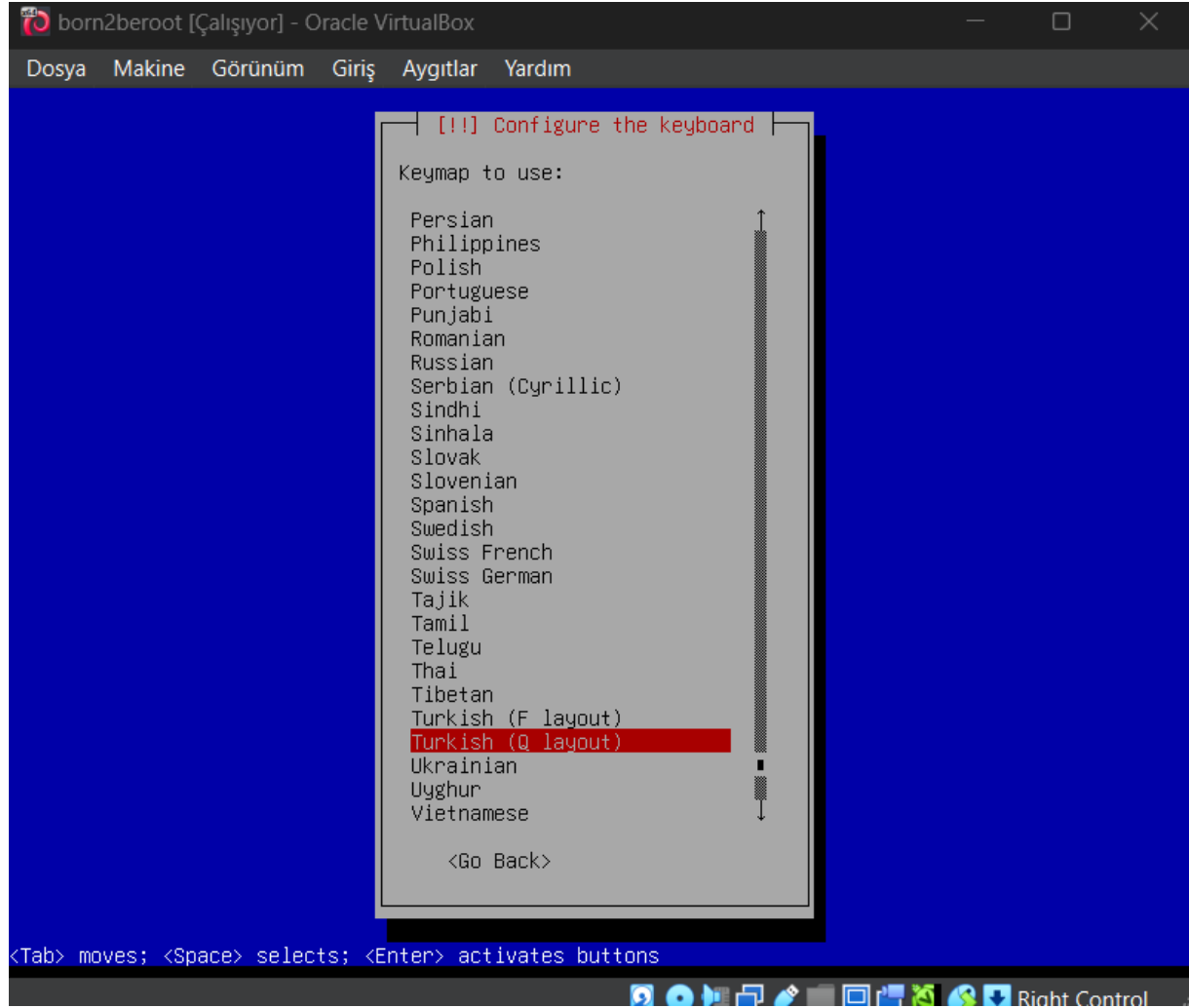
Açılan ekrandan “Install” ı seçiyoruz.



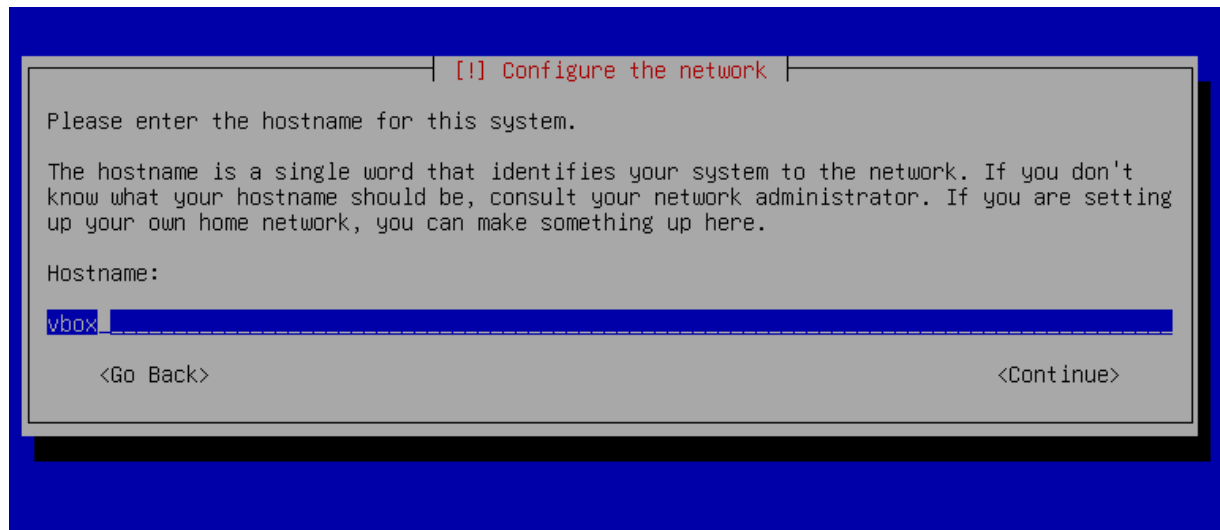
Dil ve ülke seçmemizi isteyecek. Sırasıyla English ve United States seçiyoruz.



Bizden klavye düzenini seçmemizi isteyecek. Türkçe Q klavye kullandığımdan onu seçiyorum. Siz de kendi kullandığınızı bulup onu seçin.



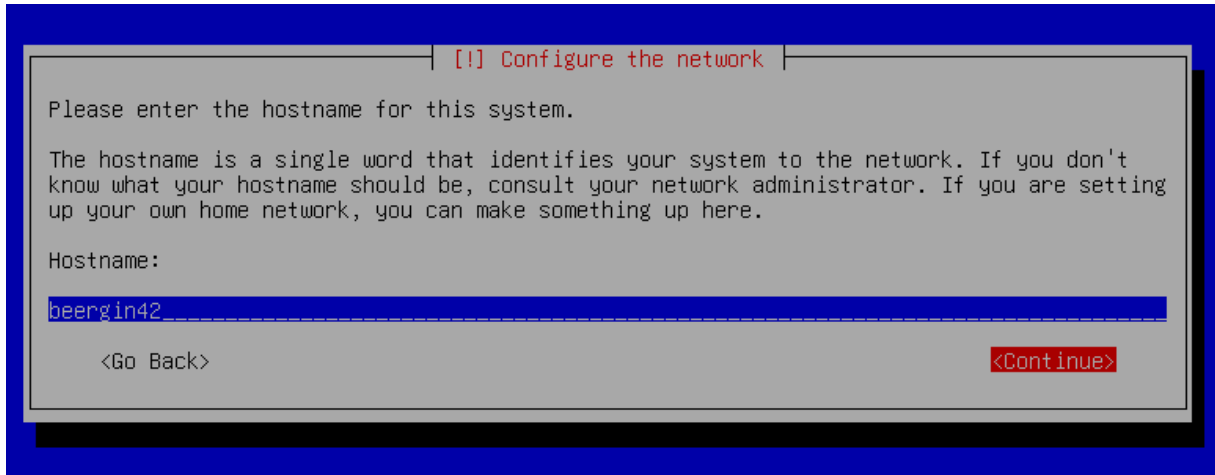
Biraz bekledikten sonra karşımıza şöyle bir ekran gelecek:



Hostname'ı <intrakullanıcıadı>42 olarak ayarlıyoruz.

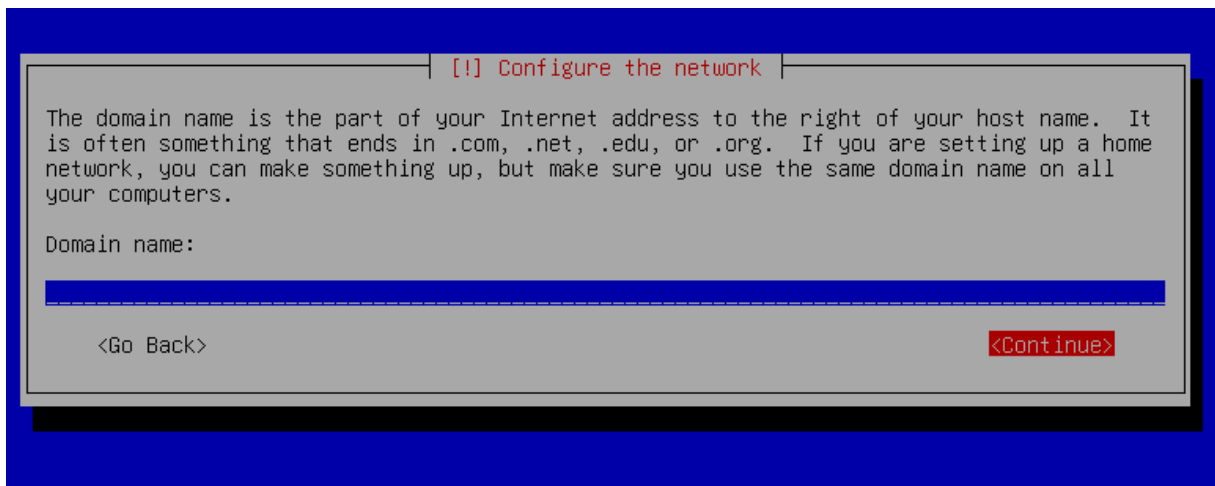
Kullanıcı adım beergin olduğundan “beergin42” olarak ayarlıyorum ve “Continue” ile ilerliyorum.

TAB tuşuna basarak seçenekler arasında gezinebilirsiniz.



The screenshot shows a terminal window titled "[!] Configure the network". The text inside reads: "Please enter the hostname for this system. The hostname is a single word that identifies your system to the network. If you don't know what your hostname should be, consult your network administrator. If you are setting up your own home network, you can make something up here." Below this, it says "Hostname:" followed by a text input field containing "beergin42". At the bottom, there are two buttons: "<Go Back>" and "<Continue>".

Domain Name kısmını boş bırakıyoruz. Sizde boş değil de “42istanbul.com.tr” tarzında da gelebilir. Gelirse silin ve boş bırakın ardından ilerleyin.



The screenshot shows a terminal window titled "[!] Configure the network". The text inside reads: "The domain name is the part of your Internet address to the right of your host name. It is often something that ends in .com, .net, .edu, or .org. If you are setting up a home network, you can make something up, but make sure you use the same domain name on all your computers." Below this, it says "Domain name:" followed by an empty text input field. At the bottom, there are two buttons: "<Go Back>" and "<Continue>".

Root için şifre isteyecek. Bu şifreyi unutmayın çünkü kullanacağız. İlerle diyoruz ve tekrar şifreyi yazıp ilerliyoruz.

!!! Set up users and passwords

You need to set a password for 'root', the system administrative account. A malicious or unqualified user with root access can have disastrous results, so you should take care to choose a root password that is not easy to guess. It should not be a word found in dictionaries, or a word that could be easily associated with you.

A good password will contain a mixture of letters, numbers and punctuation and should be changed at regular intervals.

The root user should not have an empty password. If you leave this empty, the root account will be disabled and the system's initial user account will be given the power to become root using the "sudo" command.

Note that you will not be able to see the password as you type it.

Root password:

☐ Show Password in Clear

<Go Back> <Continue>

!!! Set up users and passwords

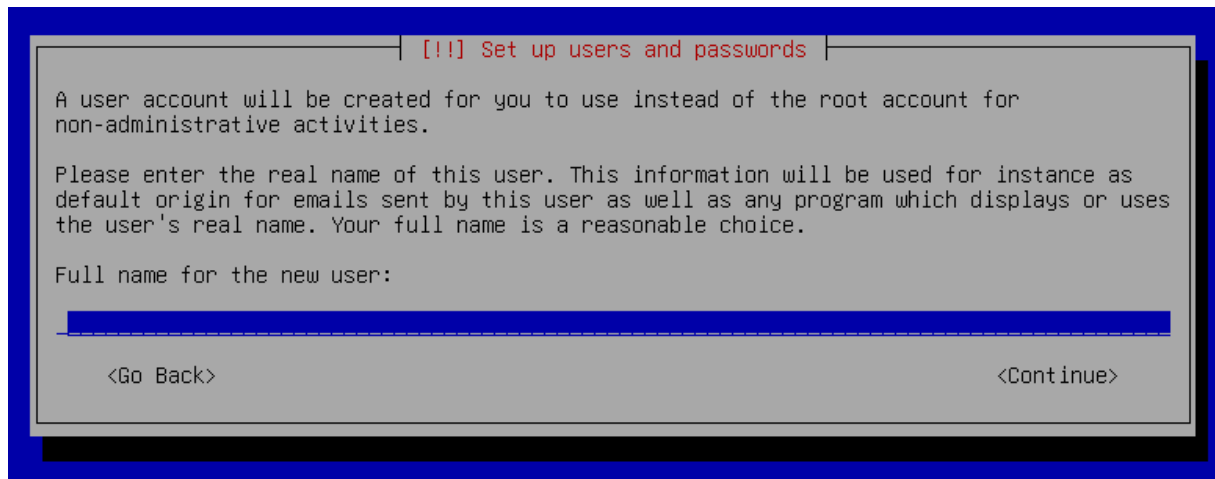
Please enter the same root password again to verify that you have typed it correctly.

Re-enter password to verify:

☐ Show Password in Clear

<Go Back> <Continue>

Bizden isim isteyen bir ekran geliyor bu kısmı boş bırakıyoruz.



!!! Set up users and passwords

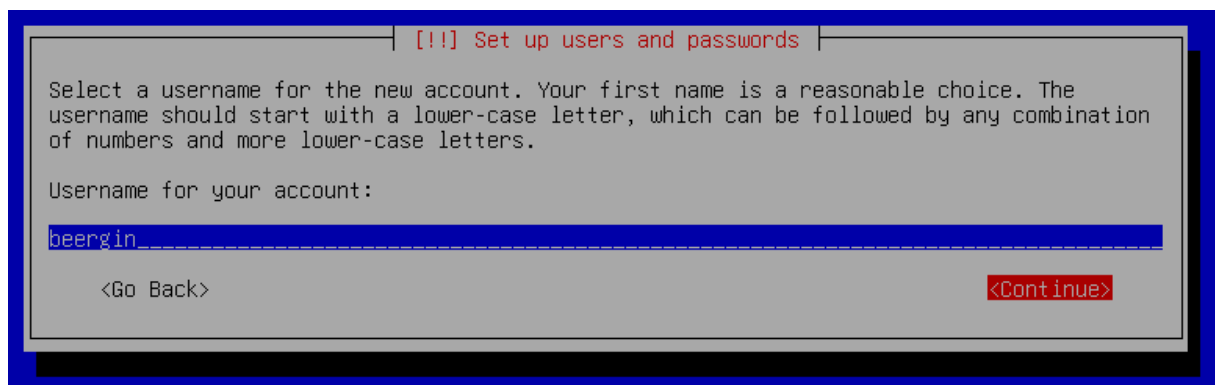
A user account will be created for you to use instead of the root account for non-administrative activities.

Please enter the real name of this user. This information will be used for instance as default origin for emails sent by this user as well as any program which displays or uses the user's real name. Your full name is a reasonable choice.

Full name for the new user:

<Go Back> <Continue>

Yeni gelen kısımda bir kullanıcı adı istiyor. Buraya intra kullanıcı adımızı girip ilerliyoruz.



!!! Set up users and passwords

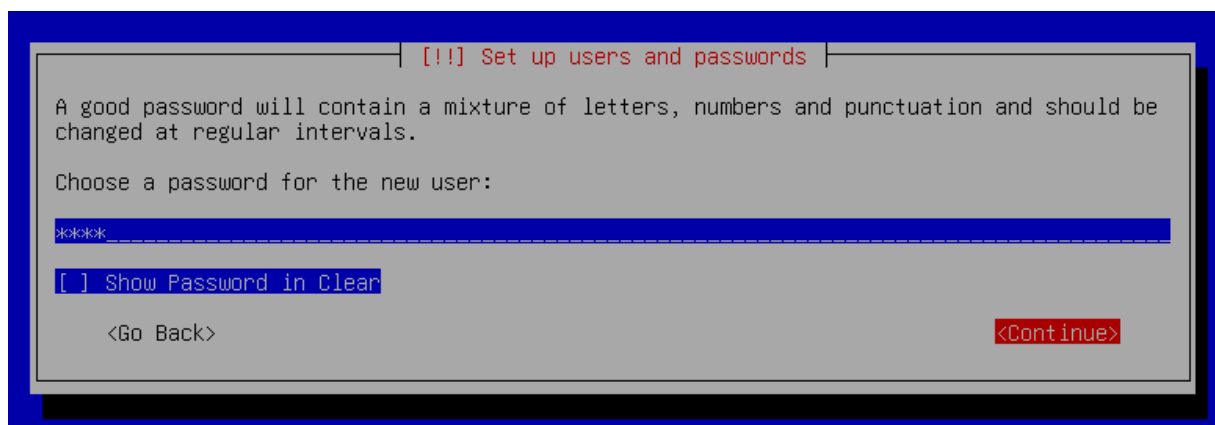
Select a username for the new account. Your first name is a reasonable choice. The username should start with a lower-case letter, which can be followed by any combination of numbers and more lower-case letters.

Username for your account:

beergin

<Go Back> <Continue>

Bu kullanıcı içinde şifre girip ilerliyoruz. Bu şifreyi de unutmayın!



!!! Set up users and passwords

A good password will contain a mixture of letters, numbers and punctuation and should be changed at regular intervals.

Choose a password for the new user:

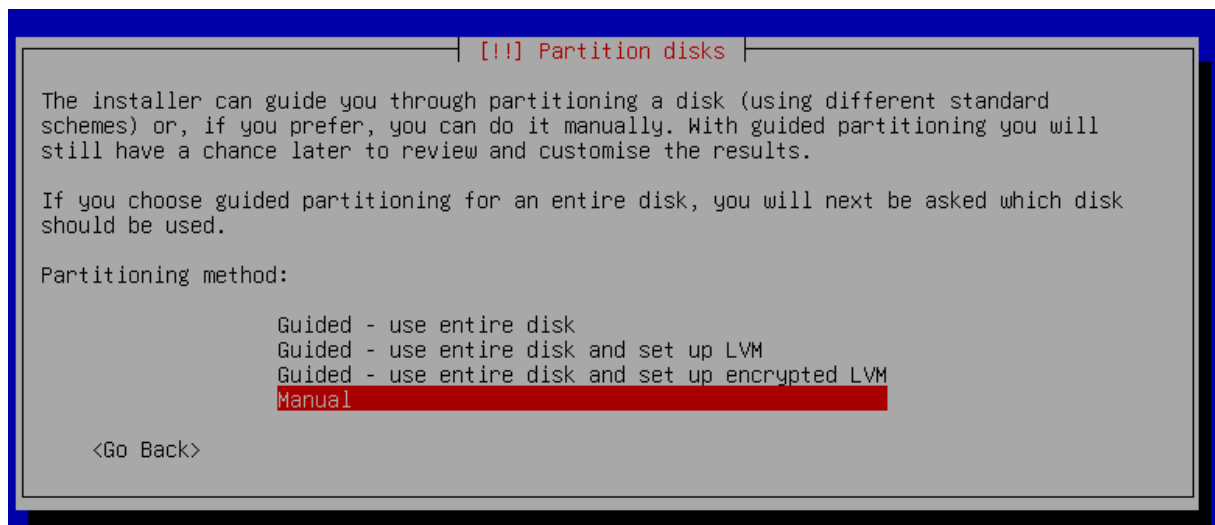
[] Show Password in Clear

<Go Back> <Continue>

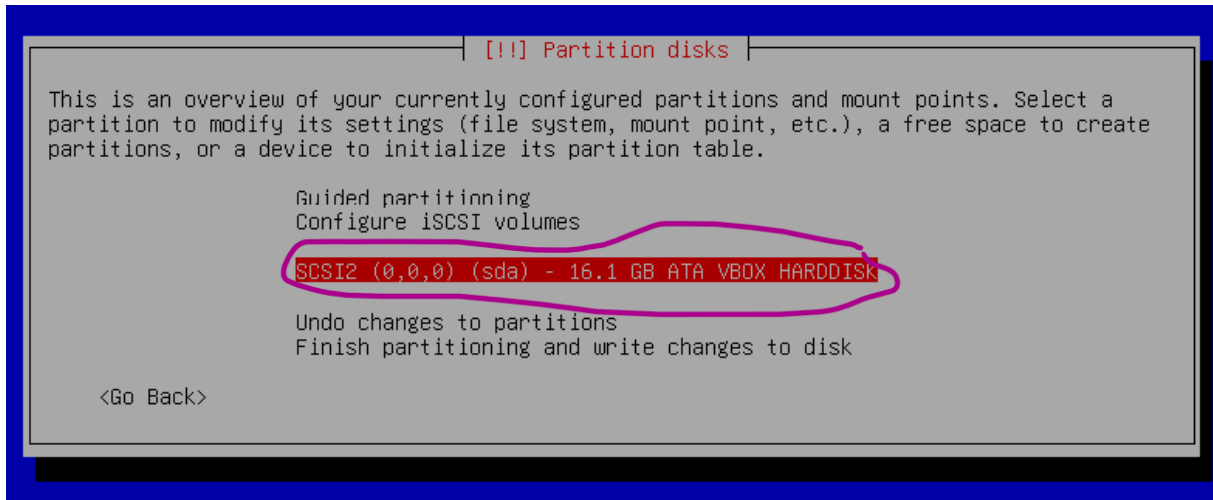
Şimdi ise bir bölge seçmemizi istiyor. Buranın çok bir önemi yok ben “Central” i seçiyorum.



Bize bölümleme işlemini hangi yöntemle yapacağımızı soruyor. Manuel olarak yapacağımızdan “Manual” ı seçiyoruz.



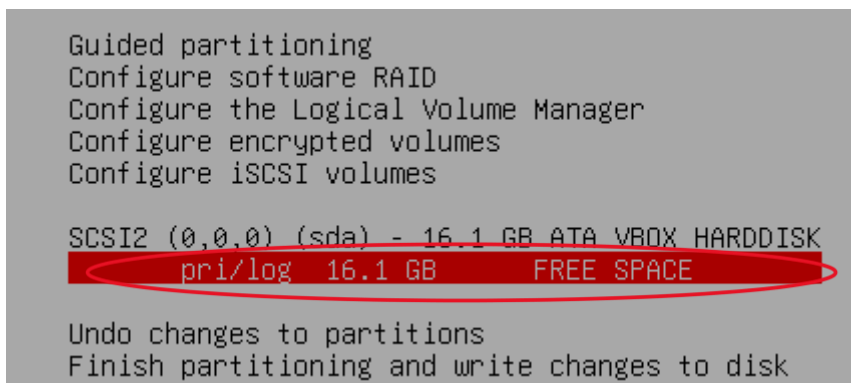
Gelen kısımdan ortadaki seçeneği seçiyoruz.



Bize cihazımızda boş bir bölümlene tablosu oluşturulsun mu diye soruyor. Buna yes diyip ilerliyoruz.



Açılan kısımda işaretlediğim yeri seçiyoruz.



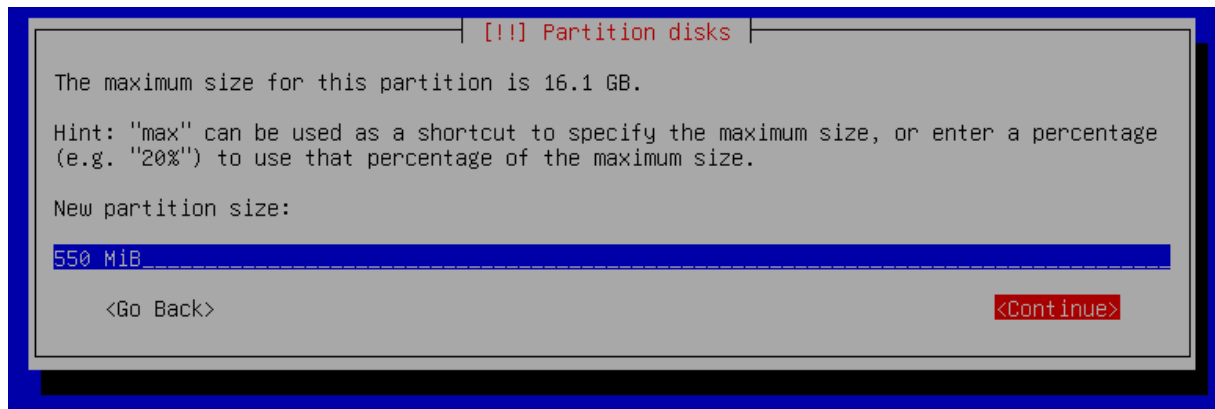
“Create a new partition” diyoruz.



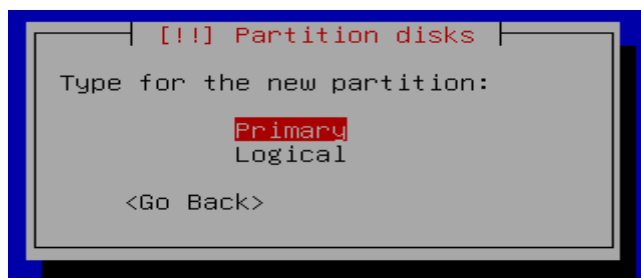
Şimdi bizden bir boyut belirlememizi istiyor. İlk olarak boot kısmını oluşturacağımızdan buraya 550 MiB yazıyoruz.

Neden GiB/MiB?

GiB (Gibibyte) ve MiB (Mebibyte), GB (Gigabyte) ve MB (Mebibyte)’tan yaklaşık %7 daha büyüktür. Eğer 550 MB yazsaydık sistemdeki boyutu daha küçük olacağından tam değeri yansıtmayacaktı. O yüzden MiB yazıyoruz ki tam olarak girdiğimiz değer ya da değere en yakın alanı açsın.

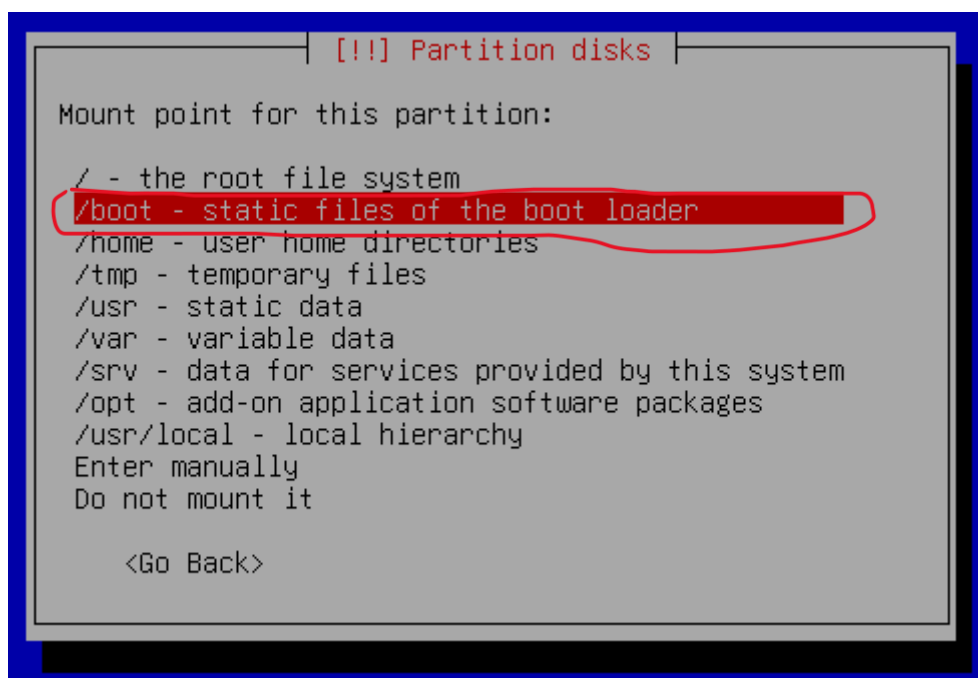
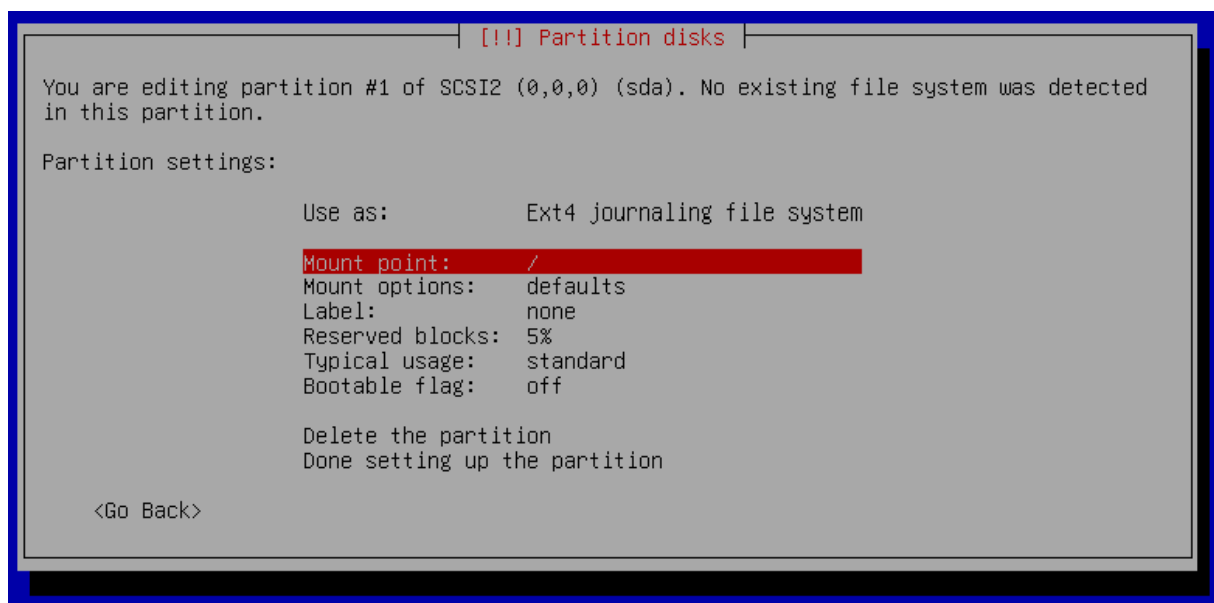


Ardından sırasıyla “Primary” ve “Beginning” seçeneğini seçiyoruz.





Sonrasında gelen ekranda “Mount point” kısmına basıp “boot” seçeneğini seçiyoruz.



Sonrasında alttan “Bootable flag” e basıp “on” a çeviriyoruz.

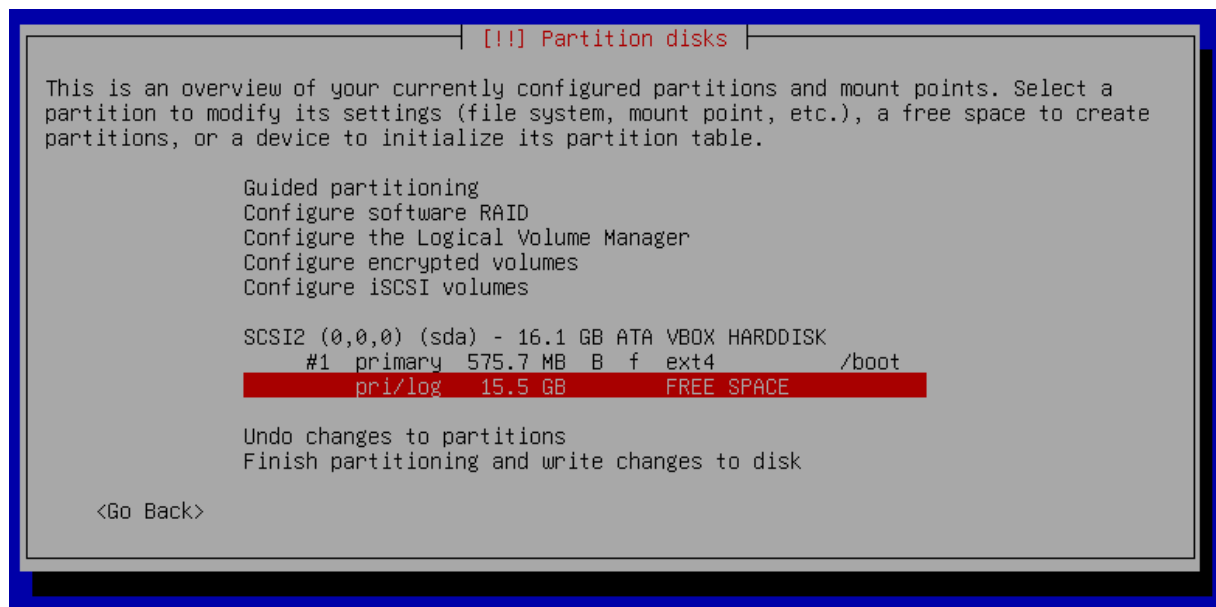
```
Use as:          Ext4 journaling file system
Mount point:     /boot
Mount options:   defaults
Label:           none
Reserved blocks: 5%
Typical usage:   standard
Bootable flag:   on
```

Yaptıktan sonra “Done setting up the partition” a basıyoruz.

```
Use as:          Ext4 journaling file system
Mount point:     /boot
Mount options:   defaults
Label:           none
Reserved blocks: 5%
Typical usage:   standard
Bootable flag:   on

Delete the partition
Done setting up the partition
```

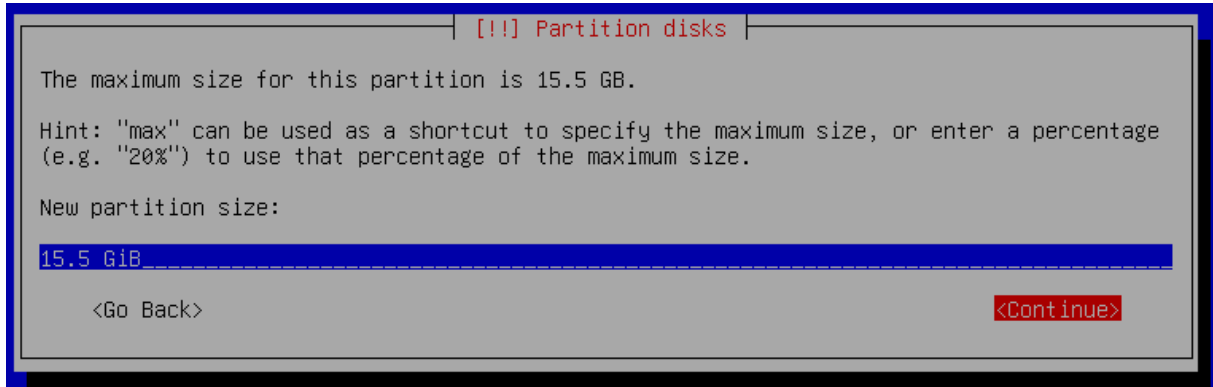
Bu işlemlerden sonra ekranımız aşağıdaki gibi görünmeli:



Şimdi de “FREE SPACE” yazan bölümü seçiyoruz.

```
SCSI2 (0,0,0) (sda) - 16.1 GB ATA VBOX HARDDISK
#1 primary 575.7 MB B f ext4 /boot
pri/log 15.5 GB FREE SPACE
```

Ardından yine “Create a new Partition” seçeneğine basıyoruz. Sonrasında bizden tekrar bir boyut vermemizi istiyor. Buraya geriye kalan tüm alanı (15.5 GiB) yazıyoruz. Ve ilerliyoruz.



!!! Partition disks

The maximum size for this partition is 15.5 GB.

Hint: "max" can be used as a shortcut to specify the maximum size, or enter a percentage (e.g. "20%") to use that percentage of the maximum size.

New partition size:

15.5 GiB

<Go Back> <Continue>

“Logical” ı seçiyoruz.



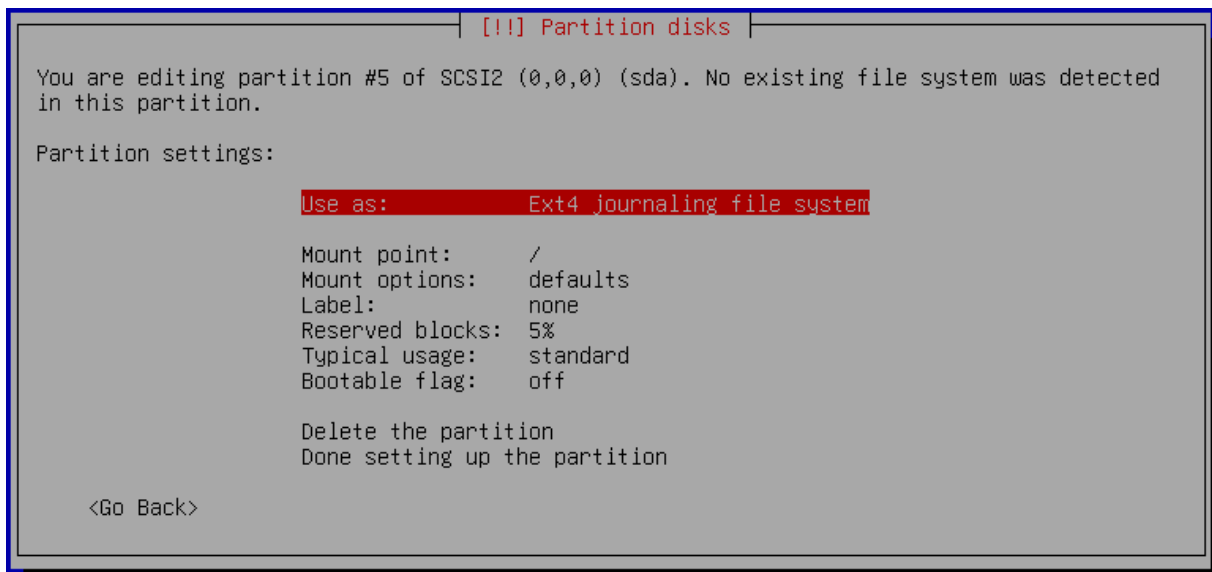
!!! Partition disks

Type for the new partition:

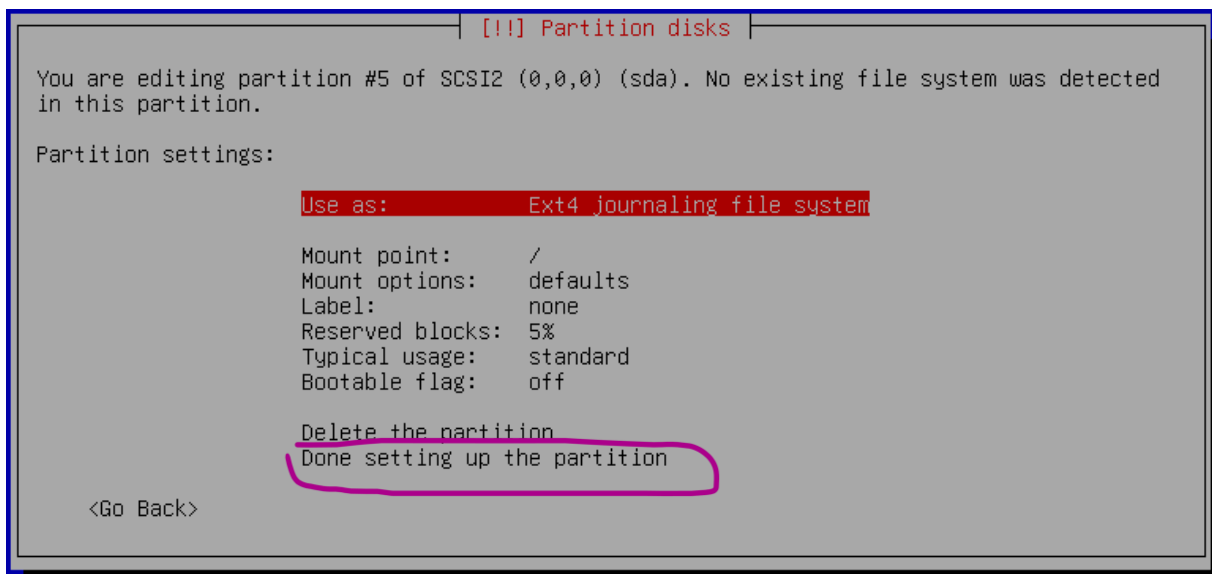
Primary
Logical

<Go Back>

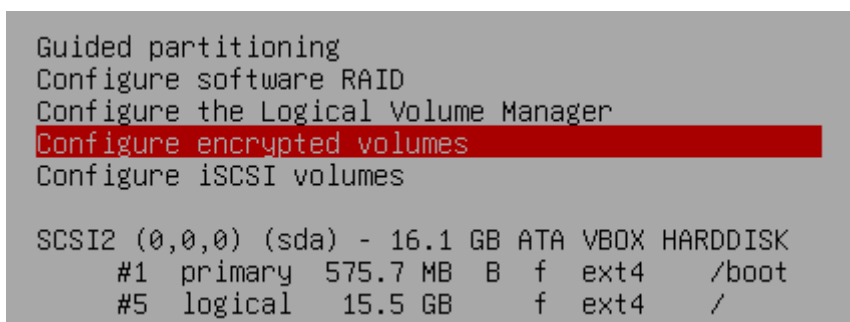
Gelen ekrandan tekrar “Beginning” seçeneğini seçtikten sonra karşımıza şöyle bir ekran gelecek:



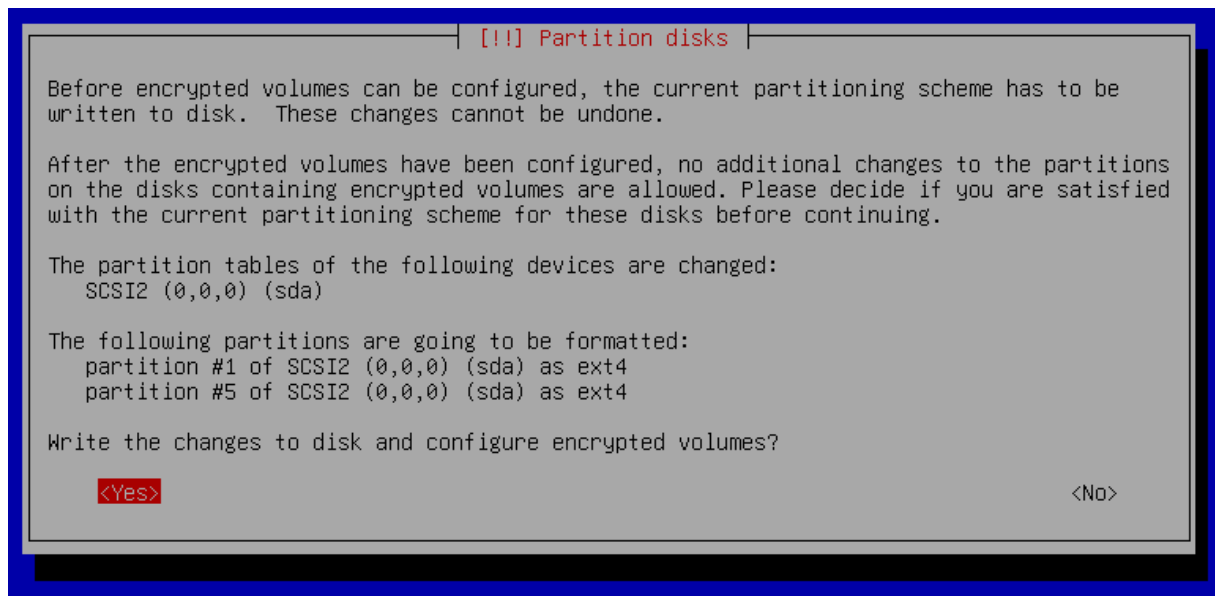
Burayı olduğu gibi bırakıp “Done setting up the partition” a basıyoruz.



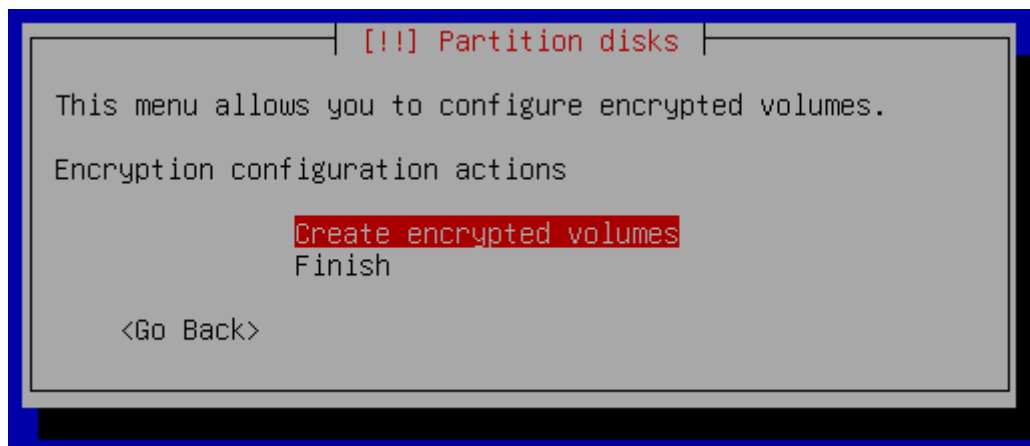
Açılan ekrandan “Configure encrypted volumes” seçeneğini seçiyoruz.



Onay ekranına yes diyip ilerliyoruz.



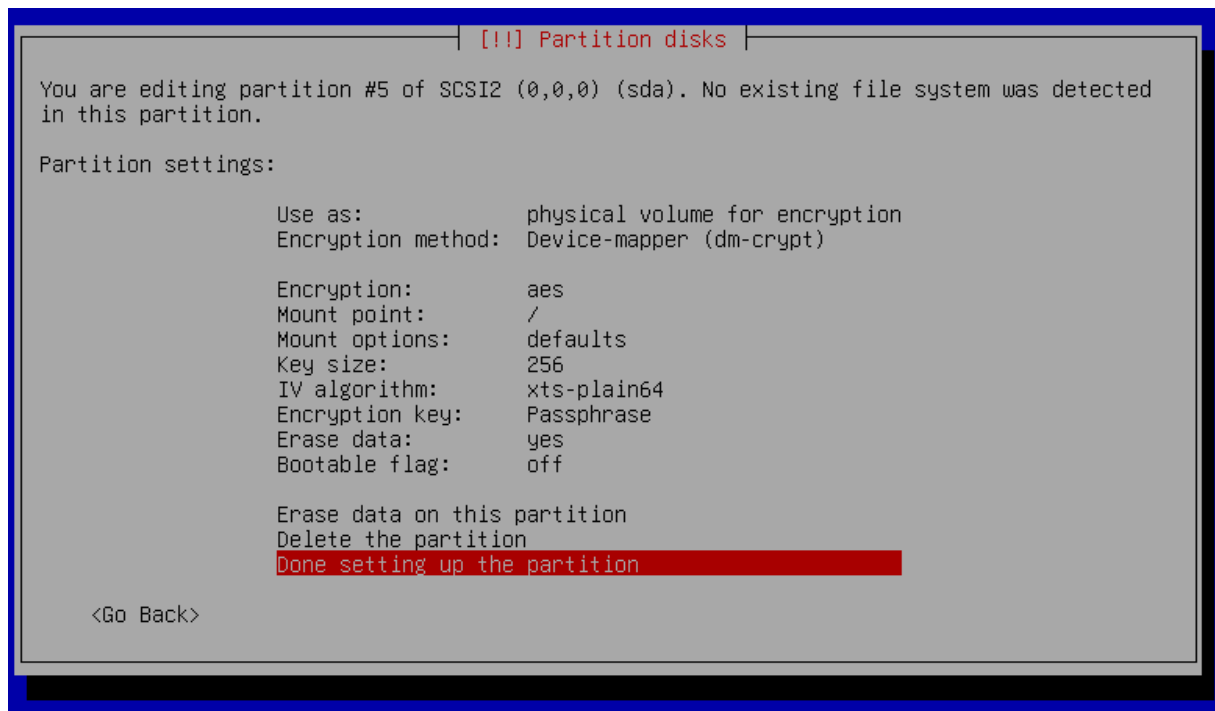
Sonrasında “Create encrypted volumes” ile devam ediyoruz.



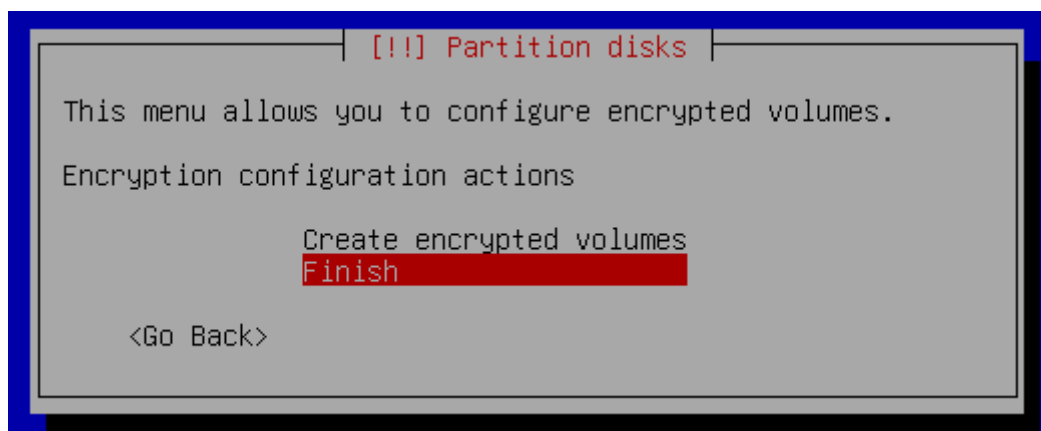
Hangi cihazı şifreleyeceğimizi seçiyoruz. “Sda5” in üstüne gelip SPACE tuşuna basıyoruz ardından “Continue” ile ilerliyoruz.



Karşımıza aşağıdaki gibi ekran gelmeli:



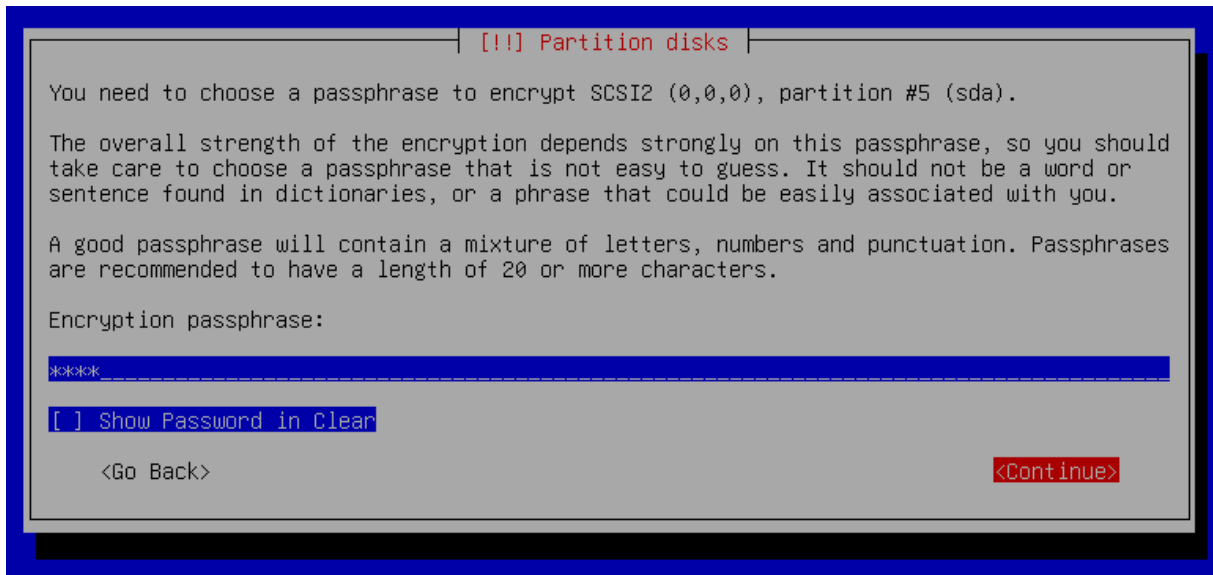
Bu kısma hiç dokunmadan “Done” diyip ilerliyoruz. Sonrasında “Finish” diyoruz.



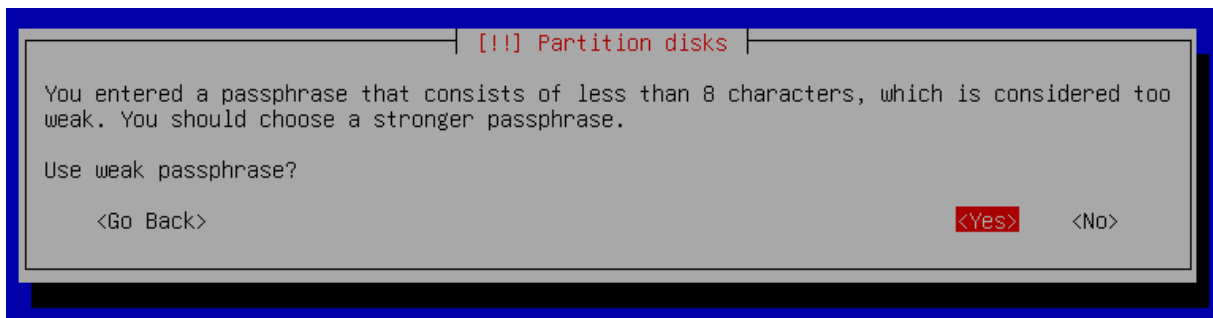
Onay sorusuna yes diyoruz.



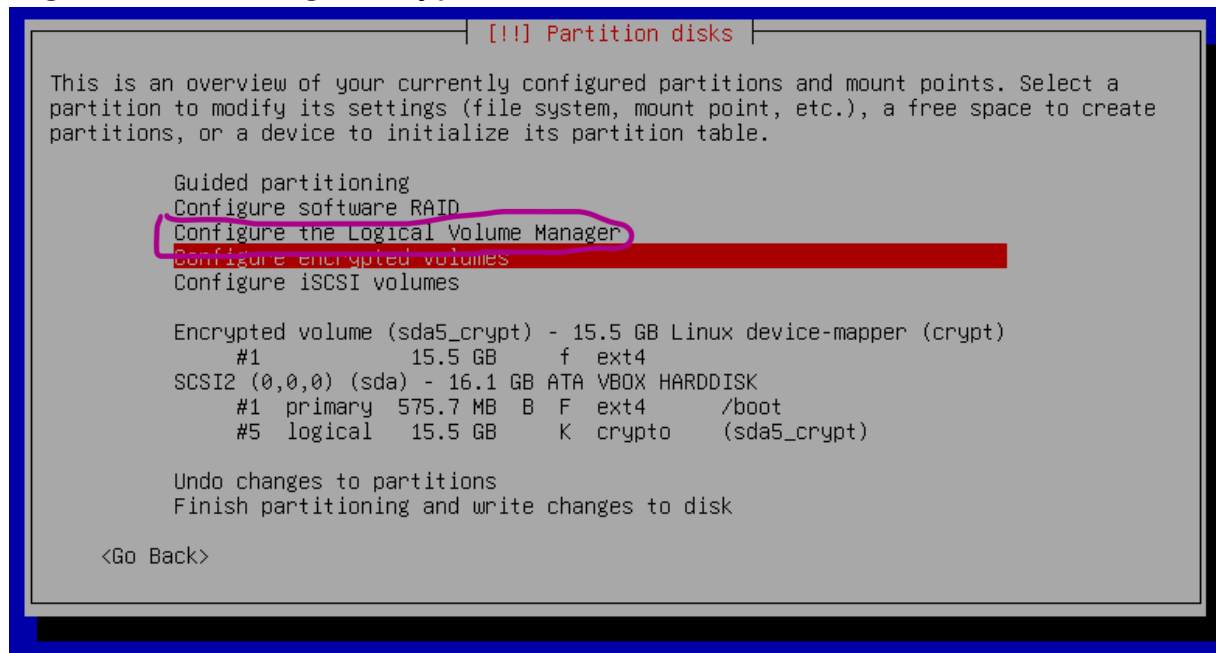
Bir süre bekliyoruz çünkü şifreleme yapıyor. Sonrasında bizden güçlü bir şifre istiyor. Girip ilerliyoruz sonra tekrar girip ilerliyoruz.



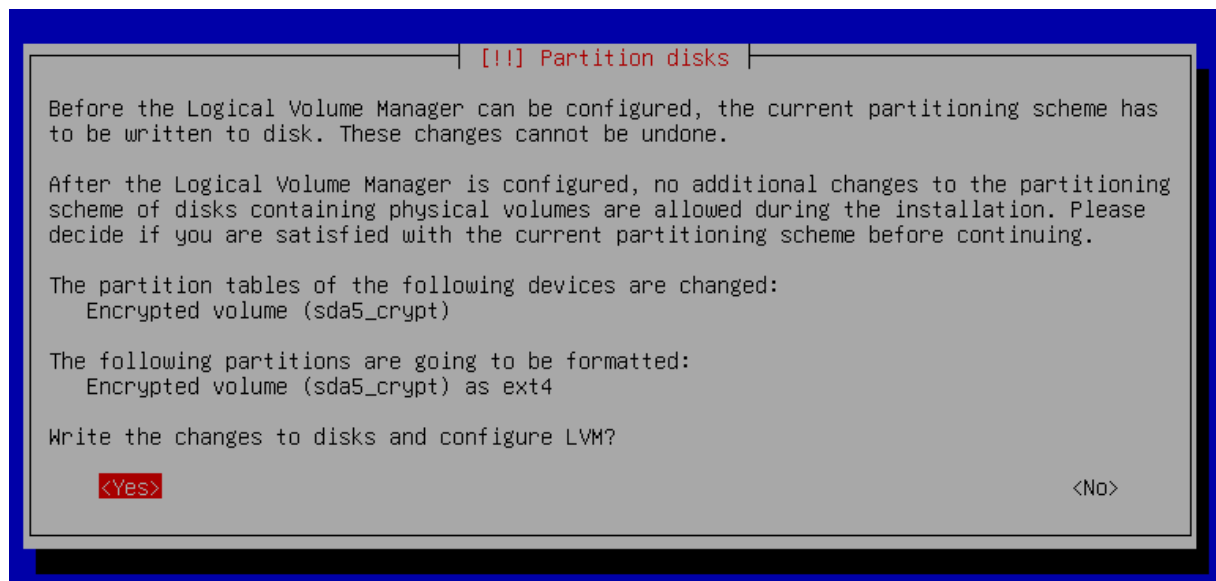
Bize yine bir onay sorusu soracak yes diyoruz.



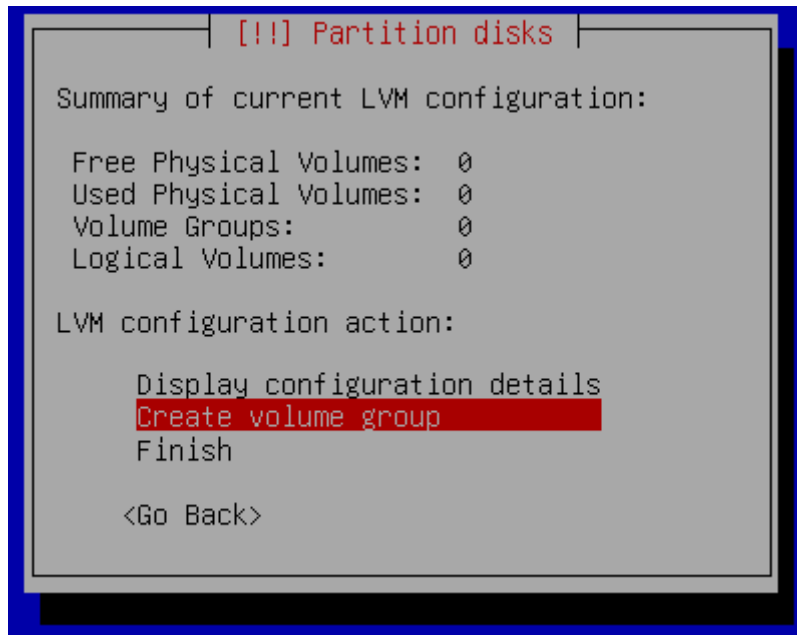
Şimdi artık bölümleri yapılandırmaya geçiyoruz. Üstteki seçeneklerden “Configure the Logical Volume Manager” ı seçiyoruz.



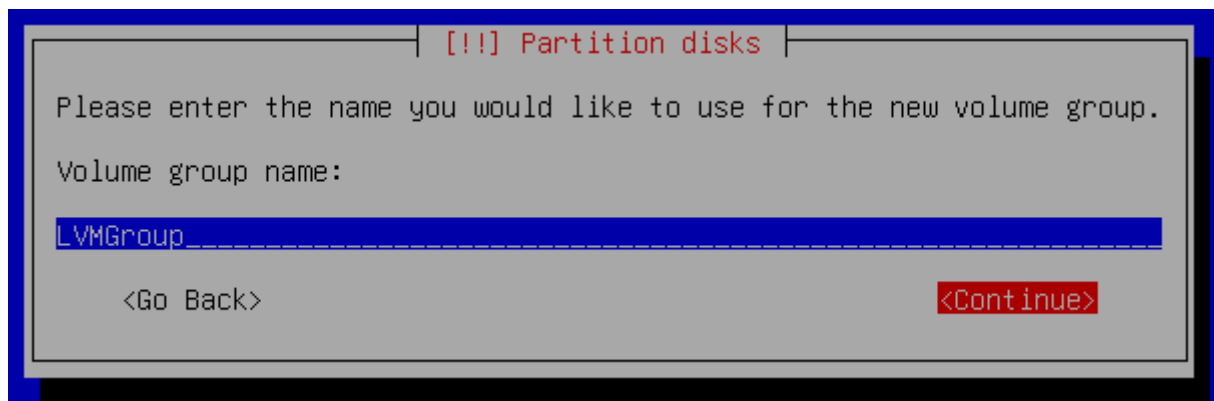
Onay sorusuna yes diyoruz.



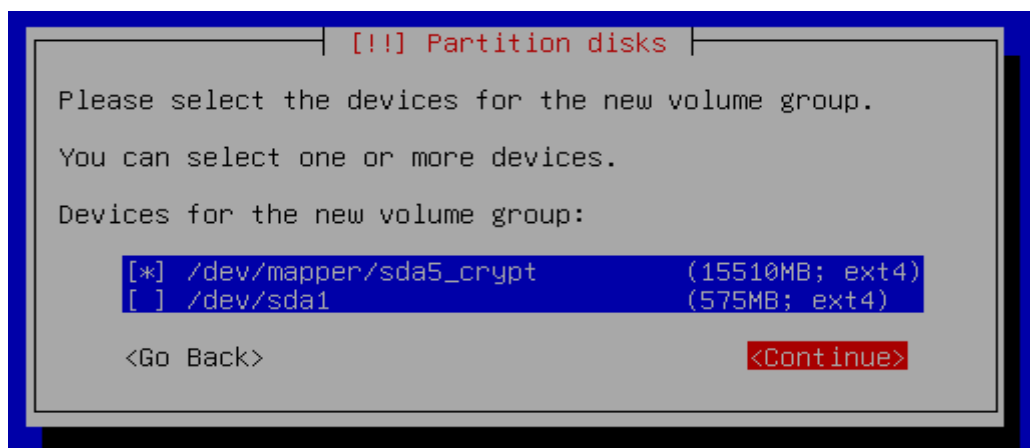
Yeni bir grup oluşturmamız lazım. Bunun için “Create volume group” u seçiyoruz.



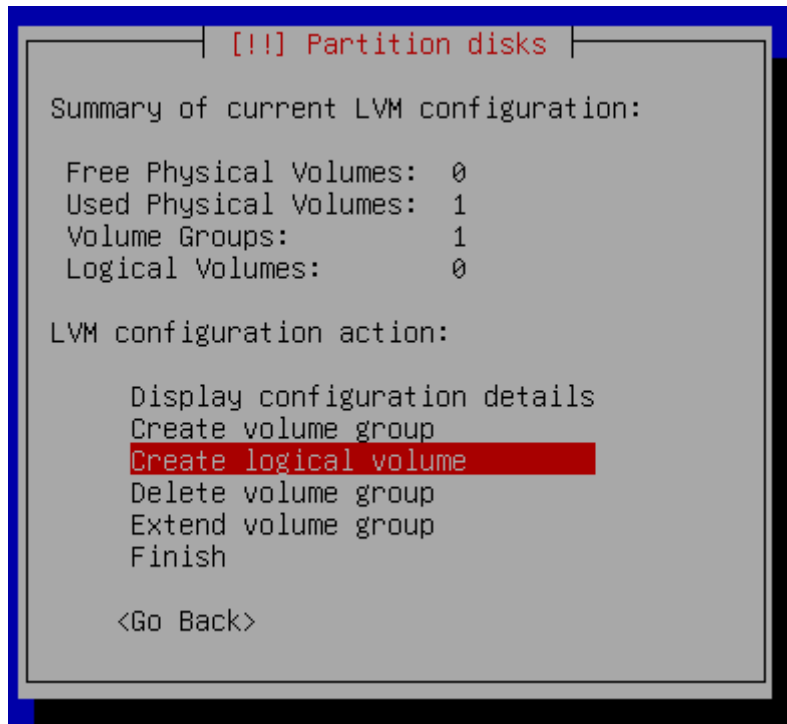
Bir isim belirlememizi istiyor. Buraya “LVGroup” diyoruz ve devam ediyoruz.



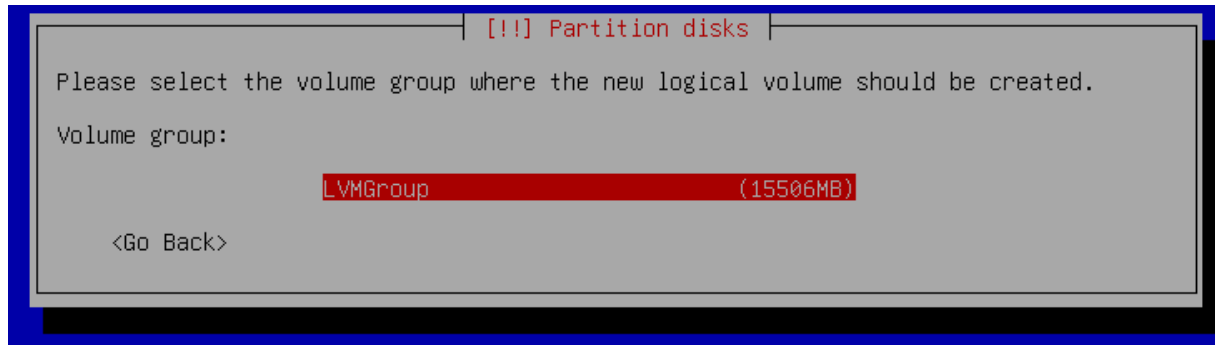
Bu kısımda grup için cihaz seçmemizi istiyor. sda5_crypt seçeneğini seçip ilerliyoruz.



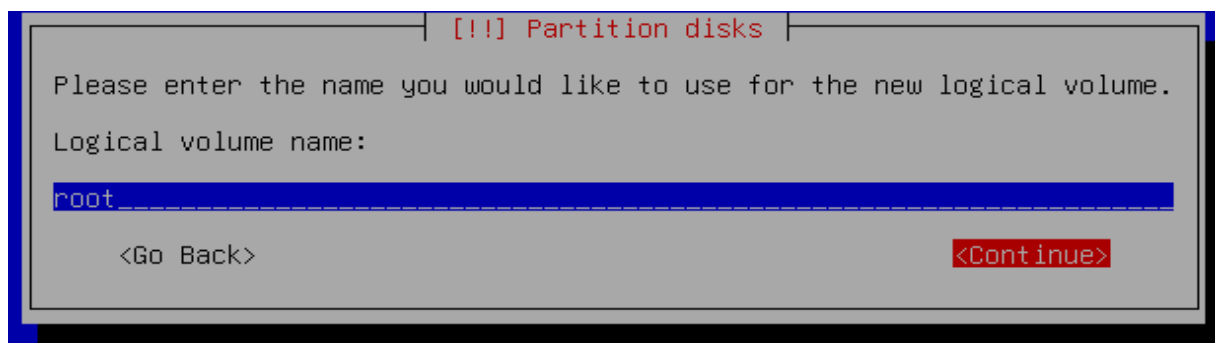
“Create logical volume” seçeneğini seçiyoruz.



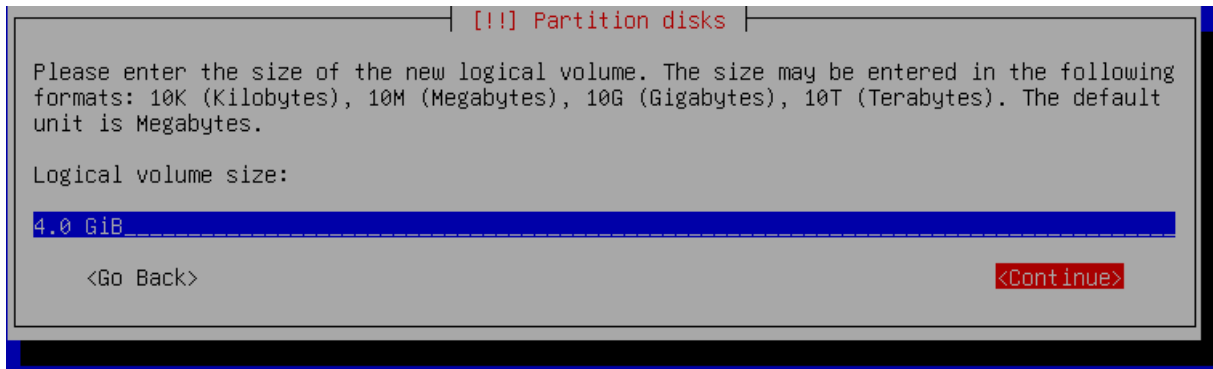
Grubumuzu seçiyoruz.



Bundan sonrasında isimlendirme ve boyutlandırma yapmamızı isteyecek. Buradaki tüm boyutlandırmaları verdiğiniz alana, isimlendirmeyi PDF’e göre yapacaksınız. Biz 15 GB vermiştik. O yüzden buna göre devam edeceğiz. İlk olarak “root” oluşturacağım.



Root'a 4 GiB alan veriyorum.



[[!]] Partition disks

Please enter the size of the new logical volume. The size may be entered in the following formats: 10K (Kilobytes), 10M (Megabytes), 10G (Gigabytes), 10T (Terabytes). The default unit is Megabytes.

Logical volume size:

4.0 GiB

<Go Back> <Continue>

Sonra tekrar "Create logical volume" diyip grubumu seçiyorum. Yine isim veriyorum. Bu sefer "home" vericem.



[[!]] Partition disks

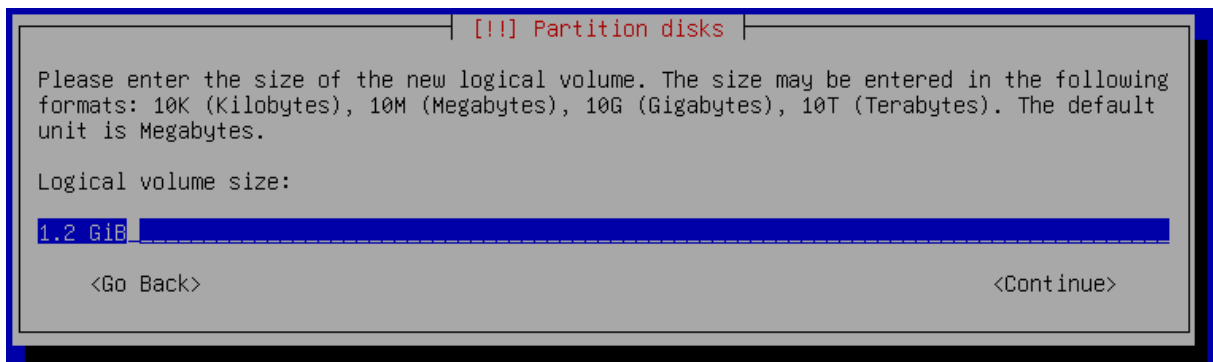
Please enter the name you would like to use for the new logical volume.

Logical volume name:

home

<Go Back> <Continue>

Boyutunu da 1.2 GiB olarak ayarlıyorum.



[[!]] Partition disks

Please enter the size of the new logical volume. The size may be entered in the following formats: 10K (Kilobytes), 10M (Megabytes), 10G (Gigabytes), 10T (Terabytes). The default unit is Megabytes.

Logical volume size:

1.2 GiB

<Go Back> <Continue>

Sonrasında yine "Create logical volume" diyip grup seçiyorum. Ve yine isimlendirme yapıp alan veriyorum. Burası sürekli tekrar edeceğinden alta text olarak veriyorum ayarlamaları:

root > 4.0 GiB

home > 1.2 GiB

swap > 3.1 GiB

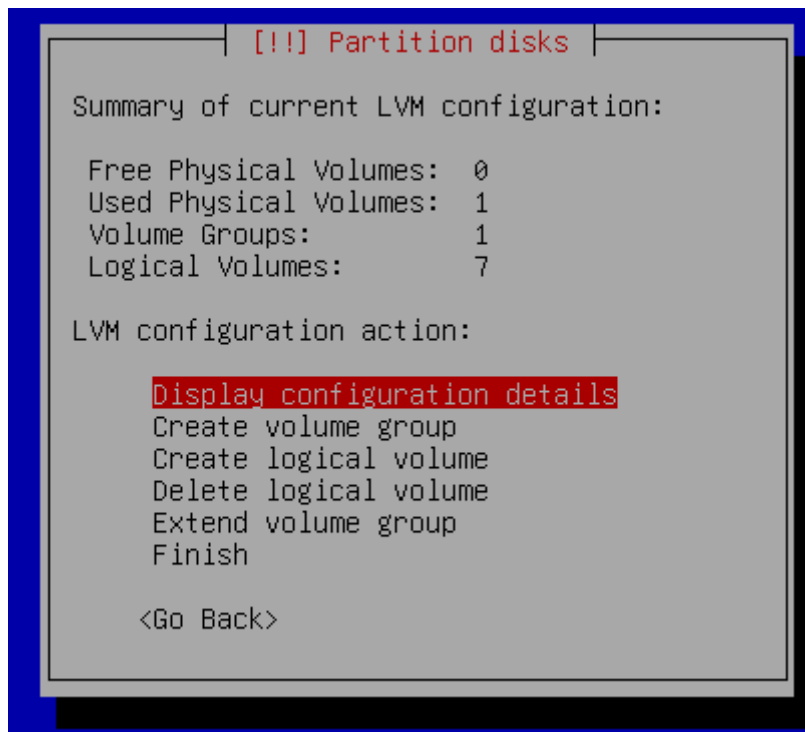
var > 1.2 GiB

srv > 1.2 GiB

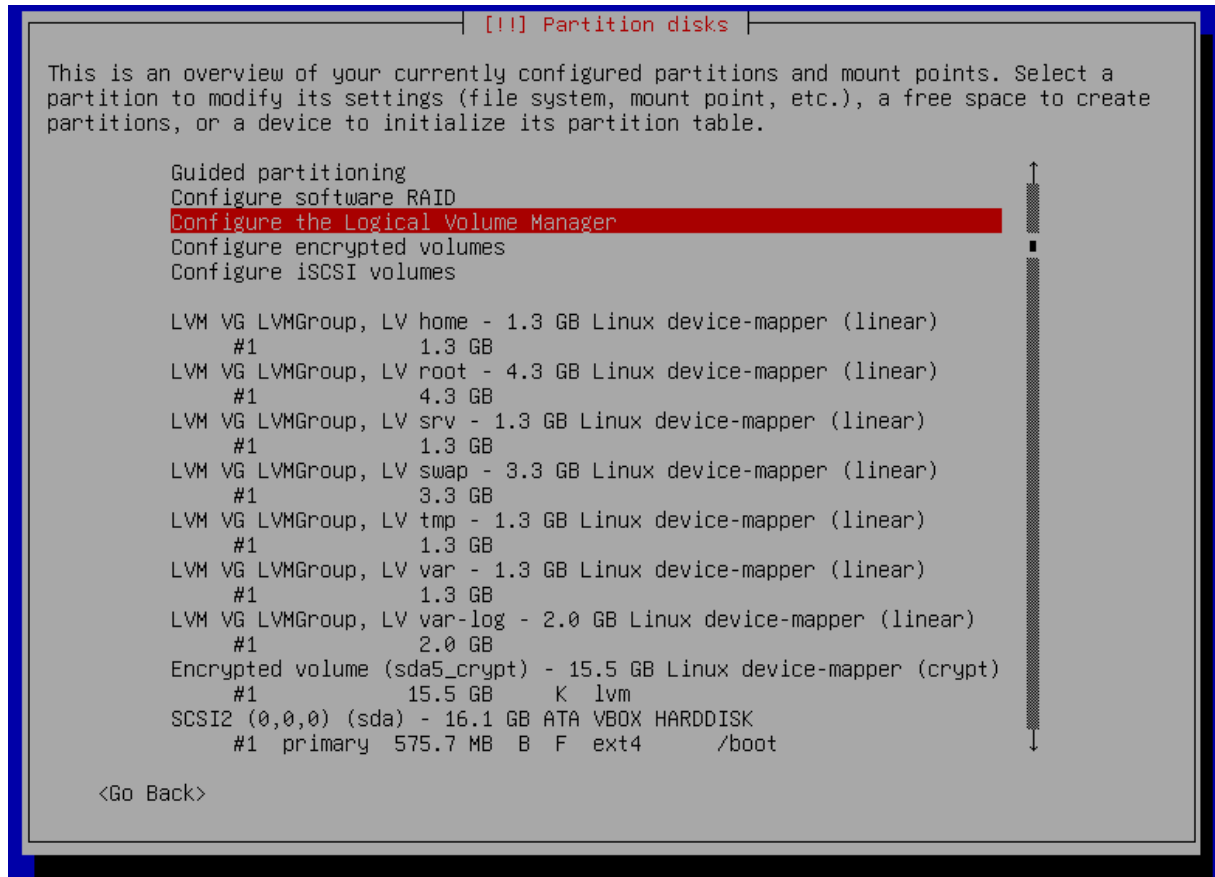
tmp > 1.2 GiB

var-log > 1.9 GiB

Tüm bu ayarlamaları yaptıktan sonra çıktımızın şöyle olması lazım:

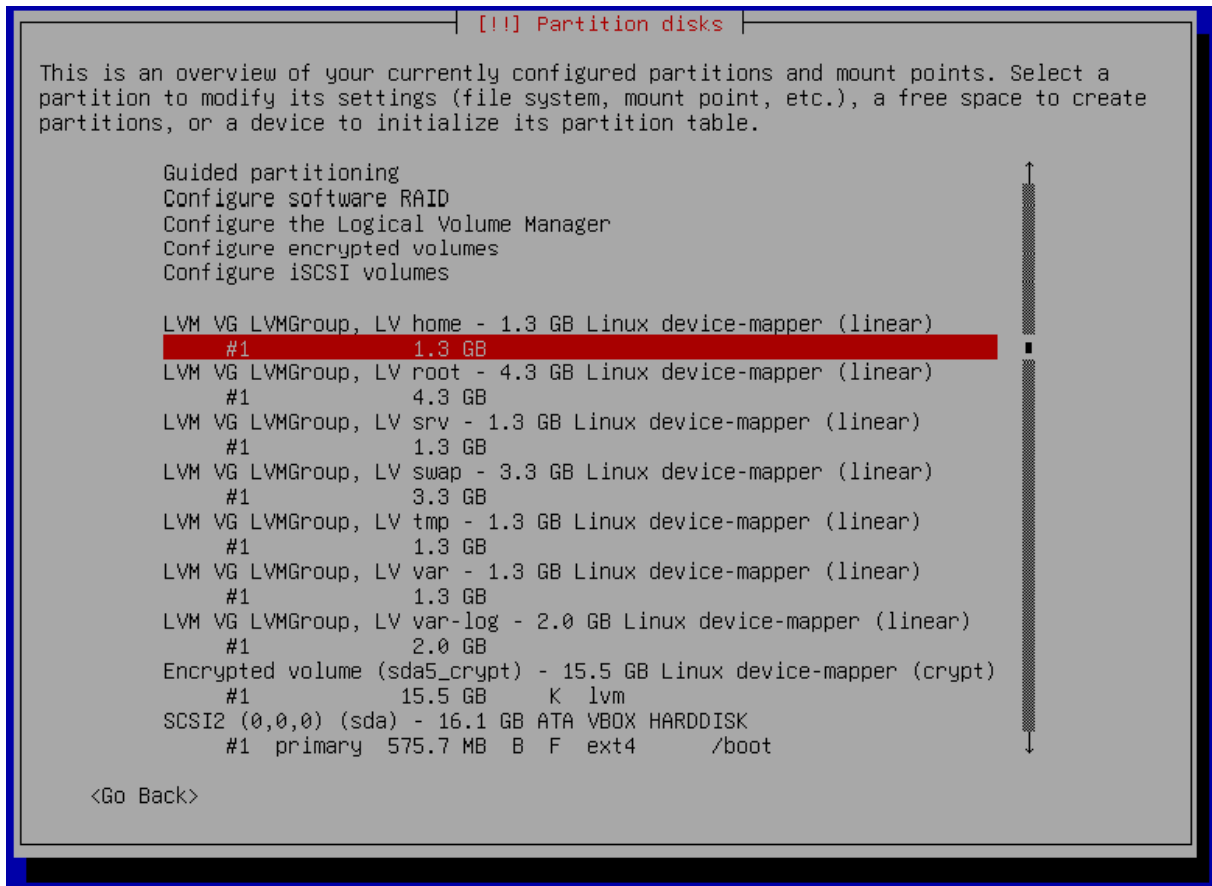


Eğer aynıysa en alttan Finish'e basıyoruz. Sonrasında aşağıdaki gibi bir ekran gelecek:

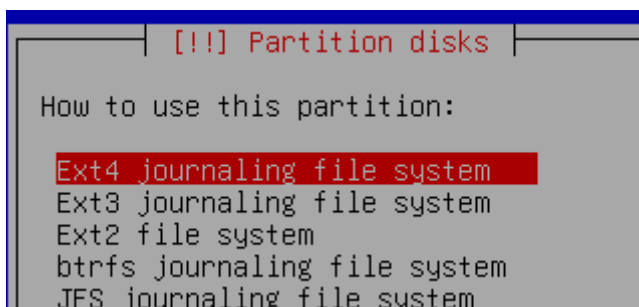
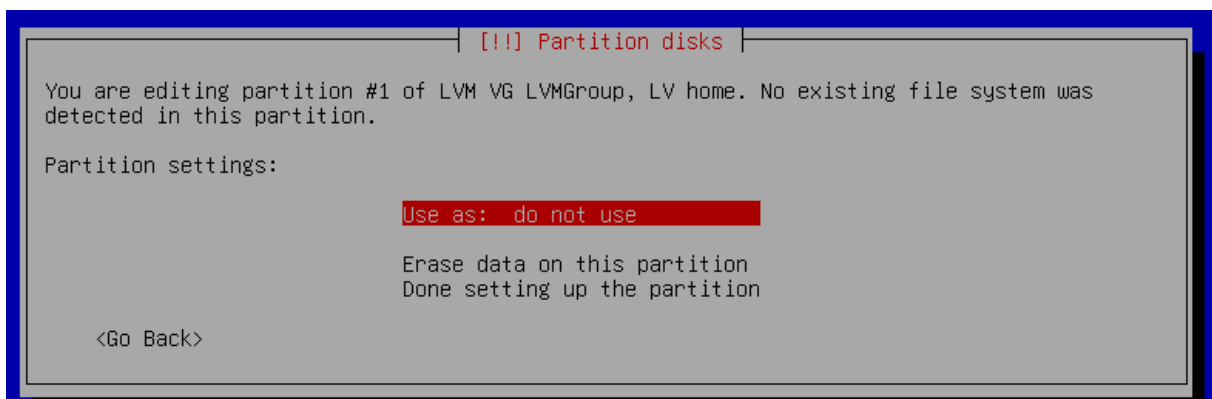


Bu ekrandan her bir klasöre bir bağlama noktası ekleyeceğiz. En baştan başlayarak “#” ile başlayan seçenekleri seçiyoruz.

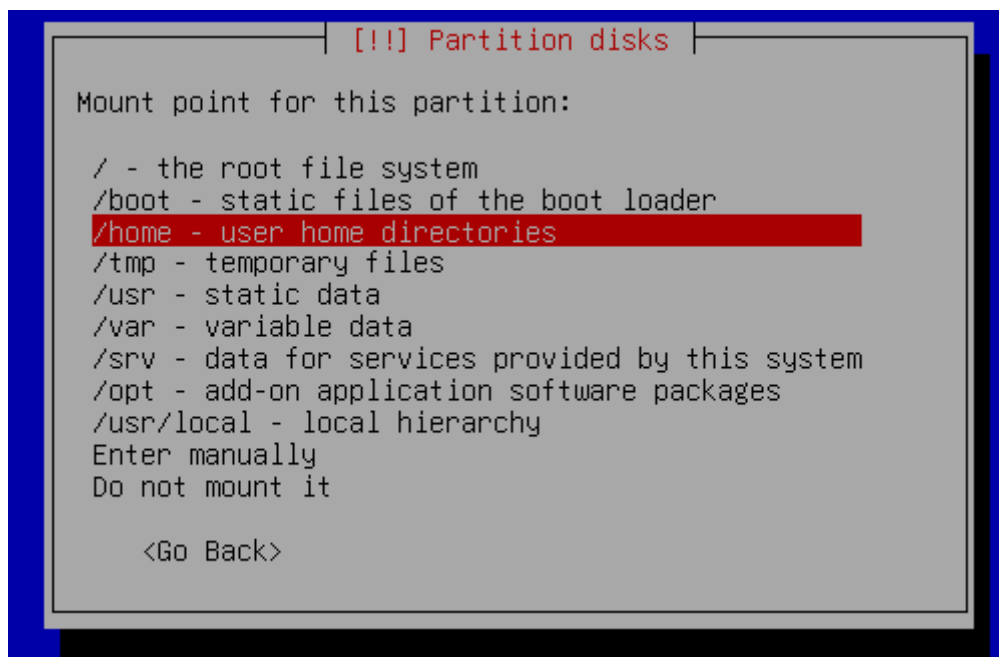
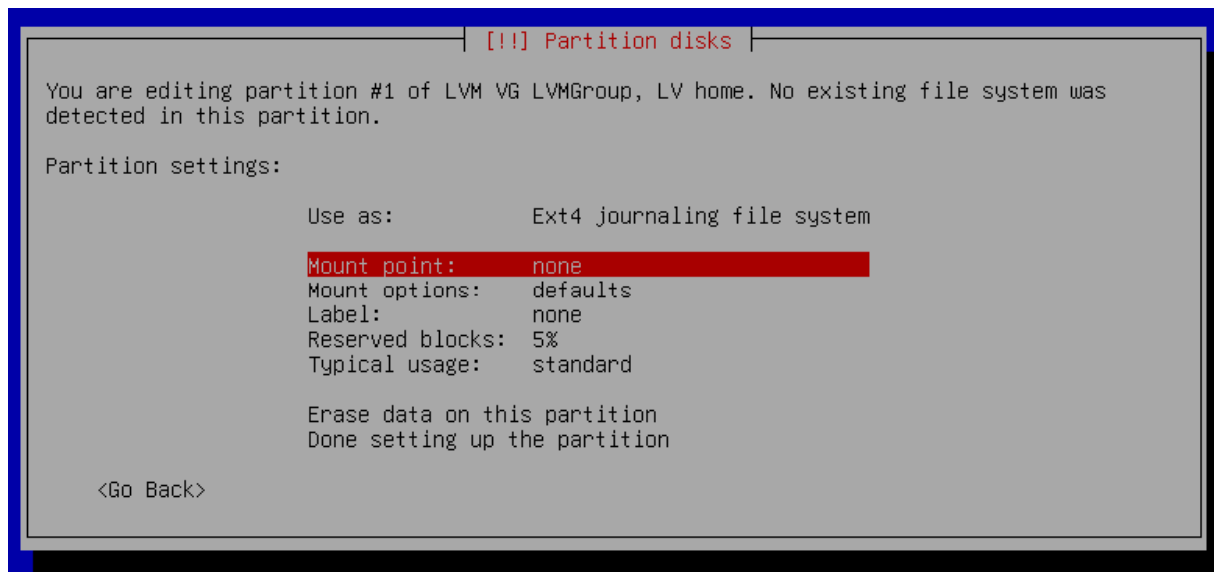
Home için:



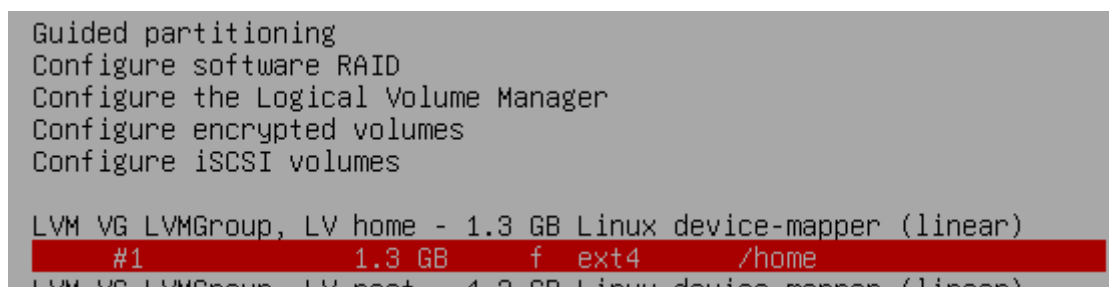
Gelen ekrandan “Use as” kısmına basıyoruz ve Ext4’ü seçiyoruz.



Ardından gelen ekranda da Mount Point'e basıp “/home” ile başlayan seçeneği seçiyoruz.



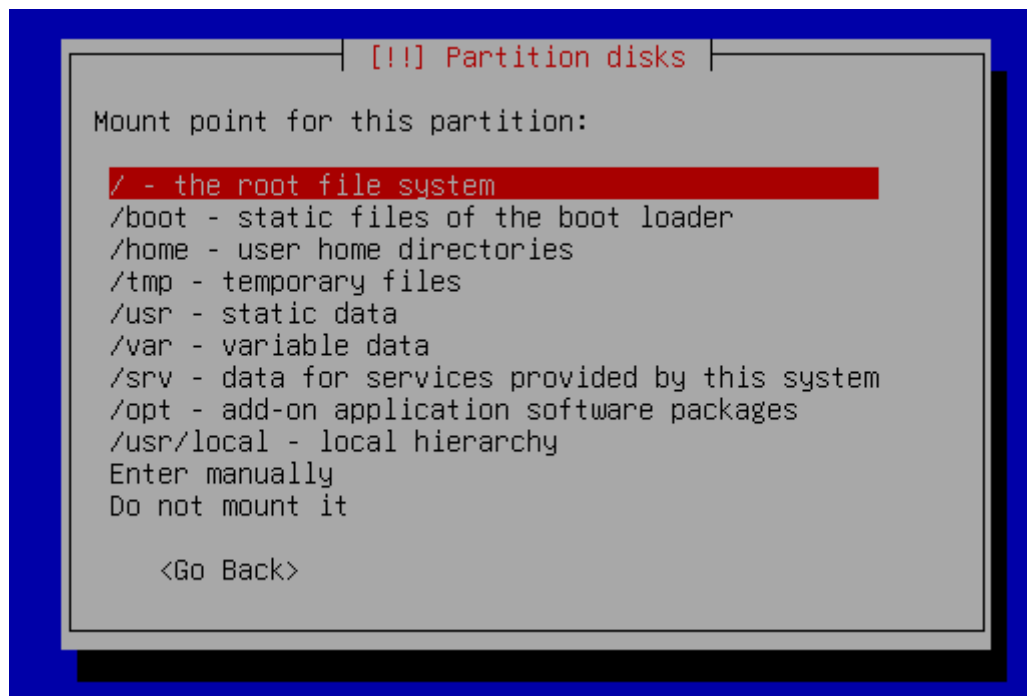
“Done” diyip çıkıyoruz. Aşağıda gördüğünüz üzere artık /home’a bağlandı. Sıra diğerlerinde.



root'un altındakini seçiyoruz.

```
LVM VG LVMGroup, LV home - 1.3 GB Linux device-mapper (linear)
#1 1.3 GB f ext4 /home
LVM VG LVMGroup, LV root - 4.3 GB Linux device-mapper (linear)
#1 4.3 GB
LVM VG LVMGroup, LV srv - 1.3 GB Linux device-mapper (linear)
#1 1.3 GB
LVM VG LVMGroup, LV swap - 3.3 GB Linux device-mapper (linear)
#1 3.3 GB
LVM VG LVMGroup, LV tmp - 1.3 GB Linux device-mapper (linear)
#1 1.3 GB
LVM VG LVMGroup, LV var - 1.3 GB Linux device-mapper (linear)
#1 1.3 GB
LVM VG LVMGroup, LV var-log - 2.0 GB Linux device-mapper (linear)
#1 2.0 GB
Encrypted volume (sda5_crypt) - 15.5 GB Linux device-mapper (crypt)
#1 15.5 GB K lvm
SCSI2 (0,0,0) (sda) - 16.1 GB ATA VBOX HARDDISK
#1 primary 575.7 MB B F ext4 /boot
```

Tekrar “use as” seçip ext4’ü seçiyoruz. Sonrasında Mount Point’ basıyoruz. Buradan en üstteki “/” olanı seçiyoruz.



Sonra tekrar “Done” diyip ilerliyoruz. Buradan sonrakiler de tekrar ettiğinden alta text olarak veriyorum:

home > use as > Ext4 > Mount Point > /home

root > use as > Ext4 > Mount Point > /

srv > use as > Ext4 > Mount Point > /srv

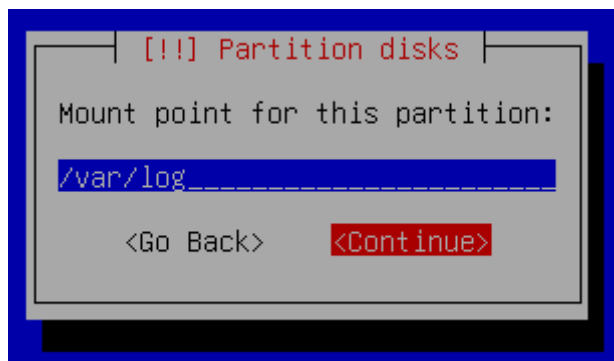
swap > use as > swap area

tmp > use as > Ext4 > Mount Point > /tmp

var > use as > Ext4 > Mount Point > /var

var-log > Ext4 > Mount Point > Enter manually

Açılan ekrana /var/log yazıp ilerliyoruz ve “Done” diyoruz.

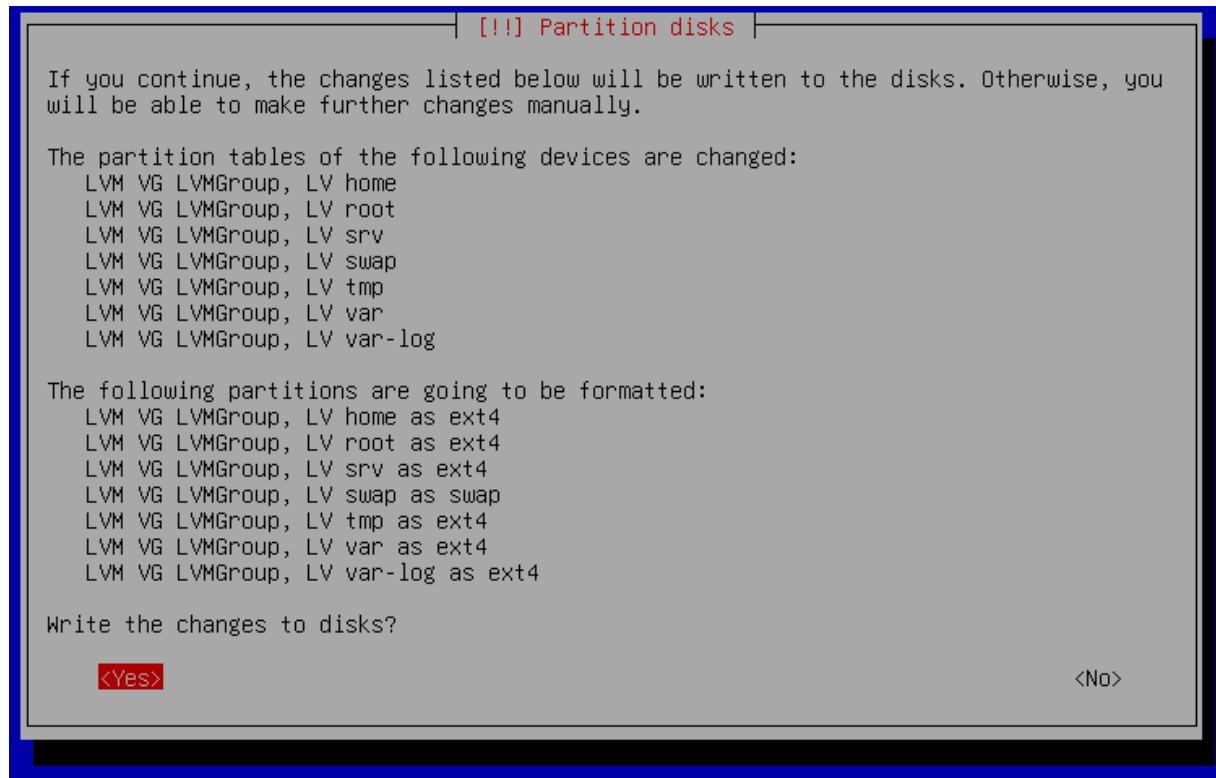


Buradaki işlemimiz bittikten sonra şöyle bir çıktımız oluyor:

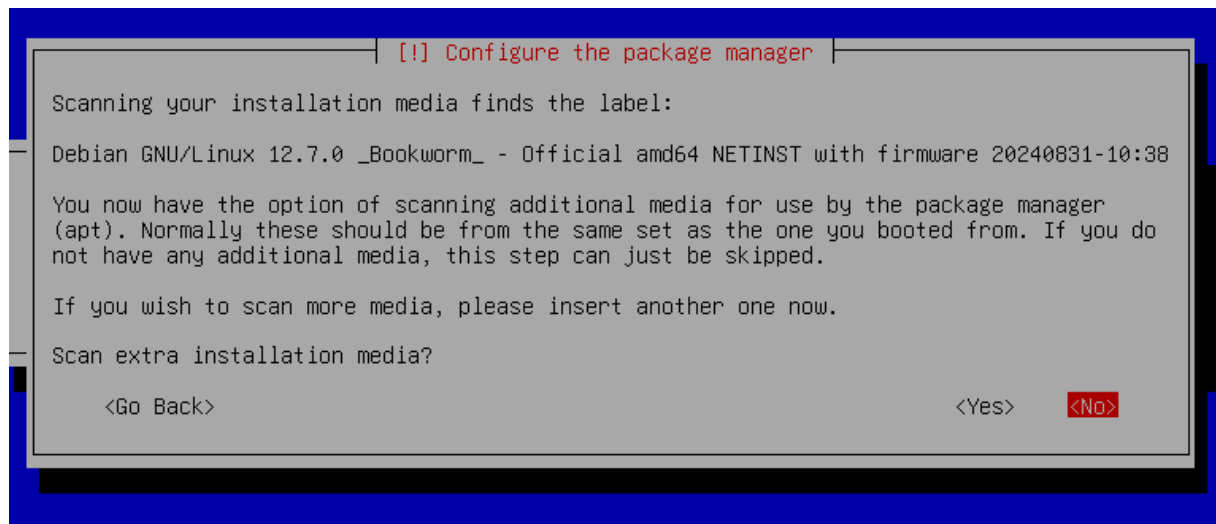
```
LVM VG LVMGroup, LV home - 1.3 GB Linux device-mapper (linear)
#1          1.3 GB      f  ext4      /home
LVM VG LVMGroup, LV root - 4.3 GB Linux device-mapper (linear)
#1          4.3 GB      f  ext4      /
LVM VG LVMGroup, LV srv - 1.3 GB Linux device-mapper (linear)
#1          1.3 GB      f  ext4      /srv
LVM VG LVMGroup, LV swap - 3.3 GB Linux device-mapper (linear)
#1          3.3 GB      f  swap      swap
LVM VG LVMGroup, LV tmp - 1.3 GB Linux device-mapper (linear)
#1          1.3 GB      f  ext4      /tmp
LVM VG LVMGroup, LV var - 1.3 GB Linux device-mapper (linear)
#1          1.3 GB      f  ext4      /var
LVM VG LVMGroup, LV var-log - 2.0 GB Linux device-mapper (linear)
#1          2.0 GB      f  ext4      /var/log
Encrypted volume (sda5_crypt) - 15.5 GB Linux device-mapper (crypt)
#1          15.5 GB      K  lvm
SCSI2 (0,0,0) (sda) - 16.1 GB ATA VBOX HARDDISK
#1 primary 575.7 MB  B  F  ext4      /boot
```

Sonrasında en alta inip Finish diyoruz.

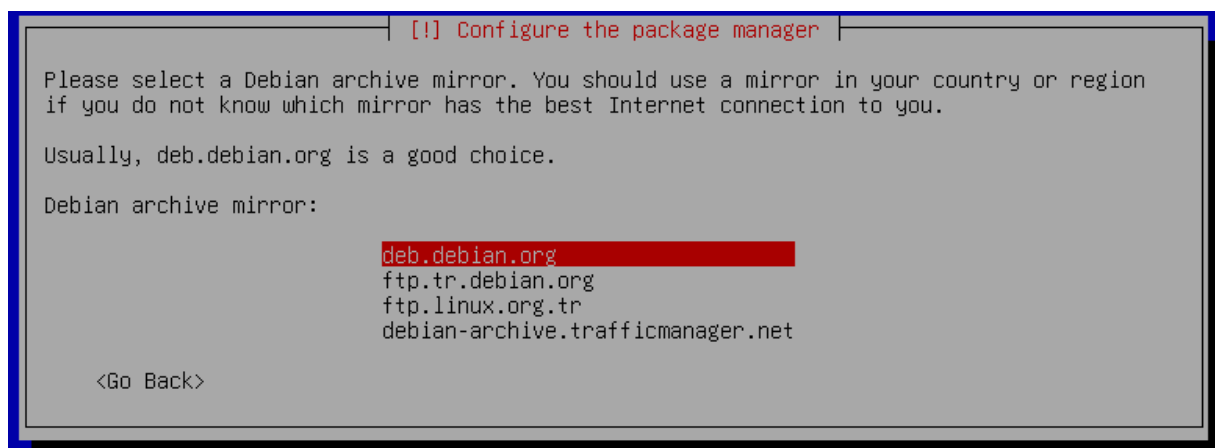
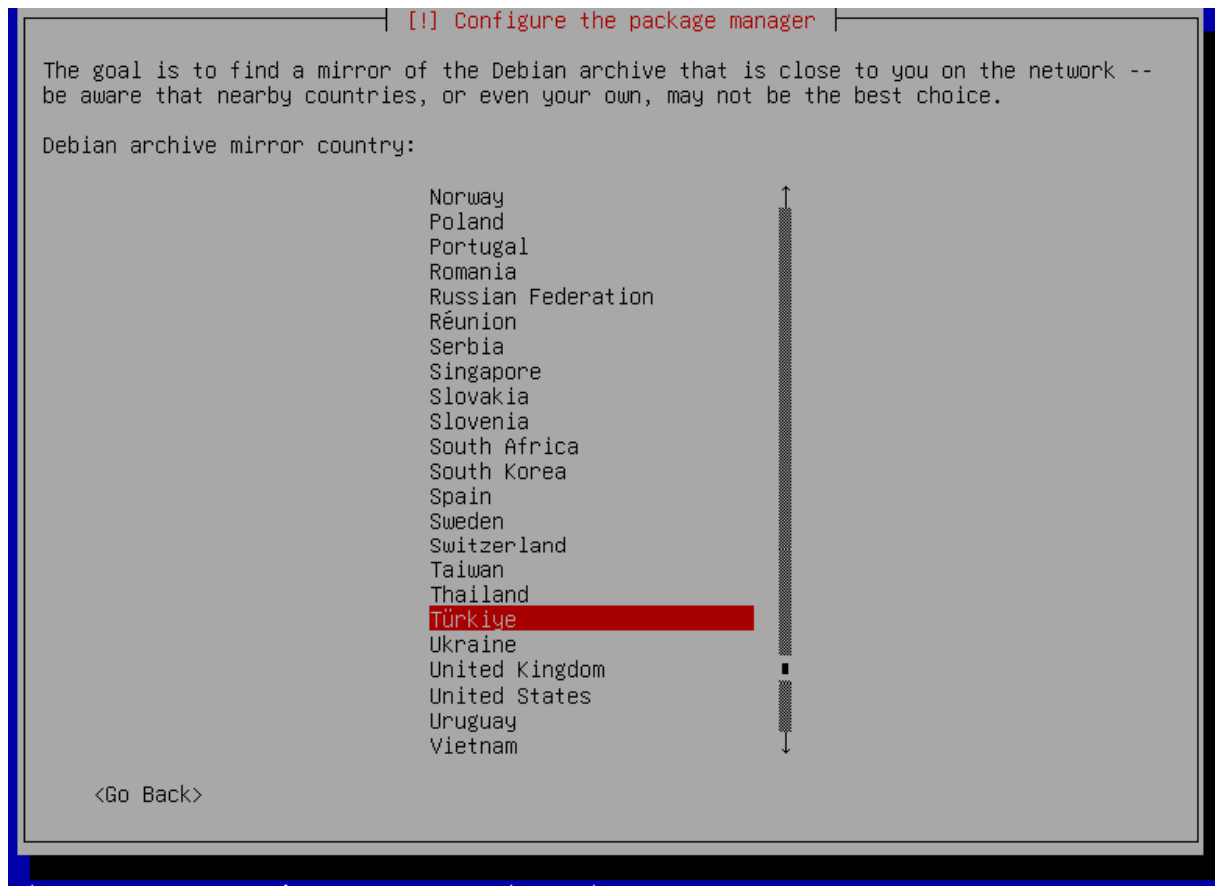
Bize bir onay sorusu soruyor. Yes diyip ilerliyoruz.



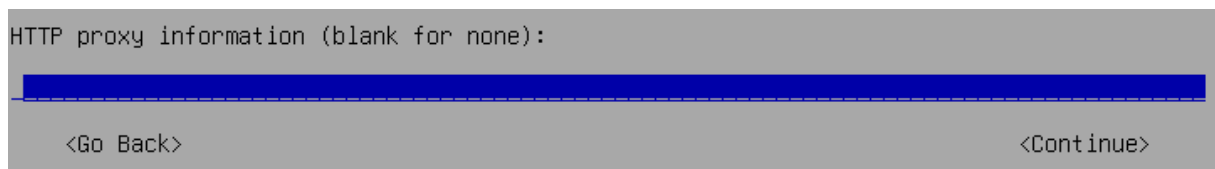
Bir onay ekranı gelecek. Ona no diyoruz.



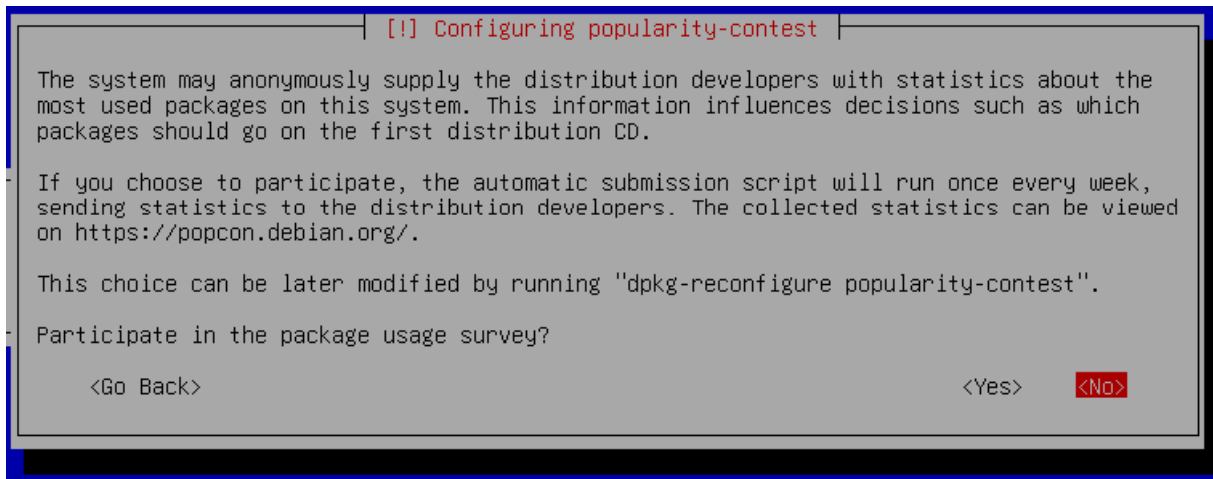
Şimdi ise debian kütüphanelerine hangi ülke sunucusu ile erişeceğimizi belirtmemiz gerekiyor. Ben buna Sırasıyla Türkiye ve "deb.debian.org" diyorum.



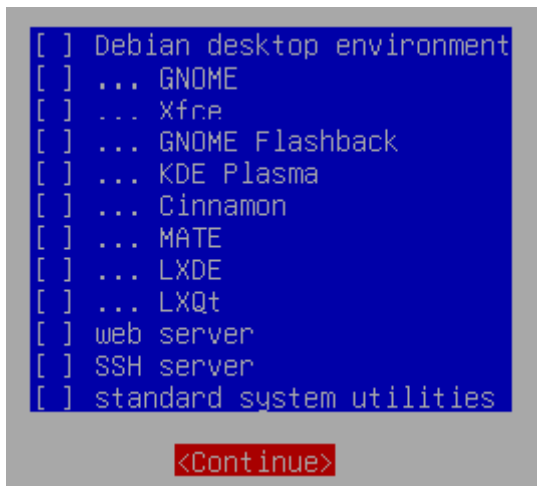
Sonrasında proxy bilgisi istiyor burayı boş bırakıyoruz.



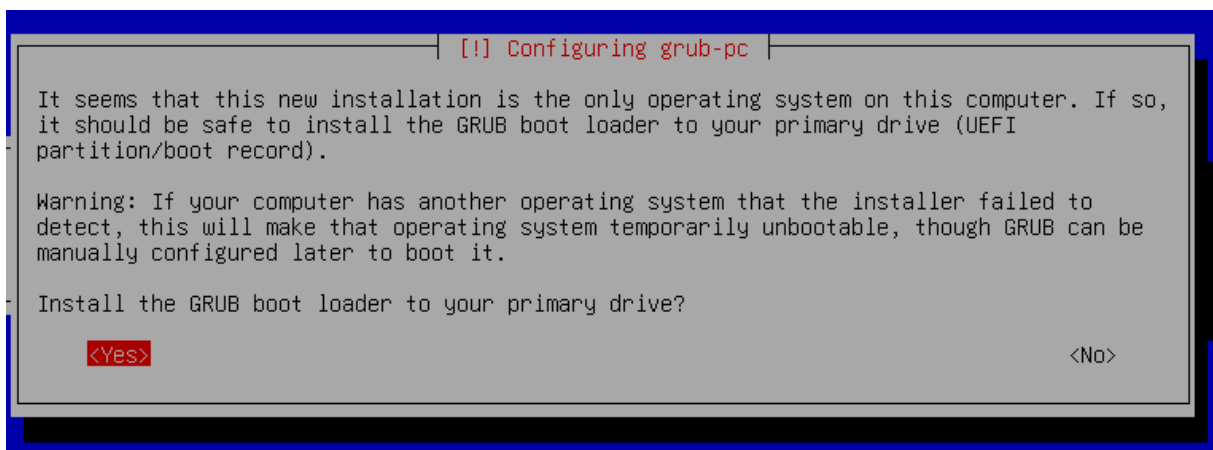
Paketleri kurup kurmayacağını soruyor buna da no diyoruz ve ilerliyoruz.



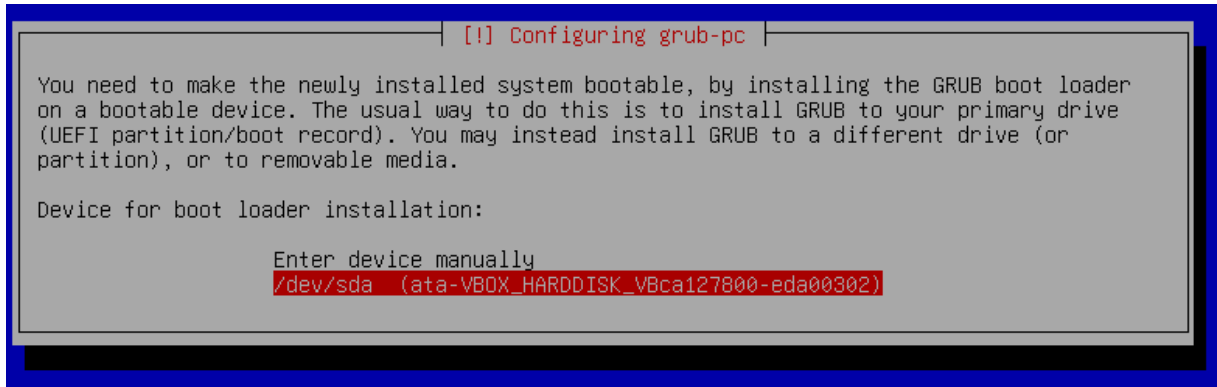
Bu adımda ki tüm yıldızları kaldırıp devam ediyoruz.



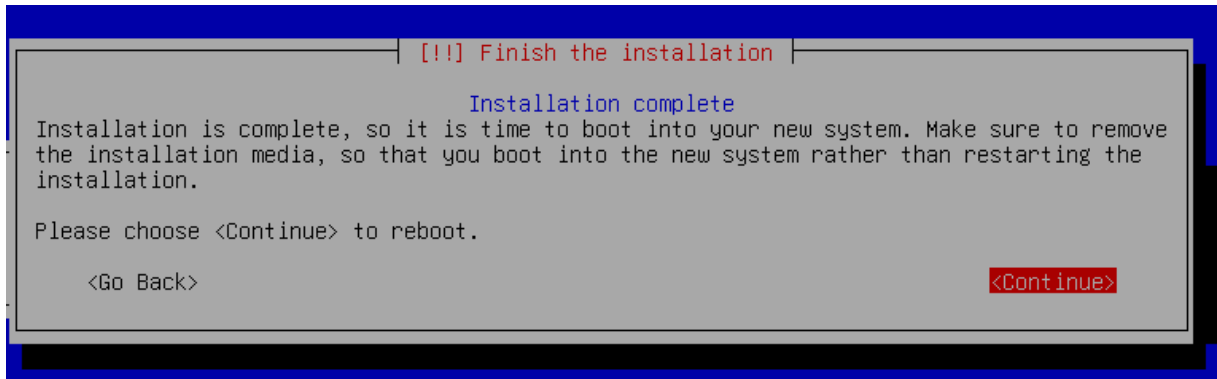
Karşımıza bir onay sorusu daha geliyor. Buna yes diyoruz.



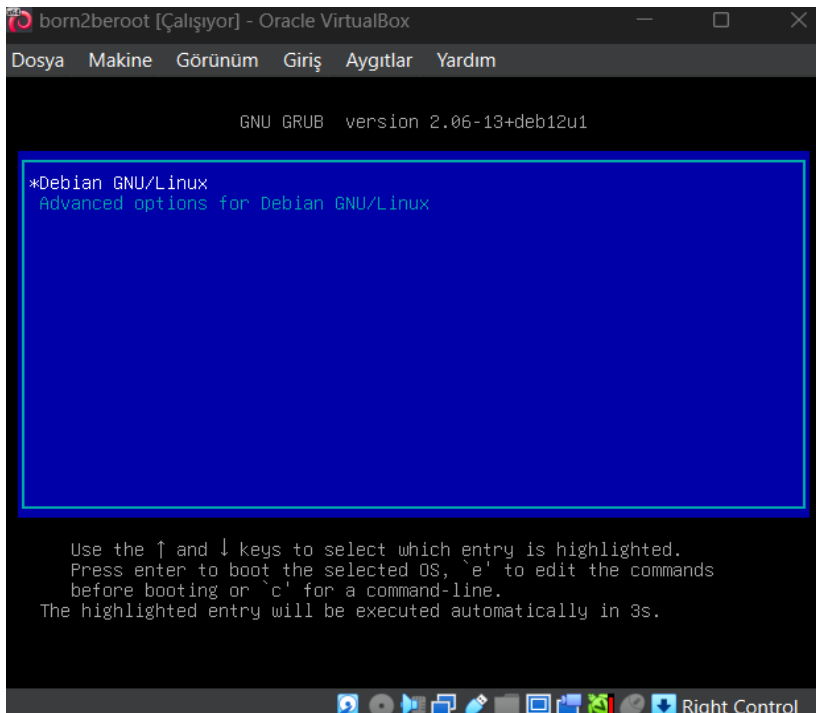
Boot loader için bir cihaz seçmemizi istiyor “/dev/sda” olanı seçiyoruz.



Son olarak yeniden başlatma istiyor. Continue diyoruz.

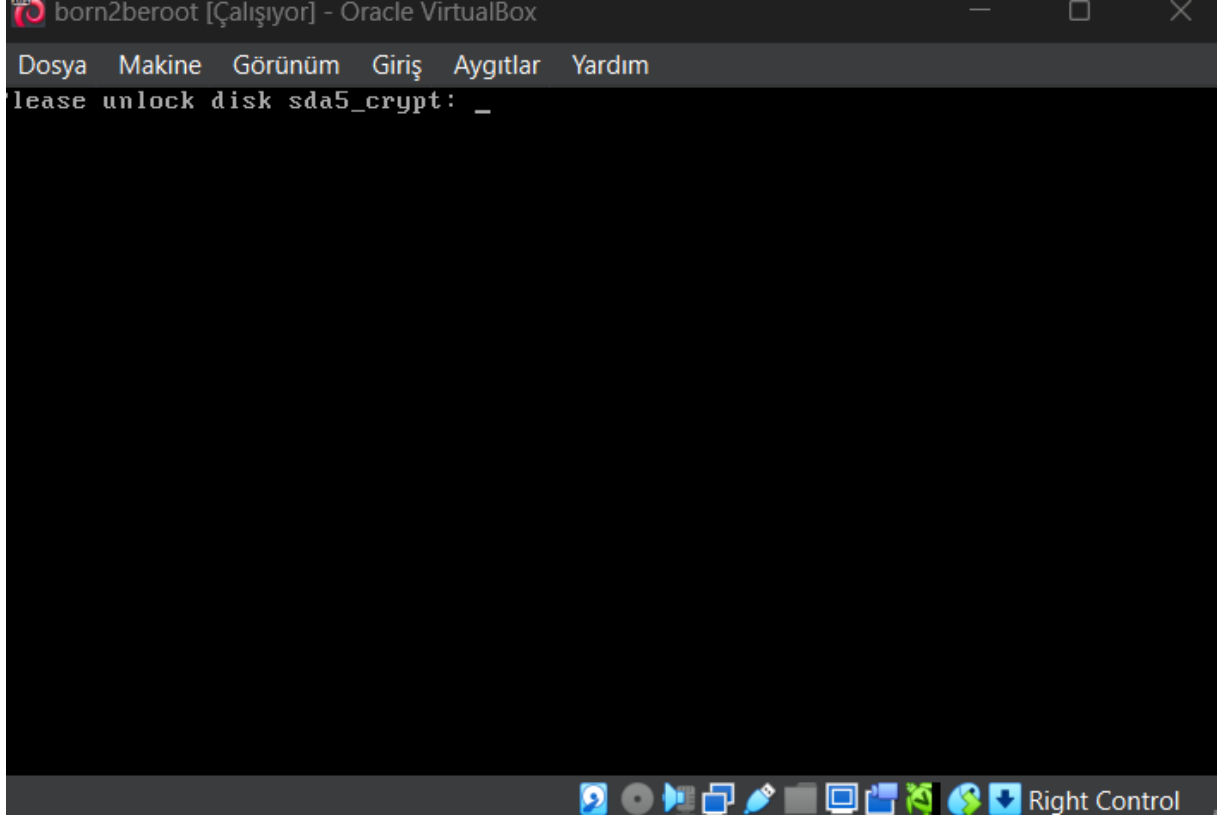


Aşağıdaki ekranı gördüğünüzde projenin kurulum kısmı bitmiş demektir.



SUDO, SSH, UFW AYARLARI

Artık ikinci aşamaya geçebiliriz. Bu aşamada ilk olarak diske girmemiz için şifre isteyecek. Bu şifre sizin root için oluşturduğunuz şifre.



Şimdi ise bizden açtığımız kullanıcı hesabını girmemizi istiyor.

```
Debian GNU/Linux 12 beergin42 tty1
beergin42 login: beergin
Password: _
```

Hesabımıza girdik:

```
The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
beergin@beergin42:~$
```

Şimdi yapmamız gereken şey sudo'yu indirmek. Bunun için öncelikle "su -" yazıp root şifremizi giriyoruz.

```
beergin@beergin42:~$ su -  
Password:  
root@beergin42:~# _
```

Sonrasında sırasıyla:

apt update -y

apt upgrade -y

apt install sudo -y yazıyoruz.

"-y" koyma sebebimiz indirme esnasındaki her soruya "yes" densin diye.

```
root@beergin42:~# apt update -y  
Hit:1 http://deb.debian.org/debian bookworm InRelease  
Hit:2 http://security.debian.org/debian-security bookworm-security InRelease  
Hit:3 http://deb.debian.org/debian bookworm-updates InRelease  
Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done  
All packages are up to date.  
root@beergin42:~# apt upgrade -y  
Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done  
Calculating upgrade... Done  
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.  
root@beergin42:~# apt install sudo -y  
Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done  
The following NEW packages will be installed:  
  sudo  
0 upgraded, 1 newly installed, 0 to remove and 0 not upgraded.  
Need to get 1,889 kB of archives.  
After this operation, 6,199 kB of additional disk space will be used.  
Get:1 http://deb.debian.org/debian bookworm/main amd64 sudo amd64 1.9.13p3-1+deb12u1 [1,889 kB]  
Fetched 1,889 kB in 1s (2,308 kB/s)  
Selecting previously unselected package sudo.  
(Reading database ... 23736 files and directories currently installed.)  
Preparing to unpack .../sudo_1.9.13p3-1+deb12u1_amd64.deb ...  
Unpacking sudo (1.9.13p3-1+deb12u1) ...  
Setting up sudo (1.9.13p3-1+deb12u1) ...  
Processing triggers for libc-bin (2.36-9+deb12u9) ...  
root@beergin42:~#
```

Sudo'nun inip inmediğini kontrol edelim. Bunun için komut satırına **dpkg -l | grep sudo** yazıyoruz.

```
root@beergin42:~# dpkg -l | grep sudo
ii  sudo                1.9.13p3-1+deb12u1      amd64        Provide limited super user privileges to specific users
root@beergin42:~#
```

Sudomuz inmiş.

Şimdi kullanıcıyı sudo grubuna ekleyelim. Komut satırına şunu yazıyoruz:

adduser <intrakullanıcıadınız> sudo

Benim için > adduser beergin sudo

```
root@beergin42:~# adduser beergin sudo
Adding user `beergin' to group `sudo' ...
Done.
root@beergin42:~# _
```

Kontrol edelim. Bunun için **getent group sudo** yazıyoruz.

```
root@beergin42:~# getent group sudo
sudo:x:27:beergin
root@beergin42:~# _
```

Kullanıcımız sudo grubuna eklenmiş. Şimdi de bu kullanıcıya sudo yetkileri verelim.

Komut satırına **sudo visudo** yazıyoruz.

Şöyle bir dosya açılıyor:

```
GNU nano 7.2 /etc/sudoers.tmp
#
# This file MUST be edited with the 'visudo' command as root.
#
# Please consider adding local content in /etc/sudoers.d/ instead of
# directly modifying this file.
#
# See the man page for details on how to write a sudoers file.
#
Defaults    env_reset
Defaults    mail_badpass
Defaults    secure_path="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin"

# This fixes CVE-2005-4890 and possibly breaks some versions of kdesu
# (#1011624, https://bugs.kde.org/show_bug.cgi?id=452532)
Defaults    use_pty

# This preserves proxy settings from user environments of root
# equivalent users (group sudo)
#Defaults:sudo env_keep += "http_proxy https_proxy ftp_proxy all_proxy no_proxy"

# This allows running arbitrary commands, but so does ALL, and it means
# different sudoers have their choice of editor respected.
#Defaults:sudo env_keep += "EDITOR"

# Completely harmless preservation of a user preference.
#Defaults:sudo env_keep += "GREP_COLOR"

# While you shouldn't normally run git as root, you need to with etckeeper
#Defaults:sudo env_keep += "GIT_AUTHOR_* GIT_COMMITTER_*"

# Per-user preferences; root won't have sensible values for them.
#Defaults:sudo env_keep += "EMAIL DEBEMAIL DEBFULLNAME"

# "sudo scp" or "sudo rsync" should be able to use your SSH agent.
#Defaults:sudo env_keep += "SSH_AGENT_PID SSH_AUTH_SOCK"

# Ditto for GPG agent
#Defaults:sudo env_keep += "GPG_AGENT_INFO"

# Host alias specification

# User alias specification

# Cmnd alias specification

# User privilege specification
```

Burada aşağıdaki satırı bulun:

```
# User privilege specification
root    ALL=(ALL:ALL) ALL
```

Bunun altına şu kodu ekleyin : **<intrakullanıcıadı> ALL=(ALL:ALL) ALL**

Benim için **> beergin ALL=(ALL:ALL) ALL**

```
# User privilege specification
root    ALL=(ALL:ALL) ALL
beergin ALL=(ALL:ALL) ALL
```

Kaydedip çıkmak için : CTRL+X e basın sonra Y'ye son olarak ENTER'a basın.

Değişikliklerin kaydedilmesi için komut satırına **reboot** yazıp enter'a basın.

Sudo'yu çalıştırmak için **sudo -v** kodunu yazıyoruz.

```
beergin@beergin42:~$ sudo -v
[sudo] password for beergin:
beergin@beergin42:~$ _
```

Artık komutlarımızın başına **sudo** koyarak yetkilerini kullanabiliriz. Test etmek için **sudo apt update** ve **sudo apt upgrade** komutlarını çalıştırabilirsiniz.

Artık sıra SSH'ı yüklemeye. **sudo login <kullanıcıadı>** ile root'tan kendi hesabınıza geçin. Ardından şu komutu yazın: **sudo apt install openssh-server -y**

Kurulumun başarıyla tamamlanıp tamamlanmadığına bakalım: **dpkg -l | grep ssh**

```
beergin@beergin42:~$ dpkg -l | grep ssh
ii  openssh-client      1:9.2p1-2+deb12u3 amd64      secure shell (SSH) client, for secure access to remote machines
ii  openssh-server      1:9.2p1-2+deb12u3 amd64      secure shell (SSH) server, for secure access from remote machines
ii  openssh-sftp-server 1:9.2p1-2+deb12u3 amd64      secure shell (SSH) sftp server module, for SFTP access from remote machines
beergin@beergin42:~$
```

Görüldüğü üzere kurulmuş. Şimdi de SSH'ın durumuna bakalım kapalıysa açalım.

Durumuna bakmak için : **sudo systemctl status ssh** ya da **sudo service ssh status**

```
beergin@beergin42:~$ sudo systemctl status ssh
• ssh.service - OpenBSD Secure Shell server
   Loaded: loaded (/lib/systemd/system/ssh.service; enabled; preset: enabled)
   Active: active (running) since Sat 2024-11-16 12:05:17 CST; 2min 15s ago
     Docs: man:sshd(8)
           man:sshd_config(5)
  Process: 564 ExecStartPre=/usr/sbin/sshd -t (code=exited, status=0/SUCCESS)
 Main PID: 577 (sshd)
    Tasks: 1 (limit: 2302)
   Memory: 6.4M
      CPU: 28ms
  CGroup: /system.slice/ssh.service
          └─577 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"

Nov 16 12:05:17 beergin42 systemd[1]: Starting ssh.service - OpenBSD Secure Shell server...
Nov 16 12:05:17 beergin42 sshd[577]: Server listening on 0.0.0.0 port 22.
Nov 16 12:05:17 beergin42 sshd[577]: Server listening on :: port 22.
Nov 16 12:05:17 beergin42 systemd[1]: Started ssh.service - OpenBSD Secure Shell server.
beergin@beergin42:~$ _
```

Bizde açık fakat kapalı olsaydı şu komutu yazarak açacaktık : **sudo service ssh start**

```
beergin@beergin42:~$ sudo service ssh start
beergin@beergin42:~$ sudo service ssh status
• ssh.service - OpenBSD Secure Shell server
   Loaded: loaded (/lib/systemd/system/ssh.service; enabled; preset: enabled)
   Active: active (running) since Sat 2024-11-16 12:10:16 CST; 991ms ago
     Docs: man:sshd(8)
           man:sshd_config(5)
  Process: 679 ExecStartPre=/usr/sbin/sshd -t (code=exited, status=0/SUCCESS)
 Main PID: 682 (sshd)
    Tasks: 1 (limit: 2302)
   Memory: 1.4M
      CPU: 16ms
  CGroup: /system.slice/ssh.service
          └─682 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"

Nov 16 12:10:16 beergin42 systemd[1]: Starting ssh.service - OpenBSD Secure Shell server...
Nov 16 12:10:16 beergin42 sshd[682]: Server listening on 0.0.0.0 port 22.
Nov 16 12:10:16 beergin42 sshd[682]: Server listening on :: port 22.
Nov 16 12:10:16 beergin42 systemd[1]: Started ssh.service - OpenBSD Secure Shell server.
beergin@beergin42:~$
```

SSH'ı tekrar başlatmak içinse **sudo service ssh restart** yapmamız yeterli olacaktır.

```
beergin@beergin42:~$ sudo service ssh restart
beergin@beergin42:~$ sudo service ssh status
• ssh.service - OpenBSD Secure Shell server
   Loaded: loaded (/lib/systemd/system/ssh.service; enabled; preset: enabled)
   Active: active (running) since Sat 2024-11-16 12:13:12 CST; 2s ago
     Docs: man:sshd(8)
           man:sshd_config(5)
   Process: 696 ExecStartPre=/usr/sbin/sshd -t (code=exited, status=0/SUCCESS)
  Main PID: 698 (sshd)
    Tasks: 1 (limit: 2302)
   Memory: 1.4M
      CPU: 16ms
   CGroup: /system.slice/ssh.service
           └─698 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"

Nov 16 12:13:12 beergin42 systemd[1]: Starting ssh.service - OpenBSD Secure Shell server...
Nov 16 12:13:12 beergin42 sshd[698]: Server listening on 0.0.0.0 port 22.
Nov 16 12:13:12 beergin42 sshd[698]: Server listening on :: port 22.
Nov 16 12:13:12 beergin42 systemd[1]: Started ssh.service - OpenBSD Secure Shell server.
beergin@beergin42:~$
```

PDF’te bizden 4242 portu üzerinden bağlanılması isteniyordu.

Bunun için **sudo nano /etc/ssh/sshd_config** yazarak config dosyasına giriyoruz.

```
GNU nano 7.2 /etc/ssh/sshd_config
# This is the sshd server system-wide configuration file. See
# sshd_config(5) for more information.

# This sshd was compiled with PATH=/usr/local/bin:/usr/bin:/bin:/usr/games

# The strategy used for options in the default sshd_config shipped with
# OpenSSH is to specify options with their default value where
# possible, but leave them commented. Uncommented options override the
# default value.

Include /etc/ssh/sshd_config.d/*.conf
#Port 22
#AddressFamily any
#ListenAddress 0.0.0.0
#ListenAddress ::

#HostKey /etc/ssh/ssh_host_rsa_key
#HostKey /etc/ssh/ssh_host_ecdsa_key
#HostKey /etc/ssh/ssh_host_ed25519_key

# Ciphers and keying
#RekeyLimit default none

# Logging
#SyslogFacility AUTH
#LogLevel INFO

# Authentication:
#AuthenticationMethods publickey
#PermitRootLogin prohibit-password
#StrictModes yes
#MaxAuthTries 6
#MaxSessions 10

#PubkeyAuthentication yes

# Expect .ssh/authorized_keys2 to be disregarded by default in future.
#AuthorizedKeysFile .ssh/authorized_keys .ssh/authorized_keys2

#AuthorizedPrincipalsFile none

#AuthorizedKeysCommand none
#AuthorizedKeysCommandUser nobody
```

Bu dosyadan işaretlediğim iki yerin de başındaki “#” işaretini kaldırıp aşağıdaki gibi düzenliyoruz:

#Port 22 > Port 4242

#PermitRootLogin prohibit-password > PermitRootLogin no

```
Port 4242
#AddressFamily any
#ListenAddress 0.0.0.0
#ListenAddress ::

#HostKey /etc/ssh/ssh_host_rsa_key
#HostKey /etc/ssh/ssh_host_ecdsa_key
#HostKey /etc/ssh/ssh_host_ed25519_key

# Ciphers and keying
#RekeyLimit default none

# Logging
#SyslogFacility AUTH
#LogLevel INFO

# Authentication:

#LoginGraceTime 2m
PermitRootLogin no
#StrictModes yes
```

CTRL + X'e basıp kaydedip çıkıyoruz. **sudo service ssh restart** ile servisi tekrar başlatıp **sudo service ssh status** ile de durumuna bakalım.

```
beergin@beergin42:~$ sudo service ssh status
• ssh.service - OpenBSD Secure Shell server
   Loaded: loaded (/lib/systemd/system/ssh.service; enabled; preset: enabled)
   Active: active (running) since Sat 2024-11-16 12:22:11 CST; 36s ago
     Docs: man:sshd(8)
           man:sshd_config(5)
   Process: 568 ExecStartPre=/usr/sbin/sshd -t (code=exited, status=0/SUCCESS)
  Main PID: 580 (sshd)
    Tasks: 1 (limit: 2302)
   Memory: 6.4M
      CPU: 24ms
   CGroup: /system.slice/ssh.service
           └─580 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"

Nov 16 12:22:11 beergin42 systemd[1]: Starting ssh.service - OpenBSD Secure Shell server...
Nov 16 12:22:11 beergin42 sshd[580]: Server listening on 0.0.0.0 port 4242.
Nov 16 12:22:11 beergin42 sshd[580]: Server listening on :: port 4242.
Nov 16 12:22:11 beergin42 systemd[1]: Started ssh.service - OpenBSD Secure Shell server.
beergin@beergin42:~$
```

Görüldüğü üzere 4242 portu dinleniyor. Eğer sizde bu çıktı gözüküyorsa **sudo reboot** ile tekrar başlatıp durumuna öyle bakın.

sudo grep Port /etc/ssh/sshd_config ile de port ayarının doğru olup olmadığına bakabiliriz.

```
beergin@beergin42:~$ sudo grep Port /etc/ssh/sshd_config
Port 4242
#GatewayPorts no
beergin@beergin42:~$
```

SSH ayarımızı hallettik sıra güvenlik duvarımızda. Bunun için **sudo apt-get install ufw -y** komutunu çalıştırıyoruz. UFW indikten sonra komut satırına

dpkg -l | grep ufw yaparak inip inmediğini kontrol ediyoruz.

```
beergin@beergin42:~$ dpkg -l | grep ufw
ii  ufw          0.36.2-1      all          program for managing a Netfilter firewall
beergin@beergin42:~$ _
```

Doğru bir şekilde inmiş, **sudo ufw enable** ile etkinleştirip **sudo systemctl status ufw** yazıp durumuna bakalım.

```
beergin@beergin42:~$ sudo ufw enable
Firewall is active and enabled on system startup
beergin@beergin42:~$ sudo systemctl status ufw
* ufw.service - Uncomplicated firewall
   Loaded: loaded (/lib/systemd/system/ufw.service; enabled; preset: enabled)
   Active: inactive (dead)
     Docs: man:ufw(8)
beergin@beergin42:~$ _
```

Aktif. Şimdi güvenlik duvarının kurallarına SSH'ı ve 4242 portunu eklemeliyiz bunun için sırasıyla şu kodları çalıştırıyoruz:

sudo ufw allow ssh

sudo ufw allow 4242

```
beergin@beergin42:~$ sudo ufw allow ssh
Rule added
Rule added (v6)
beergin@beergin42:~$ sudo ufw allow 4242
Rule added
Rule added (v6)
beergin@beergin42:~$
```

Bu sayede artık SSH bağlantısına ve 4242 portundan gelen bağlantılara izin verdik. Doğrulamak için **sudo ufw status numbered**

```
beergin@beergin42:~$ sudo ufw status numbered
Status: active

    To Action From
    --
[ 1] 22/tcp ALLOW IN Anywhere
[ 2] 4242 ALLOW IN Anywhere
[ 3] 22/tcp (v6) ALLOW IN Anywhere (v6)
[ 4] 4242 (v6) ALLOW IN Anywhere (v6)

beergin@beergin42:~$
```

Kuralı silmek istersek şu kodu yazmamız gerekiyor : **sudo ufw delete <numara>**

Buradaki numara kuralların başındaki sayılar (Örn : 1).

22 Portlarına ihtiyacımız yok biz zaten 4242 portunu kullanacağız o yüzden 22 portlarını silelim bunun için : **sudo ufw delete 3**

Bizden onay isteyecek “y” diyoruz.

```
beergin@beergin42:~$ sudo ufw delete 3
Deleting:
  allow 22
Proceed with operation (y|n)? y
Rule deleted (v6)
beergin@beergin42:~$ sudo ufw status numbered
Status: active

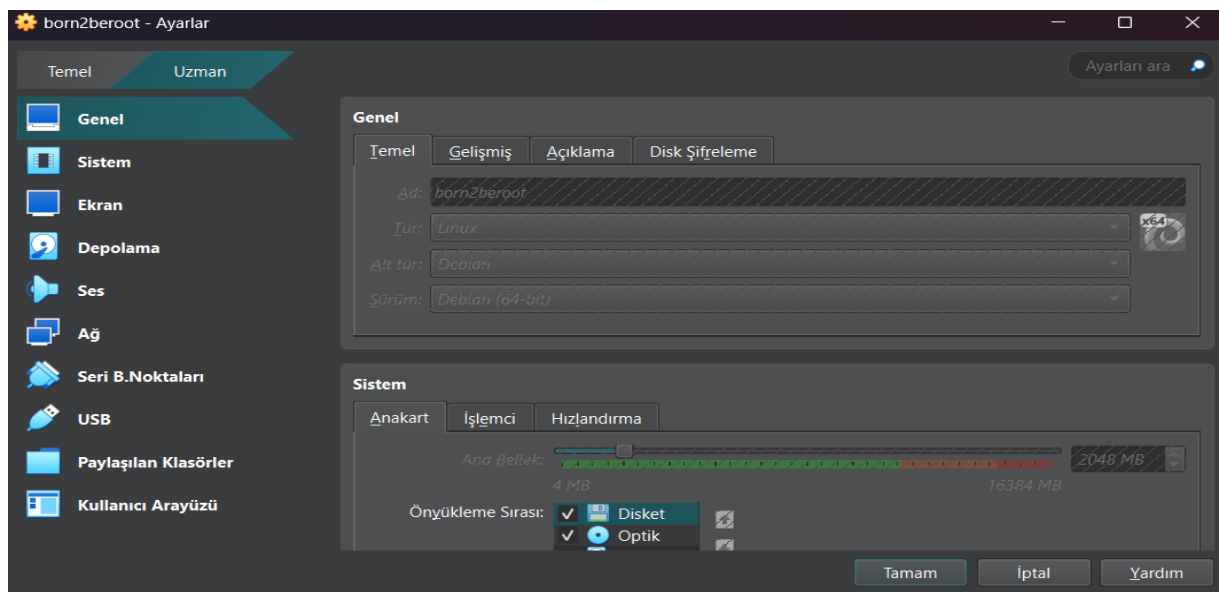
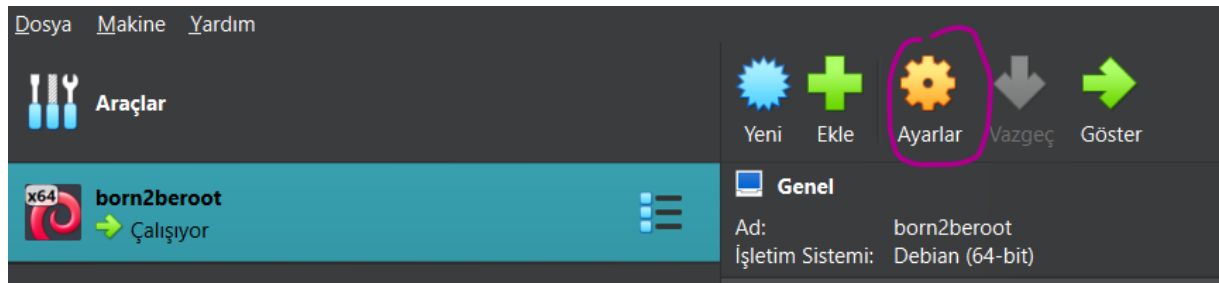
      To      Action      From
      --      -
[ 1] 4242     ALLOW IN    Anywhere
[ 2] 4242 (v6) ALLOW IN    Anywhere (v6)

beergin@beergin42:~$
```

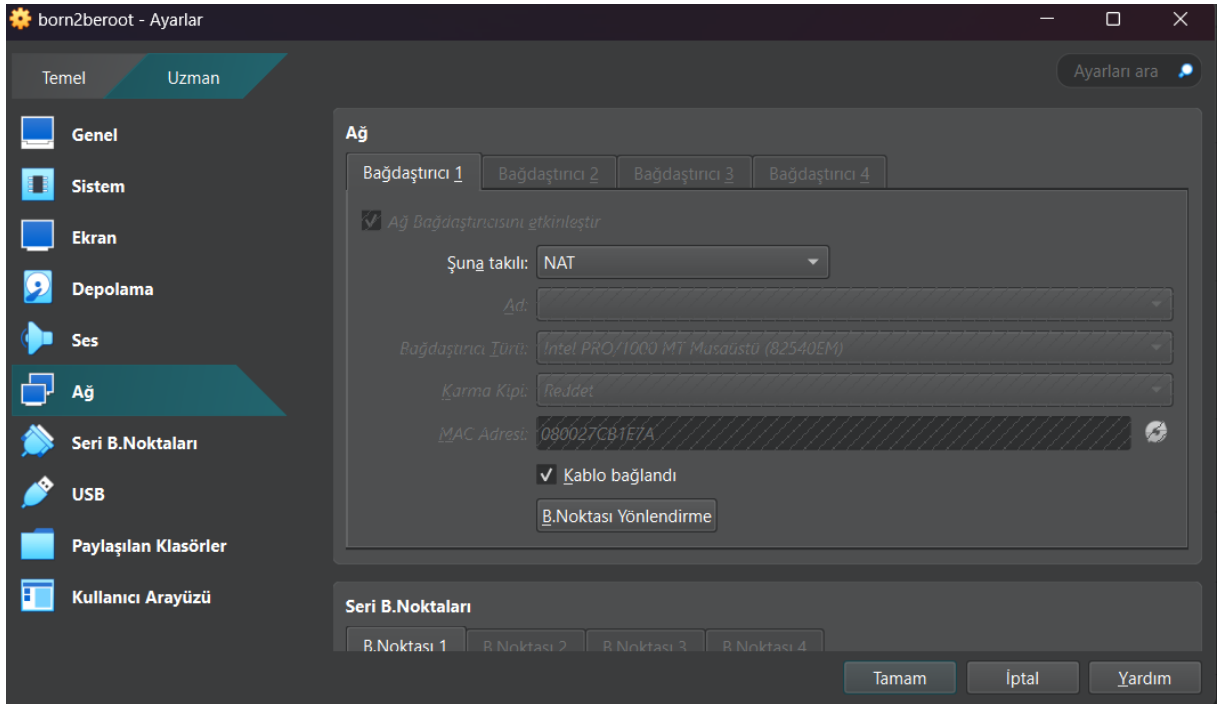
İstedığımız gibi sadece 4242 portları var.

Artık kendi bilgisayarımızın terminali ile sanal makinaya bağlanma zamanı geldi.

İlk olarak Oracle Virtual Box’u açalım ve üstten “Ayarlar” a basalım.

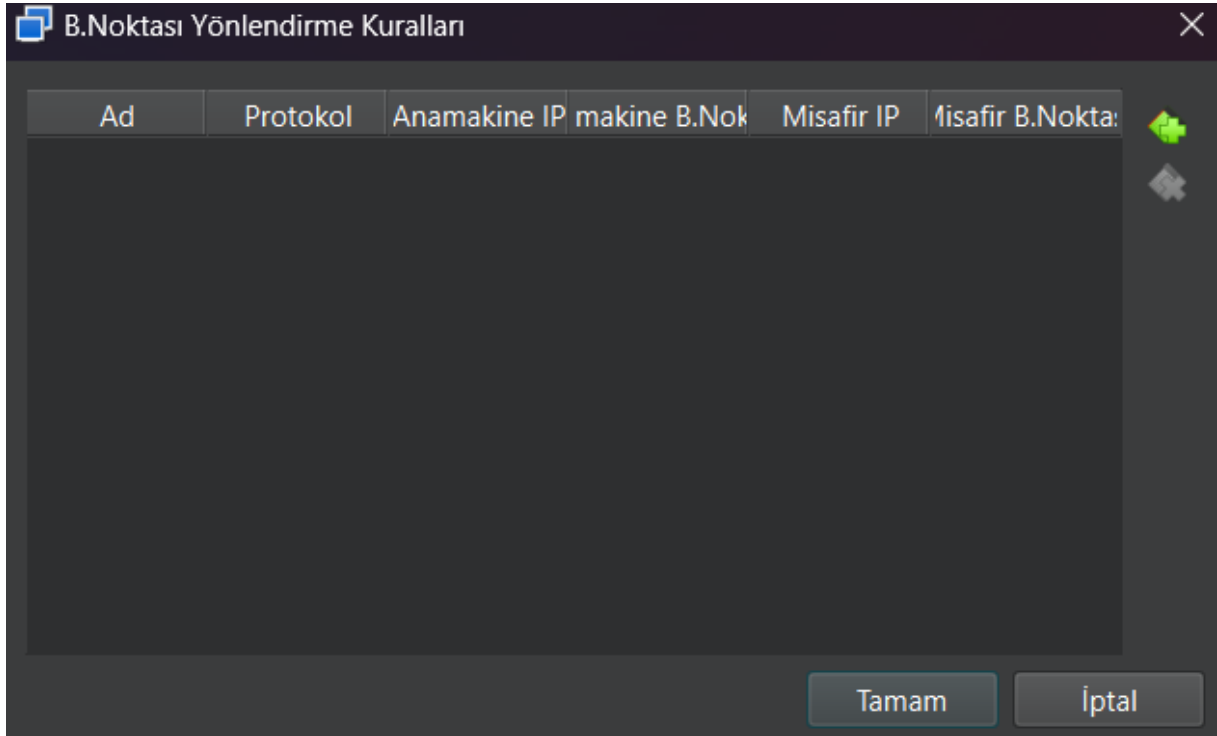


Sol menüden “Ağ” sekmesine gidelim.

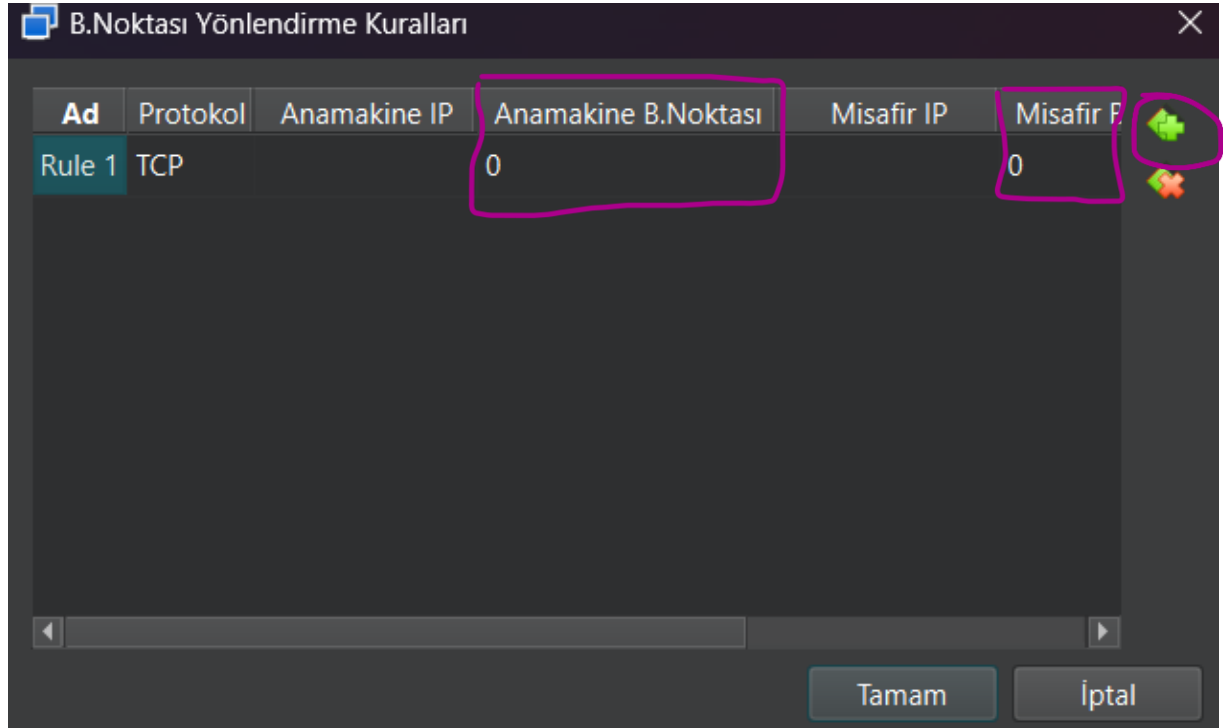


“Şuna takılı” kısmını “NAT” olarak ayarlayıp alttan “B.Noktası Yönlendirme” ye basıyoruz.

Karşımıza aşağıdaki gibi bir pencere gelecek:

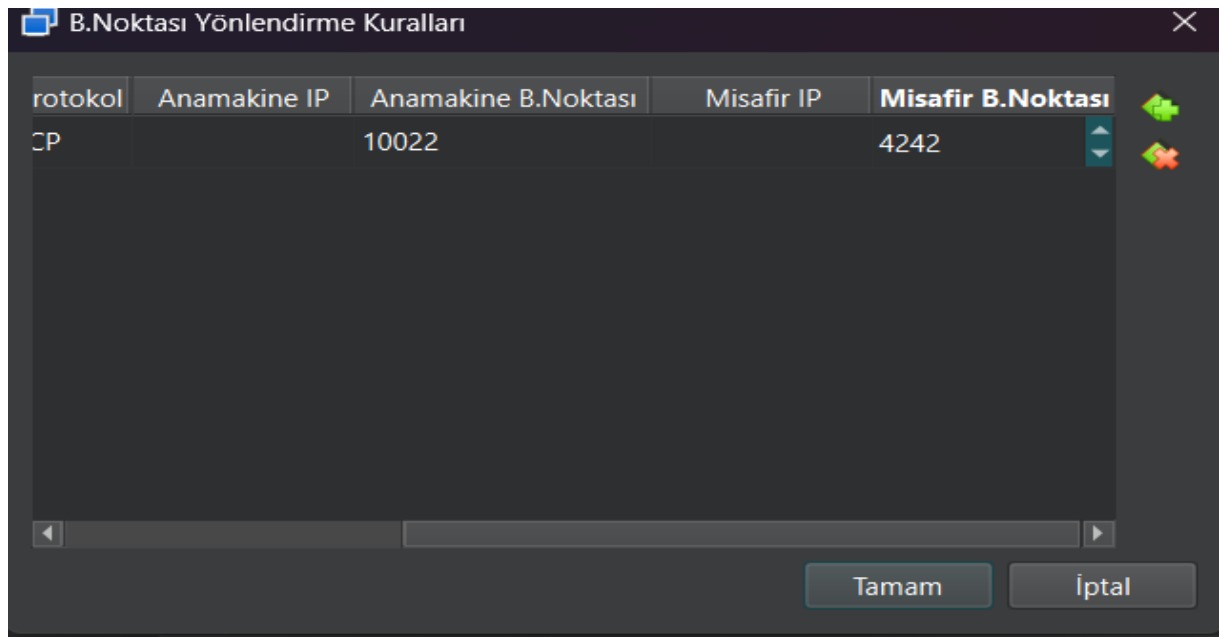


Sağ üstteki yeşil butona basıp “Anamakine B.Noktası” ve “Misafir B.Noktası” kısımlarını dolduracağız.



Burayla alakalı bir not düşmek istiyorum. Anamakine Port’una çoğu kişi 4242 yazmakta. Fakat şöyle bir sorun var. PDF’te istenen şey tüm bağlantıların 4242 portuna yönlendirilmesi. Siz anamakine portunu 4242 yaparsanız çalışmama ihtimali yüksek olur çünkü bu port çok kullanılıyor. Bu yüzden 10000 üzerinde bir port girmeniz daha iyi olur. Misafir port kısmı ise 4242 olacak. Burayı hallettiğimize göre devam edelim.

Ben anamakine portunu 10022 olarak gireceğim.



Buraya “Tamam” dedikten sonra sanal makinamızın komut satırına dönüp **sudo reboot** yazıp tekrar çalıştıralım. Açıldıktan sonra SSH’in ve UFW’nin çalışıp çalışmadığına bakalım.

```
Last login: Sat Nov 16 12:22:31 CST 2024 on tty1
beergin@beergin42:~$ sudo -v
[sudo] password for beergin:
beergin@beergin42:~$ sudo systemctl status ssh
• ssh.service - OpenBSD Secure Shell server
   Loaded: loaded (/lib/systemd/system/ssh.service; enabled; preset: enabled)
   Active: active (running) since Sat 2024-11-16 13:06:55 CST; 22s ago
     Docs: man:sshd(8)
           man:sshd_config(5)
   Process: 674 ExecStartPre=/usr/sbin/sshd -t (code=exited, status=0/SUCCESS)
   Main PID: 677 (sshd)
     Tasks: 1 (limit: 2302)
    Memory: 6.5M
       CPU: 26ms
   CGroup: /system.slice/ssh.service
           └─677 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"

Nov 16 13:06:55 beergin42 systemd[1]: Starting ssh.service - OpenBSD Secure Shell server...
Nov 16 13:06:55 beergin42 sshd[677]: Server listening on 0.0.0.0 port 4242.
Nov 16 13:06:55 beergin42 sshd[677]: Server listening on :: port 4242.
Nov 16 13:06:55 beergin42 systemd[1]: Started ssh.service - OpenBSD Secure Shell server.
beergin@beergin42:~$ sudo systemctl status ufw
• ufw.service - Uncomplicated firewall
   Loaded: loaded (/lib/systemd/system/ufw.service; enabled; preset: enabled)
   Active: active (exited) since Sat 2024-11-16 13:06:55 CST; 26s ago
     Docs: man:ufw(8)
   Process: 481 ExecStart=/lib/ufw/ufw-init start quiet (code=exited, status=0/SUCCESS)
   Main PID: 481 (code=exited, status=0/SUCCESS)
       CPU: 77ms

Nov 16 13:06:54 beergin42 systemd[1]: Starting ufw.service - Uncomplicated firewall...
Nov 16 13:06:55 beergin42 systemd[1]: Finished ufw.service - Uncomplicated firewall.
beergin@beergin42:~$ _
```

İki hizmetimiz de başarılı bir şekilde çalışıyor. Şimdi kendi bilgisayarımızın terminalini açıp şu kodu giriyoruz:

ssh -p <anamakine kısmına girdiğiniz port> <intrakullanıcıadı>@127.0.0.1

Benim için > **ssh -p 10022 beergin@127.0.0.1**

```
C:\Users\ggolg>ssh -p 10022 beergin@127.0.0.1
The authenticity of host '[127.0.0.1]:10022 ([127.0.0.1]:10022)' can't be established.
ED25519 key fingerprint is SHA256:as9UFp3SosIyt9oZ5cErp8HDSDRdDftawfAOMa63J5k.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '[127.0.0.1]:10022' (ED25519) to the list of known hosts.
beergin@127.0.0.1's password:
Linux beergin42 6.1.0-27-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.115-1 (2024-11-01) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Sat Nov 16 13:07:02 2024
beergin@beergin42:~$ |
```

Artık kendi bilgisayarımın terminali üzerinden projemi devam ettirebilirim.

Terminal bağlantısını kesmek için terminalinizin komut satırına “exit” yazabilirsiniz.

```
beergin@beergin42:~$ exit
logout
Connection to 127.0.0.1 closed.

C:\Users\ggolg>
```

Şimdi biraz daha detaylara girelim. İlk olarak sunucu ismimizi kontrol edip değiştirelim.

Terminalimize **sudo hostname** yazıyoruz. Bu komut hostname yani sunucu adlarımızı bize gösteriyor.

```
beergin@beergin42:~$ sudo hostname
beergin42
beergin@beergin42:~$ |
```

En başta “KURULUM” aşamasında girdiğimiz hostname’i bize gösterdi. Bunu değiştirelim. Komut satırına **sudo hostnamectl set-hostname beergin43** yazıyoruz. Bu komuttaki “beergin43” kısmına ne yazarsanız yeni hostname o olur. Yazdıktan sonra **sudo hostname** ile kontrol edelim.

```
beergin@beergin42:~$ sudo hostname
sudo: unable to resolve host beergin43: Name or service not known
beergin43
beergin@beergin42:~$ |
```

İlk satırda bir mesaj verdi. Bu mesajda değiştirdiğimiz hostname’in bilinmediğini söylüyor. Bunu söyleme sebebi “hosts” dosyasındaki hostname ile aynı olmaması. Bu yüzden “hosts” dosyasına giriyoruz: **sudo nano /etc/hosts**

```
beergin@beergin42:~$ sudo nano /etc/hosts
```

Bu komutu çalıştırdığımızda aşağıdaki gibi bir dosya açılacaktır:

```
GNU nano 7.2 /etc/hosts
127.0.0.1    localhost
127.0.1.1    beergin42

# The following lines are desirable for IPv6 capable hosts
::1         localhost ip6-localhost ip6-loopback
ff02::1     ip6-allnodes
ff02::2     ip6-allrouters
```

Biz hostname'i "beergin43" olarak değiştirsek de dosyadaki name hala "beergin42". Bunu da değiştiriyoruz.

```
GNU nano 7.2
127.0.0.1    localhost
127.0.1.1    beergin43|
```

Dosyayı kaydedip çıkalım ve tekrar kontrol edelim.

```
beergin@beergin42:~$ sudo hostname
beergin43
beergin@beergin42:~$ |
```

Artık uyarı mesajı vermeden hostname bilgisini veriyor. Terminale "reboot" yazarak da değişikliklerin kaydedilmesini sağlayalım. Terminal yeniden başladığında "beergin@beergin42" olan kısım güncellenmiş oldu:

```
beergin@beergin43:~$ |
```

Bundan sonraki adımda sudoyu yapılandırcaz. İlk olarak sudo'nun loglarının tutulacağı bir klasör oluşturuyoruz. Bunu **sudo mkdir /var/log/sudo** ile yapıyoruz.

```
beergin@beergin43:~$ sudo mkdir /var/log/sudo
[sudo] password for beergin:
beergin@beergin43:~$
```

Şimdi de yapılandırma dosyamıza girelim : **sudo visudo**

Dosyamız açıldığında sırasıyla şu kodları ekliyoruz:

Defaults log_input,log_output : Sudo ile yapılan işlemlerin girdilerinin ve çıktılarının loglarını tutması için

Defaults logfile="/var/log/sudo/sudo.log" : Sudo işlemlerinin loglarının kaydedileceği yeri belirlemek için

Defaults requiretty : TTY modunu aktif etmek için

Defaults secure_path="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/snap/bin" : PATH'in değiştirilme ihtimaline karşılık sadece bizim belirlediğimiz güvenli dizinlerden çalışması için

Defaults passwd_tries=3 : sudo için bir şifre istediğinde yanlış girilme durumunda en fazla kaç kez yanlış girme hakkı olduğunu ayarlamak için

Defaults badpass_message="<belirttiğiniz hata mesajı>" : Hatalı şifre girildiğinde gösterilecek olan mesaj.

Tüm bu eklemelerden sonra dosyamızın son hali böyle olmalı:

```
GNU nano 7.2 /etc/sudoers.tmp *
#
# This file MUST be edited with the 'visudo' command as root.
#
# Please consider adding local content in /etc/sudoers.d/ instead of
# directly modifying this file.
#
# See the man page for details on how to write a sudoers file.
#
Defaults      env_reset
Defaults      log_input,log_output
Defaults      logfile="/var/log/sudo/sudo.log"
Defaults      passwd_tries=3
Defaults      badpass_message="Yanlis sifre girdiniz! Tekrar deneyin!"
Defaults      requiretty
Defaults      mail_badpass
Defaults      secure_path="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/snap/bin"

# This fixes CVE-2005-4890 and possibly breaks some versions of kdesu
# (#1011624, https://bugs.kde.org/show_bug.cgi?id=452532)
Defaults      use_pty

# This preserves proxy settings from user environments of root
# equivalent users (group sudo)
#Defaults:%sudo env_keep += "http_proxy https_proxy ftp_proxy all_proxy no_proxy"

# This allows running arbitrary commands, but so does ALL, and it means
```

NOT: KULLANILAN KODLAR PDF'İN SONUNDA DETAYLIÇA AÇIKLANACAKTIR.

Şimdi ise şifremiz için güçlü bir politika ayarlayalım.

Önce komut satırında **sudo nano /etc/login.defs** komutunu çalıştırıyoruz. Karşımıza şöyle bir ekran çıkıyor:

```
GNU nano 7.2 /etc/login.defs
#
# /etc/login.defs - Configuration control definitions for the login package.
#
# Three items must be defined: MAIL_DIR, ENV_SUPATH, and ENV_PATH.
# If unspecified, some arbitrary (and possibly incorrect) value will
# be assumed. All other items are optional - if not specified then
# the described action or option will be inhibited.
#
# Comment lines (lines beginning with "#") and blank lines are ignored.
#
# Modified for Linux. --marekm

# REQUIRED for useradd/userdel/usermod
#   Directory where mailboxes reside, _or_ name of file, relative to the
#   home directory. If you _do_ define MAIL_DIR and MAIL_FILE,
#   MAIL_DIR takes precedence.
#
# Essentially:
#   - MAIL_DIR defines the location of users mail spool files
#     (for mbox use) by appending the username to MAIL_DIR as defined
#     below.
#   - MAIL_FILE defines the location of the users mail spool files as the
#     fully-qualified filename obtained by prepending the user home
#     directory before $MAIL_FILE
#
# NOTE: This is required for defining a user's MAIL environment variable
```

Şimdi ise dosyanın en altlarına giderek aşağıdaki kısmı bulun:


```
# Password aging controls:
#
# PASS_MAX_DAYS Maximum number of days a password may be used.
# PASS_MIN_DAYS Minimum number of days allowed between password changes.
# PASS_WARN_AGE Number of days warning given before a password expires.
#
PASS_MAX_DAYS 99999
PASS_MIN_DAYS 0
PASS_WARN_AGE 7
#
```

Bu kısmın bilgilerini görseldeki gibi değiştirin:

```
PASS_MAX_DAYS 30
PASS_MIN_DAYS 2
PASS_WARN_AGE 7
```

Bu ayarlamalar sonrası artık şifremiz her 30 günde bir değiştirilecek. Şifremizi değiştirdikten en az 2 gün sonra tekrar değiştirebileceğiz ve şifre süremizin dolmasına 7 gün kaldığında bir uyarı mesajı alacağız. Dosyayı kaydedip çıkalım. Yeni ekleyeceğimiz tüm kullanıcılar bu kurallara tabii olarak eklenecek. Eski kullanıcıların da (root ve kullanıcı hesabınız) bu kurallara tabii olması lazım. Bunun için terminale sırasıyla şu kodları yazıyoruz:

sudo chage --mindays 2 <user>

sudo chage --maxday 30 <user>

sudo chage --warndays 7 <user>

Sonrasında **sudo chage -l <user>** komutu ile kontrol edelim.

```
beergin@beergin43:~$ sudo chage --mindays 2 beergin
beergin@beergin43:~$ sudo chage --maxday 30 beergin
beergin@beergin43:~$ sudo chage --warndays 7 beergin
beergin@beergin43:~$ sudo chage -l beergin
Last password change           : Nov 16, 2024
Password expires                : Dec 16, 2024
Password inactive               : never
Account expires                 : never
Minimum number of days between password change : 2
Maximum number of days between password change : 30
Number of days of warning before password expires : 7
beergin@beergin43:~$ |
```

Aynı işlemi “root” için de yapıyoruz.

```

beergin@beergin43:~$ sudo chage --warndays 7 root
beergin@beergin43:~$ sudo chage --maxday 30 root
beergin@beergin43:~$ sudo chage --mindays 2 root
beergin@beergin43:~$ sudo chage -l root
Last password change           : Nov 16, 2024
Password expires                : Dec 16, 2024
Password inactive              : never
Account expires                : never
Minimum number of days between password change : 2
Maximum number of days between password change : 30
Number of days of warning before password expires : 7
beergin@beergin43:~$ |

```

Şimdi yapacağımız şey şifrenin yapısıyla ilgili olacak. İlk olarak

sudo apt install libpam-pwquality -y ile libpam-pwquality paketini kuruyoruz.

dpkg -l | grep libpam-pwquality ile de doğrulayalım.

```

beergin@beergin43:~$ dpkg -l | grep libpam-pwquality
ii  libpam-pwquality:amd64      1.4.5-1+b1          amd64      PAM module to check password strength
beergin@beergin43:~$ |

```

Paketimiz kurulmuş. Şimdi şifremizi yapılandıracağımız dosyaya girelim :

sudo nano /etc/pam.d/common-password

Bu kodu çalıştırdığımızda aşağıdaki gibi bir dosya açılacak :

```

GNU nano 7.2 /etc/pam.d/common-password *
#
# /etc/pam.d/common-password - password-related modules common to all services
#
# This file is included from other service-specific PAM config files,
# and should contain a list of modules that define the services to be
# used to change user passwords. The default is pam_unix.
#
# Explanation of pam_unix options:
# The "yescrypt" option enables
# hashed passwords using the yescrypt algorithm, introduced in Debian
# 11. Without this option, the default is Unix crypt. Prior releases
# used the option "sha512"; if a shadow password hash will be shared
# between Debian 11 and older releases replace "yescrypt" with "sha512"
# for compatibility. The "obscure" option replaces the old
# 'OBSOLETE_CHECKS_ENAB' option in login.defs. See the pam_unix manpage
# for other options.
#
# As of pam 1.0.1-6, this file is managed by pam-auth-update by default.
# To take advantage of this, it is recommended that you configure any
# local modules either before or after the default block, and use
# pam-auth-update to manage selection of other modules. See
# pam-auth-update(8) for details.
#
# here are the per-package modules (the "Primary" block)
password      requisite pam_pwquality.so retry=3
password      [success=1 default=ignore] pam_unix.so obscure use_authtok try_first_pass yescrypt
# here's the fallback if no module succeeds
password      requisite pam_deny.so
# prime the stack with a positive return value if there isn't one already;
# this avoids us returning an error just because nothing sets a success code
# since the modules above will each just jump around

```

Daire için aldığım kısım ayarlamaları yapacağımız satırlar. İlk olarak

“password requisite pam_pwquality.so retry=3”

satırını ayarlayalım. Bu satırın sonuna sırasıyla şunları ekliyoruz:

ucredit=-1 : En az 1 adet büyük karakter olması için

lcredit=-1 : En az 1 adet küçük karakter olması için

dcredit=-1 : En az 1 adet sayı olması için

maxrepeat=3 : En fazla 3 adet karakterin ardışık olması için

usercheck=1 : Şifre kullanıcı adını içeriyorsa şifrenin geçersiz olması için

difok=7 : Yeni oluşturulacak şifrenin, eski şifrenin içermediği en az 7 karakteri içermesi için

enforce_for_root : Tüm bu değişikliklerin “root” kullanıcıasına da uygulamak için

minlen=10 : Şifrenin en az 10 karakter uzunluğunda olması için

Satırımızın son hali şöyle gözükmeli:

```
password    requisite    pam_pwquality.so retry=3 ucredit=-1 lcredit=-1 dcredit=-1 maxrepeat=3 usercheck=1 difok=7 enforce_for_root minlen=10
password    [success=1 default=ignore]    pam_unix.so obscure use_authok try_first_pass yescrypt
```

İkinci satırın da sonundakileri silip aşağıdaki kodlarla düzeltin:

obscure : Şifrenin daha güvenli olması için

sha512 : Şifreyi sha512 formatında şifrelemek için

Bu düzeltmeyi de yaptıktan sonra çıktımız şu şekilde olmalıdır:

```
password    requisite    pam_pwquality.so retry=3 ucredit=-1 lcredit=-1 dcredit=-1 maxrepeat=3 usercheck=1 difok=7 enforce_for_root minlen=10
password    [success=1 default=ignore]    pam_unix.so obscure sha512
```

Dosyayı kaydedip çıkın ve **sudo reboot** ile de yeniden başlatın.

Artık kullanıcı ekleyebiliriz. **sudo adduser <username>** kodu ile kullanıcı ekleyelim. Yeni kullanıcı ismini “berat” yapıcım.

```
beergin@beergin43:~$ sudo adduser berat
[sudo] password for beergin:
Adding user `berat' ...
Adding new group `berat' (1001) ...
Adding new user `berat' (1001) with group `berat (1001)' ...
Creating home directory `/home/berat' ...
Copying files from `/etc/skel' ...
New password: |
```

Bizden bu kullanıcı için bir şifre istiyor. Şifrenizin yukarda ayarladığınız politikaya uygun olması zorunlu. Şifremizi girdikten sonra şöyle bir çıktı geliyor:

```
passwd: password updated successfully
Changing the user information for berat
Enter the new value, or press ENTER for the default
    Full Name []: |
```

Buradaki isimleri kafanıza göre doldurun ya da boş bırakın önemi yok. Sonrasında onay sorusu gelecek ona da “y” diyip geçin.

```
Enter the new value, or press ENTER for the default
    Full Name []:
    Room Number []:
    Work Phone []:
    Home Phone []:
    Other []:
Is the information correct? [Y/n] y|
```

```
Adding new user 'berat' to supplemental / extra groups 'users' ...
Adding user 'berat' to group 'users' ...
beergin@beergin43:~$ |
```

Görüldüğü üzere “berat” adlı kullanıcı “users” grubuna eklendi. Birazdan grup eklemeyi ve gruba eklemeyi de göstereceğim. Eğer bir kullanıcının olup olmadığını öğrenmek isterseniz şu komutları deneyebilirsiniz:

sudo getent passwd <username>

```
beergin@beergin43:~$ sudo getent passwd berat
berat:x:1001:1001:,,,:/home/berat:/bin/bash
```

id <username>

```
beergin@beergin43:~$ id berat
uid=1001(berat) gid=1001(berat) groups=1001(berat),100(users)
```

Bu kodlar aracılığıyla kullanıcının olup olmadığını ve hangi grupta olduğunu kontrol edebiliyoruz. Yeni kullanıcı oluşturmuşken şifre ayarının bu kullanıcıya uygulanıp uygulanmadığını da kontrol edelim. Bunun için **sudo chage -l <username>** komutunu yazıyoruz.

```
beergin@beergin43:~$ sudo chage -l berat
Last password change           : Nov 19, 2024
Password expires                : Dec 19, 2024
Password inactive               : never
Account expires                 : never
Minimum number of days between password change : 2
Maximum number of days between password change : 30
Number of days of warning before password expires : 7
beergin@beergin43:~$ |
```

Başarılı bir şekilde uygulanmış.

İçinde bulunduğumuz kullanıcı hesabının hangi grupta/gruplarda olduğunu öğrenmek için **groups** kodunu yazıyoruz.

```
beergin@beergin43:~$ groups
beergin cdrom floppy sudo audio dip video plugdev users netdev bluetooth
beergin@beergin43:~$ |
```

Bu komutun sonuna başka bir kullanıcı hesabını yazarsak o hesabın hangi gruplara ait olduğunu görebiliriz.

```
beergin@beergin43:~$ groups berat
berat : berat users
```

Kullanıcıyı ekledik. Silmemiz gerektiğinde de **sudo deluser <user>** ile siliyoruz. Bu komut sadece kullanıcının kaydını siler.

```
beergin@beergin43:~$ sudo deluser berat
Removing crontab ...
Removing user 'berat' ...
Done.
```

Başarılı bir şekilde sildik. İki farklı daha yöntem var bunlar:

sudo deluser --remove-home <user> : Bu komut, kullanıcıyı ana dizini

(Örn **/home/<user>**) ile beraber siler.

sudo deluser --remove-all-files <user> : Bu komut, kullanıcıya ait her şeyi siler.

Sistemdeki tüm kullanıcıları görmek istersek **compugen -u** komutunu kullanabiliriz.

```
beergin@beergin43:~$ compgen -u
root
daemon
bin
sys
sync
games
man
lp
mail
news
uucp
proxy
www-data
backup
list
irc
_apt
nobody
systemd-network
messagebus
avahi-autoipd
beergin
sshd
beergin@beergin43:~$ |
```

Alternatif olarak **sudo cat /etc/passwd** kullanabilirsiniz.

```
beergin@beergin43:~$ sudo cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/run/ircd:/usr/sbin/nologin
_apt:x:42:65534:./nonexistent:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
systemd-network:x:998:998:systemd Network Management:./usr/sbin/nologin
messagebus:x:100:107:./nonexistent:/usr/sbin/nologin
avahi-autoipd:x:101:108:Avahi autoip daemon:./var/lib/avahi-autoipd:/usr/sbin/nologin
beergin:x:1000:1000:./home/beergin:/bin/bash
```

Çıktıyı sadece kullanıcı adlarını göstercek şekilde ayarlamak içinse şu kodu çalıştırabiliriz: **cut -d: -f1 /etc/passwd**

```
beergin@beergin43:~$ cut -d: -f1 /etc/passwd
root
daemon
bin
sys
sync
games
man
lp
mail
news
uucp
proxy
www-data
backup
list
irc
_apt
nobody
systemd-network
messagebus
avahi-autoipd
beergin
sshd
beergin@beergin43:~$ |
```

Hangisini kullanacağınız size kalmış. Şimdi de kullanıcıların şifresini değiştirelim.

Eğer root şifresini değiştirmek istersek **sudo passwd** komutunu girmemiz gerekiyor.

```
beergin@beergin43:~$ sudo passwd
New password:
BAD PASSWORD: The password contains less than 1 uppercase letters
New password:
Retype new password:
passwd: password updated successfully
beergin@beergin43:~$ |
```

Artık “root” şifresi istendiğinde yeni şifreyi girmemiz gerekicek.

Başka bir kullanıcının şifresini değiştirmek için **sudo passwd <user>** kodunu kullanıyoruz.

```
beergin@beergin43:~$ sudo passwd berat
New password:
Retype new password:
passwd: password updated successfully
beergin@beergin43:~$ |
```

İçinde bulunduğumuz kullanıcının hesabını değiştirmek içinse ya az önce ki gibi

sudo passwd <user> komutunu kullanırız ya da sadece **passwd** komutunu kullanırız.

Aradaki fark, “sudo” ön ekiyle yaparsak bizden direkt yeni şifre ister. “sudo” yu kullanmadan yaparsak da önce eski şifreyi sorar sonra yeni şifreyi ister.

```
beergin@beergin43:~$ passwd
Changing password for beergin.
Current password:
New password:
Retype new password:
passwd: password updated successfully
beergin@beergin43:~$ |
```

Kullanıcı hesaplarıyla ilgili işimiz bitti şimdi sıra grup oluşturmada. PDF’te iki tane grup olmasını ve kullanıcımızın bu iki grupta da olmasını istiyordu. “sudo” grubu zaten var diğer grup olan “user42” yi oluşturalım. Bunun için **sudo addgroup <groupname>** komutunu çalıştırıyoruz.

```
beergin@beergin43:~$ sudo addgroup user42
Adding group 'user42' (GID 1002) ...
Done.
beergin@beergin43:~$ |
```

Grubumuz başarıyla oluştu. Kullanıcımızı bu gruba eklemek içinse şu komutu çalıştırıyoruz: **sudo adduser <user> <groupname>**

```
beergin@beergin43:~$ sudo adduser beergin user42
Adding user 'beergin' to group 'user42' ...
Done.
beergin@beergin43:~$ |
```

Kontrol edelim : **getent group <groupname>**

```
beergin@beergin43:~$ getent group sudo
sudo:x:27:beergin
beergin@beergin43:~$ getent group user42
user42:x:1002:beergin
beergin@beergin43:~$ |
```

Kullanıcımız her iki grupta da var. (**groups <user>** kodu ile de bakabilirsiniz)

Bir grubu silmek ya da bir kullanıcıyı gruptan silmek istersek de şu kodları yazmamız yeterli:

sudo delgroup <groupname> : Bir grubu silmek için

sudo deluser <user> <groupname> : Bir kullanıcıyı bir gruptan silmek için. Bu kod kullanıcıyı silmez sadece gruptan kaldırır.

Kullanıcı ve grup işlemlerimiz bitti. Sıra “**monitoring.sh**” dosyasını oluşturmakta. PDF’te her 10 dakikada bir belli bilgileri ekrana yazdırmamız isteniyor. Bunu da **crontab** dosyası ile ayarlayacağız. Öncelikle “netstat” araçlarımızı yükleyelim. Komut satırına

sudo apt install net-tools -y kodunu yazıyoruz. Araçlarımız indikten sonra shell dosyamızı ayarlayalım. İlk olarak **su -** yazarak root hesabımıza geçelim.

```
beergin@beergin43:~$ su -  
Password:  
root@beergin43:~# |
```

“monitoring.sh” dosyamız **/usr/local/bin/** dizininde olmalı bu yüzden **cd /usr/local/bin** komutu ile bu dizine erişip **touch monitoring.sh** komutuyla da shell dosyamı oluşturuyorum.

```
root@beergin43:~# cd /usr/local/bin  
root@beergin43:/usr/local/bin# touch monitoring.sh  
root@beergin43:/usr/local/bin# ls  
monitoring.sh  
root@beergin43:/usr/local/bin# |
```

nano monitoring.sh komutunu kullanarak dosya içine giriyorum. Burada “sudo” ön eki kullanmıyoruz çünkü zaten root kullanıcısındayız.

Dosyamızın içine şu kodları yazıyoruz:

```
#!/bin/bash
```

```

arc=$(uname -a)

pcpu=$(grep "physical id" /proc/cpuinfo | sort | uniq | wc -l)

vcpu=$(grep "^processor" /proc/cpuinfo | wc -l)

fram=$(free -m | awk '$1 == "Mem:" {print $2}')

uram=$(free -m | awk '$1 == "Mem:" {print $3}')

pram=$(free | awk '$1 == "Mem:" {printf("%.2f"), $3/$2*100}')

fdisk=$(df -Bg | grep '^/dev/' | grep -v '/boot$' | awk '{ft += $2} END {print ft}')

udisk=$(df -Bm | grep '^/dev/' | grep -v '/boot$' | awk '{ut += $3} END {print ut}')

pdisk=$(df -Bm | grep '^/dev/' | grep -v '/boot$' | awk '{ut += $3} {ft += $2} END {printf("%d"), ut/ft*100}')

cpul=$(top -bn1 | grep '^%Cpu' | cut -c 9- | xargs | awk '{printf("%.1f%%"), $1 + $3}')

lb=$(who -b | awk '$1 == "system" {print $3 " " $4}')

lvmt=$(lsblk | grep "lvm" | wc -l)

lvmu=$(if [ $lvmt -eq 0 ]; then echo no; else echo yes; fi)

ctcp=$(cat /proc/net/sockstat{,6} | awk '$1 == "TCP:" {print $3}')

ulog=$(users | wc -w)

ip=$(hostname -I)

mac=$(ip link show | awk '$1 == "link/ether" {print $2}')

cmds=$(journalctl _COMM=sudo | grep COMMAND | wc -l)

```

```

wall "  #Architecture: $arc

        #CPU physical: $pcpu

        #vCPU: $vcpu

        #Memory Usage: $uram/${fram}MB ($pram%)

        #Disk Usage: $udisk/${fdisk}Gb ($pdisk%)

        #CPU load: $cpul

        #Last boot: $lb

        #LVM use: $lvmu

        #Connexions TCP: $ctcp ESTABLISHED

        #User log: $ulog

        #Network: IP $ip ($mac)

        #Sudo: $cmds cmd"

```

Dosya aşağıdaki gibi olmalı:

```

GNU nano 7.2 monitoring.sh
#!/bin/bash
arc=$(uname -a)
pcpu=$(grep "physical id" /proc/cpuinfo | sort | uniq | wc -l)
vcpu=$(grep "^processor" /proc/cpuinfo | wc -l)
fram=$(free -m | awk '$1 == "Mem:" {print $2}')
uram=$(free -m | awk '$1 == "Mem:" {print $3}')
pram=$(free | awk '$1 == "Mem:" {printf("%.2f"), $3/$2*100}')
fdisk=$(df -Bg | grep '^/dev/' | grep -v '/boot$' | awk '{ft += $2} END {print ft}')
udisk=$(df -Bg | grep '^/dev/' | grep -v '/boot$' | awk '{ut += $3} END {print ut}')
pdisk=$(df -Bg | grep '^/dev/' | grep -v '/boot$' | awk '{ut += $3} {ft += $2} END {printf("%d"), ut/ft*100}')
cpul=$(top -bn1 | grep '^%Cpu' | cut -c 9- | xargs | awk '{printf("%.1f%"), $1 + $3}')
lb=$(who -b | awk '$1 == "system" {print $3 " " $4}')
lvmt=$(lsblk | grep "lvm" | wc -l)
lvmu=$(if [ $lvmt -eq 0 ]; then echo no; else echo yes; fi)
ctcp=$(cat /proc/net/sockstat{,6} | awk '$1 == "TCP:" {print $3}')
ulog=$(users | wc -w)
ip=$(hostname -I)
mac=$(ip link show | awk '$1 == "link/ether" {print $2}')
cmds=$(journalctl _COMM=sudo | grep COMMAND | wc -l)

wall " #Architecture: $arc
#CPU physical: $pcpu
#vCPU: $vcpu
#Memory Usage: $uram/${fram}MB ($pram%)
#Disk Usage: $udisk/${fdisk}Gb ($pdisk%)
#CPU load: $cpul
#Last boot: $lb
#LVM use: $lvmu
#Connexions TCP: $ctcp ESTABLISHED
#User log: $ulog
#Network: IP $ip ($mac)
#Sudo: $cmds cmd"

```

Kaydedip çıkıyoruz. Shellin çalışıp çalışmadığını test edelim bunun için terminale

sudo bash /usr/local/bin/monitoring.sh komutunu giriyoruz.

```

beergin@beergin43:~$ sudo bash /usr/local/bin/monitoring.sh

Broadcast message from root@beergin43 (pts/1) (Tue Nov 19 13:06:29 2024):

    #Architecture: Linux beergin43 6.1.0-27-amd64 #1 SMP PREEMPT_DYNAMIC Deb
ian 6.1.115-1 (2024-11-01) x86_64 GNU/Linux
    #CPU physical: 1
    #vCPU: 1
    #Memory Usage: 238/1967MB ($.2f%)
    #Disk Usage: 7/14Gb (50%)
    #CPU load: 0.0%
    #Last boot: 2024-11-19 10:02
    #LVM use: yes
    #Connexions TCP: 2 ESTABLISHED
    #User log: 4
    #Network: IP 10.0.2.15 fd00::a00:27ff:feeb:1e7a (08:00:27:cb:1e:7a)
    #Sudo: 113 cmd

Broadcast message from root@beergin43 (pts/1) (Tue Nov 19 13:06:29 2024):

    #Architecture: Linux beergin43 6.1.0-27-amd64 #1 SMP PREEMPT_DYNAMIC Deb
ian 6.1.115-1 (2024-11-01) x86_64 GNU/Linux
    #CPU physical: 1
    #vCPU: 1
    #Memory Usage: 238/1967MB ($.2f%)
    #Disk Usage: 7/14Gb (50%)
    #CPU load: 0.0%
    #Last boot: 2024-11-19 10:02
    #LVM use: yes
    #Connexions TCP: 2 ESTABLISHED
    #User log: 4
    #Network: IP 10.0.2.15 fd00::a00:27ff:feeb:1e7a (08:00:27:cb:1e:7a)
    #Sudo: 113 cmd

Broadcast message from root@beergin43 (pts/1) (Tue Nov 19 13:06:29 2024):

    #Architecture: Linux beergin43 6.1.0-27-amd64 #1 SMP PREEMPT_DYNAMIC Deb

```

Çıktımızda mesajı 3 kez yazdırması normal bunu crontab dosyası ile ayarlama yaptığımızda düzeltereğiz. **sudo crontab -u root -e** kodunu çalıştırarak root yetkisi ile dosyayı açıyoruz.

Aşağıdaki gibi bir dosya açılacak:

```
# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').
#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h dom mon dow   command
```

Bu dosyanın en altına yeni bir satır ekleyip şunu yazıyoruz

***/10 * * * * bash /usr/local/bin/monitoring.sh**

Bu komut sayesinde her 10 dakikada bir sistem bilgilerimiz ekrana yazdırılacak.

Crontab'ın durumunu görmek, durdurmak ve başlatmak için şu komutları yazıyoruz:

sudo systemctl status cron : Durumunu gösterir

sudo systemctl stop cron : Durdurur

sudo systemctl start cron : Başlatır

sudo systemctl restart cron : Yeniden başlatır

Çıktı şöyle olacak:

```
Broadcast message from root@beergin43 (somewhere) (Tue Nov 19 13:30:01 2024):
#Architecture: Linux beergin43 6.1.0-27-amd64 #1 SMP PREEMPT_DYNAMIC Deb
ian 6.1.115-1 (2024-11-01) x86_64 GNU/Linux
#CPU physical: 1
#vCPU: 1
#Memory Usage: 198/1967MB ($.2f%)
#Disk Usage: 7/14Gb (50%)
#CPU load: 0.0%
#Last boot: 2024-11-19 13:26
#LVM use: yes
#Connexions TCP: 2 ESTABLISHED
#User log: 2
#Network: IP 10.0.2.15 fd00::a00:27ff:feeb:1e7a (08:00:27:cb:1e:7a)
#Sudo: 133 cmd
```

BONUS

Zorunlu kısım bitti. Artık sadece bonus kaldı. Bonusta amacımız bir wordpress sitesi açıp sanal makinamızı ona bağlamak. İlk olarak **lighttpd** paketini kuruyoruz. Terminale

sudo apt install lighttpd -y yazıyoruz. Kurulumdan sonra kontrol edelim

dpkg -l | grep lighttpd

```
beergin@beergin43:~$ dpkg -l | grep lighttpd
ii lighttpd                    1.4.69-1          amd64        fast webserver with minimal memory footprint
ii lighttpd-mod-deflate       1.4.69-1          amd64        HTTP response compression module for lighttpd
ii lighttpd-mod-openssl       1.4.69-1          amd64        TLS support using OpenSSL module for lighttpd
```

Başarıyla inmiş. Terminale **sudo systemctl status lighttpd** yazıp çalışıp çalışmadığına bakalım.

```
beergin@beergin43:~$ sudo systemctl status lighttpd
● lighttpd.service - Lighttpd Daemon
   Loaded: loaded (/lib/systemd/system/lighttpd.service; enabled; preset: enabled)
   Active: active (running) since Tue 2024-11-19 13:37:56 CST; 5min ago
     Process: 995 ExecStartPre=/usr/sbin/lighttpd -tt -f /etc/lighttpd/lighttpd.conf (code=exited, status=0/SUCCESS)
    Main PID: 1000 (lighttpd)
      Tasks: 1 (limit: 2302)
     Memory: 840.0K
        CPU: 200ms
      CGroup: /system.slice/lighttpd.service
              └─1000 /usr/sbin/lighttpd -D -f /etc/lighttpd/lighttpd.conf

Nov 19 13:37:56 beergin43 systemd[1]: Starting lighttpd.service - Lighttpd Daemon...
Nov 19 13:37:56 beergin43 systemd[1]: Started lighttpd.service - Lighttpd Daemon.
beergin@beergin43:~$
```

Çalışıyor. Sıra porta izin vermek. Wordpress 80 portunu kullanıyor biz de o yüzden kural olarak 80 portunu ekliyoruz: **sudo ufw allow 80**

```
beergin@beergin43:~$ sudo ufw allow 80
Rule added
Rule added (v6)
beergin@beergin43:~$ sudo systemctl status ufw
● ufw.service - Uncomplicated firewall
   Loaded: loaded (/lib/systemd/system/ufw.service; enabled; preset: enabled)
   Active: active (exited) since Tue 2024-11-19 13:26:08 CST; 19min ago
     Docs: man:ufw(8)
    Main PID: 478 (code=exited, status=0/SUCCESS)
      CPU: 76ms

Nov 19 13:26:07 beergin43 systemd[1]: Starting ufw.service - Uncomplicated firewall...
Nov 19 13:26:08 beergin43 systemd[1]: Finished ufw.service - Uncomplicated firewall.
beergin@beergin43:~$
```

Kuralımız eklendi güvenlik duvarımız da güzelce çalışıyor. Şimdi de Oracle Virtual Box'u açıp ordan da 80 portuna izin verelim. **Ayarlar > Ağ > B.Noktası Yönlendirme** adımlarını izleyerek aşağıdaki yere geliyoruz:

Ad	Protokol	Anamakine IP	Anamakine B.Noktası	Misafir IP	Misafir B
Rule 1	TCP		10022		4242
Rule 2	TCP		80		80

Burada sağ üstteki yeşil butona basıp yeni kural ekliyoruz ve **Anamakine B.Noktasıyla Misafir B.Noktası** resimdeki gibi **80** olarak ayarlıyoruz. “Tamam” diyip çıkın ve “reboot” atın.

Şimdi ise wordpress için kullanacağımız veri tabanı olan MariaDB’yi kuralım.

sudo apt install mariadb-server -y kodunu girip çalıştırıyoruz. Kurulumdan sonra kontrol edelim :

```
beergin@beergin43:~$ dpkg -l | grep mariadb-server
ii mariadb-server          1:10.11.6-0+deb12u1      amd64      MariaDB database server binaries
ii mariadb-server-core     1:10.11.6-0+deb12u1      amd64      MariaDB database core server files
beergin@beergin43:~$ |
```

Doğru bir şekilde inmiş. Durumuna da bakalım: **sudo systemctl status mysql.server**

```
beergin@beergin43:~$ sudo systemctl status mysql.service
● mariadb.service - MariaDB 10.11.6 database server
   Loaded: loaded (/lib/systemd/system/mariadb.service; enabled; preset: enabled)
   Active: active (running) since Tue 2024-11-19 13:52:31 CST; 2min 21s ago
     Docs: man:mariadb(8)
           https://mariadb.com/kb/en/library/systemd/
   Main PID: 1694 (mariabdd)
   Status: "Taking your SQL requests now..."
     Tasks: 9 (limit: 2302)
    Memory: 200.0M
       CPU: 557ms
   CGroup: /system.slice/mariadb.service
           └─1694 /usr/sbin/mariabdd

Nov 19 13:52:31 beergin43 mariabdd[1694]: 2024-11-19 13:52:31 0 [Note] InnoDB: Loading buffer pool(s) from /var/lib/mysql/ib_buffer_p
Nov 19 13:52:31 beergin43 mariabdd[1694]: 2024-11-19 13:52:31 0 [Warning] You need to use --log-bin to make --expire-logs-days or --b
Nov 19 13:52:31 beergin43 mariabdd[1694]: 2024-11-19 13:52:31 0 [Note] Server socket created on IP: '127.0.0.1'.
Nov 19 13:52:31 beergin43 mariabdd[1694]: 2024-11-19 13:52:31 0 [Note] InnoDB: Buffer pool(s) load completed at 241119 13:52:31
Nov 19 13:52:31 beergin43 mariabdd[1694]: 2024-11-19 13:52:31 0 [Note] /usr/sbin/mariabdd: ready for connections.
Nov 19 13:52:31 beergin43 mariabdd[1694]: Version: '10.11.6-MariaDB-0+deb12u1' socket: '/run/mysqld/mysqld.sock' port: 3306 Debian
Nov 19 13:52:31 beergin43 systemd[1]: Started mariadb.service - MariaDB 10.11.6 database server.
Nov 19 13:52:31 beergin43 /etc/mysql/debian-start[1709]: Upgrading MySQL tables if necessary.
Nov 19 13:52:31 beergin43 /etc/mysql/debian-start[1720]: Checking for insecure root accounts.
Nov 19 13:52:32 beergin43 /etc/mysql/debian-start[1724]: Triggering myisam-recover for all MyISAM tables and aria-recover for all Ari
beergin@beergin43:~$ |
```

Veri tabanımızın güvenliği için **sudo mysql_secure_installation** kodunu çalıştıralım.

İlk olarak root şifresini isteyecek şifreyi girip enter’a basıyoruz. Sonrasında gelen soruya **N** diyoruz.

```
Enter current password for root (enter for none):
OK, successfully used password, moving on...

Setting the root password or using the unix_socket ensures that nobody
can log into the MariaDB root user without the proper authorisation.

You already have your root account protected, so you can safely answer 'n'.

Switch to unix_socket authentication [Y/n] n|
```

Root şifremizi değiştirmek isteyip istemediğimizi soruyor ona da **N** diyoruz.

```
You already have your root account protected, so you can safely answer 'n'.

Change the root password? [Y/n] n|
```

Sonrasındaki tüm sorulara **Y** diyoruz.

```
By default, a MariaDB installation has an anonymous user, allowing anyone
to log into MariaDB without having to have a user account created for
them. This is intended only for testing, and to make the installation
go a bit smoother. You should remove them before moving into a
production environment.
```

```
Remove anonymous users? [Y/n] y|
```

```
Normally, root should only be allowed to connect from 'localhost'. This
ensures that someone cannot guess at the root password from the network.
```

```
Disallow root login remotely? [Y/n] y|
```

```
By default, MariaDB comes with a database named 'test' that anyone can
access. This is also intended only for testing, and should be removed
before moving into a production environment.
```

```
Remove test database and access to it? [Y/n] y
```

```
Reloading the privilege tables will ensure that all changes made so far
will take effect immediately.
```

```
Reload privilege tables now? [Y/n] y|
```

Aşağıdaki gibi bir çıktı aldığınızda işlem başarıyla tamamlanmış demektir:

```
All done! If you've completed all of the above steps, your MariaDB
installation should now be secure.
```

```
Thanks for using MariaDB!
```

```
beergin@beergin43:~$ |
```

Servisimizi aşağıdaki kodlar ile kontrol edebiliriz:

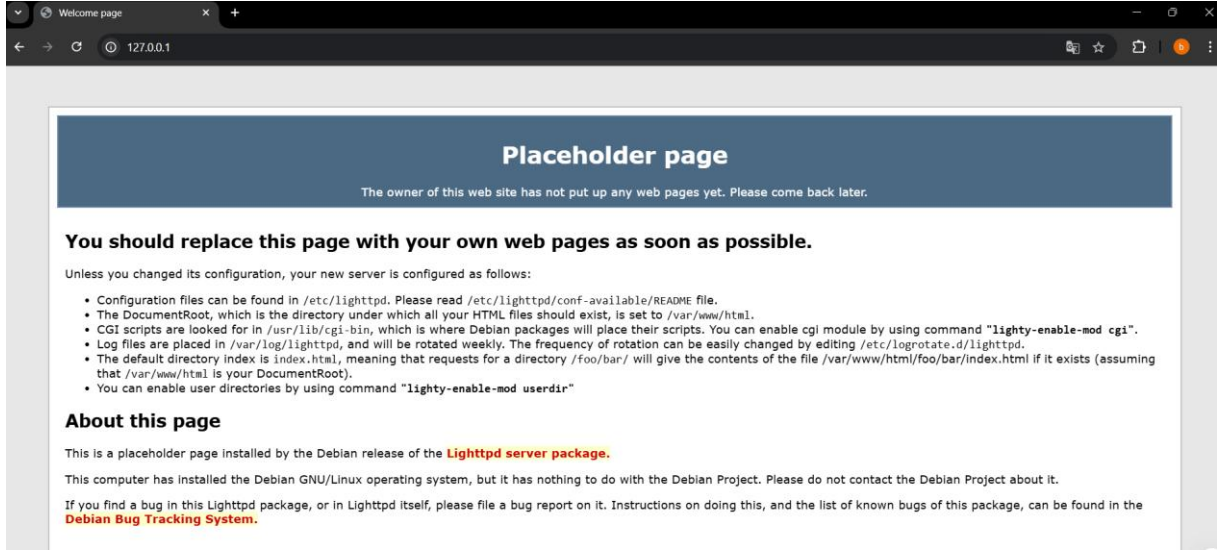
sudo systemctl status mysql.service : Durumunu gösterir

sudo systemctl stop mysql.service: Durdurur

sudo systemctl start mysql.service: Başlatır

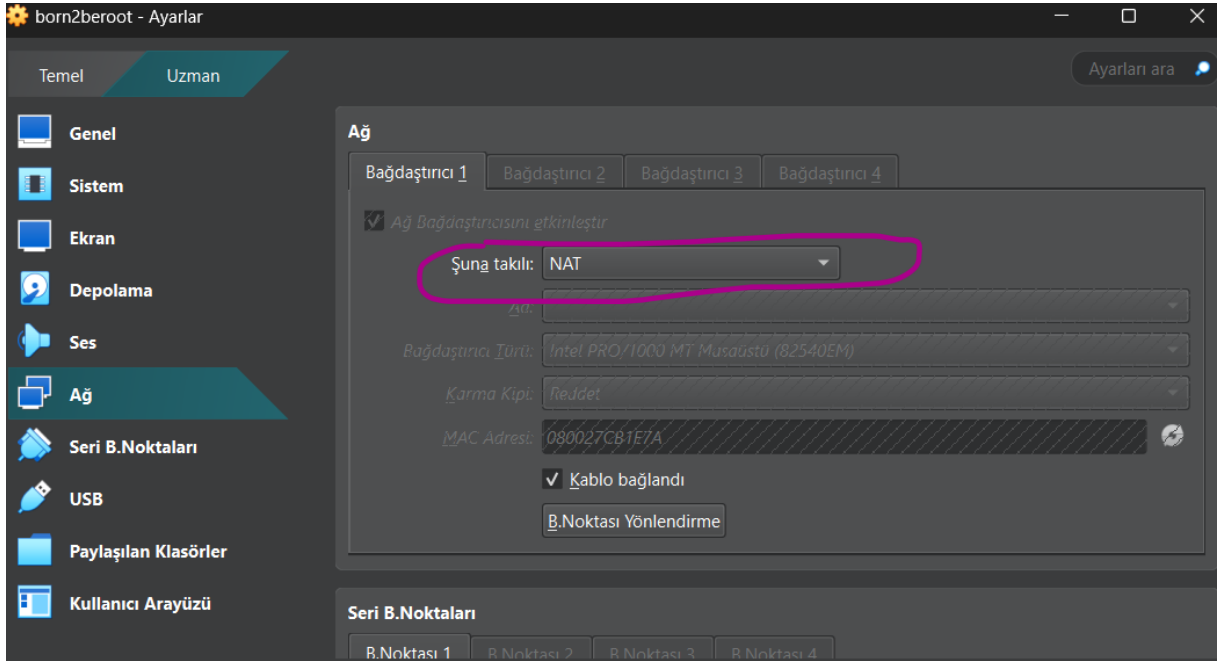
sudo systemctl restart mysql.service: Yeniden başlatır

Web sitemiz açıldı mı bakalım. Tarayıcınızdan **127.0.0.1** adresine gidin.



Site bu şekilde görünüyorsa bir sorun yoktur. Eğer site bu şekilde görünmüyorsa ve “Site Bulunamadı” tarzı bir mesaj varsa aşağıdaki adımları takip edin:

Sanal makinayı kapatıp Oracle Virtual Box’u açın ve ayarlardan “Ağ” sekmesine gidin.



İşaretlediğim yere basın ve orayı “Köprü Bağdaştırıcısı” (İngilizce kullanıyorsanız “Bridged Adapter”) olarak değiştirin. Sonrasında sanal makinanın terminaline şu kodu yazın : **hostname -I** bu komut size sanal makinanızın fiziksel IP’sini verir.

```
beergin@beergin43:~$ hostname -I
192.168.1.100
```


Bu IP'yi tarayacınıza yazdığınızda siteniz açılacak. Ek olarak bağlantı yöntemini değiştirdiğiniz için artık kendi terminalinizden sanal makineye bağlanırken yazdığınız komutu şu şekilde değiştirmeniz gerekecek **ssh -p 4242 <user>@<fiziksel_IP>**

```
PS C:\Users\ggolg> ssh -p 4242 beergin@192
beergin@192.1 's password: |
```

Bu şu yüzden oluyor: NAT dediğimiz yöntem localhost (127.0.0.1) üzerinden bağlanmaya izin veriyor fakat Bridged Adapter bağlantı için sanal makinenizin fiziksel IP'sini istiyor. Bu sorunu hallettiğimize göre devam edelim.

Sıra PHP-FPM ve LIGHTTPD'yi beraber çalışacak şekilde ayarlamakta.

İlk olarak PHP'nin bileşenlerini indirmemiz lazım.

```
sudo apt install php8.2-fpm php8.2-common php8.2-mysql php8.2-xml php8.2-xmlrpc
php8.2-curl php8.2-gd php8.2-imagick php8.2-cli php8.2-mbstring php8.2-opcache
php8.2-readline php8.2-soap php8.2-zip -y
```

Bu indirme komutunu direkt terminalde çalıştırıp gereken paketleri indirebilirsiniz.

Bu projeyi yaptığımda geçerli sürüm 8.2'ydi. İlerde sürümler değişeceğinden güncel olan sürümün paketlerini indirmeye dikkat edin.

Terminale **sudo nano /etc/php/8.2/fpm/pool.d/www.conf** yazıp entera basıyoruz. Karşımıza şöyle bir dosya açılacak:

```
Start a new pool named 'www'.
the variable $pool can be used in any directive and will be replaced by the
pool name ('www' here)
[www]

Per pool prefix
It only applies on the following directives:
- 'access.log'
- 'slowlog'
- 'listen' (unixsocket)
- 'chroot'
- 'chdir'
- 'php_values'
- 'php_admin_values'
When not set, the global prefix (or /usr) applies instead.
Note: This directive can also be relative to the global prefix.
Default Value: none
prefix = /path/to/pools/$pool

Unix user/group of the child processes. This can be used only if the master
process running user is root. It is set after the child process is created.
The user and group can be specified either by their name or by their numeric
IDs.
Note: If the user is root, the executable needs to be started with
--allow-to-run-as-root option to work.
Default Values: The user is set to master process running user by default.
If the group is not set, the user's group is used.
user = www-data
group = www-data

The address on which to accept FastCGI requests.
Valid syntaxes are:
'ip.add.re.ss:port' - to listen on a TCP socket to a specific IPv4 address on
a specific port;
```

Bu dosyada aşağıdaki satırı bulun:

```
listen = /run/php/php8.2-fpm.sock
```

Bu satırı şununla değiştirin: listen=127.0.0.1:9000

```
; Note: This value is mandatory.  
listen = 127.0.0.1:9000|
```

Burayı kaydedip çıkabiliriz. Aynı işlemi lighttpd için de yapalım.

Terminale **sudo nano /etc/lighttpd/conf-available/15-fastcgi-php.conf** yazıyoruz.

Açılan dosyada aşağıdaki satırları bulun:

```
"bin-path" => "/usr/bin/php-cgi",  
"socket" => "/run/lighttpd/php.socket",
```

Bu satırları:

"host" => "127.0.0.1",

"port" => "9000",

Şeklinde değiştirin.

```
"host" => "127.0.0.1",  
"port" => "9000",
```

Kaydedip çıkabiliriz.

Şimdi de etkinleştirmemiz gereken 2 modülümüz var onları etkinleştirelim. Modülleri etkinleştirmek için:

sudo lighty-enable-mod fastcgi

sudo lighty-enable-mod fastcgi-php

```
beergin@beergin43:~$ sudo lighty-enable-mod fastcgi  
Enabling fastcgi: ok  
Run "service lighttpd force-reload" to enable changes  
beergin@beergin43:~$ sudo lighty-enable-mod fastcgi-php  
Enabling fastcgi-php: ok  
Run "service lighttpd force-reload" to enable changes  
beergin@beergin43:~$ |
```

Değişiklikleri kaydetmek için lighttpd ve php-fpm hizmetini tekrar başlatın:

sudo systemctl restart lighttpd

sudo systemctl restart php8.2-fpm

```

beergin@beergin43:~$ sudo systemctl restart lighttpd
beergin@beergin43:~$ sudo systemctl restart php8.2-fpm
beergin@beergin43:~$ sudo systemctl status lighttpd
● lighttpd.service - Lighttpd Daemon
   Loaded: loaded (/lib/systemd/system/lighttpd.service; enabled; preset: enabled)
   Active: active (running) since Tue 2024-11-19 15:18:07 CST; 13s ago
     Process: 1123 ExecStartPre=/usr/sbin/lighttpd -tt -f /etc/lighttpd/lighttpd.conf (code=exited, status=0/SUCCESS)
    Main PID: 1130 (lighttpd)
       Tasks: 1 (limit: 2302)
      Memory: 720.0K
         CPU: 166ms
    CGroup: /system.slice/lighttpd.service
            └─1130 /usr/sbin/lighttpd -D -f /etc/lighttpd/lighttpd.conf

Nov 19 15:18:06 beergin43 systemd[1]: lighttpd.service: Deactivated successfully.
Nov 19 15:18:06 beergin43 systemd[1]: Stopped lighttpd.service - Lighttpd Daemon.
Nov 19 15:18:06 beergin43 systemd[1]: Starting lighttpd.service - Lighttpd Daemon...
Nov 19 15:18:07 beergin43 systemd[1]: Started lighttpd.service - Lighttpd Daemon.
beergin@beergin43:~$ sudo systemctl status php8.2-fpm
● php8.2-fpm.service - The PHP 8.2 FastCGI Process Manager
   Loaded: loaded (/lib/systemd/system/php8.2-fpm.service; enabled; preset: enabled)
   Active: active (running) since Tue 2024-11-19 15:18:13 CST; 19s ago
     Docs: man:php-fpm8.2(8)
     Process: 1144 ExecStartPost=/usr/lib/php/php-fpm-socket-helper install /run/php/php-fpm.sock /etc/php/8.2/fpm/po
    Main PID: 1141 (php-fpm8.2)
   Status: "Processes active: 0, idle: 2, Requests: 0, slow: 0, Traffic: 0.00req/sec"
       Tasks: 3 (limit: 2302)
      Memory: 13.4M
         CPU: 49ms
    CGroup: /system.slice/php8.2-fpm.service
            └─1141 "php-fpm: master process (/etc/php/8.2/fpm/php-fpm.conf)"
              └─1142 "php-fpm: pool www"
                └─1143 "php-fpm: pool www"

Nov 19 15:18:13 beergin43 systemd[1]: php8.2-fpm.service: Deactivated successfully.
Nov 19 15:18:13 beergin43 systemd[1]: Stopped php8.2-fpm.service - The PHP 8.2 FastCGI Process Manager.
Nov 19 15:18:13 beergin43 systemd[1]: Starting php8.2-fpm.service - The PHP 8.2 FastCGI Process Manager...

```

Şimdi ise bir **phpinfo.php** dosyası oluşturacağız. Terminale **cd /var/www/html** yazarak dizine gidin ve **sudo touch phpinfo.php** yazarak dosyamızı oluşturun.

sudo nano phpinfo.php ile de içine aşağıdaki kodları yazın:

```

<?php
phpinfo();
phpinfo(INFO_MODULES);
?>

```

Dosyayı kaydedip çıkabiliriz.

Şimdi veritabanımızı halledelim. Terminale **sudo mysql** yazıp entera bastığımızda karşımıza şöyle bir şey gelecek:

```

beergin@beergin43:/var/www/html$ sudo mysql
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 32
Server version: 10.11.6-MariaDB-0+deb12u1 Debian 12

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> |

```

Bu kısma sql kodlarımızı yazacağız. İlk olarak veritabanımızı oluşturalım. Komut satırına

CREATE DATABASE wordpress; yazıp entera basıyoruz.

```
MariaDB [(none)]> CREATE DATABASE wordpress;
Query OK, 1 row affected (0.000 sec)

MariaDB [(none)]> |
```

Bu çıktıyı aldıysak oluşmuştur. Şimdi ise kullanıcı oluşturalım.

GRANT ALL PRIVILEGES on wordpress.* TO 'admin'@'localhost' IDENTIFIED BY 'Berat42';

```
MariaDB [(none)]> GRANT ALL PRIVILEGES on wordpress.* TO 'admin'@'localhost' IDENTIFIED BY 'Berat42';
Query OK, 0 rows affected (0.004 sec)
```

Artık “admin” kullanıcı adına ve “Berat42” şifresine sahip bir kullanıcı var. Sonrasında yetkileri kullanıcımıza uygulamak için **FLUSH PRIVILEGES;** kodunu giriyoruz. Ardından **EXIT;** ile bu kısımdan çıkabiliriz.

```
MariaDB [(none)]> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.002 sec)

MariaDB [(none)]> EXIT;
Bye
beergin@beergin43: /var/www/html$ |
```

Şimdi sıra wordpress’i indirmekte. Wordpress’in kaynak dosyalarının kopyasını indirmek için öncelikle **/var/www/html** dizinine geçmemiz lazım.

```
beergin@beergin43: /var/www/html$ cd /var/www/html/
beergin@beergin43: /var/www/html$ ls
index.lighttpd.html  phpinfo.php
beergin@beergin43: /var/www/html$
```

Wget aracını kullanarak yapacağız sonraki işlemi. Bu yüzden terminale

sudo apt install wget -y yazıp aracımızı indiriyoruz. Sonrasında terminale

sudo wget <http://wordpress.org/latest.tar.gz> yazıp arşiv dosyasını indiriyoruz.

```
beergin@beergin43: /var/www/html$ sudo wget http://wordpress.org/latest.tar.gz
--2024-11-19 15:45:54-- http://wordpress.org/latest.tar.gz
Resolving wordpress.org (wordpress.org)... 198.143.164.252
Connecting to wordpress.org (wordpress.org)|198.143.164.252|:80... connected.
HTTP request sent, awaiting response... 301 Moved Permanently
Location: https://wordpress.org/latest.tar.gz [following]
--2024-11-19 15:45:55-- https://wordpress.org/latest.tar.gz
Connecting to wordpress.org (wordpress.org)|198.143.164.252|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 26927286 (26M) [application/octet-stream]
Saving to: 'latest.tar.gz'

latest.tar.gz          100%[=====] 25.68M  3.20MB/s   in 8.8s
2024-11-19 15:46:04 (2.93 MB/s) - 'latest.tar.gz' saved [26927286/26927286]
```

Arşiv dosyasının içindekileri çıkarmak için **sudo tar -xvzf latest.tar.gz** komutunu çalıştıralım. Dosyaları “**wordpress**” adında bir klasörün içinde olacak şekilde çıkardı fakat bu dosyaların direkt olarak dizinimizde olması lazım. Bu yüzden komut satırına

sudo mv wordpress/* ./ yazarak taşıma yapıyoruz.

```
beergin@beergin43:/var/www/html$ sudo mv wordpress/* ./
beergin@beergin43:/var/www/html$ ls
index.lighttpd.html  phpinfophp  wp-admin  wp-content  wp-load.php  wp-signup.php
index.php            readme.html  wp-blog-header.php  wp-cron.php  wp-login.php  wp-trackback.php
latest.tar.gz        wordpress    wp-comments-post.php  wp-includes  wp-mail.php   xmlrpc.php
license.txt          wp-activate.php  wp-config-sample.php  wp-links-opml.php  wp-settings.php
beergin@beergin43:/var/www/html$ |
```

Görüldüğü üzere dosyalarımız geldi. Artık “wordpress” klasörünü silebiliriz. Bunu da **sudo rm -rf wordpress** komutu ile yapıyoruz. Eğer varsa varsayılan .html dosyasını da silmek için **sudo rm index.html** kodunu çalıştırın. Sırada wordpress’in ayarını yapmakta.

Tarayacınızdan **127.0.0.1** adresine gidin.

Aşağıdaki gibi bir sekme gelecek:

Welcome to WordPress. Before getting started, you will need to know the following items.

1. Database name
2. Database username
3. Database password
4. Database host
5. Table prefix (if you want to run more than one WordPress in a single database)

This information is being used to create a wp-config.php file. **If for any reason this automatic file creation does not work, do not worry. All this does is fill in the database information to a configuration file. You may also simply open wp-config-sample.php in a text editor, fill in your information, and save it as wp-config.php.** Need more help? [Read the support article on wp-config.php.](#)

In all likelihood, these items were supplied to you by your web host. If you do not have this information, then you will need to contact them before you can continue. If you are ready...

Let's go!

Buna “Let’s go!” diyoruz. Diğer kısımda bizden bilgiler girmemizi istiyor. Aşağıdaki gibi doldurabilirsiniz:

Below you should enter your database connection details. If you are not sure about these, contact your host.

Database Name

The name of the database you want to use with WordPress.

Username

Your database username.

Password

[Hide](#)

Your database password.

Database Host

You should be able to get this info from your web host, if localhost does not work.

Table Prefix

If you want to run multiple WordPress installations in a single database, change this.

[Submit](#)

Bilgileri doldurduktan sonra devam edelim. “Database Host” kısmının “localhost” olması zorunlu. Diğer şekilde wordpress hata veriyor. Yani fiziksel IP’niz ile de erişiyor olsanız “localhost” yazın.

Aşağıdaki gibi bir hata almanız muhtemel:

Unable to write to wp-config.php file.

You can create the wp-config.php file manually and paste the following text into it.

Configuration rules for wp-config.php:

```
<?php
/**
 * The base configuration for WordPress
 *
 * The wp-config.php creation script uses this file during the installation.
 * You don't have to use the website, you can copy this file to "wp-config.php"
 * and fill in the values.
 *
 * This file contains the following configurations:
 *
 * * Database settings
 * * Secret keys
 * * Database table prefix
 * * ABSPATH
 */
```

After you've done that, click “Run the installation”.

[Run the installation](#)

Bu hata durumunda terminale girip **/var/www/html** dizininde **wp-config.php** dosyası oluřturun ve iine resimdeki kodları yapıřtırıp “Run the installation” butonuna basın.

Sonrasında byle bir ekran karřılayacak bizi:

Welcome

Welcome to the famous five-minute WordPress installation process! Just fill in the information below and you'll be on your way to using the most extendable and powerful personal publishing platform in the world.

Information needed

Please provide the following information. Do not worry, you can always change these settings later.

Site Title

Username

Usernames can have only alphanumeric characters, spaces, underscores, hyphens, periods, and the @ symbol.

Password

B1IUH657)!p#!Q1!FL

Hide

Strong

Important: You will need this password to log in. Please store it in a secure location.

Your Email

Double-check your email address before continuing.

Search engine visibility

☐ Discourage search engines from indexing this site

It is up to search engines to honor this request.

Install WordPress

Burayı da řyle doldurabiliriz:

Site Title

born2beroot proje

Username

admin

Usernames can have only alphanumeric characters, spaces, underscores, hyphens, periods, and the @ symbol.

Password

••••••

Show

Weak

Important: You will need this password to log in. Please store it in a secure location.

Confirm Password

☒ Confirm use of weak password

Your Email

beergin@student.42istanbul.com.tr

Double-check your email address before continuing.

Search engine visibility

☐ Discourage search engines from indexing this site

It is up to search engines to honor this request.

Install WordPress

“Install Wordpress” ile devam ediyoruz. Gelen sekmeye de “Log In” diyoruz.

Success!

WordPress has been installed. Thank you, and enjoy!

Username admin

Password *Your chosen password.*

[Log In](#)

Giriş panelini seçtiğiniz kullanıcı adı ve şifreyle doldurun:

Username or Email Address

admin

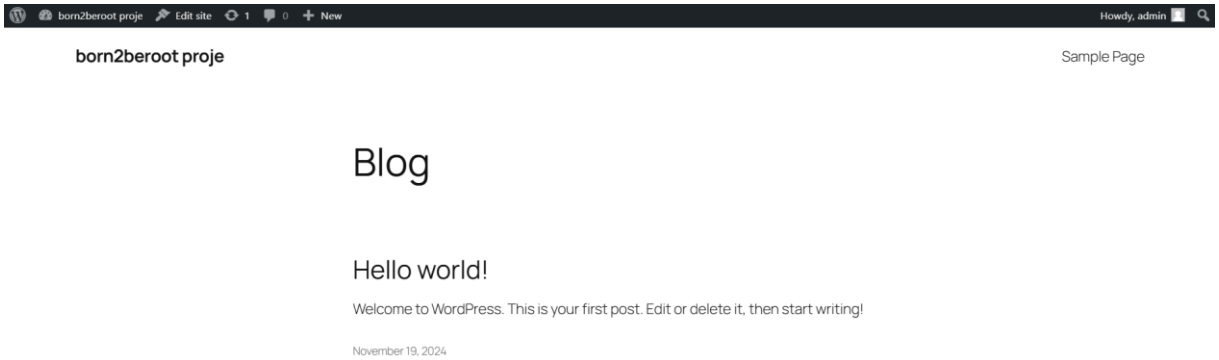
Password

•••••

☐ Remember Me

Log In

Ve sitemiz açıldı!



Arka planda resim olmaması sizi endişelendirmesin önemli olan bağlantıyı sağlayabilmek. İsterseniz kendiniz özelleştirebilirsiniz.

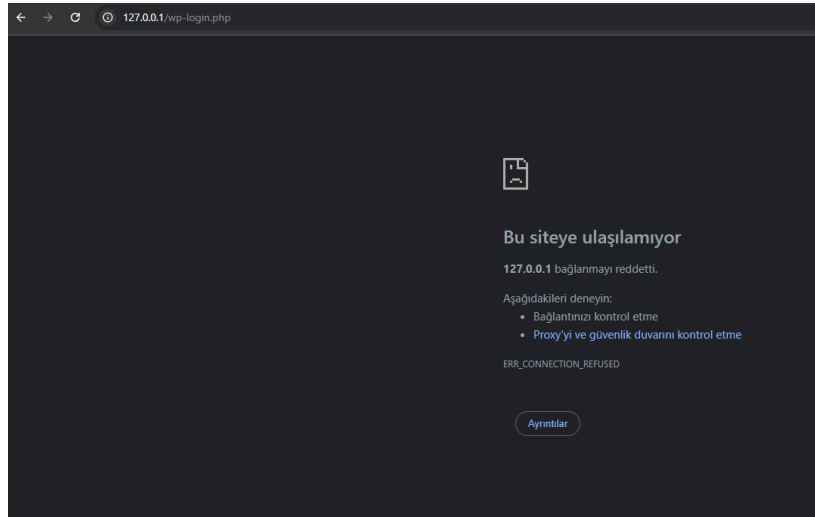
ÖNEMLİ!!

Eğer “Bridged Adapter” yani “Köprü Bağdaştırıcısı” yöntemini kullanıp fiziksel IP’niz ile wordpresse bağlanıyorsanız makinayı çalıştırdığınızda CSS ve PHP ile ilgili bir sorun yaşamanız muhtemel. Sorun örnekleri için aşağı bakınız.

CSS:



PHP:



Bunu çözmek için şu adımları izleyin:

Terminalde **cd /var/www/html** komutunu çalıştırıp dizine gidin ve

sudo nano wp-config.php kodu ile dosyanın içine girin. Dosyada şu kısma gelin:

```
/** The database collate type. Don't change this if in doubt. */
define( 'DB_COLLATE', '' );
```

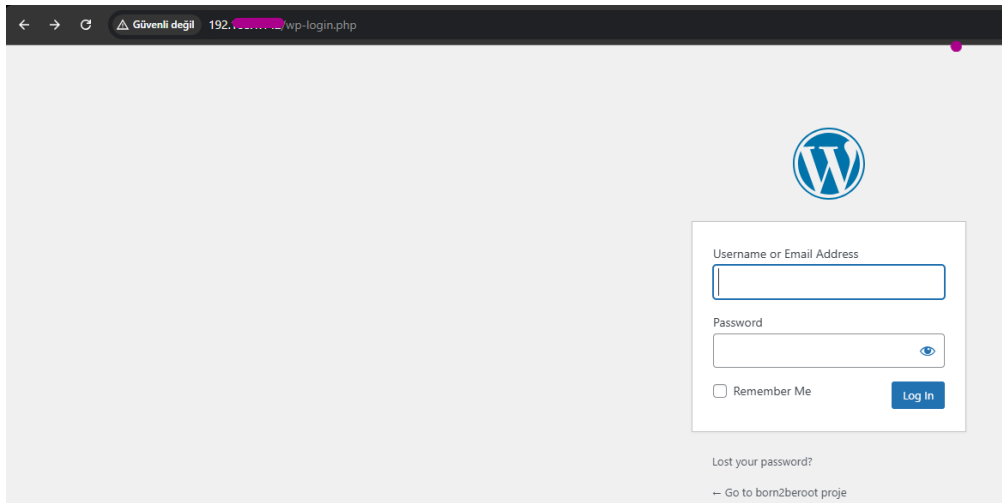
Oradaki boş kısma řu kodları yazıyoruz:

```
define('WP_HOME', 'http://<fiziksel_IP>');  
define('WP_SITEURL', 'http://<fiziksel_IP>');
```

Kullanım:

```
/** The database collate type. Don't change this  
define( 'DB_COLLATE', '' );  
define('WP_HOME', 'http://192.168.1.100');  
define('WP_SITEURL', 'http://192.168.1.100');  
/**#@+
```

Bu işlemlerden sonra site düzelecektir:



Klon oluşturduğunuzda da klonun fiziksel IP'si orijinal makinanın fiziksel IP'sinden farklı olur. Bu işlemlerin aynısını klon için de yaparak sorunsuz bir şekilde projenizi verebilirsiniz.

Born2beroot projesi bu kadardı. PDF'in devamında bazı yerleri detaylandıracağım ve evo ile ilgili bazı bilgiler vereceğim. İsteyen oraya da bakabilir.

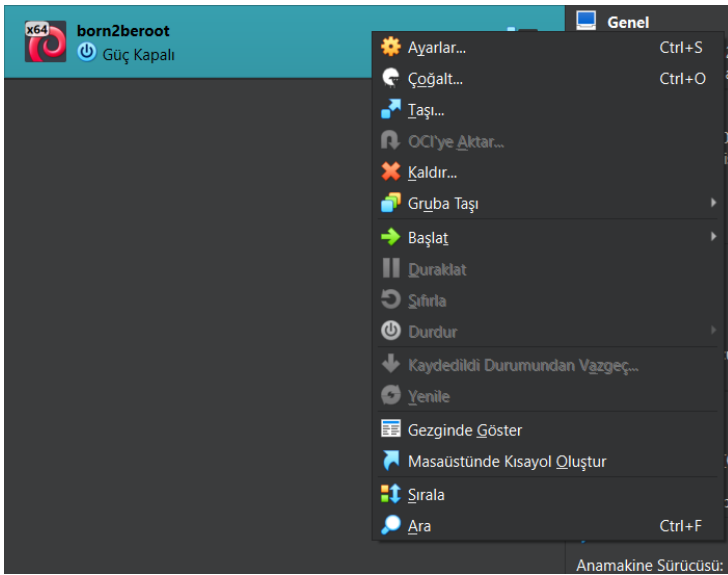
EVOYA HAZIRLIK

Proje tamamen bittikten sonra sistemdeki “root” ve sizin ana kullanıcı hesabınız yani intra kullanıcı adınıza sahip olan hesabınız (Örn: beergin) dışındaki tüm oluşturduğunuz hesapları silin. Aynı şekilde user42 ve sudo grupları dışındaki diğer tüm oluşturduğunuz grupları da silin. Eğer user42 grubunu oluşturmadıysanız oluşturun. PDF esnasında hostname’i **beergin43** olarak değiştirmiştik büyük ihtimal siz de değiştirdiniz. Hostname’i tekrar eskisine çevirin yani <username>42 ‘ye.

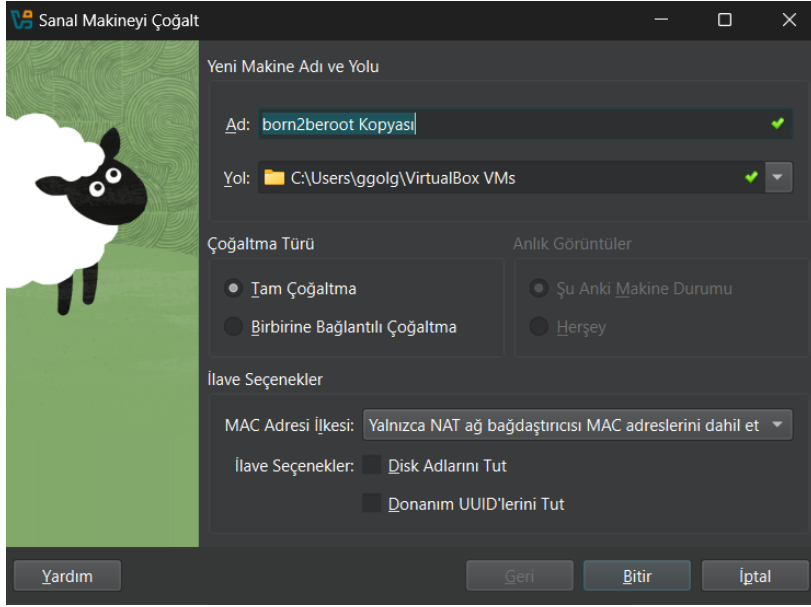
Benim için > beergin42

Eklediğiniz başka hostname varsa onu da kaldırın sadece <username>42 hostu kalacak. Ana kullanıcı hesabınızın hem sudo hem de user42 gruplarında olması lazım. Yoksa ikisine de ekleyin. Burayı hallettikten sonra sanal makinayı kapatın ve sanal makinanın bir klonunu oluşturun. Bunun için aşağıdaki adımları takip edin:

Sol menüden projenize sağ tıklayın:

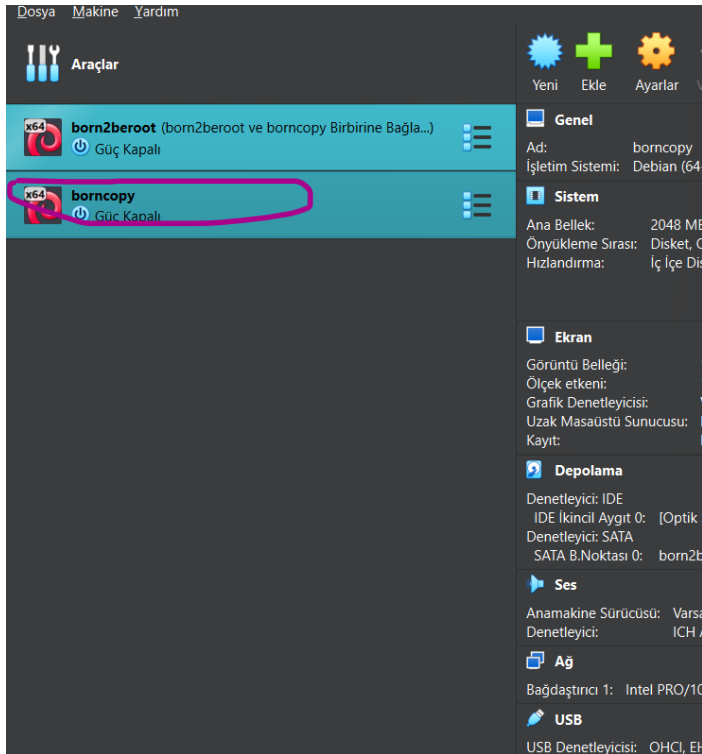


Ardından “Çoğalt” seçeneğini seçin:



Burayı doldurun. Ben windowsta olduğumdan yapamam fakat siz bunu okulda yaparken bu kopyayı “goinfre” klasörüne koymanız lazım bu yüzden “Yol:” kısmından goinfre dizinini seçin.

Sonrasında Çoğaltma Türü’nü “Birbirine Bağlantılı Çoğaltma” olarak seçip Bitir’e basın. Sol tarafa aşağıdaki gibi bir öge eklencek:



Bundan sonraki adımda terminalinizi açın. Ve sanal makinayı kurduğunuz klasöre (sizde goinfre) gidin.

```
Directory: C:\Users\ggolg\VirtualBox VMs\born2beroot

de                LastWriteTime                Length Name
--                -
----            -
19.11.2024        23:47                        Logs
20.11.2024        01:32                        Snapshots
20.11.2024        01:32                      6483 born2beroot.vbox
20.11.2024        01:32                      7002 born2beroot.vbox-prev
20.11.2024        01:29          15820914688 born2beroot.vdi

C:\Users\ggolg\VirtualBox VMs\born2beroot>
```

Yukardaki resimde gözüken sonu .vdi olan dosya bizim hedefimiz. Bu dizindeyken komut satırına şunu yazın : shasum <projeadı>.vdi

Bir süre bekleyeceksiniz (yaklaşık 30 saniye) sonrasında şu tarz bir çıktı verecek size:

beb691e1cfcd4633f66df40062f517feb39859f7 born2beroot.vdi

Bende çıkan **born2beroot.vdi**, sizde verdiğiniz isim neyse o olarak çıkacak.

Bu çıktıyı komple alıp **signature.txt** adlı bir txt dosyasına koyup pushlayacaksınız. Makinayı klonlama sebebimiz de şu : siz her sanal makinayı tekrar çalıştırdığınızda imzanın değişme ihtimali oluyor. Bu yüzden bir klon oluşturup evo esnasında yapılca her şeyi o klon üzerinden yapın ki asıl imza değişmesin. Son olarak bu projede yapay zeka yok her şey sizin anlatımınıza ve karşısındaki anlamasına bağlı. Hazırlık bu kadardı devamında konuyla ilgili bazı detaylar var isteyen bakabilir.

DETAYLAR

LVMGroup Nedir?

LVM Group (Logical Volume Manager Group), birden fazla fiziksel depolama birimini bir araya getirerek, mantıksal olarak daha esnek bir şekilde yönetim sağlayan bir disk yönetim yapısıdır. Linux işletim sistemlerinde özellikle büyük veri kümeleriyle çalışırken sıkça kullanılır.

LVM Group'un temel bileşenleri şunlardır:

1. Physical Volume (PV):

- Fiziksel diski ya da bir disk bölümünü temsil eder.
- Örneğin: /dev/sda1, /dev/sdb.

2. Volume Group (VG):

- Birden fazla fiziksel birimi birleştirerek bir grup oluşturur.
- Bu grup, mantıksal birimlerin oluşturulacağı kaynak havuzudur.
- Örneğin: VG1, storage_vg.

3. Logical Volume (LV):

- Volume Group içinde oluşturulan ve dosya sistemiyle kullanılabilen mantıksal birimlerdir.
- LV'ler, ihtiyaç oldukça boyutlandırılabilir veya yeniden yapılandırılabilir.
- Örneğin: /dev/VG1/home, /dev/VG1/root.

APT NEDİR?

APT (Advanced Package Tool), Debian tabanlı Linux sistemlerinde paket yönetimi için kullanılan bir araçtır.

- **Görevi:** Yazılımları kolayca **kurmak, güncellemek, kaldırmak** ve **yönetmek**.
- **Komutlar:**
 - `apt install <paket>` → Paket kurar.
 - `apt update` → Depo bilgilerini günceller.
 - `apt upgrade` → Sistemdeki tüm yazılımları günceller.
 - `apt remove <paket>` → Paketi kaldırır.

APT, arka planda **dpkg** aracını kullanarak çalışır ve bağımlılıkları otomatik yönetir. Bu, yazılım yüklemeyi hızlı ve sorunsuz hale getirir.

APTITUDE NEDİR?

Gelişmiş Arayüz:

- **Konsol tabanlı bir menü** sunar, böylece paketleri bir arayüz üzerinden görüp yönetebilirsiniz.
- Komut satırı modunda da çalışır.

APT'ye Göre Avantajları:

- **Bağımlılıkları daha iyi yönetir** ve çözüm önerileri sunar.
- Daha detaylı hata mesajları verir.
- APT'nin yaptığı tüm işleri (kurma, güncelleme, kaldırma) yapabilir.

Kullanımı:

- aptitude yerine apt gibi komutlarla kullanılabilir:
 - `aptitude install <paket>` → Paket yükler.
 - `aptitude remove <paket>` → Paket kaldırır.
 - `aptitude search <paket>` → Paket arar.

APT ve APTITUDE ARASINDAKİ FARK

APT ve **Aptitude**, Debian tabanlı Linux sistemlerde kullanılan iki paket yönetim aracıdır. İşlevleri benzer olsa da, kullanım ve özellik açısından farklılıkları vardır:

1. Kullanım Farklılıkları

- **APT:**
 - Daha basit ve hızlıdır.
 - Daha az kaynak kullanır.
 - Komut satırında sık kullanılan temel paket yönetimi işlevlerini gerçekleştirir.
 - **Aptitude:**
 - **Konsol tabanlı bir arayüzü** vardır (kullanıcı dostudur).
 - Bağımlılık sorunlarında daha iyi öneriler sunar.
 - Daha karmaşık komutlar ve özellikler içerir.
-

2. Bağımlılık Yönetimi

- **APT:**
 - Bağımlılıkları yükler, ancak bağımlılık sorunları ortaya çıkarsa çözümü size bırakır.
 - **Aptitude:**
 - Bağımlılık sorunlarını analiz eder ve farklı çözüm yolları önerir.
-

3. Modernlik

- **APT:**
 - Debian ve Ubuntu gibi sistemlerde standart hale gelmiştir.
 - Geliştiriciler APT'yi daha aktif günceller ve destekler.

- **Aptitude:**

- Daha eski bir araçtır.
- Özellikle bağımlılık çözümü gerektiren karmaşık sistemlerde tercih edilebilir.

4. Komut Farklılıkları

Görev	APT	Aptitude
Paket Yükleme	apt install paket	aptitude install paket
Paket Kaldırma	apt remove paket	aptitude remove paket
Paket Arama	apt search paket	aptitude search paket
Güncelleme	apt update && apt upgrade	aptitude update && aptitude upgrade

Hangisini Kullanmak Daha Avantajlı?

- **APT:** Daha hızlı ve yaygın olduğu için modern sistemlerde önerilir.
- **Aptitude:** Bağımlılık sorunlarıyla uğraşıyorsanız veya kullanıcı dostu bir arayüze ihtiyacınız varsa tercih edilebilir.

APPARMOR NEDİR?

AppArmor, Linux'ta çalışan uygulamalar için erişim kontrolü sağlayan bir güvenlik modülüdür.

- **Ne yapar?:** Uygulamaların yalnızca izin verilen kaynaklara erişmesini sağlar (örneğin dosyalar, ağ, sistem çağrıları).
- **Nasıl çalışır?:** Her uygulama için bir profil tanımlanır, bu profil uygulamanın yapabileceklerini sınırlar.
- **Modlar:**
 - **Complain Mode:** İhlalleri sadece loglar.
 - **Enforce Mode:** İhlalleri engeller.
- **Kullanım:**
 - `sudo aa-status`: Profilleri görüntüler.
 - `sudo systemctl restart apparmor`: AppArmor'u yeniden başlatır.

Ubuntu gibi sistemlerde varsayılan olarak gelir ve SELinux'a daha kolay bir alternatiftir.

SELINUX NEDİR?

SELinux (Security-Enhanced Linux), Linux çekirdeği için geliştirilmiş bir güvenlik modülüdür. Sistem güvenliğini artırmak için **Zorunlu Erişim Kontrolü (MAC)** mekanizması kullanır.

Ne Yapar?

- Kullanıcıların, süreçlerin ve uygulamaların sistem kaynaklarına erişimini **katı kurallarla sınırlar**.
- Sistem yöneticisi, hangi işlemlerin hangi kaynaklara erişebileceğini belirler.

Özellikleri

1. Erişim Kontrolü:

- Dosyalara, ağ bağlantılarına ve sistem çağrılarına erişimi sınırlar.

2. Politika Tabanlı Yönetim:

- Detaylı güvenlik politikaları ile süreçleri izler ve denetler.

3. Modlar:

- **Enforcing:** Kuralları uygular ve ihlalleri engeller.
- **Permissive:** Kuralları loglar ama engellemez.
- **Disabled:** SELinux devre dışıdır.

Kullanımı

- Durumu kontrol etmek:
sestatus
- Mod değiştirmek:
sudo setenforce 0 # Permissive moda alır
sudo setenforce 1 # Enforcing moda alır

SELINUX VE APPARMOR ARASINDAKİ FARK

SELinux ve **AppArmor**, Linux sistemlerinde güvenlik sağlamak için kullanılan iki farklı zorunlu erişim kontrolü (MAC) mekanizmasıdır. İşlevsel olarak benzerlikleri olsa da, uygulama yöntemleri ve kullanım kolaylığı açısından farklılık gösterirler.

1. Temel Farklar

Özellik	SELinux	AppArmor
Politika Türü	Etiket (Label) tabanlı	Yol (Path) tabanlı
Karmaşıklık	Daha karmaşık ve güçlü	Daha kullanıcı dostu ve basit
Kullanım Alanı	Red Hat, CentOS, Fedora gibi dağıtımlarda varsayılan	Ubuntu, Debian gibi dağıtımlarda varsayılan

Özellik	SELinux	AppArmor
Profil Yönetimi	Ayrıntılı, dinamik etiketleme	Belirli dosya ve uygulama yollarına bağlı
Performans	Biraz daha yüksek sistem yükü oluşturabilir	Daha hafif bir çözüm

2. Çalışma Yöntemleri

- **SELinux:**
 - Sistem kaynaklarını **etiketler** (ör. dosyalar, süreçler) ve etiketler üzerinden erişim kurallarını uygular.
 - Daha ayrıntılı ve esnek kontrol sağlar ama yapılandırması zordur.
 - **AppArmor:**
 - Uygulama veya dosya yollarına dayalı profiller kullanır.
 - Kullanımı daha kolaydır, özellikle küçük ölçekli sistemlerde tercih edilir.
-

3. Modlar

Mod	SELinux	AppArmor
Zorlayıcı	"Enforcing"	"Enforce Mode"
İzin Verici	"Permissive"	"Complain Mode"
Devre Dışı	"Disabled"	"Disabled"

4. Yönetim Araçları

Görev	SELinux	AppArmor
Durum kontrolü	sestatus	aa-status

Görev	SELinux	AppArmor
Politika deęiřtirme	semanage	aa-complain, aa-enforce

Hangisi Daha İyi?

- **SELinux:**
 - Büyük ve karmařık sistemler için daha güçlü ve güvenlidir.
 - Ancak yapılandırması daha zordur.
- **AppArmor:**
 - Daha küçük sistemler ve kolay yönetim isteyen kullanıcılar için uygundur.
 - Daha az karmařık ve hızlı bir çözüm sunar.

SSHD NEDİR?

SSHD, SSH hizmetini sunan bir **sunucu tarafı yazılımıdır**. Tam adı "Secure Shell Daemon"dır. SSHD, gelen SSH bağlantı taleplerini dinler, bunları işler ve uygun kimlik doğrulama işlemlerini gerçekleştirir.

Özellikleri:

1. SSH bağlantıları için bir **arka plan hizmeti** olarak çalışır.
2. Genelde bir sunucuda başlatılır ve **22 numaralı portta** istemcilerden gelen bağlantıları bekler.
3. Kullanıcı doğrulama, şifreleme, bağlantı yönetimi gibi işlemleri gerçekleştirir.

SSH VE SSHD ARASINDAKİ FARK

Özellik	SSH	SSHD
Amacı	SSH, istemci tarafıdır ve kullanıcıların sunuculara bağlanmasını sağlar.	SSHD, sunucu tarafıdır ve bağlantıları kabul eder, yönetir.
Kim Kullanır?	Kullanıcılar, SSH istemcisiyle sunucuya bağlanır.	Sunucu, SSHD hizmetini çalıştırarak bağlantıları bekler.
Çalışma Şekli	Kullanıcı tarafından çalıştırılır.	Arka planda çalışan bir hizmettir (daemon).
Örnek Yazılımlar	OpenSSH istemcisi.	OpenSSH sunucusu (sshd).

KULLANILAN KOMUT TANIMLARI

dpkg -l | grep <kelime> : “dpkg” bir paket yöneticisidir. Bilgisayarda ki var olan dosyalarla ilgili işlem yapmamız için kullanırız. Bu komutu “-l” ile kullandığımızda tüm paketleri sıralamasını isteriz. “|” işareti “ile çalış” anlamına gelir. Bu işaretin yanına yazdığımız şey “dpkg -l” komutu ile çalışacak. “grep <kelime>” kısmı ise “dpkg -l” komutunun sadece belli bir kelimeye göre çalıştırılması anlamına gelir. Ör. : **dpkg -l | grep ufw**
Bu örnek komutun çıktısında içinde “ufw” geçen satırlar gözükür.

getent <veritabanı> <anahtar> : “getent” sistemdeki veritabanlarını sorgulamak için kullanılan bir komuttur. <veritabanı> ile belirtilen veritabanını ya komple ya da <anahtar> ile sorgular. Ör. : **getent group sudo**

Bu örnek komutta “group” adlı veritabanında “sudo” grubunu aramasını istedim.

Eğer “sudo” diye bir grup varsa ekrana (varsa) içindeki veri ile yazılacak.

Örn. Çıktı: **sudo:x:27:beergin**

cut -d: -f1 /etc/passwd : Bu kod bir dosyanın içindeki verileri belli bir ayraça göre kesip bize verir. “cut” kısmı zaten adından da anlaşılacağı üzere kesmek için kullanılıyor. “-d” kısmı bir ayraç belirtmemiz için kullanılır. Bu ifadeden sonraki gelen ilk karakter ayraç olarak belirlenir. Biz ayraç için “:” kullanmışız. “-f” ifadesi sütun belirtmemiz için kullanılan bir ifade. Bu ifadeden sonra gelen ilk sayısal değer sütun olarak ele alınır. Kodda “1” yapmışız bu durumda ilk sütunu alıcak. Sonra gelen “etc/passwd” kısmı ise dizin. Tüm bu işlemlerin hangi dizinde uygulanacağını belirtiyoruz.

Yukardaki örnek için bakalım.

sudo:x:27:beergin

Ayracımız “:” idi. Kod uygulandığında ilk aşamada ayraçtan sonra gelen her kelimeyi bölüyor (ft_split yaptıysanız onun gibi). “-f1” kullandığımızda bu bölünenlerden ilki yani ilk sütunu alıyor. Bu stringin sütunlarını aşağıdaki gibi hayal edebilirsiniz:

sudo	x	27	beergin
1	2	3	4

Koda göre bana “sudo” yu vericek. Eğer “-f2” olarak yapsaydım 2. Sütun olan “x” i verecekti. Olay bu kadar.

TTY: TTY, kullanıcı ile işletim sistemi arasında etkileşim sağlamak için kullanılır. Modern bilgisayarlarda genellikle **komut satırı erişimi** veya **terminal arayüzü** anlamına gelir.

TTY'nin İşlevleri ve Ne İşe Yaradığı

1. Kullanıcı Girişi Sağlama:

- TTY, kullanıcıların sisteme giriş yapmasını sağlar. Örneğin, Linux'ta bir kullanıcı oturum açtığında, sistem bir TTY kullanır.

2. Komut Çalıştırma:

- Komut satırı üzerinden işletim sistemine talimatlar göndermek için kullanılır.
- Örneğin, ls, cd, veya mkdir gibi komutları çalıştırarak dosyaları listelemek, izin değiştirmek veya yeni bir izin oluşturmak mümkün olur.

3. Uzak Sunuculara Bağlantı (SSH ile):

- SSH kullanarak bir sunucuya bağlandığınızda, TTY benzeri bir ortam oluşturulur ve bu sayede sunucuya komut gönderilebilir.

4. Fiziksel Terminal Erişimi:

- Eğer bir sunucu veya bilgisayarın ekranı bağlı değilse, seri port üzerinden fiziksel TTY cihazları kullanılarak sisteme erişim sağlanabilir.

5. Sanal Konsollar (Virtual Consoles):

- Linux'ta **Ctrl + Alt + F1-F6** gibi tuş kombinasyonlarıyla sanal terminaller arasında geçiş yapılabilir. Bu, bir kullanıcı komut satırı tabanlı bir oturumda çalışırken başka bir terminal açmasına olanak tanır.

6. Başka Kullanıcılarla Paralel Çalışma:

- Birden fazla kullanıcı farklı sanal TTY'lerde çalışabilir. Örneğin, bir kullanıcı tty1 üzerinde çalışırken, başka bir kullanıcı tty2 üzerinden çalışabilir.

7. Grafik Ortam ile Komut Satırı Geçişi:

- Linux sistemlerinde grafik arayüz (örneğin, GNOME, KDE) genellikle tty7 veya başka bir sanal terminal üzerinde çalışır. Kullanıcılar bu grafik oturumdan komut satırı tabanlı oturumlara geçiş yapabilir.

Monitoring.sh kodları:

arc=\$(uname -a): İşletim sisteminin temel bilgilerini “arc” adında bir değişkende tutar.

pcpu=\$(grep "physical id" /proc/cpuinfo | sort | uniq | wc -l):

grep "physical id" /proc/cpuinfo: Her işlemci fiziksel kimliğini /proc/cpuinfo dosyasından alır.

sort | uniq: Aynı "physical id" değerlerini birleştirir.

wc -l: Sonuçtaki satır sayısını sayar, yani fiziksel işlemci sayısını verir.

vcpu=\$(grep "^processor" /proc/cpuinfo | wc -l):

grep "^processor" /proc/cpuinfo: Başında “processor” olan satırları sayar.

wc -l: İşlemci sayısını sayar ve sanal işlemci sayısını verir.

fram=\$(free -m | awk '\$1 == "Mem:" {print \$2}'):

free -m: RAM kullanım bilgilerini MB cinsinden verir.

awk '\$1 == "Mem:" {print \$2}':

- awk komutu ile, free çıktısındaki ilk sütunda (\$1) "Mem:" olan satır seçilir.
- Bu satırda, toplam RAM miktarı 2. sütunda (\$2) yer alır ve bu değer yazdırılır.

uram=\$(free -m | awk '\$1 == "Mem:" {print \$3}'):

awk '\$1 == "Mem:" {print \$3}':

- free çıktısındaki "Mem:" satırından kullanılan RAM miktarı 3. sütunda yer alır.
- Bu değeri alır ve yazdırır.

pram=\$(free | awk '\$1 == "Mem:" {printf("%.2f"), \$3/\$2*100}');

awk '\$1 == "Mem:" {printf("%.2f"), \$3/\$2*100}':

- awk komutu, "Mem:" satırındaki kullanılan RAM (\$3) ve toplam RAM (\$2) değerleriyle RAM kullanım oranını hesaplar.
- Hesaplanan oranı yüzde olarak yazdırır. %.2f ifadesi, sonucu iki ondalıklı şekilde yazdırır.

fdisk=\$(df -Bg | grep '^/dev/' | grep -v '/boot\$' | awk '{ft += \$2} END {print ft}');

df -Bg: Disk alanını GB cinsinden gösterir.

grep -v '/boot\$': Sonu “/boot” ile biten satırları görmezden gelir.

awk '{ft += \$2} END {print ft}':

- awk ile her satırdaki 2. sütundaki (toplam disk alanı) değeri alır ve biriktirir (ft += \$2).
- END {print ft}: Tüm satırlar işlendikten sonra birikmiş toplam değeri yazdırır.

udisk=\$(df -Bm | grep '^/dev/' | grep -v '/boot\$' | awk '{ut += \$3} END {print ut}');

awk '{ut += \$3} END {print ut}':

- Burada da, her satırdaki 3. sütunda bulunan (kullanılan disk alanı) değeri alır ve biriktirir (ut += \$3).
- END {print ut}: Tüm satırlar işlendiğinde toplam değeri yazdırır.

pdisk=\$(df -Bm | grep '^/dev/' | grep -v '/boot\$' | awk '{ut += \$3} {ft+= \$2} END {printf("%d"), ut/ft*100}');

awk '{ut += \$3} {ft+= \$2} END {printf("%d"), ut/ft*100}':

- Bu kısımda, her satırdaki 3. sütundaki (kullanılan disk) ve 2. sütundaki (toplam disk) değerleri toplanır.
- $ut/ft \times 100$ ile disk kullanım oranı hesaplanır ve yüzde olarak yazdırılır (%d tam sayı biçiminde).

cpul=\$(top -bn1 | grep '^%Cpu' | cut -c 9- | xargs | awk '{printf("%.1f%%"), \$1 + \$3}'):

top -bn1

- top komutu, sistemin anlık performansını ve kaynak kullanımını gösteren bir komuttur. top komutunun çıktısı sürekli güncellenir.
- -b: Bu seçenek, top komutunu batch mode'a alır; yani sürekli güncellenmeden sadece tek bir kez çalışmasını sağlar.
- -n1: Bu seçenek, top komutunun yalnızca bir kez çalışmasını sağlar. Yani 1 defa çıktı alırız.

cut -c 9-

- **cut** komutu, metin üzerinde sütunlar kesmeye yarar.
- -c 9-: Bu seçenek, her satırın 9. karakterinden sonrasını alır. Çünkü, %Cpu satırında CPU kullanım yüzdeleri genellikle 9. karakterden başlar.

xargs

- **xargs** komutu, gelen veriyi tek bir satıra dönüştürür ve boşluklarla ayırır.
- Bu, cut komutundan gelen birden fazla değeri (örneğin, 7.2 us, 2.3 sy, 0.0 ni, 90.3 id) tek bir satırda birbirinden ayırarak alır.

awk '{printf("%.1f%%"), \$1 + \$3}'

- **awk** komutu, veriyi işlemenin güçlü bir yoludur. Burada \$1 ve \$3 değişkenleri, metni alanlar (fields) olarak temsil eder:
 - \$1: İlk sayıyı (kullanıcı CPU zamanı) alır, yani 7.2'yi.
 - \$3: Üçüncü sayıyı (sistem CPU zamanı) alır, yani 2.3'ü.

- **\$1 + \$3:** Kullanıcı ve sistem CPU kullanım yüzdelerini toplar.
 - $7.2 + 2.3 = 9.5$ (örnek olarak).
- **printf("%.1f%%", \$1 + \$3):** Bu işlem sonucu %9.5 formatında, bir ondalıklı sayı (örneğin 9.5%) çıktısı üretir.

lb=\$(who -b | awk '\$1 == "system" {print \$3 " " \$4}):

awk '\$1 == "system" {print \$3 " " \$4}':

- **who -b** komutundan gelen son başlatılma tarihi verilerini işler.
- awk ile "system" kelimesini içeren satır bulunur ve bu satırdaki 3. ve 4. sütunlar (tarih ve saat) yazdırılır.

lvmt=\$(lsblk | grep "lvm" | wc -l):

lsblk komutu, sistemdeki tüm blok cihazlarını (diskler, bölümler, sürücüler vb.) listelemek için kullanılır. Çıktısı, fiziksel ve mantıksal diskleri ve onların sahip oldukları bölümleri (partitions) içerir.

grep "lvm": Bu komut, lsblk çıktısını tarar ve sadece LVM ile ilgili satırları arar. LVM genellikle disk adlarında lvm ifadesini içerir.

wc -l: grep'in çıktısını sayar ve bu satırların sayısını verir. Sonuç, LVM kullanılan disk sayısını sayısal olarak döndürecektir.

lvmu=\$(if [\$lvmt -eq 0]; then echo no; else echo yes; fi):

if [\$lvmt -eq 0]: Bu, lvmt (LVM kullanımı sayısı) değişkeninin 0 olup olmadığını kontrol eder. Eğer 0 ise, yani sistemde LVM kullanılmıyorsa:

- **echo no:** LVM kullanılmıyor anlamına gelir.

Eğer lvmt değeri 0 değilse, yani sistemde LVM kullanılıyorsa:

- **echo yes:** LVM kullanılıyor anlamına gelir.
- **fi:** En başta yazdığımız “if” koşulunun bittiği anlamına gelir.

ctcp=\$(cat /proc/net/sockstat{,6} | awk '\$1 == "TCP:" {print \$3}')

cat /proc/net/sockstat{,6}

- /proc/net/sockstat dosyası, sistemdeki ağ bağlantıları hakkında istatistikleri içerir. Bu dosya, TCP ve UDP soket bağlantılarını, aktif bağlantı sayıları ve bağlantı türleri hakkında bilgi sağlar.
- /proc/net/sockstat6 ise IPv6 bağlantılarına dair aynı bilgileri içerir.
- {,6} kullanımı, hem /proc/net/sockstat hem de /proc/net/sockstat6 dosyalarını birden okur. Bu, hem IPv4 hem de IPv6 bağlantılarını bir arada almanızı sağlar.

| awk '\$1 == "TCP:" {print \$3}'

- |: Bu bir "pipe" komutudur ve önceki komutun çıktısını, bir sonraki komutun girdi olarak kullanmasına olanak tanır.
- awk '\$1 == "TCP:" {print \$3}': awk komutu, verileri satır satır işler ve belirli alanlara odaklanır. Buradaki awk komutu şu şekilde çalışır:
 - \$1 == "TCP:": awk komutu, her satırın ilk kolonunda (\$1) "TCP:" ifadesi olup olmadığını kontrol eder. Bu, TCP bağlantılarının başlangıcını belirtir.
 - {print \$3}: Eğer bir satır "TCP:" ile başlıyorsa, o satırın 3. sütunundaki değeri (\$3) yazdırır. Bu sütun, o anda aktif olan TCP bağlantı sayısını içerir.

ulog=\$(users | wc -w):

Bu komut, sisteme giriş yapmış aktif kullanıcı sayısını hesaplar:

1. users: Sistemde oturum açmış kullanıcıları listeler.
2. wc -w: Listelenen kullanıcıları kelime bazında sayar, yani aktif kullanıcı sayısını verir.

Sonuç olarak, ulog değişkeni aktif kullanıcı sayısını tutar.

ip=\$(hostname -I): Sistem IP adreslerini listeler.

mac=\$(ip link show | awk '\$1 == "link/ether" {print \$2}'):

awk '\$1 == "link/ether" {print \$2}':

- **ip link show** komutundan ağ arayüzü bilgilerini alır.
- link/ether satırındaki 2. sütunda MAC adresi bulunur ve yazdırılır.

cmds=\$(journalctl _COMM=sudo | grep COMMAND | wc -l):

Bu komut, sudo komutuyla çalıştırılmış komut sayısını bulur:

1. journalctl _COMM=sudo

- **journalctl:** Sistem günlüklerini (log) görüntülemek için kullanılan bir komuttur.
- **_COMM=sudo:** journalctl komutuna verilen bu filtre, sadece sudo komutuyla çalıştırılmış işlemleri gösterir. Yani, yalnızca sudo kullanılarak yapılan işlemler kaydedilecektir.

2. | grep COMMAND

- **grep COMMAND:** journalctl çıktısındaki satırlarda "COMMAND" kelimesini arar. Bu, sudo komutuyla çalıştırılan komutların başlık kısmındaki COMMAND anahtarını arar ve sadece bu satırları alır.

3. | wc -l

- **wc -l:** wc komutu, girdi olarak aldığı satır sayısını döndürür. Bu komut, grep tarafından filtrelenmiş satırları sayarak, sudo ile çalıştırılmış toplam komut sayısını verir.

wall "... mesaj içeriği ...":

wall: Tüm terminallere bir mesaj yayınlar.

Yukarıda toplanan bilgileri formatlı bir şekilde yazdırır.

crontab komutu:

***/10 * * * * bash /usr/local/bin/monitoring.sh:** Bu kod belirlediğimiz bir işlemi belirlediğimiz bir sürede çalıştırmayı sağlar. Kodun şeması şöyledir:

* * * * * komut

| | | |
| | | | _____ Hafta günü (0 - 6) (Pazar = 0 veya 7)
| | | |
| | | _____ Ay (1 - 12)
| | |
| | _____ Gün (1 - 31)
| |
| _____ Saat (0 - 23)
|
| _____ Dakika (0 - 59)

Bizim kullanımımızda şu şekilde çalışır:

***/10:** Bu kısım komutun 10 dakikada bir çalışmasını sağlar. Eğer başına “*/” koymasaydık her 10 dakikada bir değil her saatin 10. Dakikasında çalışırdı.

***:** Eğer alanlara “*” koyarsak bu o alanın görmezden gelineceğini belirtmiş oluruz.

NOT: 0 ve “*” aynı şey değildir. 0’ında burada bir işlevi var.

bash /usr/local/bin/monitoring.sh: Bu kısım ise belirlediğimiz süre geldiğinde çalışacak olan komutu temsil ediyor. Her 10 dakikada bir monitoring.sh dosyamız çalışacak.

PHP:

phpinfo(): Bu fonksiyon, PHP yapılandırması hakkında çok kapsamlı bir bilgi sağlar.

Çalışan PHP sürümü, yapılandırma ayarları, yüklü modüller, PHP'nin derlendiği seçenekler, çevresel değişkenler, yüklü uzantılar (extensions) gibi çok sayıda bilgi içerir.

Genellikle PHP'nin doğru çalışıp çalışmadığını kontrol etmek veya geliştiricinin PHP ortamını görmesi amacıyla kullanılır.

phpinfo(INFO_MODULES):

phpinfo() fonksiyonunun bir parametresi olarak kullanılır.

INFO_MODULES parametresi, sadece **yüklü PHP modüllerinin ve uzantılarının bilgilerini** gösterir.

Bu, PHP'nin hangi modülleri yüklediği ve bu modüllerin durumları hakkında bilgi verir. Örneğin, mysqli, gd, mbstring gibi modüllerin yüklü olup olmadığını görebilirsiniz.

Veritabanı kodları:

CREATE DATABASE wordpress: “wordpress” adında bir veritabanı oluşturur.

GRANT ALL PRIVILEGES ON wordpress.* TO 'admin'@'localhost' IDENTIFIED BY 'Berat42':

Bu kod belirlenen kullanıcıya veritabanımızın tablolarına yönelik tüm izinleri verir. Adım adım gidelim

GRANT ALL PRIVILEGES ON wordpress.*: Buradaki “ALL PRIVILEGES” tüm yetkileri (INSERT, UPDATE, DELETE, CREATE vs.) içeriyor. Bu kod aracılığı ile “wordpress” adlı veri tabanındaki tüm tablolarda, belirlenen kullanıcıda tam yetki olmuş oluyor. Eğer tüm yetkiler yerine bir adet yetki vermek isteseydik (örn: update) şöyle bir güncelleme yapardık:

GRANT UPDATE ON wordpress.*

TO ‘admin’@‘localhost’: Bu kısım tüm yetkilerin hangi kullanıcıya atanacağını belirtir. “admin” kısmı kullanıcı adını, “localhost” kısmı ise bu kullanıcının hangi hosttan bağlanacağını belirtir. Bizim kodumuzda localhosttan bağlanacağımız için bu şekilde belirttik. Eğer uzak bir IP’den bağlanacak olsaydık o kısmı “%” şeklinde değiştirebilirdik.

Örnek Kullanım : **TO ‘admin’@’%’**

IDENTIFIED BY ‘Berat42’: Bu kod ise kullanıcıya şifre atamak için kullandığımız bir kod. “IDENTIFIED” kodu veri tabanı işlemlerinde kullanıcıya bir şifre atamak için kullanılan bir terimdir. Bu kod aracılığı ile “admin” adlı kullanıcının şifresi “Berat42” olarak ayarlanmış oldu.

-BU KADAR-