

main

Go to file

Code

 axfelix	bump to pnpm 10 (#349)	✓	f8de5f0 · last week	
	fix merge		2 years ago	
	Account migration gui...		2 years ago	
	make sure update scri...		2 months ago	
	bump to pnpm 10 (#3...		last week	
	pds v0.4.5001		2 months ago	
	Merge pull request #2...		10 months ago	
	build: bump Node to 2...		last week	
	add license		2 years ago	
	add license		2 years ago	
	add license		2 years ago	
	pds v0.4.5001		2 months ago	
	make sure update scri...		2 months ago	
	switch to watchtower ...		8 months ago	
	replace pdsadmin acc...		5 months ago	
	Merge pull request #2...		8 months ago	
	README and default li...		4 months ago	

README

License

Apache-2.0 license

MIT license

PDS

Welcome to the repository for the official Bluesky PDS (Personal Data Server). This repository includes container images and documentation designed to assist technical people with hosting a Bluesky PDS.

Head over to the [ATProto Touchers Discord](#) to chat with other folks hosting instances and get important updates about the PDS distribution!

Table of Contents

- [PDS](#)
 - [Table of Contents](#)
 - [FAQ](#)
 - [What is Bluesky?](#)
 - [What is AT Protocol?](#)
 - [Where is the code?](#)
 - [What is the current status of federation?](#)
 - [Self-hosting a PDS](#)
 - [Deploying a PDS onto a VPS](#)
 - [Open your cloud firewall for HTTP and HTTPS](#)
 - [Configure DNS for your domain](#)
 - [Check that DNS is working as expected](#)
 - [Installing on Ubuntu 20.04/22.04/24.04 and Debian 11/12/13](#)
 - [Verifying that your PDS is online and accessible](#)
 - [goat CLI](#)
 - [Creating an account](#)
 - [Creating an account using an invite code](#)
 - [Using the Bluesky app with your PDS](#)
 - [Setting up SMTP](#)
 - [Common SMTP issues](#)
 - [Logging](#)
 - [Updating your PDS](#)
 - [Environment Variables](#)
 - [Migrating your PDS](#)
 - [Fixing a Relay desync](#)
 - [License](#)

FAQ

What is Bluesky?

Bluesky is a social media application built on AT Protocol.

Please visit the [Bluesky website](#) for more information.

What is AT Protocol?

The Authenticated Transfer Protocol, aka atproto, is a protocol for large-scale distributed social applications.

Please visit the [AT Protocol docs](#) for additional information.

Where is the code?

- [TypeScript code](#)
- [Go code](#)

What is the current status of federation?

- The AT Protocol network is open to federation!
- ✅ Federated domain handles (e.g. @nytimes.com)
 - ✅ Federated feed generators (custom algorithms)
 - ✅ Federated relays (event firehose)
 - ✅ Federated app views (API service)
 - ✅ Federated data (PDS hosting)
 - ✅ Federated moderation (labeling)

Self-hosting a PDS

Self-hosting a Bluesky PDS means running your own Personal Data Server that is capable of federating with the wider Bluesky social network.

Deploying a PDS onto a VPS

This README provides instructions for deploying a PDS using our install script onto a Virtual Private Server. [Digital Ocean](#) and [Vultr](#) are two popular choices for VPS hosting.

Ensure that you can ssh to your server and have root access.

Server Requirements

- Public IPv4 address
- Public DNS name
- Public inbound internet access permitted on port 80/tcp and 443/tcp

Server Recommendations

Operating System	Ubuntu 24.04
Memory (RAM)	1 GB
CPU Cores	1
Storage	20 GB SSD
Architectures	amd64, arm64
Number of users	1-20

Note: It is a good security practice to restrict inbound ssh access (port 22/tcp) to your own computer's public IP address. You can check your current public IP address using [ifconfig.me](#).

Open your cloud firewall for HTTP and HTTPS

One of the most common sources of misconfiguration is not opening firewall ports correctly. Please be sure to double check this step.

In your cloud provider's console, the following ports should be open to inbound access from the public internet.

- 80/tcp (Used only for TLS certification verification)
- 443/tcp (Used for all application requests)

Note: there is no need to set up TLS or redirect requests from port 80 to 443 because the Caddy web server, included in the Docker compose file, will handle this for you.

Configure DNS for your domain

From your DNS provider's control panel, set up a domain with records pointing to your server.

Name	Type	Value	TTL
example.com	A	12.34.56.78	600
*.example.com	A	12.34.56.78	600

Note:

- Replace `example.com` with your domain name.
- Replace `12.34.56.78` with your server's IP address.
- Some providers may use the `@` symbol to represent the root of your domain.
- The wildcard record is required when allowing users to create new accounts on your PDS.
- The TTL can be anything but 600 (10 minutes) is reasonable

Check that DNS is working as expected

Use a service like [DNS Checker](#) to verify that you can resolve domain names.

Examples to check (record type `A`):

- `example.com`
- `random.example.com`
- `test123.example.com`

These should all return your server's public IP.

Installing on Ubuntu 20.04/22.04/24.04 and Debian 11/12/13

Note that this script assumes a relatively "fresh" VPS that is not also concurrently hosting a web server or anything else on port 80/443. If you intend to run a PDS alongside an existing webserver on the same VPS, you will not want to use this install script.

On your server, download the install script using `curl`:

```
curl https://raw.githubusercontent.com/bluesky-so
```

And then run the installer using `bash`. You will need `sudo` permissions to continue:

```
sudo bash installer.sh
```

The install script is interactive and will prompt for input during the install process. You will need to provide your public DNS address, an admin email address (which does not need to be from the same domain), and be prompted to create a PDS user account with its own email address and handle. If you plan to reuse an existing AT handle, you can skip user account creation, though if it is your first time deploying a PDS you may want to create an account using your domain like `account.your-domain.net` for testing purposes.

Upon completion of a successful installation, you'll receive output similar to the following:

```
=====
PDS installation successful!
-----

Check service status      : sudo systemctl statu
Watch service logs       : sudo docker logs -f |
Backup service data      : /pds
PDS Admin command        : pdsadmin

Required Firewall Ports
-----
Service      Direction  Proto
-----
HTTP TLS verification Inbound   80      TCP
HTTP Control Panel  Inbound   443     TCP

Required DNS entries
-----
Name          Type      Value
-----
your-domain.net      A        your-ip-i
*.your-domain.net    A        your-ip-i

Detected public IP of this server: your-ip-addr:

To see pdsadmin commands, run "pdsadmin help"

=====
```

And, following account creation:

```
Account created successfully!
-----
Handle      : handle.your-domain.net
DID         : did:plc:your-did
Password    : your-password
-----
Save this password, it will not be displayed aga.
```

Verifying that your PDS is online and accessible

Tip

The most common problems with getting PDS content consumed in the live network usually result from users trying to port the provided Caddy configuration to Nginx, Apache, or other reverse proxies. Getting TLS certificates, WebSockets, and virtual server names provisioned can be challenging. We are not currently providing tech support for other configurations.

After installation, your PDS should be live and accessible on the web. You can check if your server is online and healthy by making a request to `https://your-domain.net/xrpc/_health` (the healthcheck endpoint). You should see a JSON response with a version, like:

```
{ "version": "0.2.2-beta.2" }
```

You'll also need to check that WebSockets are working, for the rest of the network to pick up content from your PDS. You can test by installing a tool like `wsdump` and running a command like:

```
wsdump "wss://example.com/xrpc/com.atproto.sync.:"
```

Note that there will be no events output on the WebSocket until they are created in the PDS, so the above command may continue to run with no output immediately post-installation.

goat CLI

The PDS image includes [goat](#), our command line tool for performing admin functions. `goat` can be used locally with a `--pds-host` parameter or within the container installed by this script.

Creating an account

You can run the `goat` command included in this container with `docker exec pds goat`.

Use `goat pds admin account create` to create an account if you haven't already:

```
docker exec pds goat pds admin account create -i
```

Note

You can find `PDS_ADMIN_PASSWORD` in `/pds/pds.env` following installation. You can typically skip this arg when running `goat` from the `docker` container as `PDS_ADMIN_PASSWORD` should already be set.

Creating an account using an invite code

If needed, use `goat` to create an invite code:

```
docker exec pds goat pds admin --admin-password I
```

When creating an account using the app, enter this invite code.

Using the Bluesky app with your PDS

You can use the Bluesky app to connect to your PDS.

1. Get the Bluesky app
 - [Bluesky for Web](#)
 - [Bluesky for iPhone](#)
 - [Bluesky for Android](#)
2. Enter the URL of your PDS (e.g. `https://example.com/`)

Note: because the subdomain TLS certificate is created on-demand, it may take 10-30s for your handle to be accessible. If you aren't seeing your first post/profile, wait 30s and try to make another post.

Setting up SMTP

To be able to verify users' email addresses and send other emails, you need to set up an SMTP server.

As an alternative to running your own SMTP server, you can use an email service. [Resend](#) and [SendGrid](#) are two popular choices.

Create an account and API key on an email service, ensure your server allows access on the required ports, and then you can add these configuration variables to `/pds/pds.env` on your server (example with Resend):

```
PDS_EMAIL_SMTP_URL=smtps://resend:<your api key I
PDS_EMAIL_FROM_ADDRESS=admin@your.domain
```

If you prefer to use a standard SMTP server (a local one or from your email provider), put your account's username and password in the URL:

```
PDS_EMAIL_SMTP_URL=smtps://username:password@sm
```

Alternatively, if you're running a local sendmail-compatible mail service like Postfix or Exim on the same host, you can configure the PDS to use the sendmail transport by using such URL:

```
PDS_EMAIL_SMTP_URL=smtp://?sendmail=true
```

Note: Your PDS will need to be restarted with those variables. This varies depending on your setup. If you followed this installation guide, run `systemctl restart pds`. You might need to restart the server or recreate the container, depending on what you are using.

Common SMTP issues

If you find that your test messages using `cURL` or other sources go out correctly, but you are not receiving emails from your PDS, you may need to URL encode your username and password on `/pds/pds.env` and restart the PDS service.

If the username and/or password contain special characters, the special characters will need to be [percent encoded](#). For some email services, the username may contain an extra `@` symbol that will also need to be percent encoded. For example, the URL

`user&name@oc1:p@ssword@smtphost:465` after percent encoding for the username and password fields would become `user%26name%40oc1:p%40ssword@smtphost:465`.

If you are migrating an account, Bluesky's UI will ask you to confirm your email address. The confirmation code email is meant to come from your PDS. If you are encountering issues with SMTP and want to confirm the address before solving it, you can find the confirmation code on the `email_token` table on `accounts.sqlite`.

Logging

By default, logs from the PDS are printed to `stdout` and end up in Docker's log. You can browse them by running:

```
[sudo] docker logs pds
```

Note: these logs are not persisted, so they will be lost after server reboot.

Alternatively, you can configure the logs to be printed to a file by setting `LOG_DESTINATION`:

```
LOG_DESTINATION=/pds/pds.log
```

You can also change the minimum level of logs to be printed (default: `info`):

```
LOG_LEVEL=debug
```

Updating your PDS

It is recommended that you keep your PDS up to date with new versions. This repo sets up [watchtower](#) which handles automatic updates. You can also use the `pdsadmin` tool to manually update your PDS.

```
sudo pdsadmin update
```

Environment Variables

Environment Variable	
PDS_HOSTNAME	None
PDS_JWT_SECRET	None
PDS_ADMIN_PASSWORD	None
PDS_PLC_ROTATION_KEY_K256_PRIVATE_KEY_HEX	None
PDS_DATA_DIRECTORY	/pds
PDS_BLOBSTORE_DISK_LOCATION	/pds/blob
PDS_BLOB_UPLOAD_LIMIT	10485766
PDS_DID_PLC_URL	https://
PDS_BSKY_APP_VIEW_URL	https://
PDS_BSKY_APP_VIEW_DID	did:web:
PDS_REPORT_SERVICE_URL	https://
PDS_REPORT_SERVICE_DID	did:plc:

About

Bluesky PDS (Personal Data Server) container image, compose file, and documentation

[self-hosted](#) [bluesky](#)

[atproto](#)

 Readme

 Unknown and 2 other licenses found

 Activity

 Custom properties

 2.6k stars

 43 watching

 299 forks

Report repository

Packages 1

 **pds**

Contributors 33



[+ 19 contributors](#)

Languages

 Shell 86.7%

 TypeScript 9.0%

 Dockerfile 4.3%

PDS_CRAWLERS	https://
LOG_ENABLED	true
PDS_EMAIL_SMTP_URL	None
PDS_EMAIL_FROM_ADDRESS	None
PDS_CONTACT_EMAIL_ADDRESS	None
PDS_PRIVACY_POLICY_URL	None
PDS_TERMS_OF_SERVICE_URL	None
PDS_RATE_LIMITS_ENABLED	true
PDS_INVITE_REQUIRED	true

Migrating your PDS

Once you've deployed a PDS, it will automatically begin broadcasting events to Relay servers.

This means that if you wipe and reinstall your PDS on the same hostname, or move it to a new hostname without performing the appropriate cutover steps, the PDS and Relay can go out of sync.

To avoid this, you should always [migrate accounts individually](#) from one PDS to another when updating your PDS host configuration.

Fixing a Relay desync

If you become desynced from the Relay due to migration issues — i.e., new records created on your PDS aren't being picked up by other applications — you can fix it with these steps:

1. Look up the last known sequence number for your PDS host. You can use `goat` to retrieve the last observed cursor from the `bsky.network` relay:

```
docker exec pds sh -c 'goat relay host status "$@"'
```

```
{"hostname": "justdothings.net", "seq": 12, "status" }
```

2. Stop the `pds` service:

```
systemctl stop pds
```

3. Set the cursor sequence on your PDS to the value of `seq` from step 1, incremented by 1:

```
sqlite3 /pds/sequencer.sqlite "UPDATE sqlite_seqi
```

4. Restart your `pds` service:

```
systemctl start pds
```

New records should now be indexed as normal.

License

This project is dual-licensed under MIT and Apache 2.0 terms:

- MIT license ([LICENSE-MIT.txt](#) or <http://opensource.org/licenses/MIT>)
- Apache License, Version 2.0, ([LICENSE-APACHE.txt](#) or <http://www.apache.org/licenses/LICENSE-2.0>)

Downstream projects and end users may choose either license individually, or both together, at their discretion. The motivation for this dual-licensing is the additional software patent assurance provided by Apache 2.0.