

Jitsiverse of Madness



Jitsi Meet

Secure and high quality meetings

AdverseSiblingsFunctionSilently

Start meeting

Oct 28, 2022 hod8

6:48 PM 09:04



Jitsi Meet



Browser



Mobile

443

10000/udp

443

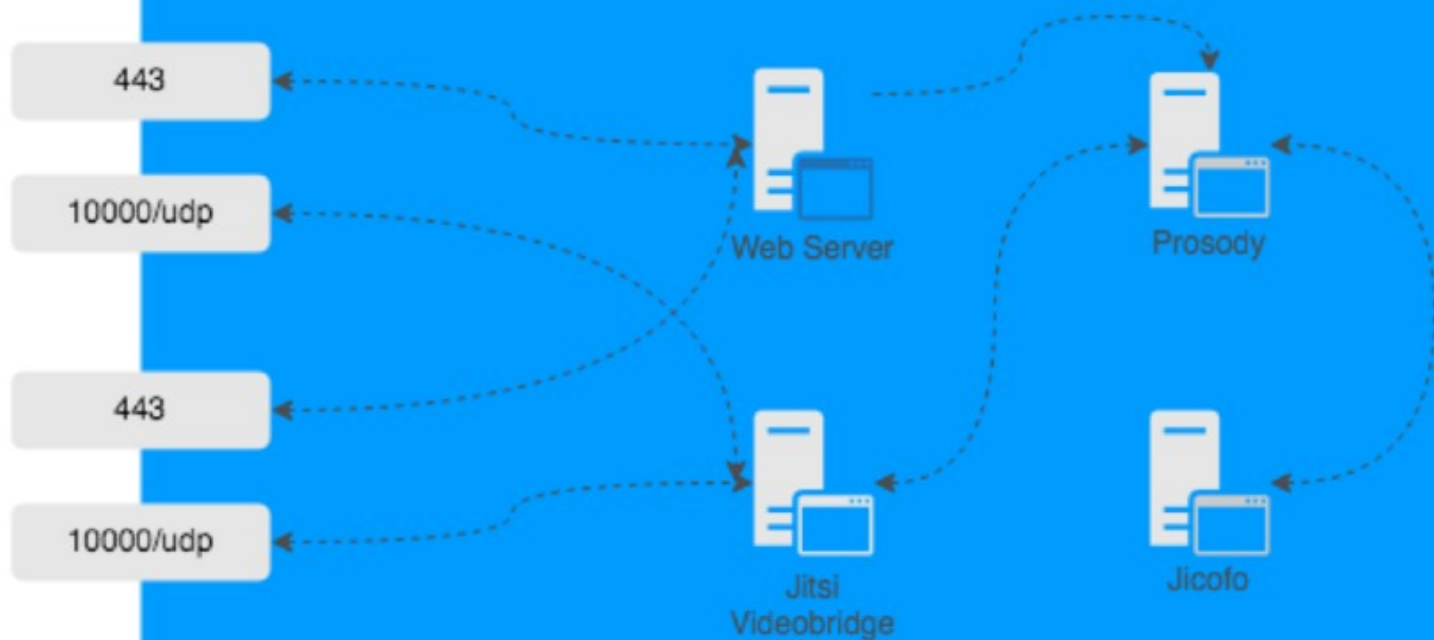
10000/udp

Web Server

Jitsi
Videobridge

Prosody

Jicofo



Jitsi Meet

- Web Interface
 - WebRTC
 - React
 - React Native
- Nginx

Jitsi Videobrigde (JVB)

- WebRTC compatible server designed to route video streams
- Multiplexer Audio/Video
- DTLS-SRTP

Jitsi Conference Focus (Jicofo)

manages media sessions and acts as load balancer between each of the participants and the videobridge.

Prosody

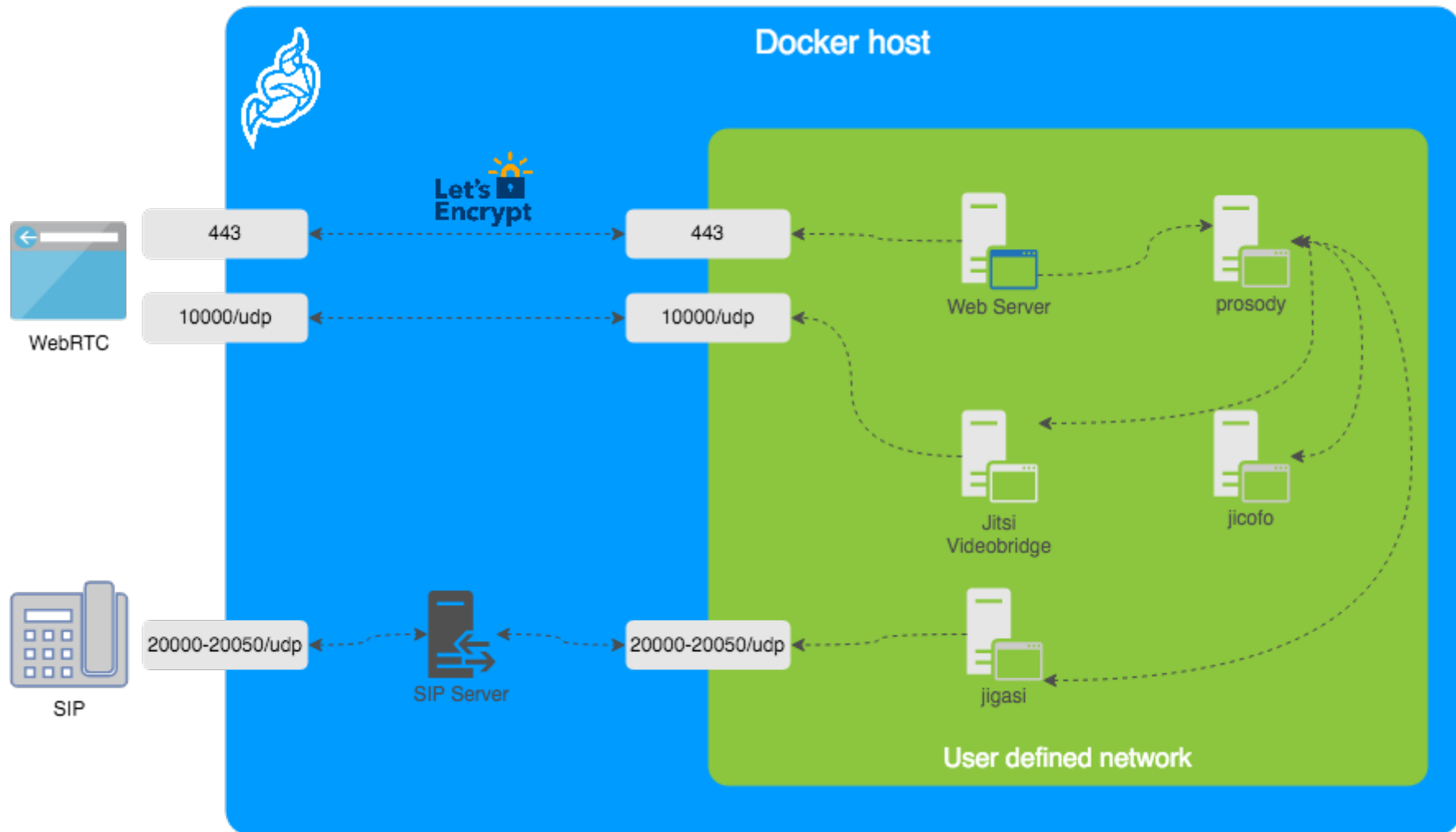
- XMPP server used for signalling
- Authentication (local users)
- Plugins

Jitsi Gateway to SIP (jigasi)

server-side application that allows regular SIP clients to join Jitsi Meet conferences

Jitsi Broadcasting Infrastructure (jibri)

set of tools for recording and/or streaming a Jitsi Meet conference that works by launching a Chrome instance rendered in a virtual framebuffer and capturing and encoding the output with ffmpeg.



Ports

- IN tcp/80
- IN tcp/443
- IN udp/10000
- OUT udp/443

Quick deploy

- Using docker-compose on <https://github.com/jitsi/docker-jitsi-meet>

Quicker deploy

- Ansible playbook doing everything for you:
<https://git.lattuga.net/panda/jitsi-quick>
 - Installs docker
 - Installs and configures telegraf to send jitsi stats to influx
 - Pulls non-test release of docker-jitsi-meet
 - No-auth = public access
 - Tested on debian11 & centos7

Tips & tricks

- Use the correct tag when pulling docker-jitsi-meet:
the default is the dev/test tag, you have to specify a release to use the stable one.

Using plugins (docker-compose)

- To add a plugin:
 - In your env file add:
`XMPP_MUC_MODULES=your_plugin_name`
 - Stop everything: `docker-compose stop`
 - Chown 101 your plugin.lua and add it in `.jitsi-meet-cfg/prosody/prosody-plugins-custom/`
 - Start everything: `docker-compose start`

Quicker-deploy demo time?

Sizing

- For small videocalls (5ppl) you won't need a big machine:
1/2core 4gb ram
- Always consider the bandwidth
- Official guideline: <https://jitsi.org/jitsi-videobridge-performance-evaluation/>
they say baremetal 2013 Xeon quad core and 16gb (?) of ram supports 1000 video streams at 550mbps and 20% cpu

Documentation

- It's ok for easy deployments
- It's lacking for more complex ones
- The last changes are more commonly found in forum posts instead of official docs

Relays (aka secure octo)

- Cascading bridges
- use ICE and DTLS/SRTP between each pair of bridges, so a secure network is not required. It uses and requires colibri websockets for the bridge-bridge connections

/etc/jitsi/videobridge/sip-communicator.properties

- org.jitsi.videobridge.ENABLE_STATISTICS=true
- org.jitsi.videobridge.STATISTICS_TRANSPORT=muc
- org.jitsi.videobridge.STATISTICS_INTERVAL=5000
- org.jitsi.videobridge.xmpp.user.shard.HOSTNAME=videocitofono.bida.im
- org.jitsi.videobridge.xmpp.user.shard.DOMAIN=auth.videocitofono.bida.im
- org.jitsi.videobridge.xmpp.user.shard.USERNAME=jvb
- org.jitsi.videobridge.xmpp.user.shard.PASSWORD=vT477ggYHYU
- org.jitsi.videobridge.xmpp.user.shard.MUC_JIDS=JvbBrewery@internal.auth.videocitofono.bida.im
- org.jitsi.videobridge.xmpp.user.shard.MUC_NICKNAME=d4393752-0bd2-4273-ba7c-03a31d50c05a
- org.jitsi.videobridge.xmpp.user.shard.DISABLE_CERTIFICATE_VERIFICATION=true
- org.jitsi.videobridge.rest.jetty.port=9090
- org.jitsi.videobridge.rest.COLIBRI_WS_TLS=true
- org.jitsi.videobridge.rest.COLIBRI_WS_DOMAIN=videocitofono.bida.im:443

Split Strategy

- `SingleBridgeSelectionStrategy`: Use the least loaded bridge, do not split a conference between bridges (Octo).
- `SplitBridgeSelectionStrategy`: Use a separate bridge for each participant (for testing).
- `RegionBasedBridgeSelectionStrategy`: Attempt to put each participant in a bridge in their local region (i.e. use Octo for geo-location).
- `IntraRegionBridgeSelectionStrategy`: Use additional bridges when a bridge becomes overloaded (i.e. use Octo for load balancing).

Load average

- `max-bridge-participants = -1`
- `// The assumed maximum packet rate that a bridge can handle.`
- `max-bridge-packet-rate = 50000`
- `// The assumed average packet rate per participant.`
- `average-participant-packet-rate-pps = 500`
- `// The default assumed average stress per participant. This value is only used when a bridge does not report its`
- `// own value.`
- `average-participant-stress = 0.01`
- `// The assumed time that an endpoint takes to start contributing fully to the load on a bridge. To avoid allocating`
- `// a burst of endpoints to the same bridge, the bridge stress is adjusted by adding the number of new endpoints`
- `// in the last [participant-rampup-time] multiplied by [average-participant-stress].`
- `participant-rampup-interval = 20 seconds`
- `// The stress level above which a bridge is considered overstressed.`
- `stress-threshold = 0.8`
- `// The amount of to wait before retrying using a failed bridge.`
- `failure-reset-threshold = 1 minute`

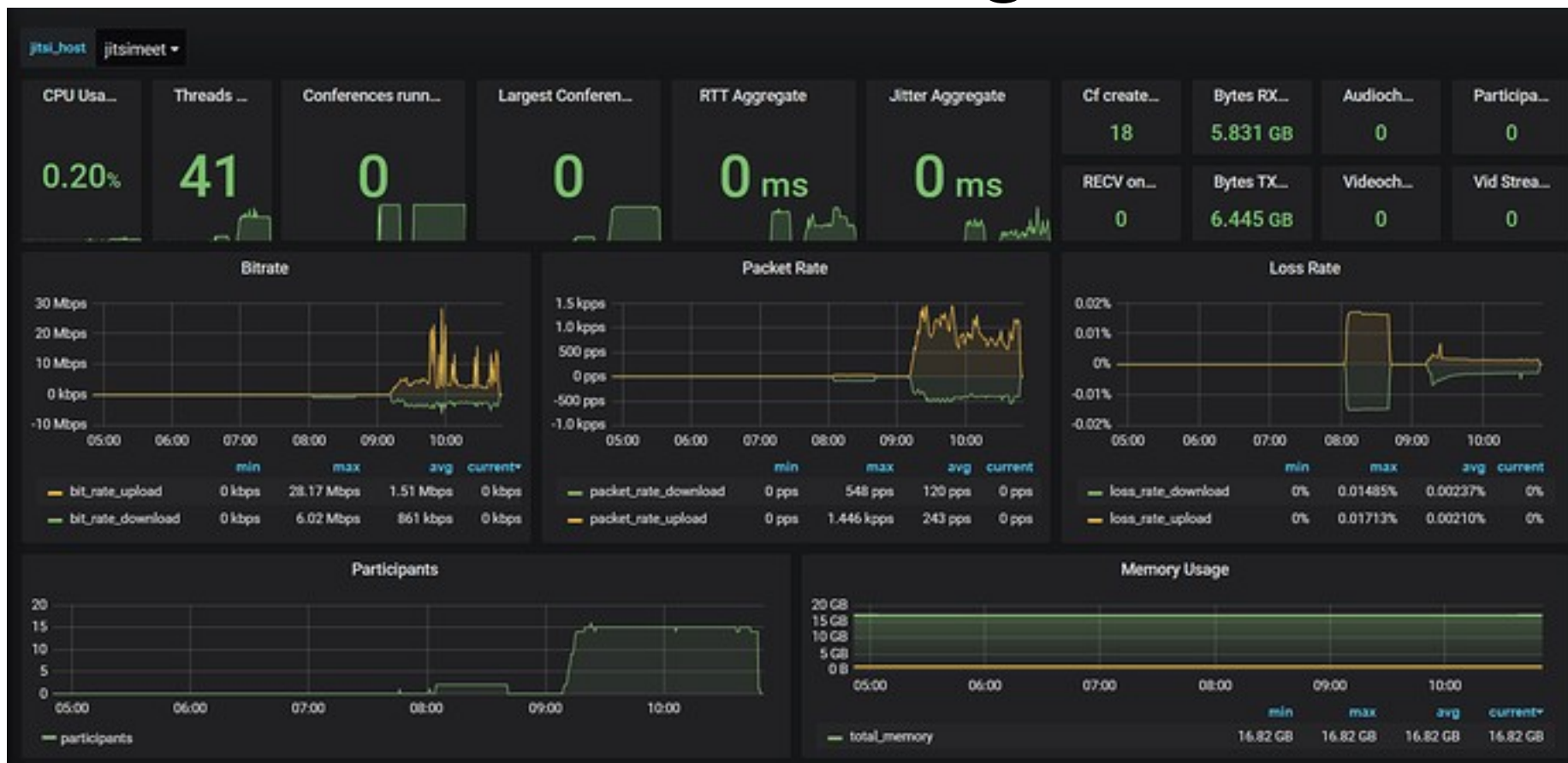
Stress Test

- <https://github.com/jitsi/jitsi-meet-torture>
- `./scripts/malleus.sh`
`--instance-url='https://videocitofono.bida.im' --`
`conferences=1 --participants=20 --senders=20 --audio-`
`senders=20 --duration=120 --room-name-prefix=TEST\n`

Stress Test



Monitoring



.. in progress

Grazie