

Scaling antispam mail servers with Puppet

Giovanni Bechis
<g.bechis@snb.it>
@g_bechis

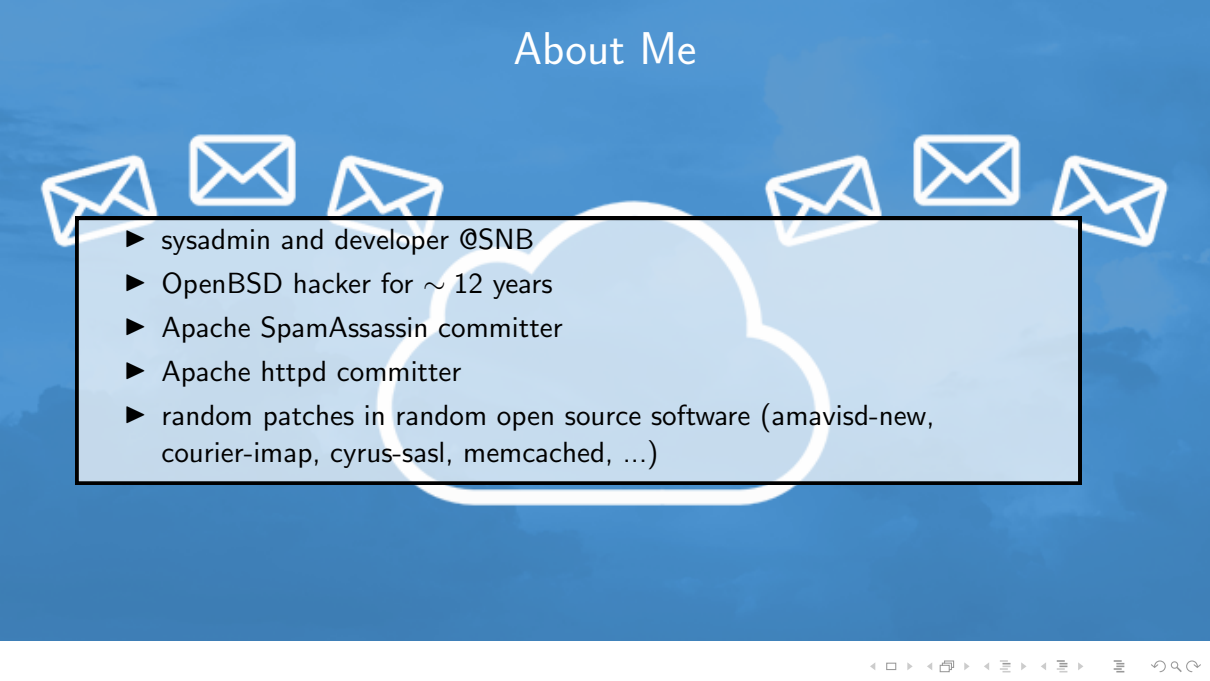
PuppetCamp 2020, Netherlands

SNB

Information Technology
& Web Solutions



About Me

- 
- ▶ sysadmin and developer @SNB
 - ▶ OpenBSD hacker for ~ 12 years
 - ▶ Apache SpamAssassin committer
 - ▶ Apache httpd committer
 - ▶ random patches in random open source software (amavisd-new, courier-imap, cyrus-sasl, memcached, ...)

"mail" software managed by Puppet

- 
- ▶ Unbound
 - ▶ Clamav
 - ▶ Postfix
 - ▶ OpenBSD spamd
 - ▶ Redis
 - ▶ Apache SpamAssassin
 - ▶ MimeDefang
 - ▶ Filebeat
 - ▶ Zabbix

Dns settings

```
class { 'resolv_conf':  
  nameservers => ['127.0.0.1', '172.16.99.1'],  
}
```

```
class { "unbound":  
  interface => ["127.0.0.1", ":::1"],  
  access    => ["127.0.0.0/8", ":::1"],  
}  
unbound::stub { "rb10.example.lan":  
  address => ["172.16.99.137@53"],  
  insecure => true,  
}  
unbound::stub { "rb11.example.lan":  
  address => ["172.16.99.137@53"],  
  insecure => true,  
}
```

Puppet Facts



Some facts are interesting to scale in a right way all involved softwares

- ▶ "os.name" to have different configuration for different operating systems
- ▶ "processors.count" and "memory.system.total_bytes" to correctly configure Clamav and Apache SpamAssassin

Puppet Facts and Erb templates

```
# Clamav template

# Maximum number of threads running at the same time.
# Default: 10
#MaxThreads 20
<% if @processors['count'] != 1 -%>
MaxThreads 20
<% end -%>

# Path to a local socket file the daemon will listen on.
# Default: disabled (must be specified by a user)
#LocalSocket /tmp/clamd.socket
<% if @os['name'] == 'OpenBSD' -%>
LocalSocket /tmp/clamd
<% end -%>
```

Puppet Facts and Erb templates



```
# Mail::SpamAssassin::Plugin::ResourceLimits  
  
<% if @memory['system']['total_bytes'] < 4142006272 -%>  
resource_limit_mem 536870912  
<% end -%>
```

Puppet Facts and Erb templates

```
redis::params { 'redis':  
  bind => ['127.0.0.1', '172.16.99.134'],  
  masterauth => 'XXX',  
  slaveof => '172.16.99.133 6379',  
}
```

```
bind <%= @bind.join(' ') %>  
  
<% if @masterauth -%>masterauth <%= @masterauth %><% end -%>  
  
<% if @slaveof -%>slaveof <%= @slaveof %><% end -%>
```

Config files deploy

Config files deploy



Puppet server does a checkout of our git repo and pulls updated configuration files.

Most of the configuration files are the same for all servers, lot of modules we are using just copy the correct files in the correct location.

OpenBSD spamd(8) configuration

```
exec {  
  'pf_test' :  
    command => '/sbin/pfctl -nf /etc/pf.conf',  
    refreshonly => true,  
    notify => Exec[pf_load] ;  
  
  'pf_load' :  
    command => '/sbin/pfctl -f /etc/pf.conf',  
    refreshonly => true ;  
}  
file { '/etc/pf-spamd-white-local':  
  ensure => file,  
  path   => '/etc/pf-spamd-white-local',  
  owner  => 'root',  
  mode   => '0640',  
  source => 'puppet:///modules/spamd/pf-spamd-white-local',  
  notify => Exec[pf_test];  
}
```

SpamAssassin rules configuration

```
file { '/etc/mail/spamassassin/20_sought.cf':  
  ensure => file,  
  path    => '/etc/mail/spamassassin/20_sought.cf',  
  owner   => 'root',  
  mode    => '0640',  
  source  => 'puppet:///modules/spamassassin/20_sought.cf',  
  notify => [  
    Service['spamassassin'],  
  ],  
}  
file { '/etc/mail/spamassassin/sendinblue_id.txt':  
  ensure => file,  
  path    => '/etc/mail/spamassassin/sendinblue_id.txt',  
  owner   => 'root',  
  mode    => '0640',  
  source  => 'puppet:///modules/spamassassin/sendinblue_id.txt',  
  notify => [  
    Service['spamassassin'],  
  ],  
}
```

Mimedefang configuration and deploy

Mimedefang deploy



Mimedefang has both code and configuration files that need to be deployed.

Mimedefang has several dependencies like SpamAssassin and Clamav, the same dependency tree is respected on Puppet code.

Mimedefang configuration

```
define mimedefang::params (
  $user_scores_dsn = '',
  $user_scores_sql_username = '',
  $user_scores_sql_password = '',
){
  include ::spamassassin
  package { 'mimedefang':
    ensure => 'present'
  }
  file { '/etc/mail/spamassassin/mimedefang.cf':
    ensure => file,
    content => template('mimedefang/mimedefang.cf.erb'),
    path    => '/etc/mail/spamassassin/mimedefang.cf',
    owner   => 'root',
    mode    => '0640',
    notify => [
      Service['spamassassin'],
    ],
  }
}
```

Mimedefang configuration

```
mimedefang::params { 'mimedefang':  
  user_scores_dsn => 'DBI:mysql:database:localhost',  
  user_scores_sql_username => 'user',  
  user_scores_sql_password => 'password',  
  md_ft_VirusClamd => 0,  
  md_ft_Spamc => 1,  
  md_ft_SpamAssassin => 0,  
  md_ft_archive_zip => 1,  
  md_ft_unrar => 1,  
}
```

```
$Features{"Spamc_max_size"} = <%= @md_ft_Spamc_max_size %>;  
$Features{"SpamAssassin"} = <%= @md_ft_SpamAssassin %>;  
$Features{"unrar"} = <%= @md_ft_unrar %>;  
  
$cemail = "<%= scope().call_function('hiera','md_cemail'); -%>";  
}
```

Filebeat and Elk configuration

```
class { 'filebeat':  
  outputs => {  
    'logstash' => {  
      'hosts' => [  
        '172.16.99.211:5001',  
      ],  
    },  
  },  
  fields_under_root => true,  
}  
  
filebeat::input { 'postfix':  
  paths => [  
    '/var/log/maillog',  
    '/var/log/spamd',  
  ],  
  doc_type => 'postfix',  
  fields => { "logtype" => "postfix" },  
}
```

Zabbix configuration

```
define zabbix::params (
  $zabbix_server = 'zabbix.example.com',
  $zabbix_port = '10050'
){
  case $facts['os']['name'] {
    'OpenBSD': {
      if ($facts['os']['release']['major'] == '6') {
        file { '/etc/zabbix/zabbix_agent.conf':
          ensure => present,
          content => template('zabbix/zabbix_agent-openbsd.conf.erb'),
          owner   => 'root',
          group   => 'wheel',
          notify  => Service['zabbix_agentd'],
        }
      }
    }
  }
}
```

Questions ?

