Instructions for bootstrap app

```
#get in as sudo
sudo bash

###### set up bootstrap app #####

## set up LAMP stack ##

#apache stuff
yum install httpd
systemctl start httpd.service
systemctl enable httpd.service

#db stuff
yum install mariadb-server mariadb
systemctl start mariadb

mysql_secure_installation
```

- change password (I used ricecrc) then y for everything # makes things more secure

```
systemctl enable mariadb.service

yum install php php-mysql
systemctl restart httpd


## make project folder and install dependencies ##
yum update
yum install python-devel mysql-devel python-setuptools mod_wsgi git
easy_install pip
pip install mysql-connector flask flask-sqlalchemy flask_migrate pymysql flask-marshmallow marshmallow-sqlalchemy watchdog flask_wtf wtforms_alchemy Flask-IIIF python-dotenv


## making project directory
cd /var/www/
```

```
git clone https://github.com/DavidYi/testiiif.git
```

```
## create config files
cd /etc/httpd/conf.d/
vi testiiif.conf

### in the file copy this into it ### (to get into insert mode press i when you first go in)
```

```
NameVirtualHost *
<VirtualHost *:80>
        ServerName [IP address]

        WSGIDaemonProcess testiiif processes=2
        WSGIScriptAlias / /var/www/testiiif/testiiif.wsgi
        WSGIApplicationGroup %{GLOBAL}

        <Directory /var/www/testiiif>
                WSGIScriptReloading On
                WSGIProcessGroup testiiif
                <IfVersion < 2.4>
                        Order allow, deny
                        Allow from all
                </IfVersion>
                <IfVersion >= 2.4>
                        Require all granted
                </IfVersion>
        </Directory>
        Alias '/stat' '/var/www/testiiif/static'
        <Directory /var/www/testiiif/static>
        <IfVersion < 2.4>
                        Order allow, deny
                        Allow from all
                </IfVersion>
                <IfVersion >= 2.4>
                        Require all granted
                </IfVersion>
        </Directory>

        Alias '/mnt/rdf/jcm10/crc_summer_dev' '/var/www/testiiif/mnt/rdf/jcm10/crc_summer_dev'
        <Directory /var/www/testiiif/mnt/rdf/jcm10/crc_summer_dev>
                <IfVersion < 2.4>
                        Order allow, deny
                        Allow from all
                </IfVersion>
                <IfVersion >= 2.4>
                        Require all granted
                </IfVersion>
        </Directory>

        Alias '/statmntpdf' '/var/www/testiiif/mnt/rdf/jcm10/
crc_summer_dev/miller_wright/HRC_Image_Archiving_Interface'

</VirtualHost>
#################################
```

```
## mount the rdf
cd /var/www/testiiif
yum install cifs-utils
chmod u+s /bin/mount /bin/umount /usr/sbin/mount.cifs
mkdir mkdir mnt
mkdir mnt/rdf

##Add the below line to /etc/fstab: (i.e. vi /etc/fstab)
//smb.rdf.rice.edu/research/ /var/www/testiiif/mnt/rdf cifs
mfsymlinks,rw,vers=3.0,sign,noperm,domain=ADRICE,user,noauto 0 0

##then run
exit
mount /var/www/testiiif/mnt/rdf
sudo bash
cd /var/www/testiiif
```

```
## set up database
# get access to db
mysql --password=ricecrc

# use address of the vm
GRANT ALL PRIVILEGES ON *.* to 'root’@‘[address of vm]’
IDENTIFIED BY 'ricecrc' WITH GRANT OPTION;
```

```
## create database to use in app (if you use different name then change in code accordingly)
CREATE DATABASE test;

# for mysql
exit

#change the connection link
sudo vi testiiif.py
##look for the database link and change the IP address to the current one (for me it was 10.134.196.58)

flask db init
flask db migrate -m “set up”
flask db upgrade


mkdir static/thumbnails

## Fix permission stuff
setsebool -P httpd_use_cifs=1

#give permission to access the folder we use ( I use mnt/rdf/jcm10/crc_summer_dev/)
chmod 755 mnt/rdf/jcm10/crc_summer_dev/
chmod 770 testiiif.wsgi
chmod 775 testiiif.py
chmod 775 static/

chown -R apache:apache testiiif.py
chown -R apache:apache testiiif.wsgi
chown -R apache:apache static/

setsebool -P httpd_can_network_connect_db=1
setsebool -P httpd_unified=1

systemctl restart httpd


##### iiif server installation ####
```

```
#turn off se linux
setenforce=0
yum install git wget mod_wsgi python-devel bzip2
cd /opt
git clone https://github.com/loris-imageserver/loris.git
cd loris

wget https://bootstrap.pypa.io/get-pip.py
python get-pip.py
pip install -r requirements.txt --ignore-installed

yum install libjpeg-turbo libjpeg-turbo-devel \
   freetype freetype-devel \
   zlib-devel \
   libtiff-devel
pip install Werkzeug
pip install Pillow
yum install gcc



##set up working directories and permissions
mkdir /var/www/loris2
useradd -d /var/www/loris2 -s /sbin/false loris


#CHANGE CONFIG FILE
###change the config file to point at the images in /opt/loris/etc file -- variable is "scr_img_root"
###Default is: /usr/local/share/images/
###I used: src_img_root = '/var/www/testiiif/mnt/rdf/jcm10/crc_summer_dev/hi' # r--


###change the config file in /etc/httpd/conf/httpd.conf file
## add the line below where listen is being described to make it organized
Listen 81
```

###Now you can install Loris:

```
cd /opt/loris
python setup.py install
```

### some dependencies will be out of date so run the line below

```
pip install flask
pip install PyMuPDF
```

####now create the below files (code in this repo):

#######loris.conf in /etc/httpd/conf.d/

###### content below ##################

```
NameVirtualHost *:81
<VirtualHost *:81>
         AllowEncodedSlashes On

        WSGIDaemonProcess loris user=loris group=loris processes=10
threads=15 maximum-requests=10000
        WSGIScriptAlias /loris /var/www/loris2/loris.wsgi
        <directory /var/www/loris2>
            WSGIProcessGroup loris
            WSGIApplicationGroup %{GLOBAL}
            WSGIScriptReloading On
        Require all granted
        </directory>

</VirtualHost>
```

######### end content ################

#######loris.wsgi in /var/www/loris2/

###### content below #################

```
#!/usr/bin/env python

from loris.webapp import create_app
application = create_app(config_file_path='/opt/loris/etc/
loris2.conf')
```

######## end content #################

#######loris.te in /opt/loris/

###### content below #################

```
module loris 1.0;

require {
        type httpd_t;
        type var_t;
        class file { write read getattr open };
}

#============= httpd_t ==============
allow httpd_t var_t:file { write read getattr open };
```

######## end content #################

####Then compile the SELinux rules as below:

```
cd /opt/loris/
checkmodule -M -m -o loris.mod loris.te
semodule_package -m loris.mod -o loris.pp
semodule -i loris.pp

mkdir /var/log/loris2
chown loris /var/log/loris2
mkdir /var/cache/loris
chown loris /var/cache/loris

httpd -k start
```

##Then turn SELinux back on:

```
setenforce=1
```