NixOS Mailserver

NixOS Mailserver Contributors

Oct 14, 2023

CONTENTS


SNM
Simple Nixos Mailserver

CHAPTER

ONE

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 NixOS Mailserver options documentation), these should be the most common ones.

```
{ config, pkgs, ... }: {
  imports = [
    (builtins.fetchTarball {
      # Pick a release version you are interested in and set its hash, e.g.
      url = "https://gitlab.com/simple-nixos-mailserver/nixos-mailserver/-/archive/nixos-
↪23.05/nixos-mailserver-nixos-23.05.tar.gz";
      # To get the sha256 of the nixos-mailserver tarball, we can use the nix-prefetch-
↪url command:
      # release="nixos-23.05"; nix-prefetch-url "https://gitlab.com/simple-nixos-
↪mailserver/nixos-mailserver/-/archive/${release}/nixos-mailserver-${release}.tar.gz" --
↪unpack
      sha256 = "0000000000000000000000000000000000000000000000000000";
    })
  ];

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

    # A list of all login accounts. To create the password hashes, use
    # nix-shell -p mkpasswd --run 'mkpasswd -sm bcrypt'
    loginAccounts = {
      "user1@example.com" = {
        hashedPasswordFile = "/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 = "acme-nginx";
  };
  security.acme.acceptTerms = true;
  security.acme.defaults.email = "security@example.com";
}
```

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`.

Warning: We don't recommend setting up a mail server if you are not able to set a reverse DNS on your public IP because sent emails would be mostly marked as spam. Note that many residential ISP providers don't allow you to set a reverse DNS entry.

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
example.com	MX	10	mail.example.com

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
example.com	10800	TXT	*v=spf1 a:mail.example.com -all*

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
mail._domainkey.example.com	10800	TXT	v=DKIM1; p=<really-long-key>

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
_dmarc.example.com	10800	TXT	v=DMARC1; p=none

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!

CHAPTER

TWO

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.

You can then run the testsuite via

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

Since Nix doesn't garantee your machine have enough resources to run all test VMs in parallel, some tests can fail. You would then haev 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 (except option documentation which is in CommonMark), built with Sphinx and published by Read the Docs.

For the syntax, see the RST/Sphinx primer.

To build the documentation, you need to enable Nix Flakes.

```
$ nix build .#documentation
$ xdg-open result/index.html
```

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
```

CHAPTER

THREE

FAQ

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.

CHAPTER

FOUR

RELEASE NOTES

4.1 NixOS 23.05

- Existing ACME certificates can be reused without configuring NGINX
- Certificate scheme is no longer a number, but a meaningful string instead

4.2 NixOS 22.11

- Allow Rspamd to send DMARC reporting (merge request)

4.3 NixOS 22.05

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

4.4 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`)

4.5 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

4.6 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

CHAPTER

FIVE

MAILSERVER OPTIONS

5.1 mailserver

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`

mailserver.domains

The domains that this mail server serves.

- type: `list of string`
- default: `[ ]`
- example:

```
[
  "example.com"
]
```

mailserver.enable

Whether to enable nixos-mailserver.

- type: `boolean`
- default: `false`
- example: `true`

mailserver.enableImap

Whether to enable IMAP with STARTTLS on port 143.

- type: `boolean`
- default: `true`

mailserver.enableImapSsl

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

- type: `boolean`
- default: `true`

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`

mailserver.enablePop3

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

- type: `boolean`
- default: `false`

mailserver.enablePop3Ssl

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

- type: `boolean`
- default: `false`

mailserver.enableSubmission

Whether to enable SMTP with STARTTLS on port 587.

- type: `boolean`
- default: `true`

mailserver.enableSubmissionSsl

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

- type: `boolean`
- default: `true`

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 Account)))`
- default: `{ }`
- example:

```
{
  "abuse@example.com" = "user1@example.com";
  "info@example.com" = "user1@example.com";
  "multi@example.com" = [
    "user1@example.com"
    "user2@example.com"
  ];
  "postmaster@example.com" = "user1@example.com";
}
```

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 *mailserver.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 string) or string)
- default: { }
- example:

```
{
  "user@example.com" = "user@elsewhere.com";
}
```

mailserver.fqdn

The fully qualified domain name of the mail server.

- type: string
- example: "mx.example.com"

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: "."

mailserver.indexDir

Folder to store search indices. If null, indices are stored along with email, which could not necessarily be desirable, especially when *mailserver.fullTextSearch.enable* is true 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: null
- example: "/var/lib/dovecot/indices"

mailserver.keyFile

(*mailserver.certificateScheme* == manual)

Location of the key file.

- type: path
- example: "/root/mail-server.key"

mailserver.lmtpSaveToDetailMailbox

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"

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

mailserver.mailDirectory

Where to store the mail.

- type: path
- default: "/var/vmail"

mailserver.mailboxes

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

- type: unspecified value
- default:

```
{
  Drafts = {
    auto = "subscribe";
    specialUse = "Drafts";
  };
  Junk = {
    auto = "subscribe";
    specialUse = "Junk";
  };
  Sent = {
    auto = "subscribe";
    specialUse = "Sent";
  };
  Trash = {
    auto = "no";
    specialUse = "Trash";
  };
}
```

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

mailserver.messageSizeLimit

Message size limit enforced by Postfix.

- type: signed integer
- default: 20971520

- example: `52428800`

mailserver.openFirewall

Automatically open ports in the firewall.

- type: `boolean`
- default: `true`

mailserver.policydSPFExtraConfig

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

- type: `strings concatenated with "\n"`
- default: `""`
- example:

```
''
  skip_addresses = 127.0.0.0/8,::ffff:127.0.0.0/104,::1
''
```

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`
- example: `true`

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"`

mailserver.recipientDelimiter

Configure the recipient delimiter.

- type: `string`
- default: `"+"`

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 string`
- default: `[ ]`
- example:

```
[
  "sales@example.com"
  "info@example.com"
]
```

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 string`
- default: `[ ]`
- example:

```
[
  "@example.com"
  "spammer@example.net"
]
```

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`

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 *mailserver.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: *mailserver.fqdn*
- example: `"myserver.example.com"`

mailserver.sieveDirectory

Where to store the sieve scripts.

- type: `path`
- default: `"/var/sieve"`

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`

mailserver.virusScanning

Whether to activate virus scanning. Note that virus scanning is *very* expensive memory wise.

- type: `boolean`
- default: `false`

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"`

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`

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.loginAccounts

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 `mkpasswd` as follows

```
nix-shell -p mkpasswd --run 'mkpasswd -sm bcrypt'
```

- type: `attribute set of (submodule)`
- default: `{ }`
- example:

```
{
  user1 = {
    hashedPassword = "$6$evQJs5CFQyPAW09S$Cn99Y8.QjZ2IBnSu4qf1vBxDRWkaIZWOtmu1Ddsm3.
↪H3CFpeVc0JU4llIq8HQXgeatvYhh5O33eWG3TSpjzu6/";
  };
  user2 = {
    hashedPassword = "$6$oE0ZNv2n7Vk9gOf$9xcZWCCLGdMflIfuA0vR1Q1Xblw6RZqPrP94mEit2/
↪81/7AKj2bqUai5yPyWE.QYPyv6wLMHZvjw3Rlg7yTCD/";
  };
}
```

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 string
- default: [ ]
- example:

```
[
  "abuse@example.com"
  "postmaster@example.com"
]
```

mailserver.loginAccounts.<name>.aliasesRegexp

Same as mailserver.aliases but using PCRE (Perl compatible regex).

- type: list of string
- default: [ ]
- example:

```
[
  "/^tom\\..*@domain\\.com$/"
]
```

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 value "example.com" (singular enum)
- default: [ ]
- example:

```
[
  "example.com"
  "example2.com"
]
```

mailserver.loginAccounts.<name>.hashedPassword

The user's hashed password. Use mkpasswd as follows

```
nix-shell -p mkpasswd --run 'mkpasswd -sm bcrypt'
```

Warning: this is stored in plaintext in the Nix store! Use *mailserver.loginAccounts.<name>.hashedPasswordFile* instead.

- type: null or string
- default: null
- example: "6evQJs5CFQyPAW09S$Cn99Y8.QjZ2IBnSu4qf1vBxDRWkaIZWOtmu1Ddsm3.H3CFpeVc0JU4llIq8HQXgeatvYhh5O33eWG3TSpjzu6/"

mailserver.loginAccounts.<name>.hashedPasswordFile

A file containing the user's hashed password. Use `mkpasswd` as follows

```
nix-shell -p mkpasswd --run 'mkpasswd -sm bcrypt'
```

- type: `null or path`
- default: `null`
- example: `"/run/keys/user1-passwordhash"`

mailserver.loginAccounts.<name>.name

Username

- type: `string`
- example: `"user1@example.com"`

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: `null`
- example: `"2G"`

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`

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."`

mailserver.loginAccounts.<name>.sieveScript

Per-user sieve script.

- type: `null or strings concatenated with "\n"`
- default: `null`
- example:

```
''
  require ["fileinto", "mailbox"];

  if address :is "from" "gitlab@mg.gitlab.com" {
    fileinto :create "GitLab";
    stop;
  }
```

(continues on next page)

(continued from previous page)

```
  # This must be the last rule, it will check if list-id is set, and
  # file the message into the Lists folder for further investigation
  elsif header :matches "list-id" "<?*>" {
    fileinto :create "Lists";
    stop;
  }
''
```

5.3 mailserver.certificate

mailserver.certificateDirectory

(*mailserver.certificateScheme* == selfsigned)

This is the folder where the self-signed 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"

mailserver.certificateDomains

(*mailserver.certificateScheme* == acme-nginx)

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

- type: list of string
- default: [ ]
- example:

```
[
  "imap.example.com"
  "pop3.example.com"
]
```

mailserver.certificateFile

(*mailserver.certificateScheme* == manual)

Location of the certificate.

- type: path
- example: "/root/mail-server.crt"

mailserver.certificateScheme

The scheme to use for managing TLS certificates:

1. manual: you specify locations via *mailserver.certificateFile* and *mailserver.keyFile* and manually copy certificates there.
2. selfsigned: you let the server create new (self-signed) certificates on the fly.
3. acme-nginx: you let the server request certificates from Let's Encrypt via NixOS' ACME module. By default, this will set up a stripped-down Nginx server for *mailserver.fqdn* and open port 80. For this to work, the FQDN must be properly configured to point to your server (see the *setup guide* for more information).

4. `acme`: you already have an ACME certificate set up (for example, you're already running a TLS-enabled Nginx server on the FQDN). This is better than `manual` because the appropriate services will be reloaded when the certificate is renewed.

- type: `(one of "manual", "selfsigned", "acme-nginx", "acme") or (one of 1, 2, 3) convertible to it`
- default: `"selfsigned"`

5.4 mailserver.dkim

mailserver.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"`

mailserver.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"`

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 (*mailserver.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`

mailserver.dkimKeyDirectory

The DKIM directory.

- type: `path`
- default: `"/var/dkim"`

mailserver.dkimSelector

The DKIM selector.

- type: `string`
- default: `"mail"`

mailserver.dkimSigning

Whether to activate dkim signing.

- type: `boolean`
- default: `true`

5.5 mailserver.dmarcReporting

mailserver.dmarcReporting.domain

The domain from which outgoing DMARC reports are served.

- type: `value "example.com" (singular enum)`
- example: `"example.com"`

mailserver.dmarcReporting.email

The email address used for outgoing DMARC reports. Read-only.

- type: `string`
- default: `"${localpart}@${domain}"`

mailserver.dmarcReporting.enable

Whether to send out aggregated, daily DMARC reports in response to incoming mail, when the sender domain defines a DMARC policy including the RUA tag.

This is helpful for the mail ecosystem, because it allows third parties to get notified about SPF/DKIM violations originating from their sender domains.

See https://rspamd.com/doc/modules/dmarc.html#reporting

- type: `boolean`
- default: `false`

mailserver.dmarcReporting.fromName

The sender name for DMARC reports. Defaults to the organization name.

- type: `string`
- default: *mailserver.dmarcReporting.organizationName*

mailserver.dmarcReporting.localpart

The local part of the email address used for outgoing DMARC reports.

- type: `string`
- default: `"dmarc-noreply"`
- example: `"dmarc-report"`

mailserver.dmarcReporting.organizationName

The name of your organization used in the `org_name` attribute in DMARC reports.

- type: `string`
- example: `"ACME Corp."`

5.6 mailserver.fullTextSearch

mailserver.fullTextSearch.autoIndex

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

- type: boolean
- default: true

mailserver.fullTextSearch.autoIndexExclude

Mailboxes to exclude from automatic indexing.

- type: list of string
- default: [ ]
- example:

```
[
  "\\Trash"
  "SomeFolder"
  "Other/*"
]
```

mailserver.fullTextSearch.enable

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

- type: boolean
- default: false
- example: true

mailserver.fullTextSearch.enforced

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

- type: one of "yes", "no", "body"
- default: "no"

mailserver.fullTextSearch.indexAttachments

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

- type: boolean
- default: false

mailserver.fullTextSearch.maintenance.enable

Regularly optmize indices, as recommended by upstream.

- type: boolean
- default: true

mailserver.fullTextSearch.maintenance.onCalendar

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

- type: string
- default: "daily"

mailserver.fullTextSearch.maintenance.randomizedDelaySec

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

- type: `signed integer`
- default: `1000`

mailserver.fullTextSearch.maxSize

Size of the largest n-gram to index.

- type: `signed integer`
- default: `20`

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: `null`
- example: `2000`

mailserver.fullTextSearch.minSize

Size of the smallest n-gram to index.

- type: `signed integer`
- default: `2`

5.7 mailserver.redis

mailserver.redis.address

Address that rspamd should use to contact redis.

- type: `string`
- default: computed from `config.services.redis.servers.rspamd.bind`

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`

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`

5.8 mailserver.ldap

mailserver.ldap.bind.dn

Distinguished name used by the mail server to do lookups against the LDAP servers.

- type: `string`
- example: `"cn=mail,ou=accounts,dc=example,dc=com"`

mailserver.ldap.bind.passwordFile

A file containing the password required to authenticate against the LDAP servers.

- type: `string`
- example: `"/run/my-secret"`

mailserver.ldap.dovecot.passAttrs

LDAP attributes to be retrieved during passdb lookups.

See the pass_attrs reference at https://doc.dovecot.org/configuration_manual/authentication/ldap_settings_auth/#pass-attrs in the Dovecot manual.

- type: `string`
- default: `"userPassword=password"`

mailserver.ldap.dovecot.passFilter

Filter for password lookups in Dovecot.

See the pass_filter reference for https://doc.dovecot.org/configuration_manual/authentication/ldap_settings_auth/#pass-filter in the Dovecot manual.

- type: `null or string`
- default: `"mail=%u"`
- example: `"(&(objectClass=inetOrgPerson)(mail=%u))"`

mailserver.ldap.dovecot.userAttrs

LDAP attributes to be retrieved during userdb lookups.

See the users_attrs reference at https://doc.dovecot.org/configuration_manual/authentication/ldap_settings_auth/#user-attrs in the Dovecot manual.

- type: `string`
- default: `""`

mailserver.ldap.dovecot.userFilter

Filter for user lookups in Dovecot.

See the user_filter reference at https://doc.dovecot.org/configuration_manual/authentication/ldap_settings_auth/#user-filter in the Dovecot manual.

- type: `string`
- default: `"mail=%u"`
- example: `"(&(objectClass=inetOrgPerson)(mail=%u))"`

mailserver.ldap.enable

Whether to enable LDAP support.

- type: `boolean`

- default: `false`
- example: `true`

mailserver.ldap.postfix.filter

LDAP filter used to search for an account by mail, where `%s` is a substitute for the address in question.

- type: `string`
- default: `"mail=%s"`
- example: `"(&(objectClass=inetOrgPerson)(mail=%s))"`

mailserver.ldap.postfix.mailAttribute

The LDAP attribute holding mail addresses for a user.

- type: `string`
- default: `"mail"`

mailserver.ldap.postfix.uidAttribute

The LDAP attribute referencing the account name for a user.

- type: `string`
- default: `"mail"`
- example: `"uid"`

mailserver.ldap.searchBase

Base DN at below which to search for users accounts.

- type: `string`
- example: `"ou=people,ou=accounts,dc=example,dc=com"`

mailserver.ldap.searchScope

Search scope below which users accounts are looked for.

- type: `one of "sub", "base", "one"`
- default: `"sub"`

mailserver.ldap.startTls

Whether to enable StartTLS upon connection to the server.

- type: `boolean`
- default: `false`

mailserver.ldap.tlsCAFile

Certifificate trust anchors used to verify the LDAP server certificate.

- type: `path`
- default: see source

mailserver.ldap.uris

URIs where your LDAP server can be reached

- type: `list of string`
- example:

```
[
  "ldaps://ldap1.example.com"
  "ldaps://ldap2.example.com"
]
```

5.9 mailserver.monitoring

mailserver.monitoring.alertAddress

The email address to send alerts to.

- type: `string`

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

mailserver.monitoring.enable

Whether to enable monitoring via monit.

- type: `boolean`
- default: `false`
- example: `true`

5.10 mailserver.backup

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: `null`

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: `null`

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 string`
- default:

```
{
  daily = "30  3  *  *  *";
  hourly = " 0  *  *  *  *";
  weekly = " 0  5  *  *  0";
}
```

mailserver.backup.enable

Whether to enable backup via rsnapshot.

- type: `boolean`
- default: `false`
- example: `true`

mailserver.backup.retain.daily

How many daily snapshots are retained.

- type: `signed integer`
- default: `7`

mailserver.backup.retain.hourly

How many hourly snapshots are retained.

- type: `signed integer`
- default: `24`

mailserver.backup.retain.weekly

How many weekly snapshots are retained.

- type: `signed integer`
- default: `54`

mailserver.backup.snapshotRoot

The directory where rsnapshot stores the backup.

- type: `path`
- default: `"/var/rsnapshot"`

5.11 mailserver.borgbackup

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: `null`

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`.

- type: `null or string`
- default: `null`

- example:

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

mailserver.borgbackup.compression.auto

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

- type: boolean
- default: false

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: null

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: null

mailserver.borgbackup.enable

Whether to enable backup via borgbackup.

- type: boolean
- default: false
- example: true

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"

mailserver.borgbackup.encryption.passphraseFile

Path to a file containing the encryption password or passphrase.

- type: null or path
- default: null

mailserver.borgbackup.extraArgumentsForCreate

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

- type: list of string
- default: [ ]

mailserver.borgbackup.extraArgumentsForInit

Additional arguments to add to the borg init command line.

- type: list of string

- default:

```
[
  "--critical"
]
```

mailserver.borgbackup.group

The group borg and its launch script is run as.

- type: `string`
- default: `"virtualMail"`

mailserver.borgbackup.locations

The locations that are to be backed up by borg.

- type: `list of path`
- default: `[ config.mailserver.mailDirectory ]`

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}"`

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 *mailserver.borgbackup.cmdPreexec* and *mailserver.borgbackup.cmdPostexec*.

- type: `string`
- default: `"/var/borgbackup"`

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"`

mailserver.borgbackup.user

The user borg and its launch script is run as.

- type: `string`
- default: `"virtualMail"`

CHAPTER

SIX

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.

CHAPTER

SEVEN

ADD RADICALE

Configuration by @dotlambda

Starting with Radicale 3 (first introduced in NixOS 20.09) the traditional crypt passwords are no longer supported. Instead bcrypt passwords have to be used. These can still be generated using *mkpasswd -m bcrypt*.

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

with lib;

let
  mailAccounts = config.mailserver.loginAccounts;
  htpasswd = pkgs.writeText "radicale.users" (concatStrings
    (flip mapAttrsToList mailAccounts (mail: user:
      mail + ":" + user.hashedPassword + "\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 ];
}
```

CHAPTER

EIGHT

ADD ROUNDCUBE, 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 ];
}
```

CHAPTER

NINE

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-Spamd-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 exemple 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)

```
  };
};
```

CHAPTER

TEN

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 resouces 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 attachements `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.

CHAPTER

ELEVEN

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;
              # ...
            };
          }
        ];
      };
    };
  };
}
```

CHAPTER

TWELVE

AUTODISCOVERY

RFC6186 allows supporting email clients to automatically discover SMTP / IMAP addresses of the mailserver. For that, the following records are required:

Record	TTL	Type	Priority	Weight	Port	Value
_submission._tcp	3600	SRV	5	0	587	mail.example.com.
_submissions._tcp	3600	SRV	5	0	465	mail.example.com.
_imap._tcp	3600	SRV	5	0	143	mail.example.com.
_imaps._tcp	3600	SRV	5	0	993	mail.example.com.

Please note that only a few MUAs currently implement this. For vendor-specific discovery mechanisms automx can be used instead.

CHAPTER

THIRTEEN

LDAP SUPPORT

It is possible to manage mail user accounts with LDAP rather than with the option loginAccounts.

All related LDAP options are described in the LDAP options section and the LDAP test provides a getting started example.

Note: The LDAP support can not be enabled if some accounts are also defined with `mailserver.loginAccounts`.

CHAPTER

FOURTEEN

INDICES AND TABLES

- genindex
- modindex
- search

INDEX

C

M