



# How to Install and Configure ESP on Ubuntu

## How to Install and Configure ESP on Ubuntu

### Revision History

Version Number	Modification Date	By	Description of Changes
1.0	16 Sep 2012	R. Schaaf	Initial version
1.1	18 Sep 2012	R. Schaaf	Create the /srv/esp30/data/fake directory
1.2	01 Feb 2013	B. Zambarano	Minor updates
1.3	7/1/2014	C. Chacin	Updates for 3.1 and merging all installation of prerequisites.
1.4	7/3/2014	C. Chacin	Adding apache and sql backup
1.5	7/8/2014	C. Chacin	Finalizing all apache configuration and formatting doc
1.6	8/21/2014	C. Chacin	Adding folder structure configuration and settings in applications.ini
1.7	8/28/2014	C. Chacin	Added how to clean data and reload
1.8	05 Oct 2014	K. Eberhardt	Added notes for dev install and modified some commands in sections 4-5
1.9	16 Jan 2015	K. Eberhardt	Updated some commands in section 8
1.10	17 Nov 2016	B. Zambarano	Updates to repository (git)
1.11	15 Feb 2017	K. Eberhardt	Modifications to installation steps and commands
1.12	13 Jul 2017	B. Zambarano	Additions of steps describing data ETL, configuration and validation

# Contents

---

<b>1 Overview</b>	<b>1</b>
<b>2 Prerequisites</b>	<b>1</b>
<b>3 Install ESP</b>	<b>1</b>
<b>4 Create an esp database and esp database user</b>	<b>2</b>
<b>5 Create the directories and files expected by ESP</b>	<b>3</b>
<b>6 Configure the ESP application settings</b>	<b>3</b>
<b>7 Setting up the data feed</b>	<b>4</b>
<b>8 Using ESP</b>	<b>5</b>
<b>9 Set up the Apache web server</b>	<b>6</b>
<b>10 Configure database backups</b>	<b>7</b>
<b>11 Configure ESP software tools daily run</b>	<b>8</b>
<b>12 Setting up disease detection</b>	<b>9</b>
<b>13 How to clean up and reload ESP data</b>	<b>9</b>
<b>Appendix A Editing apache conf file</b>	<b>12</b>
<b>Appendix B PostgreSQL database backup</b>	<b>13</b>

### 1 Overview

This document describes the procedure for configuring ESP 3.3 on Ubuntu Linux. The versions of software used to prepare these instructions are:

Linux: Ubuntu 16.04 Server Edition (64-bit)  
PostgreSQL: 9.5  
ESP: 3.3, running in a virtual environment using  
Python 2.7  
Django 1.4

The installation steps are to:

- Install ESP
- Create an esp database and database user
- Create directories and files expected by ESP
- Configure the application settings

This document may be used a guide for installing ESP on other Linux systems, but keep in mind that there are differences between Linux systems, particularly in package management and user creation and permissions.

Unless otherwise specified, all commands are executed from the linux bash shell prompt. It is assumed the installer has sudo privs.

### 2 Prerequisites

This installation is for an Ubuntu linux 16.04 server, with SSH for remote logins. The bash shell should be the default.

This installation expects that the git software version control system is installed.

The ESP installation will require either a local smtp service, or an email account on an accessible smtp server.

### 3 Install ESP

1. As the system administrator, create the esp user. Standard ESP installation will be to /srv/esp/prod or /srv/esp/test. Use /srv/esp as the esp home directory.

```
sudo useradd -d /srv/esp/ -m -p PASSWORD esp
```

2. Add the user to the 'sudo' group

```
sudo adduser esp sudo
```

3. Switch to the esp user

```
sudo su esp
```

4. Clone the esp repository in /srv/esp/[prod or test]

```
cd ~  
git clone https://bitbucket.org/Commoninf/esp.git [prod or test]
```

## **Error! No text of specified style in document.**

as in

```
git clone https://bitbucket.org/Commoninf/esp.git prod
or
git clone https://bitbucket.org/Commoninf/esp.git test
```

5. Checkout the desired branch and give it a “local” branch name. Unless you specifically require an earlier release, use the latest tagged release branch

```
cd /srv/esp/[prod or test]
git checkout v3.3.[x] -b version3.3.[x]
```

6. Setup ESP Dependencies by installing the packages that ESP depends on

```
sudo ./install-system-dependencies.sh
```

If errors are encountered, manually edit the file and make modifications to the versions as required. (For non-Ubuntu linux systems, inspect this script to determine what dependencies to install manually).

7. Install ESP

```
./install.sh
```

8. Install any disease detection plugins you require:

```
./setupPlugins.sh
```

## **4 Create an esp database and esp database user**

1. As the system administrator, create an “esp” role in the PostgreSQL database by entering the following at the command prompt:

```
sudo -u postgres createuser -P -s esp
```

Enter your sudo password, and in response to the “Enter password for new role” and “Enter it again” prompts, enter the password for the new esp user.

The options passed into the createuser command control the following:

- P: assign a password to the new role (you should be prompted)
- s: role will be superuser

If the postgres database will contain databases other than those managed by ESP, you may choose for security reasons to not provide superuser. Some distributed ESP SQL reports that use the COPY command will not work.

2. Create an “esp” database by entering the following at the command prompt:

```
sudo -u postgres createdb -O esp esp
```

The options passed into the createdb command control the following:

**-O esp:** database user to own the new database

### 5 Create the directories and files expected by ESP

1. As the ESP user, create the directories and files expected by ESP by entering:

```
mkdir -p /srv/esp/data/{case_reports,epic,load_reports}
mkdir -p /srv/esp/data/epic/{archive,incoming,error}
mkdir -p /srv/esp/data/fake
```

Use of the folder name “epic” is vestigial. These folders are used for text data from any source.

2. Create the ESP Log File

```
sudo touch /var/log/esp.log
sudo chown esp:esp /var/log/esp.log
```

### 6 Configure the ESP application settings

1. Create initial versions of ESP’s **application.ini** and **secrets.ini** configuration files:

```
cd /srv/esp/[prod or test]
./bin/esp makeini
```

These files will be created in the /srv/esp/[prod or test]/etc/ folder.

2. Edit the secrets.ini file:

```
vi etc/secrets.ini
```

For the **database\_password**, enter the password for the “esp” database role.

For the **secret\_key**, enter a random string of at least 32 characters.

3. Edit the **application.ini** file:

```
vi /etc/application.ini
```

In the **[General]** section, edit the following settings:

```
data_folder = /srv/esp/data
icd10_support = True
```

In the **[Database]** section, edit the following settings:

```
db_name = esp
username = esp
```

In the **[Web]** section, edit the following settings:

```
static_url = /media/static/
```

In the **[ETL]** section, edit the following settings:

```
load_report_dir = /srv/esp/data/load_reports/
```

## **Error! No text of specified style in document.**

In the **[Logging]** section, edit the following setting:

```
log_file = /var/log/esp.log
```

In the **[HEF]** section, edit the following setting:

```
thread_count = 0
```

Other settings may be modified as necessary.

4. As the ESP user, from the ESP installation directory, populate the media/static folder:

```
./bin/esp collectstatic
```

5. As the ESP user, from the ESP installation directory, initialize the ESP database by entering:

```
./bin/esp syncdb
```

When prompted to define a superuser, respond as follows:

Would you like to create one now? (yes/no): yes

Username (Leave blank to use 'superuser'): esp

E-mail address: <your email address>

Password: <password for the esp superuser>

Password (again): <password for the esp superuser>

Ignore the "IntegrityError: insert or update on table  
"conf\_reportabledx\_code" violates foreign key constraint  
"conf\_reportabledx\_code\_dx\_code\_id\_fkey" if received.

6. Load the static data for the ESP application by entering:

```
./bin/esp loaddata ESP/static/fixtures/dx_code.json
```

```
./bin/esp loaddata ESP/static/fixtures/loinc.json
```

```
./bin/esp loaddata ESP/static/fixtures/ndc.json
```

7. Run syncdb again to install the objects from the loaded static data. No error should be received.

```
./bin/esp syncdb
```

8. Test that basic web services are working by running the web server built into Django by entering:

```
./bin/esp runserver
```

Browse to <http://localhost:8000> and verify that you are able to log into the ESP application using the superuser account created earlier.

## **7 Setting up the data feed**

Each site will need a data feed that provides data to ESP in the standard text file format. Please see the document [ESP\\_Filespec.xlsx](#) for details. For Epic sites, Commonwealth

## How to Install and Configure ESP on Ubuntu

informatics can share SQL-based extract scripts for Epic Clarity, or MUMPS scripts for Epic Cache. These must be modified to conform to site-specific Epic configuration.

There are two data feed steps:

1. Set up the historic data extract and load. The ESP database should have at least two years of data going back from the present in order to make good determinations of disease states for conditions that may be chronic.
2. Set up the nightly data extract and load. This is for ongoing data extraction and loading

The ESP command “load\_epic” is used to load the data file. ESP was developed around Epic systems, and this name is vestigial.

These are highly site-specific activities and must be designed, developed and tested at each site. Using Commonwealth-provided scripts will greatly reduce the time for this task, but for systems starting from the ESP filespec Excel spreadsheet, expect at least 4 weeks of person-effort.

## 8 Using ESP

### ESP Commands

All ESP commands take this form:

```
$ $ESP_HOME/bin/esp command [--argument_one {optional parameters}] [--argument_two]
(Most commands have single character versions of arguments, used with a single dash)
```

For a full listing of available commands:

```
$ $ESP_HOME/bin/esp help
```

For help with a specific command:

```
$ $ESP_HOME/bin/esp command --help
```

### Load EMR data

A typical command to load data might look like this:

```
$ $ESP_HOME/bin/esp load_epic
```

The use of Epic in the command name is vestigial. This command loads text data from a set of files in the ESP defined input format. See the document “ESP\_Filespec.xlsx”.

This command will take files from the configured data directory `./epic/incoming` and load them to the ESP database tables. Details are written to the `esp.log` file, and table specific load results are written to reports in `load_reports` directory.

### Code mapping



## **Error! No text of specified style in document.**

Each individual EMR system uses a unique set of codes to designate tests. We refer to these as abstract labs. Once EMR lab data is loaded to ESP, the native codes of relevant lab tests must be mapped to the named abstract labs used by ESP's disease detection logic. Mapping is stored in the table `conf_labtestmap`.

To perform lab mapping, run:

```
$ $ESP_HOME/bin/esp concordance
```

The concordance command populates a table `emr_labtestconcordance`, which lists all unique native lab test names and code currently loaded in the ESP `emr_labresults` table. These are available for efficient searches for mapping. A tool is provided in the Administrative web interface, (setup described in next section), In the menu bar, look under *Administration* --> *Unmapped Lab Tests Report*. This form page uses the set of search strings provided for each disease detection plugin to detect potentially unmapped lab tests. In the Unmapped Lab Tests Report interface, the user can either map a lab, or mark it to be ignored. Mapped and ignored labs do not appear in the Report again.

### **Detecting cases**

Once all labs are mapped using the Admin UI interface, ESP divides the task of detecting disease cases into two distinct parts. First, a uniform table of events is generated from raw medical record data by *HEF*, the Heuristic Events Framework.

```
$ $ESP_HOME/bin/esp hef
```

Second, a table of cases is generated by searching the events table for patterns indicating disease. This function is carried out by *Nodis*, the NOTifiable DISeases framework.

```
$ $ESP_HOME/bin/esp nodis
```

Additional options for both commands are available by passing the '--help' flag.

### **Reporting cases**

The process for electronically reporting cases is heavily dependent upon the requirements, both technical and functional, of the intended recipient. ESP provides two report generation commands: `case_report` and `lab_report`. The `case_report` command generated hl7 2.3.1 version lab report message, the `lab_report` command generated hl7 2.5.1 version lab report messages. Both require extensive coding for specific recipient requirements. Unfortunately, this mapping process is not currently well documented. We are working on developing a document set for this purpose.

## **9 Set up the Apache web server**

Set up the Apache web server following the steps specified below. These steps should be executed from an interactive shell that is running as root. These steps are specific to Ubuntu Linux distributions.

1. If not already installed, install the apache package including mod wsgi and start the service.

```
sudo apt-get install apache2 libapache2-mod-wsgi
```

## How to Install and Configure ESP on Ubuntu

2. Edit `/etc/group` and add the `apache` user to the `esp` group, as in:

```
sudo useradd -G esp apache (this is the apache user name)
```

3. Make a copy of `django.wsgi.sample` in the `$ESP_HOME/share` folder, then edit it as described below:

```
sudo cp /srv/esp/prod/share/django.wsgi.sample
/srv/esp/prod/share/django.wsgi

import os
import sys

os.environ['DJANGO_SETTINGS_MODULE'] = 'ESP.settings'

# Enable this to prepend your ESP src folder to the beginning of
PYTHONPATH, in
# case an older version of Django is installed system-wide.

sys.path.insert(0, '/srv/esp/prod')

execfile('/srv/esp/prod/bin/activate_this.py')
import django.core.handlers.wsgi
application = django.core.handlers.wsgi.WSGIHandler()
```

4. Create a file named `/etc/apache2/sites-available/<any server file name>` and populate it with the configuration information listed in Error! Reference source not found.A. Give yourself permissions to create new doc and save in this location.

```
sudo a2enmod wsgi
sudo a2enmod rewrite
sudo a2ensite <any server name file>
# Start (or restart) the service
sudo /etc/init.d/apache2 restart
```

## 10 Configure database backups

1. Add a logrotate configuration to back up the `esp` database by copying the backup script from **Appendix B** to `/srv/esp/backup/postgresql-esp.logrotate` and then follow the steps specified below. These steps should be executed from an interactive shell that is running as `root`.

```
# Fix up the permissions for the postgresql-esp.logrotate script
chown root:root /srv/esp/backup/postgresql-esp.logrotate
chmod 644 /srv/esp/backup/postgresql-esp<version>.logrotate

# Create initial versions of the "log" files
cd /srv/esp/backup
chown postgres:postgres .
touch esp.daily.dump
touch esp.weekly.dump
touch esp.monthly.dump
chown postgres:postgres *.dump
chmod 644 *.dump
```

## Error! No text of specified style in document.

2. Add the following entry to the crontab for root to run the database backups at 1pm, by which time the daily ESP update should have completed.

```
# m h dom mon dow command
0 13 * * * /srv/esp/scripts/db_backup.sh
```

3. Open the **/etc/sudoers** file and search for a line specifying the **requiretty** option. If this line exists, edit the sudoers file to remove this option

```
Defaults requiretty
```

4. Create the **/srv/esp/scripts/db\_backup.sh** file and populate it with the following:

```
#!/bin/bash
#~~~~~
#~~~~~
# @author: Rich Schaaf
# @organization: Commonwealth Informatics, Inc
# @copyright: (c) 2013 Commonwealth Informatics, Inc.
# @license: LGPL
#
#~~~~~
#~~~~~
```

```
/usr/sbin/logrotate /srv/esp/backup/postgresql-esp.logrotate
```

5. Set the permissions on the **/srv/esp/scripts/db\_backup.sh** script as follows:

```
# Fix up the permissions for the postgresql-esp script
chown root:root /srv/esp/scripts/db_backup.sh
chmod 755 db_backup.sh
```

## 11 Configure ESP software tools daily run

A file called `daily_batch.sh` is distributed with ESP in the `./share` folder. This shell script can be modified for use to run the daily processes for ESP. This file should be modified and saved in a folder named something like:

```
/home/esp/esp<version>/scripts/
```

Add a cron entry in your system to run this batch script and load ESP data daily following the instructions below:

To edit type:

```
$ crontab -e
```

Add a similar text to the crontab with your desired time to run:

```
# ESP Production Daily Run
#
30 22 * * * /srv/esp30/scripts/daily_
batch_parll.sh
```

Save and quit

To see the new crontab entry type:

```
$ crontab -l
```

### 12 Setting up disease detection

ESP is distributed with a number of Python plugins for disease detection. Each disease has separate plugin. These plugins are installed via an interactive text-based screen started from the bash shell prompt:

```
$ sudo ~/esp/setupPlugins.sh
```

This is also described in section 3 above. The plugins are installed into `~/esp/src/[disease name directories]`

Configuring the disease plugins is a four-step process:

1. Run the esp concordance command

```
$ $ESP_HOME/bin/esp concordance
```

2. The concordance command populates a table `emr_labtestconcordance`, which lists all unique native lab test names and code currently loaded in the ESP `emr_labresults` table. These are available for efficient searches for mapping. A tool is provided in the Administrative web interface, In the menu bar, look under *Administration* --> *Unmapped Lab Tests Report*. This form page uses the set of search strings provided for each disease detection plugin to detect potentially unmapped lab tests. In the Unmapped Lab Tests Report interface, the user can either map a lab, or mark it to be ignored. Mapped and ignored labs do not appear in the Report again. The lab concepts required by the plugin must be mapped to the appropriate local labs. The required lab concepts are determined by inspection of the plugin file `[disease_name].py`

3. Run the esp commands `hef` and `nodis`:

```
$ $ESP_HOME/bin/esp hef
$ $ESP_HOME/bin/esp nodis
```

4. Review the resulting cases and perform appropriate clinical validation to confirm that data configuration is correct and working. Cases are available for review in the ESP administrative web interface.

Step 2 (lab concept mapping) requires site support from staff familiar with local lab naming and the corresponding lab test. Set 4 (case validation) requires site support from staff with ability to do case review in the EHR.

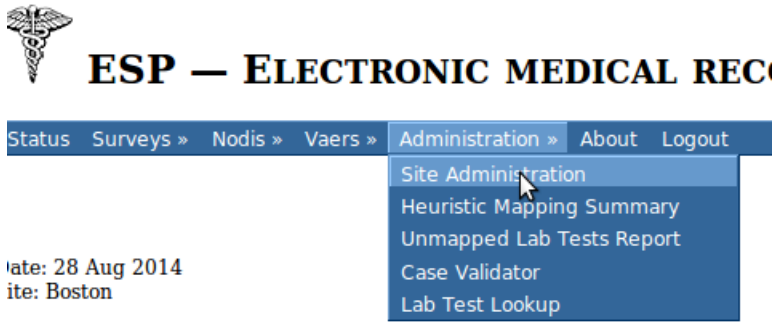
### 13 How to clean up and reload ESP data

Sometimes it is helpful to know how to clean all the data loaded and start over again. There are several procedures that can be run to accomplish this task but the simplest is to delete all the provenance entries from the administration UI except the first two and it will take care of all the related records in the db.

Below are the steps and screen shots to perform this task:

**Error! No text of specified style in document.**

1. From the ESP UI Select Administration/Site Administration from the menu



2. Click on Provenances from the Emr section

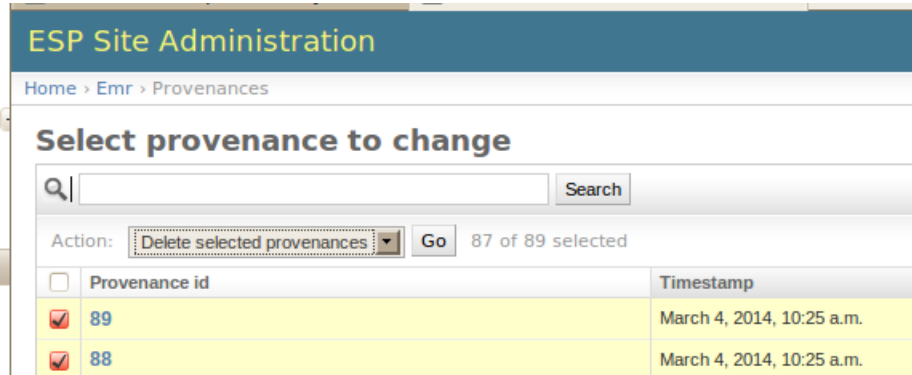
Vaccine manufacturer maps	<a href="#">+ Add</a>	<a href="#">Change</a>
<b>Emr</b>		
Allergys	<a href="#">+ Add</a>	<a href="#">Change</a>
Encounter type maps	<a href="#">+ Add</a>	<a href="#">Change</a>
Encounters	<a href="#">+ Add</a>	<a href="#">Change</a>
Hospital_ problems	<a href="#">+ Add</a>	<a href="#">Change</a>
Immunizations	<a href="#">+ Add</a>	<a href="#">Change</a>
Lab Orders	<a href="#">+ Add</a>	<a href="#">Change</a>
Lab Specimen Observations	<a href="#">+ Add</a>	<a href="#">Change</a>
Lab Specimens	<a href="#">+ Add</a>	<a href="#">Change</a>
Lab Test Results	<a href="#">+ Add</a>	<a href="#">Change</a>
Lab infos	<a href="#">+ Add</a>	<a href="#">Change</a>
Order_id infos	<a href="#">+ Add</a>	<a href="#">Change</a>
Patient_ addr	<a href="#">+ Add</a>	<a href="#">Change</a>
Patient_ extra datas	<a href="#">+ Add</a>	<a href="#">Change</a>
Patient_ guardians	<a href="#">+ Add</a>	<a href="#">Change</a>
Patients	<a href="#">+ Add</a>	<a href="#">Change</a>
Pregnancys	<a href="#">+ Add</a>	<a href="#">Change</a>
Prescriptions	<a href="#">+ Add</a>	<a href="#">Change</a>
Problems	<a href="#">+ Add</a>	<a href="#">Change</a>
<b>Provenances</b>	<a href="#">+ Add</a>	<a href="#">Change</a>

3. Select all records by clicking on the top left check box next to the Provenance id label and then unselect two checkboxes for the record 1 and 2 corresponding to CLEANUP and SYSTEM

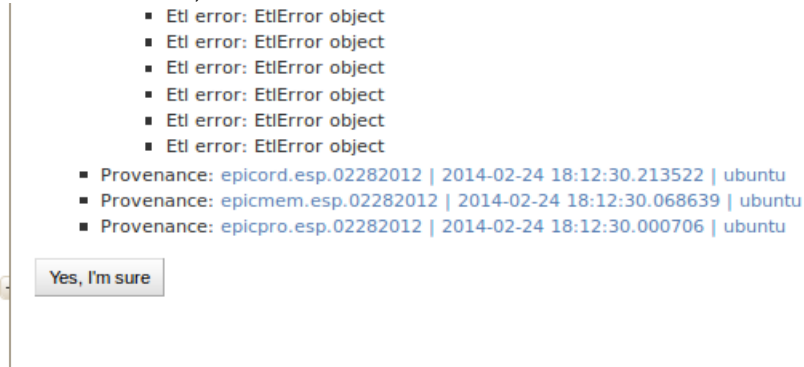
<input checked="" type="checkbox"/>	7	Feb. 24, 2014, 7:22 p.m.	epicvis.esp.02282012
<input checked="" type="checkbox"/>	6	Feb. 24, 2014, 6:12 p.m.	epicres.esp.02242012
<input checked="" type="checkbox"/>	5	Feb. 24, 2014, 6:12 p.m.	epicord.esp.02282012
<input checked="" type="checkbox"/>	4	Feb. 24, 2014, 6:12 p.m.	epicmem.esp.02282012
<input checked="" type="checkbox"/>	3	Feb. 24, 2014, 6:12 p.m.	epicpro.esp.02282012
<input type="checkbox"/>	2	Jan. 1, 1900, midnight	CLEANUP
<input type="checkbox"/>	1	Jan. 1, 1900, midnight	SYSTEM
89 provenances			

## How to Install and Configure ESP on Ubuntu

4. On that same screen select Delete selected provenances from the Action drop down and then click on the Go button next to it



5. This will calculate all the related records and elements in the database related to those loaded files and will display a summary of all the records that will be deleted for confirmation. It will prompt you: *Are you sure you want to delete the selected provenances? All of the following objects and their related items will be deleted.* Simply scroll down to the bottom and click on the "Yes, I am sure" button:



Once you have cleaned up all the records simply run the load epic command with the reload option. If you don't specify:

```
$ $ESP_HOME/bin/esp load_epic --reload
# Loads from default folder, /srv/esp-data/epic/incoming
```

**Error! No text of specified style in document.**

## Appendix A Editing apache conf file

The `/etc/apache2/sites-available/<any server file name>` file should be edited to match the following listing. The highlighted values may need to be changed depending on the location of the directory where the ESP application is installed:

```
sudo cp ./home/esp/esp<version>/share/apache.conf.sample
/etc/apache2/sites-available/<any server file name>
```

(the file name should be the domain name of the server)

```
#~~~~~
#
#                               Apache configuration for ESP
#~~~~~

WSGIRestrictStdout Off

WSGIScriptAlias / /srv/esp/prod/share/django.wsgi
WSGIDaemonProcess esp
WSGIScriptAlias / /home/esp/esp<version>/share/django.wsgi

<VirtualHost <servername>:80>
    ServerName <server name>
    ServerAdmin <server admin email>

    Alias /static/ /srv/esp/prod/ESP/media/static/
    Alias /media /srv/esp/prod/ESP/media

    <Location "/media/">
        SetHandler None
        Allow from all
    </Location>

    <LocationMatch "/media/">
        SetHandler None
        Allow from all
    </LocationMatch>

    <LocationMatch "\.(jpg|gif|png)$">
        SetHandler None
        Allow from all
    </LocationMatch>
    ### Block all trace requests (all vHosts need this!!)
    ### The following code should be included every VirtualHost.
    ### It returns a Forbidden on any TRACE or TRACK request.
    ### TRACE requests are used for debugging, they tell the server
    ### to return a text version of the REQUEST. This is not
    ### needed and can be used for abuse/cross-site-scripting
    ###
    ### See: http://www.w3.org/Protocols/rfc2616/rfc2616-sec9.html
    ### for more about TRACE
    ###
    RewriteEngine On
    RewriteCond %{REQUEST_METHOD} ^(TRACE|TRACK|CONNECT)
    RewriteRule .* - [F]
    <Directory proxy:*>
        Order Deny,Allow
        Deny from all
    </Directory>
    <IfModule mod_proxy.c>
        ProxyRequests Off
    </IfModule>
</VirtualHost>
```

### Appendix B PostgreSQL database backup

The following configuration should be saved as `/srv/esp/backup/postgresql-esp.logrotate`.

```
#
# Logrotate configuration for automatic daily backups of the PostgreSQL
# database 'esp' to '/srv/esp/backup/esp.daily.dump'.
#
# Please note, this configuration does not make any attempt to divine your
# database name from ESP's config files. Edit it if you are using something
# different than 'esp'. You may wish to run logrotate in the context of
# user 'postgres' to avoid dealing with DB authentication issues. The user
# running logrotate will need write permission in /srv/esp/backup.
#

/srv/esp/backup/esp.daily.dump {
    daily
    rotate 7
    dateext
    nomissingok
    create
    nocompress
    nocopy
    prerotate
        test -x /usr/bin/pg_dump || exit 0
        sudo -u postgres /usr/bin/pg_dump esp -F c > /srv/esp/backup/esp.daily.dump
    endscrip
}

/srv/esp/backup/esp.weekly.dump {
    weekly
    rotate 4
    dateext
    nomissingok
    create
    nocompress
    nocopy
    prerotate
        test -x /usr/bin/pg_dump || exit 0
        sudo -u postgres /usr/bin/pg_dump esp -F c > /srv/esp/backup/esp.weekly.dump
    endscrip
}

/srv/esp/backup/esp.monthly.dump {
    monthly
    rotate 12
    dateext
    nomissingok
    create
    nocompress
    nocopy
    prerotate
        test -x /usr/bin/pg_dump || exit 0
        sudo -u postgres /usr/bin/pg_dump esp -F c > /srv/esp/backup/esp.monthly.dump
    endscrip
}
```