ENS 491-492 – Graduation Project
Final Report

Project Title: Self-Hosted Video Conferencing

Group Member: Mert Ali Çelik 26958

Supervisor(s): Ayhan Bozkurt

Date: 26.05.2024

1. EXECUTIVE SUMMARY

The focus of this project is to meet the growing need for video conferencing systems that are secure, dependable, and affordable, with a special emphasis on the challenges brought about by the COVID-19 epidemic. The dominant options in the industry, including Zoom, Microsoft Teams, Google Meet, and Jitsi Meet are widely used yet frequently cause problems over privacy and data security. The objective of this project is to create an open-source video conferencing software that provides an efficient alternative, with a focus on safeguarding user privacy and security.

I started this project by exploring and integrating an open-source program known as Jitsi Meet. The purpose of this task was to find software that could be modified to provide improved privacy settings, strong data security, flexible performance, and effective memory management. The decision to use the Linux operating system as the hosting platform was crucial since it provided a safe and adaptable environment for software development and testing.

The technique entailed a thorough examination of the chosen open-source tools, followed by an elaborate procedure of creation and testing. My primary objective was to design an interface that uses Jitsi Meet tools efficiently, while also guaranteeing that the program can handle the demands of large-scale group conversations. A comprehensive evaluation was carried out to determine the dependability and efficiency of the program across different scenarios.

The main conclusions derived from this experiment demonstrate the possibility of developing a reliable and effective open-source video conferencing application that rivals commercial offerings. The created program provides a robust solution for customers that prioritize privacy and data protection, while yet delivering the expected functions of a contemporary video conferencing application. This research highlights the capacity of open-source technology to tackle existing digital communication difficulties and paves the way for future advancements in this field.

2. PROBLEM STATEMENT

Original Problem

The primary concern that this project addresses is the increased need for video conferencing solutions that experienced a significant rise during the COVID-19 epidemic. Although commercial services like as Zoom, Microsoft Teams, Google Meet, and Jitsi Meet are readily accessible, there are ongoing worries surrounding their privacy and data security capabilities. The objective of this project is to create a safe, efficient, and open-source alternative that effectively deals with these challenges.

Motivation and Goals

I am highly motivated by a strong passion for improving the privacy and security of digital communication. Upon witnessing the rapid integration of video conferencing technologies and the accompanying hazards, I acknowledged the necessity for a more secured and dependable platform. The objective was to utilize open-source technology in order to construct a video conferencing solution that prioritizes user privacy and data security, while maintaining functionality and user-friendliness. This project aspires to contribute to the existing body of work by developing a tool that not only rivals commercial offerings in terms of performance and features but also surpasses them in terms of security and privacy safeguards.

Objectives/Tasks

1) **Development of the Software:** Utilizing open-source tools such as Jitsi Meet and eduMeet to create a robust video conferencing software. The intended result is a secure, scalable, and user-friendly platform.
2) **Obtaining a Cloud Hosting and a Domain Name:** Buying a cloud hosting and domain name for the website.
3) **Cost and Comparative Analysis:** Comparing the developed software with commercial counterparts to assess its cost-effectiveness and competitive edge.
4) **Testing for Scalability and Memory Management:** Ensuring the software can handle multiple sessions and large groups effectively, without compromising performance.

Realistic Constraints

1) **Technical Constraints:** Given the reliance on open-source tools, technical constraints such as compatibility and integration challenges were anticipated. I addressed these by carefully selecting compatible software components and focusing on effective integration strategies.
2) **Resource Limitations:** Operating within a limited budget and time frame required efficient management of resources. This was managed by prioritizing essential features and leveraging free, open-source tools.
3) **Compliance with Standards:** The project required adherence to specific software development standards, particularly those related to data security and privacy. I ensured compliance by staying updated with current best practices in software development and implementing rigorous testing protocols.
4) **Skillset and Knowledge:** The need for expertise in Linux and various programming languages posed a constraint. I addressed this by dedicating time to learning and applying new skills pertinent to the project requirements. I mostly searched for new tools to analyze my Jitsi Meet server.

By meticulously planning and adapting to these constraints, the project aims to deliver a video conferencing tool that is not only functional and secure but also accessible to a wide range of users with high performance.

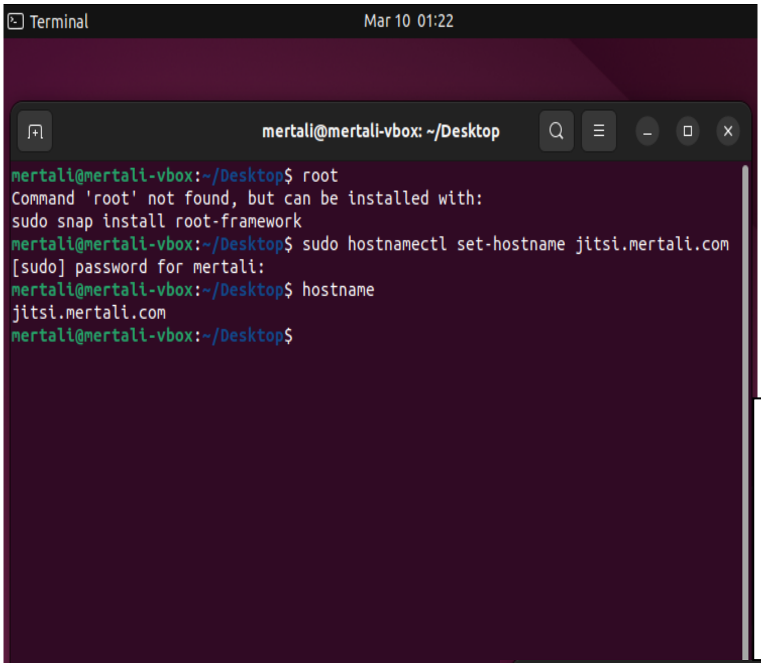
3. METHODOLOGY

The methodology for this project was a multi-step process, focusing on the development, adaptation, and testing of open-source video conferencing software, Jitsi Meet. The approach combined practical implementation with iterative testing to ensure the software's functionality, security, and user-friendliness.

Selection and Cloning of Open-Source Software

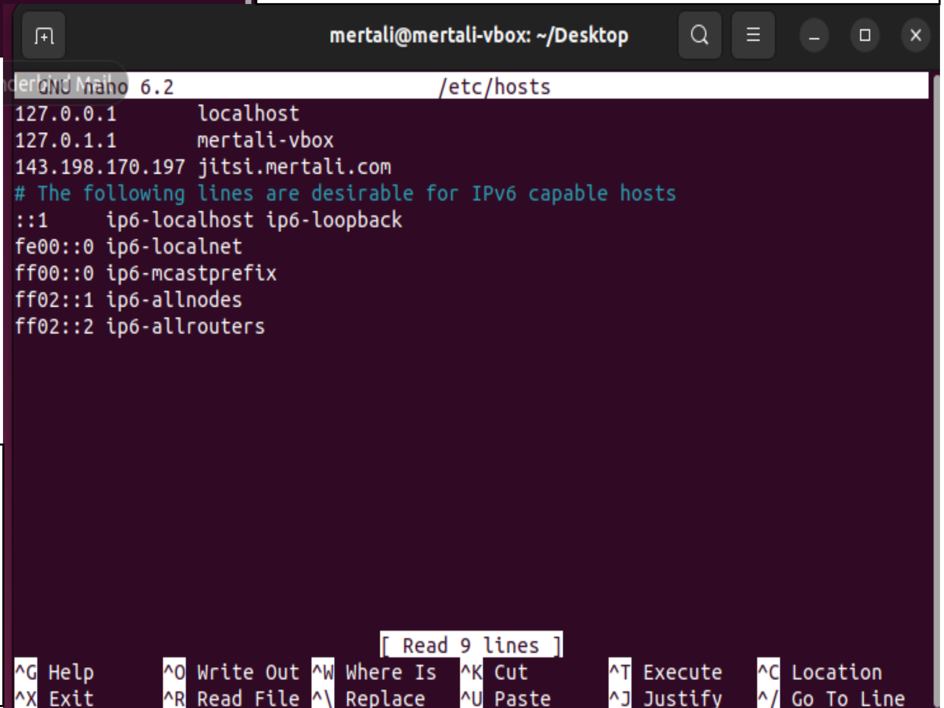
The initial step involved selecting an appropriate open-source platform as the base for the project. I chosed Jitsi Meet for its robust features and compatibility with the project's objectives. Jitsi Meet has a very detailed webpage with necessary guide to download the basic Jitsi Meet open source software if you have a cloud hosting adress and a domain name.

Installation Screenshots



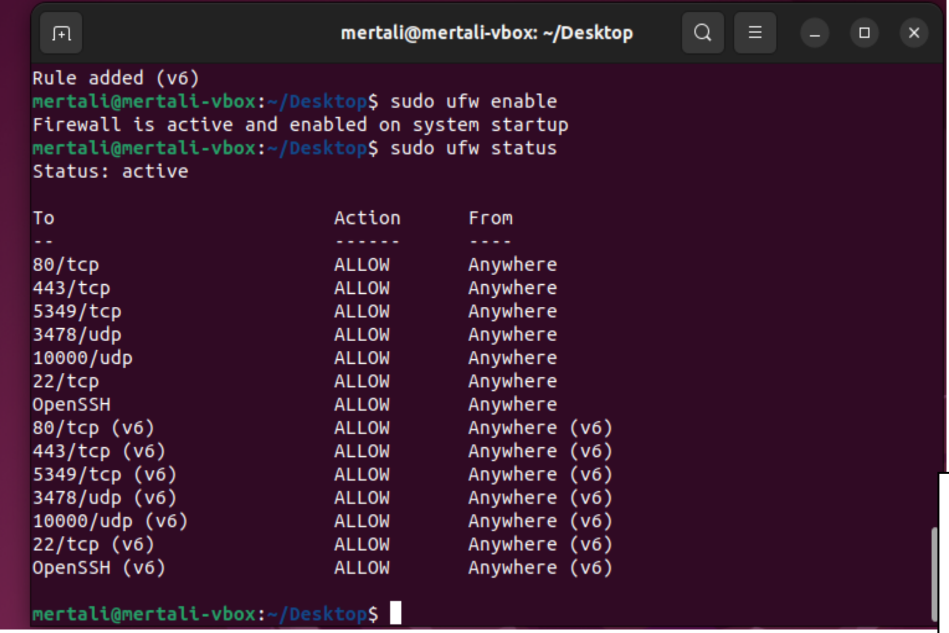
```
mertali@mertali-vbox:~/Desktop$ root
Command 'root' not found, but can be installed with:
sudo snap install root-framework
mertali@mertali-vbox:~/Desktop$ sudo hostnamectl set-hostname jitsi.mertali.com
[sudo] password for mertali:
mertali@mertali-vbox:~/Desktop$ hostname
jitsi.mertali.com
mertali@mertali-vbox:~/Desktop$
```

I used hostnamectl with sudo access to create a host domain name for my app. In this screenshot, I used jitsi.mertali.com domain name on my local privately to test the environment.



```
 GNU nano 6.2                        /etc/hosts
127.0.0.1       localhost
127.0.1.1       mertali-vbox
143.198.170.197 jitsi.mertali.com
# The following lines are desirable for IPv6 capable hosts
::1     ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters

                              [ Read 9 lines ]
^G Help       ^O Write Out  ^W Where Is   ^K Cut        ^T Execute    ^C Location
^X Exit       ^R Read File  ^\ Replace    ^U Paste      ^J Justify    ^/ Go To Line
```

Then I save my server IP address to /etc/hosts directory to reach it.

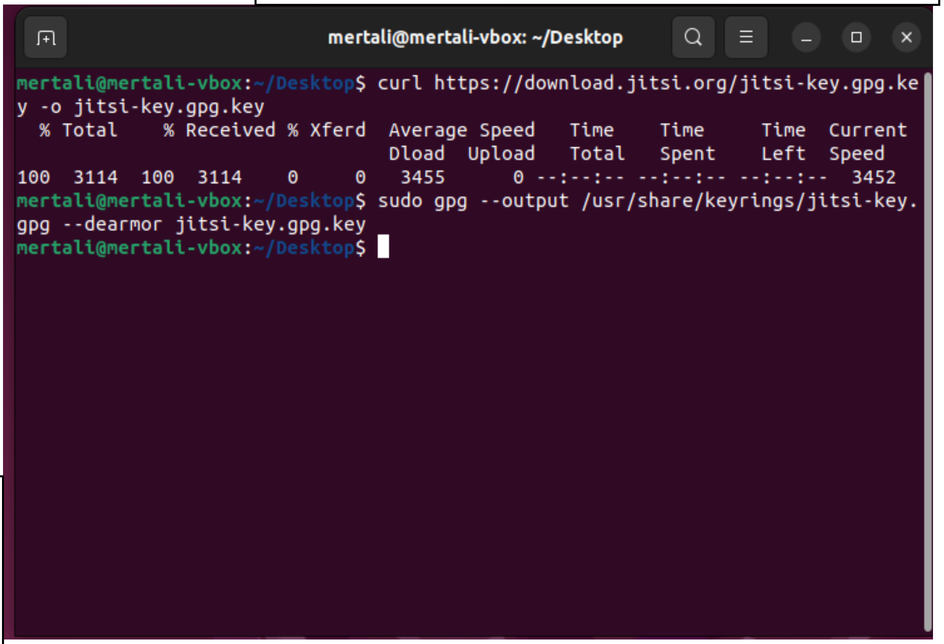


```
Rule added (v6)
mertali@mertali-vbox:~/Desktop$ sudo ufw enable
Firewall is active and enabled on system startup
mertali@mertali-vbox:~/Desktop$ sudo ufw status
Status: active

To                         Action      From
--                         ------      ----
80/tcp                     ALLOW       Anywhere
443/tcp                    ALLOW       Anywhere
5349/tcp                   ALLOW       Anywhere
3478/udp                   ALLOW       Anywhere
10000/udp                  ALLOW       Anywhere
22/tcp                     ALLOW       Anywhere
OpenSSH                    ALLOW       Anywhere
80/tcp (v6)                ALLOW       Anywhere (v6)
443/tcp (v6)               ALLOW       Anywhere (v6)
5349/tcp (v6)              ALLOW       Anywhere (v6)
3478/udp (v6)              ALLOW       Anywhere (v6)
10000/udp (v6)             ALLOW       Anywhere (v6)
22/tcp (v6)                ALLOW       Anywhere (v6)
OpenSSH (v6)               ALLOW       Anywhere (v6)

mertali@mertali-vbox:~/Desktop$ 
```

Then I enabled the important ports such as: OpenSSH, and 80tcp for Jitsi Meet to work efficiently with its tools.

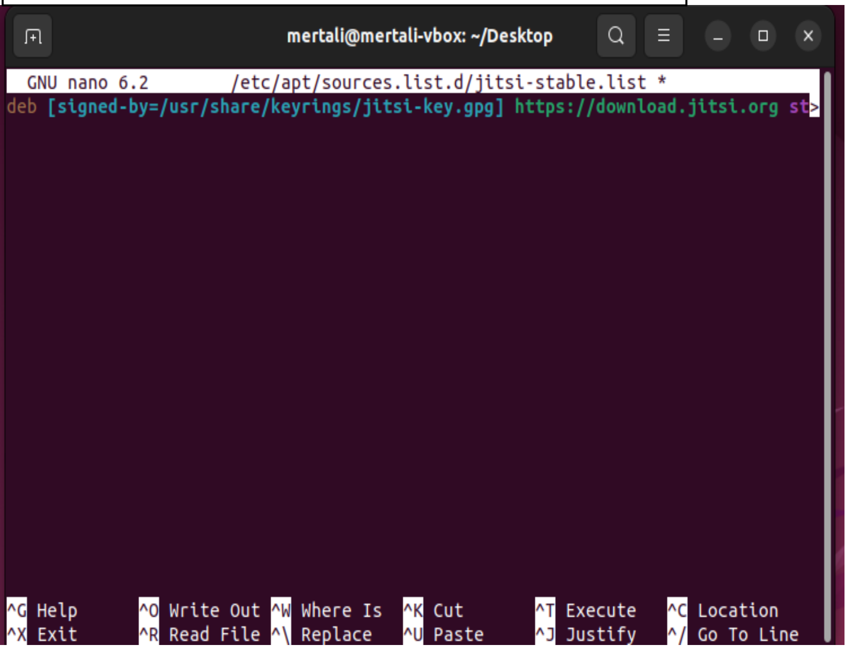
Sudo ufw enable to enable firewall.

Sudo ufw status to see firewall status.



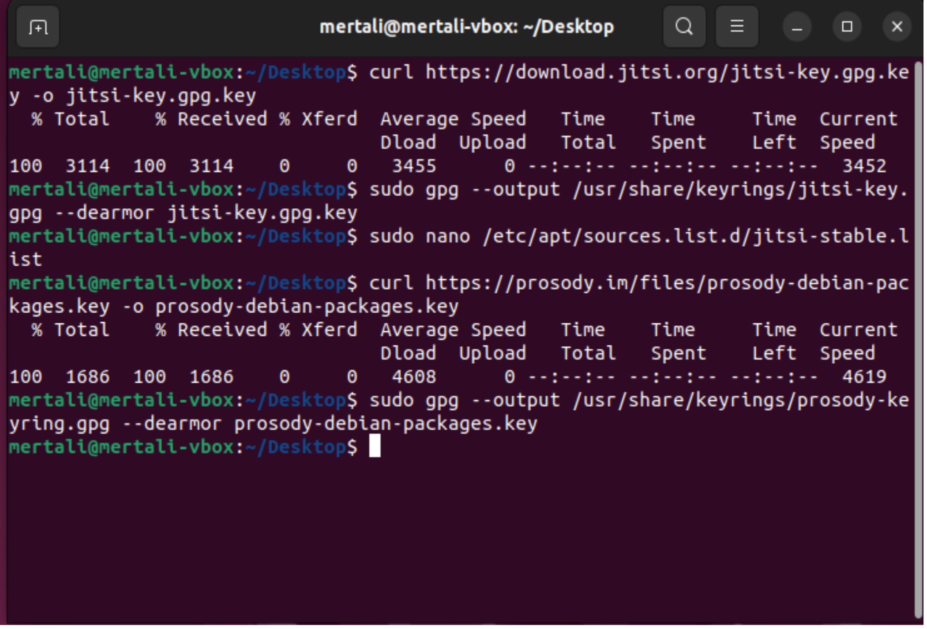
```
mertali@mertali-vbox:~/Desktop$ curl https://download.jitsi.org/jitsi-key.gpg.ke
y -o jitsi-key.gpg.key
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100  3114  100  3114    0     0   3455      0 --:--:-- --:--:-- --:--:--  3452
mertali@mertali-vbox:~/Desktop$ sudo gpg --output /usr/share/keyrings/jitsi-key.
gpg --dearmor jitsi-key.gpg.key
mertali@mertali-vbox:~/Desktop$ 
```

Downloaded the Jitsi GPG key with curl command and added key to my system.



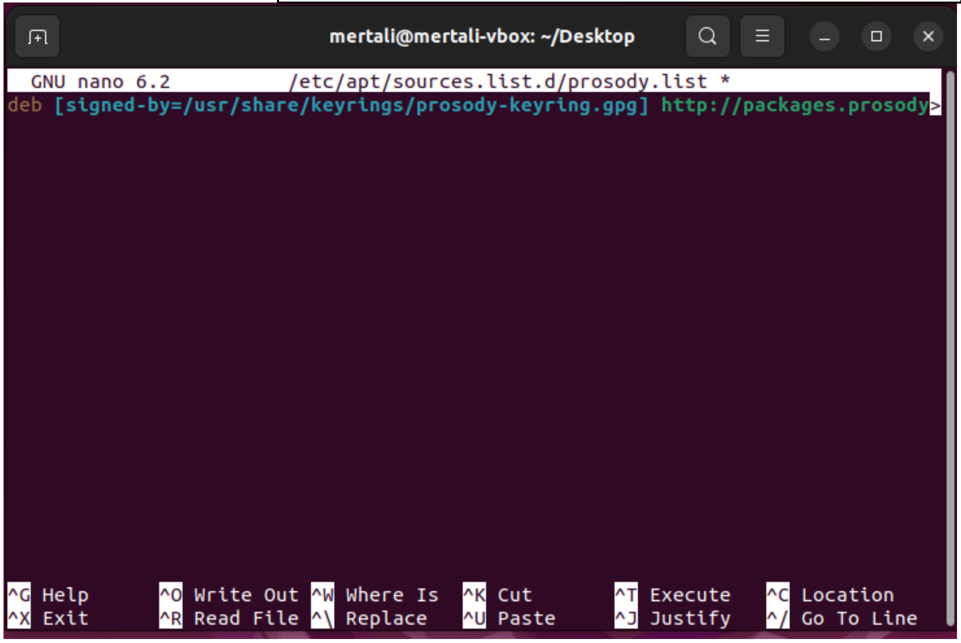
```
  GNU nano 6.2         /etc/apt/sources.list.d/jitsi-stable.list *
deb [signed-by=/usr/share/keyrings/jitsi-key.gpg] https://download.jitsi.org st>
```

Added Jitsi repository with the following line.

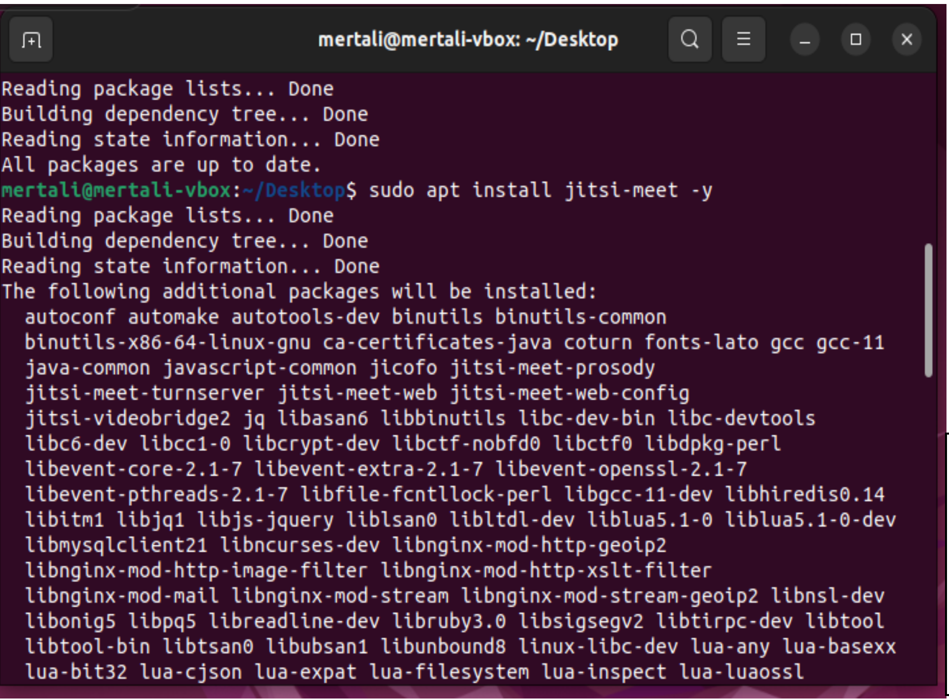


```
mertali@mertali-vbox:~/Desktop$ curl https://download.jitsi.org/jitsi-key.gpg.ke
y -o jitsi-key.gpg.key
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100  3114  100  3114    0     0   3455      0 --:--:-- --:--:-- --:--:--  3452
mertali@mertali-vbox:~/Desktop$ sudo gpg --output /usr/share/keyrings/jitsi-key.
gpg --dearmor jitsi-key.gpg.key
mertali@mertali-vbox:~/Desktop$ sudo nano /etc/apt/sources.list.d/jitsi-stable.l
ist
mertali@mertali-vbox:~/Desktop$ curl https://prosody.im/files/prosody-debian-pac
kages.key -o prosody-debian-packages.key
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100  1686  100  1686    0     0   4608      0 --:--:-- --:--:-- --:--:--  4619
mertali@mertali-vbox:~/Desktop$ sudo gpg --output /usr/share/keyrings/prosody-ke
yring.gpg --dearmor prosody-debian-packages.key
mertali@mertali-vbox:~/Desktop$ 
```

Downloaded and followed same steps for the Prosody GPG key too.

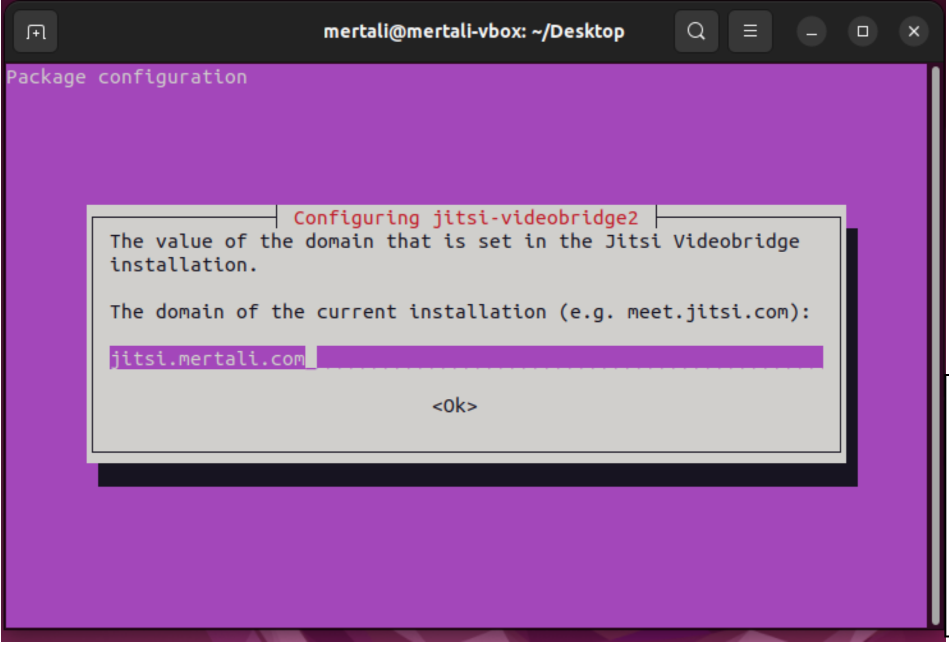


```
  GNU nano 6.2           /etc/apt/sources.list.d/prosody.list *
deb [signed-by=/usr/share/keyrings/prosody-keyring.gpg] http://packages.prosody>









^G Help      ^O Write Out ^W Where Is  ^K Cut       ^T Execute   ^C Location
^X Exit      ^R Read File ^\ Replace   ^U Paste     ^J Justify   ^/ Go To Line
```



```
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
All packages are up to date.
mertali@mertali-vbox:~/Desktop$ sudo apt install jitsi-meet -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  autoconf automake autotools-dev binutils binutils-common
  binutils-x86-64-linux-gnu ca-certificates-java coturn fonts-lato gcc gcc-11
  java-common javascript-common jicofo jitsi-meet-prosody
  jitsi-meet-turnserver jitsi-meet-web jitsi-meet-web-config
  jitsi-videobridge2 jq libasan6 libbinutils libc-dev-bin libc-devtools
  libc6-dev libcc1-0 libcrypt-dev libctf-nobfd0 libctf0 libdpkg-perl
  libevent-core-2.1-7 libevent-extra-2.1-7 libevent-openssl-2.1-7
  libevent-pthreads-2.1-7 libfile-fcntllock-perl libgcc-11-dev libhiredis0.14
  libitm1 libjq1 libjs-jquery liblsan0 libltdl-dev liblua5.1-0 liblua5.1-0-dev
  libmysqlclient21 libncurses-dev libnginx-mod-http-geoip2
  libnginx-mod-http-image-filter libnginx-mod-http-xslt-filter
  libnginx-mod-mail libnginx-mod-stream libnginx-mod-stream-geoip2 libnsl-dev
  libonig5 libpq5 libreadline-dev libruby3.0 libsigsegv2 libtirpc-dev libtool
  libtool-bin libtsan0 libubsan1 libunbound8 linux-libc-dev lua-any lua-basexx
  lua-bit32 lua-cjson lua-expat lua-filesystem lua-inspect lua-luaossl
```

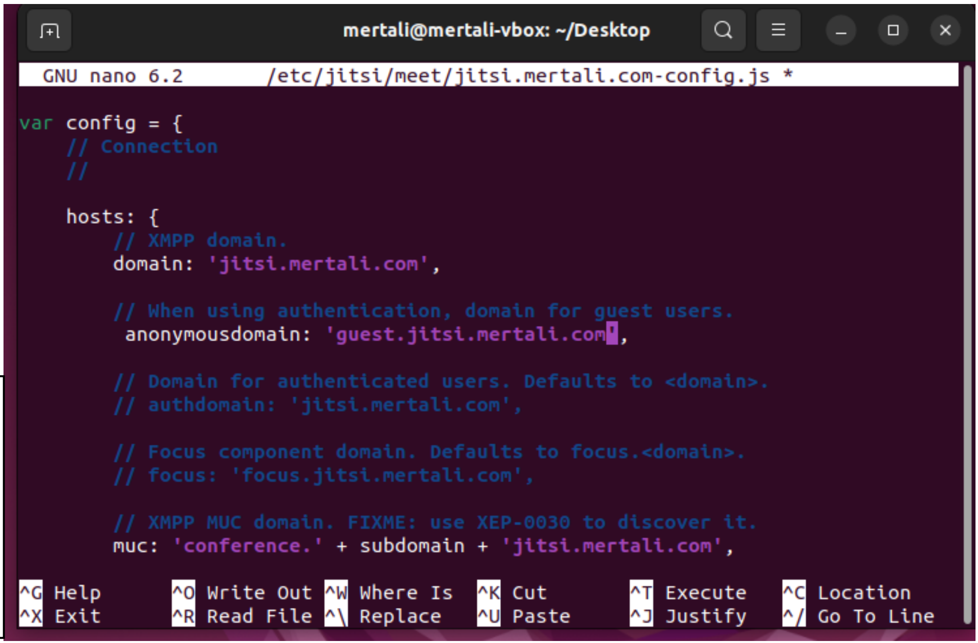
Then I refreshed my repository with a very general command sudo apt update and started to install Jitsi.



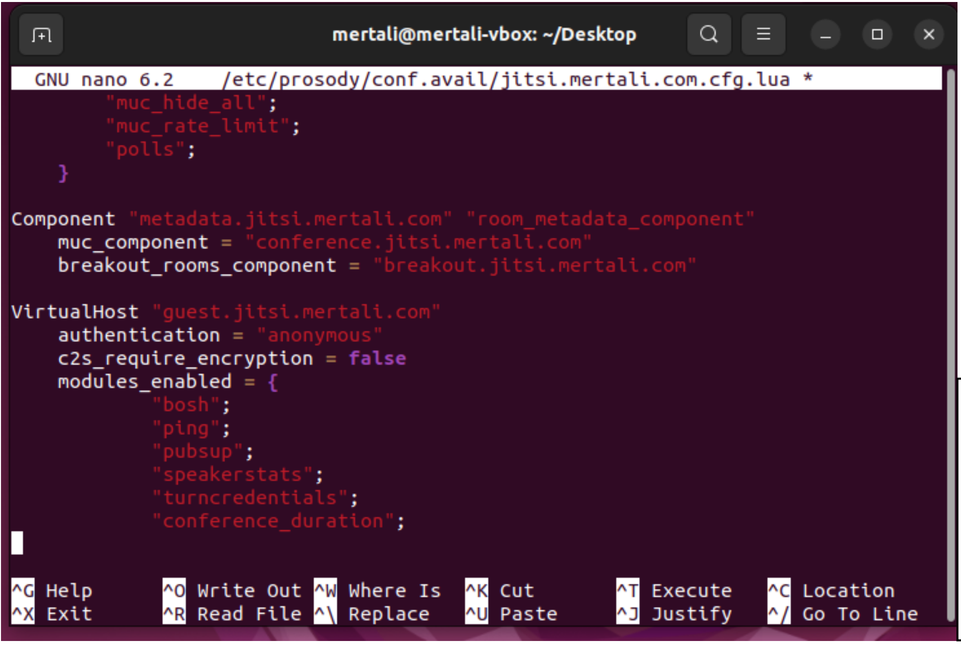
```
Package configuration

Configuring jitsi-videobridge2
The value of the domain that is set in the Jitsi Videobridge
installation.

The domain of the current installation (e.g. meet.jitsi.com):

jitsi.mertali.com
```

Then I added my test domain name to my Jitsi server. Later on, I did it on public domain with all the cloud hosting setup on live.jitsi.mertaliclk.com.tr server.

Configuration setup from config JavaScript files to create a domain link for guests.

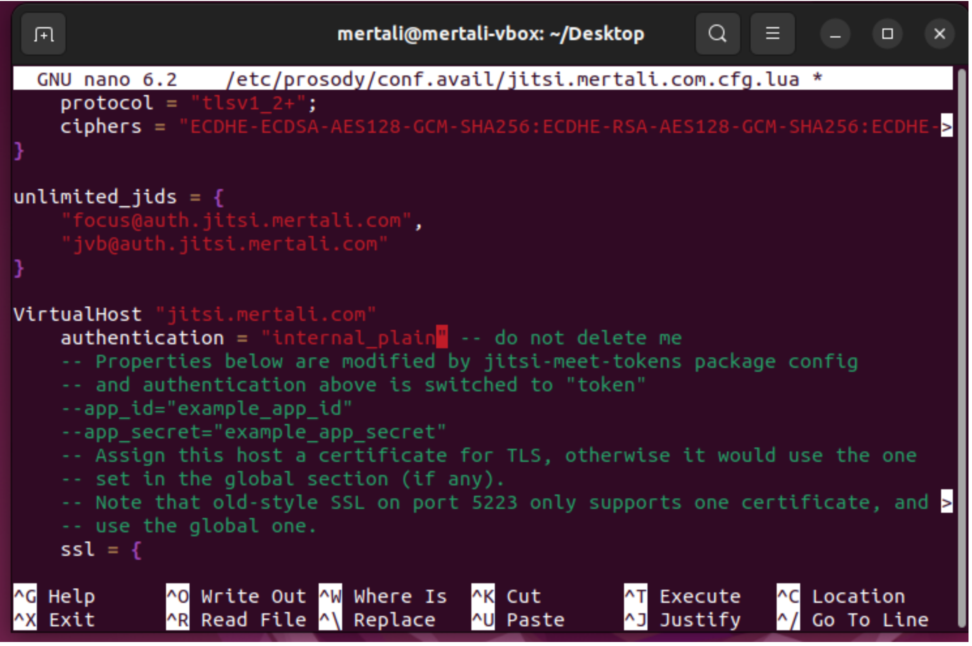


```
  GNU nano 6.2         /etc/jitsi/meet/jitsi.mertali.com-config.js *

var config = {
    // Connection
    //

    hosts: {
        // XMPP domain.
        domain: 'jitsi.mertali.com',

        // When using authentication, domain for guest users.
         anonymousdomain: 'guest.jitsi.mertali.com',

        // Domain for authenticated users. Defaults to <domain>.
        // authdomain: 'jitsi.mertali.com',

        // Focus component domain. Defaults to focus.<domain>.
        // focus: 'focus.jitsi.mertali.com',

        // XMPP MUC domain. FIXME: use XEP-0030 to discover it.
        muc: 'conference.' + subdomain + 'jitsi.mertali.com',
```

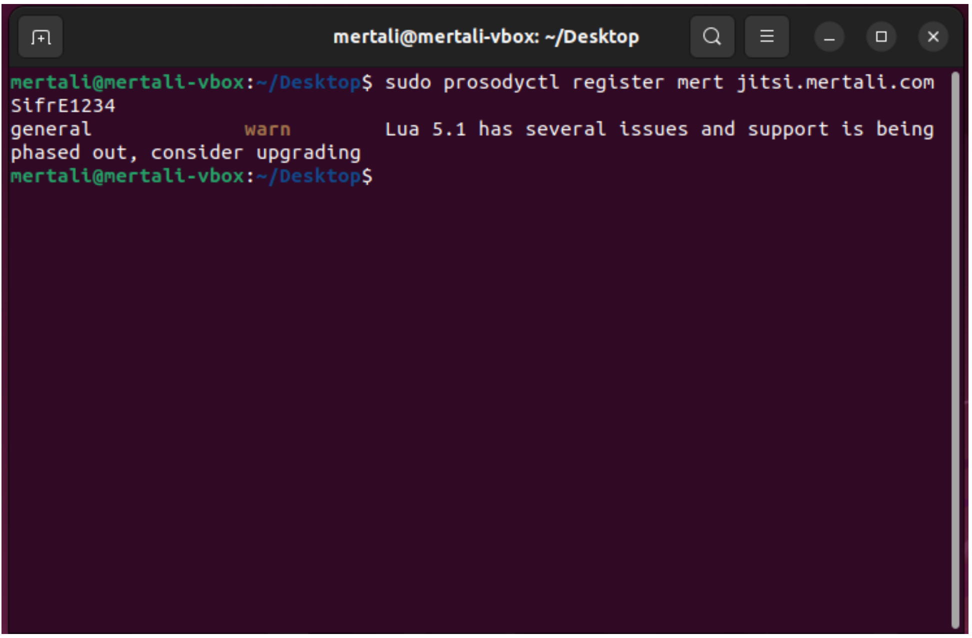


```
  GNU nano 6.2     /etc/prosody/conf.avail/jitsi.mertali.com.cfg.lua *
        "muc_hide_all";
        "muc_rate_limit";
        "polls";
    }

Component "metadata.jitsi.mertali.com" "room_metadata_component"
    muc_component = "conference.jitsi.mertali.com"
    breakout_rooms_component = "breakout.jitsi.mertali.com"

VirtualHost "guest.jitsi.mertali.com"
    authentication = "anonymous"
    c2s_require_encryption = false
    modules_enabled = {
            "bosh";
            "ping";
            "pubsup";
            "speakerstats";
            "turncredentials";
            "conference_duration";
```

Created a Virtual Host variable for guests with several modules enabled. I searched the modules from internet then I used these models with different usages.

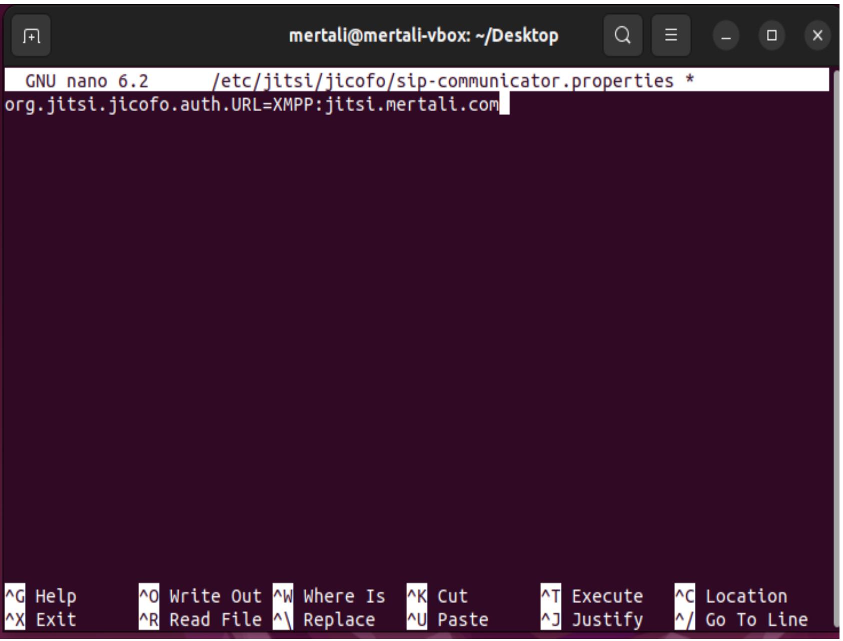


```
/etc/prosody/conf.avail/jitsi.mertali.com.cfg.lua *
    protocol = "tlsv1_2+";
    ciphers = "ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-RSA-AES128-GCM-SHA256:ECDHE->
}

unlimited_jids = {
    "focus@auth.jitsi.mertali.com",
    "jvb@auth.jitsi.mertali.com"
}

VirtualHost "jitsi.mertali.com"
    authentication = "internal_plain" -- do not delete me
    -- Properties below are modified by jitsi-meet-tokens package config
    -- and authentication above is switched to "token"
    --app_id="example_app_id"
    --app_secret="example_app_secret"
    -- Assign this host a certificate for TLS, otherwise it would use the one
    -- set in the global section (if any).
    -- Note that old-style SSL on port 5223 only supports one certificate, and >
    -- use the global one.
    ssl = {
```



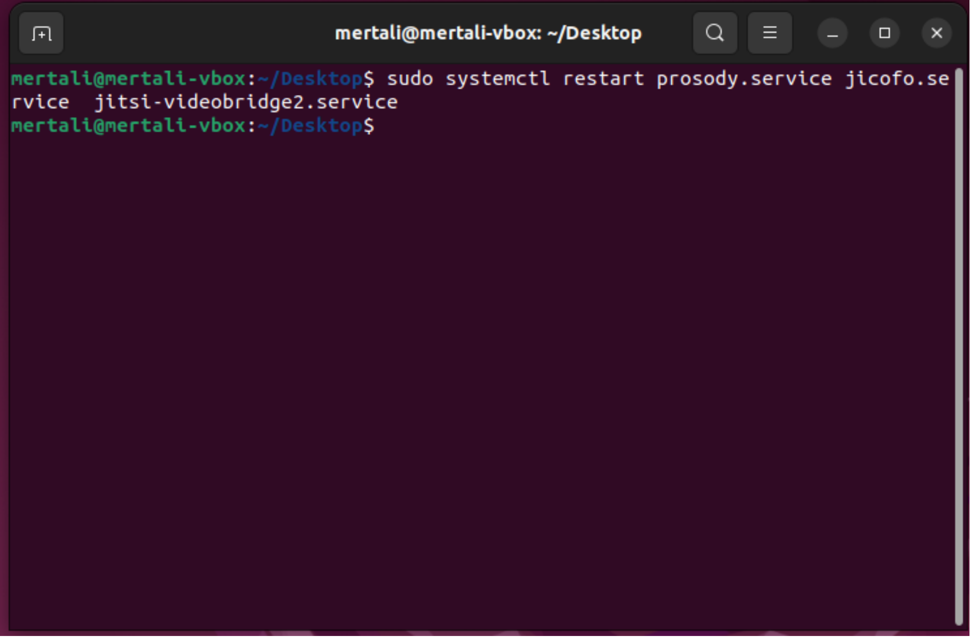
```
mertali@mertali-vbox:~/Desktop$ sudo prosodyctl register mert jitsi.mertali.com
SifrE1234
general                 warn        Lua 5.1 has several issues and support is being
phased out, consider upgrading
mertali@mertali-vbox:~/Desktop$
```

Registered an Admin account for my test server to gain more security and monitoring advantages.



```
/etc/jitsi/jicofo/sip-communicator.properties *
org.jitsi.jicofo.auth.URL=XMPP:jitsi.mertali.com
```

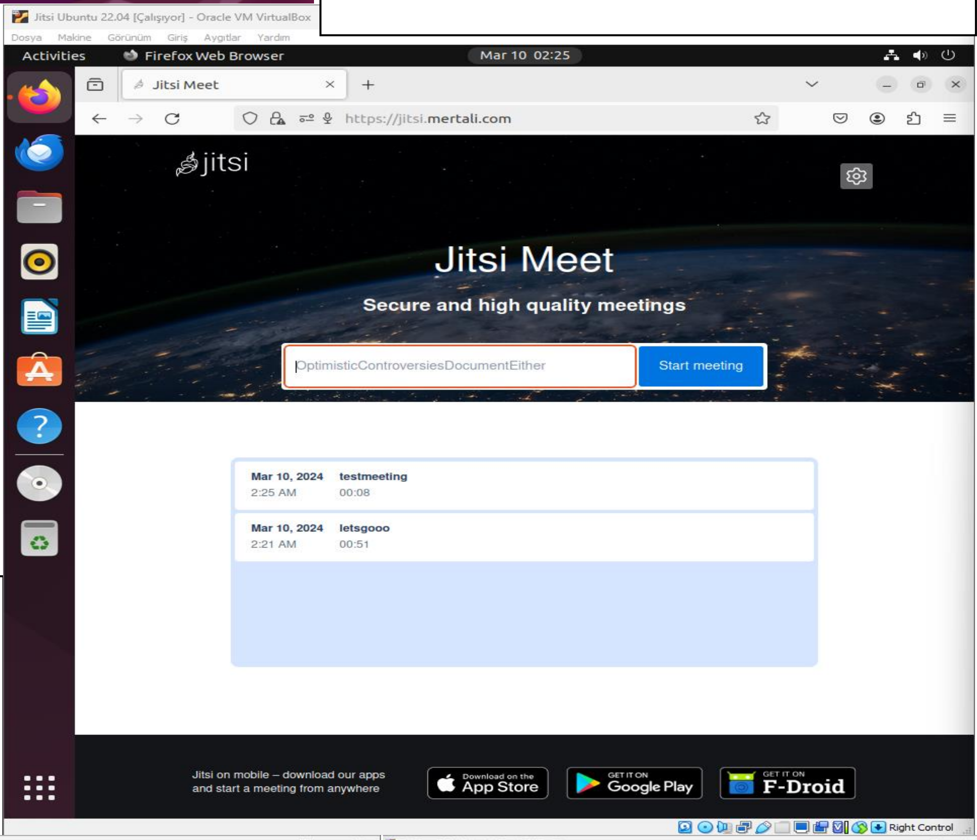
Set up configurations for Jicofo component which is a useful tool of Jitsi software which controls conference, participant authentication, media session, health monitoring, and more.



```
mertali@mertali-vbox:~/Desktop$ sudo systemctl restart prosody.service jicofo.se
rvice  jitsi-videobridge2.service
mertali@mertali-vbox:~/Desktop$
```

I restarted all the systems and services that I downloaded to start my test server.

Then I saw the following page on my localhost.



Jitsi Ubuntu 22.04 [Çalışıyor] - Oracle VM VirtualBox
Activities
Firefox Web Browser
Mar 10 02:25
Jitsi Meet
https://jitsi.mertali.com
jitsi

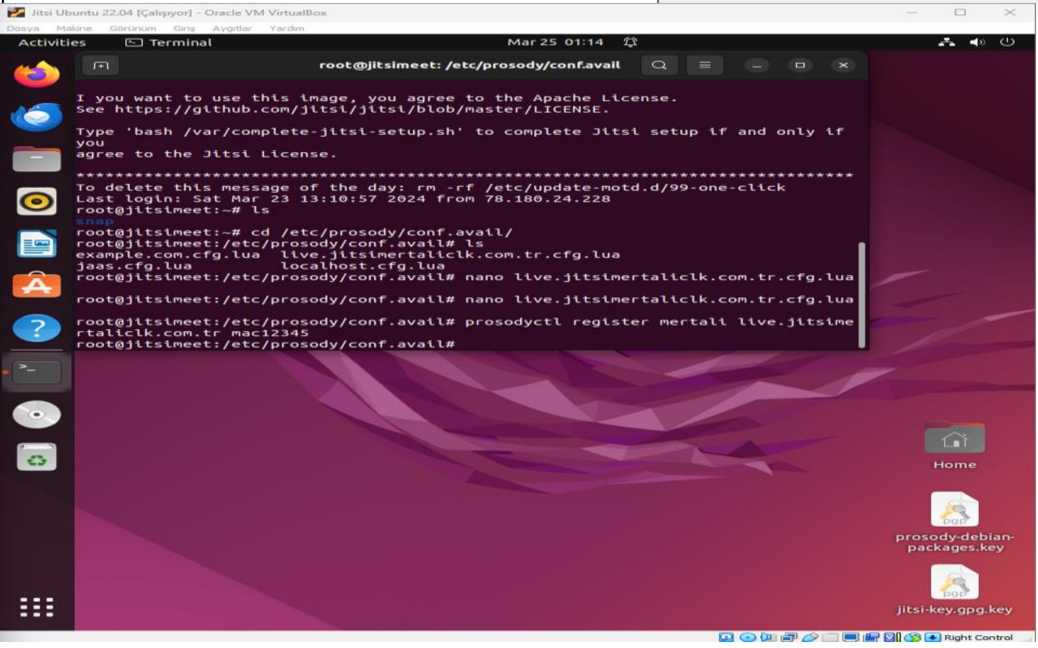
Jitsi Meet

Secure and high quality meetings

OptimisticControversiesDocumentEither | Start meeting

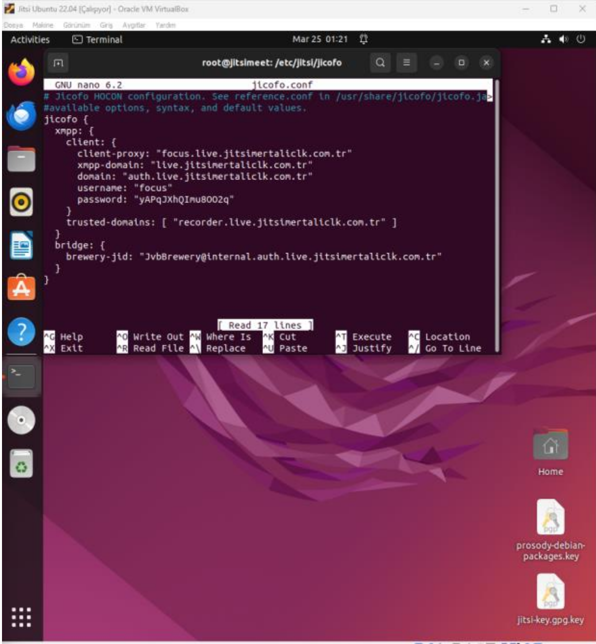
Mar 10, 2024 2:25 AM	testmeeting 00:08
Mar 10, 2024 2:21 AM	letsgooo 00:51

Jitsi on mobile – download our apps and start a meeting from anywhere



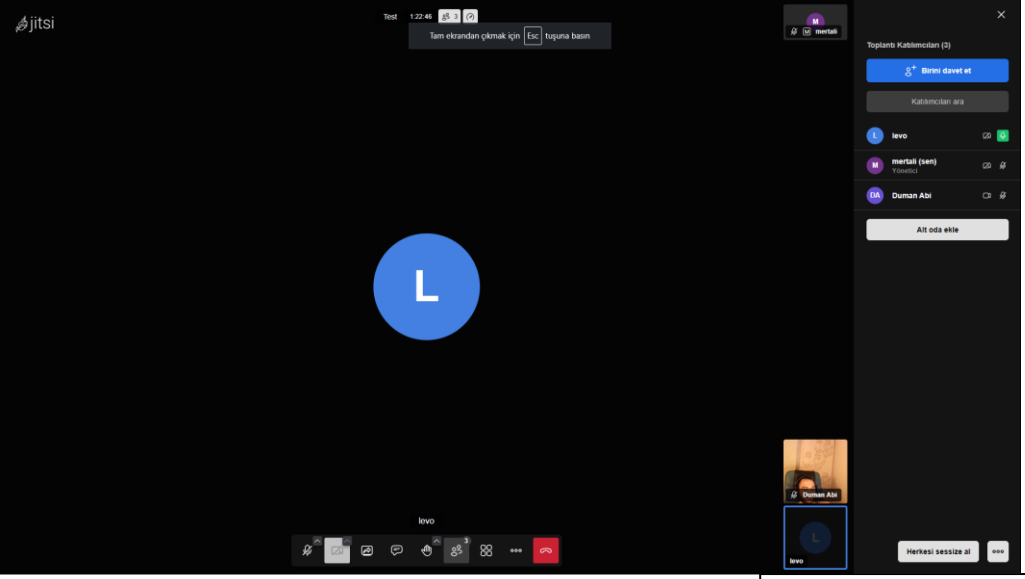
Jitsi Ubuntu 22.04 [Çalışıyor] - Oracle VM VirtualBox
Activities
Terminal
Mar 25 01:14
root@jitsimeet: /etc/prosody/conf.avail
Home
prosody-debian-packages.key
jitsi-key.gpg.key

```
I you want to use this image, you agree to the Apache License.
See https://github.com/jitsi/jitsi/blob/master/LICENSE.

Type 'bash /var/complete-jitsi-setup.sh' to complete Jitsi setup if and only if
you
agree to the Jitsi License.

********************************************************************************
To delete this message of the day: rm -rf /etc/update-motd.d/99-one-click
Last login: Sat Mar 23 13:10:57 2024 from 78.180.24.228
root@jitsimeet:~# ls
snap
root@jitsimeet:~# cd /etc/prosody/conf.avail/
root@jitsimeet:/etc/prosody/conf.avail# ls
example.com.cfg.lua  live.jitsimertaliclk.com.tr.cfg.lua
jaas.cfg.lua         localhost.cfg.lua
root@jitsimeet:/etc/prosody/conf.avail# nano live.jitsimertaliclk.com.tr.cfg.lua

root@jitsimeet:/etc/prosody/conf.avail# nano live.jitsimertaliclk.com.tr.cfg.lua

root@jitsimeet:/etc/prosody/conf.avail# prosodyctl register mertali live.jitsime
rtaliclk.com.tr mac12345
root@jitsimeet:/etc/prosody/conf.avail#
```



Jitsi Ubuntu 22.04 [Çalışıyor] - Oracle VM VirtualBox
Activities
Terminal
Mar 25 01:21
root@jitsimeet: /etc/jitsi/jicofo
Home
prosody-debian-packages.key
jitsi-key.gpg.key

```
  GNU nano 6.2                        jicofo.conf
# Jicofo HOCON configuration. See reference.conf in /usr/share/jicofo/jicofo.ja>
#available options, syntax, and default values.
jicofo {
  xmpp: {
    client: {
      client-proxy: "focus.live.jitsimertaliclk.com.tr"
      xmpp-domain: "live.jitsimertaliclk.com.tr"
      domain: "auth.live.jitsimertaliclk.com.tr"
      username: "focus"
      password: "yAPqJXhQImu8OO2q"
    }
    trusted-domains: [ "recorder.live.jitsimertaliclk.com.tr" ]
  }
  bridge: {
    brewery-jid: "JvbBrewery@internal.auth.live.jitsimertaliclk.com.tr"
  }
}

                              [ Read 17 lines ]
^G Help      ^O Write Out ^W Where Is  ^K Cut       ^T Execute   ^C Location
^X Exit      ^R Read File ^\ Replace   ^U Paste     ^J Justify   ^/ Go To Line
```



After that, I obtained a cloud hosting server from digital ocean for 2 months free and bought a domain name for 70 TL for two years. Then I set an "A Record" for my domain name to connect my cloud host. After these domain and cloud setup, I replayed these steps again to install Jitsi Meet to my server with ssh@ipdress connection from terminal. At the end, I published my website with Jitsi Meet configuration from the following link: live.jitsimertaliclk.com.tr.

Setting Up the Development Environment

I set up a local development environment to test and modify Jitsi Meet. This required configuring a Linux-based system, which was selected for its security and versatility. The installation process was challenging, involving multiple steps:

1) **Installation of Dependencies:** Before installing Jitsi Meet, I installed necessary dependencies, such as specific versions of Node.js, nginx, and other supporting software, ensuring compatibility with Jitsi Meet.
2) **Configuration Adjustments:** Post-cloning, I made several adjustments to the configuration files. This included setting up server parameters, defining secure authentication methods, and adjusting the user interface to better suit the project's needs.
3) **Overcoming Installation Challenges:** The installation process was intricate, involving troubleshooting various issues that arose, such as dependency conflicts and configuration errors. This phase required significant research, trial-and-error, and consultation of community forums and documentation to resolve issues.

Development and Customization

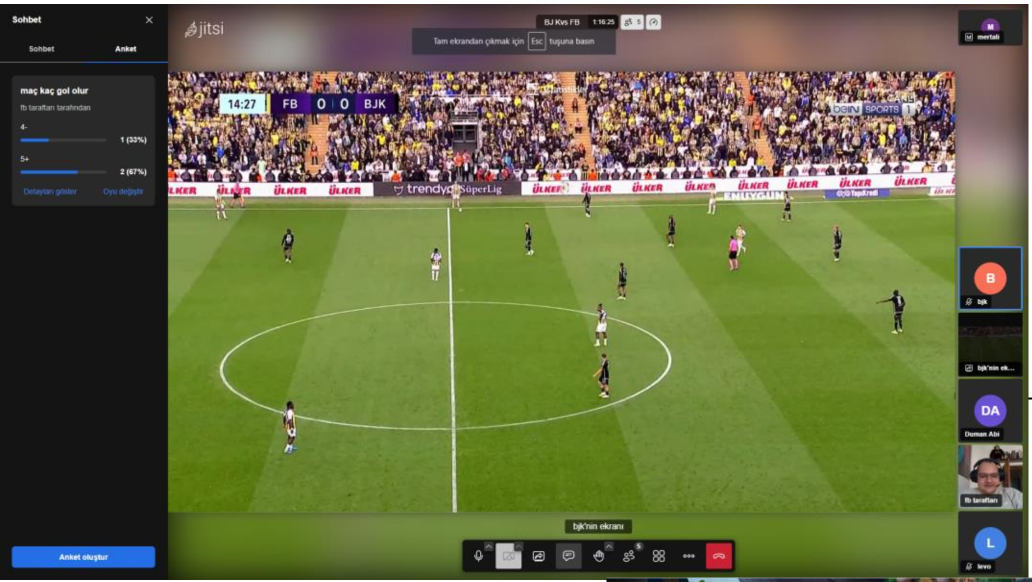
Once Jitsi Meet was successfully installed, the next phase involved customizing and developing additional features:

1) **Matcing Server and Client:** After installing Jitsi software to my ip adress, created a dns record to display it on the internet.
2) **User Interface Customization:** The user interface was modified to enhance usability and incorporate additional privacy features.
3) **Feature Enhancement:** New features, such as shortcut emotes, surveys, and scalable video coding for efficient bandwidth usage, were integrated.

Testing and Iterative Improvements

Testing was a continuous process throughout the project:

1) **Functional Testing:** I conducted thorough testing of all features to ensure they worked as intended.
2) **Performance Testing:** The software was tested under various loads to evaluate its scalability and memory management capabilities.
3) **Security Testing:** Security features were rigorously tested to ensure data protection and user privacy.



Sohbet

maç kaç gol olur

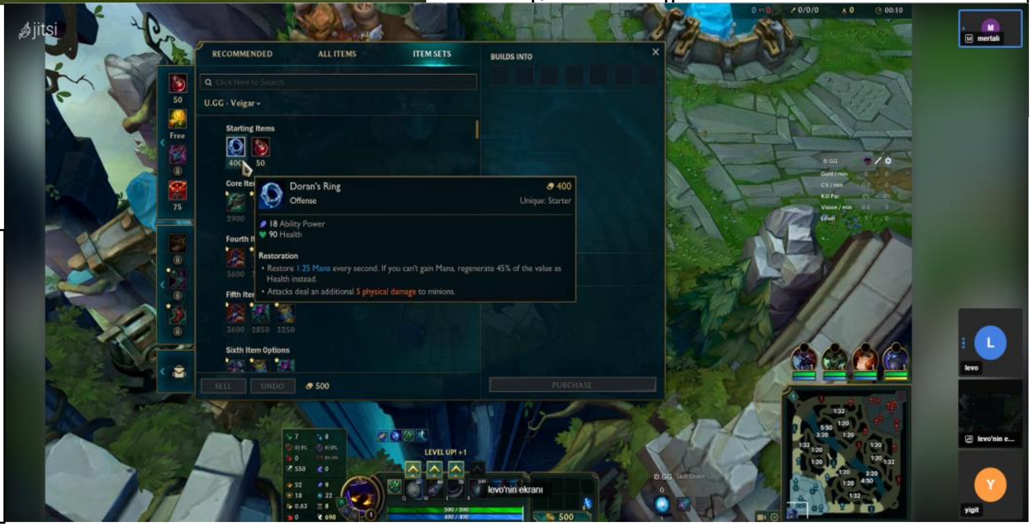
fb taraftarı tarafından

4- 1 (33%)

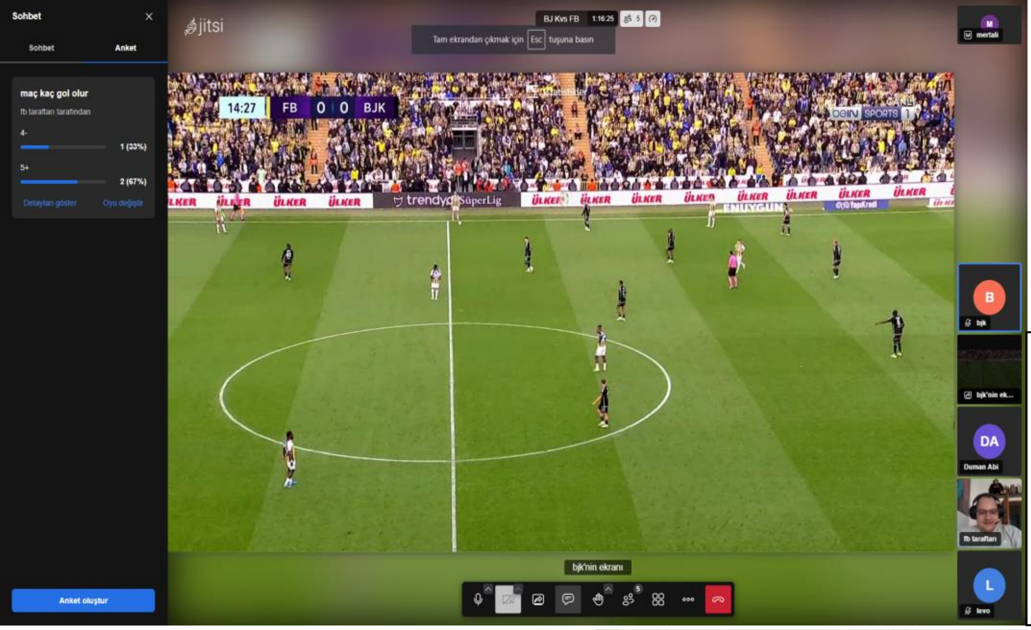
5+ 2 (67%)

Testing Screenshots

Once I run my server publicly, I invited my friends to my Jitsi Meet channel to test screenshot quality, sound quality, additional features, CPU GPU usage, and network



Another screen share test with some popular multiplayer games.



maç kaç gol olur

fb taraftarı tarafından

4-

1 (33%)

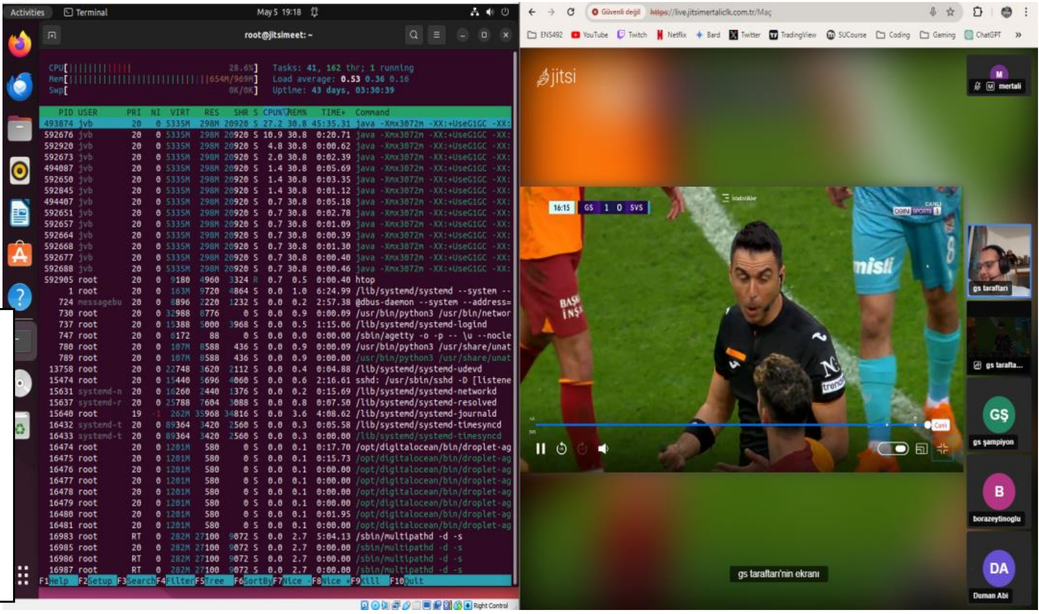
5+

2 (67%)

Detayları göster

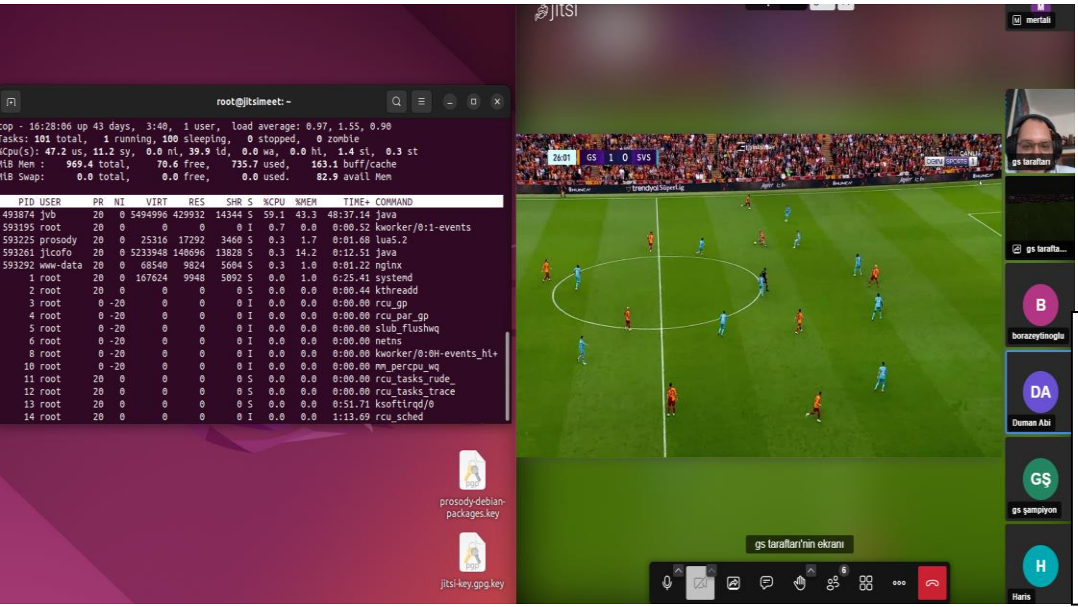
Oyu değiştir

There is a survey feature where all the attendees can give multiple answers.



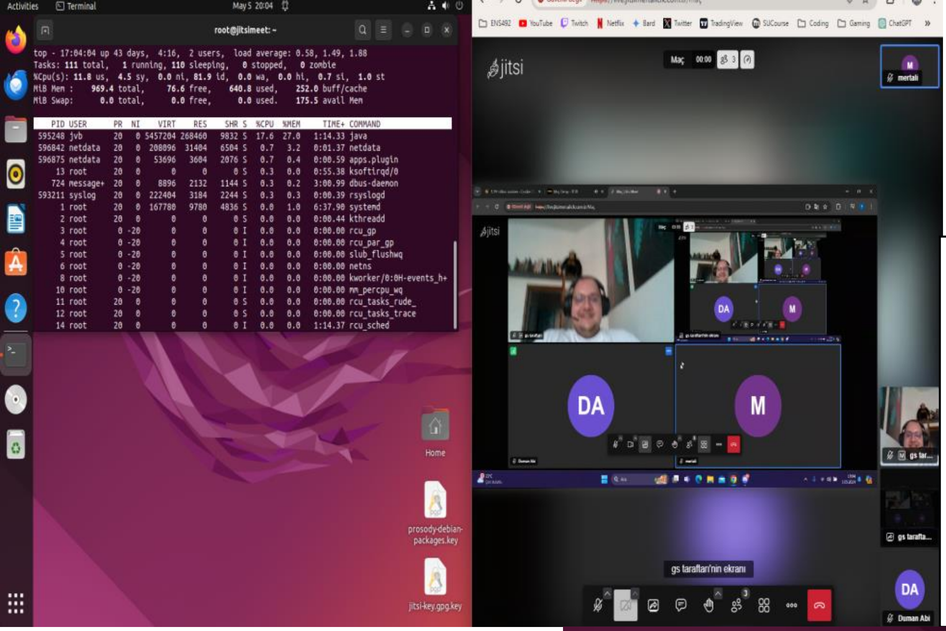
```
  CPU[||||||||||||||                 28.6%]   Tasks: 41, 162 thr; 1 running
  Mem[|||||||||||||||||||||||||||||654M/969M]   Load average: 0.53 0.36 0.16
  Swp[                                0K/0K]   Uptime: 43 days, 03:30:39

    PID USER      PRI  NI  VIRT   RES   SHR S CPU%▽MEM%   TIME+  Command
 493874 jvb        20   0 5335M  298M 20920 S 27.2 30.8 45:35.31 java -Xmx3072m -XX:+UseG1GC -XX:
 592676 jvb        20   0 5335M  298M 20920 S 10.9 30.8  0:20.71 java -Xmx3072m -XX:+UseG1GC -XX:
 592920 jvb        20   0 5335M  298M 20920 S  4.8 30.8  0:00.62 java -Xmx3072m -XX:+UseG1GC -XX:
 592673 jvb        20   0 5335M  298M 20920 S  2.0 30.8  0:02.39 java -Xmx3072m -XX:+UseG1GC -XX:
 494087 jvb        20   0 5335M  298M 20920 S  1.4 30.8  0:05.69 java -Xmx3072m -XX:+UseG1GC -XX:
 592650 jvb        20   0 5335M  298M 20920 S  1.4 30.8  0:03.35 java -Xmx3072m -XX:+UseG1GC -XX:
 592845 jvb        20   0 5335M  298M 20920 S  1.4 30.8  0:01.12 java -Xmx3072m -XX:+UseG1GC -XX:
 494407 jvb        20   0 5335M  298M 20920 S  0.7 30.8  0:05.18 java -Xmx3072m -XX:+UseG1GC -XX:
 592651 jvb        20   0 5335M  298M 20920 S  0.7 30.8  0:02.78 java -Xmx3072m -XX:+UseG1GC -XX:
 592657 jvb        20   0 5335M  298M 20920 S  0.7 30.8  0:01.09 java -Xmx3072m -XX:+UseG1GC -XX:
 592664 jvb        20   0 5335M  298M 20920 S  0.7 30.8  0:00.39 java -Xmx3072m -XX:+UseG1GC -XX:
 592668 jvb        20   0 5335M  298M 20920 S  0.7 30.8  0:01.30 java -Xmx3072m -XX:+UseG1GC -XX:
 592677 jvb        20   0 5335M  298M 20920 S  0.7 30.8  0:00.40 java -Xmx3072m -XX:+UseG1GC -XX:
 592688 jvb        20   0 5335M  298M 20920 S  0.7 30.8  0:00.46 java -Xmx3072m -XX:+UseG1GC -XX:
 592905 root       20   0  9180  4960  3324 R  0.7  0.5  0:00.40 htop
      1 root       20   0  163M  9720  4864 S  0.0  1.0  6:24.99 /lib/systemd/systemd --system --
    724 messagebu  20   0  8896  2220  1232 S  0.0  0.2  2:57.38 @dbus-daemon --system --address=
    730 root       20   0 32988  8776     0 S  0.0  0.9  0:00.09 /usr/bin/python3 /usr/bin/networ
    737 root       20   0 15388  5000  3968 S  0.0  0.5  1:15.06 /lib/systemd/systemd-logind
    747 root       20   0  6172    88     0 S  0.0  0.0  0:00.00 /sbin/agetty -o -p -- \u --nocle
    780 root       20   0  107M  8588   436 S  0.0  0.9  0:00.09 /usr/bin/python3 /usr/share/unat
    789 root       20   0  107M  8588   436 S  0.0  0.9  0:00.00 /usr/bin/python3 /usr/share/unat
  13758 root       20   0 22748  3620  2112 S  0.0  0.4  0:04.88 /lib/systemd/systemd-udevd
  15474 root       20   0 15440  5696  4060 S  0.0  0.6  2:16.61 sshd: /usr/sbin/sshd -D [listene
  15631 systemd-n  20   0 16260  2440  1376 S  0.0  0.2  0:15.69 /lib/systemd/systemd-networkd
  15637 systemd-r  20   0 25788  7604  3088 S  0.0  0.8  0:07.50 /lib/systemd/systemd-resolved
  15640 root       19  -1  262M 35968 34816 S  0.0  3.6  4:08.62 /lib/systemd/systemd-journald
  16432 systemd-t  20   0 89364  3420  2560 S  0.0  0.3  0:05.58 /lib/systemd/systemd-timesyncd
  16433 systemd-t  20   0 89364  3420  2560 S  0.0  0.3  0:00.00 /lib/systemd/systemd-timesyncd
  16474 root       20   0 1201M   580     0 S  0.0  0.1  0:17.70 /opt/digitalocean/bin/droplet-ag
  16475 root       20   0 1201M   580     0 S  0.0  0.1  0:15.73 /opt/digitalocean/bin/droplet-ag
  16476 root       20   0 1201M   580     0 S  0.0  0.1  0:00.00 /opt/digitalocean/bin/droplet-ag
  16477 root       20   0 1201M   580     0 S  0.0  0.1  0:00.00 /opt/digitalocean/bin/droplet-ag
  16478 root       20   0 1201M   580     0 S  0.0  0.1  0:00.00 /opt/digitalocean/bin/droplet-ag
  16479 root       20   0 1201M   580     0 S  0.0  0.1  0:00.00 /opt/digitalocean/bin/droplet-ag
  16480 root       20   0 1201M   580     0 S  0.0  0.1  0:01.95 /opt/digitalocean/bin/droplet-ag
  16481 root       20   0 1201M   580     0 S  0.0  0.1  0:00.00 /opt/digitalocean/bin/droplet-ag
  16983 root       RT   0  282M 27100  9072 S  0.0  2.7  5:04.13 /sbin/multipathd -d -s
  16985 root       20   0  282M 27100  9072 S  0.0  2.7  0:00.00 /sbin/multipathd -d -s
  16986 root       RT   0  282M 27100  9072 S  0.0  2.7  0:00.00 /sbin/multipathd -d -s
  16987 root       RT   0  282M 27100  9072 S  0.0  2.7  0:00.00 /sbin/multipathd -d -s
F1Help  F2Setup F3SearchF4FilterF5Tree  F6SortByF7Nice -F8Nice +F9Kill  F10Quit
```

CPU and memory usage test by using htop with 5 participants. Also, each task has a detailed explanation about its process with its own process id.



```
top - 16:28:06 up 43 days,  3:40,  1 user,  load average: 0.97, 1.55, 0.90
Tasks: 101 total,   1 running, 100 sleeping,   0 stopped,   0 zombie
%Cpu(s): 47.2 us, 11.2 sy,  0.0 ni, 39.9 id,  0.0 wa,  0.0 hi,  1.4 si,  0.3 st
MiB Mem :    969.4 total,     70.6 free,    735.7 used,    163.1 buff/cache
MiB Swap:      0.0 total,      0.0 free,      0.0 used.     82.9 avail Mem

    PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
 493874 jvb       20   0 5494996 429932  14344 S  59.1  43.3  48:37.14 java
 593195 root      20   0       0      0      0 I   0.7   0.0   0:00.52 kworker/0:1-events
 593225 prosody   20   0   25316  17292   3460 S   0.3   1.7   0:01.68 lua5.2
 593261 jicofo    20   0 5233948 140696  13828 S   0.3  14.2   0:12.51 java
 593292 www-data  20   0   68540   9824   5604 S   0.3   1.0   0:01.22 nginx
      1 root      20   0  167624   9948   5092 S   0.0   1.0   6:25.41 systemd
      2 root      20   0       0      0      0 S   0.0   0.0   0:00.44 kthreadd
      3 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 rcu_gp
      4 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 rcu_par_gp
      5 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 slub_flushwq
      6 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 netns
      8 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 kworker/0:0H-events_hi+
     10 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 mm_percpu_wq
     11 root      20   0       0      0      0 S   0.0   0.0   0:00.00 rcu_tasks_rude_
     12 root      20   0       0      0      0 S   0.0   0.0   0:00.00 rcu_tasks_trace
     13 root      20   0       0      0      0 S   0.0   0.0   0:51.71 ksoftirqd/0
     14 root      20   0       0      0      0 I   0.0   0.0   1:13.69 rcu_sched
```

Another test by using top command to see the similar information about the processes and CPU usage, buff/cache actions. With 6 participants.

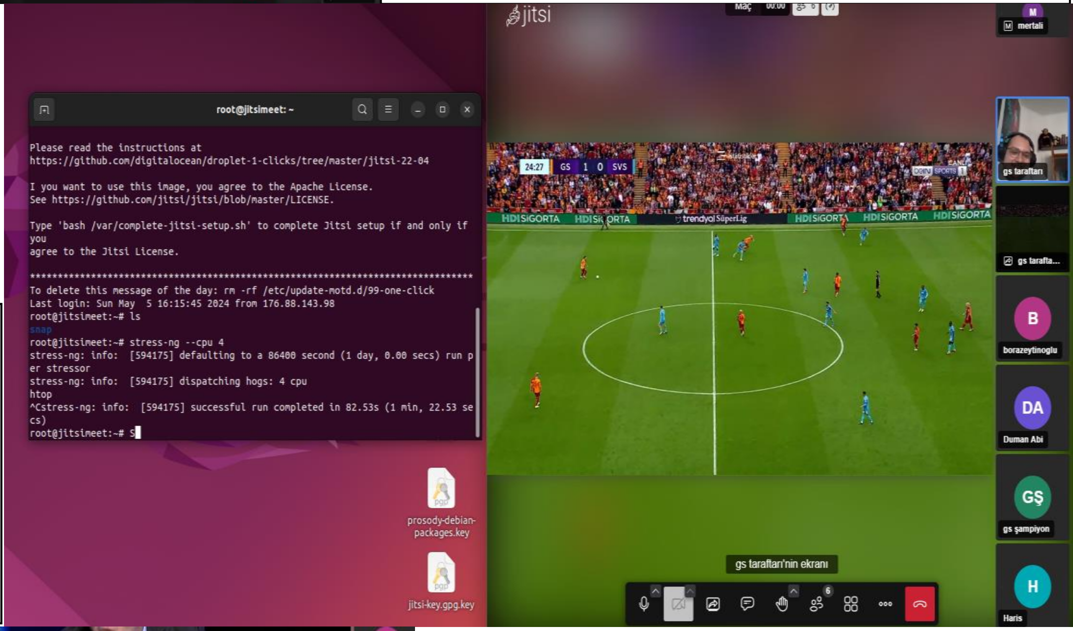


```
top - 17:04:04 up 43 days,  4:16,  2 users,  load average: 0.58, 1.49, 1.88
Tasks: 111 total,   1 running, 110 sleeping,   0 stopped,   0 zombie
%Cpu(s): 11.8 us,  4.5 sy,  0.0 ni, 81.9 id,  0.0 wa,  0.0 hi,  0.7 si,  1.0 st
MiB Mem :    969.4 total,     76.6 free,    640.8 used,    252.0 buff/cache
MiB Swap:      0.0 total,      0.0 free,      0.0 used.    175.5 avail Mem

    PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
 595248 jvb       20   0 5457204 268460   9832 S  17.6  27.0   1:14.33 java
 596842 netdata   20   0  208896  31404   6504 S   0.7   3.2   0:01.37 netdata
 596875 netdata   20   0   53696   3604   2076 S   0.7   0.4   0:00.59 apps.plugin
     13 root      20   0       0      0      0 S   0.3   0.0   0:55.38 ksoftirqd/0
    724 message+  20   0    8896   2132   1144 S   0.3   0.2   3:00.99 dbus-daemon
 593211 syslog    20   0  222404   3184   2244 S   0.3   0.3   0:00.39 rsyslogd
      1 root      20   0  167780   9780   4836 S   0.0   1.0   6:37.90 systemd
      2 root      20   0       0      0      0 S   0.0   0.0   0:00.44 kthreadd
      3 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 rcu_gp
      4 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 rcu_par_gp
      5 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 slub_flushwq
      6 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 netns
      8 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 kworker/0:0H-events_h+
     10 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 mm_percpu_wq
     11 root      20   0       0      0      0 S   0.0   0.0   0:00.00 rcu_tasks_rude_
     12 root      20   0       0      0      0 S   0.0   0.0   0:00.00 rcu_tasks_trace
     14 root      20   0       0      0      0 I   0.0   0.0   1:14.37 rcu_sched
```

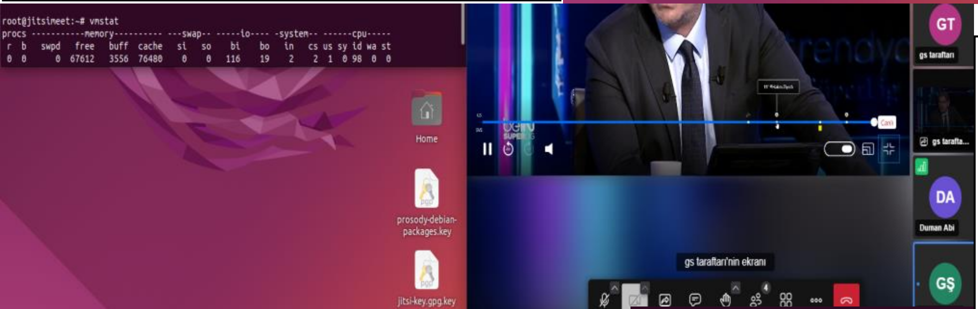
Once again using top command to see the performance test without screenshare and with 3 participants.

After the last 3 screenshots, we can easily say that CPU and memory usage increases nearly %5 percent by CPU and %8 by memory. To better control the stress level let's make a stress test. (There are also other screenshots under Appendix with different amount of participants CPU and memory usage test.)

Used stress-ng –cpu 4 command to divide processes into 4 parts of job for our cpu and see the successful run time under heavily load to test stability and performance. Which is 82.53s (1 min 22.53 sec) There is no overheating or throttling issue happened in my server or my system.



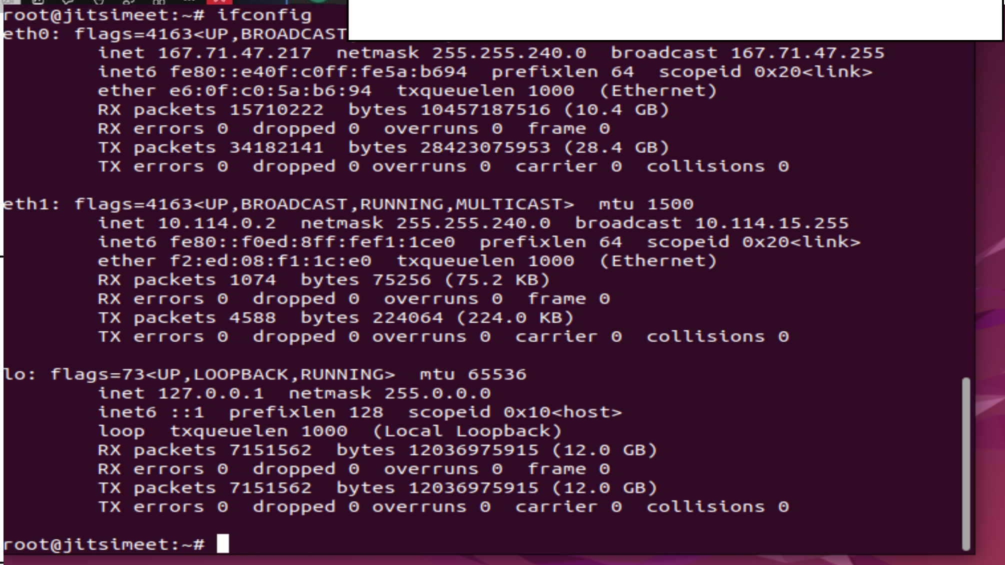
```
Please read the instructions at
https://github.com/digitalocean/droplet-1-clicks/tree/master/jitsi-22-04

I you want to use this image, you agree to the Apache License.
See https://github.com/jitsi/jitsi/blob/master/LICENSE.

Type 'bash /var/complete-jitsi-setup.sh' to complete Jitsi setup if and only if
you
agree to the Jitsi License.

********************************************************************************
To delete this message of the day: rm -rf /etc/update-motd.d/99-one-click
Last login: Sun May  5 16:15:45 2024 from 176.88.143.98
root@jitsimeet:~# ls
snap
root@jitsimeet:~# stress-ng --cpu 4
stress-ng: info:  [594175] defaulting to a 86400 second (1 day, 0.00 secs) run p
er stressor
stress-ng: info:  [594175] dispatching hogs: 4 cpu
htop
^Cstress-ng: info:  [594175] successful run completed in 82.53s (1 min, 22.53 se
cs)
root@jitsimeet:~# S
```



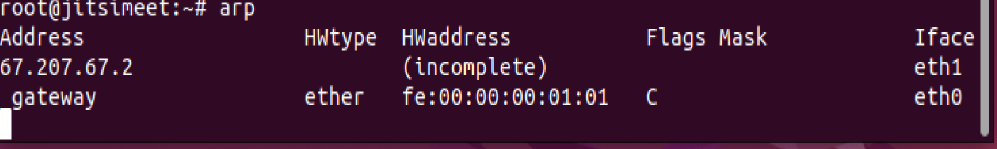
```
root@jitsimeet:~# vmstat
procs -----------memory---------- ---swap-- -----io---- -system-- ------cpu-----
 r  b   swpd   free   buff  cache   si   so    bi    bo   in   cs us sy id wa st
 0  0      0  67612   3556  76480    0    0   116    19    2    2  1  0 98  0  0
```

Used vmstat to display virtual memory statics about processes, traps, memory information, and CPU activity.

Used ifconfig to see and configure network interfaces. Where we can change ethernet interfaces by using eth0 down or up to activate or deactivate. Moreover, we can change MAC addresses and IP addresses too, but I didn't need to change them.



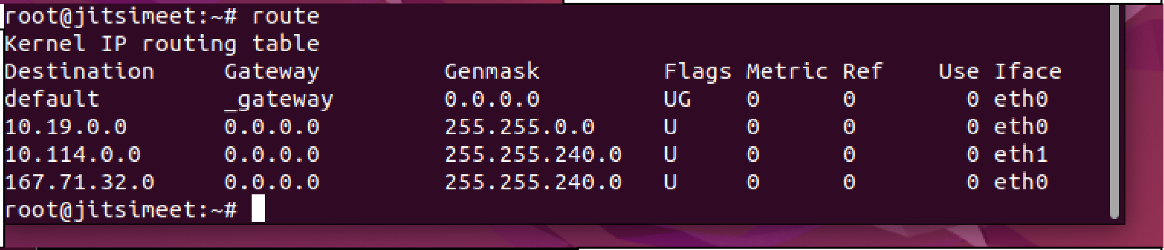
```
root@jitsimeet:~# ifconfig
eth0: flags=4163<UP,BROADCAST
        inet 167.71.47.217  netmask 255.255.240.0  broadcast 167.71.47.255
        inet6 fe80::e40f:c0ff:fe5a:b694  prefixlen 64  scopeid 0x20<link>
        ether e6:0f:c0:5a:b6:94  txqueuelen 1000  (Ethernet)
        RX packets 15710222  bytes 10457187516 (10.4 GB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 34182141  bytes 28423075953 (28.4 GB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

eth1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.114.0.2  netmask 255.255.240.0  broadcast 10.114.15.255
        inet6 fe80::f0ed:8ff:fef1:1ce0  prefixlen 64  scopeid 0x20<link>
        ether f2:ed:08:f1:1c:e0  txqueuelen 1000  (Ethernet)
        RX packets 1074  bytes 75256 (75.2 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 4588  bytes 224064 (224.0 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 7151562  bytes 12036975915 (12.0 GB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 7151562  bytes 12036975915 (12.0 GB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

root@jitsimeet:~# 
```



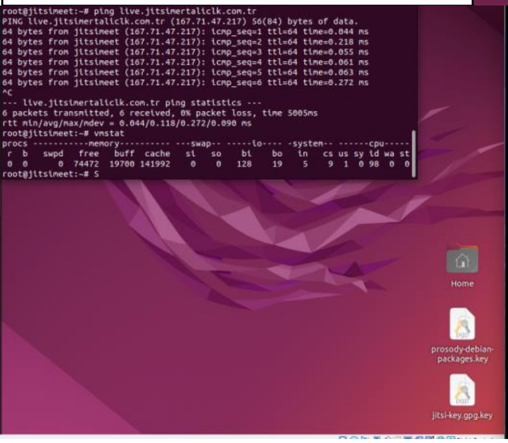
```
root@jitsimeet:~# arp
Address                  HWtype  HWaddress           Flags Mask            Iface
67.207.67.2                      (incomplete)                              eth1
_gateway                 ether   fe:00:00:00:01:01   C                     eth0
```

Used arp command to see the ARP caches which contains a mapping of IP addresses to MAC addresses.

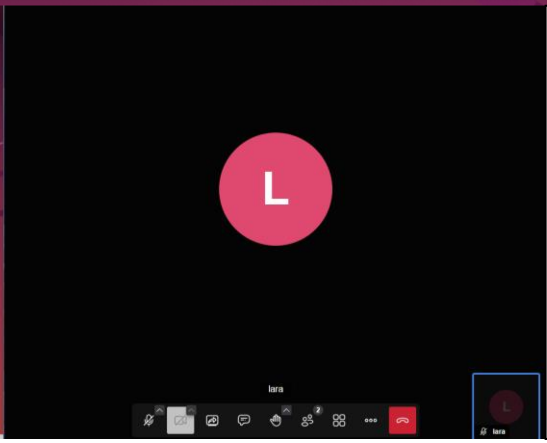
Used route to see route tables which shows the gatewats, netmasks, the destination network, and the interface for each route.



```
root@jitsimeet:~# route
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
default         _gateway        0.0.0.0         UG    0      0        0 eth0
10.19.0.0       0.0.0.0         255.255.0.0     U     0      0        0 eth0
10.114.0.0      0.0.0.0         255.255.240.0   U     0      0        0 eth1
167.71.32.0     0.0.0.0         255.255.240.0   U     0      0        0 eth0
root@jitsimeet:~# 
```



```
root@jitsimeet:~# ping live.jitsimertaliclk.com.tr
PING live.jitsimertaliclk.com.tr (167.71.47.217) 56(84) bytes of data.
64 bytes from jitsimeet (167.71.47.217): icmp_seq=1 ttl=64 time=0.044 ms
64 bytes from jitsimeet (167.71.47.217): icmp_seq=2 ttl=64 time=0.218 ms
64 bytes from jitsimeet (167.71.47.217): icmp_seq=3 ttl=64 time=0.055 ms
64 bytes from jitsimeet (167.71.47.217): icmp_seq=4 ttl=64 time=0.061 ms
64 bytes from jitsimeet (167.71.47.217): icmp_seq=5 ttl=64 time=0.063 ms
64 bytes from jitsimeet (167.71.47.217): icmp_seq=6 ttl=64 time=0.272 ms
^C
--- live.jitsimertaliclk.com.tr ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5005ms
rtt min/avg/max/mdev = 0.044/0.118/0.272/0.090 ms
root@jitsimeet:~# vmstat
procs -----------memory---------- ---swap-- -----io---- -system-- ------cpu-----
 r  b   swpd   free   buff  cache   si   so    bi    bo   in   cs us sy id wa st
 0  0      0  74472  19700 141992    0    0   128    19    5    9  1  0 98  0  0
root@jitsimeet:~# S
```



Used ping command to see the network connectivity of my server.

Documentation

Throughout the project, documentation with screenshots was maintained. This included keeping a record of the installation process, customization changes, and testing results. Documentation is crucial for future development and for other users who might use or contribute to the project.

This methodology, combining the selection of a suitable open-source base, overcoming installation challenges, custom development, and rigorous testing, ensured the development of a reliable and secure video conferencing tool. The process was characterized by adaptability and problem-solving, essential in navigating the complexities of working with open-source software.

4. RESULTS & DISCUSSION

Achievement of Objectives

The project has predominantly succeeded in its initial objectives of developing a secure, scalable, and user-friendly video conferencing tool utilizing open-source software, specifically Jitsi Meet. The major achievements include:

1) **Development of the Software:** A robust video conferencing platform has been created. The platform is distinguished safely and securely.

2) **Scalability and Memory Management:** The software's ability to efficiently handle multiple sessions and large groups has been confirmed, demonstrating commendable scalability and memory management capabilities.
3) **Cost and Comparative Analysis:** The developed software stands out in terms of cost-effectiveness when compared to commercial alternatives, maintaining similar functionalities at a reduced or negligible cost.

Project Completion Status

The project is completed. The core functionalities of the Jitsi Meet software are in place, but crucial elements require further development and testing. The performance under real-time conditions, especially with multiple users employing video and audio simultaneously, is tested with several screenshots. This testing is critical for evaluating the software's scalability and stability under load. Stress testing with varying participant numbers will further illuminate the tool's capacity and responsiveness.

Additionally, Beta testing to gather user feedback can be made in further, which will inform necessary adjustments to enhance usability for a diverse user base. This iterative process of refinement is vital for ensuring that the software is not only technically sound but also user-friendly and accessible.

Contribution to State-of-the-Art

This project contributes meaningfully to the field of video conferencing solutions. It showcases the potential of open-source software in creating a secure, efficient video conferencing tool. The project's focus on user privacy and data security is particularly noteworthy, offering a compelling alternative to existing commercial products. This work highlights the viability and advantages of open-source solutions in the digital communication landscape, especially in the context of remote communication and collaboration.

In conclusion, the project has largely met its initial objectives, with ongoing efforts to finalize and refine certain aspects. The development thus far has resulted in a functional, secure video conferencing tool. Extensive real-world testing and user experience enhancement are made, with a view to establishing the software as a competitive and user-centric solution in the open-source community. The journey of this project demonstrates the challenges and rewards of developing open-source software, contributing valuable insights and tools to the ever-evolving domain of digital communication technologies.

5. IMPACTS

Scientific and Technological Impacts

The development of an open-source video conferencing tool using Jitsi Meet has notable scientific and technological impacts:

1) **Advancement in Open-Source Technology:** By successfully leveraging and customizing an open-source platform, this project contributes to the field of software development, particularly in the realm of digital communication technologies. It showcases the capability of open-source software to meet and even surpass the standards set by commercial counterparts, especially in terms of security and privacy.
2) **Enhanced Understanding of Software Scalability:** Through the project's focus on scalability and memory management, it contributes to a deeper understanding of how open-source tools can be optimized for high-performance video conferencing, catering to a wide range of group sizes and dynamics.
3) **Innovation in Video Conferencing:** The project emphasizes user privacy and data security, areas often overlooked in commercial video conferencing tools. Implementing advanced security protocols represents a significant advancement in the technology's application for secure digital communications.

Socio-Economic Impacts

The project has several socio-economic implications:

1) **Accessibility and Cost-Effectiveness:** By providing a cost-effective alternative to commercial video conferencing solutions, this project has the potential to make digital communication tools more accessible to individuals and organizations with limited budgets, such as small businesses, educational institutions, and non-profit organizations.
2) **Promotion of Privacy and Security Awareness:** The project highlights the importance of privacy and security in digital communications, potentially influencing user behavior and expectations towards these aspects in software products.

Innovative and Commercial Aspects

This project opens avenues for innovation and commercialization in several ways:

1. **Potential for Customization Services:** The open-source nature of the software provides opportunities for offering customization services tailored to specific organizational needs, fostering an entrepreneurial environment.
2. **Foundation for Future Development:** The project lays the groundwork for future innovations in secure and efficient video conferencing technology. It can serve as a base for other developers to build upon, potentially leading to new features or derivative products.

Freedom-to-Use (FTU) Considerations

Regarding Freedom-to-Use issues:

1. **Open Source Licensing:** The project, based on Jitsi Meet, adheres to open-source licensing agreements. This ensures that the software remains free to use, modify, and distribute, aligning with the principles of the open-source community.
2. **SSL Certificate Licensing:** The website might need a SSL Certificate in order to reach webpage directly but it is not necessary for this project. Also, there are free ways to get SSL Certificate.
3. **No FTU Conflicts:** There have been no identified conflicts with Freedom-to-Use. The project's reliance on open-source components ensures that it remains accessible and free from proprietary constraints, encouraging wider adoption and contribution from the community.

In summary, the impact of this project extends beyond the technological realm, influencing socio-economic dynamics by making secure and effective communication tools more accessible. It fosters an environment of innovation and potential commercialization while adhering to the principles of open-source development, ensuring freedom-to-use and encouraging community collaboration.

6. ETHICAL ISSUES

The development and implementation of an open-source video conferencing tool, as conducted in this project, necessitate a thorough consideration of ethical issues.

Potential Ethical Concerns

1. **Use of Patent-Protected Designs or Concepts:** The project's reliance on Jitsi Meet, an open-source tool, mitigates the risk of infringing on patent-protected designs. Continuous monitoring is in place to ensure no such infringement occurs.
2. **Privacy and Data Security:** User privacy and data security are paramount, especially in video conferencing software. This project implements robust security measures, along with transparency about data usage.
3. **Health Concerns:** The project acknowledges the potential health effects related to screen time but recognizes this as an industry-wide issue beyond the project's direct scope.
4. **Non-Exploitative Design:** The project avoids unethical designs, such as those for secret surveillance. Open-source transparency and community oversight prevent exploitative uses.

Addressing Ethical Issues

1. **Adherence to Open-Source Ethics:** The project adheres to open-source principles, ensuring transparency and free access.
2. **Compliance with Legal Standards:** Development and implementation comply with legal standards for privacy and digital communication.

There is no significant ethical issue have been identified in the project. The commitment to open-source principles and a strong focus on user privacy and security help mitigate many potential concerns. However, the open-source nature of the project introduces specific cybersecurity vulnerabilities that require continuous attention and proactive management. As the project evolves and new features are added, maintaining high ethical standards and robust security measures will be critical.

7. PROJECT MANAGEMENT

Initial Project Plan

The initial plan for this project was structured to develop a secure, open-source video conferencing tool using Jitsi Meet. The stages were laid out as follows:

1. **Research and Selection of Tools:** The first stage involved researching various open-source platforms and selecting Jitsi Meet based on its features and compatibility with the project's objectives.
2. **Setting Up Development Environment:** This included configuring a Linux-based system for development and testing, with an emphasis on creating a stable and secure environment.
3. **Development and Customization:** Focused on modifying Jitsi Meet to suit the project requirements, this stage involved enhancing user interface, security features, and scalability.
4. **Initial Testing:** Testing in controlled environments to assess functionality, security, and performance.
5. **Documentation:** Keeping detailed records of the development process, configurations, and testing results.

Changes in the Project Plan

As the project progressed, several adjustments were made to the original plan:

1. **Increased Emphasis on User Experience:** Feedback from initial testing phases indicated the need for a more intuitive user interface. This led to an additional development phase focused on enhancing the user interface and overall user experience.
2. **Expanded Testing:** To better evaluate the software's real-world applicability, the testing phase was expanded to include stress testing with multiple users and various usage scenarios.

Final Project Plan

The final project plan incorporated these changes and was structured as follows:

1. **Refined Development:** Focused on reinforcing security features and improving the user interface based on feedback and new insights.
2. **Comprehensive Testing:** This crucial phase now includes stress testing the server-client architecture of JitsiMeet with multiple users having their cameras on and one user screen-shares videos for them. The aim is to simulate real-world conditions to assess the software's performance, scalability, and stability.
3. **Feedback Incorporation:** Gathering and analyzing user feedback from testing to make final adjustments.
4. **Final Documentation:** Updating the documentation to reflect all changes and findings from the development and testing phases.

Lessons Learned in Project Management

Throughout the project, several valuable lessons were learned in managing such a complex endeavor:

1. **Adaptability:** The ability to adapt the project plan in response to new findings, challenges, and user feedback was crucial. Flexibility in the face of changing requirements or obstacles proved to be a key asset.
2. **Importance of Security:** The project reinforced the importance of prioritizing security in software development, especially in communication tools.
3. **User-Centric Approach:** Understanding user needs and expectations is vital. The feedback loop created during the testing phase was instrumental in refining the software to be more user-friendly.
4. **Time Management and Resource Allocation:** Efficient management of time and resources was necessary to keep the project on track, particularly when accommodating the expanded scope of testing and development.
5. **Documentation and Record Keeping:** Maintaining detailed documentation throughout the process was invaluable, both for troubleshooting and for future reference.

In summary, the project management experience underscored the importance of adaptability, security-focus, user-centered design, efficient resource management, and thorough documentation in successfully bringing a software development project to fruition.

8. CONCLUSION AND FUTURE WORK

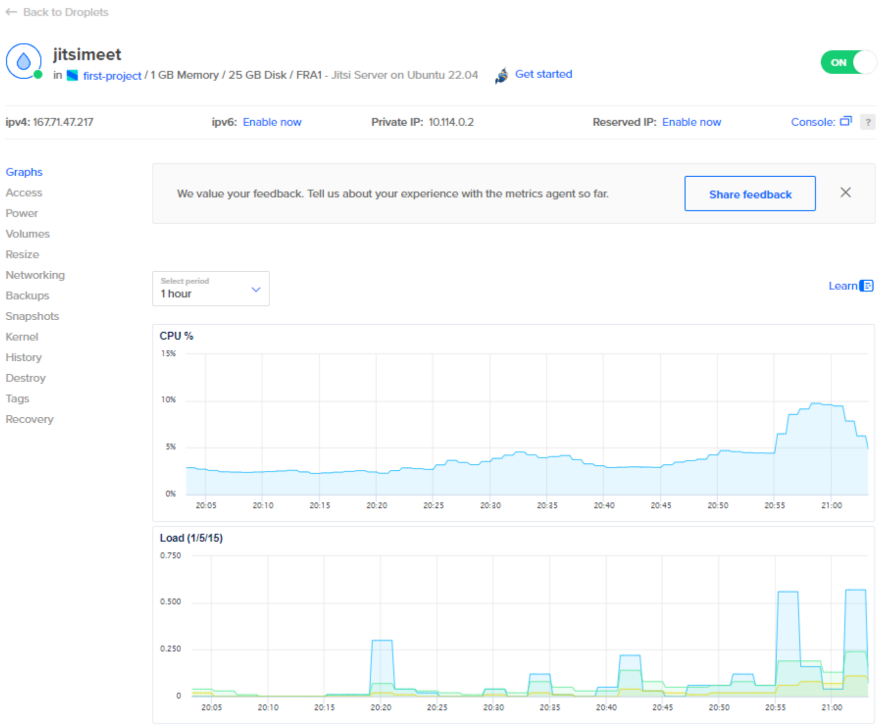
The main achievement of the project is the creation of a secure and user-friendly video conferencing tool, which is based on Jitsi Meet. This tool showcases the successful customization of open-source software, with a specific emphasis on improving user privacy and security. It also incorporates strong security measures and a more user-friendly interface.

Although the tool shows impressive scalability and efficiency, it is important to note that the majority of tests were done in controlled situations. Additionally, the development and testing process was hindered by resource restrictions. Following studies should prioritize thorough real-world experimentation to thoroughly assess performance and scalability, broaden functionality to incorporate virtual backdrops, create a mobile platform iteration, and persistently improve security measures.

Students who want to develop this project might pursue several options, including completing comprehensive user surveys, researching interaction with other services, and examining emerging technologies such as blockchain for improved security or augmented reality for immersive meetings.

To summarize, this project has laid a solid groundwork for an open-source video conferencing system, with much potential for future advancements and improvements in both technological elements and user experience.

9. APPENDIX



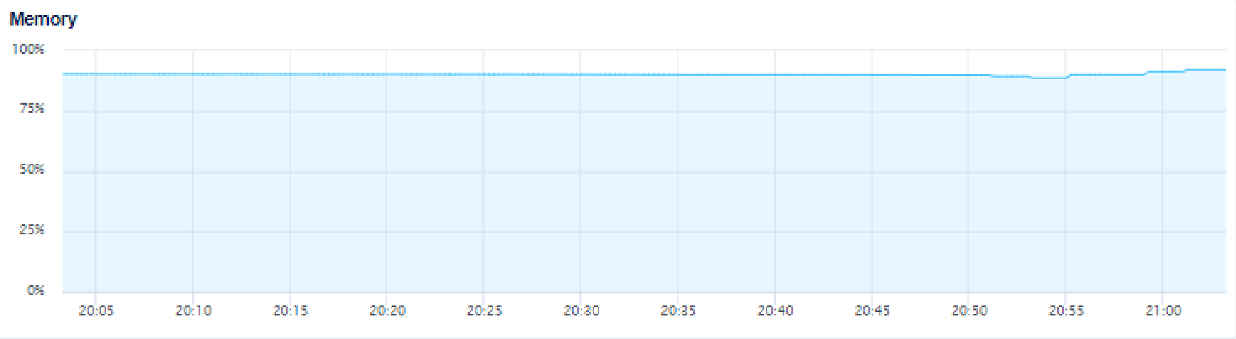
jitsimeet

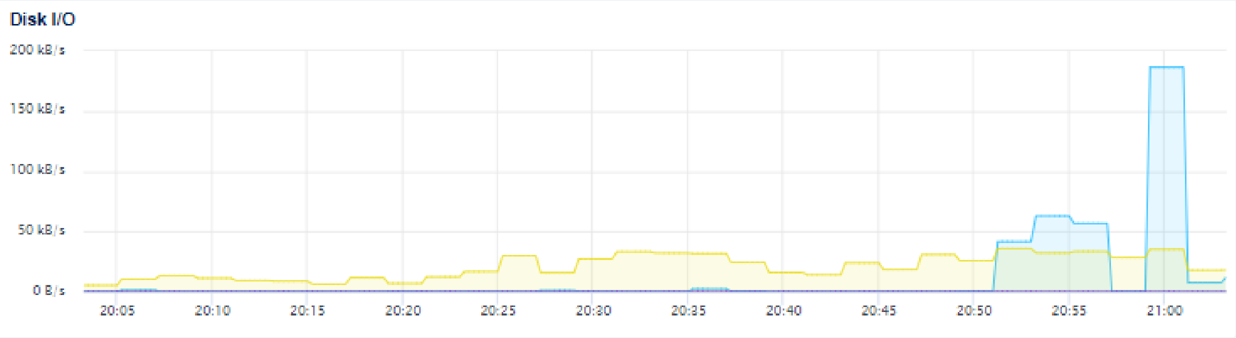
in first-project / 1 GB Memory / 25 GB Disk / FRA1 - Jitsi Server on Ubuntu 22.04

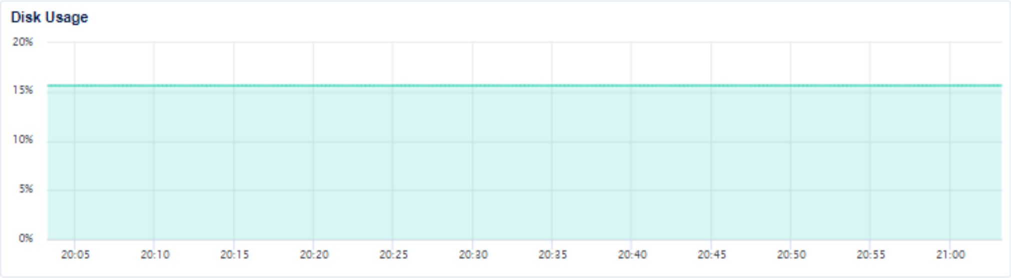
ipv4: 167.71.47.217 ipv6: Enable now Private IP: 10.114.0.2 Reserved IP: Enable now Console:

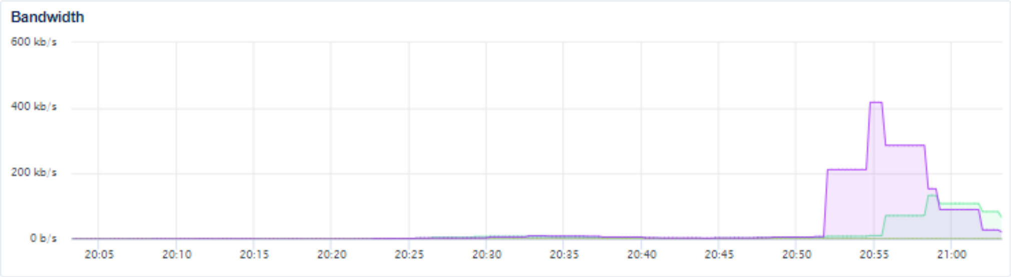
We value your feedback. Tell us about your experience with the metrics agent so far.

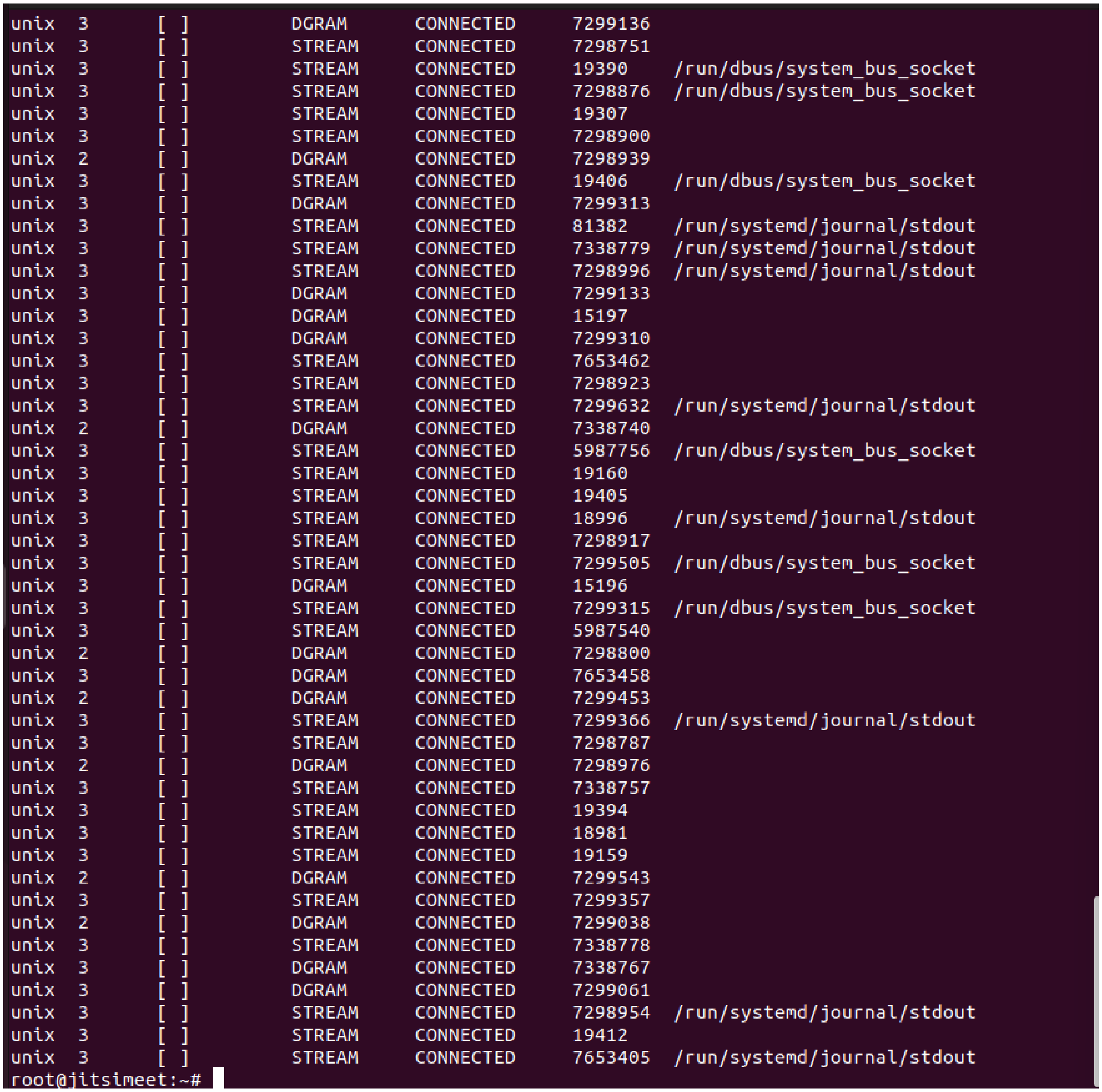
Select period: 1 hour


Memory
100%
75%
50%
25%
0%
20:05
20:10
20:15
20:20
20:25
20:30
20:35
20:40
20:45
20:50
20:55
21:00

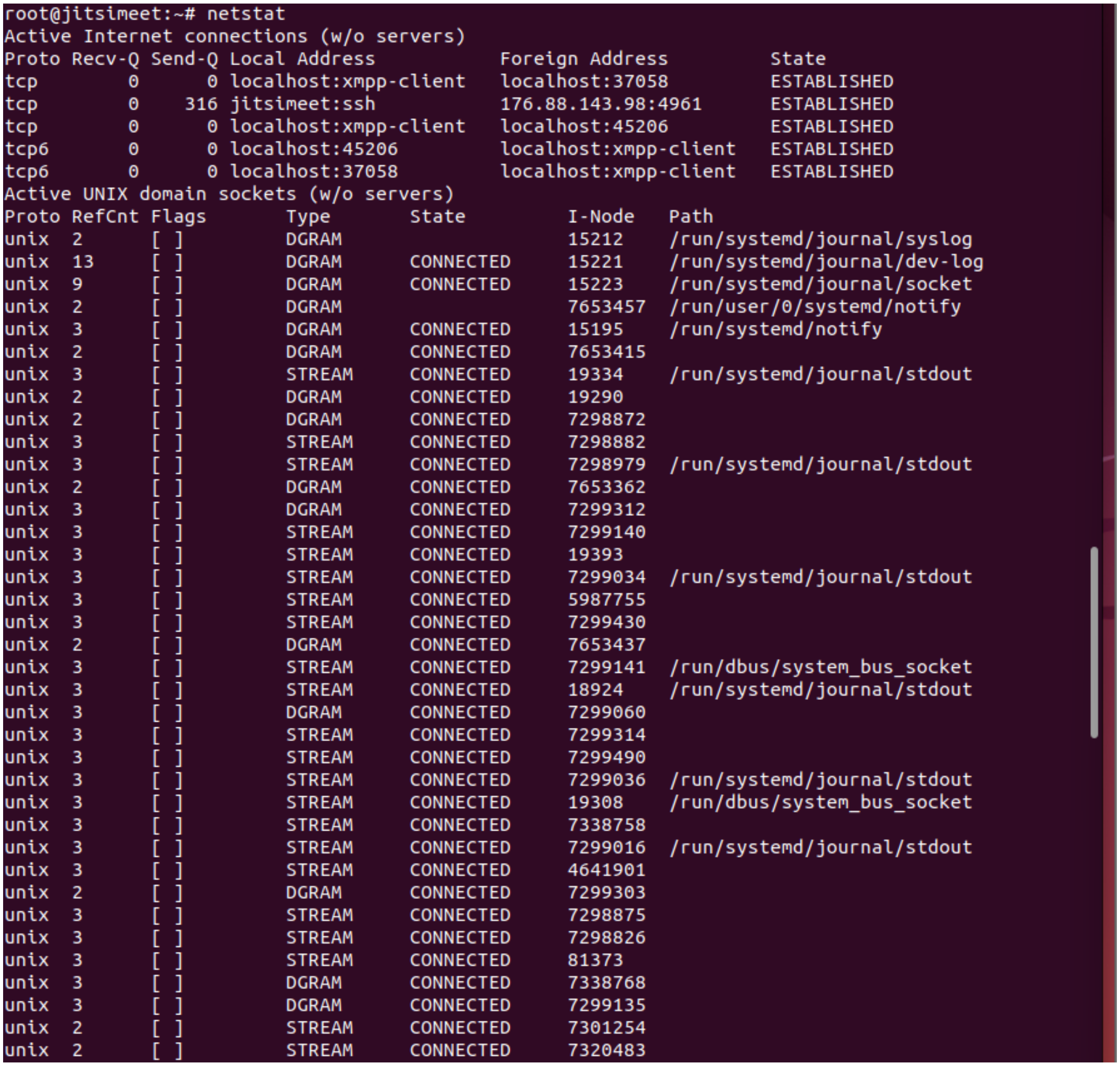

Disk I/O
200 kB/s
150 kB/s
100 kB/s
50 kB/s
0 B/s
20:05
20:10
20:15
20:20
20:25
20:30
20:35
20:40
20:45
20:50
20:55
21:00


Disk Usage
20%
15%
10%
5%
0%
20:05
20:10
20:15
20:20
20:25
20:30
20:35
20:40
20:45
20:50
20:55
21:00

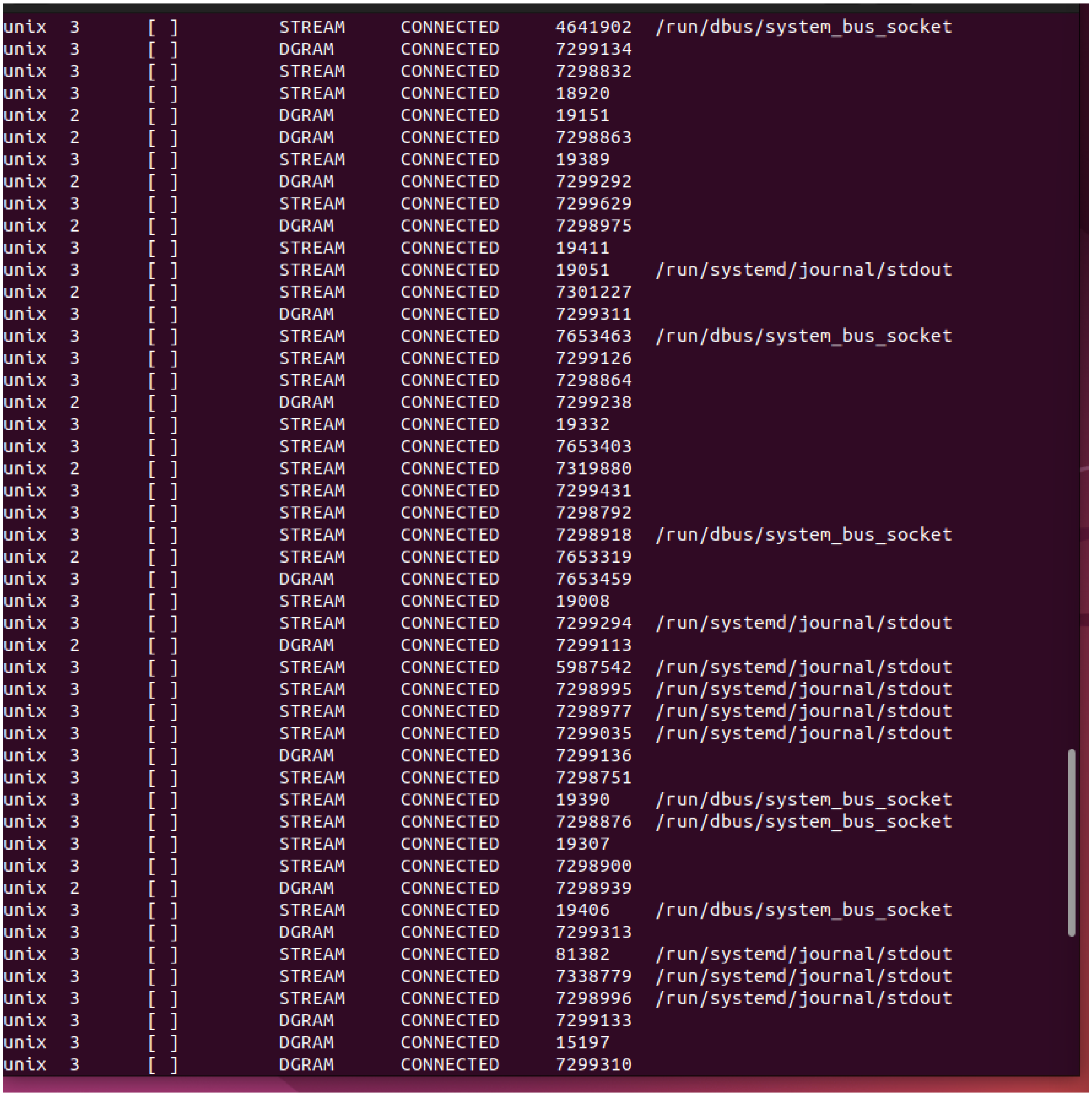

Bandwidth
600 kb/s
400 kb/s
200 kb/s
0 b/s
20:05
20:10
20:15
20:20
20:25
20:30
20:35
20:40
20:45
20:50
20:55
21:00



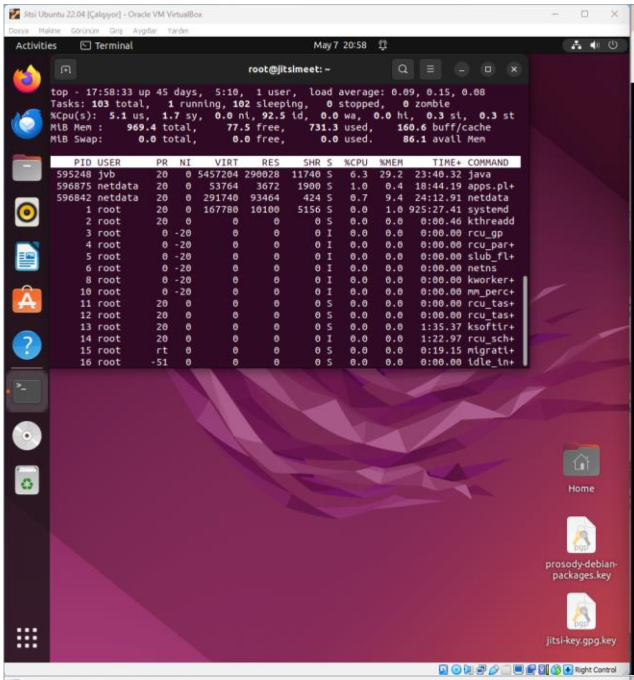
```
unix  3      [ ]         DGRAM      CONNECTED     7299136
unix  3      [ ]         STREAM     CONNECTED     7298751
unix  3      [ ]         STREAM     CONNECTED     19390    /run/dbus/system_bus_socket
unix  3      [ ]         STREAM     CONNECTED     7298876  /run/dbus/system_bus_socket
unix  3      [ ]         STREAM     CONNECTED     19307
unix  3      [ ]         STREAM     CONNECTED     7298900
unix  2      [ ]         DGRAM      CONNECTED     7298939
unix  3      [ ]         STREAM     CONNECTED     19406    /run/dbus/system_bus_socket
unix  3      [ ]         DGRAM      CONNECTED     7299313
unix  3      [ ]         STREAM     CONNECTED     81382    /run/systemd/journal/stdout
unix  3      [ ]         STREAM     CONNECTED     7338779  /run/systemd/journal/stdout
unix  3      [ ]         STREAM     CONNECTED     7298996  /run/systemd/journal/stdout
unix  3      [ ]         DGRAM      CONNECTED     7299133
unix  3      [ ]         DGRAM      CONNECTED     15197
unix  3      [ ]         DGRAM      CONNECTED     7299310
unix  3      [ ]         STREAM     CONNECTED     7653462
unix  3      [ ]         STREAM     CONNECTED     7298923
unix  3      [ ]         STREAM     CONNECTED     7299632  /run/systemd/journal/stdout
unix  2      [ ]         DGRAM      CONNECTED     7338740
unix  3      [ ]         STREAM     CONNECTED     5987756  /run/dbus/system_bus_socket
unix  3      [ ]         STREAM     CONNECTED     19160
unix  3      [ ]         STREAM     CONNECTED     19405
unix  3      [ ]         STREAM     CONNECTED     18996    /run/systemd/journal/stdout
unix  3      [ ]         STREAM     CONNECTED     7298917
unix  3      [ ]         STREAM     CONNECTED     7299505  /run/dbus/system_bus_socket
unix  3      [ ]         DGRAM      CONNECTED     15196
unix  3      [ ]         STREAM     CONNECTED     7299315  /run/dbus/system_bus_socket
unix  3      [ ]         STREAM     CONNECTED     5987540
unix  2      [ ]         DGRAM      CONNECTED     7298800
unix  3      [ ]         DGRAM      CONNECTED     7653458
unix  2      [ ]         DGRAM      CONNECTED     7299453
unix  3      [ ]         STREAM     CONNECTED     7299366  /run/systemd/journal/stdout
unix  3      [ ]         STREAM     CONNECTED     7298787
unix  2      [ ]         DGRAM      CONNECTED     7298976
unix  3      [ ]         STREAM     CONNECTED     7338757
unix  3      [ ]         STREAM     CONNECTED     19394
unix  3      [ ]         STREAM     CONNECTED     18981
unix  3      [ ]         STREAM     CONNECTED     19159
unix  2      [ ]         DGRAM      CONNECTED     7299543
unix  3      [ ]         STREAM     CONNECTED     7299357
unix  2      [ ]         DGRAM      CONNECTED     7299038
unix  3      [ ]         STREAM     CONNECTED     7338778
unix  3      [ ]         DGRAM      CONNECTED     7338767
unix  3      [ ]         DGRAM      CONNECTED     7299061
unix  3      [ ]         STREAM     CONNECTED     7298954  /run/systemd/journal/stdout
unix  3      [ ]         STREAM     CONNECTED     19412
unix  3      [ ]         STREAM     CONNECTED     7653405  /run/systemd/journal/stdout
root@jitsimeet:~# 
```



```
root@jitsimeet:~# netstat
Active Internet connections (w/o servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 localhost:xmpp-client   localhost:37058         ESTABLISHED
tcp        0    316 jitsimeet:ssh           176.88.143.98:4961      ESTABLISHED
tcp        0      0 localhost:xmpp-client   localhost:45206         ESTABLISHED
tcp6       0      0 localhost:45206         localhost:xmpp-client   ESTABLISHED
tcp6       0      0 localhost:37058         localhost:xmpp-client   ESTABLISHED
Active UNIX domain sockets (w/o servers)
Proto RefCnt Flags       Type       State         I-Node   Path
unix  2      [ ]         DGRAM                    15212    /run/systemd/journal/syslog
unix  13     [ ]         DGRAM      CONNECTED     15221    /run/systemd/journal/dev-log
unix  9      [ ]         DGRAM      CONNECTED     15223    /run/systemd/journal/socket
unix  2      [ ]         DGRAM                    7653457  /run/user/0/systemd/notify
unix  3      [ ]         DGRAM      CONNECTED     15195    /run/systemd/notify
unix  2      [ ]         DGRAM      CONNECTED     7653415
unix  3      [ ]         STREAM     CONNECTED     19334    /run/systemd/journal/stdout
unix  2      [ ]         DGRAM      CONNECTED     19290
unix  2      [ ]         DGRAM      CONNECTED     7298872
unix  3      [ ]         STREAM     CONNECTED     7298882
unix  3      [ ]         STREAM     CONNECTED     7298979  /run/systemd/journal/stdout
unix  2      [ ]         DGRAM      CONNECTED     7653362
unix  3      [ ]         DGRAM      CONNECTED     7299312
unix  3      [ ]         STREAM     CONNECTED     7299140
unix  3      [ ]         STREAM     CONNECTED     19393
unix  3      [ ]         STREAM     CONNECTED     7299034  /run/systemd/journal/stdout
unix  3      [ ]         STREAM     CONNECTED     5987755
unix  3      [ ]         STREAM     CONNECTED     7299430
unix  2      [ ]         DGRAM      CONNECTED     7653437
unix  3      [ ]         STREAM     CONNECTED     7299141  /run/dbus/system_bus_socket
unix  3      [ ]         STREAM     CONNECTED     18924    /run/systemd/journal/stdout
unix  3      [ ]         DGRAM      CONNECTED     7299060
unix  3      [ ]         STREAM     CONNECTED     7299314
unix  3      [ ]         STREAM     CONNECTED     7299490
unix  3      [ ]         STREAM     CONNECTED     7299036  /run/systemd/journal/stdout
unix  3      [ ]         STREAM     CONNECTED     19308    /run/dbus/system_bus_socket
unix  3      [ ]         STREAM     CONNECTED     7338758
unix  3      [ ]         STREAM     CONNECTED     7299016  /run/systemd/journal/stdout
unix  3      [ ]         STREAM     CONNECTED     4641901
unix  2      [ ]         DGRAM      CONNECTED     7299303
unix  3      [ ]         STREAM     CONNECTED     7298875
unix  3      [ ]         STREAM     CONNECTED     7298826
unix  3      [ ]         STREAM     CONNECTED     81373
unix  3      [ ]         DGRAM      CONNECTED     7338768
unix  3      [ ]         DGRAM      CONNECTED     7299135
unix  2      [ ]         STREAM     CONNECTED     7301254
unix  2      [ ]         STREAM     CONNECTED     7320483
```

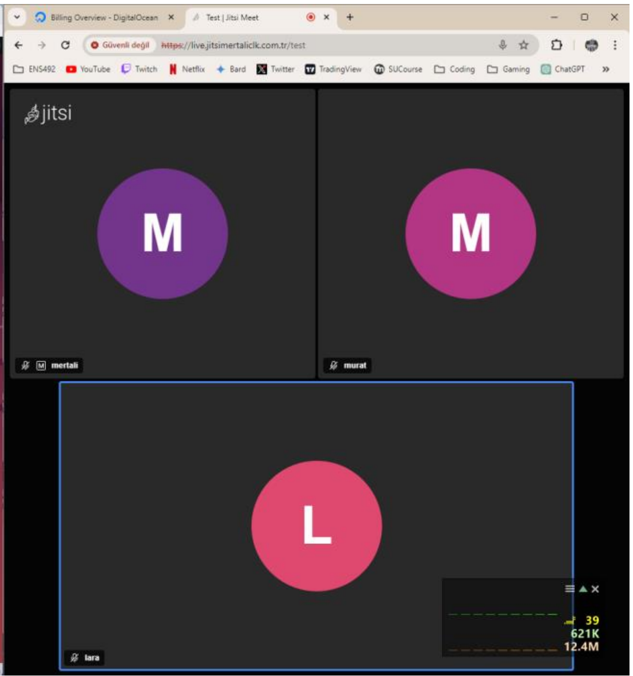


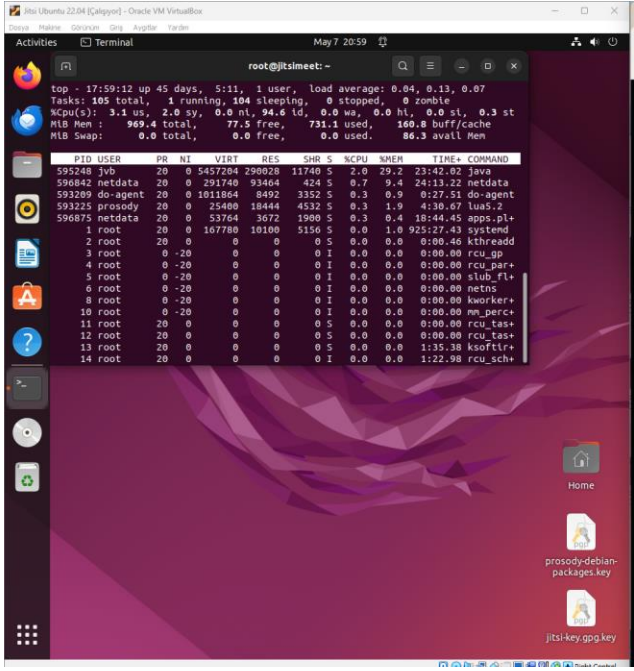
```
unix  3      [ ]         STREAM     CONNECTED     4641902  /run/dbus/system_bus_socket
unix  3      [ ]         DGRAM      CONNECTED     7299134
unix  3      [ ]         STREAM     CONNECTED     7298832
unix  3      [ ]         STREAM     CONNECTED     18920
unix  2      [ ]         DGRAM      CONNECTED     19151
unix  2      [ ]         DGRAM      CONNECTED     7298863
unix  3      [ ]         STREAM     CONNECTED     19389
unix  2      [ ]         DGRAM      CONNECTED     7299292
unix  3      [ ]         STREAM     CONNECTED     7299629
unix  2      [ ]         DGRAM      CONNECTED     7298975
unix  3      [ ]         STREAM     CONNECTED     19411
unix  3      [ ]         STREAM     CONNECTED     19051    /run/systemd/journal/stdout
unix  2      [ ]         STREAM     CONNECTED     7301227
unix  3      [ ]         DGRAM      CONNECTED     7299311
unix  3      [ ]         STREAM     CONNECTED     7653463  /run/dbus/system_bus_socket
unix  3      [ ]         STREAM     CONNECTED     7299126
unix  3      [ ]         STREAM     CONNECTED     7298864
unix  2      [ ]         DGRAM      CONNECTED     7299238
unix  3      [ ]         STREAM     CONNECTED     19332
unix  3      [ ]         STREAM     CONNECTED     7653403
unix  2      [ ]         STREAM     CONNECTED     7319880
unix  3      [ ]         STREAM     CONNECTED     7299431
unix  3      [ ]         STREAM     CONNECTED     7298792
unix  3      [ ]         STREAM     CONNECTED     7298918  /run/dbus/system_bus_socket
unix  2      [ ]         STREAM     CONNECTED     7653319
unix  3      [ ]         DGRAM      CONNECTED     7653459
unix  3      [ ]         STREAM     CONNECTED     19008
unix  3      [ ]         STREAM     CONNECTED     7299294  /run/systemd/journal/stdout
unix  2      [ ]         DGRAM      CONNECTED     7299113
unix  3      [ ]         STREAM     CONNECTED     5987542  /run/systemd/journal/stdout
unix  3      [ ]         STREAM     CONNECTED     7298995  /run/systemd/journal/stdout
unix  3      [ ]         STREAM     CONNECTED     7298977  /run/systemd/journal/stdout
unix  3      [ ]         STREAM     CONNECTED     7299035  /run/systemd/journal/stdout
unix  3      [ ]         DGRAM      CONNECTED     7299136
unix  3      [ ]         STREAM     CONNECTED     7298751
unix  3      [ ]         STREAM     CONNECTED     19390    /run/dbus/system_bus_socket
unix  3      [ ]         STREAM     CONNECTED     7298876  /run/dbus/system_bus_socket
unix  3      [ ]         STREAM     CONNECTED     19307
unix  3      [ ]         STREAM     CONNECTED     7298900
unix  2      [ ]         DGRAM      CONNECTED     7298939
unix  3      [ ]         STREAM     CONNECTED     19406    /run/dbus/system_bus_socket
unix  3      [ ]         DGRAM      CONNECTED     7299313
unix  3      [ ]         STREAM     CONNECTED     81382    /run/systemd/journal/stdout
unix  3      [ ]         STREAM     CONNECTED     7338779  /run/systemd/journal/stdout
unix  3      [ ]         STREAM     CONNECTED     7298996  /run/systemd/journal/stdout
unix  3      [ ]         DGRAM      CONNECTED     7299133
unix  3      [ ]         DGRAM      CONNECTED     15197
unix  3      [ ]         DGRAM      CONNECTED     7299310
```



Jitsi Ubuntu 22.04 [Çalışıyor] - Oracle VM VirtualBox
Activities
Terminal
May 7 20:58
root@jitsimeet: ~
Home
prosody-debian-packages.key
jitsi-key.gpg.key
Right Control

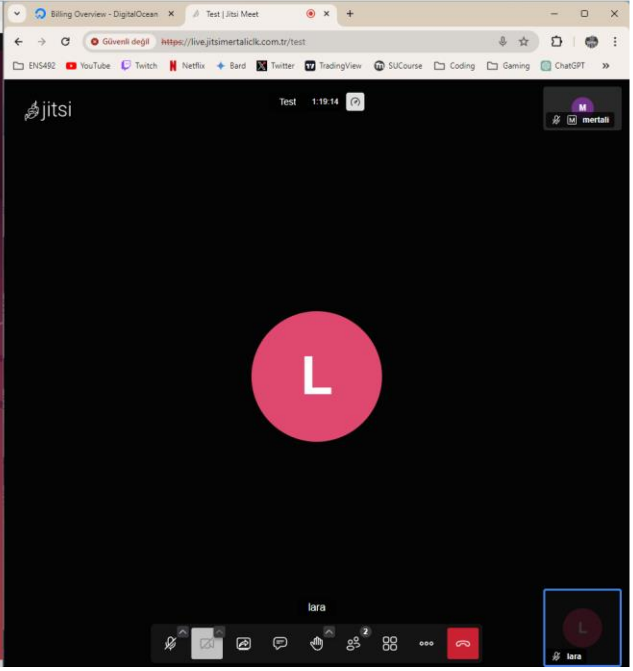
```
top - 17:58:33 up 45 days,  5:10,  1 user,  load average: 0.09, 0.15, 0.08
Tasks: 103 total,   1 running, 102 sleeping,   0 stopped,   0 zombie
%Cpu(s):  5.1 us,  1.7 sy,  0.0 ni, 92.5 id,  0.0 wa,  0.0 hi,  0.3 si,  0.3 st
MiB Mem :    969.4 total,     77.5 free,    731.3 used,    160.6 buff/cache
MiB Swap:      0.0 total,      0.0 free,      0.0 used.     86.1 avail Mem

    PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
 595248 jvb       20   0 5457204 290028  11740 S   6.3  29.2  23:40.32 java
 596875 netdata   20   0   53764   3672   1900 S   1.0   0.4  18:44.19 apps.pl+
 596842 netdata   20   0  291740  93464    424 S   0.7   9.4  24:12.91 netdata
      1 root      20   0  167780  10100   5156 S   0.0   1.0 925:27.41 systemd
      2 root      20   0       0      0      0 S   0.0   0.0   0:00.46 kthreadd
      3 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 rcu_gp
      4 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 rcu_par+
      5 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 slub_fl+
      6 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 netns
      8 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 kworker+
     10 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 mm_perc+
     11 root      20   0       0      0      0 S   0.0   0.0   0:00.00 rcu_tas+
     12 root      20   0       0      0      0 S   0.0   0.0   0:00.00 rcu_tas+
     13 root      20   0       0      0      0 S   0.0   0.0   1:35.37 ksoftir+
     14 root      20   0       0      0      0 I   0.0   0.0   1:22.97 rcu_sch+
     15 root      rt   0       0      0      0 S   0.0   0.0   0:19.15 migrati+
     16 root     -51   0       0      0      0 S   0.0   0.0   0:00.00 idle_in+
```

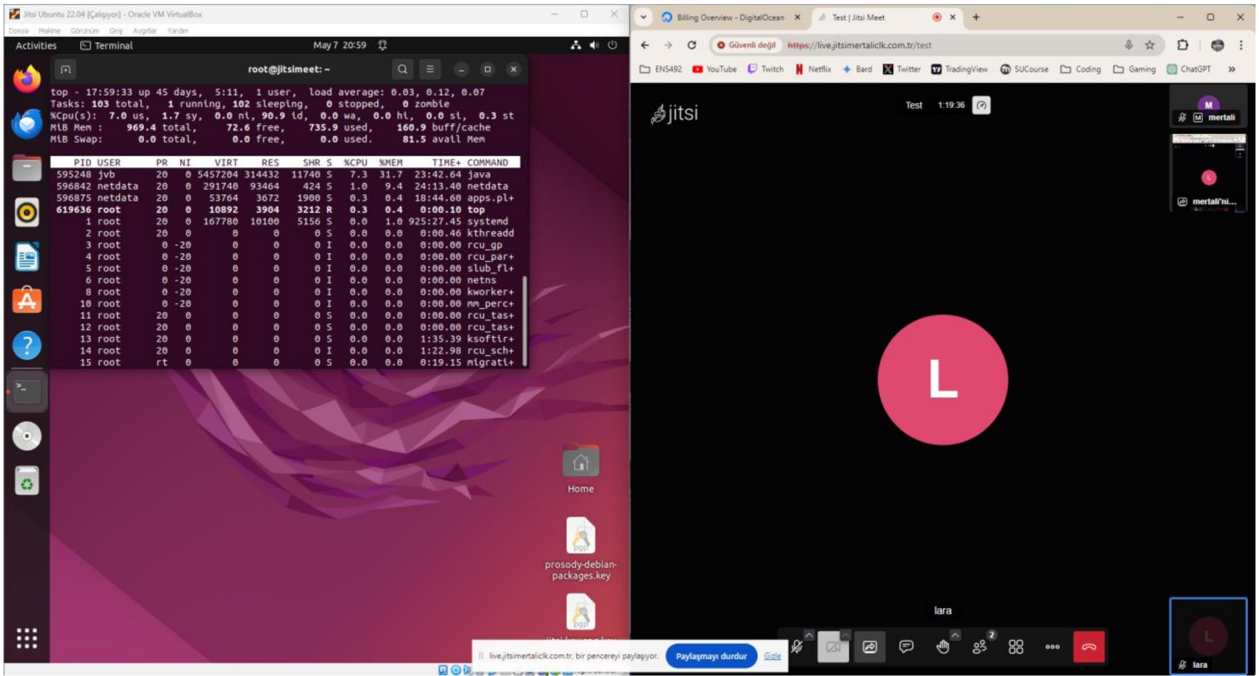




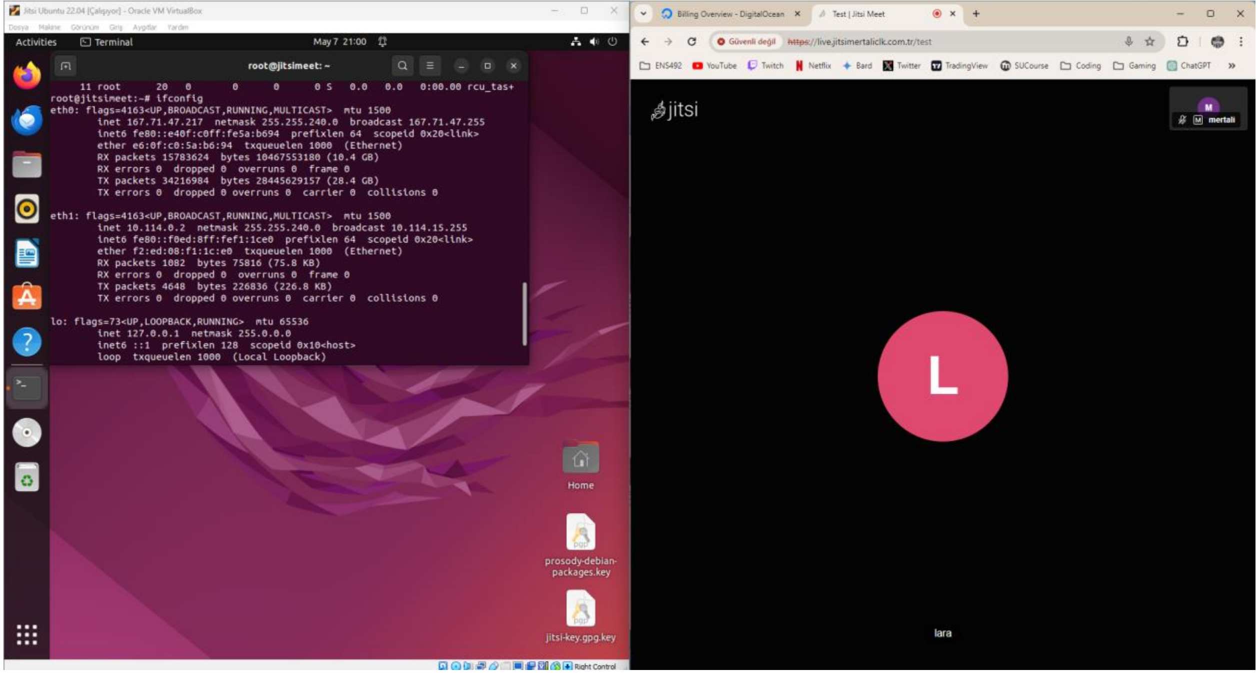
Jitsi Ubuntu 22.04 [Çalışıyor] - Oracle VM VirtualBox
Activities
Terminal
May 7 20:59
root@jitsimeet: ~
Home
prosody-debian-packages.key
jitsi-key.gpg.key

```
top - 17:59:12 up 45 days,  5:11,  1 user,  load average: 0.04, 0.13, 0.07
Tasks: 105 total,   1 running, 104 sleeping,   0 stopped,   0 zombie
%Cpu(s):  3.1 us,  2.0 sy,  0.0 ni, 94.6 id,  0.0 wa,  0.0 hi,  0.0 si,  0.3 st
MiB Mem :    969.4 total,     77.5 free,    731.1 used,    160.8 buff/cache
MiB Swap:      0.0 total,      0.0 free,      0.0 used.     86.3 avail Mem

    PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
 595248 jvb       20   0 5457204 290028  11740 S   2.0  29.2  23:42.02 java
 596842 netdata   20   0  291740  93464    424 S   0.7   9.4  24:13.22 netdata
 593209 do-agent  20   0 1011864   8492   3352 S   0.3   0.9   0:27.51 do-agent
 593225 prosody   20   0   25400  18444   4532 S   0.3   1.9   4:30.67 lua5.2
 596875 netdata   20   0   53764   3672   1900 S   0.3   0.4  18:44.45 apps.pl+
      1 root      20   0  167780  10100   5156 S   0.0   1.0 925:27.43 systemd
      2 root      20   0       0      0      0 S   0.0   0.0   0:00.46 kthreadd
      3 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 rcu_gp
      4 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 rcu_par+
      5 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 slub_fl+
      6 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 netns
      8 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 kworker+
     10 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 mm_perc+
     11 root      20   0       0      0      0 S   0.0   0.0   0:00.00 rcu_tas+
     12 root      20   0       0      0      0 S   0.0   0.0   0:00.00 rcu_tas+
     13 root      20   0       0      0      0 S   0.0   0.0   1:35.38 ksoftir+
     14 root      20   0       0      0      0 I   0.0   0.0   1:22.98 rcu_sch+
```

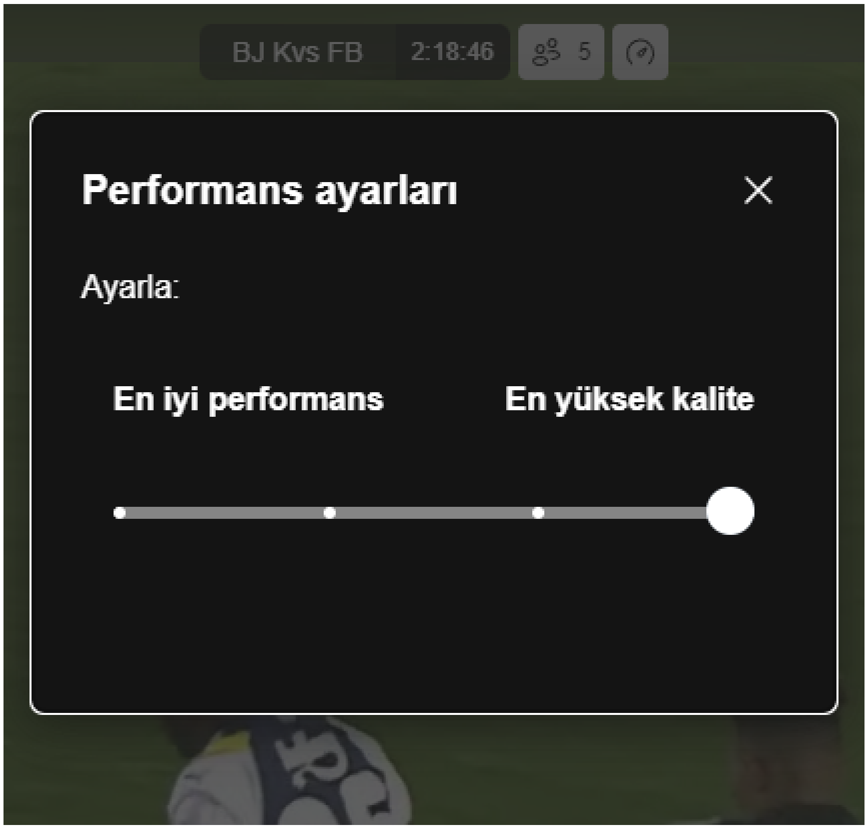



Jitsi Ubuntu 22.04 [Çalışıyor] - Oracle VM VirtualBox
Dosya Makine Görünüm Giriş Aygıtlar Yardım
Activities
Terminal
May 7 20:59
root@jitsimeet: ~
Home
prosody-debian-packages.key
Billing Overview - DigitalOcean
Test | Jitsi Meet
Güvenli değil
https://live.jitsimertaliclk.com.tr/test
ENS492 YouTube Twitch Netflix Bard Twitter TradingView SUCourse Coding Gaming ChatGPT
jitsi
Test 1:19:36
mertali
mertali'ni...
L
lara
live.jitsimertaliclk.com.tr; bir pencereyi paylaşıyor.
Paylaşmayı durdur
Gizle

```
top - 17:59:33 up 45 days,  5:11,  1 user,  load average: 0.03, 0.12, 0.07
Tasks: 103 total,   1 running, 102 sleeping,   0 stopped,   0 zombie
%Cpu(s):  7.0 us,  1.7 sy,  0.0 ni, 90.9 id,  0.0 wa,  0.0 hi,  0.0 si,  0.3 st
MiB Mem :    969.4 total,     72.6 free,    735.9 used,    160.9 buff/cache
MiB Swap:      0.0 total,      0.0 free,      0.0 used.     81.5 avail Mem

    PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
 595248 jvb       20   0 5457204 314432  11740 S   7.3  31.7  23:42.64 java
 596842 netdata   20   0  291740  93464    424 S   1.0   9.4  24:13.40 netdata
 596875 netdata   20   0   53764   3672   1900 S   0.3   0.4  18:44.60 apps.pl+
 619636 root      20   0   10892   3904   3212 R   0.3   0.4   0:00.10 top
      1 root      20   0  167780  10100   5156 S   0.0   1.0 925:27.45 systemd
      2 root      20   0       0      0      0 S   0.0   0.0   0:00.46 kthreadd
      3 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 rcu_gp
      4 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 rcu_par+
      5 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 slub_fl+
      6 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 netns
      8 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 kworker+
     10 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 mm_perc+
     11 root      20   0       0      0      0 S   0.0   0.0   0:00.00 rcu_tas+
     12 root      20   0       0      0      0 S   0.0   0.0   0:00.00 rcu_tas+
     13 root      20   0       0      0      0 S   0.0   0.0   1:35.39 ksoftir+
     14 root      20   0       0      0      0 I   0.0   0.0   1:22.98 rcu_sch+
     15 root      rt   0       0      0      0 S   0.0   0.0   0:19.15 migrati+
```


Jitsi Ubuntu 22.04 [Çalışıyor] - Oracle VM VirtualBox
Dosya Makine Görünüm Giriş Aygıtlar Yardım
Activities
Terminal
May 7 21:00
root@jitsimeet: ~
Home
prosody-debian-packages.key
jitsi-key.gpg.key
Right Control
Billing Overview - DigitalOcean
Test | Jitsi Meet
Güvenli değil
https://live.jitsimertaliclk.com.tr/test
ENS492 YouTube Twitch Netflix Bard Twitter TradingView SUCourse Coding Gaming ChatGPT
jitsi
mertali
L
lara

```
     11 root      20   0       0      0      0 S   0.0   0.0   0:00.00 rcu_tas+
root@jitsimeet:~# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 167.71.47.217  netmask 255.255.240.0  broadcast 167.71.47.255
        inet6 fe80::e40f:c0ff:fe5a:b694  prefixlen 64  scopeid 0x20<link>
        ether e6:0f:c0:5a:b6:94  txqueuelen 1000  (Ethernet)
        RX packets 15783624  bytes 10467553180 (10.4 GB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 34216984  bytes 28445629157 (28.4 GB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

eth1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.114.0.2  netmask 255.255.240.0  broadcast 10.114.15.255
        inet6 fe80::f0ed:8ff:fef1:1ce0  prefixlen 64  scopeid 0x20<link>
        ether f2:ed:08:f1:1c:e0  txqueuelen 1000  (Ethernet)
        RX packets 1082  bytes 75816 (75.8 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 4648  bytes 226836 (226.8 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
```

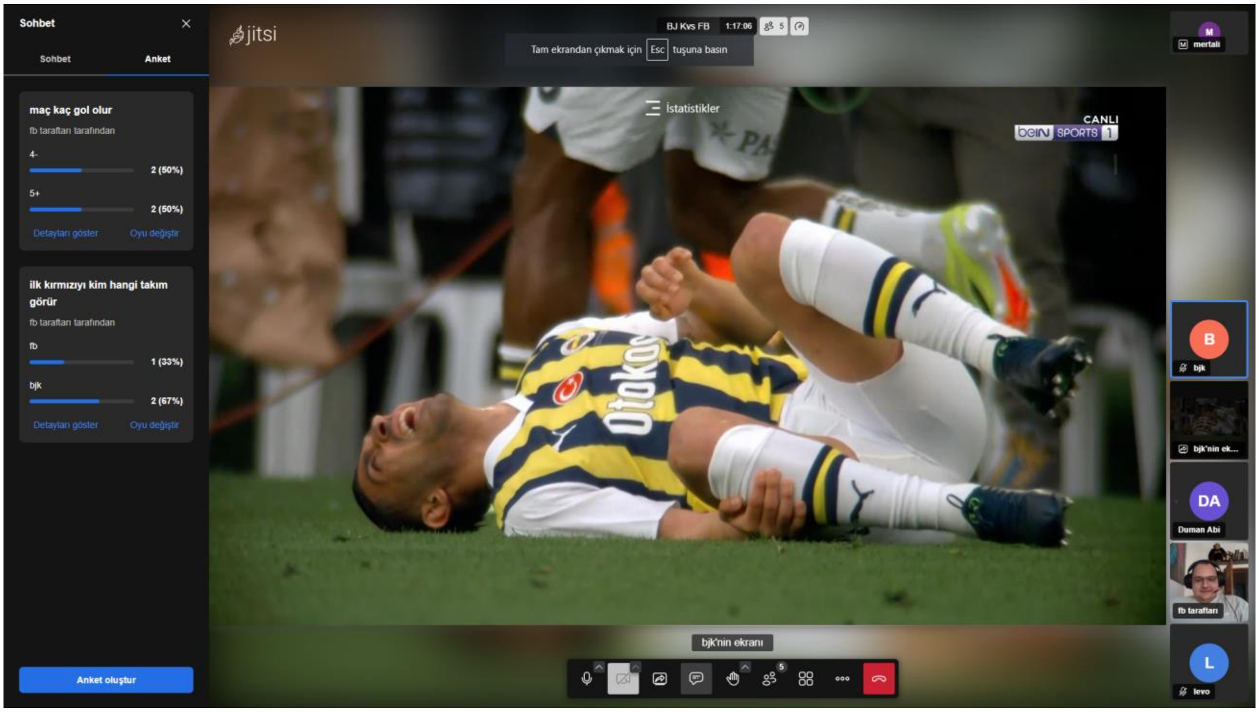


Performans ayarları

Ayarla:

En iyi performans

En yüksek kalite



Sohbet

Sohbet | Anket

maç kaç gol olur

fb taraftarı tarafından

4- 2 (50%)

5+ 2 (50%)

Detayları göster | Oyu değiştir

ilk kırmızıyı kim hangi takım görür

fb taraftarı tarafından

fb 1 (33%)

bjk 2 (67%)

Detayları göster | Oyu değiştir

Anket oluştur

10. REFERENCES

1. Dudding, C. C. (2009). Digital Videoconferencing: Applications Across the Disciplines. Communication Disorders Quarterly, 30(3), 178–182.

2. Renfrew, M., Morton, D., Morton, J., Hinze, J., Beamish, P., Przybylko, G., & Craig, B. (2020). A Web- and Mobile App–Based Mental Health Promotion Intervention Comparing Email, Short Message Service, and Videoconferencing Support for a Healthy Cohort: Randomized Comparative Study. Journal of Medical Internet Research, 22(1), e15592.

3. https://github.com/jitsi/jitsi-meet

4. https://dashboard.ngrok.com/login?state=uIUZ1nKviOnteHRRePFo8iGsJCPBYv6ndgs4wVxNVBhh4OyjMhvrhNSy2Xzg7FyU8DrDtp4HNe25aYBGYJhQN6LkN2SW6hRJb3gkBOwDY49Eqzx3bqjT27BNqVNEU5TBNwqFgQE7E24h3QJDQE3Dj4Zy6GbzgBGBQXanx9GvwGtHtsHrmJ3vLWs%3D

5. https://cloud.digitalocean.com/databases?i=8ad8ef

6. https://app.metunic.com.tr/client/login/

7. https://www.namecheap.com/support/knowledgebase/article.aspx/10375/2208/how-do-i-link-a-domain-to-my-digitalocean-account/?_ga=2.133298830.1277036912.1710767761-1201122972.1710767761