ESP DOCS

How to Install and Configure ESP 3.7 on Ubuntu and Red Hat Enterprise Linux

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
1.13	29 Nov 2017	K. Eberhardt	Rewrite of Apache section. Reformatting. Rewrite of db backups section. Rewrite of daily batch process. Additional minor changes throughout. Added section for configuring daily status emails.
1.14	01 Mar 2019	J. Miller	Updated for ubuntu 18.04 – other minor updates throughout.
1.15	11 Mar 2019	J. Miller	Minor updates
1.16	May 28, 2019	J Miller	Cleaned up and reordered with Karen
2.0	July 9 2019	J miller	Converted to redhat 7.6 Installation Guide
2.1	August 15, 2019	J.Miller	Added ESP log rotation and fixes to DB backup steps
2.2	August 28 2019	J Miller	Updated Log Monitoring Instructions Redhat 7.6 v3
2.3	October 21, 2019	J. Miller	Added note about no special characters in Secret Key.
2.4	May 2020	J Miller	Updated for ESP 3.5
2.5	June 2020	J. Boyer	Consolidated document containing instructions for multiple environments
2.6	July 2020	J. Miller	Minor updates from Bob Z's feedback
2.7	Sept 2020	J.Miller	Added section 13 – Configure esp log file rotation
2.8	March 2021	J.Miller	Minor typos and path location updates – added esp_tools
2.9	April 2021	J. Miller	Updates to log monitoring section
3.0	June 2021	J.Miller	Updated for postgresql 13
3.1	Sep. 2023	J. Miller	Updated for Python3.8 and ESP 3.7 and removed mssql install steps – these will be put in a separate install doc

Contents

1 Document Formatting

Instructions in this document are color-coded to indicate which Operating System they apply to.

General instructions are in normal black text, but will be preceded by the appropriate OS as needed.

UBUNTU
RHEL

2 Installation Overview

This document describes the procedure for configuring ESP on an Ubuntu or Redhat Linux Server.

The versions of software used to prepare these instructions were:

Operating System:
UBUNTU 20.04.x Server Edition (64-bit)
UBUNTU 22.04.x Server Edition (64-bit)

OR
RHEL Linux: 8.6 (64-bit)
RHEL Linux: 8.9 (64-bit)

NOTE: Later O.S versions are generally supported - but may require slight modifications to these instructions.
Please check with the Commonwealth Informatics team if you have questions (esp_support@commoninf.com)

Common Pkgs:
  PostgreSQL: 13.x
  Apache or Httpd version 2 .4+
  ESP: 3.7.x Running in a virtual environment using
      Python 3.8
      Django 3.2

The Linux server should have ssh, git, access to a local SMTP service, and access to gitlab.com and pypi.org

These Installation steps are covered in Chapters 1 - 13:

- Create the esp user and install system prerequisites
- Download the esp project and run the install system dependencies shell script
- Create the esp database and esp database user and directories and files needed by ESP
- Install the Python virtual environment and configure the application settings and create the UI user
- Configure the Apache or httpd Web server
- Configure Database Backups

Additional topics once the system is running are covered starting at Chapter 14. These include the following:

- Updating to the latest version of ESP
- Setting up esp log rotation and log file monitoring
- Setting up the data import to ESP
- ESP Basics
- Daily Status Reports
- Daily Cron job setup
- Basic disease detection setup
- Deleting test data

This document should be used as 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 privileges to create the needed account and directories.

3 Prerequisites

SSH: This installation is for a Linux server with SSH for remote logins. The bash shell should be the default.

GIT: This installation expects that the git software version control system is installed.
To install git, run the following command:

```
UBUNTU:
 sudo apt install git

RHEL:
 sudo  yum install @development redhat-lsb-core
```

If asked - accept any keys by typing y - and hitting enter

NOTE: RHEL servers need to register with a subscription manager to install packages and updates.
Type the following to see if a server is registered:

```
 sudo subscription-manager list
```

eMail: In order to send status and error emails, the ESP installation will require either a local SMTP service, or an email account on an accessible SMTP server.

Access for Admin: Ensure firewall rules permit the following inbound connections on the application server:
 Port: 22 (needed for SSH by sysadmins)
 Ports: 443, 80 (needed for HTTP/HTTPS access for ESP user interface)

DB Access: If the database is on a different server, or remote access to the database is required, ensure that firewall rules permit an inbound connection to the database server to the following port :
 Default port is 5432

NOTE on SELinux: SELinux security context enforcement is specific to how SELinux is configured on your system. If SELinux is enabled, you will need to update object security contexts.

NOTE on PHI: The ESP database will contain PHI once data has been loaded for a participating site. Every precaution should be taken to ensure that the Server is not directly accessible to anyone outside of that Organization. The Server and the Network where it is located should be locked down to only allow access from certain Internal hosts or networks and/or VPN connections.

4 Create the ESP User and setup prerequisites

1. As the system administrator, create the esp user.

The standard ESP installation will install to /srv/esp/prod/
If possible also use /srv/esp as the esp home directory as shown below.
Run the command below and enter the account information and password when prompted.

```
sudo adduser --home /srv/esp/ --shell /bin/bash esp
```

2. Add the user to the 'sudo' group.

```
UBUNTU:
  sudo usermod -aG sudo esp
RHEL:
  sudo usermod -aG wheel esp
```

3. Make the /srv/esp directory readable for other users (like postgres)

```
sudo chmod 755 /srv/esp
```

5 Download the ESP projects and run system installation script

1. Switch to the esp user

```
sudo su - esp
```

2. Clone the ESP Project to /srv/esp/prod

```
cd /srv/esp
git clone https://gitlab.com/ESP-Project/ESP.git prod
```

> **Note**: If you receive git errors about "server certificate verification failed" , you may be going through a firewall or repository that does not trust this host. Try to resolve by running the following command and then rerunning the git clone:

```
git config –global http.sslverify "false"
```

> The permanent solution would be to have gitlab.com added to an "allowed" list on the local repository.

3. Clone the ESP Tools Project to /srv/esp/esp_tools

```
git clone https://gitlab.com/ESP-Project/esp_tools.git esp_tools
```

4. Checkout the required ESP branch and give it a "local" name.

Note: Unless you specifically require an earlier release, use the latest tagged release branch.

To see the latest tagged release enter the following commands:

```
cd /srv/esp/prod
git tag
```

To checkout the desired branch:

```
git checkout v*BRANCH NUMBER HERE* -b version*BRANCH NUMBER HERE*
```

as in:

```
git checkout v3.7 -b version3.7
```

This creates a local branch called **version3.7** from the release branch **v3.7**

5. Update the system pkg list

```
UBUNTU:
sudo apt-get update


RHEL:
sudo yum update
```

If asked – accept any keys by typing **Y <ENTER>**

6. Run the install-system-dependencies.sh script to install the repositories and packages that ESP depends on.

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

RHEL:
  sudo ./install-system-dependencies-rhel.sh
```

If asked – accept any keys by typing **Y <ENTER>**

Note: if you receive a pip error about "server certificate verification failed." You can attempt to run pip with the trusted-host flag as shown below, or have the hosts lists in the allowed list for the local firewall or repository.

```
UBUNTU:
    pip install --trusted-host pypi.python.org --trusted-host python.org  --trusted-host
    pypi.org  --upgrade -v -r requirements.pypi.txt

RHEL7:
    pip install --trusted-host pypi.python.org --trusted-host python.org  --trusted-host
    pypi.org  --upgrade -v -r requirements-rhel7.pypi.txt

RHEL8:
    pip install --trusted-host pypi.python.org --trusted-host python.org  --trusted-host
    pypi.org  --upgrade -v -r requirements-rhel8.pypi.txt
```

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

6a. RHEL 7.9 Only - Enable SCL Python

1. On older Red Hat systems we need to enable the use of Python installed from the Software Collection and to permanently add Python3 to the ESP user's path. To do this execute the following commands:

```
sudo su esp   (if not already the esp user)
vi ~/.bashrc     (adapt commands for editor)
```

Arrow down and over to the end of the line "# User specific aliases and functions"
and Type "o" (lower case letter o) to start a new line and enter insert mode,
then add the following 2 lines:

```
#Add RHSCL Python 3 to my login environment
source scl_source enable rh-python38
```

To exit and save:
hit the ESC key
Type :wq (colon wq)   and hit Enter

To exit without saving (so you can start over if need be):
hit ESC key
Type :q! (colon q exclamation point) and Enter)

Now CLOSE all terminal windows and log back in.
sudo and/or login as the esp user again before proceeding

6 Verify Python3 is installed

Perform the following as the esp user

```
sudo su - esp
```

type:

```
which python3.8
```

REQUIRED OUTPUT:

UBUNTU:
```
/usr/bin/python3.8
```

RHEL:
```
/usr/bin/python3.8                          # RHEL 8.x
   OR
/opt/rh/rh-python38/root/usr/bin/python3    # RHEL 7.x
```

To Check with specific version type:
```
python3.8 -V
```

SAMPLE OUTPUT:
python3.8.16

**Note: If these do not line up with expected responses, please contact Commonwealth Informatics for
       assistance @ esp_support@commoninf.com**

7   Run the ESP install.sh script.

UBUNTU:
```
./install.sh
```

RHEL 8:
```
./install-rhel8.sh
```

RHEL 7.9:
```
./install-rhel7.sh
```

If errors are encountered – you may need to manually update pip to version 20.1 or above..

As the esp user type:

```
source /srv/esp/prod/bin/activate
python -m pip install pip==20.1
pip list
deactivate
```

```
Then try to rerun the installation script:
```

UBUNTU:
```
./install.sh
```

RHEL:
```
./install-rhel7.sh  or  ./install-rhel8.sh
```

If errors are encountered, manually edit the install(-rhel7).sh file and make modifications to the versions as required or contact CII for assistance.

8 Verify PostgreSQL is installed

1. To Verify that postgres is installed - type:

```
UBUNTU:
 sudo apt list –installed | grep postgresql
```

```
RHEL:
 sudo yum list installed | grep postgresql
```

A list of pkgs should be displayed. If postgresql is not installed, contact CII for support.

9 Initialize and create the ESP Database and ESP Database User

1. Initialize and start the Postgres database.

NOTE: This step may not be necessary if the installation automatically starts the database,
This is usually the case for Ubuntu Servers.

First, you can to see if the Database is running by typing this:

```
sudo -u postgres psql -c '\l'
```

(Sample output is posted below)

RHEL – you need to initialize Postgresql with initdb – run as root or sudo:

```
sudo /usr/pgsql-13/bin/postgresql-13-setup initdb
```

2. Enable and Start the postgresql Server

Type:

```
sudo systemctl enable postgresql-13
sudo systemctl start postgresql-13
```

Verify the postgres is running by logging in as the postgres user and running \l;

```
sudo -u postgres psql -c '\l'
```

(See sample output below in step 4)

3. Create the "esp" role in the Database

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

First enter in the current user's password for sudo if required
Then in response to the "Enter password for new role" prompts enter the password for the new esp user.

The password for the esp db user will also need to be entered in the etc/secrets.ini file as specified later in these instructions.

4. Create the "esp" database

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

NOTE: the -O option sets the esp: database owner (to esp) user to own the new database.

If received, Ignore the error "could not to change to directory "/srv/esp/prod": permission denied"

Optional: Verify the database was created by connecting to it via psql:

```
sudo -u postgres psql -d esp -c '\l'
```

that is a backslash and a lowercase "L" in the single quotes

you should see output similar to below:

```
 Name      |  Owner   | Encoding |   Collate   |   Ctype     |  Access privileges
-----------+----------+----------+-------------+-------------+----------------------
 esp       | esp      | UTF8     | en_US.UTF-8 | en_US.UTF-8 |
 postgres  | postgres | UTF8     | en_US.UTF-8 | en_US.UTF-8 |
 template0 | postgres | UTF8     | en_US.UTF-8 | en_US.UTF-8 | =c/postgres        +
 template1 | postgres | UTF8     | en_US.UTF-8 | en_US.UTF-8 | =c/postgres        +
```

5. Update the postgres pg_hba.conf to allow esp user to connect locally and restart the DB.

Configure pg_hba.conf for local connections.
  NOTE: this file may be in a different location – use "sudo find / -name pg_hba.conf" to locate it
  The below location are samples for Postgresql version 13:

```
sudo su - postgres
UBUNTU:  cd /etc/postgresql/13/main/
REDHAT:  cd /var/lib/pgsql/13/data/
vi pg_hba.conf
```

Replace the contents at the bottom of the file with the following:

```
# "local" is for Unix domain socket connections only
#local  all        all                    trust
local   all        postgres               ident
local   esp        esp                    ident
local   all        all              md5

# IPv4 local connections:
#host   all        all      127.0.0.1/32      trust
host    all        all      127.0.0.1/32      md5
host    all        all      0.0.0.0/0         md5

# IPv6 local connections:
#host   all        all      ::1/128          trust
host    all        all      ::1/128          md5
```

Exit as postgres, and become the esp user again

6. Restart Postgres

```
sudo systemctl restart postgresql-13
```

10 Create the Directories and Files required by ESP

As the ESP user, copy and paste the following to create the directories expected by ESP:

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

Note: The use of the folder name "epic" for inbound data is vestigial.
These folders are used for text data from any source.

1. Create the ESP Log File

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

```
sudo chown esp:esp /var/log/esp.log
sudo chmod 666 /var/log/esp.log
```

11 Configure the ESP Application and create the UI user

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

```
cd /srv/esp/prod
./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 you created earlier.

For the **secret_key**, enter a random string of at least 32 characters.

NOTE: Use only Alpha-Numeric characters (a-z A-Z 0-9)
Do not use any special chars like $ or # or quotes

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

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

You should confirm, modify, or enter details for the attributes listed here.

In the **[General]** section, edit the following settings:
```
site_name = Your Site Name
data_folder = /srv/esp/data
admins = your_email@your_host.com, another_email@your_host.com
managers = your_email@your_host.com, another_email@your_host.com
icd10_support = True
```

In the **[Database]** section, edit the following settings as needed depending on the Database you are using:
db_name, username, engine, port, etc.. Standard configs details are below:

```
db_name = esp
username = esp
host = localhost
engine = django.db.backends.postgresql_psycopg2
port = 5432
transaction_row_limit = 100
```

In the **[Web**] section, edit the following settings:
```
allowed_hosts = localhost, <**Add the ESP server hostname here**>
```
** this should match the VirtualName you configure in Apache/httpd. See Section 10 **

In the **[ETL**] section, edit the following settings:
```
load_report_dir = /srv/esp/data/load_reports/
archive = False
```

In the **[Email]** section, edit the following setttings:
```
host = <enter your corporate mail server here>
server_email = esp-no-reply@your_host.com
default_from_email = esp-no-reply@your_host.com
```

In the **[Logging]** section, edit the following settings:
```
log_level_console = info
log_level_file = info
log_level_syslog = info
```

In the **[Site]** section, edit the following settings:
```
site_header = ESP-YOURSITE
case_report_site_name = YOURSITE
site_clia = <enter your primary site CLIA here>
site_last_name = Jones
site_first_name = Bob
site_address1 = 133 AnyStreet Avenue
site_city = Boston
site_state = MA
site_zip = 02215
site_country = USA
site_email = bjones@yourhost.org
site_area_code = 617
site_tel_numeric = 1234567
site_app_name = ESP
site_sending_facility = YOURSITE
```

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 migrate
```

NOTE: If you receive an error that ends with

```
"TypeError: 'class Meta' got invalid attribute(s): constraints",
```

Edit the following file with the command:
```
vi /srv/esp/prod/lib/python3.*/site-
packages/rest_framework/authtoken/migrations/0003_tokenproxy.py
```

Comment out the following line (or delete it) :

```
'constraints': [],
As in:
#'constraints': [],
```

Run the migrate command again:

```
/srv/esp/prod/bin/esp migrate
```

NOTE: If you still receive migration errors, reach out to CII at esp_support@commoninf.com for assistance.

6. Install the Disease Detection Plugins you require:

    ```
    ./setupPlugins.sh
    ```

7. Create an ESP UI superuser:

    ```
    ./bin/esp createsuperuser
    Username (Leave blank to use 'esp'): esp
    E-mail address: <your email address>
    Password: <password for the esp superuser>   enter secure password
    Password (again): <password for the esp superuser>
    ```

    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.

    Exit the server using Ctrl-C

12 Configure the Web server for http access

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.

The following pkgs should already be installed:

```
UBUNTU:
apache2 libapache2-mod-wsgi-py3

RHEL:
httpd (httpd24-httpd)  rh-python38-mod_wsgi
```

1. Startup and Enable automatic startup at system reboot

    ```
    UBUNTU:
    sudo systemctl start apache2
    sudo systemctl enable apache2

    RHEL:
    sudo systemctl start httpd
    sudo systemctl enable httpd
    ```

WSGI setup

1a. Make a copy of django.wsgi.sample located in the $ESP_HOME/share folder and copy it to the $ESP_HOME/etc directory and then edit it as described below:

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

1b. Update the contents of the file to set the proper PATH if needed.

   NOE: Skip this step If you are using the standard /srv/esp/prod,
              otherwise, replace the directory path **/srv/esp/prod/** to match your environment.

```
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/')

exec(compile(open('/srv/esp/prod/bin/activate_this.py').read(), '/srv/esp/prod/bin/activate_this.py', 'exec'))

from django.core.wsgi import get_wsgi_application
application = get_wsgi_application()
```

2.   Create the Configuration files for the Web Server

   2a. Create the configuration file. The name should match the name of your server
   Update the highlighted parts referenced below to match your server

   Copy the sample file from /srv/esp/prod/share to the appropriate directory

   UBUNTU:

```
sudo cp /srv/esp/prod/share/apache.conf.sample /etc/apache2/sites-
available/servername.conf
```

   RHEL:

```
sudo cp /srv/esp/prod/share/apache.conf.sample
/etc/httpd/conf.d/vhost_servername.conf
```

   2b. Edit the Virtual Host conf file and modify the **VirtualHost, ServerName and ServerAdmin values**.
        Also update the location of the **WSGIScriptAlias** path for the ESP install if different then /srv/esp/prod

   UBUNTU:

```
sudo vi /etc/apache2/sites-available/servername.conf
```

   RHEL:

```
sudo vi /etc/httpd/conf.d/vhost_servername.conf
```

   See sample excerpt from /etc/apache2/sites-available/esp-server.conf below:

….

```
WSGIRestrictStdout Off
WSGIDaemonProcess esp
WSGIScriptAlias  /  /srv/esp/prod/etc/django.wsgi

# Change the VirtualHost and the Servername from "esp-server" to reflect your actual hostname
# this should match this filename - esp-server.conf or esp-server.org.conf
<VirtualHost esp-server:80>
ServerName esp-server
ServerAdmin  esp-support@your-org.com

Alias /static/ /srv/esp/prod/ESP/media/static/
Alias /media /srv/esp/prod/ESP/media
….
```

2c. Edit the Server's apache or httpd .conf file and Add the required directory configurations

UBUNTU:

```
sudo vi /etc/apache2/apache2.conf
```

RHEL:

```
sudo vi /etc/httpd/conf/httpd.conf
```

Beneath this default section:

```
<Directory /var/www/>

  Options Indexes FollowSymLinks
  AllowOverride None
  Require all granted
</Directory>
```

Add the following:

```
<Directory /srv/esp/prod/etc/>
  Options Indexes FollowSymLinks
  AllowOverride All
  Require all granted
</Directory>

<Directory /srv/esp/prod/ESP/media/>
  Options Indexes FollowSymLinks
  AllowOverride All
  Require all granted
</Directory>
```

2d. Enable the required Apache modules. (Ubuntu Only)

UBUNTU:
```
sudo a2enmod wsgi
sudo a2enmod rewrite
sudo a2ensite servername.conf
```

Note: The site name (server-name.conf) has to match the file name created in step 2a.

3. Restart the Web Server/Verify it is setup to startup

   UBUNTU:

   ```
   sudo systemctl restart apache2
   ```

   RHEL:

   ```
   sudo systemctl restart httpd
   ```

```
Login via a web browser – check the logs if you get any errors.
```

13 Configure the Web server for https access

```
TBD
```

14 Configure Database Backups

1. As the esp user, Create the backup directory (modify steps as needed for a different location)

   ```
   mkdir /srv/esp/backup
   ```

2. Modify the ownership on the directory

   ```
   sudo chown postgres:postgres /srv/esp/backup/
   ```

3. Modify the permissions on the directory

   ```
   sudo chmod 755 /srv/esp/backup/
   ```

Create the **/srv/esp/scripts/db_backup.sh** file from the share directory

```
cp /srv/esp/prod/share/db_backup.sh /srv/esp/scripts/db_backup.sh
```

Set the permissions on the script:

```
chmod 755 /srv/esp/scripts/db_backup.sh
```

4. Create the logrotate configuration file

```
 sudo -u postgres cp /srv/esp/prod/share/postgres-sql.logrotate
/srv/esp/backup/esp-sql.logrotate
```

Edit the logrotate file and modify it as described below:

```
sudo vi /srv/esp/backup/esp-sql.logrotate
```

```
        A - verify/update the location of the pg_dump command and backup directory
        B - adjust the number of files to be saved if desired.
```

5. Create the empty dump files needed for logrotate as the user that will run the backups – by default this is the local postgres user.

6. Create the crontab file entry for the db_backup.sh script as the DB process owner

    As the postgres user, add the following entry to the postgres user crontab to run the database backups. Choose a time that does not conflict with daily ESP processing. The example below is set to run at 11pm each day. The script referenced will be created in the next step.

```
    sudo -u postgres crontab -e

    #m     h      dom      mon      dow      command
     0     23      *        *        *       /srv/esp/scripts/db_backup.sh

    (ESC and :wq) to save the file (in vi)
```

```
   To verify the crontab type:

       sudo -u postgres crontab -l
```

```
Note:   After the initial setup of the backup/logrotate job, check back on a
regular basis to make sure that the process is running properly.  The backup
directory should start to get populated with -yyyymmdd dated files as shown below
```

```
-rw-r--r-- 1 postgres postgres         0 Mar 24 23:02 esp.daily.dump
-rw-r--r-- 1 postgres postgres 711858897 Mar 18 23:02 esp.daily.dump-20210318
-rw-r--r-- 1 postgres postgres 712799083 Mar 19 23:02 esp.daily.dump-20210319
-rw-r--r-- 1 postgres postgres 713975672 Mar 20 23:02 esp.daily.dump-20210320
-rw-r--r-- 1 postgres postgres 714477947 Mar 21 23:02 esp.daily.dump-20210321
-rw-r--r-- 1 postgres postgres 714858619 Mar 22 23:02 esp.daily.dump-20210322
-rw-r--r-- 1 postgres postgres 715945353 Mar 23 23:02 esp.daily.dump-20210323
-rw-r--r-- 1 postgres postgres 717050394 Mar 24 23:02 esp.daily.dump-20210324
-rw-r--r-- 1 postgres postgres         0 Mar  1 23:04 esp.monthly.dump
-rw-r--r-- 1 postgres postgres 641133808 Dec 16 12:28 esp.monthly.dump-20201216
-rw-r--r-- 1 postgres postgres 653218502 Jan  1 23:03 esp.monthly.dump-20210101
-rw-r--r-- 1 postgres postgres 675364055 Feb  1 23:04 esp.monthly.dump-20210201
-rw-r--r-- 1 postgres postgres 696714943 Mar  1 23:04 esp.monthly.dump-20210301
-rw-r--r-- 1 postgres postgres         0 Mar 21 23:04 esp.weekly.dump
-rw-r--r-- 1 postgres postgres 696400523 Feb 28 23:04 esp.weekly.dump-20210228
-rw-r--r-- 1 postgres postgres 702511352 Mar  7 23:04 esp.weekly.dump-20210307
-rw-r--r-- 1 postgres postgres 707950650 Mar 14 23:04 esp.weekly.dump-20210314
-rw-r--r-- 1 postgres postgres 714477947 Mar 21 23:04 esp.weekly.dump-20210321
-bash-4.2$
```

15 Configure ESP log file rotation

As the root user - copy the esp.logrotate configuration file from the /srv/esp/prod/share directory to /etc/logrotate.d/

```
sudo cp /srv/esp/esp_tools/Install_Scripts/esp.logrotate /etc/logrotate.d/
```

Note: Modify this file as necessary based on the location of the esp log file
       The default is /var/log/esp.log

To test this - type the following commands to verify the log has been saved.

```
sudo logrotate -f /etc/logrotate.conf
ls -al /var/log/esp.log*
```

16 Configure Log Monitoring

This process will monitor a few key files for specific strings that are known to represent a possible error and send an email to the specified recipients with the line from the log file that triggered the error.

When an error message is received, the recipient should access the full log file for more details and information as only a snippet of the error message may be received.

Mail Prerequisite for monitoring

Please note that your ESP server must be configured and capable of sending email. Prior to following the setup instructions, you should verify that the server is configured with a mail program. If it isn't, you will need to do this or reach out to your IT staff for assistance.

One way to test this is by issuing the following from the command line (replace the email address with a valid email address):

```
mail -s "This is a test mail" myname@mycompany.com
this is a test
```

When you have finished composing the e-mail enter <CTL>-D on a new line to send it.

Verify that the email is received.

--

The process should be initiated from the ESP application server where the ESP log files reside.

Commands should be executed as the root or esp user (via sudo).

Instructions are provided for both Ubuntu and RHEL.

1. Install swatch (or swatch-dog)

```
UBUNTU:
   sudo apt-get install swatch    (or swatchdog if swatch is not found)


RHEL:
   sudo yum install swatch    (or swatchdog if swatch is not found)
```

If apt-get or yum fails to find or install either pkg, manually download and install swatchdog with the steps below:

```
cd /srv/esp/esp_tools
git clone https://github.com/ToddAtkins/swatchdog.git
cd swatchdog/
perl Makefile.PL
make
sudo make install
sudo make realclean
```

Verify command is available

`which swatchdog`   (or `which swatch` – this will return the path - /usr/local/bin/swatchdog)

2. Install xtail

UBUNTU:
```
sudo apt-get install xtail
```

RHEL:
```
sudo yum install xtail
```

If the apt-get or yum fails to find or install the pkg, manually download and install xtail with the steps below:

```
cd /srv/esp/esp_tools
sudo wget http://www.unicom.com/files/xtail-2.1.tar.gz
gunzip xtail-2.1.tar.gz
tar -xvf xtail-2.1.tar
cd xtail-2.1
sudo sh
configure
make
make install
```
(you can ignore any errors about creating the man file)
```
exit (to exit the sh)
```

Verify command is available

`which xtail –` (this will return the path – such as /usr/local/bin/xtail)

3. Grab latest versions of conf files and scripts from ESP tools repository

As the esp user type:

```
cd /srv/esp/esp_tools
git pull
cd Install_Scripts/log_monitoring
```

4. Implement the conf files

For systems that send email with the "mail" command copy these files:
```
cp swatch_esp.conf /srv/esp/scripts
cp swatch_cron.conf /srv/esp/scripts
```

For systems that send email with the "sendmail" command copy these files:
```
cp swatch_esp.conf.sendmail /srv/esp/scripts/swatch_esp.conf
cp swatch_cron.conf.sendmail /srv/esp/scripts/swatch_cron.conf
```

5. Modify the conf files to direct email to appropriate recipients.

```
cd /srv/esp/scripts
```

5a. Modify the email recipients in the swatch_esp conf file

```
vi swatch_esp.conf
```

For systems that use the "mail" command, update the following addresses (**separated by colo**n):

```
mail addresses=testemail1\@test.org:testemail2\@test.org,
```

For systems using the "sendmail" command, update the following addresses **(separate by semicolon**):

```
sendmail testemail1@test.org;testemail2@test.org
```

5b.   REPEAT – modify the email recipients in the swatch_cron.conf file

```
vi swatch_cron.conf
```

6. Implement the startup scripts

```
cd /srv/esp/esp_tools/Install_Scripts/log_monitoring
cp swatch_esp /srv/esp/scripts
cp swatch_cron /srv/esp/scripts
```

7. Update the locations in the ESP Log monitor startup script.

You will need to verify and possibly update the script file for the following parameters listed below:

1. The location of the esp.log files   (if other then /var/log/esp.log)
2. The path to xtail   (if other then /usr/bin/xtail)
      a.   Use the command: **which xtail**   to confirm the path
3. The Name and path of the swatch process. Swatch had to change its name so depending on the OS you are running the package may be installed as swatch or swatchdog.
      a.   Use the commands **which swatch**   and   **which swatchdog**   to determine which process name and which path to use. (the default is /usr/bin/swatchdog)

```
cd /srv/esp/scripts
vi swatch_esp
```

as described above, confirm the highlighted parameters within the file:

```
case "$1" in 'start')
        /usr/bin/swatchdog --daemon --config-file=/etc/swatch_esp.conf --tail-
file='/var/log/esp.log' --pid-file=/tmp/swatch_esp.pid --tail-program-name
/usr/bin/xtail
```

8. Update the locations in the ESP Cron Log monitor startup script.

You will need to verify and possibly update the script file for the following parameters listed below:

1. The directory of the ESP cron logs (if other then /srv/esp/logs/)
2. The path to xtail   (if other then /usr/bin/xtail)
   a. Use the command: **which xtail** to confirm the path
3. Name and path of the swatch process. Swatch had to change its name so depending on the OS you are running the package may be installed as swatch or swatchdog.
   a. Use the commands `which swatch` and `which swatchdog` to determine which process name and the path, only one should return a path..

```
cd /srv/esp/scripts
vi swatch_cron
```

as described above, confirm the highlighted parameters within the file:

```
case "$1" in 'start')
        sudo /usr/bin/swatchdog --daemon --config-file=/etc/swatch_cron.conf --
tail-file='/srv/esp/logs/' --pid-file=/var/run/swatch_cron.pid --tail-program-
name /usr/bin/xtail
```

9. Set the Permissions on the script files

```
sudo chmod 755 /srv/esp/scripts/swatch_cron
sudo chmod 755 /srv/esp/scripts/swatch_esp
```

10. Implement the service files

```
cd /srv/esp/esp_tools/Install_Scripts/log_monitoring/
sudo cp swatch_esp.service /etc/systemd/system/
sudo cp swatch_cron.service /etc/systemd/system/
```

11. Reload the service daemon and Start the monitors

```
sudo systemctl daemon-reload
sudo systemctl start swatch_esp
sudo systemctl start swatch_cron
```

12. Verify the services are running

```
ps -elf | grep swatch | grep -v grep
```

you should see output showing two processes running similar to below:

1 S root   5258   1 0 80 0 - 14094 -   18:35 ?   00:00:00 /usr/bin/swatchdog --daemon --config-
file=/srv/esp/scripts/**swatch_esp.conf** --tail-file=/var/log/esp.log --pid-file=/tmp/swatch_esp.pid --tail-program-name=/usr/bin/xtail
1 S root   5259   1 0 80 0 - 14088 -   18:35 ?   00:00:00 /usr/bin/swatchdog --daemon --config-
file=/srv/esp/scripts/**swatch_cron.conf** --tail-file=/srv/esp/logs/ --pid-file=/tmp/swatch_cron.pid --tail-program-name=/usr/bin/xtail

13. Add the new Services to the system startup

```
sudo systemctl enable swatch_esp
sudo systemctl enable swatch_cron
```

14. Test it out.

(NOTE: update file names and locations as appropriate)
As the esp user copy and paste the following two commands:

```
echo "CRITICAL This is just a test. If this was a real error you would see it
here" >> /var/log/esp.log

echo "ERROR This is just a test. If this was a real error you would see it here"
>> /srv/esp/logs/daily_log_cron.test
```

You should receive an email with the errors above.

17 Setting up the data feed - Importing data to ESP

Each site requires an extract that provides data to ESP in the standard text file format. Please see the document ESP_Filespec for details. For Epic sites, Commonwealth 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 configurations.

There are two types of files accepted for input to ESP, "epic" or flat files and CDA files.

This Chapter covers the load_epic method, Chapter 16 covers the CDA method.

Either way, there are two data feed steps, historic and daily.

First, we need to 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.

Then we can set up the daily data extract and load. This is for ongoing data extraction and loading of data.

The ESP command "load_epic" is used to load the flat ^ separated data file.
   NOTE: 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.

Sample commands for loading files:

/srv/esp/prod/bin/esp load_epic --file /srv/esp/data/epic/incoming/epicmem.esp.03112019
/srv/esp/prod/bin/esp load_epic --input /srv/esp/data/epic/incoming/ --reload

18 Using ESP Basics – loading data & case reporting

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

NOTE: If using CDA files, please refer to Chapter 16

A typical command to load data might look like this:
(NOTE: file names need to follow this format epic*xyz*.esp.yyyymmdd – where xyz is the file type – see the latest ESP FileSpec for more information

```
$ $ESP_HOME/bin/esp load_epic  --file
/srv/esp/data/epic/incoming/epicmem.esp.20190301
```

The use of the name 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

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 the table conf_labtestmap.

To perform lab mapping, run:

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

The concordance command populates a table emr_labtestconcordnace, 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 generates hl7 2.3.1 version lab report message, the lab_report command generates 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. Please contact CII for more information. Example case_report and case_requeue commands are below.

```
$ESP_HOME/bin/esp case_report --mdph --transmit --status 'Q'

$ESP_HOME/bin/esp case_requeue --status=RS
```

19 Loading CDA Data

The CDA REST web service will accept CDAs via a REST endpoint and place them in /srv/esp/data/cda/incoming.

To consume these CDAs and import the data into the ESP database, execute the following command as the esp user:

```
$ESP_HOME/bin/esp load_cda --proc=16
```

Copyright © 2021 Commonwealth Informatics Inc.

The --proc (abbreviation for processing threads) is optional and indicates the number of processing threads to use. If not specified the default value of 1 will be used.

CDAs will be read from the /srv/esp/data/incoming folder, consumed, and then moved to the /srv/esp/data/archive folder.

A daily cron job using logrotate will compress (zip) these .xml files and move them to the /processed sub-folder. The cron job also deletes the the old .gz files in /processed after 30 days.

The folder permissions must be set to allow the rest web service (which runs as user www-data under Apache2) and logrotate script (which runs as user esp). To read and write these files. Folder permissions must be set as follows:

```
$ cd ~/data/cda
$ sudo chown www-data:esp archive
```

The following commands will use the sample scripts to setup the scripts that should be run nightly as cron jobs.

```
$ cp ~/prod/ESP/cda/scripts/cda.logrotate.sample ~/scripts/cda.logrotate
$ sudo chown root:root ~/scripts/cda.logrotate
$ cp ~/prod/ESP/cda/scripts/cda_ingest.sh.sample ~/scripts/cda_ingest.sh
$ chmod +x ~/scripts/cda_ingest.sh
$ cp ~/prod/ESP/cda/scripts/cda_compress_clean.sh.sample
~/scripts/cda_compress_clean.sh
$ chmod +x ~/scripts/cda_compress_clean.sh
```

These scripts are very simple and typically they will not need to be edited for an installation.

Cronjobs to run these scripts can be created by the following command:

```
sudo crontab -e
```

A sample of these cronjobs is listed below:

```
0  1  *  *  * /srv/esp/scripts/cda_ingest.sh
0  2  *  *  * /srv/esp/scripts/cda_compress_clean.sh
```

At 0100 cda_ingest.sh runs. This script will ingest all of the .xml CDAs in ~/data/cda/incoming and move them to ~/data/cda/archive

At 0200 cda_compress_clean.sh runs. This script will use logrotate to move all of the .xml files from ~/data/cda/archive and compress them to ~/data/cda/archive/compressed with an extension of .gz

20 Configure ESP Daily Status Reports

ESP will send a daily email showing information about the cases created and transmitted. Additionally, it will show unmapped lab tests and other information related to ESP.

1. Configure the application.ini file

```
sudo su esp

sudo vi /srv/esp/prod/etc/application.ini
```

In the **[General]** section at the top of the file, configure the managers to match your desired recipients.

```
  # Managers are emailed a copy of the daily status report, if it is enabled in
  the Batch section
  managers = recip1@youremail.com, recip2@youremail.com,
```

  Note the ending comma

In the **[Email]** section, configure your parameters. Set the host, port, server_email, and default_from_email to match your configuration.

```
[Email]
host = your_mail_server_or_ip_address
port = your_port_number (i.e. 25)
username = ""
use_tls = False
server_email = esp-no-reply@youremail.com
default_from_email = youremail@youremail.com
subject_prefix = "[ESP] "
```

**Note: If your server requires authentication, you will want to enter the username here and the password in the secrets.ini file

In the **[Batch]** section, set:

```
mail_status_report = True
```

2. To test if it is working, run the following command from the command line. It just sends the email and doesn't change any data, etc.

```
/srv/esp/prod/bin/esp status_report --send-mail
```

21 Configure crontab to run Daily jobs

A sample daily_batch.sh script can be found within the core system repository. If using the default paths, this file will be found in the `/srv/esp/prod/share` folder. This is the script that is utilized to perform the daily processes for ESP (data load, event creation, case creation, etc).

This is just a sample script and should be modified/customized for your environment.

To enable daily data loading and processing, please the customized daily_batch.sh file in the `/srv/esp/scripts` folder.

As the ESP user, add a cron entry in your system:

```
sudo su esp
crontab – e
```

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

```
# ESP Production Daily Run
#
30      22      *       *       *       /srv/esp/scripts/daily_batch.sh
```

Save and quit

To verify the new crontab entry type:

```
crontab –l
```

22 Setting Up Basic Disease Detection

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

```
$ /srv/esp/prod/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 codes 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. From the menu bar, browse to *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. This step requires site support from staff familiar with local lab naming and the corresponding lab test.

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. Case validation requires site support from staff with the ability to do case review in the EHR.

23 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. This method will automatically delete all of the related records in the db.

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

1. From the ESP UI Select Administration -> Site Administration from the menu

2. Click on "Provenances" from the "Emr" section

Vaccine manufacturer maps		✚ Add	✏ Change
Emr			
Allergys		✚ Add	✏ Change
Encounter type maps		✚ Add	✏ Change
Encounters		✚ Add	✏ Change
Hospital_ problems		✚ Add	✏ Change
Immunizations		✚ Add	✏ Change
Lab Orders		✚ Add	✏ Change
Lab Specimen Observations		✚ Add	✏ Change
Lab Specimens		✚ Add	✏ Change
Lab Test Results		✚ Add	✏ Change
Lab infos		✚ Add	✏ Change
Order_id infos		✚ Add	✏ Change
Patient_ addrs		✚ Add	✏ Change
Patient_ extra datas		✚ Add	✏ Change
Patient_ guardians		✚ Add	✏ Change
Patients		✚ Add	✏ Change
Pregnancys		✚ Add	✏ Change
Prescriptions		✚ Add	✏ Change
Problems		✚ Add	✏ Change
Provenances		✚ Add	✏ Change

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

☑	7	Feb. 24, 2014, 7:22 p.m.	epicvis.esp.02282012
☑	6	Feb. 24, 2014, 6:12 p.m.	epicres.esp.02242012
☑	5	Feb. 24, 2014, 6:12 p.m.	epicord.esp.02282012
☑	4	Feb. 24, 2014, 6:12 p.m.	epicmem.esp.02282012
☑	3	Feb. 24, 2014, 6:12 p.m.	epicpro.esp.02282012
☐	2	Jan. 1, 1900, midnight	CLEANUP
☐	1	Jan. 1, 1900, midnight	SYSTEM

89 provenances

4.   From the same screen, select "Delete selected provenances" from the "Action" drop down. Next, click "Go" via the button to the right of the dropdown.
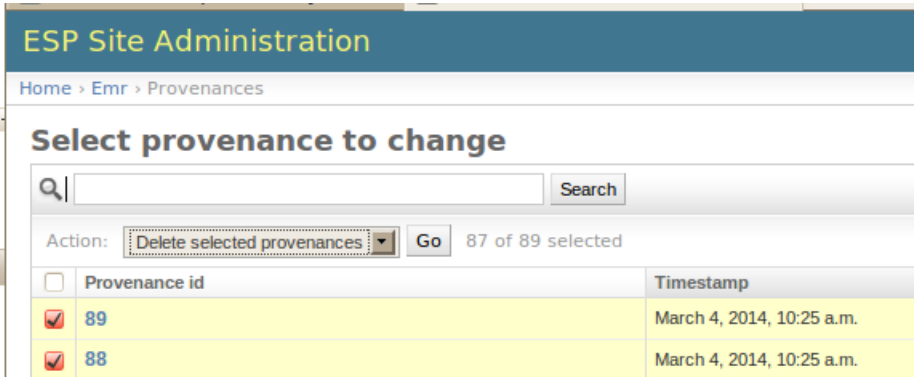
5. This will calculate all the related records and elements in the database related to those loaded files and will display a summary of all the records that will be deleted.

A confirmation window with the prompt "Are you sure you want to delete the selected provenances? All of the following objects and their related items will be deleted." will appear.

To continue, scroll down to the bottom and click on the "Yes, I am sure" button: