

---

# Installing Katzenpost

## Table of Contents

Requirements .....	1
Obtain the Katzenpost code .....	1
Install the latest Go version .....	1
Build server components .....	2
Install the server components .....	2
Create service accounts .....	2
Create configuration files .....	2
Configure systemd .....	3
Generate keys .....	3
Build the chat client .....	4

The section provides an overview of how to download Katzenpost, set up a development environment, build the code, install the Katzenpost binaries, and configure the components.

## Requirements

An up-to-date Debian [<https://www.debian.org>] or Ubuntu [<https://ubuntu.com>] Linux system is assumed as the build and hosting environment for all Katzenpost server-side components. Required packages include the following:

- **git**
- **gcc**
- **build-essential**
- **libc-dev-bin**

## Obtain the Katzenpost code

Complete the following steps to set up a local Katzenpost git repository.

1. Clone Katzenpost.

```
$ git clone git@github.com:katzenpost/katzenpost.git
```

2. Get the latest tagged commit of the Katzenpost with the following commands.

```
$ git fetch --tags
$ tag=$(git describe --tags `git rev-list --tags --max-count=1`)
$ git checkout $tag
```

## Install the latest Go version

Download the latest version of the Go programming language from <https://go.dev/dl> and unzip it in a suitable location. As root, set the necessary environment variables:

```
# export PATH=$PATH:/<your Go location>/bin
# export GOMODCACHE=$PATH
# export CGO_CFLAGS_ALLOW="-DPARAMS=sphincs-shake-256f"
```

The `go/bin` path must be included in your user `$PATH` environment variable.

### Note

Do not use the Debian/Ubuntu **golang** packages. They are probably too old.

## Build server components

To build each of the Katzenpost components, navigate to the directory containing its source code and run **go build**. The paths shown are relative to the Katzenpost repository root.

**Table 1. Component directories**

Component	Source code directory	Binary
Mix, gateway, or service node	server/cmd/server/	server
Directory authority	authority/cmd/dirauth/	dirauth

## Install the server components

To install the server binaries, run the following commands from the katzenpost repository root.

```
# cp server/cmd/server/server /usr/local/bin/pq-katzenpost-mixserver
# cp authority/cmd/dirauth/dirauth /usr/local/bin/pq-katzenpost-authority
```

## Create service accounts

Create a service account account for each of the node types that you deploy.

**To create a service user for a directory authority.**

```
# adduser \
  --disabled-login \
  --disabled-password \
  --system \
  --group \
  --home /var/lib/pq-katzenpost-authority \
  pq-katzenpost-authority
```

**To create a service user for a mix, gateway, or service node.**

```
# adduser \
  --disabled-login \
  --disabled-password \
  --system \
  --group \
  --home /var/lib/pq-katzenpost-mixserver \
  pq-katzenpost-mixserver
```

## Create configuration files

It is possible, though challenging, to construct a node configuration file based on the published component parameters, the example of the Docker test image, and the latest state of the code tree. Katzenpost currently has no configuration automation tool that is ready for general use. If you plan to implement a mix network from scratch, we suggest that you contact the development team for assistance with configuration.

## Configure systemd

If you are running your Katzenpost components under systemd [<https://systemd.io/>], create and install a systemd service file for each node type that you plan to deploy. The following scripts are examples of how to do this.

**To create a systemd service file for a directory authority.**

```
#!/bin/bash -x
```

```
cat << EOF > /etc/systemd/system/pq-katzenpost-mixserver.service
```

```
[Unit]
```

```
Description=pq Katzenpost Mix Server
```

```
After=network.target
```

```
[Service]
```

```
IPAccounting=yes
```

```
Type=simple
```

```
User=pq-katzenpost-mixserver
```

```
WorkingDirectory=/var/lib/pq-katzenpost-mixserver
```

```
ExecStart=/usr/local/bin/pq-katzenpost-mixserver -f /etc/pq-katzenpost-mixserver
```

```
PrivateTmp=yes
```

```
NoNewPrivileges=yes
```

```
# RestartSec=5
```

```
Restart=on-failure
```

```
[Install]
```

```
WantedBy=default.target
```

```
EOF
```

**To create a systemd service file for a mix, gateway, or service node.**

```
#!/bin/bash -x
```

```
cat << EOF > /etc/systemd/system/pq-katzenpost-authority.service
```

```
[Unit]
```

```
Description=pq Katzenpost Authority
```

```
After=network.target
```

```
[Service]
```

```
Type=simple
```

```
IPAccounting=yes
```

```
User=pq-katzenpost-authority
```

```
WorkingDirectory=/var/lib/pq-katzenpost-authority
```

```
ExecStart=/usr/local/bin/pq-katzenpost-authority -f /etc/pq-katzenpost-authority
```

```
PrivateTmp=yes
```

```
NoNewPrivileges=yes
```

```
Restart=on-failure
```

```
[Install]
```

```
WantedBy=default.target
```

```
EOF
```

## Generate keys

The first time that you run a server binary directly or using systemd, identity and encryption keys are automatically generated and installed if they are not already present. The key location is speci-

fied by the value of `DataDir` in the `[Server]` section of the configuration. For configuration parameter details, see Components and configuration of the Katzenpost mixnet [<https://katzenpost.network/components.html>]. For server binary commandline options, see the Quickstart guide [<https://katzenpost.network/quickstart.html>].

Once the keys are in place, restart the server to begin operations.

## Build the chat client

To build the Katzen chat client, navigate to the katzen repository and run **go build**.

### Note

The Katzen client is under development and not currently usable..