# Lektionstillfälle 7

Säker kommunikation
med Mikael Larsson

NACKADEMIN

# Återblick

Sist pratade vi om programexekvering:

- PATH
- Argument
- Inmatning, utmatning och omdirigering
- Förgrunds- och bakgrundsprocesser
- Jobbkontroll
- Exit codes
- Signaler

# Dagens lektion

**Mål: Att förstå och använda `ssh`, generera och installera nycklar.**

**TLCL s209-215**

- `ssh`, `scp`, `sftp`
- Serverinställningar
- Klientinställningar
- Nycklar och lösenordslös inloggning
- Tunnlar *
- **Inlämningsuppgift 1**

**överkurs*

# Termer och begrepp

- ssh
- OpenSSH
- ssh-nycklar
- Nätverksportar
- Tunnlar

# Vad är ssh?

**SSH** – Secure SHell, är ett protokoll och en samling verktyg för fjärråtkomst.

För att kunna koppla upp sig till en server via ssh måste man ha en **ssh-klient** och servern måste köra en **ssh-server**.

Vi kommer att använda oss av OpenSSH, som är standardvalet för de flesta Linux-distributionerna.

För klient-operationer kommer jag visa OpenSSH – verktygen, även om det finns många andra alternativ med GUI etc.

*ssh-servern kallas ofta ”sshd” – ssh daemon, eftersom tjänster i Linux går under namnet daemon.*

# Vad gör ssh säkert?

Det är framför allt tre saker som gör ssh säkert:

- Krypterad trafik
- Verifiering av servrar
- Nycklar för klientverifiering och lösenordslös inloggning

# Vad gör ssh säkert – simplifierad process

**Dubbel låsning**

- Sändaren låser lådan med sitt hänglås och skickar den till dig.
- Du låser lådan med ditt hänglås och skickar tillbaka lådan.
- Sändaren tar bort sitt lås och skickar lådan igen
- Du tar bort ditt hänglås och kommer åt innehållet.


*Komplicerat och mycket trafik fram och åter.*

# Vad gör ssh säkert – simplifierad process

**Publik och privat nyckel – asymmetrisk kryptering**

- Du förser sändaren med ett olåst hänglås.
- Sändaren lägger in innehållet och låser lådan med ditt öppna hänglås och skickar den till dig.
- Du använder din nyckel (privat) och låser upp lådan så att du kommer åt innehållet.


- *Öppna hänglås motsvarar **publik nyckel***
- *Nyckeln till hänglåsen är din **privata nyckel***
- *Lådan är ett, av många, TCP-paket*
- *Du kan utan risk dela ut öppna hänglås till vem som helst, de kan bara låsa inte öppna.*

# OpenSSH server

Installeras från paketet “openssh-server”:

```
sudo apt install openssh-server
```

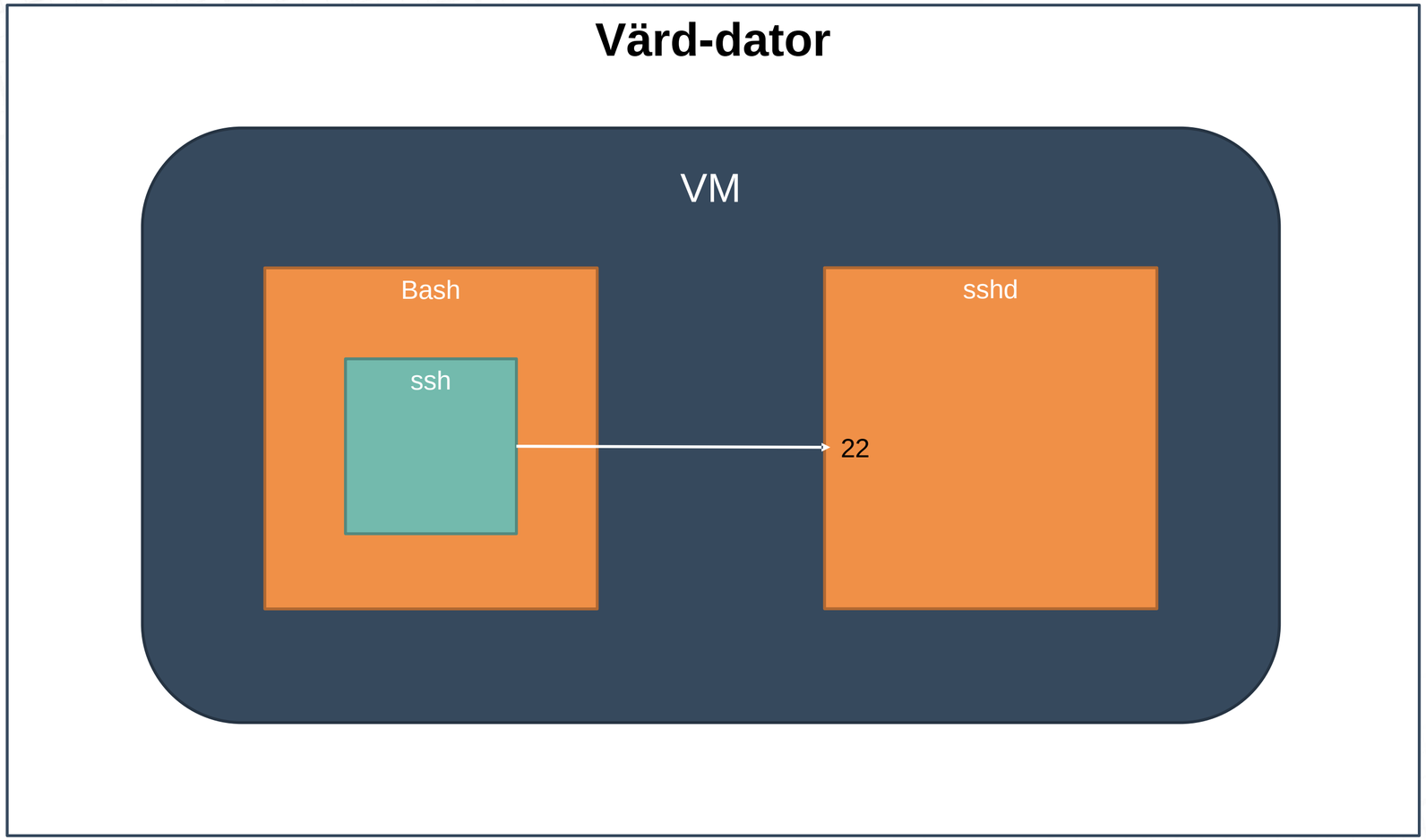
Konfigureras i “/etc/ssh/sshd_config”.
(man sshd_config)

För att ansluta med ssh till en server används kommandot ssh:

```
ssh user@server
```

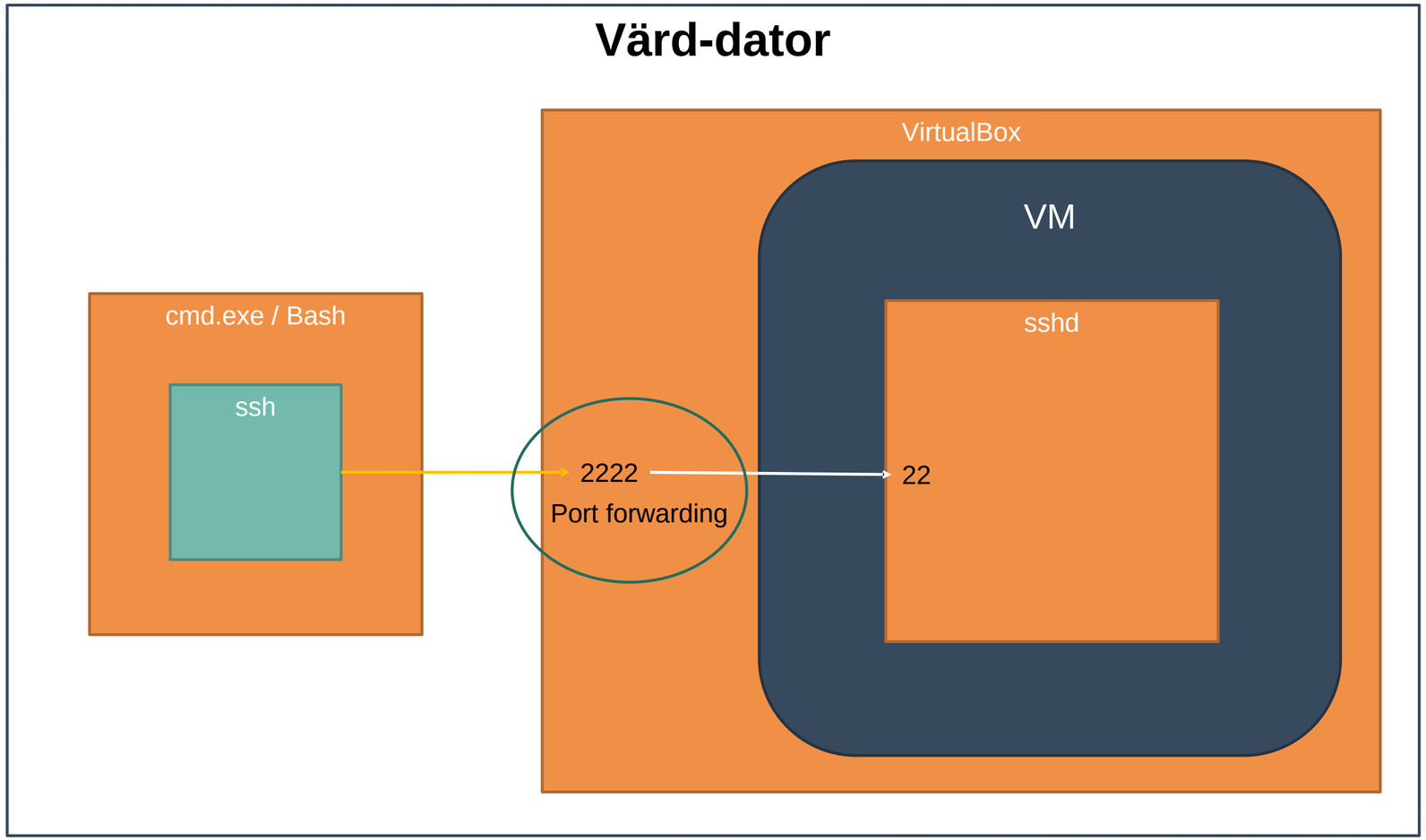
Om inte “user” anges används nuvarande inloggade användares namn.

# Lokal uppkoppling i VM




```
ssh user@localhost
```

# Uppkoppling från värd till VM




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

NACKADEMIN

# Inloggning med lösenord

Är ”säkert”, eftersom ingen kan läsa av lösenordet på vägen. Trafiken är krypterad.

Är mindre säkert, eftersom:

- Lösenord bör hållas unika, svårt med många servrar
- Lösenord är relativt korta och ofta lätta att kopiera
- Lösenord måste skrivas in vid varje inloggning
- Samma lösenord används oavsett varifrån inloggning sker
- Lösenordet kan plockas från ditt tangentbord med med en mellandel, akustiskt eller trådlöst

https://www.youtube.com/watch?v=2OjzI9m7W10

# Inloggning med nyckel

Baseras på ett nyckelpar med en privat och en publik del.

- Den privata delen måste hållas hemlig, och lämnar i princip aldrig din dator. Den är oftast skyddad med en passphrase.
- Den publika delen används för att verifiera den privata, och läggs in på alla servrar man ska kunna komma åt.

En Linux-användare på en server kan ha många olika nycklar som tillåter inloggning. Med andra ord kan flera sysops dela på ett konto, men ändå ha olika identiteter.

För att slippa skriva passphrase om igen kan **ssh-agent** hjälpa till att komma ihåg upplåsta privata nycklar.

***ssh-agent** körs på klienten, startas oftast vid login och skyddas på så sätt av användarens lokala inloggning.*

# Skapa nyckelpar

Alla ssh-klientverktyg sparar sina filer i ”~/.ssh” – mappen.

”ssh-keygen” skapar ett nytt nyckelpar, och ber om passphrase för att skydda den privata nyckeln.

Använd **alltid passphrase**, om du inte har mycket goda skäl att inte göra så.

OBS att din passphrase, och den privata nyckeln aldrig förs över till servern.

Standard RSA-nyckel anses osäkert. **Ed25519** är säkrare och dessutom snabbare:

```
ssh-keygen -o -a 100 -t ed25519 -f ~/.ssh/id_rsa -C "user@example.com"
```

# Filen authorized_keys

På en server innehåller filen ”~/.ssh/authorized_keys” en rad med en publik nyckel för varje tillåten identitet som får logga in.

OBS att filen ligger i hemkatalogen för den användare man vill logga in som.

Man skiljer alltså på ”inloggad identitet” och den ”användare” man loggar in som.

*För att slippa redigera ”authorized_keys” för hand finns ett verktyg ”ssh-copy-id” som kan lägga in publika nycklar från en klient till en server.*

*Men – verktyget finns inte överallt, och man måste ändå ha access på något sätt för att kunna lägga en nyckel.*

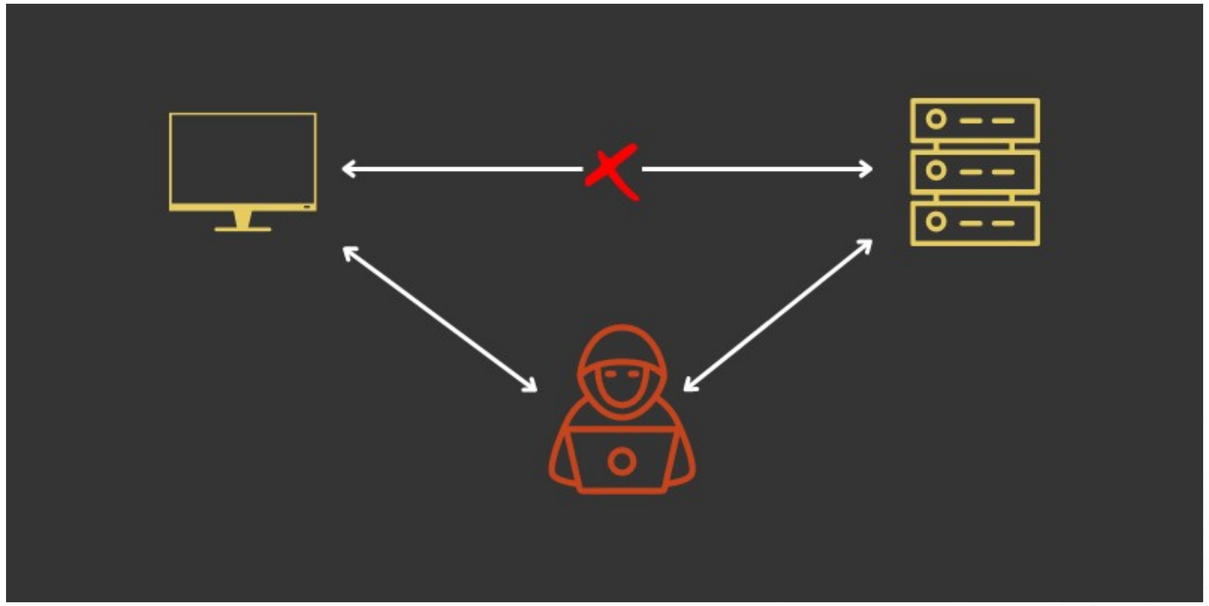
# Filen known_hosts

På en klient innehåller filen ”~/.ssh/known_hosts” en rad med en nyckel för varje ”host” eller ssh-server som man tidigare accepterat.

Om en host skulle byta nyckel kommer ssh vägra koppla upp sig, och varna användaren.

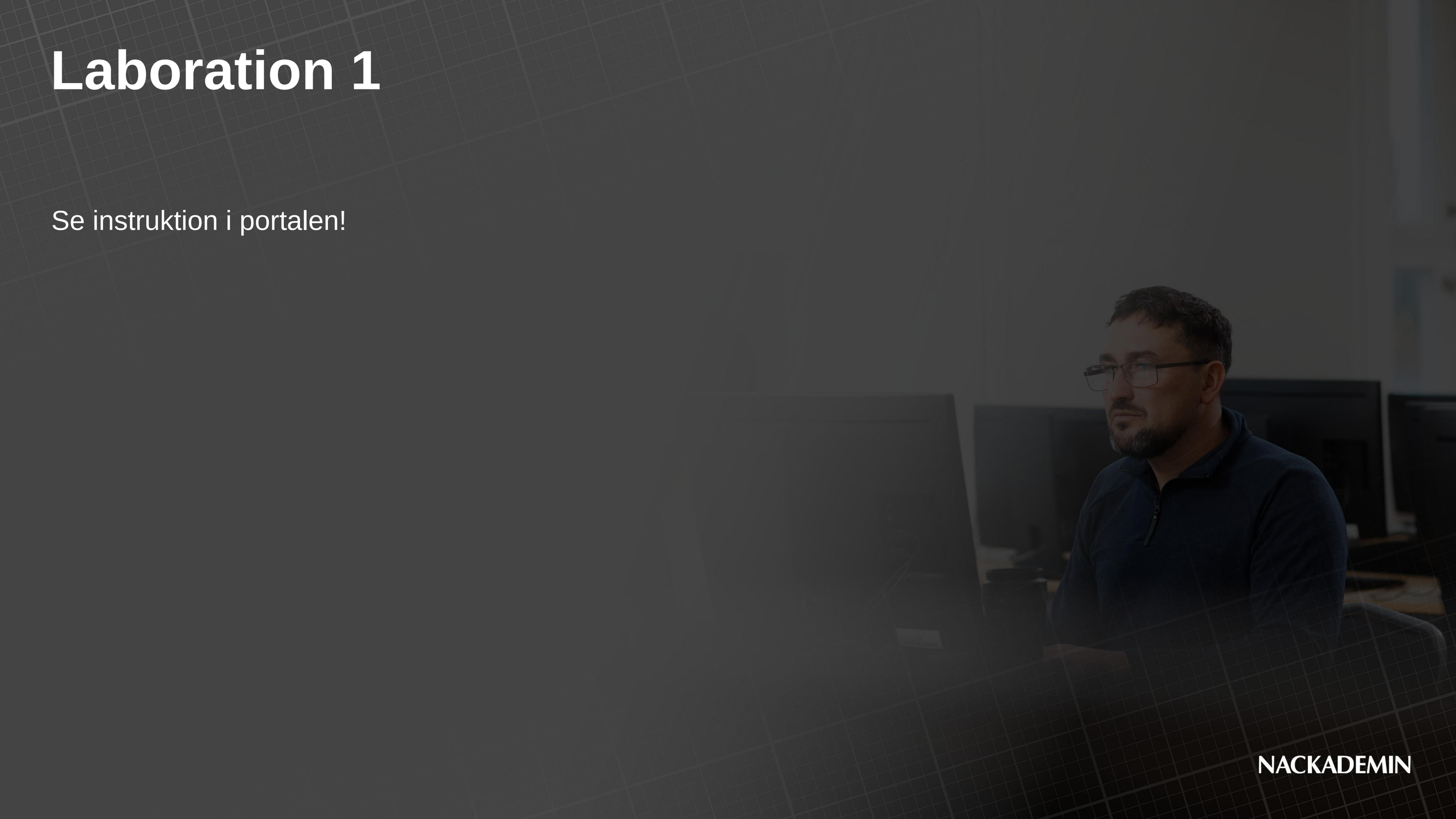
Det är samma sorts verifiering som servern gör av klienternas identiteter, fast tvärt om.

Om du får en sådan varning, tänk efter innan du accepterar den nya identiteten från servern.
I realiteten kan det vara en Man In the Middle-attack på gång.



# Laboration 1

Se instruktion i portalen!



NACKADEMIN

# ssh som generellt fjärrkommando

Man kan använda ssh för att köra kommandon:

```
ssh user@host kommando
```

Obs att det är en hel kommandorad som körs i skalet på servern, det går att skicka med pipes, wildcards, etc.

*Ansible – ett mycket vanligt verktyg för devops – använder sig bara av ssh för att komma åt de maskiner som ska arbetas på.*

*Det gör att kraven på förinstallerade tjänster för en ”naken” maskin kan hållas låga.*

# scp, sftp kopierar filer

scp använder ssh för att kopiera filer.
Det går att kopiera i båda riktningarna, upp eller ned.

Filnamnet hakas på serveradressen med ett kolon:

```
scp local.file user@host:remote.file
```

Om ingen destinationsfil anges behålls
originalnamnet. Men kolonet behövs fortfarande:

```
scp local.file user@host:
```

***sftp** ger ett mer interaktivt sätt att lista och ladda ner/upp filer. Undersök gärna själva!*

# ssh-tunnlar

En tunnel är ett sätt att skicka nätverkstrafik inkapslat i ett annat nätverkskoppel.

På så sätt man kan till viss del kringgå nätverksfilter eller komma åt maskiner som står innanför en brandvägg.

Tunnlar kan också göra så att program på klienten kan komma åt tjänster på servern, eller tvärt om.

ssh – tunnlar är så vanliga att de är inbyggda i många andra klienter. Exempelvis DBeaver, en databasklient.

Vissa molntjänster kräver att man använder ssh – tunnlar för att använda säkerheten i ssh till att säkra upp mindre säkra protokoll.

# Användarspecifika serverinställningar

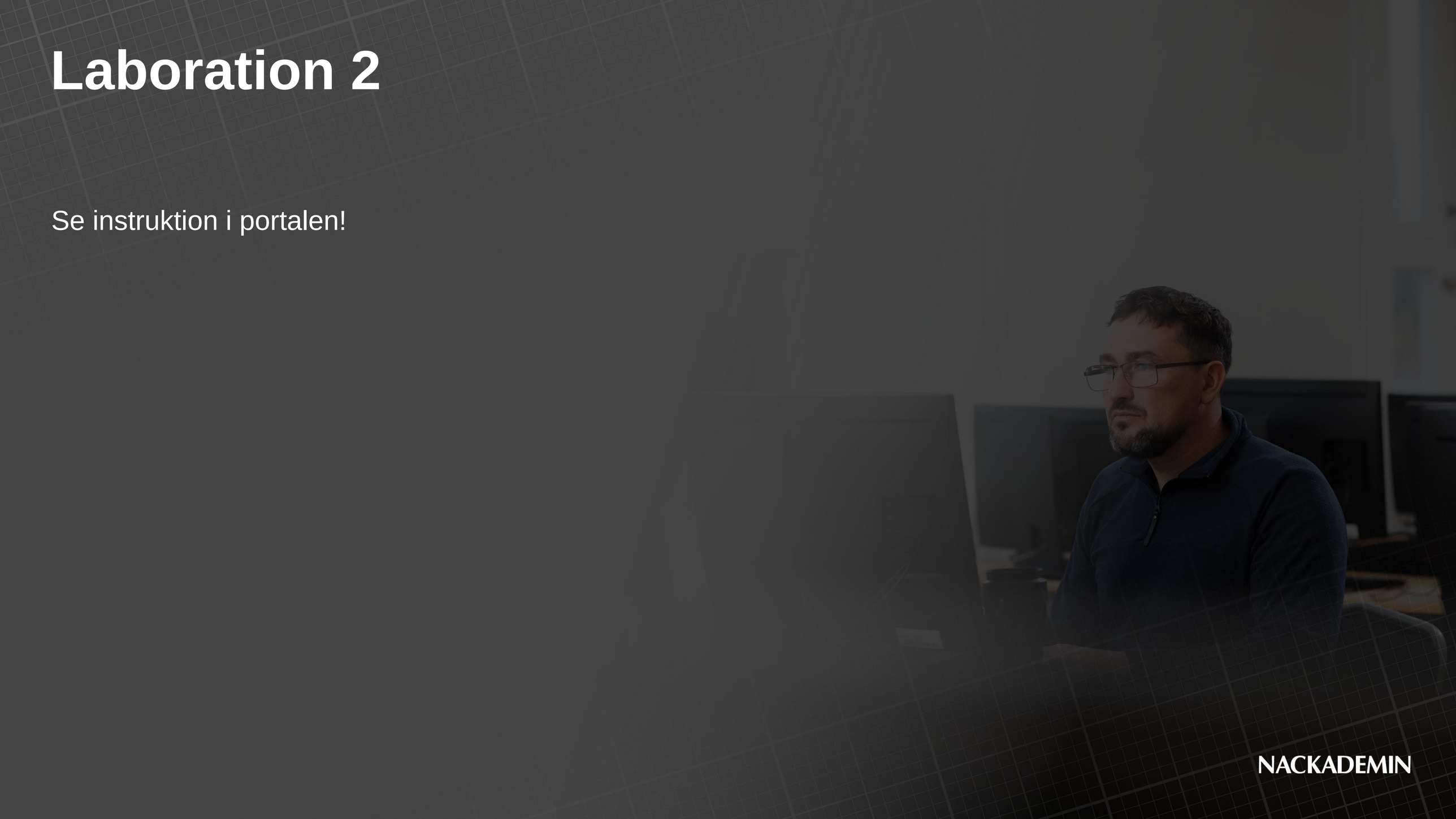
Förutom den globala inställningen för alla användare som ansluter till en ssh-server kan man in OpenSSL ställa in inställningar för specifika användare.

Exempelvis man kan tillåta lösenordsinloggning för en viss användare, medan alla andra måste använda nycklar.

Det finns exempel på den typen av inställning i ”sshd_config”.

# Laboration 2

Se instruktion i portalen!



NACKADEMIN

# Inlämningsuppgift 1

För inlämningsuppgiften behöver ni känna till ett nytt verktyg: ”sha256sum”, som beräknar SHA 256, en kryptologisk hash, på en fil eller ström med data.

Detta används för att verifiera att en fil inte ändrats efter att hashvärdet beräknats.

# Summering

**Mål: Att förstå och använda ssh, generera och installera nycklar.**

Vi har pratat om:

- ssh och hur det kan användas på olika sätt
- Hur man installerar och konfigurerar OpenSSH - server
- Nyckelpar och inloggning utan lösenord
- Tunnlar

# Nästa gång

**Mål: Att kunna installera paket och hålla systemet uppdaterat.**

- Pakethantering
- Beroenden
- Installera nfs / samba
- curl
- Kontrollsummor / hashvärden
- Installera från källkod

# Stort tack!

NACKADEMIN