ESP DOCS

How to Install and Configure ESP on ~~Ubuntu~~RedHat (7.6-v3)

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

Contents

1   Installation Overview

This document describes the procedure for configuring ESP on a RedHat n UbuntuEnterprise Linux Server.

The versions of software used to prepare these instructions were:

Linux: RedHat Enterprise Linux 7.6 Ubuntu 18.04 Server Edition (64-bit)
PostgreSQL: 9.6
Apache2: 2.4
ESP: 3.4.x running in a virtual environment using
     Python 2.7
     Django 1.8

The Linux Ubuntu server should have ssh, git, and access to a local SMTP service.

These Installation steps are covered in Chapters 1 - 11:

- Create the esp user and system prerequisites
- Download the esp project and run the install system dependencies shell script
- Create the esp database and esp database user
- Create directories and files expected by ESP
- Configure the application settings and create the UI user
- Configure the Apache web server
- Database Backups
- Log rotation
- Log monitoring

Additional topics for configuration once the system is running are covered starting at Chapter 10. These include the following:

- Configure Daily Cron job
- Configure the data import to ESP
- ESP Basics
- Daily Status Reports
- Basic disease detection setup
- Deleting test data

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 privileges.

2   Prerequisites

This installation Guide is for an Ubuntu Linux 18RedHat 7.6. .04 server   SSH for remote login should be available. The bash shell should be the default for the esp user.

The Installation will require either a local SMTP service, or an email account on an accessible SMTP server.

Ensure firewall rules permit the following inbound connections on the application server:
Port: 22 - needed for SSH by sysadmins for installation and maintenance.

Port: 80 - needed for HTTP access to the browser based ESP user interface

NOTE: 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 your Organization. The Server and the Network where it is located should be locked down using whitelisting and firewalls to allow access only from specific Internal hosts or networks and/or VPN connections to prevent unauthorized access.

3   Create the ESP User and setup prerequisites

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

    Standard ESP installation will be to /srv/esp/prod. Use /srv/esp as the esp home directory. Enter a password for the ESP user and account details when prompted.

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

2. Add the user to the 'sudo' group

    ```
    sudo usermod -aG sudo wheel esp
    ```
2.

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

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

4   Download the ESP project and run the esp install

1. Switch to the esp user
    1.
    ```
    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
    ```
    2. d

3. Checkout the desired branch and give it a "local" branch name. 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:

```
cd /srv/esp/prod
```

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

as in:
```
git checkout v3.4.11.21 -b version3.4.11.211
```

4. Run the install-system-Setup ESP Ddependencies.sh script to —by iInstalling the packages that ESP depends on

```
—sudo ./install-system-dependencies-rhel7.sh
```

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

5. Run the ESP install.sh script.

```
./install-rhel7.sh
```

If errors are encountered, manually edit the "requirements-rhel7.pypi.txt " file and make modifications as needed.

5 Initialize and create Create the ESP Database and ESP Database User

1. Enter the following commands to initialize and start the PostgreSQL Database:

```
sudo /usr/pgsql-9.6/bin/postgresql96-setup initdb
sudo systemctl enable postgresql-9.6
sudo systemctl start postgresql-9.6
```

1. 2. 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
```

first enter your (or the esp user's) password (for sudo (if required))
then in response to the "Enter password for new role" and "Enter it again" prompts,
—enter the password for the new esp user. **(later you will enter thisthis will be entered in the secrets.ini file)**

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

2. Create the "esp" database by entering the following at the command prompt:

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

The options passed into the createdb command control the following:
NOTE: the -O option sets the database owner (to esp) to own the new database
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:

```
psql -d esp -c '\l'      # that is a slash 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         +
           |          |          |            |            | postgres=CTc/postgres
 template1 | postgres | UTF8     | en_US.UTF-8 | en_US.UTF-8 | =c/postgres         +
           |          |          |            |            | postgres=CTc/postgres
(4 rows)
```

6 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
mkdir -p /srv/esp/scripts
mkdir -p /srv/esp/logs
```

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

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

```
As the ESP user, create the directories and files expected by ESP by entering:Create the ESP Log Filesudo su 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/fake
mkdir -p /srv/esp/scripts
mkdir -p /srv/esp/logs
```

Use of the folder name "epic" for inbound data is vestigial.
These folders are used for text data from any source system.

2. 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
```

7 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.

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

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

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

Confirm, modify, or enter details for the attributes listed below in the application.ini file.

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

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. See Section 8 **
```

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 = EMR
    ___  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
```

Modify other settings as necessary – save and exit the file.

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
```

If you get an error "ValueError: Cannot query "ConditionConfig object": Must be "ConditionConfig" instance.", please re-run the above command. Login errors are often caused by a locked down postgres configuration – please check the files following files and see Appendix C for a sample pg_hba.conf file.

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>
```

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.

Exit the server using Ctrl-C

8 Configure the ~~Apache~~ httpd web server

Set up the Apache web server following the steps specified below.

1. As the esp user - make a copy of django.wsgi.sample located in the srv/esp/prod/share folder and copy it to the /srv/esp/prod/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
```

NOTE: Update the contents of the file so directory paths match if not using the default location.

```
sys.path.insert(0, '/srv/esp/prod/')
execfile('/srv/esp/prod/bin/activate_this.py')
```

2. As root - Create the Apache configuration file from the sample in the share directory – it is also listed in Appendix C - The name should match the name of your server.

For example:

```
sudo cp /srv/esp/prod/share/esp-server.domain.com
/etc/httpd/conf.d/vhost_servername.conf
```

Modify the first below settings for your environment.

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

```
ServerName servername
ServerAdmin esp_support@yourdomain.com
```

3. Add required configurations to httpd.conf

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

Beneath this existing section:

```
<Directory "/var/www">
        AllowOverride None
        # Allow open access:
        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>
```

4. Restart httpd

```
sudo systemctl restart httpd
```

5. Verity that httpd is set to start on a system reboot

```
sudo systemctl enable httpd
```

    verity connectivity to the server with a browser http://127.0.0.1/login

6. SELinux may prevent connection to the web server. This can be remedied by executing the
    following two commands to create a new policy and enable it.

```
grep httpd /var/log/audit/audit.log | audit2allow -M esp_policy
semodule -i esp_policy.pp
```

    verity connectivity to the server with a browser http://127.0.0.1/login

    Troubleshoot other connectivity issues as needed

9 Configure Database Backups

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

```
sudo mkdir -p /srv/esp/backup
```

1. Create the logrotate configuration file

2. Copy the logrotate configuration file from the /srv/esp/prod/share directory to /srv/esp/backup

```
NOTE: sudo su
mkdir -p /srv/esp/backup
vi /srv/esp/backup/postgresql-esp.logrotate
```

Populate the file by copying the configuration The full contents of the file "esp-postgres.logrotate" are in shown in Appendix A

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

3. Modify the permissions and ownership on the file and /directory

```
sudo chown -R postgres.postgres /srv/esp/backup
chmsudo chmod 644 /srv/esp/backup/esp-postgres.logrotate od 644
/srv/esp/backup/postgresql-esp.logrotate
```

4. Create initial versions of the backup or "log" files as the postgres user

```
sudo su postgres
cd /srv/esp/backup

touch esp.daily.dump
touch esp.weekly.dump
touch esp.monthly.dump
```

5. Prime each of the three files by running a backup manually

```
/usr/bin/pg_dump esp -F c > /srv/esp/backup/esp.daily.dump
/usr/bin/pg_dump esp -F c > /srv/esp/backup/esp.weekly.dump
/usr/bin/pg_dump esp -F c > /srv/esp/backup/esp.monthly.dump
```

6. While still the postgres users, add the following entry to the postgres user's 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 10pm each day.

```
crontab -e

#  min      hour      dom    mon    dow     command
0     22      *      *      *      /srv/esp/scripts/db_backup.sh
```

7. Create the **/srv/esp/scripts/db_backup.sh** file

```
sudo su esp

vi /srv/esp/scripts/db_backup.sh
```

Populate it with the following:

```
#!/bin/bash
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
# @contact   esp_support@commoninf.comauthor: Rich Schaaf
# @organization: Commonwealth Informatics, Inc
# @copyright: (c) 20139 Commonwealth Informatics, Inc.
# @license: LGPL
#
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
/usr/sbin/logrotate -s /tmp/esp.logrotate.status /srv/esp/backup/esp-
   postgres.logrotate
```

```
Save and Exit.
```

8. Set the permissions on script:

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

10 Configure ESP log file rotation

1. 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/prod/share/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   This file is also listed in Appendix B for reference

2. 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*
   ```

11 Configure Log Monitoring

This process will monitor the 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

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.

1. **Install swatch**

```
sudo yum install swatch
```

2. **Install xtail**

```
sudo yum install xtail
```

or
```
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
sudo sh make
sudo sh make install ( you can ignore any errors about creating the man file)
```

3. **Create the esp.log conf file**

```
cd /etc
sudo vi swatch_esp.conf
```

Add the following: (customize email addresses as appropriate – separated by a colon)

```
    watchfor /CRITICAL/
     mail addresses=testemail1\@test.org:testemail2\@test.org,
subject=CRITICAL_Error_In_ESP_Log_File
```

```
      threshold track_by=CRITICAL, type=limit, count=1, seconds=60
```

4. Create the cron log conf file
(Note: this will monitor all files in the ESP cron log directory.)

```
cd /etc
sudo vi swatch_cron.conf
```

Add the following: (customize email addresses as appropriate – separated by a colon)

```
ignore /spiffymcgee|records with 0 errors|Errors:/
watchfor /ERROR|error|Error|No such file|Permission denied|Traceback/
 mail addresses=testemail1\@test.org:testemail2\@test.org,
subject=ERROR_Found_In_Cron_Log_File_Directory
    threshold track_by=ERROR, type=limit, count=1, seconds=60
```

5. Create the esp startup script.

You must customize the file for the following: (the parameters are highlighted below):
   1. The location of the esp.log files
   2. The path to xtail
        a. Use the command: `which xtail` to confirm the path
   3. Name 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 to use. The one that returns a path is the correct one.

```
cd /etc/init.d
sudo vi swatch_esp
```

Add the following as the contents of the file:

```
#!/bin/sh
# Simple Log Watcher Program

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

        ;;
'stop')
        PID=`cat /var/run/swatch_esp.pid`
        kill $PID
        ;;
*)
        echo "Usage: $0 { start | stop }"
        ;;
```

```
esac
exit 0
```

Set the Permissions:

```
sudo chmod 755 swatch_esp
```

6. Create the cron startup script.

You must customize the file for the following: (the parameters are highlighted below):
   a. The DIRECTORY of the ESP cron logs
   b. The path to xtail
        i. Use the command: `which xtail` to confirm the path
   c. Name 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.
        i. Use the commands `which swatch` and `which swatchdog` to determine which
            process name to use. The one that returns a path is the correct one.

```
cd /etc/init.d
sudo vi swatch_cron
```

Add the following as the contents of the file:

```
#!/bin/sh
# chkconfig: 2345 80 20
# Simple Log Watcher Program

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


        ;;
'stop')
        PID=`cat /var/run/swatch_cron.pid`
        kill $PID
        ;;
*)
        echo "Usage: $0 { start | stop }"
        ;;
esac
exit 0
```

Set the Permissions:

```
        sudo chmod 755 swatch_cron
```

7. Start the monitors
```
        sudo /etc/init.d/swatch_esp start
        sudo /etc/init.d/swatch_cron start
```

8. Test it out.
(NOTE: run the following commands - update file names and locations as appropriate)

```
        sudo su esp

        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/log/daily_log_cron.test
```

You should receive an email with the errors above.

9. Add it to system startup

```
cd /etc/init.d
        sudo chkconfig --add swatch_esp
        sudo chkconfig --add swatch_cron
```

Copyright © 2019 Commonwealth Informatics Inc.

~~11~~12 Setting up the data feed/Importing data to ESP

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 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:

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.

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.

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

~~12~~13 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

A typical command to load data might look like this:
        (NOTE: file names need to follow this format)

```
$ $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 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

Copyright © 2019 Commonwealth Informatics Inc.

process is not currently well documented. We are working on developing a document set for this purpose.

```
/srv/esp/prod/bin/esp case_report --mdph --transmit --status 'Q'

/srv/esp/prod/bin/esp case_requeue --status=RS
```

~~13~~14 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
```

~~14~~15   Configure crontab to run Daily ESP jobs

As the ESP user, copy the sample daily_batch.sh script from 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.

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

Often this file is initially set to only load the daily files and then as system validation progresses, additional commands like hef, nodis and case_report will be enabled.

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
```

~~15~~16 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

    ```
    $ $/srv/eps/prodESP_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

Copyright © 2019 Commonwealth Informatics Inc.

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:

```
/srv/eps/prod/$ $ESP_HOME/bin/esp hef
/srv/eps/prod$ $ESP_HOM/E/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.

1617 How to Clean Up and Reload ESP Data files (Provenances)

Every time a file is loaded into ESP, a Provenance is created and associated with it. Sometimes it is helpful to clean all or some of the data loaded. There are several ways to accomplish this task but the simplest is to delete all the Provenance entries from the administration UI (except #s 1 and 2) This method will automatically delete all of the related records in the database.

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

3. Select the records you want to delete by clicking on the left check box next to the Provenance id

DO NOT DELETE Provenances #1 and #2!!

4. From the same screen at the top - select "Delete selected provenances" from the "Action" drop down.

5. Next, click the "Go" button to the right of the drop down.

6. Before deleting anything, the system will calculate all the 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:
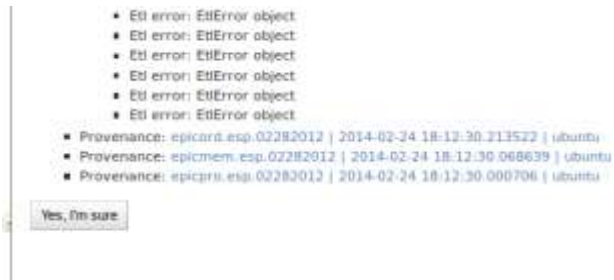
You are done!

Appendix A - 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
        /usr/bin/pg_dump esp -F c > /srv/esp/backup/esp.daily.dump
    endscript
}

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

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

Appendix B - ESP Log Rotation Script

```
# esplogrotate - this is the logrotate Config file for /var/log/esp.log
# Modify as needed for longer log retention
#   the default is rotate the esp log weekly and to keep 4 weeks
# This file should be copied to /etc/logrotate.d/
#
/var/log/esp.log {
    weekly
    missingok
    rotate 4
    delaycompress
    dateext
    shred
}
```

Appendix C – ESP httpd vhost file

```
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
#
#                     Apache/httpd configuration for ESP
#
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
# Old prod code may have 'print' statements scattered throughout; this
# directive prevents mod_wsgi from bombing out when it hits such a statement.
WSGIRestrictStdout Off

<VirtualHost *:80>
    ServerName scooby
    ServerAdmin esp_support@yourdomain.com
    Alias /static/ /srv/esp/prod/ESP/media/static/
    Alias /media /srv/esp/prod/ESP/media/
    WSGIDaemonProcess esp
    WSGIScriptAlias / /srv/esp/prod/etc/django.wsgi
    <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>
```