# **Server Installation for Jitsi Meet**

Reference: <https://github.com/jitsi/jitsi-meet/blob/master/doc/manual-install.md>

1. General setup
2. Install prosody and otalk modules
3. Configure prosody
4. Install nginx
5. Install Jitsi Videobridge
6. Install Jitsi Conference Focus(jicofo)
7. Deploy Jitsi Meet
8. Install Turn server
9. Hold your first conference

**General setup**:

Domain: prosody.htk.com

IP: 192.168.1.70

Password: 123qwe

Install PKG\_DEPENCIES: apt-get install unzip vim nano git ant python-pip nginx default-jre default-jdk npm node g++ make lib32z1 mercurial

Install PKG\_LIBRARIES:

Lua-expat 1.3:

wget <http://launchpadlibrarian.net/173842910/lua-expat_1.3.0-1_amd64.deb>

dpkg -i lua-expat\_1.3.0-1\_amd64.deb

another packages:

apt-get install lua-zlib lua-dbi-sqlite3 liblua5.1-bitop-dev liblua5.1-bitop0 python-ldap libldap-dev liblua5.1-0-dev lua-event-dev

**Install prosody and otalk modules:**

Install Prosody:

*echo deb http://packages.prosody.im/debian $(lsb\_release -sc) main | sudo tee -a /etc/apt/sources.list*

*wget --no-check-certificate https://prosody.im/files/prosody-debian-packages.key -O- | sudo apt-key add -*

*apt-get update*

*sudo apt-get install --yes -q prosody-0.10*

Install Otalk:

*git clone https://github.com/andyet/otalk-server.git*

*cd otalk-server*

*cp -r mod\* /usr/lib/prosody/modules*

**Configure prosody**

Modify the config file in /etc/prosody/prosody.cfg.lua

vi /etc/prosody/prosody.cfg.lua

|  |
| --- |
| -- Prosody XMPP Server Configuration  --  -- Information on configuring Prosody can be found on our  -- website at http://prosody.im/doc/configure  --  -- Tip: You can check that the syntax of this file is correct  -- when you have finished by running: prosodyctl check config  -- If there are any errors, it will let you know what and where  -- they are, otherwise it will keep quiet.  --  -- Good luck, and happy Jabbering!  ---------- Server-wide settings ----------  admins = {"focus@auth.prosody.htk.com"}  --use\_libevent = true  modules\_enabled = {  -- Generally required  "roster"; -- Allow users to have a roster. Recommended ;)  "saslauth"; -- Authentication for clients and servers. Recommended if you want to log in.  "tls"; -- Add support for secure TLS on c2s/s2s connections  "dialback"; -- s2s dialback support  "disco"; -- Service discovery  -- Not essential, but recommended  "private"; -- Private XML storage (for room bookmarks, etc.)  "vcard"; -- Allow users to set vCards    -- These are commented by default as they have a performance impact  --"blocklist"; -- Allow users to block communications with other users  "compression"; -- Stream compression (requires the lua-zlib package installed)  -- Nice to have  "version"; -- Replies to server version requests  "uptime"; -- Report how long server has been running  "time"; -- Let others know the time here on this server  "ping"; -- Replies to XMPP pings with pongs  "pep"; -- Enables users to publish their mood, activity, playing music and more  "register"; -- Allow users to register on this server using a client and change passwords  -- Admin interfaces  "admin\_adhoc"; -- Allows administration via an XMPP client that supports ad-hoc commands  --"admin\_telnet"; -- Opens telnet console interface on localhost port 5582    -- HTTP modules  "bosh"; -- Enable BOSH clients, aka "Jabber over HTTP"  --"http\_files"; -- Serve static files from a directory over HTTP  -- Other specific functionality  --"groups"; -- Shared roster support  --"announce"; -- Send announcement to all online users  --"welcome"; -- Welcome users who register accounts  --"watchregistrations"; -- Alert admins of registrations  --"motd"; -- Send a message to users when they log in  --"legacyauth"; -- Legacy authentication. Only used by some old clients and bots.  --HTK-INC  "smacks";  "carbons";  "mam";  "lastactivity";  "offline";  "pubsub";  "adhoc";  "websocket";  "http\_altconnect";  "turncredentials";  }  --HTK-INC  daemonize = true  cross\_domain\_bosh = true;  storage = {archive2 = "sql2"}  sql = { driver = "SQLite3", database = "prosody.sqlite" }  default\_archive\_policy = "roster"  allow\_registration = false  ssl = {  key = "/var/lib/prosody/prosody.htk.com.key";  certificate = "/var/lib/prosody/prosody.htk.com.crt";  }  c2s\_require\_encryption = true  s2s\_secure\_auth = false  -- Required for init scripts and prosodyctl  pidfile = "/var/run/prosody/prosody.pid"  -- Logging configuration  -- For advanced logging see http://prosody.im/doc/logging  log = {  debug = "/var/log/prosody/prosody.log"; -- Change 'info' to 'debug' for verbose logging  error = "/var/log/prosody/prosody.err";  "\*syslog";  }  ----------- Virtual hosts -----------  VirtualHost "prosody.htk.com"  --authentication = "anonymous"  authentication = "external"  external\_auth\_command ="/opt/VNCtalk/prosody-packaging/prosody\_authProvider/vnc\_prosody\_auth\_provider.sh"  external\_auth\_protocol = "generic"    ssl = {  key = "/var/lib/prosody/prosody.htk.com.key";  certificate = "/var/lib/prosody/prosody.htk.com.crt";  }  turncredentials\_secret = "123qwe";  turncredentials = {  { type = "turn", host = "turn.192.168.1.70", port = 3478, transport = "tcp" }  }  VirtualHost "auth.prosody.htk.com"  authentication = "internal\_plain"  ssl = {  key = "/var/lib/prosody/prosody.htk.com.key";  certificate = "/var/lib/prosody/prosody.htk.com.crt";  }  turncredentials\_secret = "123qwe";  turncredentials = {  { type = "turn", host = "turn.192.168.1.70", port = 3478, transport = "tcp" }  }  --admins = { "focus@auth.prosody.htk.com" }  ------ Components ------  Component "conference.prosody.htk.com" "muc"  Component "jitsi-videobridge.prosody.htk.com"  component\_secret = "123qwe"  Component "focus.prosody.htk.com"  component\_secret = "123qwe" |

Generate certs for the domain:

*prosodyctl cert generate prosody.htk.com*

Create conference focus user:

*prosodyctl register focus auth.prosody.htk.com 123qwe*

Restart prosody XMPP server with the new config

*prosodyctl restart*

**Install nginx**

apt-get install nginx

nginx config for domain in /etc/nginx/nginx.conf:

|  |
| --- |
| user www-data;  worker\_processes 4;  pid /run/nginx.pid;  events {  worker\_connections 768;  # multi\_accept on;  }  http {  ##  # Basic Settings  ##  sendfile on;  tcp\_nopush on;  tcp\_nodelay on;  keepalive\_timeout 65;  types\_hash\_max\_size 2048;  # server\_tokens off;  server\_names\_hash\_bucket\_size 64;  # server\_name\_in\_redirect off;  include /etc/nginx/mime.types;  default\_type application/octet-stream;  ##  # Logging Settings  ##  access\_log /var/log/nginx/access.log;  error\_log /var/log/nginx/error.log;  ##  # Gzip Settings  ##  gzip on;  gzip\_disable "msie6";    ##  # Virtual Host Configs  ##  include /etc/nginx/conf.d/\*.conf;  include /etc/nginx/sites-enabled/\*;  } |

Add a new file prosody.htk.com in /etc/nginx/sites-available:

|  |
| --- |
| server {  listen 443;  ssl on;  ssl\_certificate /var/lib/prosody/prosody.htk.com.crt;  ssl\_certificate\_key /var/lib/prosody/prosody.htk.com.key;  server\_name prosody.htk.com;  access\_log /var/log/nginx/nginx.vhost.access.log;  error\_log /var/log/nginx/nginx.vhost.error.log;  root /srv/prosody.htk.com;  index index.html;  location ~ ^/([a-zA-Z0-9]+)$ {  rewrite ^/(.\*)$ / break;  }  # BOSH  location /http-bind {  proxy\_pass https://localhost:5281/http-bind;  proxy\_set\_header X-Forwarded-For $remote\_addr;  proxy\_set\_header Host $host;  proxy\_set\_header X-Real-IP $remote\_addr;  }  location / {  ssi on;  }  }  server {  listen 80;  server\_name prosody.htk.com;  # set the root  root /srv/prosody.htk.com;  index index.html;  location ~ ^/([a-zA-Z0-9]+)$ {  rewrite ^/(.\*)$ / break;  }  # BOSH  location /http-bind {  proxy\_pass http://prosody.htk.com:5280/http-bind;  proxy\_set\_header X-Forwarded-For $remote\_addr;  proxy\_set\_header Host $http\_host;  }  # xmpp websockets  location /xmpp-websocket {  proxy\_pass http://localhost:5280;  proxy\_http\_version 1.1;  proxy\_set\_header Upgrade $http\_upgrade;  proxy\_set\_header Connection "upgrade";  proxy\_set\_header Host $host;  tcp\_nodelay on;  }  } |

Add link for the added configuration

*cd /etc/nginx/sites-enabled  
ln -s ../sites-available/prosody.htk.com prosody.htk.com*

**Install Jitsi Videobridge**

*sudo add-apt-repository --yes 'deb http://download.jitsi.org/nightly/deb unstable/'*

*sudo wget -qO - https://download.jitsi.org/nightly/deb/unstable/archive.key | sudo apt-key add -*

*sudo apt-get update > /dev/null*

*sudo apt-get install --yes -q jitsi-videobridge*

In the user home that will be starting Jitsi Videobridge create .sip-communicator folder and add the file sip-communicator.properties with one line in it:

*sudo mkdir -p /home/jvb/.sip-communicator*

*sudo chown jvb:jitsi /home/jvb/*

*sudo chown jvb:jitsi /home/jvb/.sip-communicator*

*echo "org.jitsi.impl.neomedia.transform.srtp.SRTPCryptoContext.checkReplay=false" | sudo tee -a /home/jvb/.sip-communicator/sip-communicator.properties*

Config Jitsi Videobridge

vi /etc/jitsi/videobridge/config

|  |
| --- |
| # Jitsi Videobridge settings  JVB\_HOSTNAME=prosody.htk.com  JVB\_HOST=localhost  JVB\_PORT=5347  JVB\_SECRET=123qwe  JVB\_OPTS="" |

*sudo update-rc.d jitsi-videobridge defaults*

*sudo service jitsi-videobridge restart*

**Install Jitsi Conference Focus(jicofo)**

*sudo apt-get install --yes -q jicofo*

vi /etc/jitsi/jicofo/config

|  |
| --- |
| *# Jitsi Conference Focus settings*  *JICOFO\_HOST=localhost*  *JICOFO\_HOSTNAME=prosody.htk.com*  *JICOFO\_SECRET=123qwe*  *JICOFO\_PORT=5347*  *JICOFO\_AUTH\_DOMAIN=auth.prosody.htk.com*  *JICOFO\_AUTH\_USER=focus*  *JICOFO\_AUTH\_PASSWORD=123qwe*  *JICOFO\_OPTS="”* |

**Deploy Jitsi Meet**

Checkout and configure Jitsi Meet:

*cd /srv  
git clone* [*https://github.com/jitsi/jitsi-meet.git*](https://github.com/jitsi/jitsi-meet.git)

*(git checkout 239f271caf81080fc34328e22ea99cc7351079eb)*

*mv jitsi-meet/ prosody.htk.com*

Edit host names in /srv/*prosody.htk.com*/config.js

|  |
| --- |
| var config = {  hosts: {  domain: 'prosody.htk.com',  muc: 'conference.prosody.htk.com',  bridge: 'jitsi-videobridge.prosody.htk.com'  },  useNicks: false,  bosh: '//prosody.htk.com/http-bind', // FIXME: use xep-0156 for that  desktopSharing: 'false' // Desktop sharing method. Can be set to 'ext', 'webrtc' or false to disable.  //chromeExtensionId: 'diibjkoicjeejcmhdnailmkgecihlobk', // Id of desktop streamer Chrome extension  //minChromeExtVersion: '0.1' // Required version of Chrome extension  }; |

Restart nginx to get the new configuration:

*invoke-rc.d nginx restart*

**Install Turn server**

**Install libre**

|  |
| --- |
| **wget http://creytiv.com/pub/re-0.4.10.tar.gz -O re-0.4.10.tar.gz**  **tar xfvz re-0.4.10.tar.gz**  **cd re-0.4.10/**  **make**  **sudo make install** |

Install restund

|  |
| --- |
| **git clone** [**https://github.com/otalk/restund.git**](https://github.com/otalk/restund.git)  **cd restund/**  **git pull --all -v**  **make**  **sudo make install**  **sudo ldconfig**  **cd /etc**  **wget** [**https://raw.github.com/andyet/otalk-server/master/restund/restund.conf**](https://raw.github.com/andyet/otalk-server/master/restund/restund.conf)  **cp restund.conf /etc/restund.conf** |

vi /etc/restund.conf

|  |
| --- |
| # core  daemon yes  debug no  realm  syncinterval 600  udp\_listen :3478  udp\_sockbuf\_size 524288  tcp\_listen :3478  #tls\_listen 1.2.3.4:3479,/path/to/keyandcert.pem  #udp\_internal\_listen    # modules  module\_path /usr/local/lib/restund/modules  module stat.so  module binding.so  module auth.so  module turn.so  #module mysql\_ser.so  module syslog.so  module status.so  # auth  auth\_nonce\_expiry 3600  auth\_shared\_expiry 86400  auth\_shared # share this with your prosody server  # turn  turn\_max\_allocations 512  turn\_max\_lifetime 600  turn\_relay\_addr  #turn\_relay\_addr6 ::1  # syslog  syslog\_facility 24  # status  status\_udp\_addr 127.0.0.1  status\_udp\_port 33000  status\_http\_addr 127.0.0.1  status\_http\_port 8080 |

*cp debian/restund.init /etc/init.d/restund*

*sudo update-rc.d restund defaults*

*sudo service restund start*

**Hold your first conference:** visit to https://prosody.htk.com