

Vem vill inte spela lite Minecraft?

Innehållsförteckning

1. Implementera tjänsteisolering och minimera Root-användning.....	1
1.1 Konfigurera Systemd tjänst.....	2
1.2 Tjänstehantering med least privileges för Systemctl.....	2
2. Skit i säkerheten, polarna vill bara spela.....	4

Krav: Optimistisk uppsättning 20 Gb lagring, 4 Gb RAM, enhet med Debian 13, VM, Proxmox, statisk IP-adressering

Bakgrund: Denna dokumentation beskriver tjänstehantering från ett säkerhetsperspektiv för Linux-server. Oavsett tjänst, bör principen gälla andra tjänster som driftsätts i en företagsmiljö. Majoriteten av systemkonfigurationen ska utföras som root.

1. Implementera tjänsteisolering och minimera Root-användning

Utgå från root

Skapa en separat användare där tjänsten ska administreras från
> adduser gamer

Skapa även en katalog under användaren
> mkdir /home/gamer/server/

Installera både java 21 och server-paketet i katalogen

> apt install openjdk-21-jre-headless -y
> wget

<https://piston-data.mojang.com/v1/objects/64bb6d763bed0a9f1d632ec347938594144943ed/server.jar>
-O server.jar

! hashen kan ändras, kopiera därför den gröna länken från den officiella hemsidan
→ <https://www.minecraft.net/en-us/download/server>

Exekvera

> java -Xms2G -Xmx2G -jar server.jar nogui

! Första gången tjänsten körs kommer den att krascha pga. eula
> nano eula.txt → eula=true

! All konfiguration har utförts av root vilket gör att root även blir user- och group owner i användarens katalog. Det behöver ändras eftersom tanken är att användaren "gamer" ska kunna hantera tjänsten.

Stå i användarens katalog och skriv in

> chown -R gamer:gamer server/

Logga ut från root och logga in som gamer med

```
> su -l gamer
```

Testa att starta och stoppa tjänsten

```
> java -Xms2G -Xmx2G -jar server.jar nogui
```

```
> stop
```

1.1 Konfigurera Systemd tjänst

Logga in som root igen, det är nu dags att skapa en riktig systemd tjänst.

```
> nano /etc/systemd/system/minecraft.service
```

```
[Unit]
Description=Minecraft Server
After=network.target

[Service]
User=gamer
Group=gamer
WorkingDirectory=/home/gamer/server
ExecStart=/usr/bin/java -Xms2G -Xmx2G -jar server.jar nogui
Restart=on-failure

[Install]
WantedBy=multi-user.target
```

```
> systemctl daemon-reload
```

```
> systemctl enable minecraft.service
```

```
> systemctl start minecraft.service
```

```
> systemctl status minecraft.service
```

Nu när tjänsten körs kan det vara bra att verifiera om en användare faktiskt kan ansluta till Minecraft servern.

```
> journalctl -fu minecraft.service // visar tjänstens logg i realtid, ska även visa att en användare lyckas ansluta
```

1.2 Tjänstehantering med least privileges för Systemctl

Systemctl kommandon förutom “status” är begränsat till sudo användare, därför behöver “gamer” läggas till i sudoers filen för att utöka Systemctl kommandon som endast avser den specifika tjänsten.

```
> nano /etc/sudoers.d/gamer
```

```
gamer ALL=(ALL) NOPASSWD: /bin/systemctl start minecraft.service, /bin/systemctl stop
minecraft.service, /bin/systemctl restart minecraft.service, /bin/systemctl status minecraft.service,
/bin/systemctl enable minecraft.service, /bin/systemctl disable minecraft.service
```

Verifiera och applicera ändringar med

```
> sudo visudo -cf /etc/sudoers.d/gamer  
→ du ska få tillbaka "parsed OK"
```

Logga in som användaren och verifiera med

```
> sudo -l
```

Gå in i katalogen server och verifiera med

```
> sudo systemctl status minecraft.service  
> sudo systemctl stop minecraft.service  
> sudo systemctl start minecraft.service
```

! sudo systemctl stop ssh ... eller andra tjänster på servern ska inte kunna påverkas // det ska inte fungera eftersom rättigheter med sudo är begränsade!

! Det går att lägga till alias för behöriga sudo kommandon som används för att hantera tjänsten genom /home/gamer/.bashrc. Alias är korta smeknamn för långa repetitiva kommandon, det gör livet lättare för en administratör.

```
alias mcstatus='sudo systemctl status minecraft.service'  
alias mcstop='sudo systemctl stop minecraft.service'  
alias mcstart='sudo systemctl start minecraft.service'  
alias mcrestart='sudo systemctl restart minecraft.service'  
alias mcenable='sudo systemctl enable minecraft.service'  
alias mcdisable='sudo systemctl disable minecraft.service'
```

```
> source .bashrc // laddar om filen
```

! Om andra ska ansluta är det bara att konfigurera port forwarding i routern

Det ska vara allt, bra jobbat!

2. Skit i säkerheten, polarna vill bara spela

Det absolut enklaste sättet att skapa en minecraft server för LAN

Logga in som root och skapa katalog /home/ANVÄNDARE/minecraft

I katalogen

```
> apt install openjdk-21-jre-headless -y
```

```
> wget
```

```
https://piston-data.mojang.com/v1/objects/64bb6d763bed0a9f1d632ec347938594144943ed/server.jar  
-O server.jar
```

! hashen kan ändras, kopiera därför den gröna länken från den officiella hemsidan

→ <https://www.minecraft.net/en-us/download/server>

Exekvera

```
> java -Xms2G -Xmx2G -jar server.jar nogui
```

nano eula.txt → eula=true

Starta om servern igen med “java -Xms2G -Xmx2G -jar server.jar nogui” och anslut med spelklient

Skriv stop för att stänga av tjänsten

! Om andra ska ansluta är det bara att konfigurera port forwarding i routern

Minecraft är ett axplock av tjänster som kan köras i en server, men uppsättningen som beskrivs ovanför är inte alls säker. Varför då?

1 - tjänsten körs som root

2 - ingen isolering

3 - ingen begränsning av användarprivilegier

4 - ingen riktig tjänst