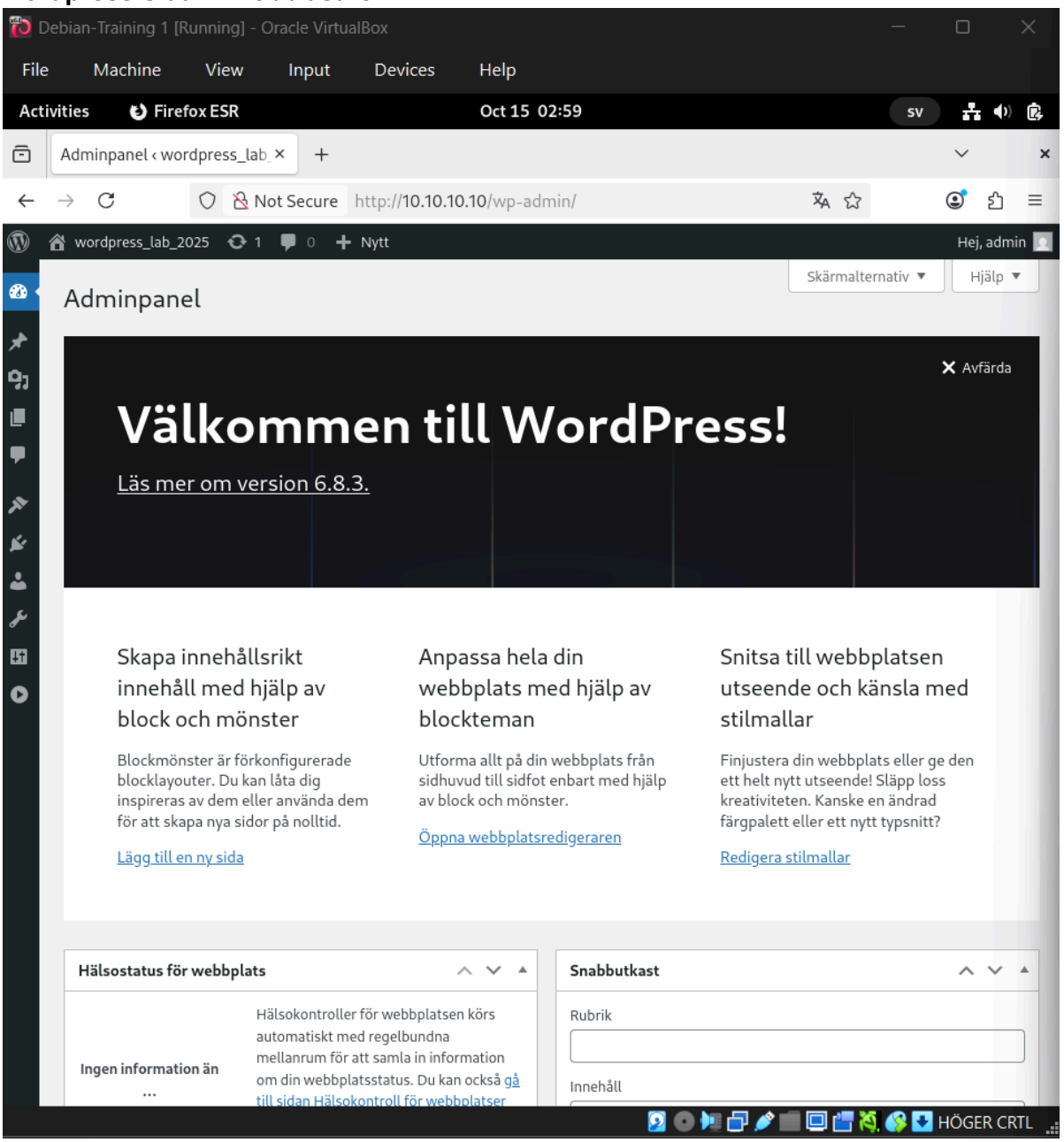
Ping mellan VM:

Wordpress-sidan i webbläsaren:



Brandväggsstatus:

```
root@wp-web:/home/webuser# sudo ufw status numbered
Status: active

     To                         Action      From
     --                         ------      ----
[ 1] 22/tcp                     ALLOW IN    Anywhere
[ 2] 80/tcp                     ALLOW IN    Anywhere
[ 3] 22/tcp (v6)                ALLOW IN    Anywhere (v6)
[ 4] 80/tcp (v6)                ALLOW IN    Anywhere (v6)
```

```
root@wp-db:/home/dbuser# sudo ufw status numbered
Status: active

     To                         Action      From
     --                         ------      ----
[ 1] 3306/tcp                   ALLOW IN    10.10.10.10
[ 2] 22/tcp                     ALLOW IN    Anywhere
[ 3] 22/tcp (v6)                ALLOW IN    Anywhere (v6)
```

Allt nedanför pysslade jag med i klassrummet när jag inte lyckades få mina ubuntu VMs att starta korrekt 🙂

VG-uppgiften:

- **UFW Brandvägg:** Nätverkssegmentering uppnåddes genom att tillåta nödvändig trafik enligt principen om least privilege.
- **Fail2ban**: Genom att installera fail2ban kan man skydda mot enklare brute force-attacker. Fail2ban fungerar genom att automatiskt lägga till regler i brandväggen (ufw/iptables) som blockerar IP-adressen temporärt. Applikationen noterar misslyckade inloggningsförsök från samma IP-adress inom en viss tid i en intern logg. Om antalet försök når 5 avvisas nya försök från IP-adressen i 10 minuter. Efter 10 minuter tas regeln bort automatiskt. På detta vis låter systemet inte denna användare/IP-adress sluka för mycket resurser av servern.
    - Installation:
      sudo apt install fail2ban -y
    - Verifiering:
      sudo fail2ban-client status sshd

Riskanalys:

1. **Distribuerade Brute Force-attacker**: Problemet med Fail2ban är att det är en åtgärd på network layer som saktar ner en angripare men inte helt stoppar sofistikerade attacker. En angripare med resurser kan använda distribuerade botnets med tusentals olika IP-adresser vilket gör att varje IP bara behöver göra 1-2 försök. Fail2ban blir då ineffektivt. En striktare åtgärd vore att införa rate limit med hårdare regler exempelvis så att 3 misslyckade försök ‘bannar’ IP-adressen permanent och kräver en manuell återställning av en systemadmin. Införandet av MFA funkar också bra liksom OAuth för att slippa lösenord men tror att en rate limit är ett enklare fix att börja med.
2. **Överdrivna databasbehörigheter**: Inställningen GRANT ALL PRIVILEGES är en risk då en enskild och vanlig databasanvändare har obegränsad tillgång till

databasen. Skulle en angripare komma över inloggningsuppgifterna är hela databasen exponerad eftersom användaren kan läsa, modifiera, radera, och ändra databasstrukturen osv. Det borde vi ändra på med hänsyn till principen om least privilege. WordPress behöver endast SELECT, INSERT, UPDATE och DELETE för vardaglig drift. För systemadministrativa uppgifter är det klokare att skapa en separat superanvändare med högre behörighet som endast används vid underhåll eller vid behov. Denna användare ska bara logga in lokalt på databasservern och inte över nätverket för att isolera anslutningen och därmed minimera sårbarheten. Tydlig policy att denna användare sällan använts eller inte tillkännages utanför IT-avdelningen är ett plus. På så vis avgränsas skadan och ev utredning underlättas vid olaga intrång.

3. **Okrypterad data at rest**: Om en angripare får fysisk åtkomst till servern, hårddiskarna, eller VM-filerna kan de läsa all data direkt från disken utan att behöva logga in i systemet. Därför behöver disken krypteras på disk-nivå.
4. **Förutsägbara användarnamn**: I databasen skapade vi "wpuser" som är en risk eftersom namnet låter som ett standardnamn som en angripare eller något program kan gissa sig till snabbt. Kombinerat med standardport (3306) och vanliga lösenord blir systemet sårbart för automatiserade attacker. Använd säkrare och/eller något mer random användarnamn som inte avslöjar systemets funktion. Även lösenordet kan behöva ändras kanske med en password manager då man initialt ofta får ett enkelt lösenord av sin databasadmin.
5. **Dataförlust**: Utan backups riskerar man permanent dataförlust vid ransomware-attack, hårdvarufel, mänskliga fel, programfel eller utsatta databas. Med jämna mellanrum bör det tas backups ifall det uppstår Indicators of Compromise/Attack eller om databasen bör underhållas. 3-2-1-regeln innebär 3 kopior av data, 2 olika media, och 1 kopia offsite vilket är en utmärkt åtgärd. Vid manuell handläggning: mysql -u wpuser -p wordpress > /tmp/ny_backup.sql && echo “backup lyckad”
   (Det vore bra att automatisera och schemalägga backups med mysqldump via cron)

Varför Två VM Istället för En?

- **Säkerhet** om webbservern röjs via WordPress är databasen fortfarande skyddad bakom nätverkssegmentering och brandvägg. Tänk defense in depth.
- **Skalbarhet**: Kan lägga till fler webbservrar som alla ansluter till samma databas när trafiken ökar. Tänk load balancing och SAN.
- **Underhåll**: Kan lättare underhålla en server utan att påverka den andra.
- **Enklare att ta backup** av endast databasen och återskapa webbservern snabbt om något går fel.
- Tänker att **man gör så här i praktiken** dvs separera för säkerhet, tillförlitlighet och compliance.