



Collaborative Virtual Environment Quick start guide

Virtual research environment (from Wikipedia):

A virtual research environment (VRE) or virtual laboratory is an online system helping researchers collaborate. Features usually include collaboration support (Web forums and wikis), document hosting, and some discipline-specific tools, such as data analysis, visualisation, or simulation management. In some instances, publication management, and teaching tools such as presentations and slides may be included. VREs have become important in fields where research is primarily carried out in teams which span institutions and even countries: the ability to easily share information and research results is valuable.

Note:

We'll use Ubuntu server 22.x

LINUX first steps:

First of all: login as root on linux server and create all users we'll need (each user will access in a separated environment on jupyterhub and RStudio)

```
>> adduser guest1  
(Not necessary: adduser guest1 sudo)  
>> adduser guest2  
(Not necessary: adduser guest2 sudo)  
>> adduser guest3  
(Not necessary: adduser guest3 sudo)
```



JUPYTER HUB quick install:

```
>> apt update
>> apt install python3-pip
>> apt install nodejs npm
>> python3 -m pip install jupyterhub
>> npm install -g configurable-http-proxy
>> python3 -m pip install jupyterlab notebook
>> jupyterhub -h
>> configurable-http-proxy -h
>> nohup jupyterhub &
```

JULIA Kernel (repeat JULIASHELL for each account)

```
>> wget https://julialang-s3.julialang.org/bin/linux/x64/1.8/julia-1.8.5-linux-
x86_64.tar.gz
>> tar -xvzf julia-1.8.5-linux-x86_64.tar.gz
>> mv julia-1.8.5/ /opt/
>> ln -s /opt/julia-1.8.5/bin/julia /usr/local/bin/julia
>> julia
JULIASHELL> using Pkg
JULIASHELL> Pkg.add("IJulia")
JULIASHELL> using IJulia
JULIASHELL> installkernel("Julia")
```

R kernel

Install

```
>> apt install r-cran-littler
>> sudo apt update -qq
>> sudo apt install --no-install-recommends software-properties-common dirmngr
>> wget -qO- https://cloud.r-project.org/bin/linux/ubuntu/marutter_pubkey.asc |
sudo tee -a /etc/apt/trusted.gpg.d/cran_ubuntu_key.asc
>> add-apt-repository deb https://cloud.r-project.org/bin/linux/ubuntu $
(lsb_release -cs)-cran40/
>> apt install --no-install-recommends r-base
>> add-apt-repository ppa:c2d4u.team/c2d4u4.0+
```

Add kernel to Jupyter

```
>> R
```

```
R SHELL> install.packages('IRkernel')
R SHELL> IRkernel::installspec(user = FALSE)
```

OCTAVE Kernel Install

```
>> add-apt-repository ppa:octave/stable
>> apt-get install -y --no-install-recommends octave gnuplot ghostscript
>> pip3 install octave_kernel
>> apt-get install -y --no-install-recommends octave gnuplot ghostscript qt
>> apt-get install -y --no-install-recommends octave gnuplot ghostscript qt5
>> apt-get install -y --no-install-recommends octave octave-symbolic octave-
miscellaneous python-sympy gnuplot ghostscript
>> apt-get install -y --no-install-recommends octave octave-miscellaneous gnuplot
ghostscript
>> apt install fonts-freefont-otf
```

MATLAB Kernel

```
>> pip install matlab_kernel
```

LUA Kernel

```
>> pip install ilua
```

MicroPython Kernel

```
>> pip install jupyter_micropython_kernel
>> python -m jupyter_micropython_kernel.install
```

Install voila

Voilà turns Jupyter notebooks into standalone web applications.
Starting with JupyterLab 3.0, the extension is automatically installed after installing voila with

```
>> pip install voila
```

Security TIP: Disable Jupyter Terminal!

```
>> pip3 uninstall -y terminado
```

Let's go with the browser for JupyterHub:

<http://myserver.com:8000/>

NOTE: for Jupyter some PIP packages needs the root access by shell to install dependencies
eg.

```
>> apt-get install libgeos-dev
>> pip install Cartopy
```



Install RStudio Server

Add R-base repository

```
>> sudo add-apt-repository "deb https://cloud.r-project.org/bin/linux/ubuntu focal-cran40/"
```

Run update

```
>> sudo apt update
```

Install Gdebi

To easily install Rstudio Desktop or Server, we will use the Gdebi Package manager that will automatically fetch and install the required dependencies needed by the tool.

```
>> sudo apt-get install gdebi-core
```

Install OLD dependencies (broken in Ubuntu 22.x) necessities for RStudio

```
>> wget  
http://security.ubuntu.com/ubuntu/pool/main/o/openssl/libssl1.1_1.1.1f-1ubuntu2_amd64.deb  
>> sudo apt install ./libssl1.1_1.1.1f-1ubuntu2_amd64.deb
```

Download RStudio Server package

```
>> wget https://download2.rstudio.org/server/bionic/amd64/rstudio-server-1.3.1093-amd64.deb  
>> sudo gdebi rstudio-server-1.3.1093-amd64.deb
```

Let's go with the browser for Rstudio:

<http://myserver.com:8787/>



Install PostgreSQL database and GITEA repository

```
>> apt update
>> apt install postgresql postgresql-contrib
>> systemctl start postgresql.service
>> cd /etc/postgresql/14/main/
>> nano postgresql.conf
```

1. For remote database setup, configure PostgreSQL on database instance to listen to your IP address by editing listen_addresses on postgresql.conf to:
listen_addresses='localhost, 203.0.113.3'

2. PostgreSQL uses md5 challenge-response encryption scheme for password authentication by default. Now a scheme is not considered secure anymore. Use SCRAM-SHA-256 scheme instead by editing the postgresql.configuration file on the database server to:

```
password_encryption=scram-sha-256
```

```
>> systemctl restart postgresql.service
```

On the database server, login to the database console as superuser:

```
>> su -c psql - postgres
```

Create database user (role in PostgreSQL terms) with login privilege and password.

```
POSTGRESQL >> CREATE ROLE gitea WITH LOGIN PASSWORD
'myStrongPasswordForGitea';
```

Create database with UTF-8 charset and owned by the database user created earlier.

```
POSTGRESQL >> CREATE DATABASE giteadb WITH OWNER gitea TEMPLATE
template0 ENCODING UTF8 LC_COLLATE 'en_US.UTF-8' LC_CTYPE 'en_US.UTF-8';
```

```
>> cd /etc/postgresql/14/main/
>> nano pg_hba.conf
```

Allow the database user to access the database created above by adding the following authentication rule to pg_hba.conf.

For local database:

```
local giteadb gitea scram-sha-256
```

For remote database:

```
host giteadb gitea 192.0.2.10/32 scram-sha-256
```

Replace database name, user, and IP address of Gitea instance with your own.

```
>> systemctl restart postgresql.service
```

On your Gitea server, test connection to the database.

For local database:

```
>> psql -U gitea -d giteadb
```

For remote database:

```
>> psql "postgres://gitea@203.0.113.3/giteadb"
```

Add repo signing key to apt

```
>> curl -sL -o /etc/apt/trusted.gpg.d/morph027-gitea.asc  
https://packaging.gitlab.io/gitea/gpg.key
```

Add repo to apt

```
>> echo deb https://packaging.gitlab.io/gitea gitea main | sudo tee  
/etc/apt/sources.list.d/morph027-gitea.list
```

Install

```
>> apt-get update  
>> apt-get install gitea morph027-keyring
```

Start

```
>> systemctl enable --now gitea
```

Let's go with the browser for Gitea:

<http://myserver.com:3000>



Install and prepare Maria DB for Mediawiki:

```
>> apt update
>> apt upgrade
>> apt-get install apache2 mariadb-server php php-mysql libapache2-modphp
php-xml php-mbstring
>> apt-get install imagemagick php-curl
>> mysql -u root -p
MYSQL >> CREATE USER 'new_mysql_user'@'localhost' IDENTIFIED BY
'MySuperMariaDBPassword';
```




Install Wikimedia

#Download the MediaWiki software

```
>> cd /tmp/
```

```
>> wget https://releases.wikimedia.org/mediawiki/1.39/mediawiki-1.39.3.tar.gz
```

```
>> tar -xvzf /tmp/mediawiki-*.tar.gz
```

```
>> mkdir /var/lib/mediawiki
```

```
>> mv mediawiki-*/ * /var/lib/mediawiki
```

```
>> mysql -u root -p
```

```
MYSQL >> CREATE DATABASE my_wiki;
```

```
MYSQL >> CREATE USER 'wikiuser'@'localhost' IDENTIFIED BY 'database_password';
```

```
MYSQL >> GRANT ALL PRIVILEGES ON my_wiki.* TO 'wikiuser'@'localhost' WITH  
GRANT OPTION;
```

Only if your database is not running on the same server as your web server, you need to give the appropriate web server hostname (mediawiki.example.com in the example below):

```
MYSQL >> GRANT ALL PRIVILEGES ON my_wiki.* TO
```

```
'wikiuser'@'mediawiki.example.com' IDENTIFIED BY 'databaseSpassword';
```

1) Open a web browser and browse to the website address or path that you configured for MediaWiki in the web server software.

2) You will see the version of MediaWiki which you extracted and a link to "Please set up the wiki first." Click on the link to begin the configuration script. For reference (in case you want to go there directly), this is located in MediaWiki's mw-config directory (so using the example above you would go to <http://www.example.com/mw-config/>, or on a local machine <http://localhost/mw-config/>).

3) After selecting the language, the configuration script performs some environmental checks.

NOTE:

If need, you'll have to:

```
>> ln -s /var/lib/mediawiki /var/www/html/mediawiki
```

```
>> phpenmod mbstring  
>> phpnmod xml  
>> apt-get install php-intl  
>> systemctl restart apache2.service
```

4) Follow the instructions on the pages to connect to the database, create the MediaWiki administrator account, and to set further settings such as the logo or skin.

5) The script will populate the database and write a configuration file.

6) After the configuration script has finished, download the generated configuration file LocalSettings.php and place the file in the base directory of your MediaWiki installation.

LocalSettings.php contains all the information needed by MediaWiki to run.

Let's go with the browser for Mediawiki:

<http://myserver.com/mediawiki/>



PHPBB forum Install

wget <https://download.phpbb.com/pub/release/3.3/3.3.10/phpBB-3.3.10.zip>

Create the DB

```
mysql -u root -p
MYSQL >> CREATE DATABASE myphpbb;
MYSQL >> CREATE USER 'phpbbuser'@'localhost' IDENTIFIED BY
'superPasswordphpbb';
MYSQL >> GRANT ALL PRIVILEGES ON myphpbb.* TO 'phpbbuser'@'localhost' WITH
GRANT OPTION;
```

Decompress the phpBB archive to a local directory on your system.

Upload all the files contained in this archive (retaining the directory structure) to a web accessible directory on your server or hosting account.

Change the permissions on **config.php** to be writable by all (**666** or **-rw-rw-rw-** within your FTP Client)

Change the permissions on the following directories to be writable by all (**777** or **-rwxrwxrwx** within your FTP Client):

store/, cache/, files/ and images/avatars/upload/.

Point your web browser to the location where you uploaded the phpBB files with the addition of **install/app.php** or simply **install/**,

e.g.

<http://www.example.com/phpBB3/install/app.php>

or

<http://www.example.com/forum/install/>

Click the **INSTALL** tab, follow the steps and fill out all the requested information.

Change the permissions on **config.php** to be writable only by yourself (**644** or **-rw-r--r--** within your FTP Client)

phpBB should now be available, please **MAKE SURE** you read at least Section 6 below for important, security related post-installation instructions, and also take note of Section 7 regarding anti-spam measures.

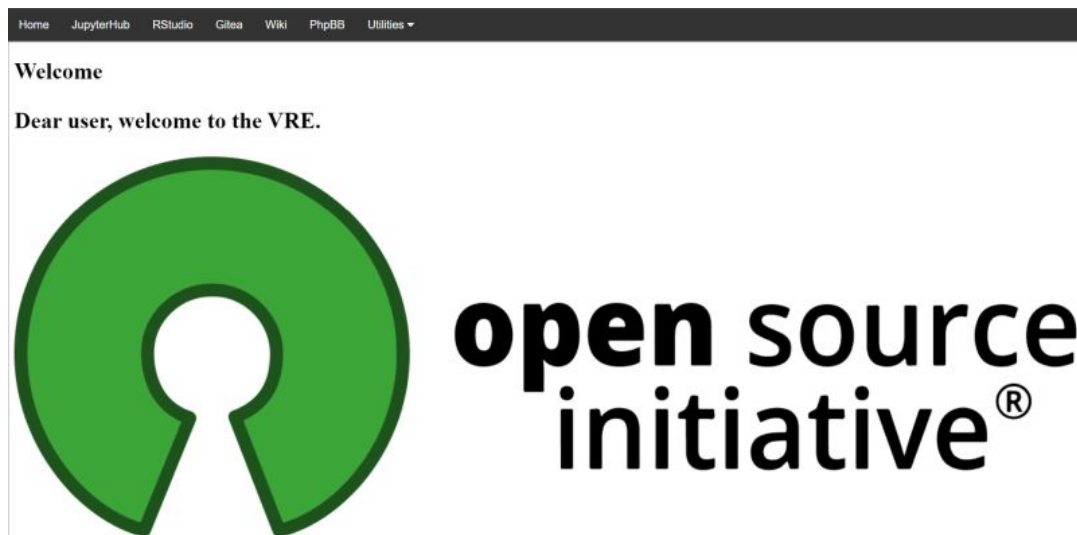
Security TIPS – rename (or delete) the "install" directory:

```
>> mv install/ nopeinstall/
```

For Apache there are .htaccess files already in place to do this for the most sensitive files and folders. We do however recommend to completely deny all access to the aforementioned folders and their respective subfolders in your Apache configuration.

On Apache 2.4, denying access to the phpbb folder in a phpBB instance located at /var/www/html/ would be accomplished by adding the following access rules to the Apache configuration file (typically apache.conf):

```
<Directory /var/www/html/phpbb/*>  
Require all denied  
</Directory>  
<Directory /var/www/html/phpbb>  
Require all denied  
</Directory>
```



Create a web page to connect all services in `/var/www/html/`

eg. `index.html`

```
<!DOCTYPE html>
<html>
<head>
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css">
<title>Dashboard</title>
<style>
* {
  box-sizing: border-box;
}

body {
  margin: 0;
}

.navbar {
  overflow: hidden;
  background-color: #333;
  font-family: Arial, Helvetica, sans-serif;
}

.navbar a {
  float: left;
  font-size: 16px;
  color: white;
  text-align: center;
  padding: 14px 16px;
  text-decoration: none;
}

.dropdown {
  float: left;
  overflow: hidden;
}

.dropdown .dropbtn {
  font-size: 16px;
  border: none;
  outline: none;
  color: white;
  padding: 14px 16px;
  background-color: inherit;
  font: inherit;
  margin: 0;
}
```

```

.navbar a:hover, .dropdown:hover .dropbtn {
  background-color: #33CEFF;
}

.dropdown-content {
  display: none;
  position: absolute;
  background-color: #f9f9f9;
  width: 100%;
  left: 0;
  box-shadow: 0px 8px 16px 0px rgba(0,0,0,0.2);
  z-index: 1;
}

.dropdown-content .header {
  background: #33CEFF;
  padding: 16px;
  color: white;
}

.dropdown:hover .dropdown-content {
  display: block;
}

/* Create three equal columns that floats next to each other */
.column {
  float: left;
  width: 33.33%;
  padding: 10px;
  background-color: #ccc;
  height: 550px;
}

.column a {
  float: none;
  color: black;
  padding: 16px;
  text-decoration: none;
  display: block;
  text-align: left;
}

.column a:hover {
  background-color: #ddd;
}

/* Clear floats after the columns */
.row:after {
  content: "";
  display: table;
  clear: both;
}

/* Responsive layout - makes the three columns stack on top of each other instead of next to each other */
@media screen and (max-width: 600px) {
  .column {
    width: 100%;
    height: auto;
  }
}
</style>
</head>
<body>

<div class="navbar">
  <a href="welcome.html" target="iframeMain">Home</a>
  <a href="http://www.mywebserver.org:8000/" target="_blank">JupyterHub</a>
  <a href="http://www.mywebserver.org:8787/" target="_blank">RStudio</a>
  <a href="http://www.mywebserver.org:3000/" target="_blank">Gitea</a>
  <a href="http://www.mywebserver.org/mediawiki/" target="iframeMain">Wiki</a>
  <a href="http://www.mywebserver.org/phpBB3/" target="iframeMain">PhpBB</a>
  <div class="dropdown">
    <button class="dropbtn">Utilities
      <i class="fa fa-caret-down"></i>
    </button>
    <div class="dropdown-content">
      <div class="row">
        <div class="column">
          <h3>Linux distros</h3>
          <a href="https://www.stackscale.com/blog/ubuntu/" target="iframeMain">Ubuntu</a>
          <a href="https://www.debian.org/index.en.html" target="iframeMain">Debian</a>

```

```

        <a href="https://www.redhat.com/en/technologies/linux-platforms/enterprise-linux"
target="iframeMain">Red Hat</a>
        <a href="https://www.gentoo.org/" target="iframeMain">Gentoo</a>
        <a href="https://getfedora.org/en/" target="_blank">Fedora</a>
        <a href="https://www.opensuse.org/" target="_blank">Open SUSE</a>
    </div>
    <div class="column">
        <h3>Other links</h3>
        <a href="http://www.google.com" target="iframeMain">GOOGLE</a>
        <a href="https://www.youtube.com/" target="iframeMain">You Tube</a>
        <a href="https://github.com/" target="iframeMain">GitHub</a>
    </div>
</div>
</div>
</div>
</div>
<iframe src="welcome.html" width="100%" height="2000" name="iframeMain"></iframe>
</html>

```

eg. welcome.html

```

<!DOCTYPE html>
<html>
<body onload="typeWriter()">
<h1>Welcome</h1>
<h1><p id="welcome"></p></h1>
<script>
var i = 0;
var txt = 'Dear user, welcome to the VRE.';
var speed = 50;

function typeWriter() {
    if (i < txt.length) {
        document.getElementById("welcome").innerHTML += txt.charAt(i);
        i++;
        setTimeout(typeWriter, speed);
    }
}
</script>

</body>
</html>

```