Almizan

Configuration système persistante

`/etc/rc.conf` et `rc.conf.d/`

La configuation des scripts système et service se fait en principe dans le fichier `/etc/rc.conf`.

On peut diviser la configuration par script et service rc en utilisant des fichiers dans le dossier `/etc/rc.conf.d/`. Les fichiers de config doivent porter le nom du script rc.

Pour savoir quel script utilise telle variable, on peux faire la commande `grep -l "NOM_VARIABLE" /etc/rc.d/*`. Ex :

```
root@almizan /e/rc.conf.d# grep -l "ipv6_defaultrouter" /etc/rc.d/*
/etc/rc.d/routing
```

La configuration basique :

Les fichiers suivant sont dans `/etc/rc.conf.d`

- **/hostname** : defini le nom de la machine
- **/network** : permet de faire la configuration ifconfig
- **/routing**: defini les routes

Réseau

Les mail

Trois logiciels

- Dovecot
- opensmtpd
- dkimsign

Les mails sont dans `/var/mailbox`

Pour ajouter un nouvel identifiant, les fichiers sont dans `/etc/mailbox/`

Pour générer le hash du mot de passe: `doveadm pw -s SHA512-CRYPT`

Opensmtpd

/usr/local/etc/mail/smtpd.conf

```
pki mail.ppsfleet.navy cert "/usr/local/etc/mail/certs/mail.ppsfleet.navy.crt"
pki mail.ppsfleet.navy key "/usr/local/etc/mail/certs/mail.ppsfleet.navy.key"

# --- Filtre rdns --- # Reject if no reverse dns

filter check_rdns phase connect match !rdns \
    disconnect "550 no rDNS"

# --- Filtre fcrdns --- # Reject if no "Forward-confirmed_reverse_DNS" dns(reverse(domain)) =
domain

filter check_fcrdns phase connect match !fcrdns \
    disconnect "550 no FCrDNS"

filter dkimsign proc-exec "/usr/local/libexec/opensmtpd/filter-dkimsign -d ppsfleet.navy -s
mail -k /usr/local/etc/mail/dkim/ppsfleet.navy.key" user _smtpd group _smtpd

table aliases file:/etc/mailbox/aliases.txt
table domains file:/etc/mailbox/domains.txt
table password file:/etc/mailbox/passwd.txt
```

```
# --- mail entrant --- #
listen on vtnet0 port 25 tls pki mail.ppsfleet.navy filter { check_rdns, check_fcrdns}

# --- mail sortant --- #
listen on vtnet0 port submission tls-require pki mail.ppsfleet.navy auth <password> filter {
dkimsign }

action "reception" lmtp "/var/run/dovecot/lmtp" rcpt-to virtual <aliases>

action "envoi" relay helo almizan.ppsfleet.navy

# -- entrant --
match from any for domain <domains> action "reception"

# -- sortant --
# Demande authentification si "any auth"
match from any auth for any action "envoi"
match from local for any action "envoi"
```

Dovecot

/usr/local/etc/mail/dovecot/dovecot.conf

```
ssl_cert = </var/db/caddy/data/caddy/certificates/acme-v02.api.letsencrypt.org-
directory/mail.ppsfleet.navy/mail.ppsfleet.navy.crt
ssl_key = </var/db/caddy/data/caddy/certificates/acme-v02.api.letsencrypt.org-
directory/mail.ppsfleet.navy/mail.ppsfleet.navy.key

ssl_min_protocol = TLSv1.2
ssl_prefer_server_ciphers = yes
ssl = required
disable_plaintext_auth = yes

protocols = lmtp imap
# sieve



service lmtp {
```

```
  unix_listener lmtp {
    user  = vmail
    group = vmail
  }

}

protocol lmtp {
  mail_plugins = $mail_plugins sieve
}

service managesieve-login {
  inet_listener sieve {
    port = 4190
  }

  #inet_listener sieve_deprecated {
  #  port = 2000
  #}

  # Number of connections to handle before starting a new process. Typically
  # the only useful values are 0 (unlimited) or 1. 1 is more secure, but 0
  # is faster. <doc/wiki/LoginProcess.txt>
  service_count = 1

  # Number of processes to always keep waiting for more connections.
  process_min_avail = 0

  # If you set service_count=0, you probably need to grow this.
  vsz_limit = 64M
}

service imap-login {
        inet_listener imap {
                port = 143
        }
        inet_listener imaps {
                port = 993
        }
}
```

```
#service auth {
# SASL
#  unix_listener auth-client {
#    mode = 0660
#    user = mail
#    group = mail
#  }
#}

passdb {
        driver = passwd-file
        args = scheme=SHA512-CRYPT /etc/mailbox/passwd.txt
}



userdb {
    args   = uid=vmail gid=vmail home=/var/mailbox/%d/%n
    driver = static
}




namespace inbox {
  # Namespace type: private, shared or public
  type = private

  # Hierarchy separator to use. You should use the same separator for all
  # namespaces or some clients get confused. '/' is usually a good one.
  # The default however depends on the underlying mail storage format.
  separator = '/'

  inbox = yes

}

mail_location = maildir:/var/mailbox/%d/%n
```

Jails

Généralitées

https://docs.freebsd.org/en/books/handbook/jails/

Les jails sont installées dans `/usr/local/jails`.

`/usr/local/jails/containers/` - contient les instances des jails

`/usr/local/jails/media/` - contient les archives d'installation de freebsd

`/usr/local/jails/templates` - contient des images zfs sur lesquels baser ses jails. `base` est une image basique sans rien.

Créer une jail classique à la main (Fat Jail)

Installer et mettre à jour la jail

```
mkdir /usr/local/jails/conainers/<nom de la jail>
tar -xf /usr/local/jails/media/15.0-RELEASE-base.txz -C /usr/local/jails/containers/<nom de la jail> --unlink

cp /etc/resolv.conf /usr/local/jails/containers/classic/etc/resolv.conf
cp /etc/localtime /usr/local/jails/containers/classic/etc/localtime

freebsd-update -b /usr/local/jails/containers/classic/ fetch install
```

Créer le fichier de conf de la jail

dans `/etc/jail.conf.d/<nom de la jail>`

Utiliser les jail

Lister les jails

```
jls
```

Installer des packages dans la jail

```
pkg -j <nom de la jail> install ..
```

Executer une commande dans une jail

```
jexec -l <nom de la jail> commande sans guillemets
```