
NixOS Mailserver

NixOS Mailserver Contributors

Aug 10, 2022

CONTENTS

1	Setup Guide	3
1.1	Setup DNS A record for server	3
1.2	Setup the server	4
1.3	Setup all other DNS requirements	4
1.4	Test your Setup	6
2	Contribute or troubleshoot	7
2.1	Run NixOS tests	7
2.2	Contributing to the documentation	7
2.3	Nixops	8
2.4	Imap	8
3	FAQ	9
3.1	catchAll users can't send email as user other than themself	9
4	Release Notes	11
4.1	NixOS 22.05	11
4.2	NixOS 21.11	11
4.3	NixOS 21.05	11
4.4	NixOS 20.09	11
5	Mailserver Options	13
5.1	mailserver	13
5.2	mailserver.loginAccount	20
5.3	mailserver.certificate	21
5.4	mailserver.dkim	22
5.5	mailserver.fullTextSearch	23
5.6	mailserver.redis	25
5.7	mailserver.monitoring	26
5.8	mailserver.backup	26
5.9	mailserver.borg	28
6	Backup Guide	31
7	Add Radicale	33
8	Add Roundcube, a webmail	35
9	Tune spam filtering	37
9.1	Auto-learning	37
9.2	Train from existing folders	37

9.3	Tune symbol weight	37
9.4	Tune action thresholds	38
9.5	Access the rspamd web UI	38
10	Full text search	41
10.1	Enabling full text search	41
10.2	Resource requirements	41
10.3	Mitigating resources requirements	42
11	Nix Flakes	43
12	Indices and tables	45



SETUP GUIDE

Mail servers can be a tricky thing to set up. This guide is supposed to run you through the most important steps to achieve a 10/10 score on <https://mail-tester.com>.

What you need is:

- a server running NixOS with a public IP
- a domain name.

Note: In the following, we consider a server with the public IP 1.2.3.4 and the domain example.com.

First, we will set the minimum DNS configuration to be able to deploy an up and running mail server. Once the server is deployed, we could then set all DNS entries required to send and receive mails on this server.

1.1 Setup DNS A record for server

Add a DNS record to the domain example.com with the following entries

Name (Subdomain)	TTL	Type	Value
mail.example.com	10800	A	1.2.3.4

You can check this with

```
$ ping mail.example.com
64 bytes from mail.example.com (1.2.3.4): icmp_seq=1 ttl=46 time=21.3 ms
...
```

Note that it can take a while until a DNS entry is propagated. This DNS entry is required for the Let's Encrypt certificate generation (which is used in the below configuration example).

1.2 Setup the server

The following describes a server setup that is fairly complete. Even though there are more possible options (see the `default.nix` file), these should be the most common ones.

```
{ config, pkgs, ... }:
{
  imports = [
    (builtins.fetchTarball {
      # Pick a commit from the branch you are interested in
      url = "https://gitlab.com/simple-nixos-mailserver/nixos-mailserver/-/archive/A-
      ↪COMMIT-ID/nixos-mailserver-A-COMMIT-ID.tar.gz";
      # And set its hash
      sha256 = "0000000000000000000000000000000000000000000000000000000000000000";
    })
  ];

  mailserver = {
    enable = true;
    fqdn = "mail.example.com";
    domains = [ "example.com" ];

    # A list of all login accounts. To create the password hashes, use
    # nix run nixpkgs.apacheHttpd -c htpasswd -nbB "" "super secret password" | cut -d: -
    ↪f2
    loginAccounts = {
      "user1@example.com" = {
        hashedPasswordField = "/a/file/containing/a/hashed/password";
        aliases = ["postmaster@example.com"];
      };
      "user2@example.com" = { ... };
    };

    # Use Let's Encrypt certificates. Note that this needs to set up a stripped
    # down nginx and opens port 80.
    certificateScheme = 3;
  };
}
```

After a `nixos-rebuild switch` your server should be running all mail components.

1.3 Setup all other DNS requirements

1.3.1 Set rDNS (reverse DNS) entry for server

Wherever you have rented your server, you should be able to set reverse DNS entries for the IP's you own. Add an entry resolving `1.2.3.4` to `mail.example.com`

You can check this with

```
$ nix-shell -p bind --command "host 1.2.3.4"
4.3.2.1.in-addr.arpa domain name pointer mail.example.com.
```


Note that it can take a while until a DNS entry is propagated.

1.3.2 Set a MX record

Add a MX record to the domain `example.com`.

Name (Subdomain)	Type	Priority	Value
<code>example.com</code>	MX	10	<code>mail.example.com</code>

You can check this with

```
$ nix-shell -p bind --command "host -t mx example.com"
example.com mail is handled by 10 mail.example.com.
```

Note that it can take a while until a DNS entry is propagated.

1.3.3 Set a SPF record

Add a SPF record to the domain `example.com`.

Name (Subdomain)	TTL	Type	Value
<code>example.com</code>	10800	TXT	<code>v=spf1 a:mail.example.com -all</code>

You can check this with

```
$ nix-shell -p bind --command "host -t TXT example.com"
example.com descriptive text "v=spf1 a:mail.example.com -all"
```

Note that it can take a while until a DNS entry is propagated.

1.3.4 Set DKIM signature

On your server, the `opendkim` systemd service generated a file containing your DKIM public key in the file `/var/dkim/example.com.mail.txt`. The content of this file looks like

```
mail._domainkey IN TXT "v=DKIM1; k=rsa; s=email; p=<really-long-key>" ; ----- DKIM mail_
↪for domain.tld
```

where `really-long-key` is your public key.

Based on the content of this file, we can add a DKIM record to the domain `example.com`.

Name (Subdomain)	TTL	Type	Value
<code>mail._domainkey.example.com</code>	10800	TXT	<code>v=DKIM1; p=<really-long-key></code>

You can check this with

```
$ nix-shell -p bind --command "host -t txt mail._domainkey.example.com"
mail._domainkey.example.com descriptive text "v=DKIM1;p=<really-long-key>"
```

Note that it can take a while until a DNS entry is propagated.

1.3.5 Set a DMARC record

Add a DMARC record to the domain `example.com`.

Name (Subdomain)	TTL	Type	Value
<code>_dmarc.example.com</code>	10800	TXT	<code>v=DMARC1; p=none</code>

You can check this with

```
$ nix-shell -p bind --command "host -t TXT _dmarc.example.com"
_dmarc.example.com descriptive text "v=DMARC1; p=none"
```

Note that it can take a while until a DNS entry is propagated.

1.4 Test your Setup

Write an email to your aunt (who has been waiting for your reply far too long), and sign up for some of the finest newsletters the Internet has. Maybe you want to sign up for the [SNM Announcement List](#)?

Besides that, you can send an email to [mail-tester.com](#) and see how you score, and let [mxtoolbox.com](#) take a look at your setup, but if you followed the steps closely then everything should be awesome!

CONTRIBUTE OR TROUBLESHOOT

To report an issue, please go to <https://gitlab.com/simple-nixos-mailserver/nixos-mailserver/-/issues>.

You can also chat with us on the Libera IRC channel `#nixos-mailserver`.

2.1 Run NixOS tests

To run the test suite, you need to enable *Nix Flakes* <https://nixos.wiki/wiki/Flakes#Installing_flakes>.

You can then run the testsuite via

```
$ nix flake check -L
```

Since Nix doesn't guarantee your machine have enough resources to run all test VMs in parallel, some tests can fail. You would then have to run tests manually. For instance:

```
$ nix build .#hydraJobs.x86_64-linux.external-unstable -L
```

2.2 Contributing to the documentation

The documentation is written in RST, build with Sphinx and published by [Read the Docs](#).

For the syntax, see [RST/Sphinx Cheatsheet](#).

The `shell.nix` provides all the tooling required to build the documentation:

```
$ nix-shell  
$ cd docs  
$ make html  
$ firefox ./_build/html/index.html
```

Note if you modify some NixOS mailserver options, you would also need to regenerate the `options.rst` file:

```
$ nix-shell --run generate-rst-options
```

2.3 Nixops

You can test the setup via nixops. After installation, do

```
$ nixops create nixops/single-server.nix nixops/vbox.nix -d mail
$ nixops deploy -d mail
$ nixops info -d mail
```

You can then test the server via e.g. telnet. To log into it, use

```
$ nixops ssh -d mail mailserver
```

2.4 Imap

To test imap manually use

```
$ openssl s_client -host mail.example.com -port 143 -starttls imap
```

3.1 catchAll users can't send email as user other than themself

To allow a `catchAll` user to send mail with the address used as recipient, the option `aliases` has to be used instead of `catchAll`.

For instance, to allow `user@example.com` to catch all mails to the domain `example.com` and send mails with any address of this domain:

```
mailserver.loginAccounts = {  
  "user@example.com" = {  
    aliases = [ "@example.com" ];  
  };  
};
```

See also [this discussion](#) for details.

RELEASE NOTES

4.1 NixOS 22.05

- Make NixOS Mailserver options discoverable from search.nixos.org
- Add a roundcube setup guide in the documentation

4.2 NixOS 21.11

- Switch default DKIM body policy from simple to relaxed ([merge request](#))
- Ensure locally-delivered mails have the X-Original-To header ([merge request](#))
- NixOS Mailserver options are detailed in the [documentation](#)
- New options `dkimBodyCanonicalization` and `dkimHeaderCanonicalization`
- New option `certificateDomains` to generate certificate for additional domains (such as `imap.example.com`)

4.3 NixOS 21.05

- New `fullTextSearch` option to search in messages (based on Xapian) ([Merge Request](#))
- Flake support ([Merge Request](#))
- New `openFirewall` option defaulting to `true`
- We moved from Freenode to Libera Chat

4.4 NixOS 20.09

- IMAP and Submission with TLS wrapped-mode are now enabled by default on ports 993 and 465 respectively
- OpenDKIM is now sandboxed with Systemd
- New `forwards` option to forwards emails to external addresses ([Merge Request](#))
- New `sendingFqdn` option to specify the fqdn of the machine sending email ([Merge Request](#))
- Move the Gitlab wiki to [ReadTheDocs](#)

MAILSERVER OPTIONS

5.1 mailserver

5.1.1 mailserver.debug

Whether to enable verbose logging for mailserver related services. This intended be used for development purposes only, you probably don't want to enable this unless you're hacking on nixos-mailserver.

- Type: boolean
- Default: False

5.1.2 mailserver.domains

The domains that this mail server serves.

- Type: list of strings
- Default: []

5.1.3 mailserver.enable

Whether to enable nixos-mailserver.

- Type: boolean
- Default: False

5.1.4 mailserver.enableImap

Whether to enable IMAP with STARTTLS on port 143.

- Type: boolean
- Default: True

5.1.5 mailserver.enableImapSsl

Whether to enable IMAP with TLS in wrapper-mode on port 993.

- Type: boolean
- Default: True

5.1.6 mailserver.enableManageSieve

Whether to enable ManageSieve, setting this option to true will open port 4190 in the firewall.

The ManageSieve protocol allows users to manage their Sieve scripts on a remote server with a supported client, including Thunderbird.

- Type: boolean
- Default: False

5.1.7 mailserver.enablePop3

Whether to enable POP3 with STARTTLS on port on port 110.

- Type: boolean
- Default: False

5.1.8 mailserver.enablePop3Ssl

Whether to enable POP3 with TLS in wrapper-mode on port 995.

- Type: boolean
- Default: False

5.1.9 mailserver.enableSubmission

Whether to enable SMTP with STARTTLS on port 587.

- Type: boolean
- Default: True

5.1.10 mailserver.enableSubmissionSsl

Whether to enable SMTP with TLS in wrapper-mode on port 465.

- Type: boolean
- Default: True

5.1.11 mailserver.extraVirtualAliases

Virtual Aliases. A virtual alias “*info@example.com*” = “*user1@example.com*” means that all mail to *info@example.com* is forwarded to *user1@example.com*. Note that it is expected that *postmaster@example.com* and *abuse@example.com* is forwarded to some valid email address. (Alternatively you can create login accounts for *postmaster* and (or) *abuse*). Furthermore, it also allows the user *user1@example.com* to send emails as *info@example.com*. It’s also possible to create an alias for multiple accounts. In this example all mails for *multi@example.com* will be forwarded to both *user1@example.com* and *user2@example.com*.

- Type: attribute set of Login Account or non-empty list of Login Accountss
- Default: {}

5.1.12 mailserver.forwards

To forward mails to an external address. For instance, the value {“*user@example.com*” = “*user@elsewhere.com*”;} means that mails to *user@example.com* are forwarded to *user@elsewhere.com*. The difference with the *extraVirtualAliases* option is that *user@elsewhere.com* can’t send mail as *user@example.com*. Also, this option allows to forward mails to external addresses.

- Type: attribute set of list of strings or strings
- Default: {}

5.1.13 mailserver.fqdn

The fully qualified domain name of the mail server.

- Type: string

5.1.14 mailserver.hierarchySeparator

The hierarchy separator for mailboxes used by dovecot for the namespace ‘inbox’. Dovecot defaults to “.” but recommends “/”. This affects how mailboxes appear to mail clients and sieve scripts. For instance when using “.” then in a sieve script “*example.com*” would refer to the mailbox “*com*” in the parent mailbox “*example*”. This does not determine the way your mails are stored on disk. See <https://wiki.dovecot.org/Namespaces> for details.

- Type: string
- Default: .

5.1.15 mailserver.indexDir

Folder to store search indices. If null, indices are stored along with email, which could not necessarily be desirable, especially when the *fullTextSearch* option is enable since indices it creates are voluminous and do not need to be backed up.

Be careful when changing this option value since all indices would be recreated at the new location (and clients would need to resynchronize).

Note the some variables can be used in the file path. See https://doc.dovecot.org/configuration_manual/mail_location/#variables for details.

- Type: null or string
- Default: None

5.1.16 mailserver.keyFile

Scheme 1) Location of the key file

- Type: path

5.1.17 mailserver.ImtpSaveToDetailMailbox

If an email address is delimited by a “+”, should it be filed into a mailbox matching the string after the “+”? For example, `user1+test@example.com` would be filed into the mailbox “test”.

- Type: one of "yes", "no"
- Default: yes

5.1.18 mailserver.localDnsResolver

Runs a local DNS resolver (kresd) as recommended when running rspamd. This prevents your log file from filling up with `rspamd_monitored_dns_mon` entries.

- Type: boolean
- Default: True

5.1.19 mailserver.mailDirectory

Where to store the mail.

- Type: path
- Default: `/var/vmail`

5.1.20 mailserver.mailboxes

The mailboxes for dovecot. Depending on the mail client used it might be necessary to change some mailbox’s name.

- Type: unspecified
- Default:

```
{'Drafts': {'auto': 'subscribe', 'specialUse': 'Drafts'}, 'Junk': {'auto': 'subscribe', 'specialUse': 'Junk'}, 'Sent': {'auto': 'subscribe', 'specialUse': 'Sent'}, 'Trash': {'auto': 'no', 'specialUse': 'Trash'}}
```

5.1.21 mailserver.maxConnectionsPerUser

Maximum number of IMAP/POP3 connections allowed for a user from each IP address. E.g. a value of 50 allows for 50 IMAP and 50 POP3 connections at the same time for a single user.

- Type: signed integer
- Default: 100

5.1.22 mailserver.messageSizeLimit

Message size limit enforced by Postfix.

- Type: signed integer
- Default: 20971520

5.1.23 mailserver.openFirewall

Automatically open ports in the firewall.

- Type: boolean
- Default: True

5.1.24 mailserver.policydSPFExtraConfig

Extra configuration options for policyd-spf. This can be used to among other things skip spf checking for some IP addresses.

- Type: strings concatenated with "\n"
- Default: ""

5.1.25 mailserver.rebootAfterKernelUpgrade.enable

Whether to enable automatic reboot after kernel upgrades. This is to be used in conjunction with `system.autoUpgrade.enable = true`

- Type: boolean
- Default: False

5.1.26 mailserver.rebootAfterKernelUpgrade.method

Whether to issue a full “reboot” or just a “systemctl kexec”-only reboot. It is recommended to use the default value because the quicker kexec reboot has a number of problems. Also if your server is running in a virtual machine the regular reboot will already be very quick.

- Type: one of "reboot", "systemctl kexec"
- Default: reboot

5.1.27 mailserver.recipientDelimiter

Configure the recipient delimiter.

- Type: string
- Default: +

5.1.28 mailserver.rejectRecipients

Reject emails addressed to these local addresses from unauthorized senders. Use if a spammer has found email addresses in a catchall domain but you do not want to disable the catchall.

- Type: list of strings
- Default: []

5.1.29 mailserver.rejectSender

Reject emails from these addresses from unauthorized senders. Use if a spammer is using the same domain or the same sender over and over.

- Type: list of strings
- Default: []

5.1.30 mailserver.rewriteMessageId

Rewrites the Message-ID's hostname-part of outgoing emails to the FQDN. Please be aware that this may cause problems with some mail clients relying on the original Message-ID.

- Type: boolean
- Default: False

5.1.31 mailserver.sendingFqdn

The fully qualified domain name of the mail server used to identify with remote servers.

If this server's IP serves purposes other than a mail server, it may be desirable for the server to have a name other than that to which the user will connect. For example, the user might connect to `mx.example.com`, but the server's IP has reverse DNS that resolves to `myserver.example.com`; in this scenario, some mail servers may reject or penalize the message.

This setting allows the server to identify as `myserver.example.com` when forwarding mail, independently of *fqdn* (which, for SSL reasons, should generally be the name to which the user connects).

Set this to the name to which the sending IP's reverse DNS resolves.

- Type: string
- Default: `config.mailserver.fqdn`

5.1.32 mailserver.sieveDirectory

Where to store the sieve scripts.

- Type: path
- Default: `/var/sieve`

5.1.33 mailserver.useFsLayout

Sets whether dovecot should organize mail in subdirectories:

- `/var/vmail/example.com/user/.folder.subfolder/` (default layout)
- `/var/vmail/example.com/user/folder/subfolder/` (FS layout)

See <https://wiki2.dovecot.org/MailboxFormat/Maildir> for details.

- Type: `boolean`
- Default: `False`

5.1.34 mailserver.virusScanning

Whether to activate virus scanning. Note that virus scanning is `_very_` expensive memory wise.

- Type: `boolean`
- Default: `False`

5.1.35 mailserver.vmailGroupName

The user name and group name of the user that owns the directory where all the mail is stored.

- Type: `string`
- Default: `virtualMail`

5.1.36 mailserver.vmailUID

The unix UID of the virtual mail user. Be mindful that if this is changed, you will need to manually adjust the permissions of `mailDirectory`.

- Type: `signed integer`
- Default: `5000`

5.1.37 mailserver.vmailUserName

The user name and group name of the user that owns the directory where all the mail is stored.

- Type: `string`
- Default: `virtualMail`

5.2 mailserver.loginAccount

5.2.1 mailserver.loginAccounts

The login account of the domain. Every account is mapped to a unix user, e.g. *user1@example.com*. To generate the passwords use *htpasswd* as follows

```
` nix run nixpkgs.apacheHttpd -c htpasswd -nbB "" "super secret password" | cut -d: -f2`
```

- Type: attribute set of submodules
- Default: {}

5.2.2 mailserver.loginAccounts.<name>.aliases

A list of aliases of this login account. Note: Use list entries like “@example.com” to create a catchAll that allows sending from all email addresses in these domain.

- Type: list of strings
- Default: []

5.2.3 mailserver.loginAccounts.<name>.catchAll

For which domains should this account act as a catch all? Note: Does not allow sending from all addresses of these domains.

- Type: list of impossible (empty enum)s
- Default: []

5.2.4 mailserver.loginAccounts.<name>.hashedPassword

The user’s hashed password. Use *htpasswd* as follows

```
` nix run nixpkgs.apacheHttpd -c htpasswd -nbB "" "super secret password" | cut -d: -f2`
```

Warning: this is stored in plaintext in the Nix store! Use *hashedPasswordFile* instead.

- Type: null or string
- Default: None

5.2.5 mailserver.loginAccounts.<name>.hashedPasswordFile

A file containing the user’s hashed password. Use *htpasswd* as follows

```
` nix run nixpkgs.apacheHttpd -c htpasswd -nbB "" "super secret password" | cut -d: -f2`
```

- Type: null or path
- Default: None

5.2.6 mailserver.loginAccounts.<name>.name

Username

- Type: string

5.2.7 mailserver.loginAccounts.<name>.quota

Per user quota rules. Accepted sizes are *xx k/M/G/T* with the obvious meaning. Leave blank for the standard quota *100G*.

- Type: null or string
- Default: None

5.2.8 mailserver.loginAccounts.<name>.sendOnly

Specifies if the account should be a send-only account. Emails sent to send-only accounts will be rejected from unauthorized senders with the `sendOnlyRejectMessage` stating the reason.

- Type: boolean
- Default: False

5.2.9 mailserver.loginAccounts.<name>.sendOnlyRejectMessage

The message that will be returned to the sender when an email is sent to a send-only account. Only used if the account is marked as send-only.

- Type: string
- Default: This account cannot receive emails.

5.2.10 mailserver.loginAccounts.<name>.sieveScript

Per-user sieve script.

- Type: null or strings concatenated with `"\n"`
- Default: None

5.3 mailserver.certificate

5.3.1 mailserver.certificateDirectory

Scheme 2) This is the folder where the certificate will be created. The name is hardcoded to `"cert-DOMAIN.pem"` and `"key-DOMAIN.pem"` and the certificate is valid for 10 years.

- Type: path
- Default: `/var/certs`

5.3.2 mailserv.certificateDomains

Secondary domains and subdomains for which it is necessary to generate a certificate.

- Type: list of strings
- Default: []

5.3.3 mailserv.certificateFile

Scheme 1) Location of the certificate

- Type: path

5.3.4 mailserv.certificateScheme

Certificate Files. There are three options for these.

- 1) You specify locations and manually copy certificates there.
- 2) You let the server create new (self signed) certificates on the fly.
- 3) You let the server create a certificate via *Let's Encrypt*. Note that this implies that a stripped down webserver has to be started. This also implies that the FQDN must be set as an *A* record to point to the IP of the server. In particular port 80 on the server will be opened. For details on how to set up the domain records, see the guide in the readme.

- Type: one of 1, 2, 3
- Default: 2

5.4 mailserv.dkim

5.4.1 mailserv.dkimBodyCanonicalization

DKIM canonicalization algorithm for message bodies.

See <https://datatracker.ietf.org/doc/html/rfc6376/#section-3.4> for details.

- Type: one of "relaxed", "simple"
- Default: relaxed

5.4.2 mailserv.dkimHeaderCanonicalization

DKIM canonicalization algorithm for message headers.

See <https://datatracker.ietf.org/doc/html/rfc6376/#section-3.4> for details.

- Type: one of "relaxed", "simple"
- Default: relaxed

5.4.3 mailserver.dkimKeyBits

How many bits in generated DKIM keys. RFC6376 advises minimum 1024-bit keys.

If you have already deployed a key with a different number of bits than specified here, then you should use a different selector (dkimSelector). In order to get this package to generate a key with the new number of bits, you will either have to change the selector or delete the old key file.

- Type: signed integer
- Default: 1024

5.4.4 mailserver.dkimKeyDirectory

- Type: path
- Default: /var/dkim

5.4.5 mailserver.dkimSelector

- Type: string
- Default: mail

5.4.6 mailserver.dkimSigning

Whether to activate dkim signing.

- Type: boolean
- Default: True

5.5 mailserver.fullTextSearch

5.5.1 mailserver.fullTextSearch.autoIndex

Enable automatic indexing of messages as they are received or modified.

- Type: boolean
- Default: True

5.5.2 mailserver.fullTextSearch.autoIndexExclude

Mailboxes to exclude from automatic indexing.

- Type: list of strings
- Default: []

5.5.3 mailserver.fullTextSearch.enable

Whether to enable Full text search indexing with xapian. This has significant performance and disk space cost..

- Type: boolean
- Default: False

5.5.4 mailserver.fullTextSearch.enforced

Fail searches when no index is available. If set to `<literal>body</literal>`, then only body searches (as opposed to header) are affected. If set to `<literal>no</literal>`, searches may fall back to a very slow brute force search.

- Type: one of "yes", "no", "body"
- Default: no

5.5.5 mailserver.fullTextSearch.indexAttachments

Also index text-only attachments. Binary attachments are never indexed.

- Type: boolean
- Default: False

5.5.6 mailserver.fullTextSearch.maintenance.enable

Regularly optimize indices, as recommended by upstream.

- Type: boolean
- Default: True

5.5.7 mailserver.fullTextSearch.maintenance.onCalendar

When to run the maintenance job. See `systemd.time(7)` for more information about the format.

- Type: string
- Default: daily

5.5.8 mailserver.fullTextSearch.maintenance.randomizedDelaySec

Run the maintenance job not exactly at the time specified with `<literal>onCalendar</literal>`, but plus or minus this many seconds.

- Type: signed integer
- Default: 1000

5.5.9 mailserver.fullTextSearch.maxSize

Size of the largest n-gram to index.

- Type: signed integer
- Default: 20

5.5.10 mailserver.fullTextSearch.memoryLimit

Memory limit for the indexer process, in MiB. If null, leaves the default (which is rather low), and if 0, no limit.

- Type: null or signed integer
- Default: None

5.5.11 mailserver.fullTextSearch.minSize

Size of the smallest n-gram to index.

- Type: signed integer
- Default: 2

5.6 mailserver.redis

5.6.1 mailserver.redis.address

Address that rspamd should use to contact redis.

- Type: string
- Default: computed from `<option>config.services.redis.servers.rspamd.bind</option>`

5.6.2 mailserver.redis.password

Password that rspamd should use to contact redis, or null if not required.

- Type: null or string
- Default: `config.services.redis.servers.rspamd.requirePass`

5.6.3 mailserver.redis.port

Port that rspamd should use to contact redis.

- Type: 16 bit unsigned integer; between 0 and 65535 (both inclusive)
- Default: `config.services.redis.servers.rspamd.port`

5.7 mailserver.monitoring

5.7.1 mailserver.monitoring.alertAddress

The email address to send alerts to.

- Type: string

5.7.2 mailserver.monitoring.config

The configuration used for monitoring via monit. Use a mail address that you actively check and set it via 'set alert ...'.

- Type: string
- Default: see source

5.7.3 mailserver.monitoring.enable

Whether to enable monitoring via monit.

- Type: boolean
- Default: False

5.8 mailserver.backup

5.8.1 mailserver.backup.cmdPostexec

The command to be executed after each backup operation. This is wrapped in a shell script to be called by rsnapshot.

- Type: null or string
- Default: None

5.8.2 mailserver.backup.cmdPreexec

The command to be executed before each backup operation. This is wrapped in a shell script to be called by rsnapshot.

- Type: null or string
- Default: None

5.8.3 mailserver.backup.cronIntervals

Periodicity at which intervals should be run by cron. Note that the intervals also have to exist in configuration as retain options.

- Type: attribute set of strings
- Default: {'daily': '30 3 * * *', 'hourly': '0 * * * *', 'weekly': '0 5 * * 0'}

5.8.4 mailserver.backup.enable

Whether to enable backup via rsnapshot.

- Type: boolean
- Default: False

5.8.5 mailserver.backup.retain.daily

How many daily snapshots are retained.

- Type: signed integer
- Default: 7

5.8.6 mailserver.backup.retain.hourly

How many hourly snapshots are retained.

- Type: signed integer
- Default: 24

5.8.7 mailserver.backup.retain.weekly

How many weekly snapshots are retained.

- Type: signed integer
- Default: 54

5.8.8 mailserver.backup.snapshotRoot

The directory where rsnapshot stores the backup.

- Type: path
- Default: /var/rsnapshot

5.9 mailserver.borg

5.9.1 mailserver.borgbackup.cmdPostexec

The command to be executed after each backup operation. This is called after borg create completed successfully and in the same script that runs cmdPreexec, borg init and create.

- Type: null or string
- Default: None

5.9.2 mailserver.borgbackup.cmdPreexec

The command to be executed before each backup operation. This is called prior to borg init in the same script that runs borg init and create and cmdPostexec. Example:

```
export BORG_RSH="ssh -i /path/to/private/key"
```

- Type: null or string
- Default: None

5.9.3 mailserver.borgbackup.compression.auto

Leaves it to borg to determine whether an individual file should be compressed.

- Type: boolean
- Default: False

5.9.4 mailserver.borgbackup.compression.level

Denotes the level of compression used by borg. Most methods accept levels from 0 to 9 but zstd which accepts values from 1 to 22. If null the decision is left up to borg.

- Type: null or signed integer
- Default: None

5.9.5 mailserver.borgbackup.compression.method

Leaving this unset allows borg to choose. The default for borg 1.1.4 is lz4.

- Type: null or one of "none", "lz4", "zstd", "zlib", "lzma"
- Default: None

5.9.6 mailserver.borgbackup.enable

Whether to enable backup via borgbackup.

- Type: boolean
- Default: False

5.9.7 mailserver.borgbackup.encryption.method

The backup can be encrypted by choosing any other value than 'none'. When using encryption the password / passphrase must be provided in passphraseFile.

- Type: one of "none", "authenticated", "authenticated-blake2", "repokey", "keyfile", "repokey-blake2", "keyfile-blake2"
- Default: none

5.9.8 mailserver.borgbackup.encryption.passphraseFile

Path to a file containing the encryption password or passphrase.

- Type: null or path
- Default: None

5.9.9 mailserver.borgbackup.extraArgumentsForCreate

Additional arguments to add to the borg create command line e.g. '-stats'.

- Type: list of strings
- Default: []

5.9.10 mailserver.borgbackup.extraArgumentsForInit

Additional arguments to add to the borg init command line.

- Type: list of strings
- Default: ['--critical']

5.9.11 mailserver.borgbackup.group

The group borg and its launch script is run as.

- Type: string
- Default: virtualMail

5.9.12 mailserver.borgbackup.locations

The locations that are to be backed up by borg.

- Type: list of paths
- Default: ['/var/vmail']

5.9.13 mailserver.borgbackup.name

The name of the individual backups as used by borg. Certain placeholders will be replaced by borg.

- Type: string
- Default: {hostname}-{user}-{now}

5.9.14 mailserver.borgbackup.repoLocation

The location where borg saves the backups. This can be a local path or a remote location such as `user@host:/path/to/repo`. It is exported and thus available as an environment variable to `cmdPreexec` and `cmdPostexec`.

- Type: string
- Default: /var/borgbackup

5.9.15 mailserver.borgbackup.startAt

When or how often the backup should run. Must be in the format described in `systemd.time 7`.

- Type: string
- Default: hourly

5.9.16 mailserver.borgbackup.user

The user borg and its launch script is run as.

- Type: string
- Default: virtualMail

BACKUP GUIDE

First off you should have a backup of your `configuration.nix` file where you have the server config (but that is already in a git repository right?)

Next you need to backup `/var/vmail` or whatever you have specified for the option `mailDirectory`. This is where all the mails reside. Good options are a cron job with `rsync` or `scp`. But really anything works, as it is simply a folder with plenty of files in it. If your backup solution does not preserve the owner of the files don't forget to `chown` them to `virtualMail:virtualMail` if you copy them back (or whatever you specified as `vmailUserName`, and `vmailGoupName`).

Finally you can (optionally) make a backup of `/var/dkim` (or whatever you specified as `dkimKeyDirectory`). If you should lose those don't worry, new ones will be created on the fly. But you will need to repeat step B)5 and correct all the `dkim` keys.

ADD RADICALE

Configuration by @dotlambda

Starting with Radicale 3 (first introduced in NixOS 20.09) the traditional crypt passwords, as generated by *mkpasswd*, are no longer supported. Instead bcrypt passwords have to be used which can be generated using *htpasswd*.

```
{ config, pkgs, lib, ... }:

with lib;

let
  mailAccounts = config.mailserver.loginAccounts;
  htpasswd = pkgs.writeText "radicale.users" (concatStrings
    (flip mapAttrsToList mailAccounts (mail: user:
      mail + ":" + user.hashedException + "\n"
    ))
  );
in {
  services.radicale = {
    enable = true;
    config = ''
      [auth]
      type = htpasswd
      htpasswd_filename = ${htpasswd}
      htpasswd_encryption = bcrypt
    '';
  };

  services.nginx = {
    enable = true;
    virtualHosts = {
      "cal.example.com" = {
        forceSSL = true;
        enableACME = true;
        locations."/" = {
          proxyPass = "http://localhost:5232/";
          extraConfig = ''
            proxy_set_header X-Script-Name /;
            proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
            proxy_pass_header Authorization;
          '';
        };
      };
    };
  };
}
```

(continues on next page)

(continued from previous page)

```
};  
};  
};  
};  
networking.firewall.allowedTCPPorts = [ 80 443 ];  
}
```

ADD ROUNDKUBE, A WEBMAIL

The NixOS module for roundcube nearly works out of the box with SNM. By default, it sets up a nginx virtual host to serve the webmail, other web servers may require more work.

```
{ config, pkgs, lib, ... }:  
  
with lib;  
  
{  
  services.roundcube = {  
    enable = true;  
    # this is the url of the vhost, not necessarily the same as the fqdn of  
    # the mailserver  
    hostName = "webmail.example.com";  
    extraConfig = ''  
      # starttls needed for authentication, so the fqdn required to match  
      # the certificate  
      $config['smtp_server'] = "tls://${config.mailserver.fqdn}";  
      $config['smtp_user'] = "%u";  
      $config['smtp_pass'] = "%p";  
    '';  
  };  
  
  services.nginx.enable = true;  
  
  networking.firewall.allowedTCPPorts = [ 80 443 ];  
}
```


TUNE SPAM FILTERING

SNM comes with the `rspamd` spam filtering system enabled by default. Although its out-of-the-box performance is good, you can increase its efficiency by tuning its behaviour.

9.1 Auto-learning

Moving spam email to the Junk folder (and false-positives out of it) will trigger an automatic training of the Bayesian filters, improving filtering of future emails.

9.2 Train from existing folders

If you kept previous spam, you can train the filter from it. Note that the `rspamd` FAQ indicates that *you should always learn both classes with almost equal amount of messages to increase performance of the statistical engine.*

You can run the training in a root shell as follows:

```
# Path to the controller socket
export RSOCK="/var/run/rspamd/worker-controller.sock"

# Learn the Junk folder as spam
rspamc -h $RSOCK learn_spam /var/vmail/$DOMAIN/$USER/.Junk/cur/

# Learn the INBOX as ham
rspamc -h $RSOCK learn_ham /var/vmail/$DOMAIN/$USER/cur/

# Check that training was successful
rspamc -h $RSOCK stat | grep learned
```

9.3 Tune symbol weight

The `X-Spam-Result` header is automatically added to your emails, detailing the scoring decisions. The `modules` documentation details the meaning of each symbol. You can tune the weight if a symbol if needed.

```
services.rspamd.locals = {
  "groups.conf".text = ''
  symbols {
    "FORGED_RECIPIENTS" { weight = 0; }
  }
}
```

(continues on next page)

(continued from previous page)

```
}'';  
};
```

9.4 Tune action thresholds

After scoring the message, rspamd decides on an action based on configurable thresholds. By default, rspamd will tell postfix to reject any message with a score higher than 15. If you experience issues in scoring or want to stay on the safe side, you can disable this behaviour by tuning the configuration. For example:

```
services.rspamd.extraConfig = ''  
  actions {  
    reject = null; # Disable rejects, default is 15  
    add_header = 6; # Add header when reaching this score  
    greylist = 4; # Apply greylisting when reaching this score  
  }  
'';
```

9.5 Access the rspamd web UI

Rspamd comes with a [web interface](#) that displays statistics and history of past scans. **We do NOT recommend using it to change the configuration** as doing so will override values from the configuration set in the previous sections.

The UI is served on the `/var/run/rspamd/worker-controller.sock` Unix socket. Here are two ways to access it from your browser.

9.5.1 With ssh forwarding

For occasional access, the simplest way is to forward the socket to localhost and open `http://localhost:3333` in your browser.

```
ssh -L 3333:/run/rspamd/worker-controller.sock $HOSTNAME
```

9.5.2 With an nginx reverse-proxy

If you have a secured nginx reverse proxy set on the host, you can use it to expose the socket. **Keep in mind the UI is unsecured by default, you need to setup an authentication scheme**, for example with [basic auth](#):

```
services.nginx.virtualHosts.rspamd = {  
  forceSSL = true;  
  enableACME = true;  
  basicAuthFile = "/basic/auth/hashes/file";  
  serverName = "rspamd.example.com";  
  locations = {  
    "/" = {  
      proxyPass = "http://unix:/run/rspamd/worker-controller.sock:/";  
    };  
  };  
};
```

(continues on next page)

(continued from previous page)

```
};  
};
```


FULL TEXT SEARCH

By default, when your IMAP client searches for an email containing some text in its *body*, dovecot will read all your email sequentially. This is very slow and IO intensive. To speed body searches up, it is possible to *index* emails with a plugin to dovecot, `fts_xapian`.

10.1 Enabling full text search

To enable indexing for full text search here is an example configuration.

```
{
  mailserver = {
    # ...
    fullTextSearch = {
      enable = true;
      # index new email as they arrive
      autoIndex = true;
      # this only applies to plain text attachments, binary attachments are never indexed
      indexAttachments = true;
      enforced = "body";
    };
  };
}
```

The `enforced` parameter tells dovecot to fail any body search query that cannot use an index. This prevents dovecot to fall back to the IO-intensive brute force search.

If you set `autoIndex` to `false`, indices will be created when the IMAP client issues a search query, so latency will be high.

10.2 Resource requirements

Indices created by the full text search feature can take more disk space than the emails themselves. By default, they are kept in the emails location. When enabling the full text search feature, it is recommended to move indices in a different location, such as `(/var/lib/dovecot/indices)` by using the option `mailserver.indexDir`.

Warning: When the value of the `indexDir` option is changed, all dovecot indices needs to be recreated: clients would need to resynchronize.

Indexation itself is rather resource intensive, in CPU, and for emails with large headers, in memory as well. Initial indexation of existing emails can take hours. If the indexer worker is killed or segfaults during indexation, it can be that it tried to allocate more memory than allowed. You can increase the memory limit by eg `mailserver.fullTextSearch.memoryLimit = 2000` (in MiB).

10.3 Mitigating resources requirements

You can:

- disable indexation of attachments `mailserver.fullTextSearch.indexAttachments = false`
- reduce the size of ngrams to be indexed `mailserver.fullTextSearch.minSize` and `maxSize`
- disable automatic indexation for some folders with `mailserver.fullTextSearch.autoIndexExclude`. Folders can be specified by name ("Trash"), by special use ("\\Junk") or with a wildcard.

NIX FLAKES

If you're using `flakes`, you can use the following minimal `flake.nix` as an example:

```
{
  description = "NixOS configuration";

  inputs.simple-nixos-mailserver.url = "gitlab:simple-nixos-mailserver/nixos-mailserver/
↪nixos-20.09";

  outputs = { self, nixpkgs, simple-nixos-mailserver }: {
    nixosConfigurations = {
      hostname = nixpkgs.lib.nixosSystem {
        system = "x86_64-linux";
        modules = [
          simple-nixos-mailserver.nixosModule
            {
              mailserver = {
                enable = true;
                # ...
              };
            }
        ];
      };
    };
  };
}
```


INDICES AND TABLES

- genindex
- modindex
- search