Self-hosting

Hashdump Security Club

Cloud-based services

Why use cloud-based services

Advantages

- Infrastructure is managed for you
- Greater exposure (w/ public-facing platforms like GitHub)
- Better reliability (backups, automatic failover)

Overall: less effort/overhead

Disadvantages

- Must trust provider not to make mistakes / lose your data
- Provider will use your data for their own purposes
- Data may be shared with third parties
- May need to pay subscription fees

Overall: less privacy/control

Alternatives to the cloud

- **Self-hosting:** running services on your own hardware, often to replace cloud-based applications
- *Disadvantages:* more effort to set up and maintain own servers
- *Advantages:* more control over your own data, greater flexibility, lower latency
- Hardware requirements are low: if you have an old desktop or laptop lying around, it can become a server
- Many free and open-source programs are available

Use cases

Looking at software we've used in practice

Nextcloud

Alternative to Google/Microsoft suites

File/photo storage

Calendar

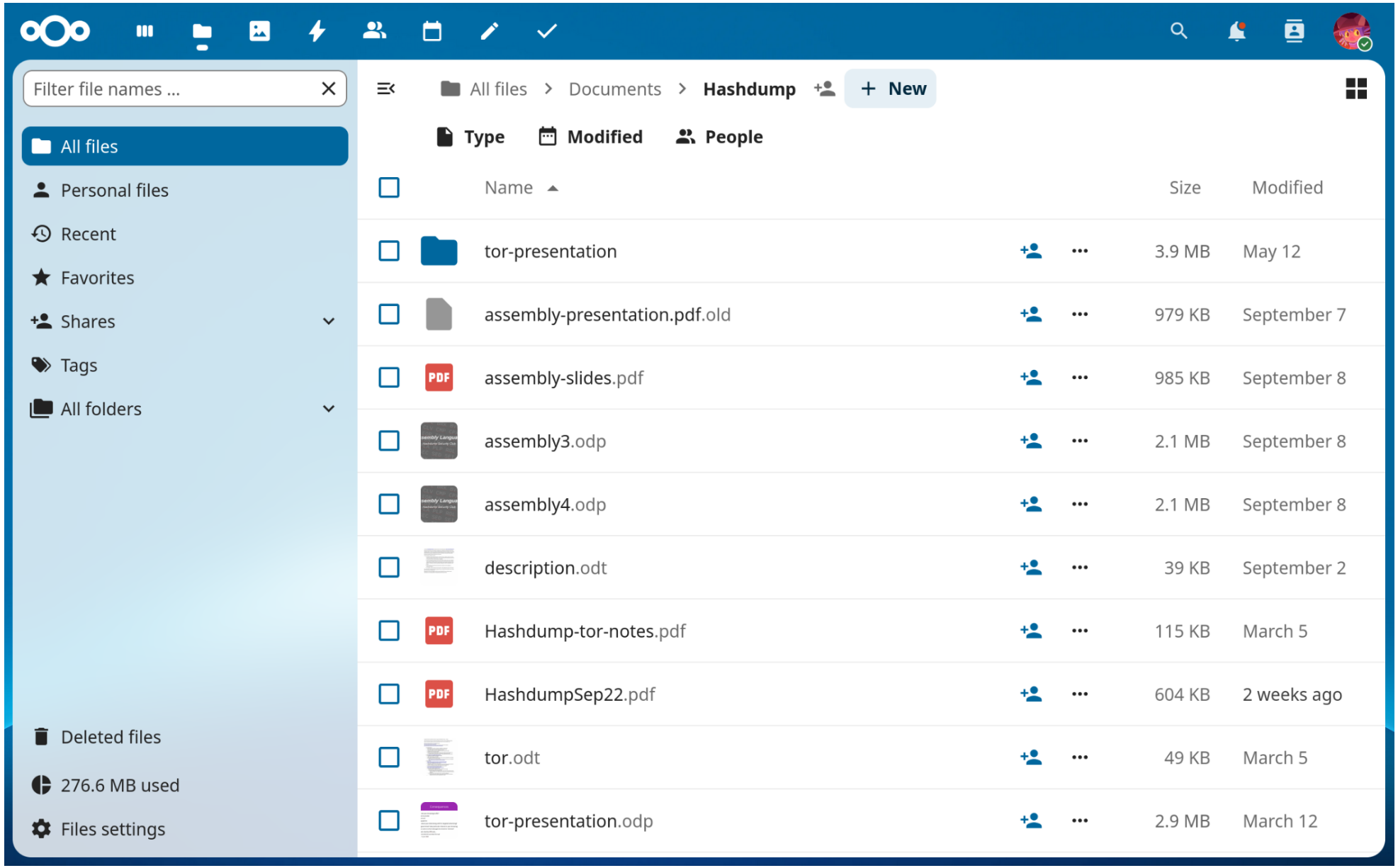
Contacts

Notes

Tasks

Office suite

Mobile app integration



Forgejo, Gitea

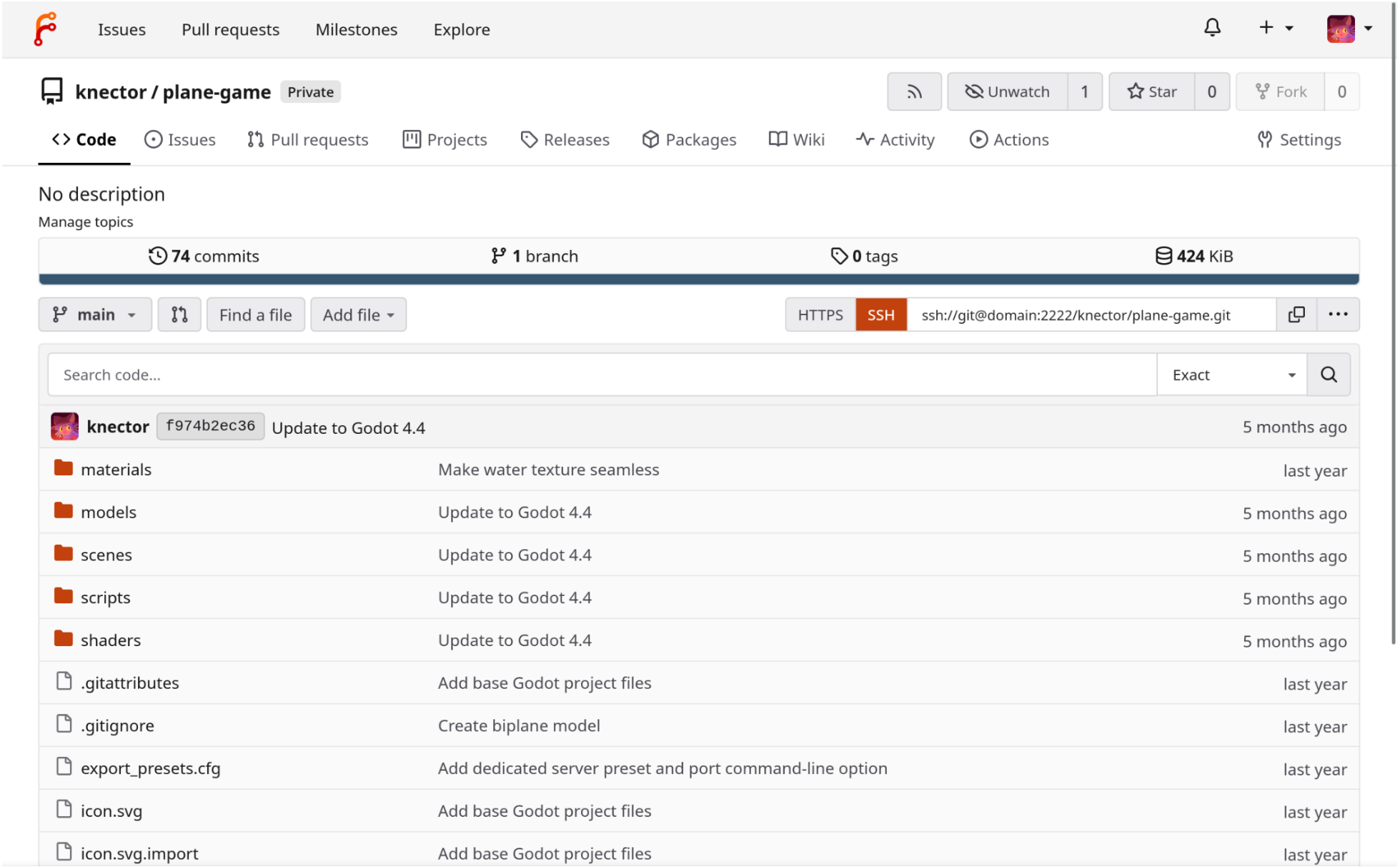
Host Git source code repositories

Allow downloading release binaries

Built-in container registry

Forgejo is community-run fork of Gitea, itself based on Gogs

See also: GitLab



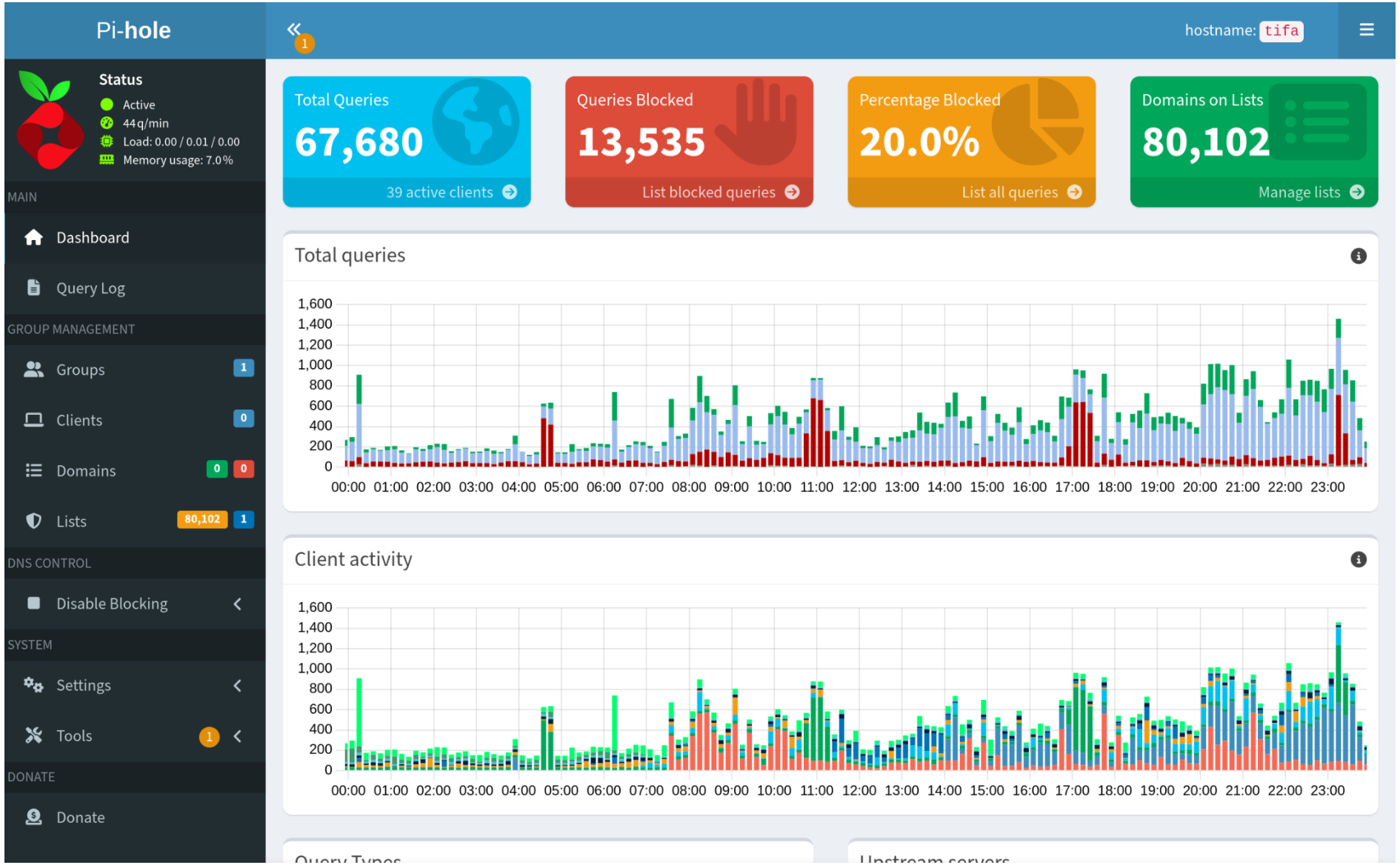
Pi-hole

Network-wide ad blocker

DNS-based blocking: apply filter rules for most devices

Allows setting custom DNS records for internal webpages

See also: AdGuard Home

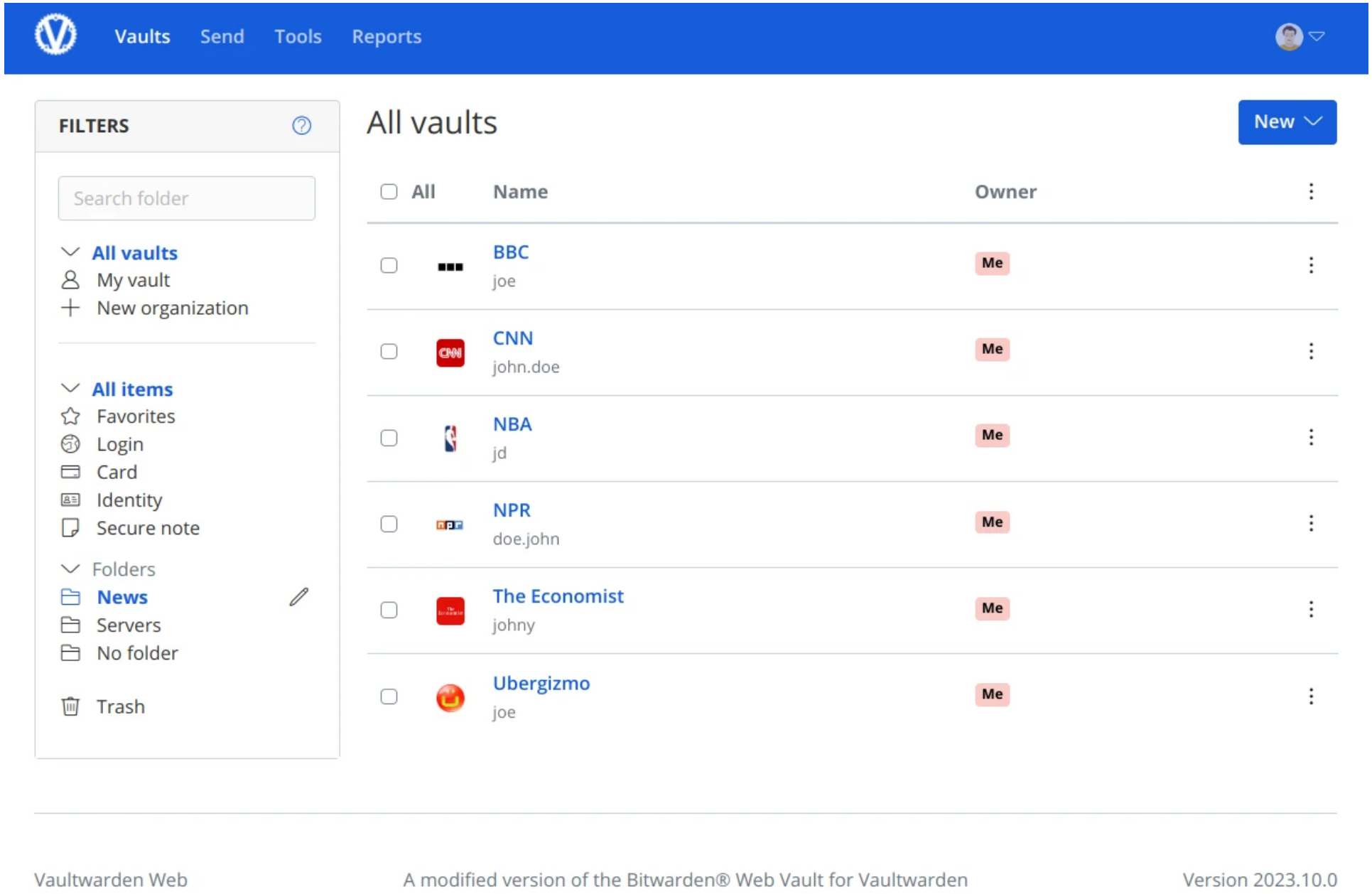


Vaultwarden

Self-hosted password manager

Alternative to LastPass, Dashlane, 1Password, etc.

For local-only password storage, see KeePassXC



Screenshot: https://etke.cc/help/extras/vaultwarden/

https://en.wikipedia.org/wiki/Element_(software)

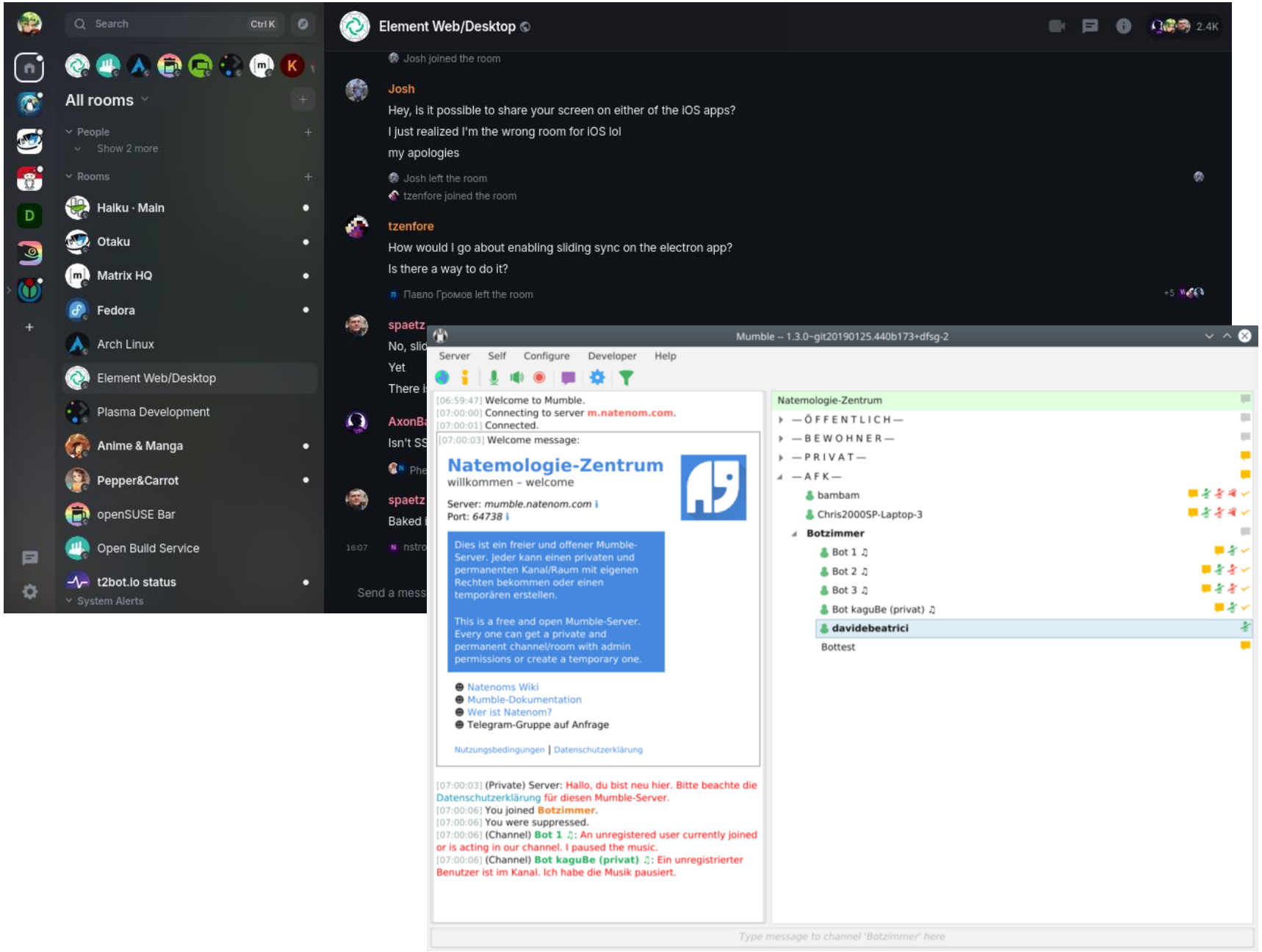
Communication

Can run your own chat servers

Matrix: like Discord or Slack, has variety of clients + supports video calling

Mumble: voice chat like Teamspeak

Honorable mentions: Rocket.Chat, Zulip, Mattermost, Stoat (previously Revolt), Jitsi Meet



https://en.wikipedia.org/wiki/Mumble_(software)

Email

Various software exists to run your own email server

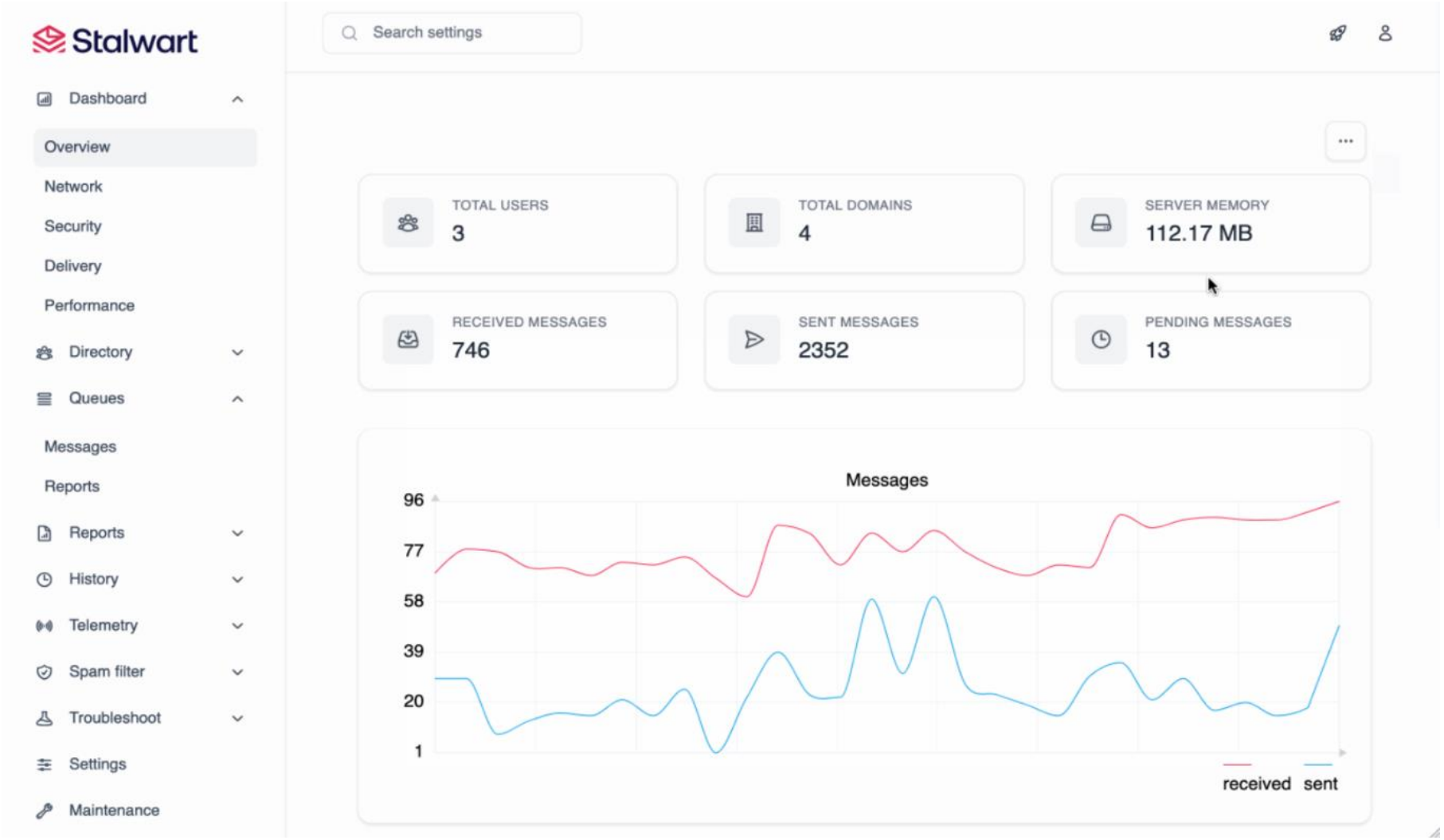
Stalwart (pictured)

Mail-in-a-Box

Mailcow

Difficult self-hosting task, due to security requirements

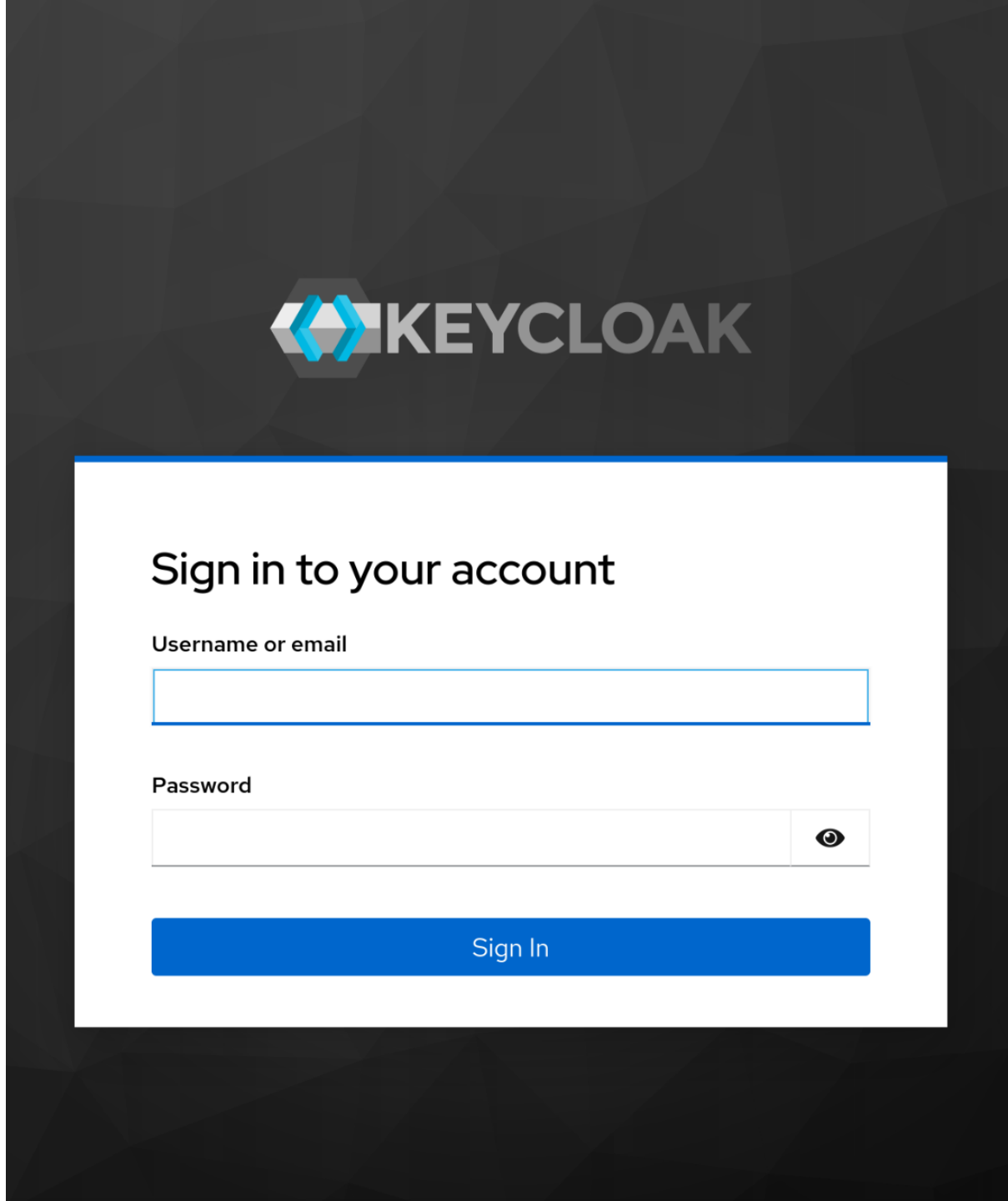
Many of these projects attempt to streamline setup



https://stalw.art/slides/#/57

User authentication

- Centralized account system simplifies access if you have many apps
- **Single Sign-On** (SSO): consistent login flow using SAML or OpenID Connect protocols
- Lightweight Directory Access Protocol (LDAP) is a standard for user account management
- Popular solutions: Authelia, Authentik, FreeIPA, Pocket ID, Keycloak, Kanidm, Shibboleth, LLDAP, OpenLDAP



Virtualization: Proxmox VE

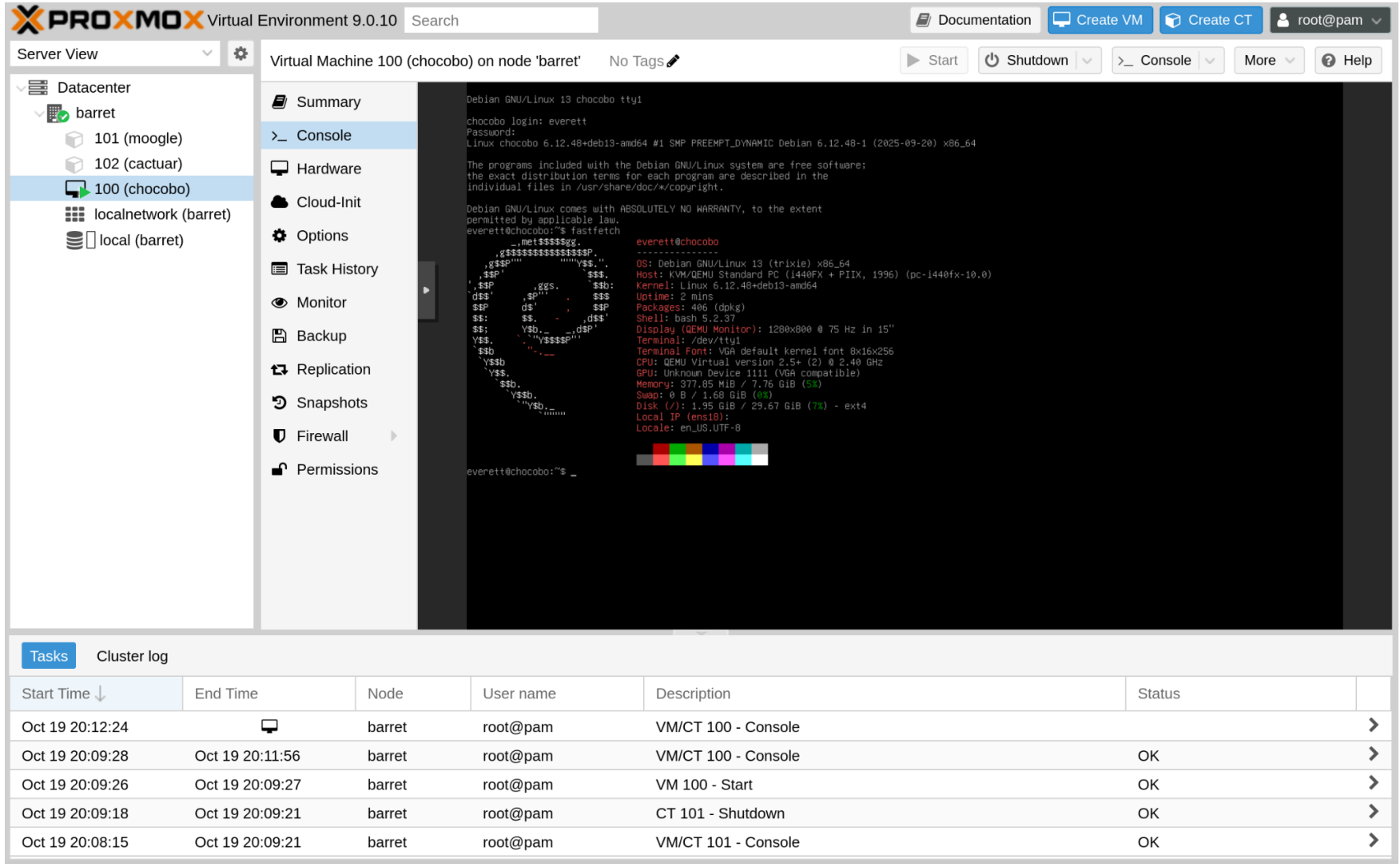
Run miniature Linux environments to host server applications

Allows quick provisioning of

- Virtual machines (VMs)
- Operating system containers

Benefits: better isolation, leverage features of varied Linux distributions

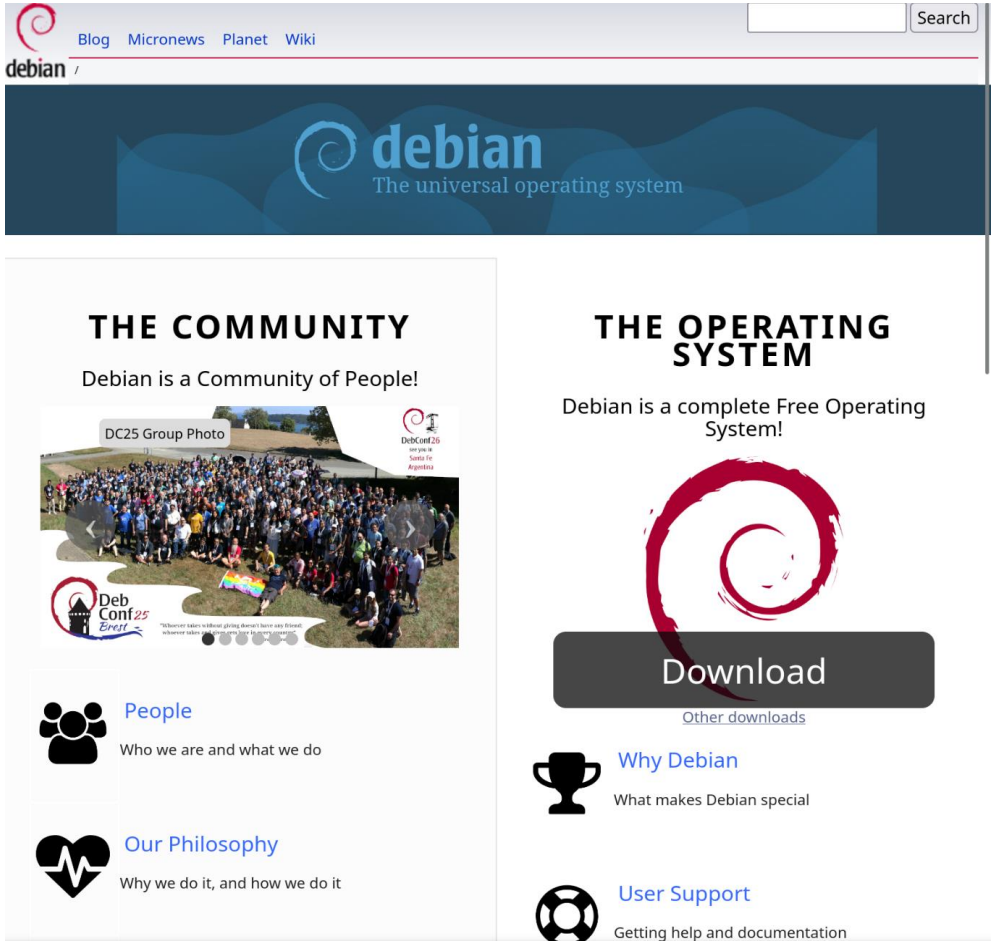
Virtual machines can host other OS platforms such as Windows as well



Setup

Software environment

- After finding a computer, you'll typically install a server OS on it
  - Commonly a Linux distribution (Debian, Ubuntu, AlmaLinux…)
- Some OS platforms aid in self-hosting
  - TrueNAS (free community version)
  - HexOS (paid, based on TrueNAS)
  - Unraid (paid)



Installing software

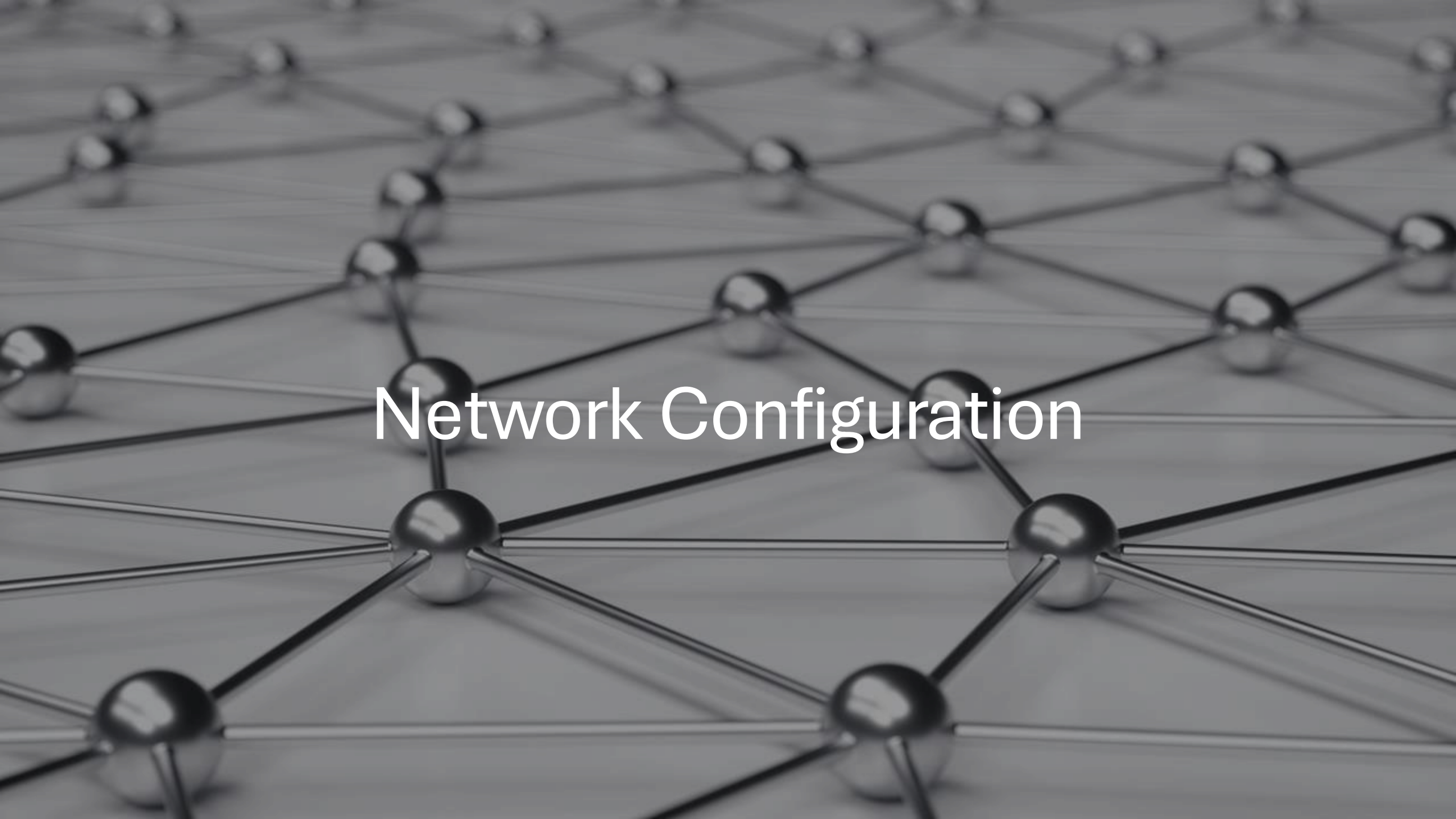
- Linux distributions provide many programs through *packages* in their *repositories*
  - Collections of software assembled by distribution maintainers
  - Includes core ones like Nginx (reverse proxy) or PostgreSQL (database)
  - Simple installation through terminal commands like "apt install nginx"
- In cases where distributions don't carry a program, there are other methods
- For example, *container images*
  - Package a server application together with the Linux OS needed to run it
  - Require a *container runtime* such as Docker or Podman
- Other programs may provide their own binaries or packages

Initial app configuration

- *Configuration files* adjust server behavior
- This may include port numbers where you can access them over the network
  - Commonly HTTP (Hypertext Transfer Protocol), the protocol of the Web
- May require IP addresses + ports for external programs, e.g. databases
- After configuring + starting program, it should be reachable over the network

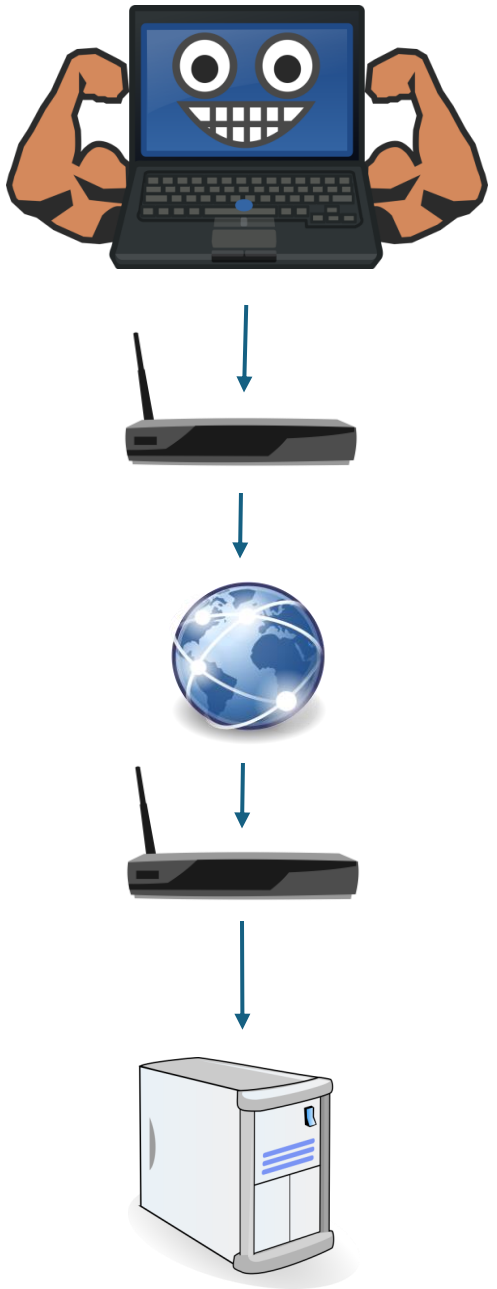
```php
<?php
$CONFIG = array (
  'instanceid' => 'oc8c0fd71e03',
  'passwordsalt' => '515a13302a6b3950a9d0fdb970191a',
  'trusted_domains' =>
  array (
    0 => 'localhost',
    1 => 'studio',
    2 => '192.168.10.155'
  ),
  'datadirectory' => '/var/www/nextcloud/data',
  'dbtype' => 'mysql',
   'version' => '7.0.2.1',
  'dbname' => 'nextcloud',
  'dbhost' => 'localhost',
  'dbtableprefix' => 'oc_',
  'dbuser' => 'oc_carla',
  'dbpassword' => '67336bcdf7630dd80b2b81a413d07',
  'installed' => true,
);
```

https://docs.nextcloud.com/server/stable/admin_manual/configuration_server/config_sample_php_parameters.html


Network Configuration

Allowing others access

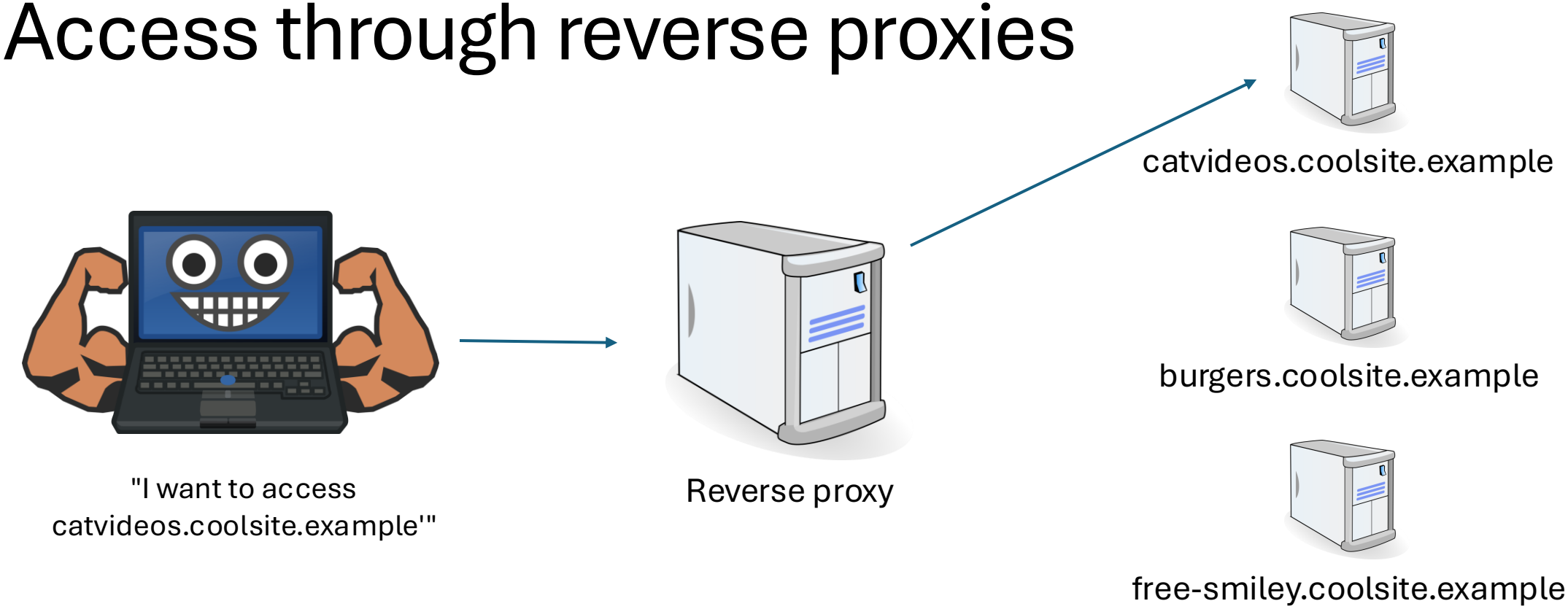
- If you're on your network, you can reach other devices on that same network.
  - But how do you allow people outside your household to connect?
- **Port forwarding**
  - Sets your router to forward traffic it receives over a specific port (often 80 and 443) to your server
  - What if that's unavailable?
- Relay servers (e.g. Cloudflare tunnels)
  - Uses an intermediary to facilitate connections to your computer without needing to open ports.
  - Your server IP remains private from people accessing your site.



Domain Name System (DNS)

- IP addresses are how computers on the Internet identify each other
- But they aren't human readable: to visit google.com, you don't enter its IP address "142.250.69.238"
- For a yearly fee, you can purchase a *domain name* on the Internet and point it toward your own server's address
  - Best to acquire domain privacy protection as well to avoid spam
- After doing so, you will want to acquire a TLS certificate to encrypt traffic to/from your domain
  - Let's Encrypt allows you to obtain certificates for free

Access through reverse proxies



- If you don't want users to remember port numbers and which machine hosts what service, reverse proxies have you covered
- Proxy receives and forwards HTTP traffic to the appropriate destination server
- Particularly allows for multiple domains & routes to be associated with one open port on a network.
  - Examples: Apache httpd, nginx, HAProxy, Envoy, Traefik, Caddy

Managing systems remotely

- Port-forwarding SSH?
  - Consideration: attack surface
  - Disallow password auth and use public-key instead – eliminate possibility of weak passwords
  - Fail2ban can ban IPs that attempt to brute-force login
- VPNs allow restricting access to apps and management ports
  - WireGuard: can open a port for outside VPN connection, public-key auth
  - TailScale: service built on WireGuard, including relay servers
  - WARP-gated Cloudflare tunnels: accessible through zero-trust auth

Conclusions

- Self-hosting is a hands-on way to learn about networking technology and server administration
- System can be simple or complex depending on your needs
- There are many communities online that provide further resources, including more applications you can run
  - https://github.com/awesome-selfhosted/awesome-selfhosted
  - https://reddit.com/r/selfhosted
  - https://lemmy.world/c/selfhosted

Try it yourself!

- Download Gitea for your device at the following link
  - https://about.gitea.com/products/gitea/
- Run a local Gitea server on your machine
  - For simplicity, use the built-in SQLite database
- Access your server at http://localhost:3000
  - Fill out the setup wizard to create the administrator account

Thank you

Any questions?