Provision & Setup Ubuntu Server 22.04

>prøjectsecurity_

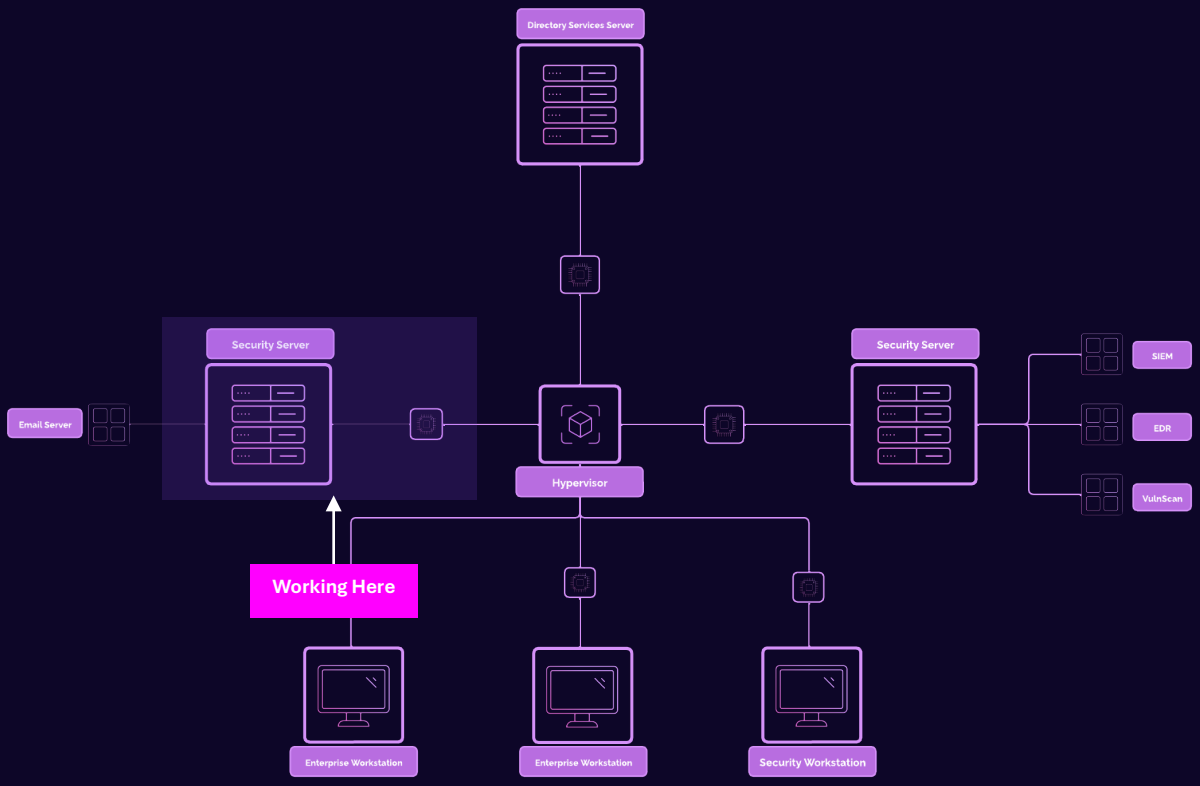
>prøjectsecurity_

Table of Contents

Prerequisites

1. Virtualbox installed.
2. Virtual Machine with Ubuntu 22.04 ISO Server has been configured and provisioned (the ISO should be attached to the new VM).
3. Windows Server 2025 with Active Directory Domain Services (ADDS) configured.

Network Topology



Email Security Server Overview

Overview

An **email server** is a system designed to send, receive, store, and manage email communication for users. It uses protocols such as SMTP (Simple Mail Transfer Protocol) for sending emails, and IMAP (Internet Message Access Protocol) or POP3 (Post Office Protocol) for receiving and managing email messages.

We will be configuring Postfix as a Mail Transfer Agent (MTA), which is used for sending and routing emails on Linux servers.

Email servers are much less common today than they were 20+ years ago. Running an email server requires expertise with configuring DNS records, securing against spam, developing a good reputation via IP address for email delivery, and more. With the emergence of third-party application such as Gmail, Microsoft 365, and ProtonMail (for personal use), these managed services provide scalable, secure email without having to manage the infrastructure of an email server.

Why are we configuring an email server in this project then?

It provides good insight into how email works and why it's important to secure your email gateways.

This guide will be used to set up the underlying Operating System, Ubuntu 22.04 Server LTS, and connection through Active Directory. Additional guides will be provided for configuring Postfix.

Security Implications

While running an email server like Postfix on Ubuntu Server 22.04 gives you control, it also introduces several security considerations:

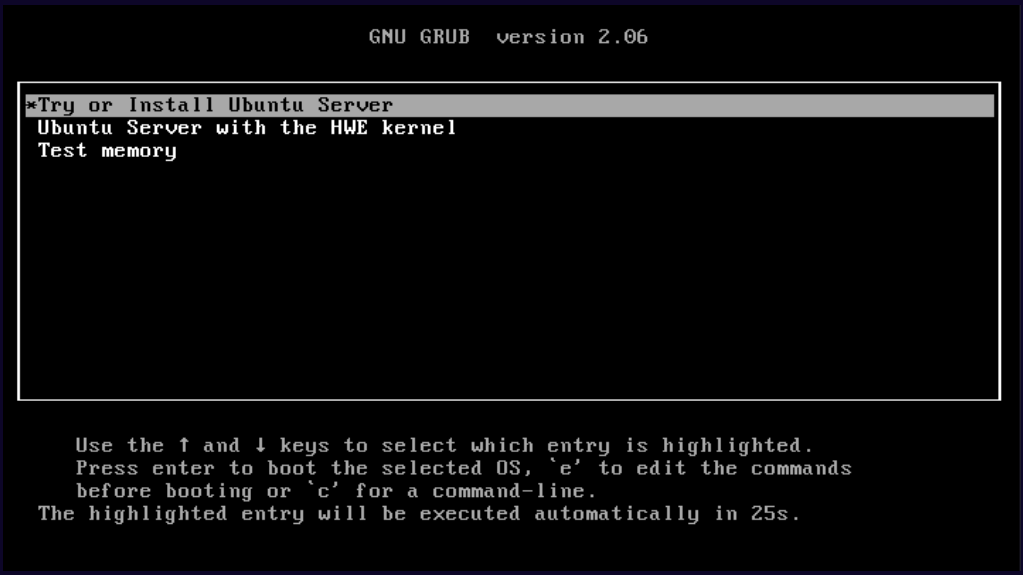
Common Threats

1. **Open Relay Exploitation**: If improperly configured, your server can be used by spammers to send large volumes of email, damaging your IP reputation.
2. **Brute Force Attacks**: Attackers often attempt to compromise accounts via brute force or credential stuffing.
3. **Spam and Phishing**: Attackers may spoof your domain or use your server for phishing campaigns.
4. **Data Breaches**: Poorly secured servers can expose sensitive emails and user credentials.
5. **Malware Delivery**: Your server could inadvertently become a vehicle for spreading malware if attachments are not scanned.

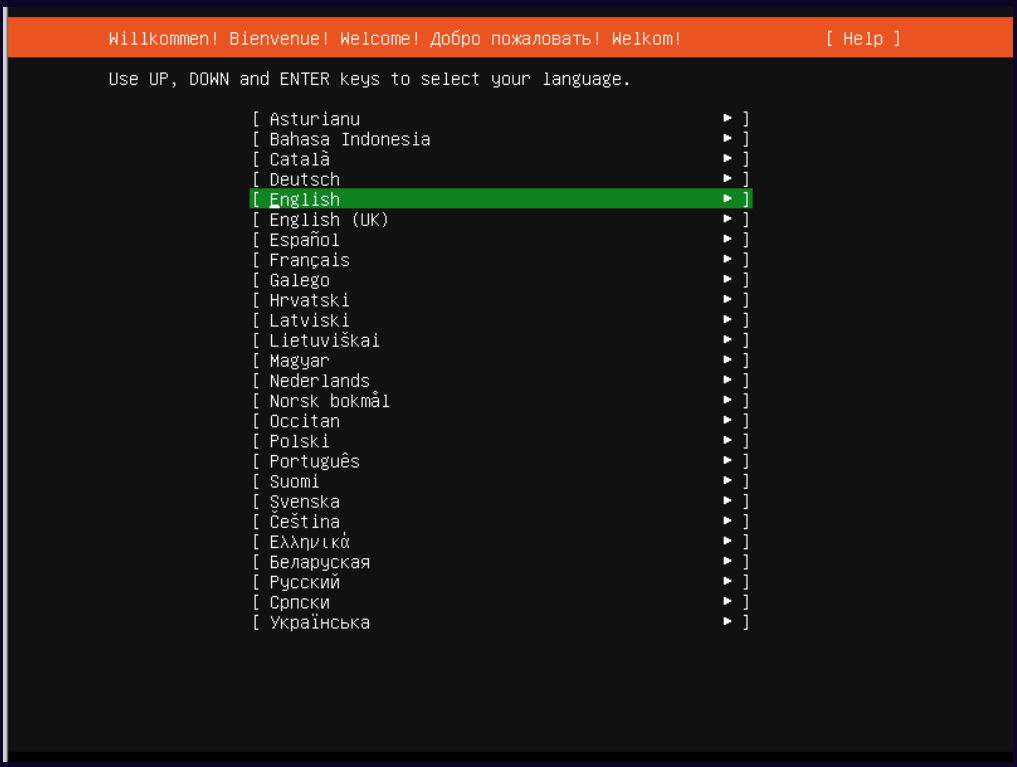
Setup Security Server

Step 1

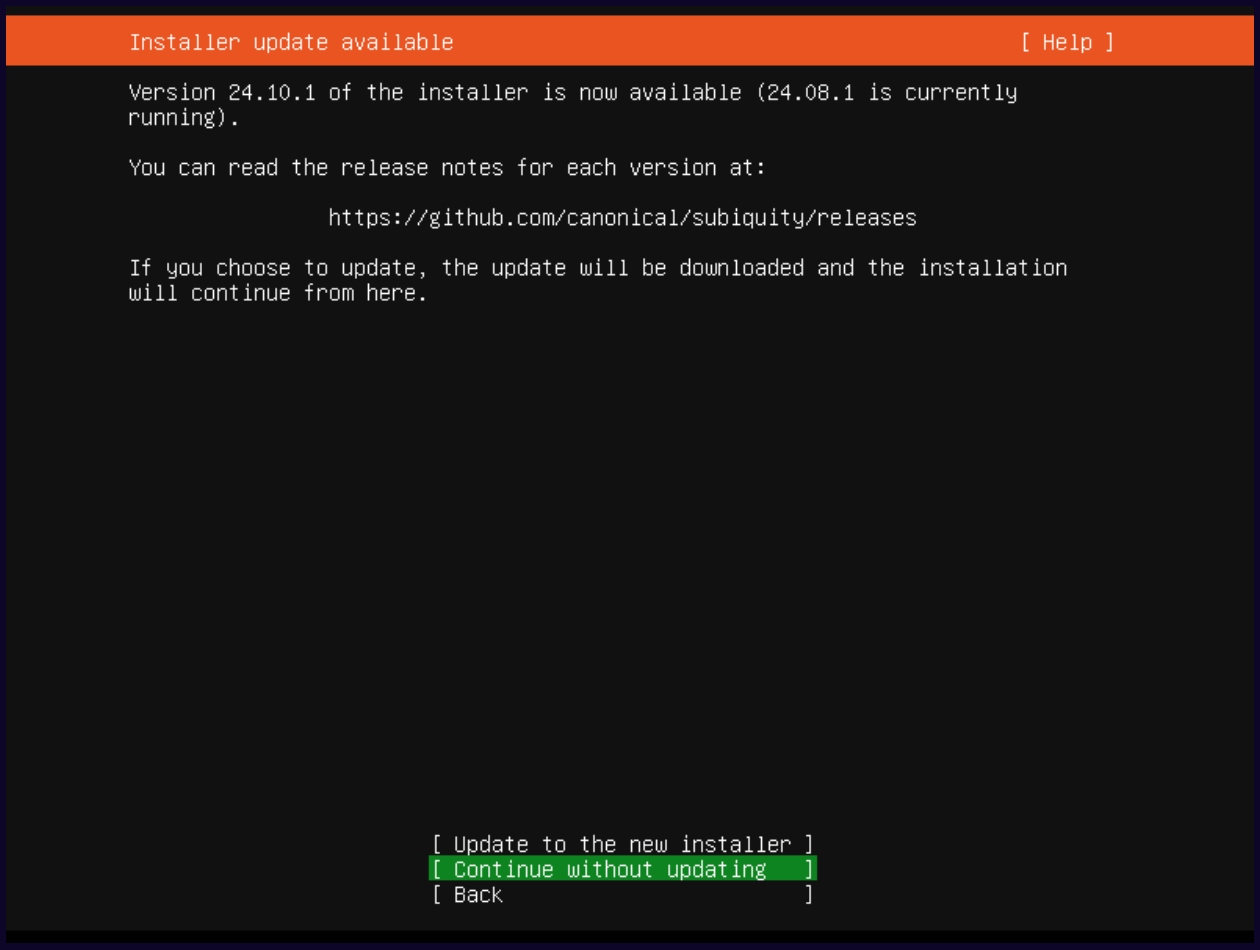
Press “Enter”.



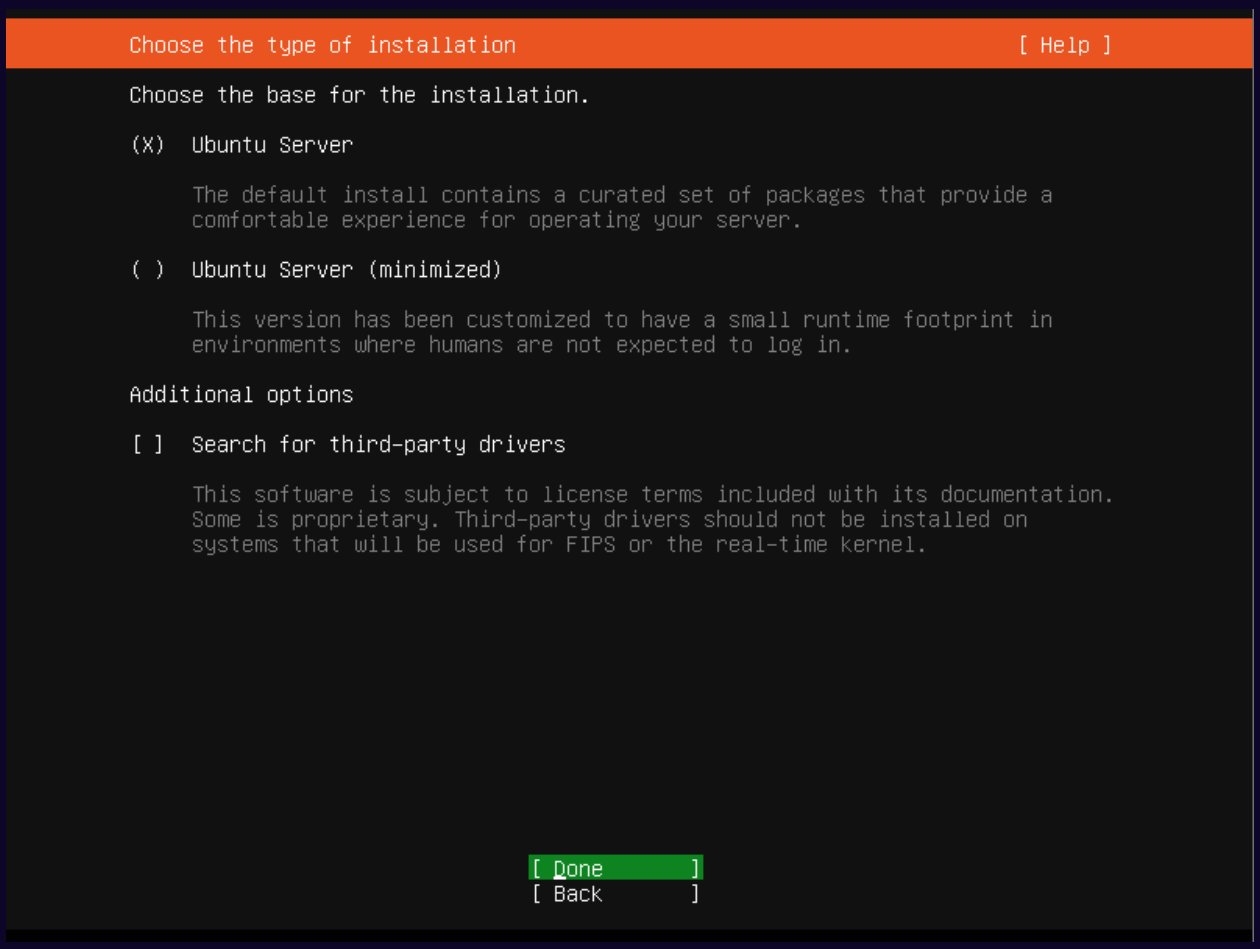
Select language.



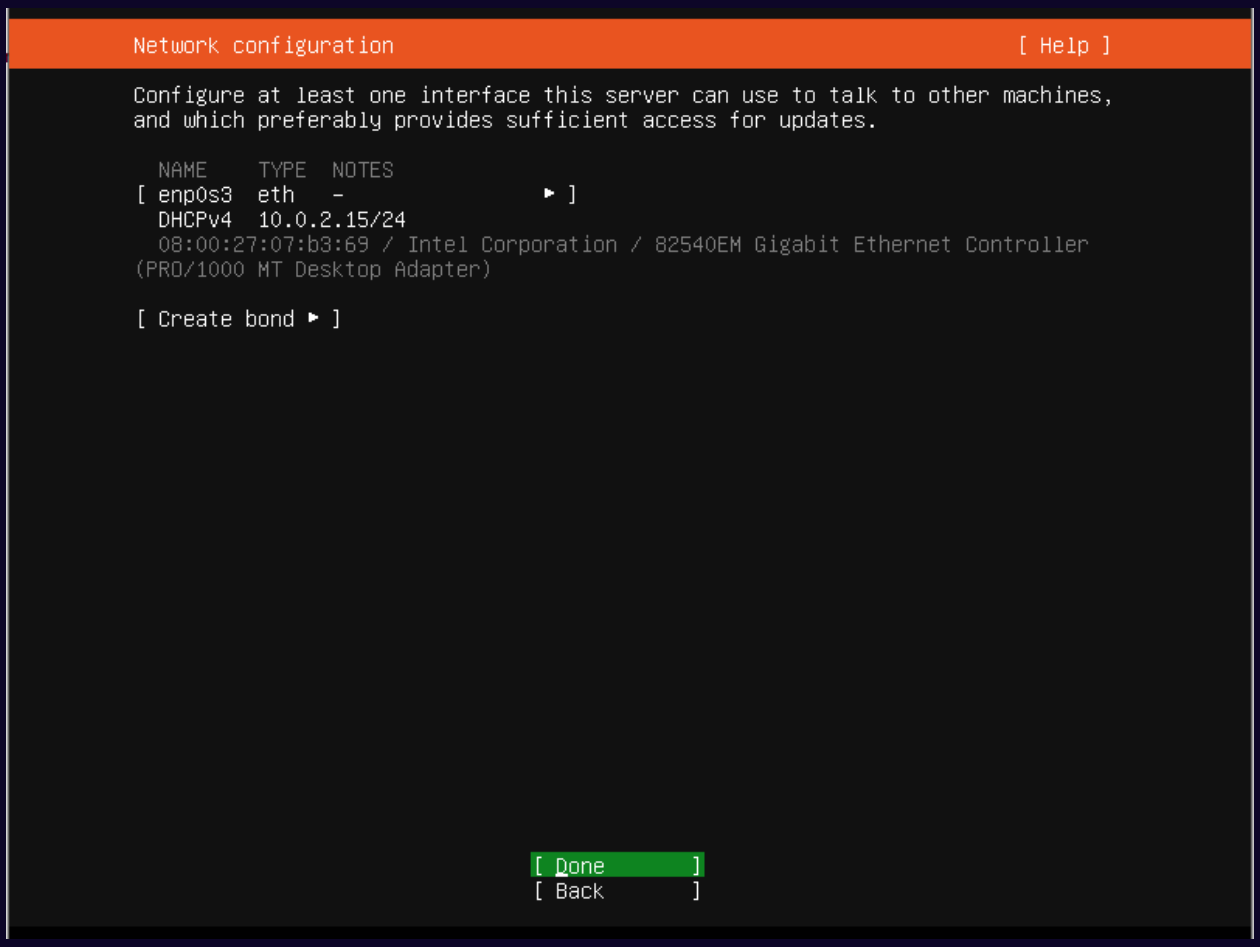
Continue without updating.



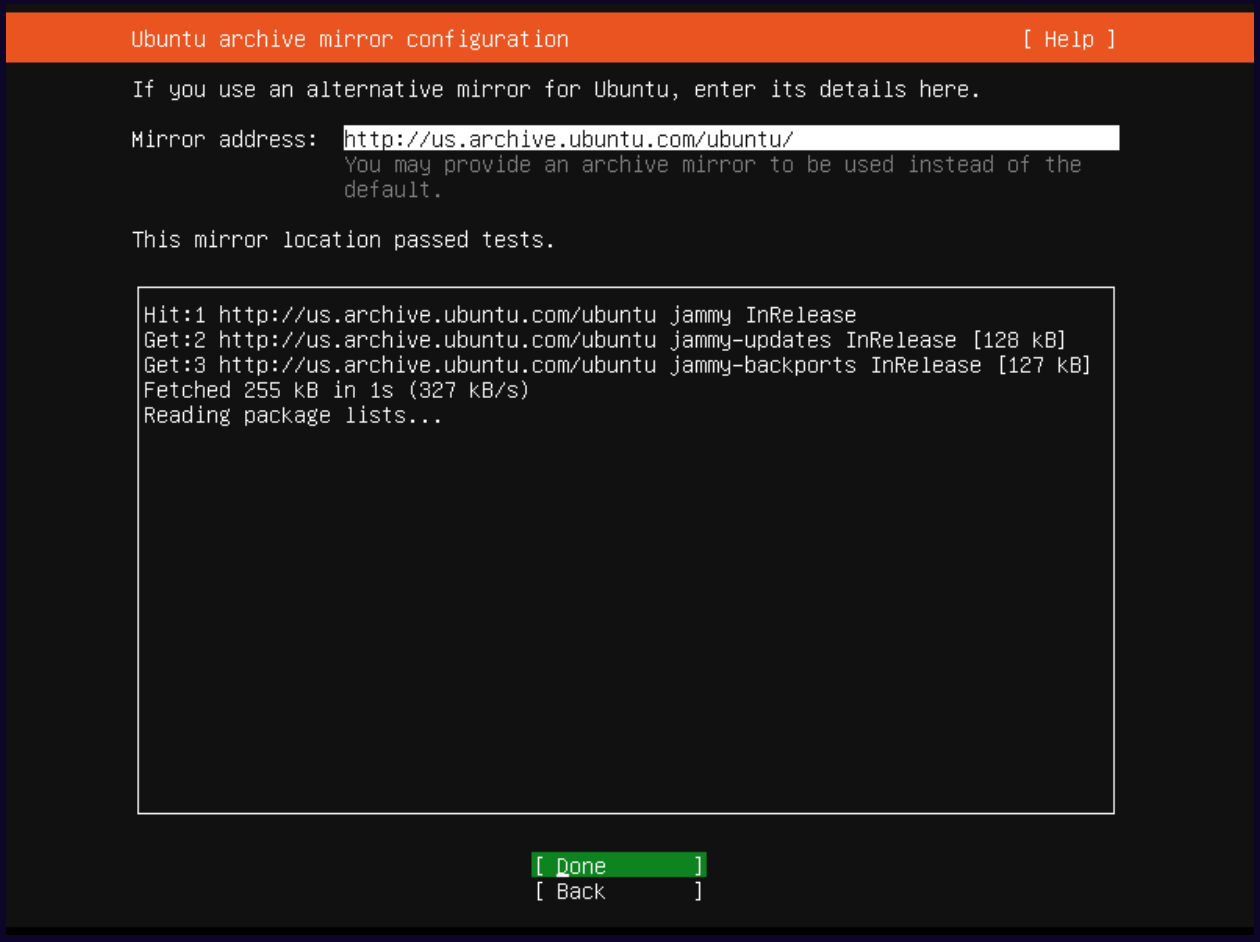
Select “Ubuntu Server”.



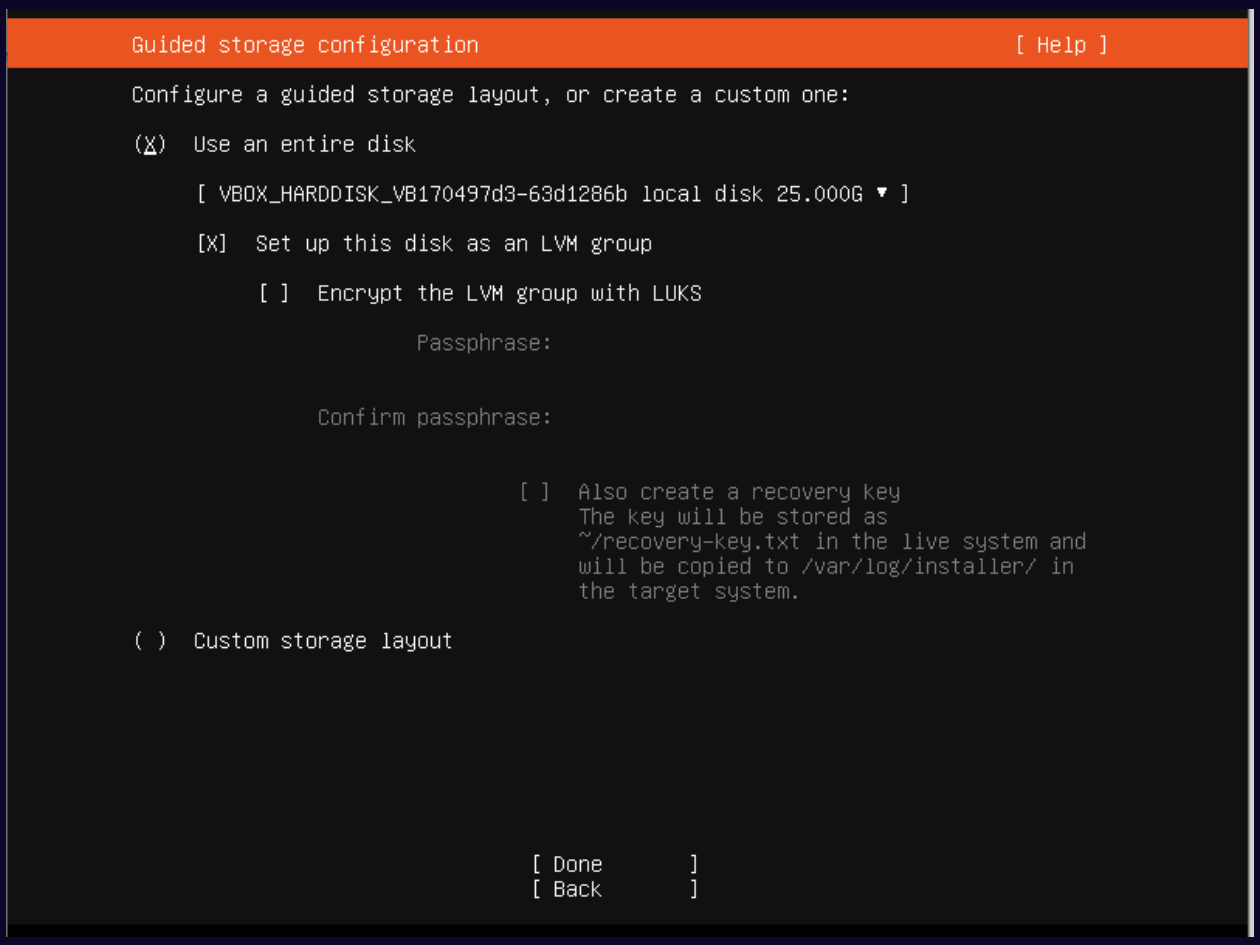
Leave default Network configuration → Leave Proxy Page empty.



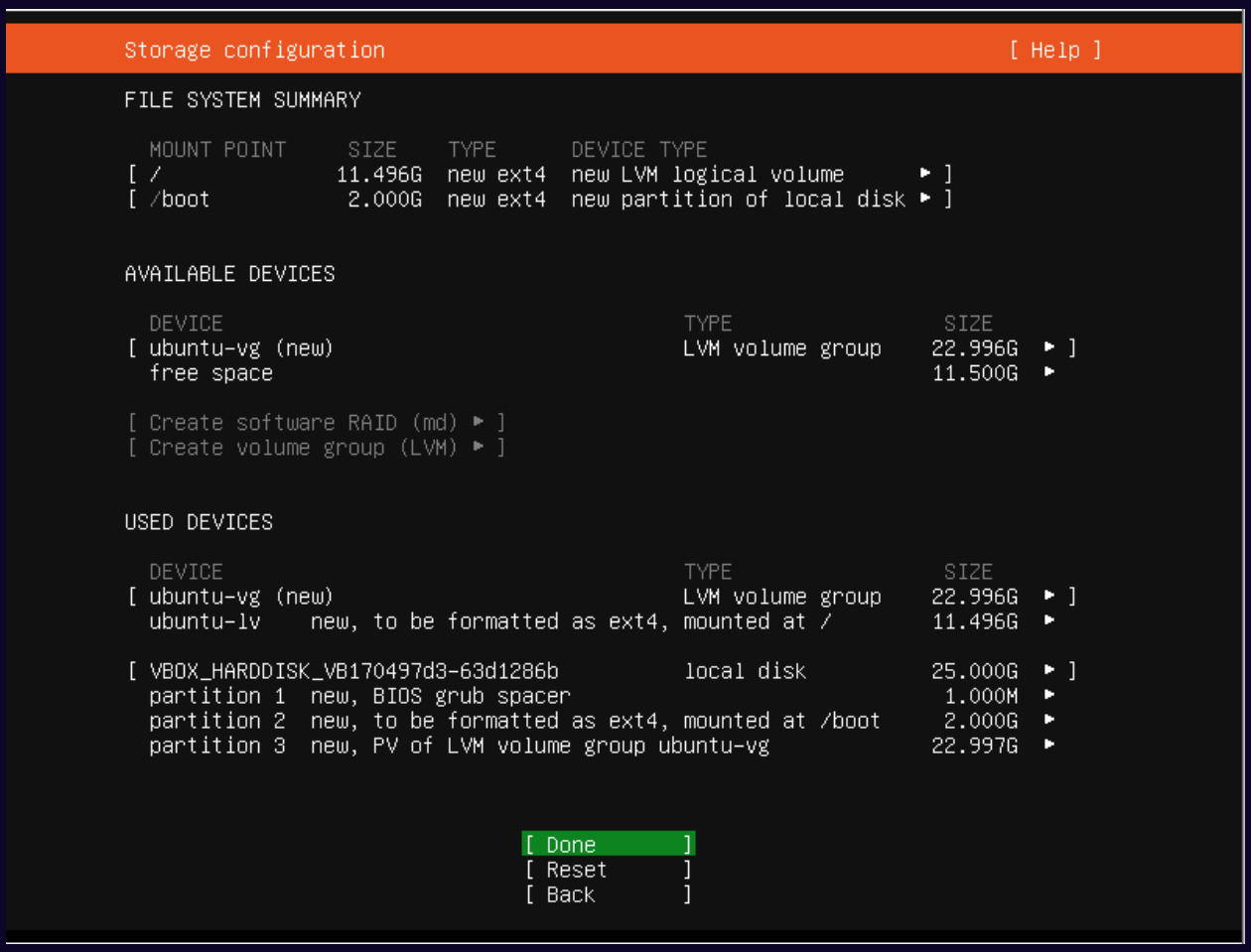
Leave “Mirror” configuration empty → “Done”



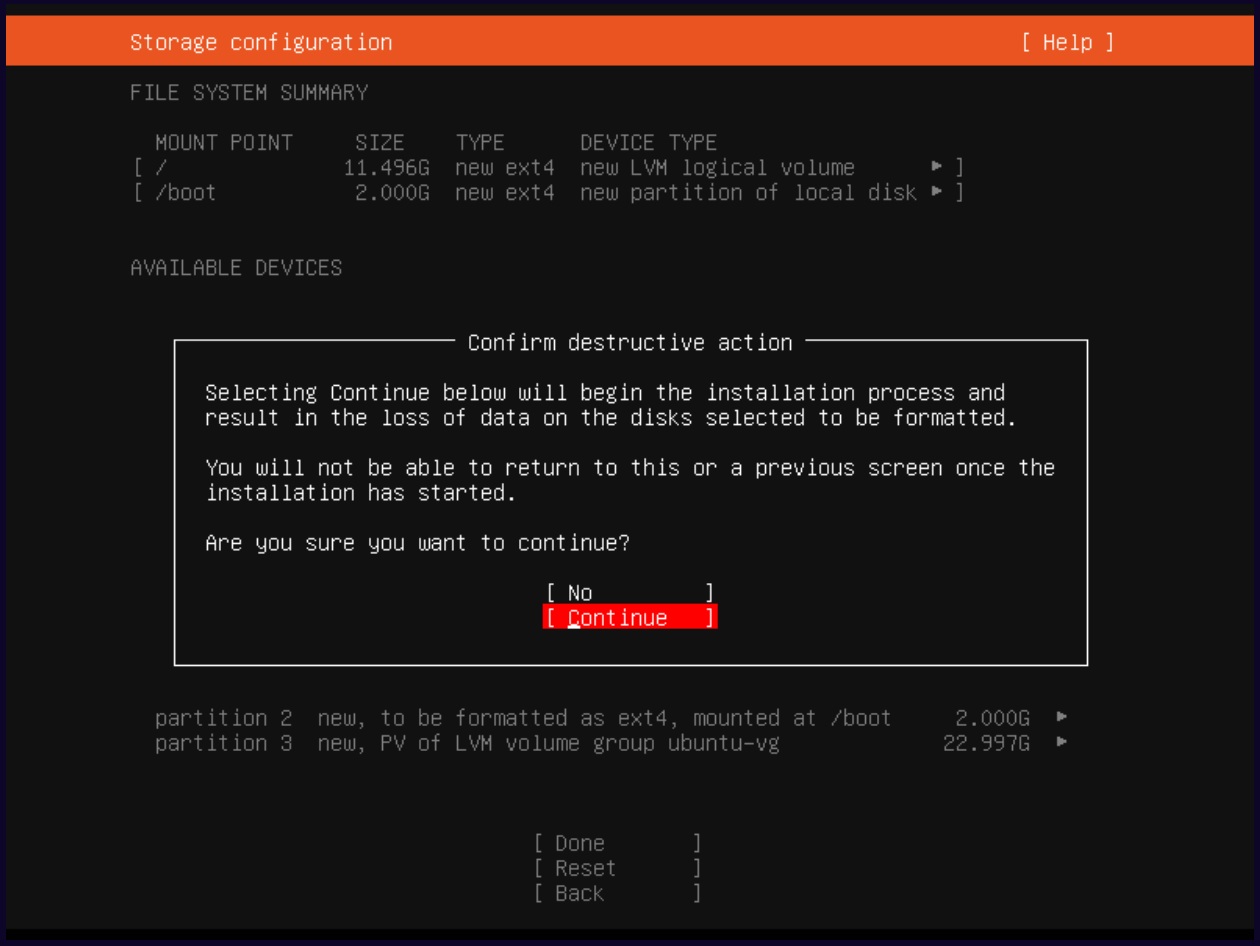
Select “Use an entire disk”. Press the “Tab” key until selecting “Done.”



Leave Storage Configuration as default.

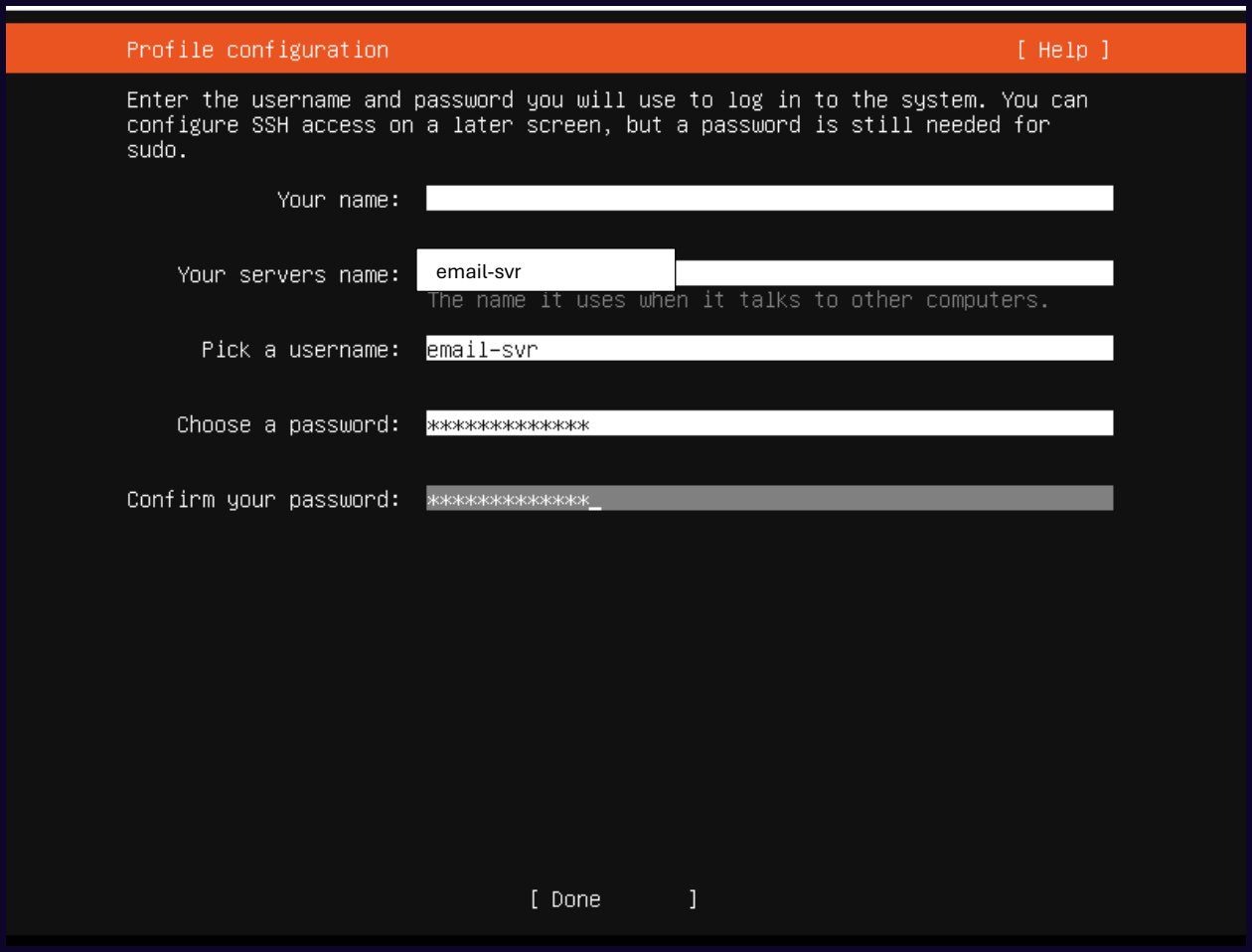


Arrow to “Continue”.

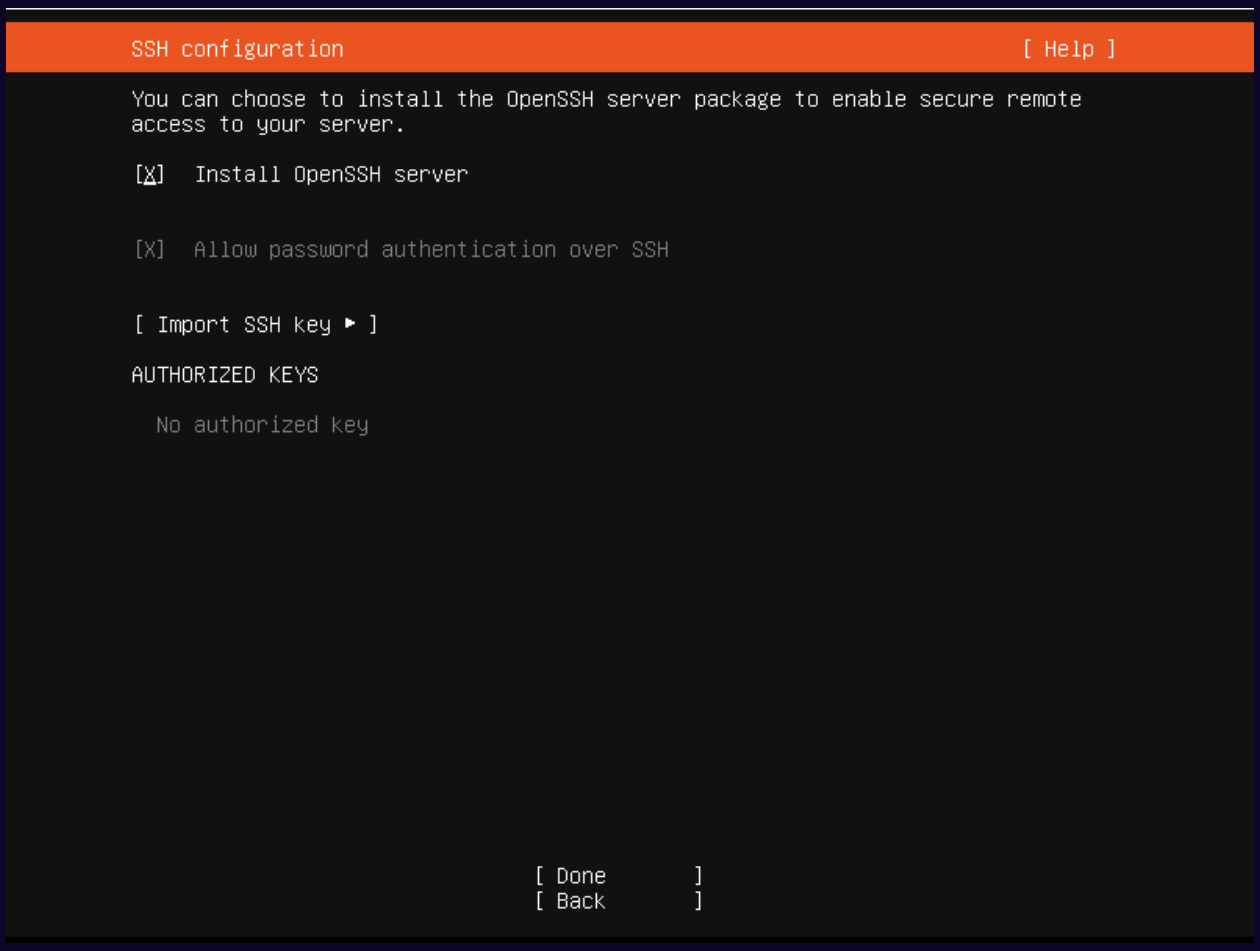


Enter in the server's hostname, username, and password.

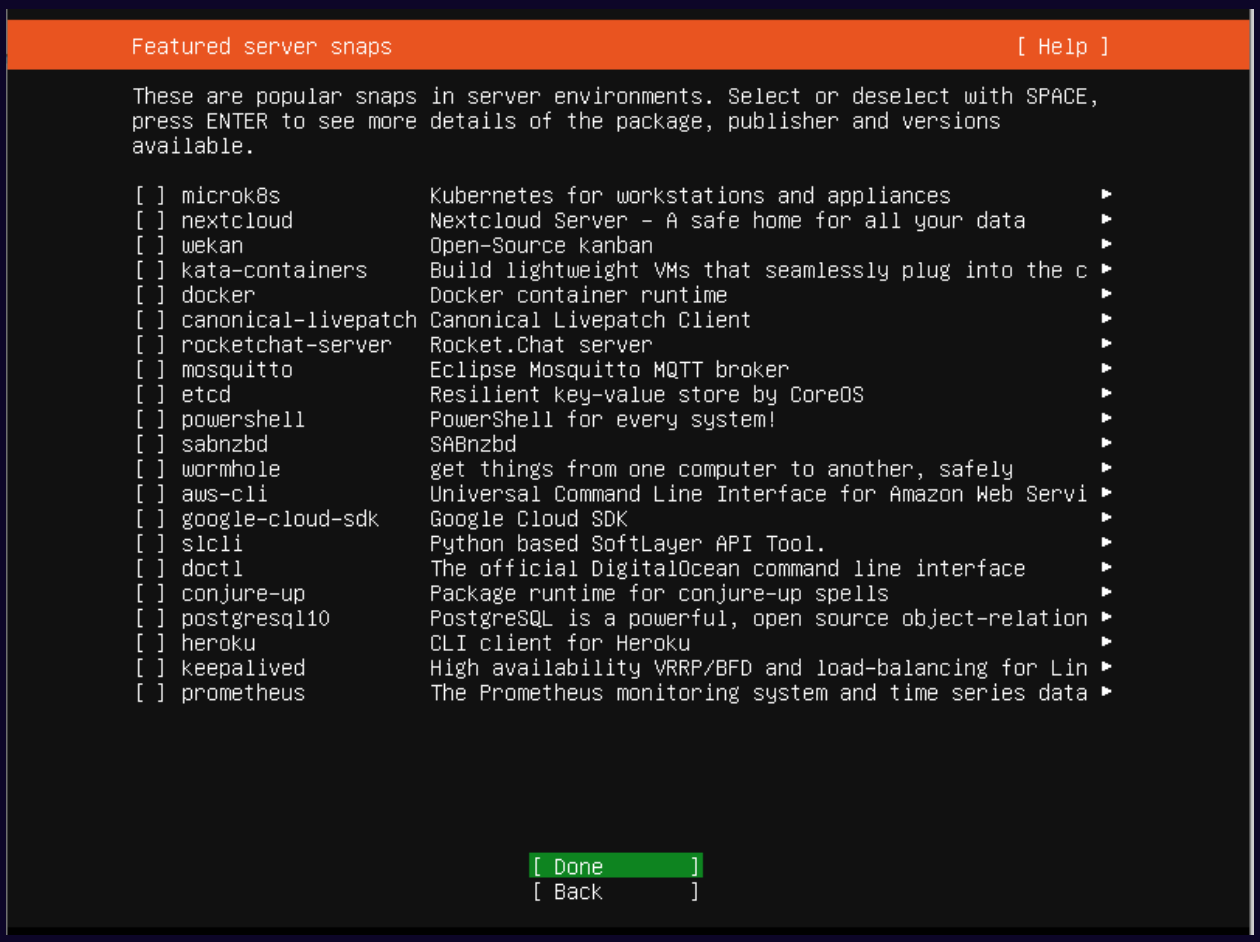
👉 Refer to the "**Project Overview**" guide for more information on default usernames and passwords.



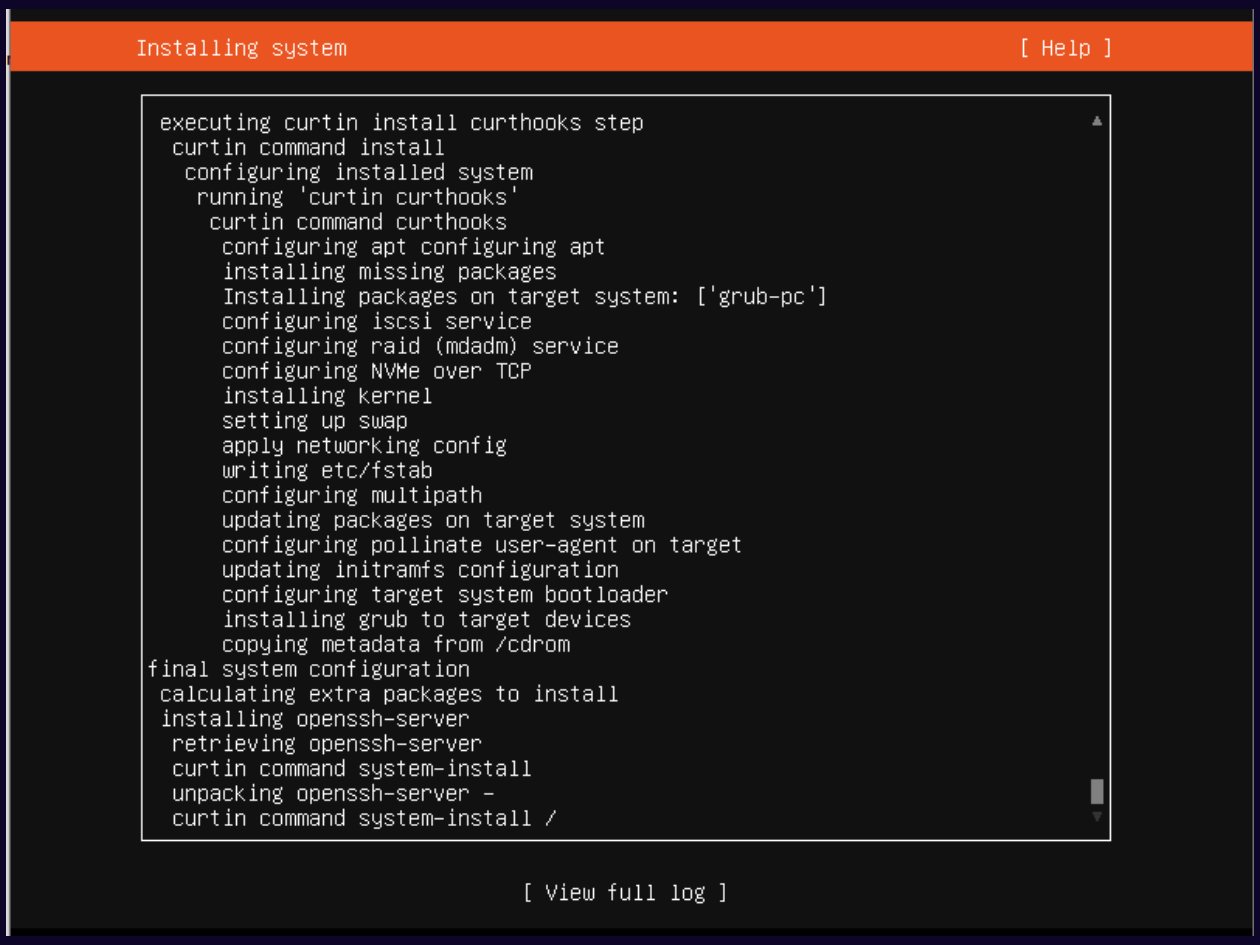
Select “Skip for now” for the Ubuntu Pro → Select “Install OpenSSH server”. Use the Tab key until selecting “Done”.



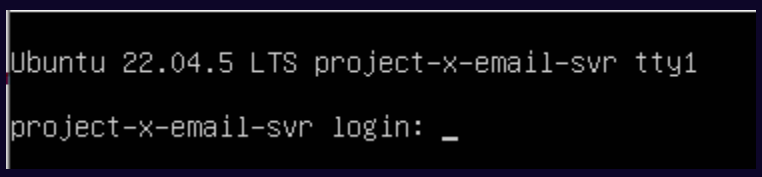
Use the Tab key until selecting “Done”.



Wait for the OS to install, press “enter for a reboot”.



Success!



Connect Ubuntu Desktop to Active Directory

👉 Switch the Network from “NAT Network” → Bridged.

👉 Refer to this guide if you would like to insert VirtualBox Guest Additions (for copy/paste controls).

Connecting Ubuntu (and Debian-based systems) to Active Directory can be accomplished in a couple ways. The easiest way is to connect Ubuntu to Active Directory with **realmd** and

SSSD (System Security Services Daemon). **Samba Winbind** can also be used to join Linux systems if realmd / SSSD is not working.

❗ Currently realmd and SSSD integration do not work for Windows Server 2025 and Debian/Ubuntu-based systems.

About SSSD / Realmd

- **System Security Services Daemon (SSSD):** A service on Linux systems that provides a central access point for identity management and authentication. When connecting a Linux system to Active Directory (AD), SSSD allows for the integration by acting as an intermediary between the Linux system and AD needing to know what files should be edited.
- **realmd**: A tool that simplifies the process of joining Linux machines to AD domains. It automates the discovery, configuration, and enrollment of Linux systems in Active Directory, making it easier to integrate Linux systems into existing AD environments. Realmd is especially useful for administrators because it manages the complexities of setting up Kerberos, configuring LDAP settings, and ensuring proper authentication protocols.
- **realmd** is a tool that automates domain joining and manages configurations for **sssd,** which provides caching, more flexible configuration options, and better performance.

About Samba Winbind

- **Samba Winbind**: A component of the Samba suite that allows Linux systems to authenticate users against Windows Active Directory (AD) and integrate with Windows network environments. Is a more direct integration, especially useful for legacy systems and environments where tight compatibility with Windows protocols is necessary. It's often preferred when working in older Windows Server environments or where native Samba compatibility is crucial.

Realmd + SSSD

Step 1

Open a new terminal session.

Update the system with:

```
sudo apt update
```

Step 2

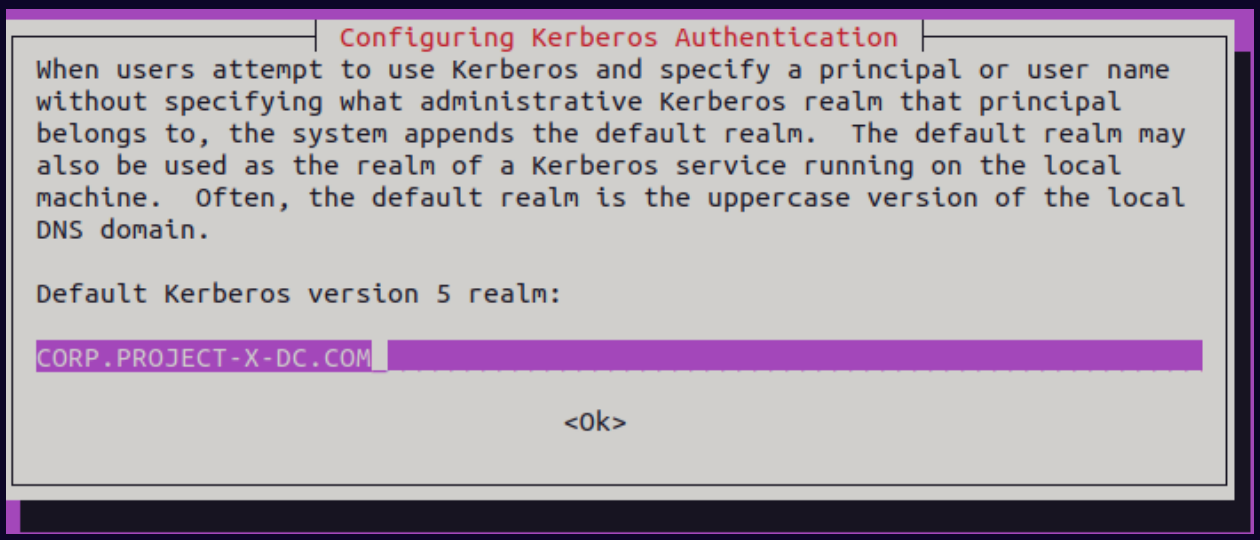
Adding the following under the [Time] block.

```
sudo nano /etc/systemd/timesyncd.conf
```

```
[Time]
NTP=corp.project-x-dc.com
#FallbackNTP=ntp.ubuntu.com
RootDistanceMaxSec=30
```

Install the necessary packages:

```
sudo apt install realmd sssd sssd-tools samba-common krb5-user
packagekit libnss-sss libpam-sss adcli samba-common-bin
```



Step 3

Use the realm command to discover the domain.

```
jane@linux-client:~$ sudo realm discover corp.project-x-dc.com
corp.project-x-dc.com
  type: kerberos
  realm-name: CORP.PROJECT-X-DC.COM
  domain-name: corp.project-x-dc.com
  configured: no
  server-software: active-directory
  client-software: sssd
  required-package: sssd-tools
  required-package: sssd
  required-package: libnss-sss
  required-package: libpam-sss
  required-package: adcli
  required-package: samba-common-bin
```

Step 4

Enter the following command, enter the Administrator password:

```
sudo realm join --verbose --user=Administrator corp.project-x-dc.com
```

Step 5

If no output is shown in the console, then the VM has been connected.

Enter the following command to confirm:

```
realm list
```

Samba Winbind

Step 1

Open a new terminal session.

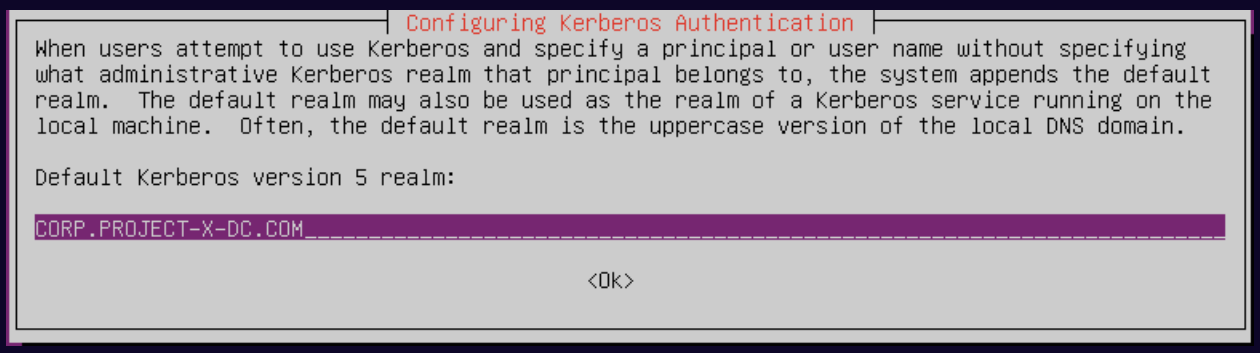
Update the system with:

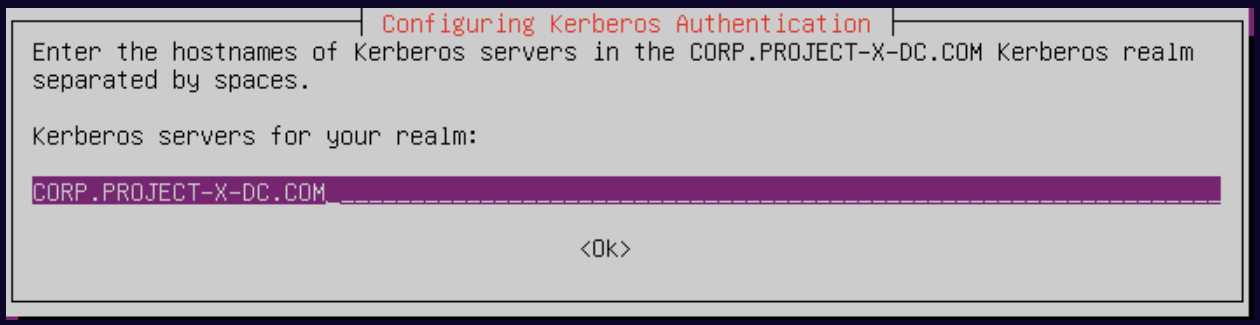
```
sudo apt update
```

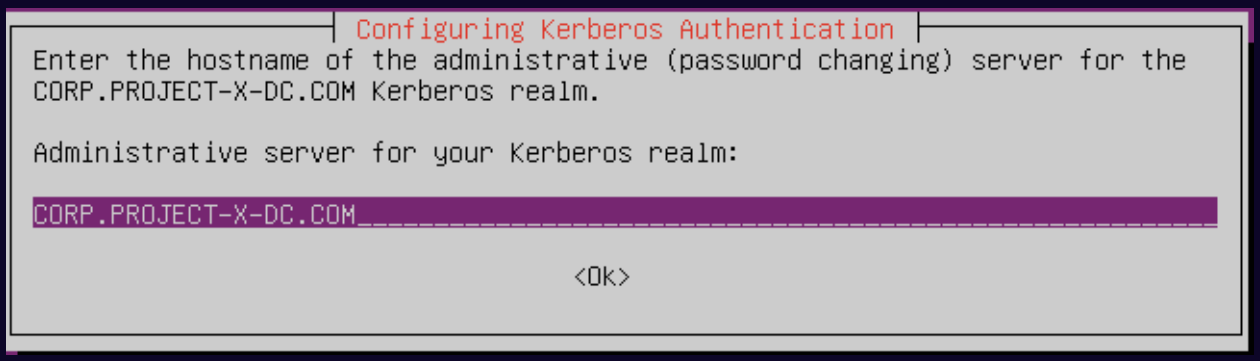
Install the necessary packages (note that **krb5-user** is installed this time):

```
sudo apt -y install winbind libpam-winbind libnss-winbind krb5-config krb5-user samba-dsdb-modules samba-vfs-modules
```

Add `CORP.PROJECT-X-DC.COM` for the two Kerberos Authentication pages.







Step 2

Move the smb.conf.org file:

```
mv /etc/samba/smb.conf /etc/samba/smb.conf.org
```

Step 3

```
sudo nano /etc/samba/smb.conf
```

Replace realm and workgroup with the following:

```
[global]
   kerberos method = secrets and keytab
   realm = CORP.PROJECT-X-DC.COM
   workgroup = CORP
```

```
security = ads

template shell = /bin/bash

winbind enum groups = Yes

winbind enum users = Yes

winbind separator = +

idmap config * : rangesize = 1000000

idmap config * : range = 1000000-19999999

idmap config * : backend = autorid
```

Step 4

Confirm `passwd` and `group` have `winbind` set as a value.

```
sudo nano /etc/nsswitch.conf
```

Add if needed.

```
passwd:         files systemd winbind
group:          files systemd winbind
shadow:         files
gshadow:        files
```

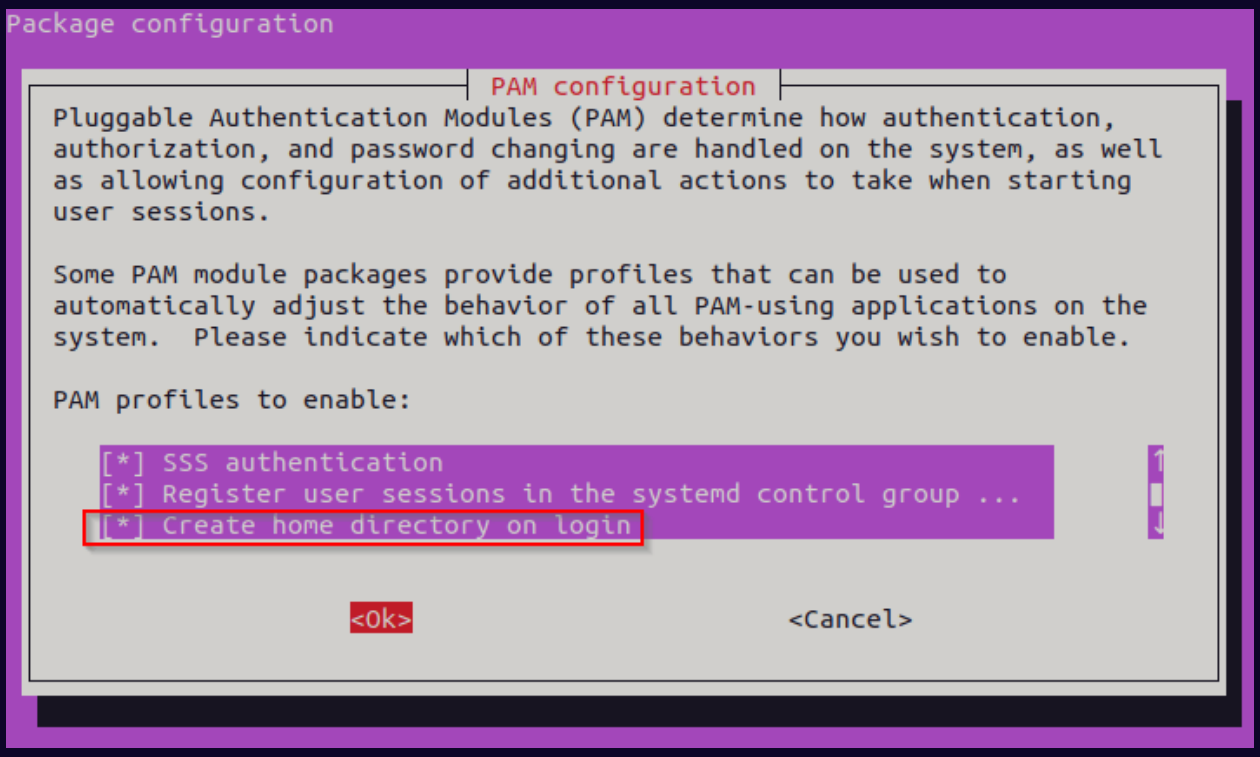
Step 5

On Ubuntu, every user that has an interactive logon to the system needs a home directory. For domain users, we need to set this before a user is able to successfully logon and start working.

Issue the following command

```
sudo pam-auth-update
```

Scroll down up to the point where it states:" Create *home directory on login*". Use the space bar to select, tab to "*OK*" and hit enter.



Kudos to Michael Waterman for the screenshot!

Step 6

Change DNS settings to refer to AD.

```
sudo nano /etc/resolv.conf
```

```
nameserver 10.0.0.5
nameserver 127.0.0.53
options edns0 trust-ad
search localdomain
```

Add `corp.project-x-dc.com` to `/etc/hosts`:

```
sudo nano /etc/hosts
```

```
  GNU nano 6.2                                  /etc/hosts
127.0.0.1 localhost
127.0.1.1 email-svr
10.0.0.5 corp.project-x-dc.com
# The following lines are desirable for IPv6 capable hosts
::1     ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
```

👉 Change Network settings from Bridged → NAT Network (project-x-network).

Step 7

Join the domain with Administrator:

```
sudo net ads join -U Administrator
```

Step 8

Restart winbind:

```
systemctl restart winbind
```

Step 9

Get Active Directory services information listing.

```
net ads info
```

Step 10

List all available users.

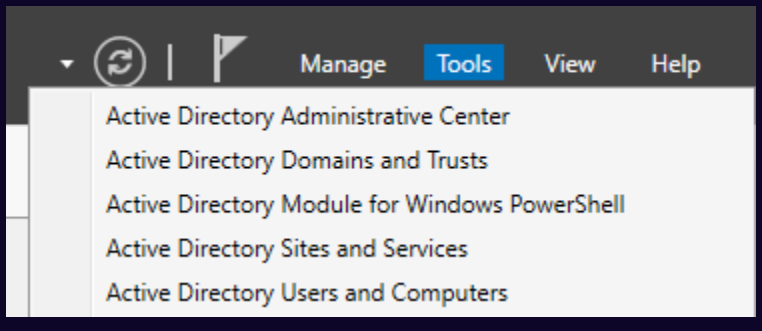
wbinfo -u

```
email-svr@email-svr:~$ wbinfo -u
CORP+administrator
CORP+guest
CORP+krbtgt
CORP+johnd
CORP+janed
CORP+secuser
```

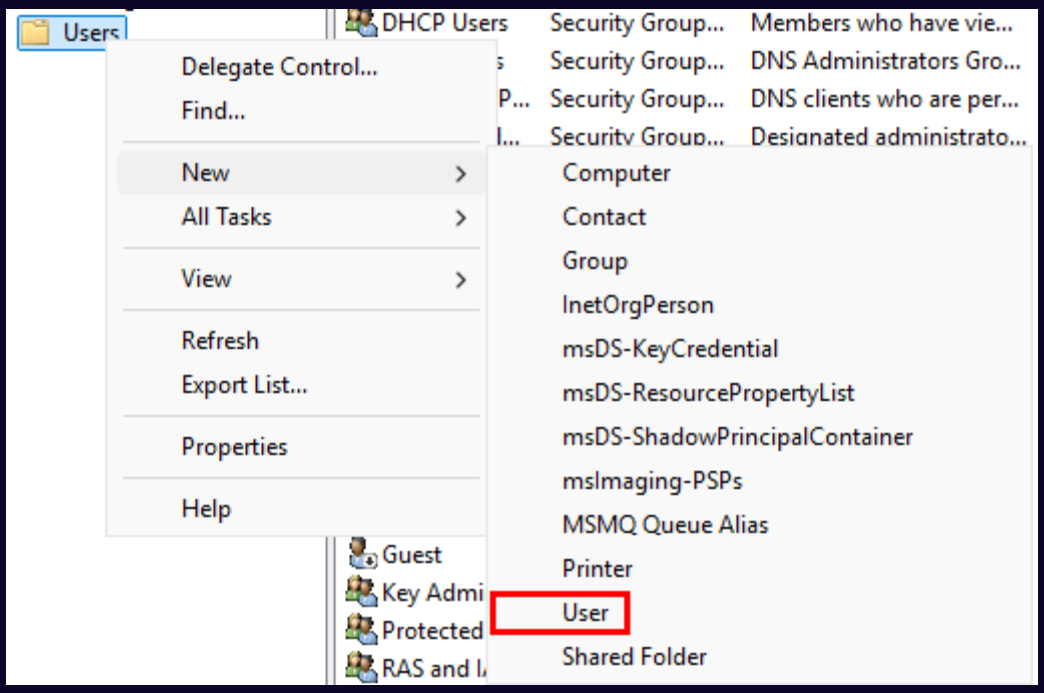
Step 11

Let's create `email-svr's` AD account in our Domain Controller.

Go to Server Manager, then on the top right "Tools" → "Active Directory Users and Computers"

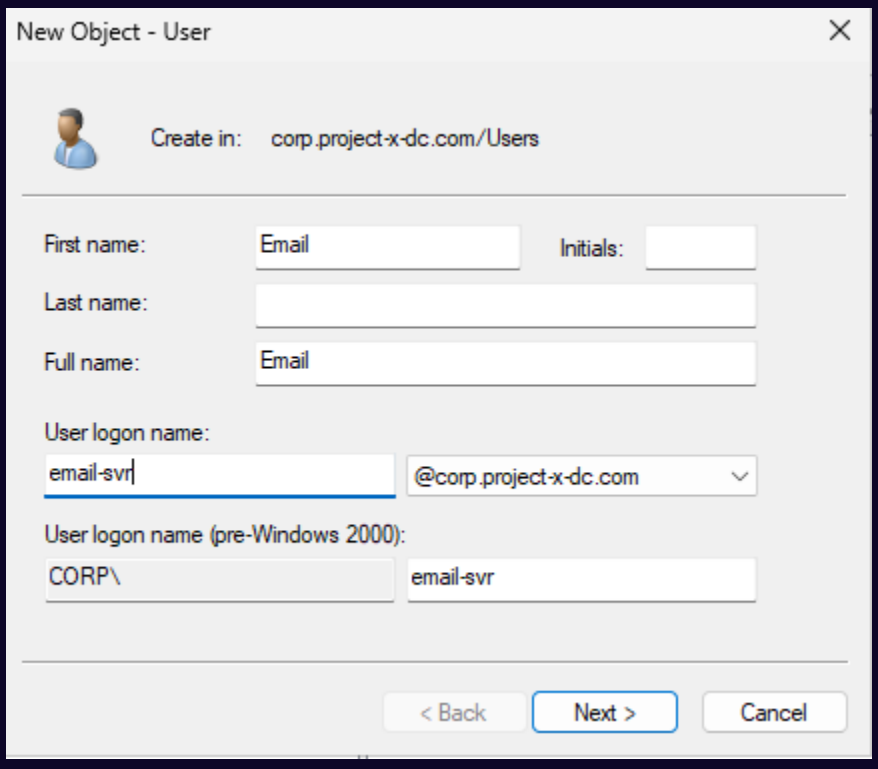


Navigate to the "Users" folder. Right-click, then go to "New" → "User"



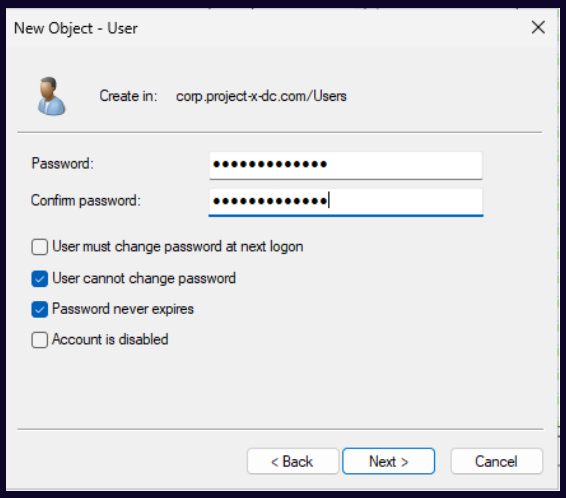
Add the following information.

`emailsvr@corp.project-x-dc.com`



Set `email-svr's` password (`@password123!`).

👉 Refer to the “Project Overview” guide for more information on default usernames and passwords.



Clear the winbind cache by restarting the service, then see the changes reflected.

```
sudo systemctl restart winbind
```

```
wbinfo -u
```

```
email-svr@email-svr:~$ wbinfo -u
CORP+administrator
CORP+guest
CORP+krbtgt
CORP+johnd
CORP+janed
CORP+secuser
CORP+email-svr
```

Step 12

Login as `email-svr (CORP+email-svr):`

```
sudo login
```

```
email-svr@email-svr:~$ sudo login
email-svr login: CORP+email-svr
```

Step 13

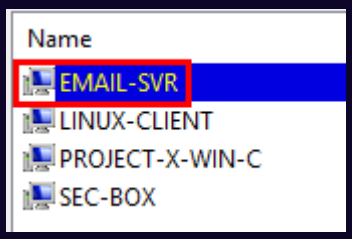
Issue an `id` command to view status:

```
id
```

Success!

```
Creating directory '/home/CORP/email-svr'.
CORP+email-svr@email-svr:~$ id
uid=2001120(CORP+email-svr) gid=2000513(CORP+domain users) groups=2000513(CORP+domain users),2001120
(CORP+email-svr)
CORP+email-svr@email-svr:~$ _
```

Going back to the Server Manager, we should see “LINUX-CLIENT” under the “Computers” folder.



📸 **Take Snapshot!**