Lektion 7 – Laboration 1

Om du vill köra i VirtualBox gå till nästa sida istället!

1: Köra med podman eller docker från WSL eller linux

Ladda ner en image med docker eller podman så här:

docker pull docker.io/romram/ubuntu:2204-base-ssh

eller

podman pull docker.io/romram/ubuntu:2204-base-ssh

Starta sedan containern med:

podman run -it --rm –hostname=sandbox-base-ssh -p 2222:22 romram/ubuntu:2204-base-ssh

Du behöver två terminalfönster, i ett startar du containern enligt ovan och i det andra loggar du in och kör alla andra kommandon nedan. Antingen **en till flik** eller **ett till terminal-fönster**.

- Testa att logga in från wsl med lösenord: Använd **ssh -p 2222 user@localhost** för att ansluta till din docker-container. Lösenord till user är **user**
- Testa att logga in som en annan användare inifrån containern (utan att logga ut):
  ssh temp1@localhost. *OBS! inne i containern behöver vi inte använda specialport*
  Tips: Har du inte någon användare kan du skapa en.

2: Testa server-verifiering

- Se till att du är inloggad som **user@sandbox-base-ssh**:~$ ▉
- Skapa om nycklarna som identifierar servern:

```
sudo rm /etc/ssh/ssh_host*
sudo dpkg-reconfigure openssh-server
sudo service ssh restart
```

- **Utan att logga ut!** - testa att logga in med ssh igen. Det borde gå dåligt och du får en varning - läs den.
- Eftersom vi vet att vi inte är utsatta för en attack, följ instruktionerna från varningen för att ta bort den ”trasiga” raden i filen **known_hosts**.
- Logga in igen och acceptera serverns nya host-key.

3: Stäng av lösenordsinloggning

- I filen ”/etc/ssh/sshd_config”, sätt inställningen ”PasswordAuthentication” till ”no”.
- Starta om ssh-server:

```
sudo service ssh restart
```

- Testa att logga in med ssh igen. Det borde inte gå.

4: Skapa nyckelpar och logga in lokalt (från VM till VM)

- Använd ”ssh-keygen” för att skapa ett nyckelpar. Ange ett lösenord för att skydda din privata nyckel!
- Lägg till din publika nyckel i filen ”authorized_keys” i mappen ”~/.ssh”:

```
cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_keys
chmod 600 ~/.ssh/authorized_keys
```

- Testa att logga in igen, nu borde det fungera!

*Observera att ssh ber om **passphrase** till din privata nyckel, inte lösenordet till användaren user*

Varför ändrar vi rättigheterna på ”authorized_keys”? Diskutera!

Lektion 7 – Laboration 1

Om du hellre vill köra i docker/podman gå till föregående sida istället!

1: Installera ssh-server

- Installera paketet ”openssh-server”. Paket installeras med ”apt install <paketnamn>”. Kom ihåg att du måste vara superanvändare för att installera paket.
- Testa att logga in lokalt med lösenord. Använd ”ssh localhost” för att ansluta till din VM med ssh.
- Logga ut för att komma tillbaka till konsolens bash.
- Testa att logga in som en av dina andra användare med ”ssh user@localhost”. Har du inte kvar en extra användare från tidigare får du lägga till en ny.

2: Testa server-verifiering

- Skapa om nycklarna som identifierar servern:

```
sudo rm /etc/ssh/sshd_host*

sudo dpkg-reconfigure openssh-server
```

- Testa att logga in igen. Det borde gå dåligt.
- Eftersom vi vet att vi inte är utsatta för en attack, följ instruktionerna från ssh för att ta bort den ”trasiga” raden i ”known_hosts”.
- Logga in igen och acceptera det nya host-IDt.

3: Stäng av lösenordsinloggning

- I filen ”/etc/ssh/sshd_config”, sätt inställningen ”PasswordAuthentication” till ”no”.
- Starta om sshd:

```
sudo systemctl restart ssh
```

- Testa att logga in med ssh igen. Det borde inte gå.

4: Skapa nyckelpar och logga in lokalt (från VM till VM)

- Använd ”ssh-keygen” för att skapa ett nyckelpar. Ange ett lösenord för att skydda din privata nyckel!
- Lägg till din publika nyckel i filen ”authorized_keys” i mappen ”~/.ssh”:

```
cat id_rsa.pub >> authorized_keys

chmod 600 authorized_keys
```

- Testa att logga in igen, nu borde det fungera!

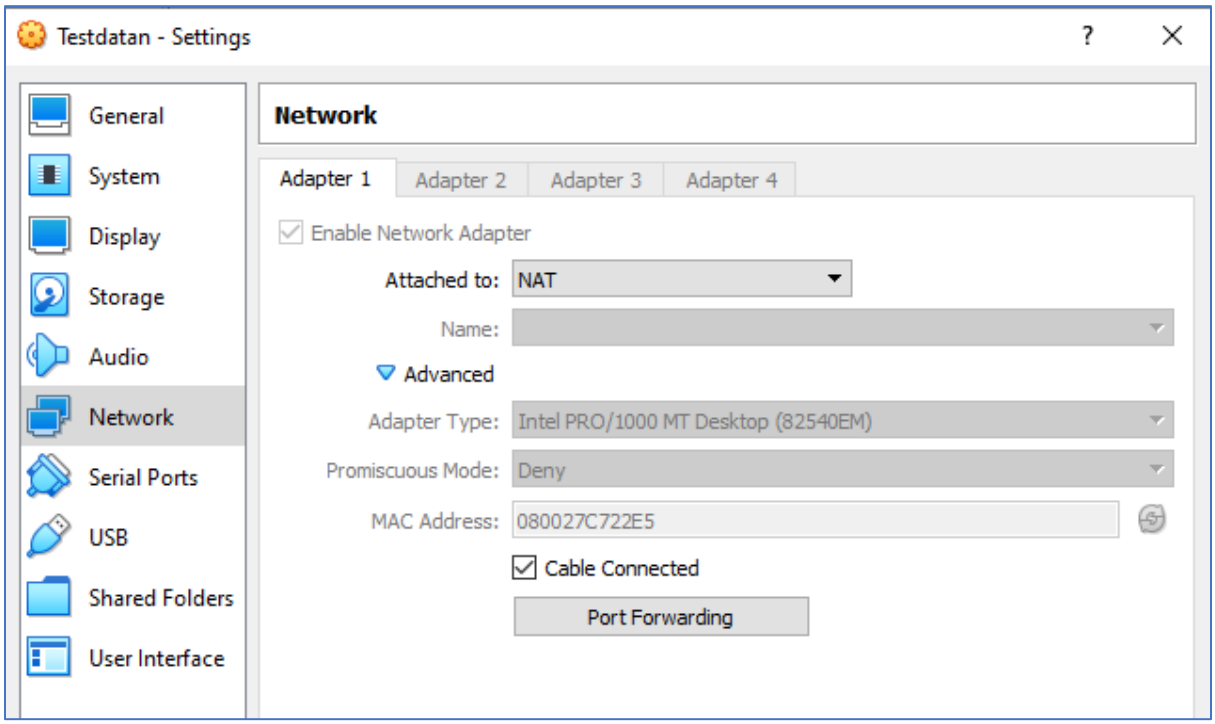
Observera att ssh ber om lösenordet till din privata nyckel, inte till användaren i din VM.

Varför ändrar vi rättigheterna på ”authorized_keys”? Diskutera!

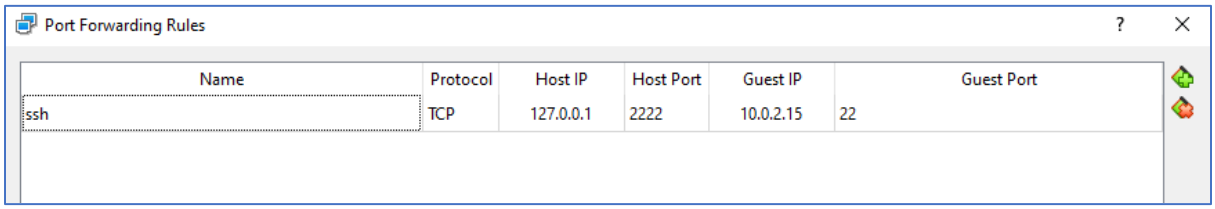
5: Logga in till din VM från din värd-dator - gäller ej docker

Ställ in port-forwarding för port 22 i VirtualBox:

- Använd ”ip addr” i din VM för att ta reda på vilken ip-address ditt ethernet-interface har
- Öppna nätverksinställningarna till din VM, klicka på ”Advanced” och gå till ”Port Forwarding” – dialogen:

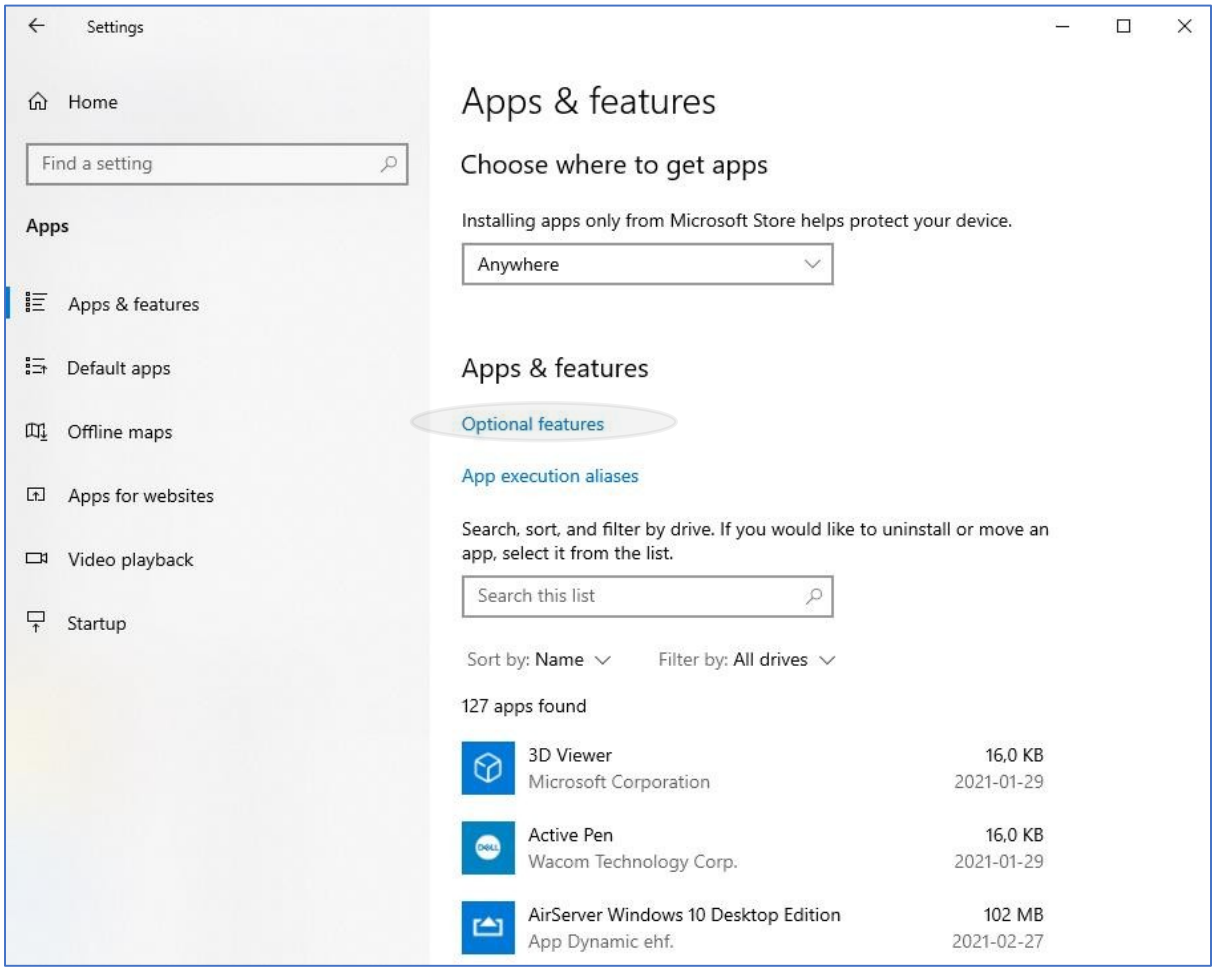


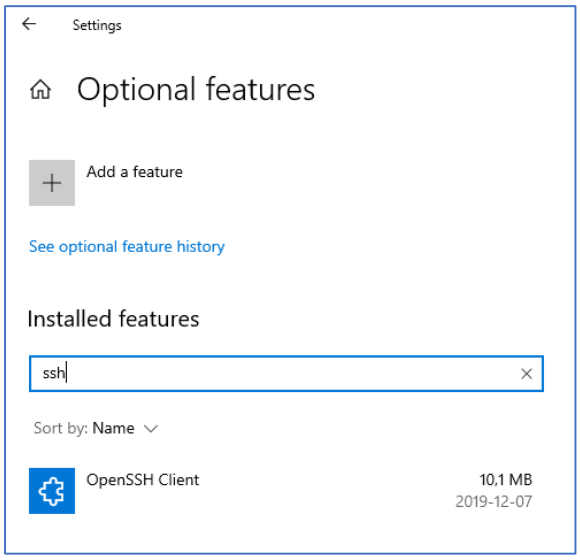
- Lägg till en ny rad med följande inställningar, byt Guest IP mot den ip-address du hittade tidigare, om den skiljer sig:



Det här gör så att man kan ansluta till ssh-servern i VM:en (som lyssnar på port 22) genom att på värd-datorn ansluta till port 2222.

- Om du kör Windows, installera ssh-klienten genom ”Settings/Apps & features”:





För en trevligare terminal på Windows kan du även installera ”Windows Terminal” från ”Microsoft Store”.

På macOS / Linux har du redan ssh – klienten installerad!

- Skapa ett nytt nyckelpar i din värddator med ”ssh-keygen”.
- Gå till din VM och tillåt lösenordsinloggning för nästa steg.
- Använd scp för att kopiera över din nya publika nyckel till din VM:
  scp -P 2222 ~/.ssh/id_rsa.pub frasse@localhost:another.pub
- Lägg till den publika nyckeln till ”~/.ssh/authorized_keys” i din VM som tidigare.

Du borde nu ha två rader i ”authorized_keys” – en för din lokala VM-användare, och en från din värd-dator.

- Slå av lösenordsinloggning igen.
- Logga in från värd-datorn till din VM:

```
ssh -p 2222 user@localhost
```

Byt ut ”user” mot användarnamnet i din VM. ”-p 2222” gör så att ssh-klienten använder port 2222 istället för standardvärdet 22.

Nu har du satt upp ssh så att du kan ansluta till din VM med flera olika terminaler och fönster, klippa och klistra, scrolla, et.c.