PUFFY PRESENTS VIDEO CONFERENCING / JITSI ON OPENBSD

Philipp Bühler, sysfive.com GmbH

EuroBSDCon - Vienna - 18. September 2022

sysfive
com securing performance

WHAT'S AHEAD

RIDDLES JITSI + OPENBSD

COMPONENTS / TERMINOLOGY

INSTALL OVERVIEW

CONFIGURATION OPENBSD / JITSI

WATCH OUT / PITFALLS

PRAYERS AND DEMO

STATUS + OUTLOOK

https://is.gd/ontSw3

- smtp: pb@sysfive.com
- bird: @pb_double ; matrix/irc: double-p

RIDDLES JITSI

- so many components and than some
- even more communication, firewall nightmare?
- all localhost and “discovery magic”
- `location ~ ^/([^/?&:'"]+)/(.*)$`
- help? -> “quick install!”, “one-debian VM”
- java, uuuh, two versions of ’em!
- how many in DNS again?
- who/where/what/wtf - towel launcher

RIDDLES OPENBSD

- need Linux-VMs? no/yes:Alpine/NixOS
- vm.conf examples / hoowl-toooo scarce
- inter VM networking (“the switch is dead”)
- inside/outside networking (VMM as router)
- scale out (VMM machines, bare-metal, cloud)
- java + rcctl
- just `pkg_add`?

COMPONENTS OPENBSD

vmm(4) - virtual machine monitor:

kernel driver isolating/providing the required resources for the VMs (“hypervisor”)

vmd(8):

userland daemon to interact with `vmm`

vmctl(8):

administrative tool to create, start/stop, .. VMs

vm.conf(5):

persist VMs resource configuration

COMPONENTS JITSI

nginx(8): `web`

serving web assets and reverse proxy BOSH or websockets

prosody(8): `xmpp`

conference chat + internal components communication (esp. “PubSub” for health/discovery)

COMPONENTS JITSI (C.)

focus: `jicofo` Jitsi COnference FOcus

room+session handling in conferences (who's talking to whom and where)

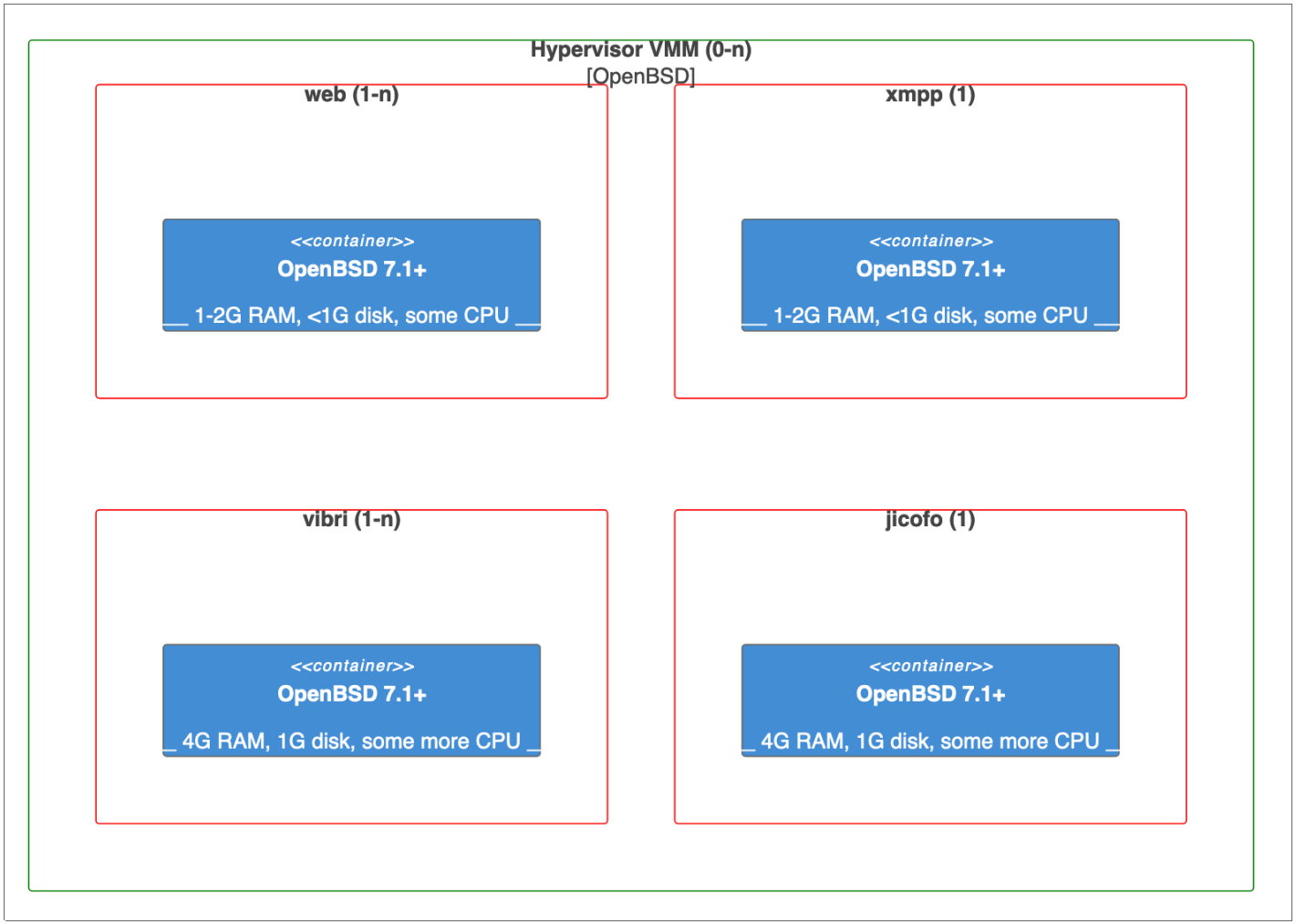
videobridge: `jvb`

mediastream (WebRTC) handlings between participants (SFU)

jibri: Jitsi BRoadcasting Infrastructure

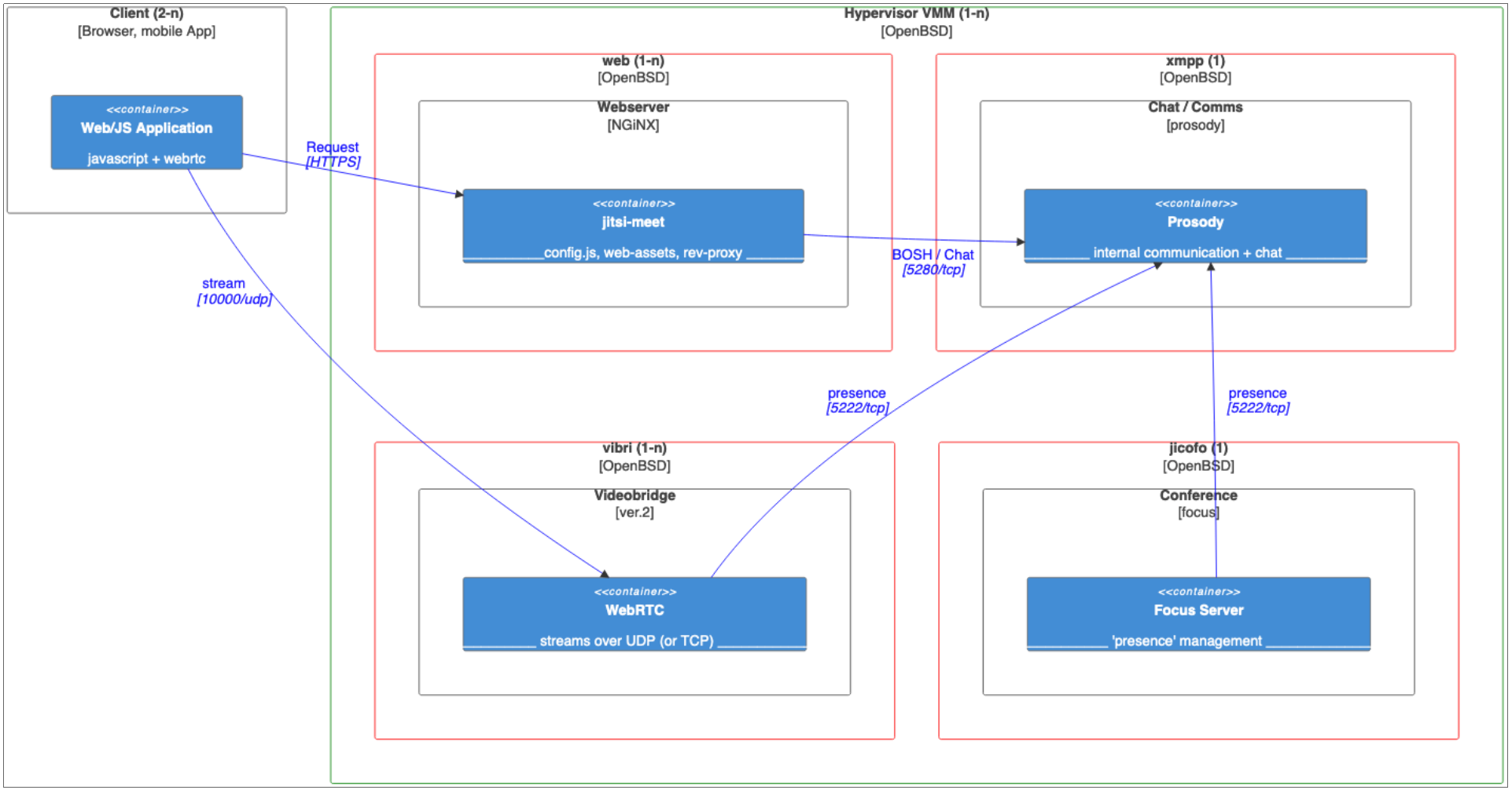
recording + streaming conferences

ARCHITECTURE OPENBSD

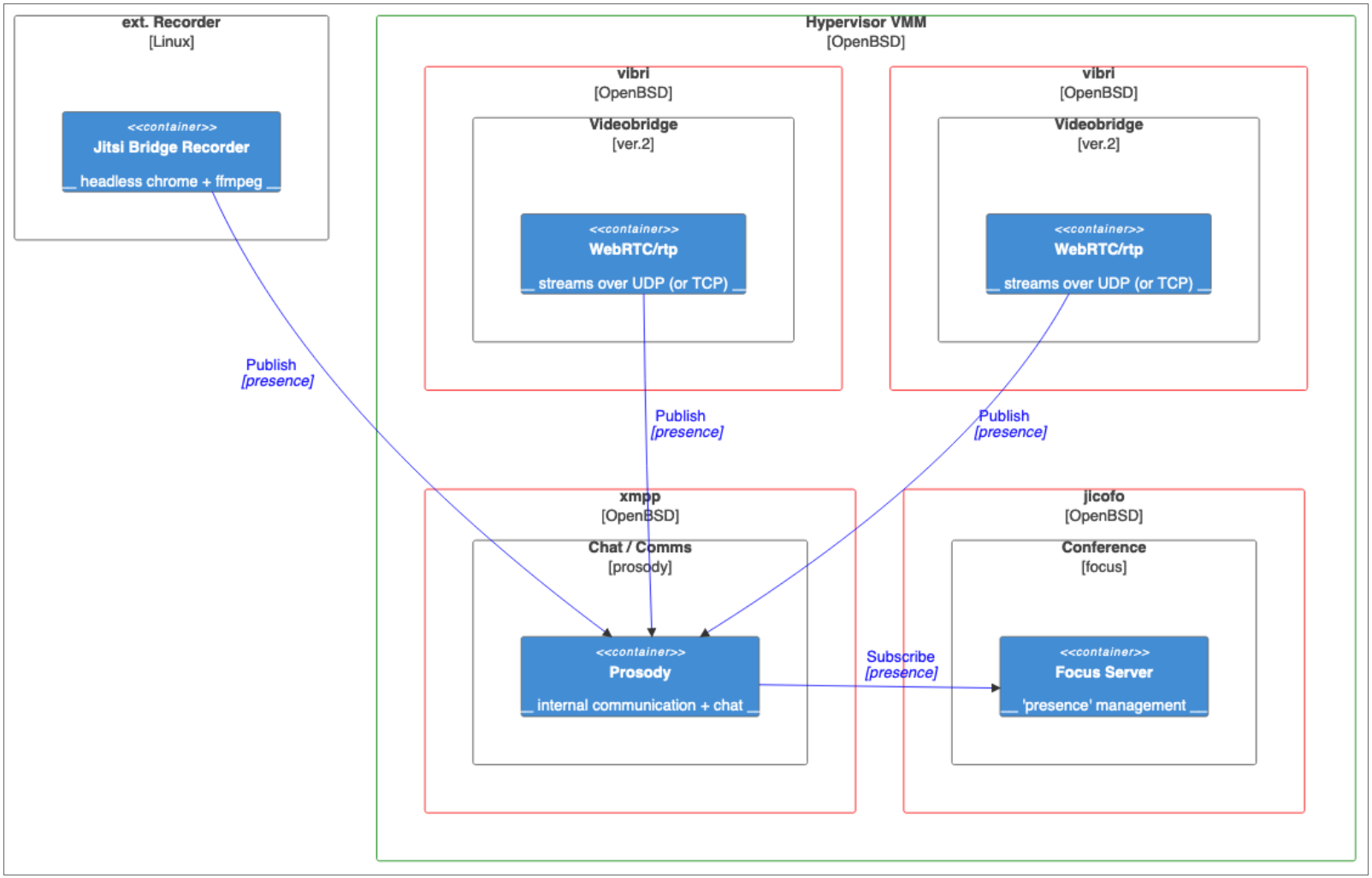


!

ARCHITECTURE JITSI (NET)



ARCHITECTURE JITSI (LOGICAL / EXT.)



COMMUNICATIONS WORKFLOW

INSTALL OVERVIEW

- create VM images
- /etc/vm.conf
- hosts / DNS
- nginx: install, config, certs
- prosody: install, config, certs, users
- jicofo: install, config
- jvb: install, config
- DONE

VM INSTALLATION

```
1  rcctl enable vmd; rcctl start vmd
2  mkdir /home/vmm; cd /home/vmm
3  vmctl create -s 5G web.qcow2
4  ftp https://cdn.openbsd.org/pub/OpenBSD/7.1/amd64/install71.iso
5  vmctl start -m 2G -L -i 1 -r install71.iso -d /home/vmm/web.qcow2 wel
6  vmctl console web
7  ## run the (I)nstaller, default options. only one 'a' slice on (w)ho
8  ## halt -p (so new sshd_keys per VM)
9  vmctl stop web
10 for vm in xmpp jicofo jvb ; do cp web.qcow2 $vm.qcow2; done
11 echo 'net.inet.ip.forwarding=1' >> /etc/sysctl.conf
```

VMM /etc/vm.conf

```
vm "web" {
  enable
  memory 2G
  disk "/home/vmm/web.qcow2" format qcow2
  local interface { up }
}
vm "web" instance "xmpp" {
  disk "/home/vmm/xmpp.qcow2" format qcow2
}
vm "web" instance "jicofo" {
  memory 4G
  disk "/home/vmm/jicofo.qcow2" format qcow2
}
vm "web" instance "jvb" {
  memory 4G
  disk "/home/vmm/jvb.qcow2" format qcow2
}
```

DNS, `/etc/hosts`

> **Note**
>
> DNS: ONE A-RR for jts.fips.de; but local `hosts` for jicofo (or split DNS)

```
1 100.64.1.3    web
2 100.64.2.3    xmpp jts.fips.de
3 100.64.3.3    jicofo
4 100.64.4.3    jvb
```

> **Note**
>
> adapt `/etc/myname` in each VM accordingly

`/etc/pf.conf` EACH VM, VMM

> **Note**
>
> Not needed for jitsi itself, rather common admin care

```
1 block return log
2 pass out quick on egress proto { tcp udp } to any port { 123 53 80 4
3 pass in quick on egress proto tcp from $admin to port 22
```

> **Note**
>
> block both ways; allow NTP, DNS, HTTP(S), SSH

`/etc/pf.conf` VMM (AS ROUTER)

Assuming all traffic hits the VMM external IP-address (`on egress`)

- jitsi specifics:

```
1  pass in on egress proto tcp to any port { 80 443 } rdr-to web
2  pass in on egress proto udp to any port { 10000 } rdr-to jvb
3  pass in proto tcp from { jvb jicofo } to xmpp port 5222 # native
4  pass in proto tcp from web to xmpp port 5280 # http/BOSH
5  pass in on egress proto tcp to any port 5280 rdr-to xmpp # debug
```

- DNS

```
1  vms={ web xmpp jicofo jvb }
2  pass in proto { udp tcp } from $vms to any port domain rdr-to $resol
```

/etc/pf.conf WEB / NGINX

```
1 pass in quick on egress proto tcp to self port { 80 443 }
2 pass out quick on egress proto tcp to xmpp port 5280
```

> **Note**
>
> BOSH or XMPP-websocket both run via 5280/tcp

/etc/pf.conf XMPP / PROSODY

```
1 pass in proto tcp from { jicofo jvb } to self port { 5222 }
2 pass in proto tcp from web to self port 5280
3 pass in proto tcp from { $admin } to self port { 5280 5347} # debug
```

> **Note**
>
> 5347/tcp for explicit authentication if need be (not here)

/etc/pf.conf JVB / VIDEOBRIDGE

```
1 pass out quick on egress proto tcp to xmpp port { 5222 5280 }
2 pass in quick on egress proto udp to self port 10000
3 pass in quick on egress proto tcp from $monitor to self port 8080
```

Note

for scale out the 10000/udp can be changed and would make use of a port range e.g. ‘10000:10050’ vertically or explicit rdr-to in VMM for horizontally

Note

jvb has a REST API on 8080/tcp for health/metrics (prometheus)

/etc/pf.conf JICOFO / FOCUS SERVER

```
1 pass out quick on egress proto tcp to xmpp port 5222
2 pass in quick on egress proto tcp from $monitor to self port 8888
```

> **Note**
>
> jicofo has a REST API on 8888/tcp for health/metrics (prometheus) - BROKEN?

INSTALL / CONFIGURATION PROSODY (XMPP)

Besides the package itself, we need some additional modules

```
1 pkg_add unzip-- prosody
2 prosodyctl install --server=https://modules.prosody.im/rocks/ mod_cl
3 prosodyctl install --server=https://modules.prosody.im/rocks/ mod_ro
```

> **Note**
>
> The modules do not need further configuration in prosody.cfg.lua: client_proxy gets loaded with “Component” configuration. roster_command is CLI only.

CONFIGURATION PROSODY (XMPP)

- `/etc/prosody/prosody.cfg.lua`: (shortened)

```
 1  http_interfaces = { "*", "::" }
 2  VirtualHost "jts.fips.de"
 3      authentication = "anonymous";
 4      modules_enabled = { "bosh"; "pubsub"; }
 5      c2s_require_encryption = false
 6
 7  VirtualHost "auth.jts.fips.de"
 8      admins = { "focus@auth.jts.fips.de", "jvb@auth.jts.fips.de" }
 9      ssl = { key = "/var/prosody/auth.jts.fips.de.key";
10              certificate = "/var/prosody/auth.jts.fips.de.crt"; }
11      authentication = "internal_hashed"
```

> **Note**
>
> `admins` usage details unclear

CONFIGURATION PROSODY (XMPP) (CONT.)

- `/etc/prosody/prosody.cfg.lua`: (shortened)

```
1  Component "conference.jts.fips.de" "muc"
2  Component "jvb.jts.fips.de"
3      component_secret = "CHANGE_jvb"
4  Component "focus.jts.fips.de" "client_proxy"
5      target_address = "focus@auth.jts.fips.de"
6  Component "internal.auth.jts.fips.de" "muc"
7      muc_room_locking = false
8      muc_room_default_public_jids = true
```

> **Note**
>
> No extra DNS needed! Like “Host:” HTTP-Header.
> `focus` was using component_secret like `jvb` in earlier versions

PROSODY USERS

The connection for `jvb` uses a shared secret as shown on the previous page.

```
1 rcctl enable prosody ; rcctl start prosody
```

Jicofo's "focus" user:

```
1 prosodyctl register focus auth.jts.fips.de CHANGE_FOCUS
2 prosodyctl mod_roster_command subscribe focus.jts.fips.de focus@auth
```

> **Note**
>
> Documentation a bit scarce about what's in for this subscription part - but needed.

TLS CERTIFICATES (PROSODY / JKS)

```
1 prosodyctl cert generate auth.jts.fips.de
2 # fill in `openssl req` dialog
3 cd /var/prosody
4 yes | /usr/local/jdk-11/bin/keytool -import -alias prosody -file \
5   auth.jts.fips.de.crt -keystore jicofo-key.store -storepass jitsico
6 cp jicofo-key.store jvb-key.store # copy to VM jicofo, jvb according
```

Note

`keytool` comes with JDK, this task can also be done on jicofo or jvb VM - or copy over the resulting store-files to jicofo/jvb VM respectivly

Important

Do NOT change JDK's `lib/security/cacerts` - any later upgrade from jdk-11 would "forget" the keystore

INSTALL NGINX / WEB

nginx and the jitsi web elements

```
pkg_add nginx
pkg_add jitsi-meet
```

Any TLS setup is mandatory or Chrome/Firefox/.. will refuse to let you use the camera+microphone.
Using Let's Encrypt with `acme-client` for the TLS setup is easily possible and included in `./testing-config/nginx.conf`.

> **Note**
>
> only needed jitsi configuration is config.js, see below

WEB / NGINX.CONF

/etc/nginx/nginx.conf

```
 1  server_name  jts.fips.de;
 2  root         /var/www/jitsi-meet;
 3  ssi on;
 4  ssi_types application/x-javascript application/javascript;
 5  location ~ ^/(libs|css|static|images|fonts|lang|sounds|connection_opt
 6    add_header 'Access-Control-Allow-Origin' '*';
 7    alias /var/www/jitsi-meet/$1/$2; }
 8  location /external_api.js { alias /var/www/jitsi-meet/libs/external_a
 9  location = /http-bind {   # BOSH
10    proxy_pass      http://xmpp:5280/http-bind;
11    proxy_set_header X-Forwarded-For $remote_addr;
12    proxy_set_header Host $http_host; }
13  location ~ ^/([a-zA-Z0-9=\?]+)$ {
14    rewrite ^/(.*)$ / break; }
```

If using LE, put the `.well-known` location first (above L5)

WEB / MOBILE CLIENT

- `/var/www/jitsi-meet/config.js`:

```
var config = {
  hosts: {
    domain: 'jts.fips.de',
    muc: 'conference.jts.fips.de' // no DNS
  },
  bosh: '//jts.fips.de/http-bind',
  useTurnUdp: false, enableWelcomePage: true,
  prejoinConfig: {
    enabled: true,
    hideExtraJoinButtons: ['no-audio', 'by-phone'] },
  p2p: {
    stunServers: [
      { urls: 'stun:meet-jit-si-turnrelay.jitsi.net:443' } ] }
}
```

TURN depends on NAT environment(s)

INSTALL / CONFIG JICOFO

- `pkg_add jicofo`
- adapt `/etc/jicofo/jicofo.in.sh` if need be

```
1 JICOFO_CONF=/etc/jicofo/jicofo.conf
2 JICOFO_LOG_CONFIG=/usr/local/share/jicofo/lib/logging.properties
3 JICOFO_TRUSTSTORE=/etc/ssl/jicofo-key.store
4 JICOFO_TRUSTSTORE_PASSWORD=jitsicool
5 JICOFO_MAXMEM=3G
6 JICOFO_DHKEYSIZE=2048
7 JAVA_SYS_PROPS=""
```

> **Tip**
>
> jicofo-key.store is generated from prosody certificate, see prosody slide.
>
> you can enable an XMPP-packet-debug-log in logging.properties

JITSI / JICOFO

- `/etc/jicofo/jicofo.conf`: (shortened)

```
jicofo { bridge {
  brewery-jid = "JvbBrewery@internal.auth.jts.fips.de"
  xmpp-connection-name = Client } // enum
  sctp { enabled = false }
  xmpp {
    client {
      port = 5222
      domain = "auth.jts.fips.de"
      username = "focus"
      password = "CHANGE_FOCUS"
      use-tls = true
    }
    // trusted service domains. Logged in -> advance to bridges
    trusted-domains = [ "auth.jts.fips.de" ]
  }
}
```

JITSI / JICOFO (CONT.)

- `/etc/rc.conf.local`:

```
1 jicofo_flags="--host=jts.fips.de"
```

> **Important**
>
> Needs `/etc/hosts` or split-DNS. Used for TCP connect AND virtualhost

- `/etc/syslog.conf`

```
1 !jicofo
2 *.*     /var/log/jicofo
```

```
rcctl enable jicofo ; rcctl start jicofo
```

INSTALL / CONFIG JVB `jvb.in.sh`

- `pkg_add jitsi-videobrige`

```
JVB_CONF=/etc/jvb/jvb.conf
JVB_LOG_CONFIG=/usr/local/share/jvb/lib/logging.properties
JVB_TRUSTSTORE=/etc/ssl/jvb-key.store
JVB_TRUSTSTORE_PASSWORD=jitsicool
JVB_MAXMEM=3G
JVB_DHKEYSIZE=2048
JVB_GC_TYPE=G1GC
JAVA_SYS_PROPS=""
# reads /etc/jvb/sip-communicator.properties
JVB_SC_HOME_LOCATION='/etc'
JVB_SC_HOME_NAME='jvb'
```

> **Tip**
>
> jvb-key.store is generated from prosody certificate, see earlier slide. can be same file as /etc/jicofo/jicofo-key.store on one VM

CONFIGURATION /etc/jvb/jvb.conf

```
videobridge { apis {
  xmpp-client {
   configs {
    ourprosody {
     hostname = "xmpp"
     domain = "auth.jts.fips.de" // 'realm'
     username = "jvb"
     password = "CHANGE_jvb"
     muc_jids = "JvbBrewery@internal.auth.jts.fips.de"
     muc_nickname = "jvb-foo"
     disable_certificate_verification = true } } } }
  sctp { enabled = false } // n/a on OpenBSD
  ice { tcp {
    enabled = false
    port = 443
   } udp {
    port = 10000
   }
  }
```

JITSI / JVB (BONUSSLIDE)

/etc/jvb/sip-communicator.properties:

```
1 org.ice4j.ice.harvest.NAT_HARVESTER_LOCAL_ADDRESS=100.64.4.3
2 org.ice4j.ice.harvest.NAT_HARVESTER_PUBLIC_ADDRESS=87.253.170.146
3 org.ice4j.ice.harvest.DISABLE_AWS_HARVESTER=true
```

> **Note**
>
> ice4j has not been migrated yet, thus not available in “new” JICO config format

/etc/syslog.conf

```
1 !jvb
2 *.*     /var/log/jvb
```

```
rcctl enable jvb ; rcctl start jvb
```

PITFALLS / HINTS OPENBSD

- `rc.conf.local`: jicofo_flags: IP instead DNS/hosts
- startup ordering:
  - web
  - xmpp
  - jicofo
  - jvb

  avoid retries with long wait times (health interval)

PITFALLS / HINTS JITSI

- xmpp: host vs. virtualhost vs. domain
- DNS: one and only one (or mess up xmpp fallback)
- not disabling sctp (jvb AND jicofo)
- (hidden) version bumps (“no longer component!”)
- jicofo has an XMPP-packetlogger (see `logging.properties`)
- Patience, be really patient (at initial start)
- always check the `,` in config.js (JSON after all..)

PRAYERS FOR A LIVEDEMO

FIPS.DE

https://jts.fips.de/EBC

.

Infrastructure kindly provided by Mischa Peters from

STATUS OPENBSD

> **Important**
>
> IT WORKS!

Ports / Packages (Bonusslide)

- `net/jitsi/videobridge`: **current/7.2**
- `net/jitsi/jicofo`: **current/7.2**
- `net/jitsi/meet`: **current/7.2**
- `www/jitsi-meta`: planned, include above and coherent configuration (all on localhost)

OUTLOOK

SCALE / CLOUDIFICATION

- `nginx`: easy scale out; protect nginx->xmpp (new CPU sink with TLS)
- `jvb`: ok; use statistics and one IP only per JVM instance
- `xmpp`: harder; what parameter to split on the frontend/nginx. how to tell jicofo?
- `jicofo`: unsure or not feasible; not the workload horse, also

OUTLOOK (C.)

JITSI

- `srtp`: faster crypto (.so inline)
- `xmpp/jvb`: might move from BOSH to websockets eventually (who takes the perf hit)
- `jibri`: likely linux only (chrome-headless/-driver, x-fb+ffmpeg rip), nixos?; comms like `jvb`
- `jigasi`: feel free / just no POTS/SIP dialin?

HELP

- community.jitsi.org: it’s a rough place of quick-install

QUESTIONS ? / K-THX-BYE

Thanks to:

- OpenBSD / Jitsi
- sysfive.com GmbH
- Aisha Tammy (ports)
- Mischa (obsd.ams)

Misc:

- QUESTIONS
- Meet: Hallway
- No Lunch

> **Presentation**
>
> Created with `Quarto / revealjs`

presentation (+testing-config) `https://is.gd/ontSw3`

BONUS

-CURRENT PORTS

- fetch snapshots ports.tar.gz and unpack in a temporary path

```
1  CVSROOT=anoncvs.XX.openbsd.org:cvs
2  cd temppath/ports/net ; cvs up -Pd -A
3  cp -pr jitsi /usr/ports/net
4  cp ../infrastructure/db/user.list /usr/ports/infrastructure/db/
5  cd /usr/ports/net/jitsi ; make package # get a coffee/tea/..
6  scp /usr/ports/packages/amd64/all/jitsi-meet* web:/tmp
7  scp /usr/ports/packages/amd64/all/jitsi-videobridge* jvb:/tmp
8  scp /usr/ports/packages/amd64/all/jicofo* jicofo:/tmp
```

BONUS 2

on each vm, install as so

```
TRUSTED_PKG_PATH="/tmp:https://cdn.openbsd.org/pub/OpenBSD
/7.1/packages/amd64/" pkg_add jicofo
```

BONUS LOGS (XMPP/JICOFO)

```
 wrong/missing keystore:
 Sep 16 14:35:58 c2s60a86b47480  info    Client connected
 Sep 16 14:35:58 c2s60a86b47480  info    Client disconnected: sslv3
 alert certificate unknown

 wrong password:
 Sep 16 14:36:20 c2s60b1aced0c0  info    Stream encrypted (TLSv1.3 with
 TLS_AES_256_GCM_SHA384)
 Sep 16 14:36:21 c2s60b1aced0c0  info    Client disconnected:
 connection closed

 all good:
 Sep 16 14:37:00 c2s60a5b549900  info    Stream encrypted (TLSv1.3 with
 TLS_AES_256_GCM_SHA384)
 Sep 16 14:37:01 c2s60a5b549900  info    Authenticated as
 focus@auth.jts.fips.de
:

 cant use bridge before (might take; "presence-interval = 120s")
 Jicofo 2022-09-16 14:45:24.260 INFO: [35] [type=bridge
 brewery=jvbbrewery] BaseBrewery.addInstance#341: Added brewery
 instance: jvbbrewery@internal.auth.jts.fips.de/jvb
```

Error