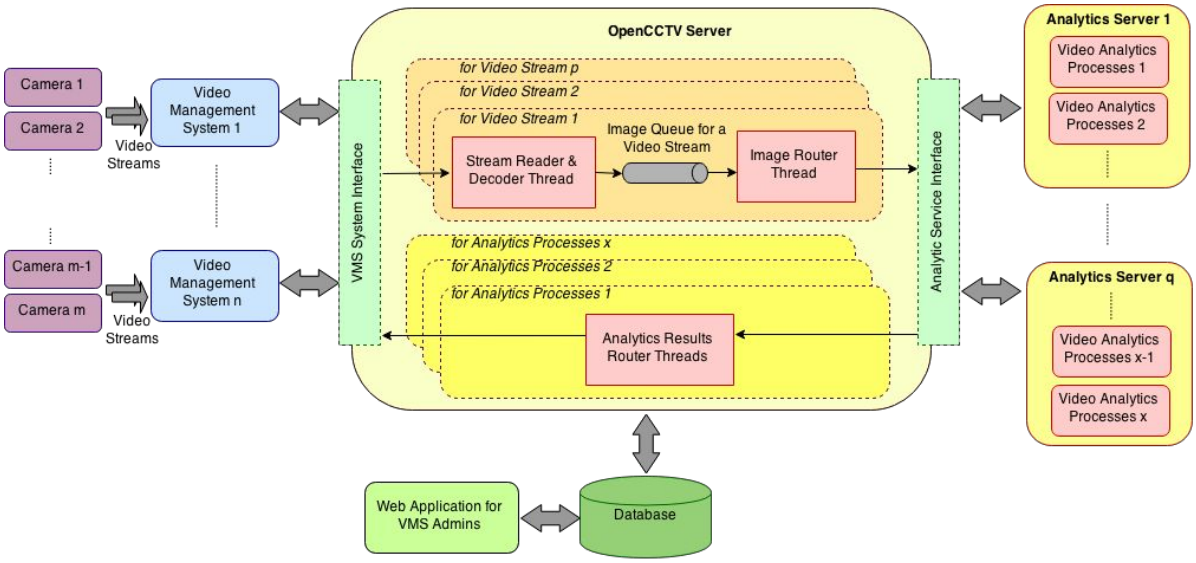
OpenCCTV Platform Installation Manual

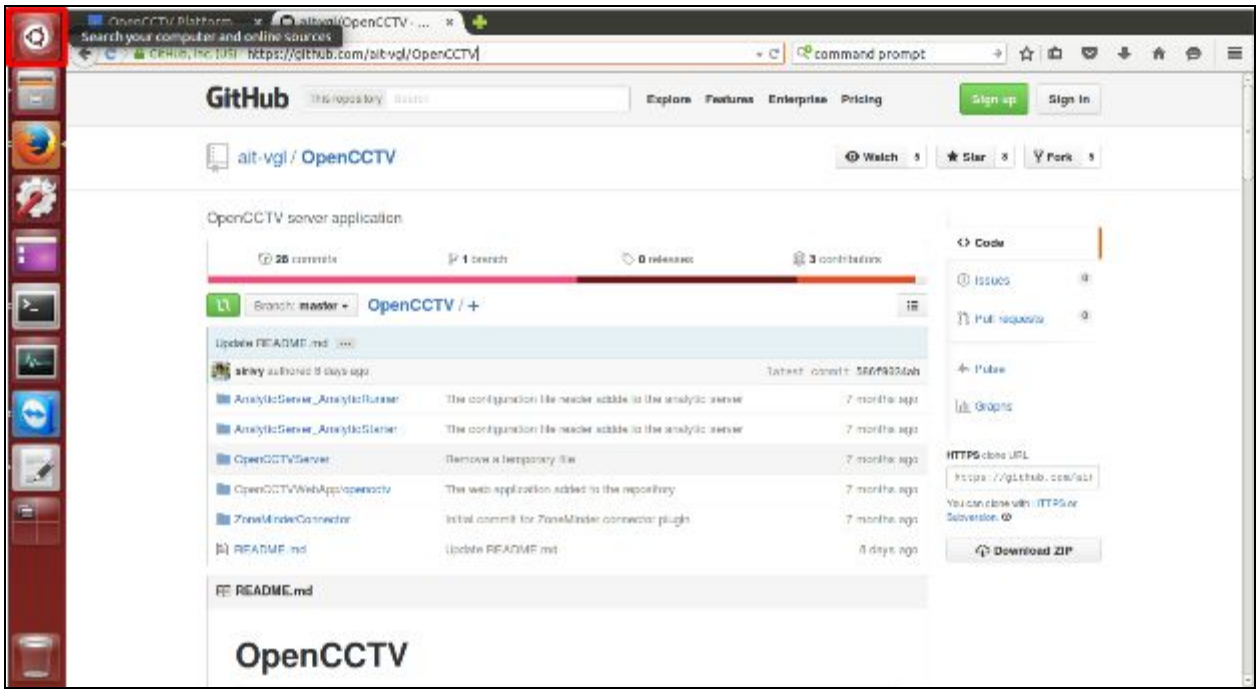
System diagram of OpenCCTV system



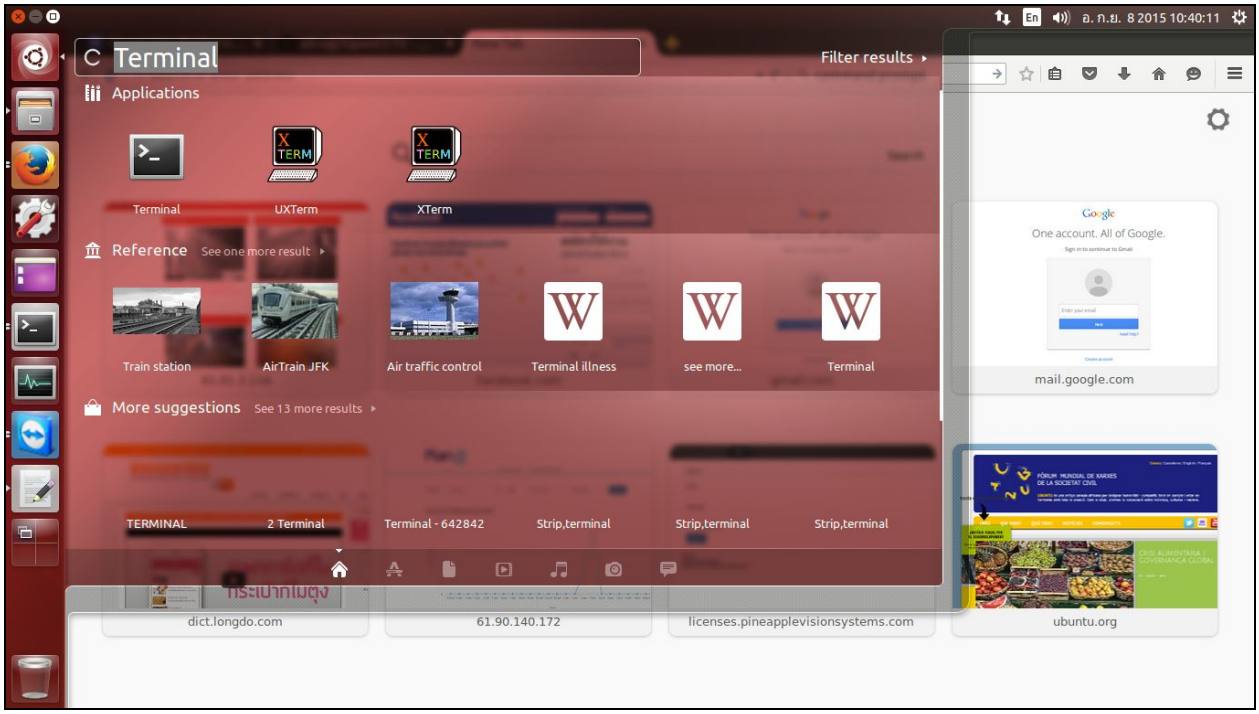
The OpenCCTV platform is developed on Ubuntu operating system. A web application is developed using Ruby on Rails architecture. The backend system is developed using C/C++. Ubuntu 14.04 is required for the OpenCCTV deployment. User can download Ubuntu ISO image from http://www.ubuntu.com/download/desktop. After user finishes Ubuntu installation, user can follow the following steps to deploy OpenCCTV platform.

Step 1: Clone OpenCCTV Platform repository

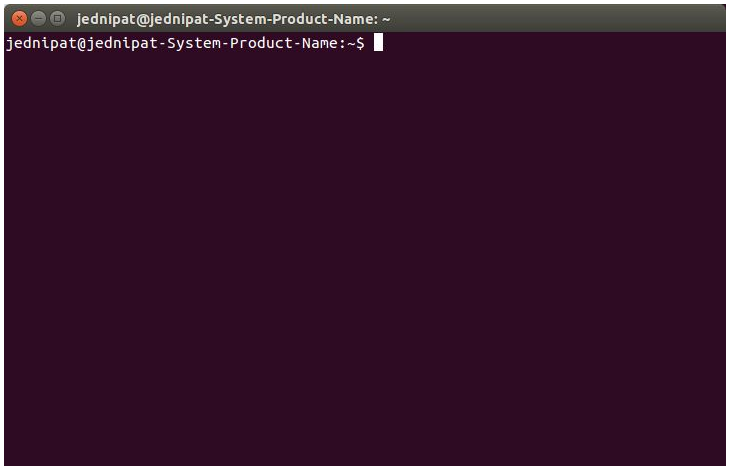
The OpenCCTV platform is an open source project. Source code is available on Github: https://github.com/ait-vgl/OpenCCTV. The first step of installation is cloning a repository to user's computer. User first needs to open a Terminal window, which is a command prompt tool in Ubuntu, by clicking at the icon on the top of toolbar on the left side of desktop screen.



Then, user has to type a word "Terminal" in a text field as shown in figure below. After that, user has to click on an icon "Terminal".



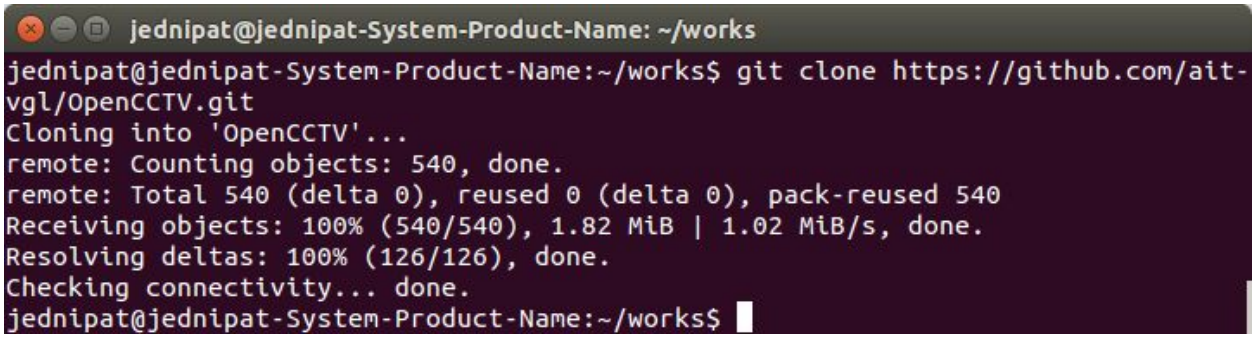
A terminal window will be pop-up and waits for a command line from user. A default directory of terminal window is /home/your_username.



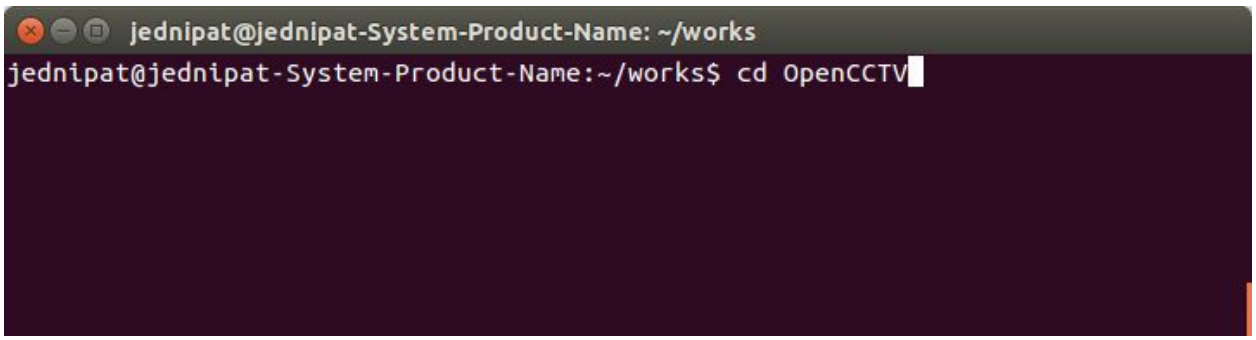
At this step, user has to clone the OpenCCTV repository to user's computer by executing the following command:

git clone https://github.com/ait-vgl/OpenCCTV.git

A result from executing this command is shown in figure below.



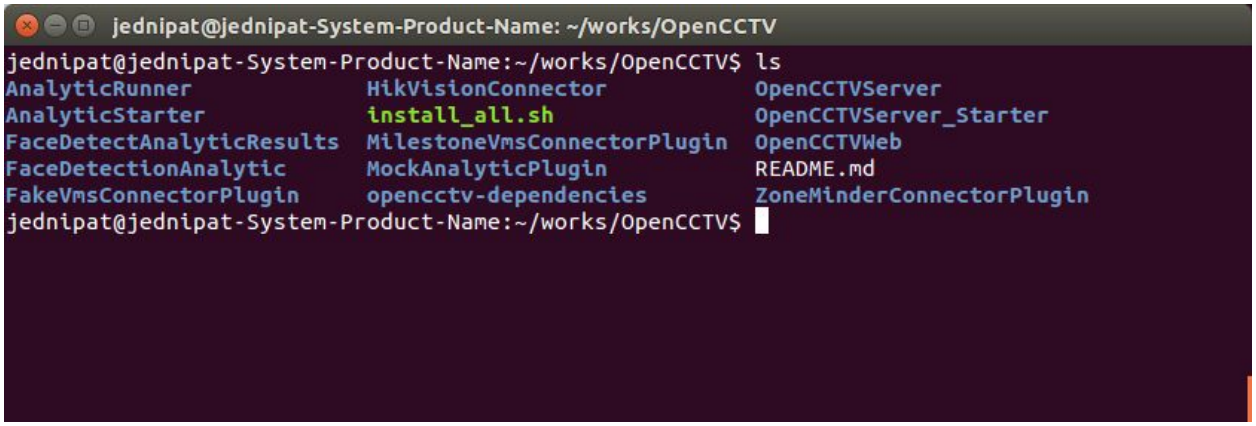
At this step, user will get a folder named “OpenCCTV”. To change directory, user has to type in a command “cd OpenCCTV” and press an enter button.



Now your current directory is changed to OpenCCTV directory. An OpenCCTV folder consists of 13 subfolders which are

- **AnalyticRunner** is a module which works with AnalyticStarter to start/stop analytic module.
- **AnalyticStarter**
- **FaceDetectAnalyticResults** is a simple Ruby on Rails web application for displaying the output result from FaceDetectionAnalytic.
- **FaceDetectionAnalytic** is a simple face detection software. The output of the software is stored in the database used by FaceDetectAnalyticResults.
- **FakeVmsConnectorPlugin** is an example of VMS connector.
- **HikVisionConnector** is a connector for getting a video stream from HikVision NVR.
- **MilestoneVmsConnectorPlugin** is a connector for getting a video stream from Milestone VMS.
- **MockAnalyticPlugin** is an example of analytic module. If user would like to use your own analytic, user has to follow the APIs in this example. User also can look into the FaceDetectionAnalytic as an example too.
- **opencctv-dependencies** contains libraries required for OpenCCTV platform.
- **OpenCCTVServer** is a module which works with OpenCCTVServer_Starter. This module is used to start/stop server.
- **OpenCCTVServer_Starter**

- **OpenCCTVWeb** is a main Ruby on Rails web application source code.
- **ZoneMinderConnectorPlugin** is a connector used for getting a video stream from ZoneMinder VMS.



Step 2: Ruby on Rails installation

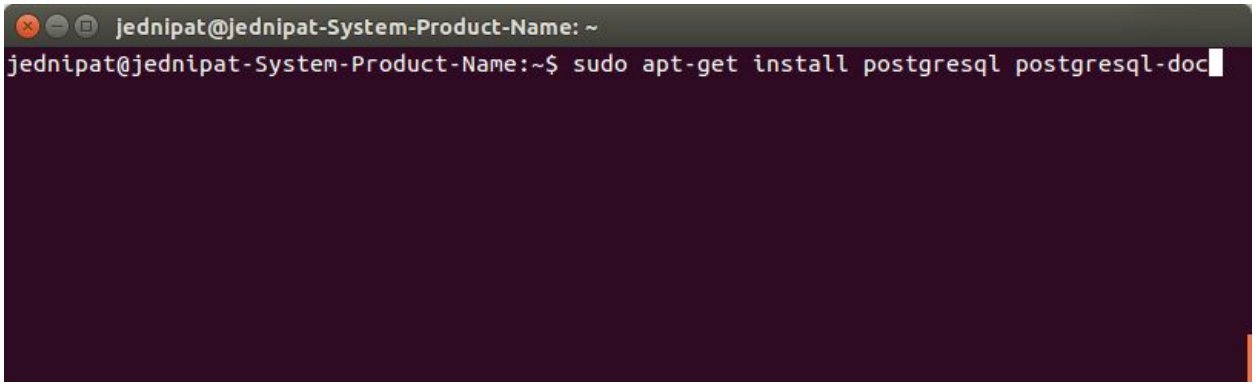
Reference: https://cis.ait.asia/course_offerings/224/rails-companion

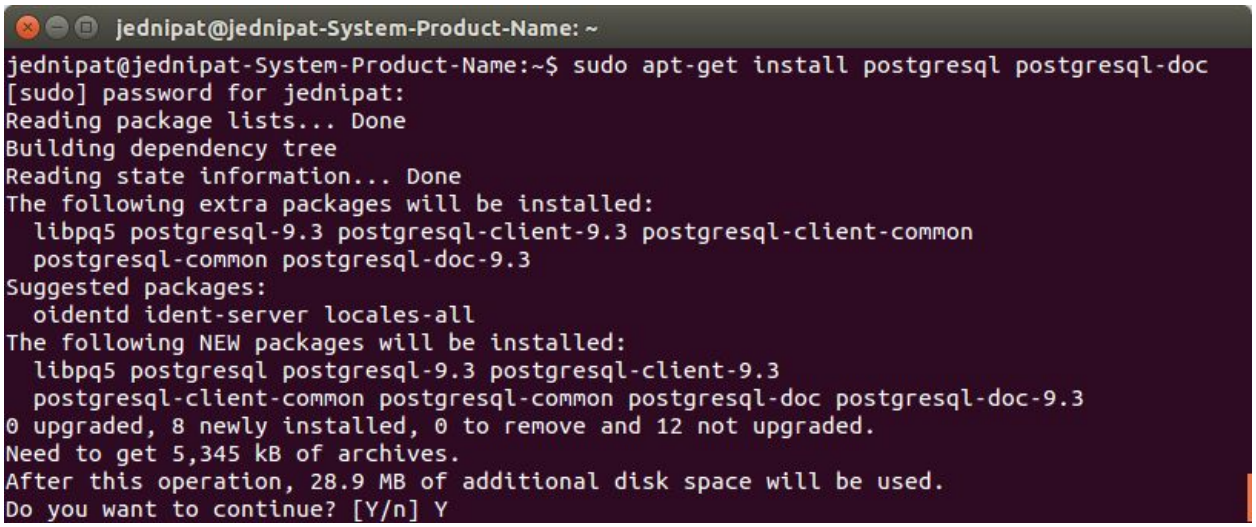
To set up OpenCCTV Web application, user has to install Ruby on Rails framework. The instruction about Ruby on Rails installation was written by Dr. Matthew N. Dailey. User can visit https://cis.ait.asia/course_offerings/224/rails-companion and follow the instuction step-by-step. If user can not access the reference link, user can follow the instruction below, which is copied from the reference link.

These instructions are oriented towards the **Ubuntu 14.04 (Trusty Tahr)** desktop installation. For other Linux distros, the instructions can vary.
Please note that this document provides quick-start information only. If you need more detailed information, look through the reference section or search the Web.

PostgreSQL Installation

Install PostgreSQL before you create your Rails projects. This is simple with Ubuntu: 14.04:
$ sudo apt-get install postgresql postgresql-doc



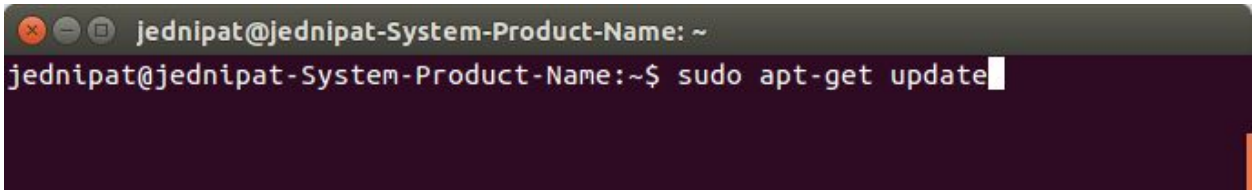


Rails Installation

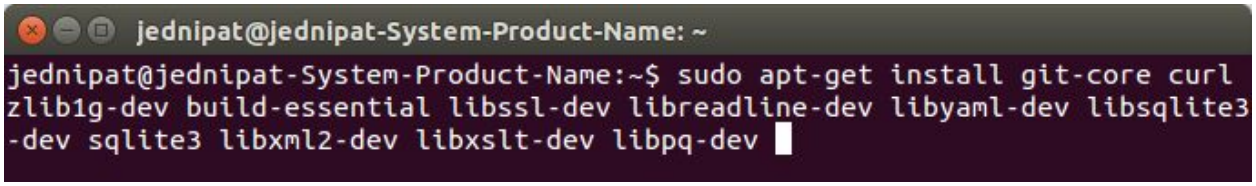
You can install Ruby, Ruby Gems, and other Ruby-related packages using apt-get or from source packages. But these approaches are outdated and lead to major headaches when you need to update to new version or work with older versions.
We will be using rbenv (a new lightweight Ruby version management tool) to install Ruby, Ruby Gems, Rails, and other Ruby-related packages. rbenv is a command-line tool allowing you to easily install, manage, and work with multiple Ruby environments including interpreters and sets of Ruby gems.
Ruby Gems is a package manager for the Ruby programming language that provides a standard way of distributing Ruby programs and libraries (in a self-contained format called a "gem"), a tool designed to easily manage the installation of gems, and a server for distributing them.

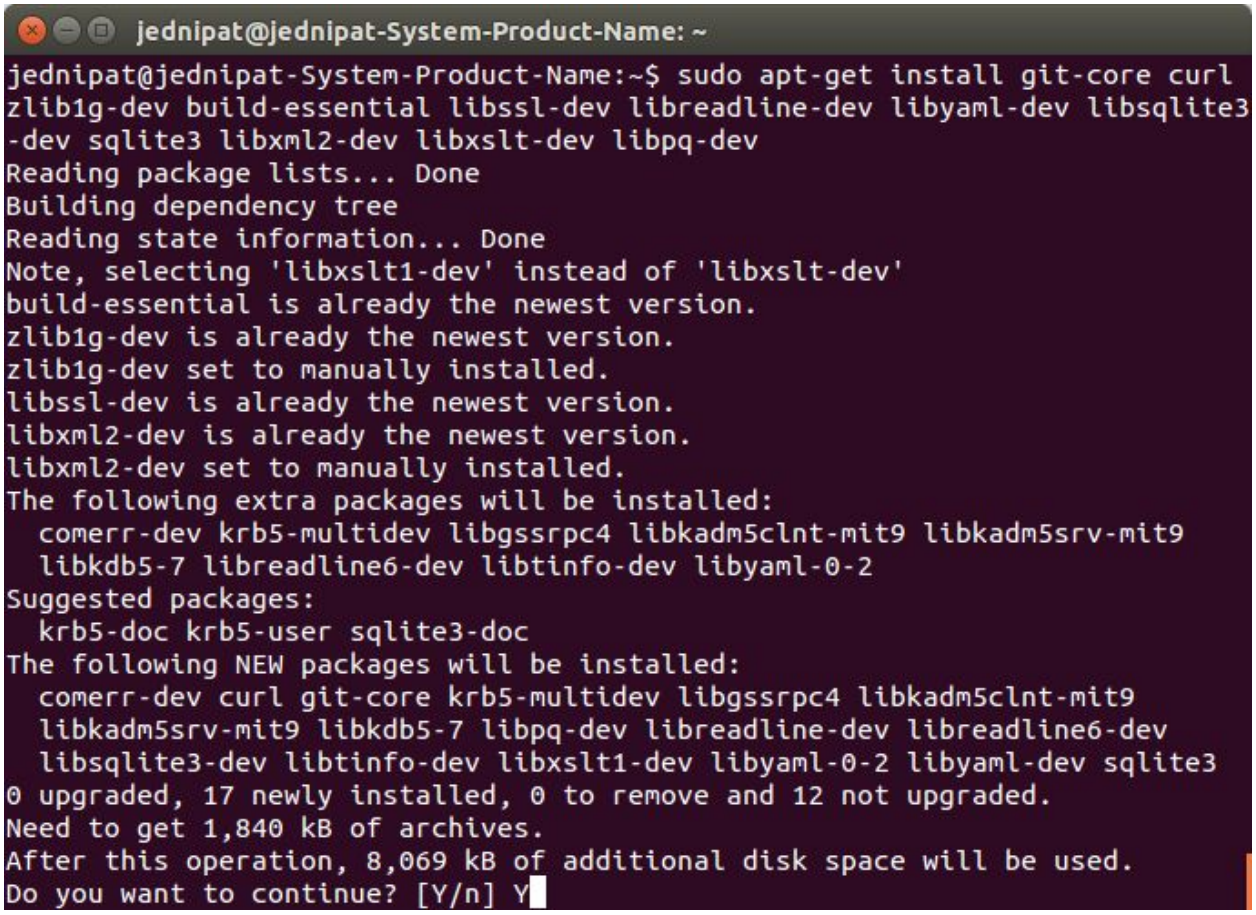
Housekeeping

First of all, run
$ sudo apt-get update



so that we have the latest sources and update on our machine.
Next, we're going to install git-core and curl which are both required to install and use rbenv, and build-essential which is required to compile Ruby and some of the Ruby gems.
$ sudo apt-get install git-core curl zlib1g-dev build-essential libssl-dev libreadline-dev libyaml-dev libsqlite3-dev sqlite3 libxml2-dev libxslt-dev libpq-dev



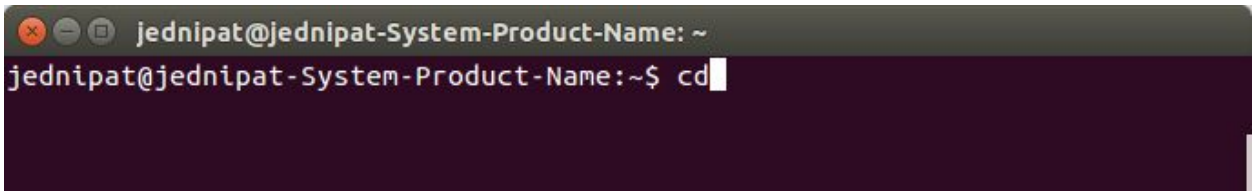


Installing rbenv

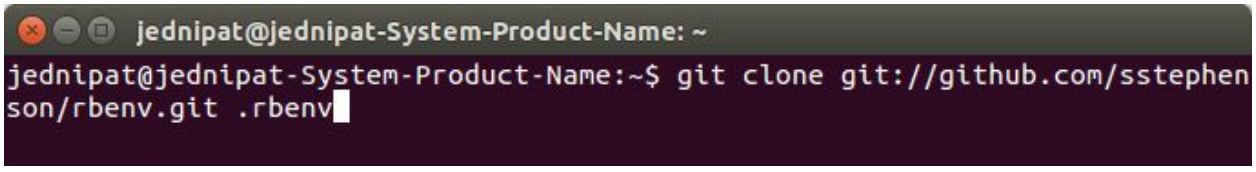
Installing with rbenv is a simple two step process. First you install rbenv, then you install the ruby-build plugin for rbenv.

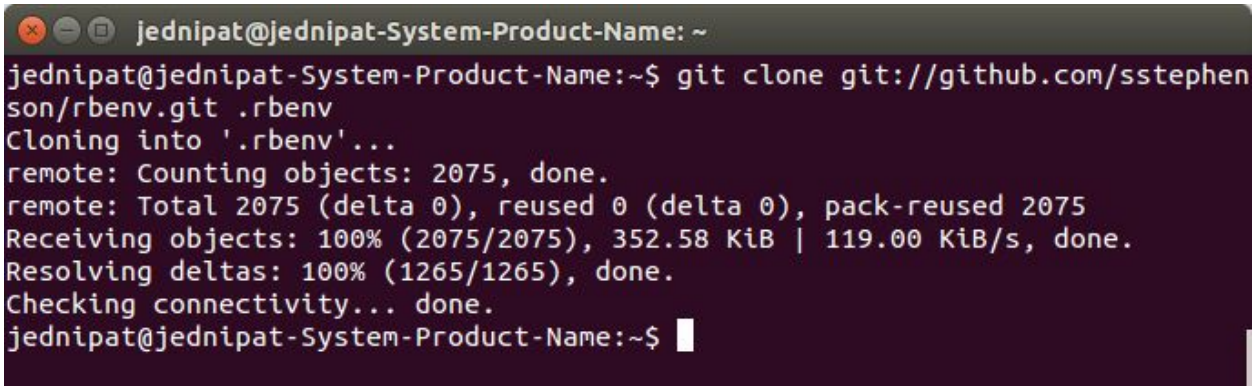
First install the latest version of rbenv:

$ cd

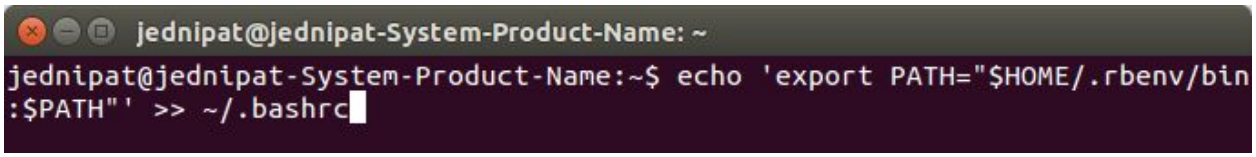


$ git clone git://github.com/sstephenson/rbenv.git .rbenv

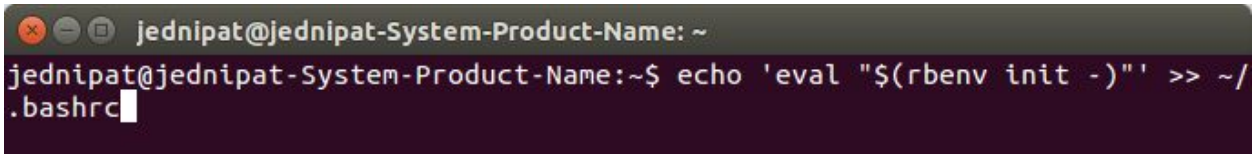




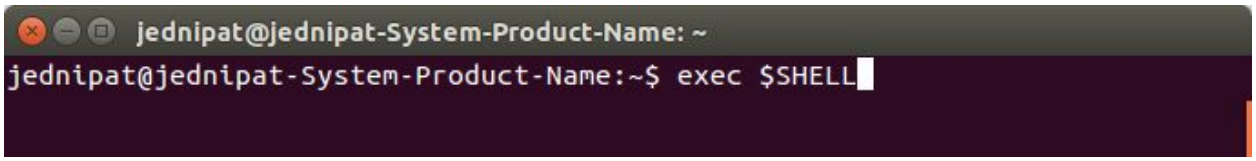
$ echo 'export PATH="$HOME/.rbenv/bin:$PATH"' >> ~/.bashrc



$ echo 'eval "$(rbenv init -)"' >> ~/.bashrc

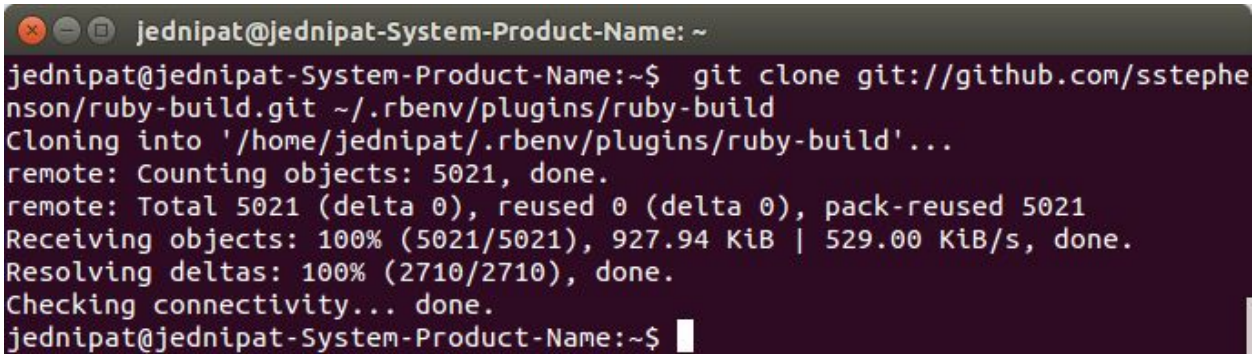


$ exec $SHELL

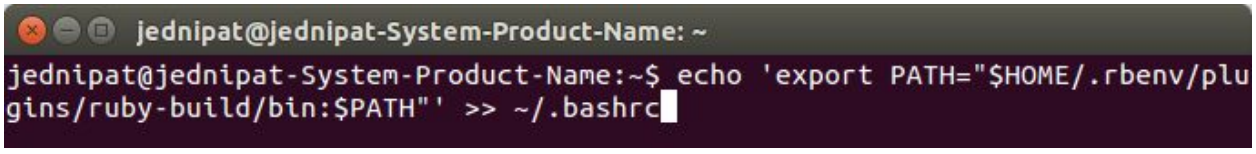


Next install the ruby-build plugin for rbenv:

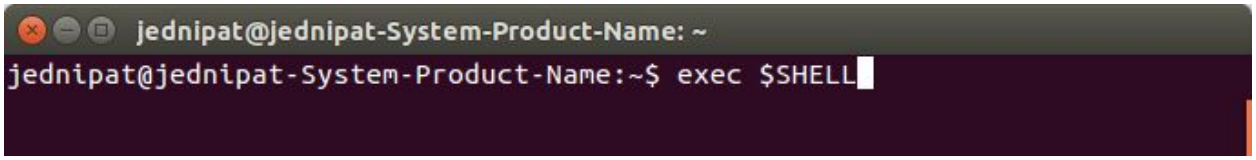
$ git clone git://github.com/sstephenson/ruby-build.git ~/.rbenv/plugins/ruby-build



$ echo 'export PATH="$HOME/.rbenv/plugins/ruby-build/bin:$PATH"' >> ~/.bashrc



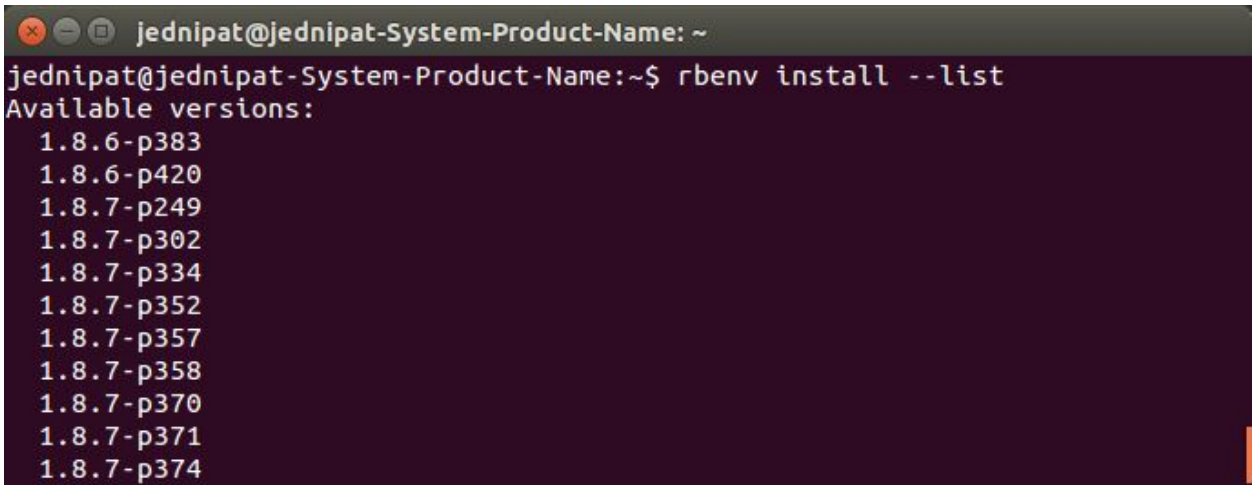
$ exec $SHELL



Installing Ruby

Now we can install any version of Ruby we like. To check what releases of Ruby are available, you could run

$ rbenv install --list



and see that (at the time I'm writing) the latest release is 2.1.2-p95. Install it:

$ rbenv install 2.1.2

$ rbenv global 2.1.2

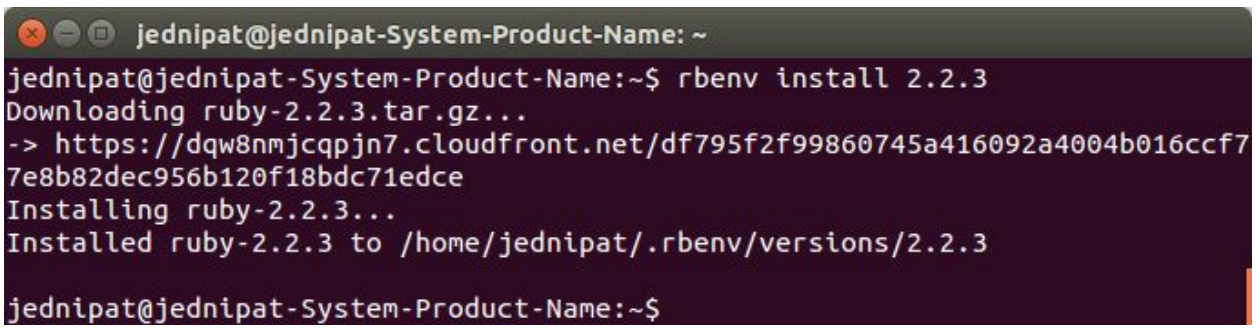
$ ruby -v

You should see output similar to this:

ruby 2.1.2p95 (2014-05-08 revision 45877) [x86_64-linux]

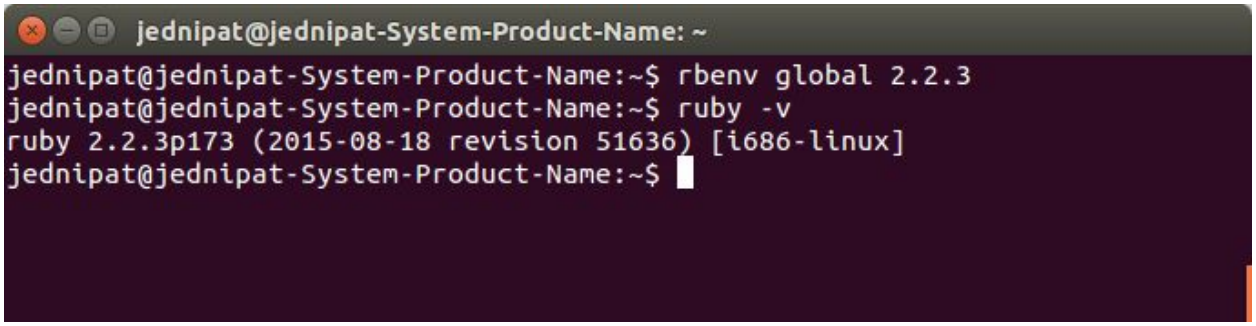
NOTE: On 8 September 2015, the latest release is 2.2.3. User can install the latest release by executing the following commands:

$ rbenv install 2.2.3



NOTE: This step may take some time.

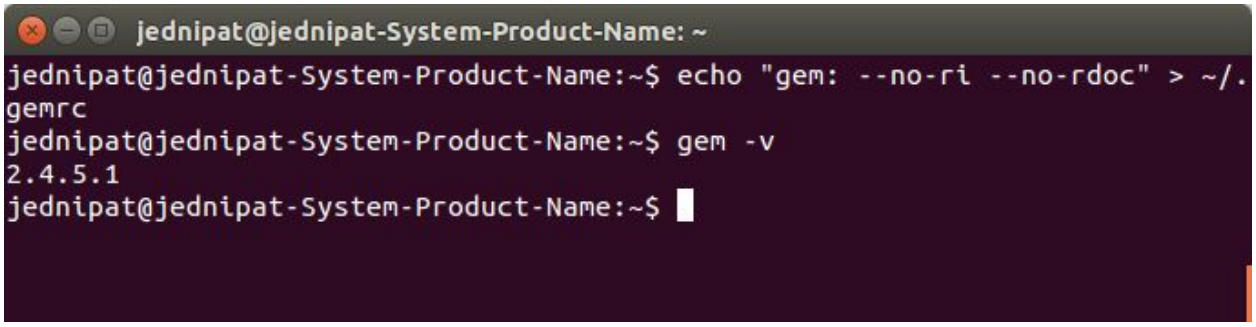
$ rbenv global 2.2.3
$ ruby -v
You should see this output:



Ruby Gems

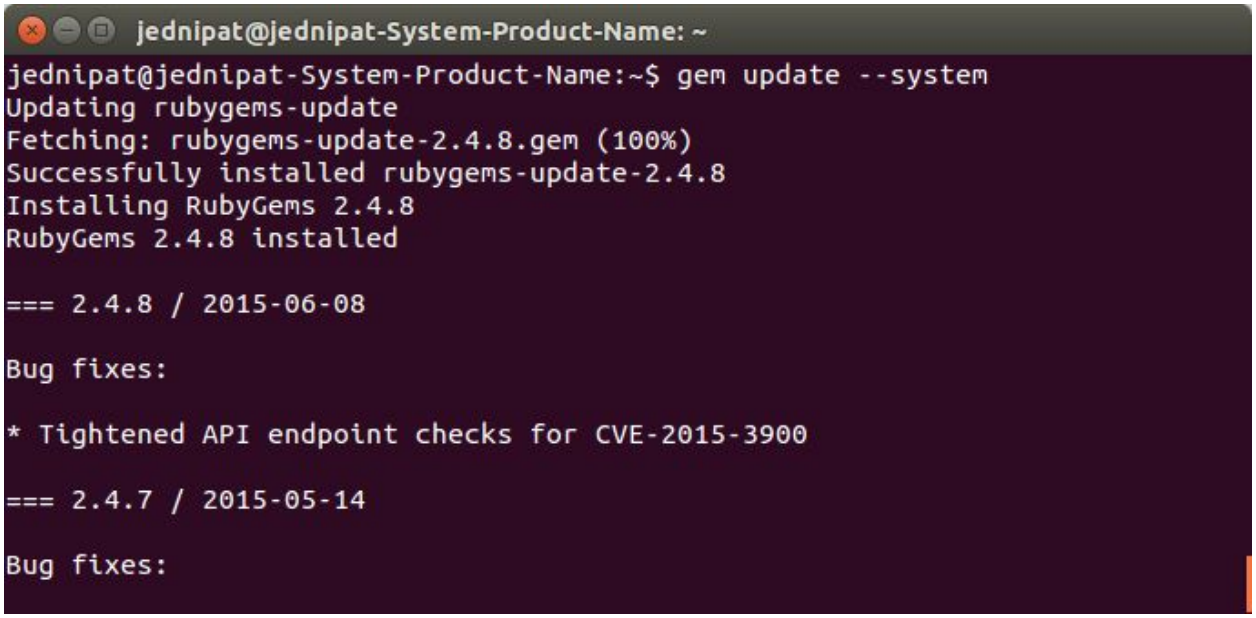
The following step is to tell Rubygems not to install the documentation for each package locally. you can simply update your ~/.gemrc file.
$ echo "gem: --no-ri --no-rdoc" > ~/.gemrc

Next, check your Ruby Gems version:
$ gem -v

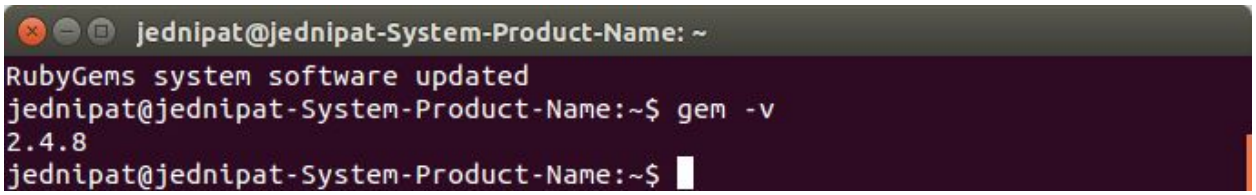


It should report version 2.2.2. Run the following command to update to the latest version:

$ gem update --system

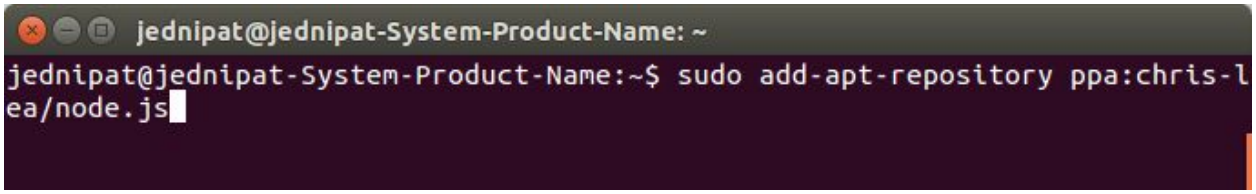


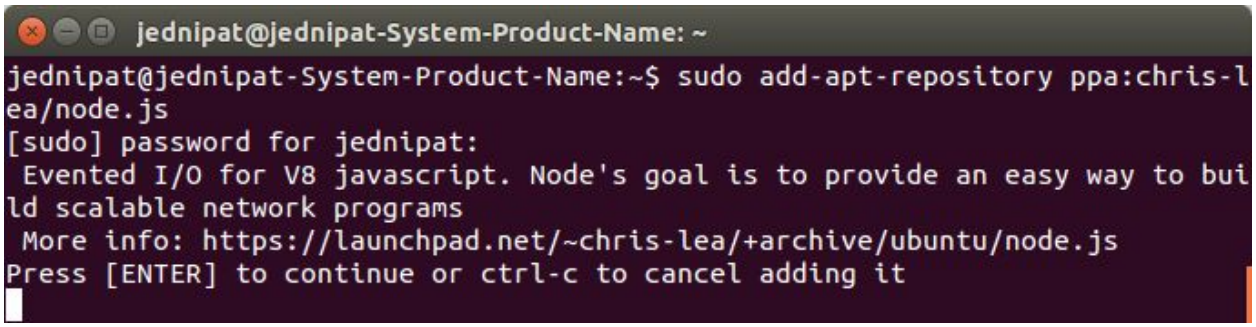
$ gem -v



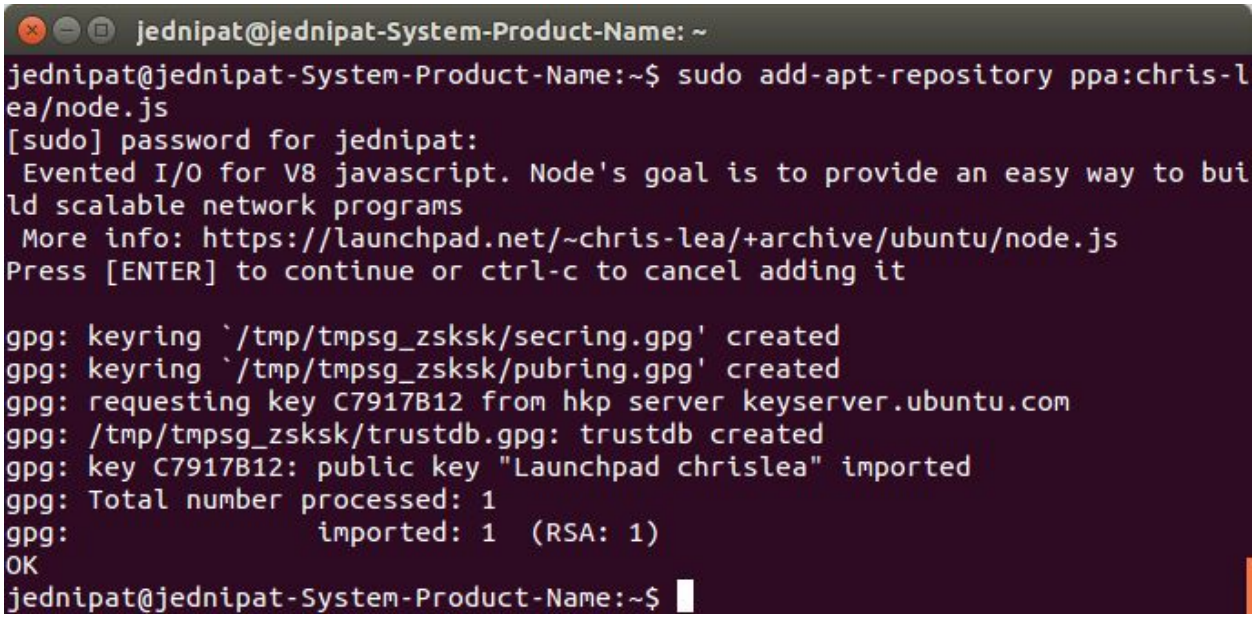
Installing Rails

Next let's install the latest version of Rails (4.1.5 at the time of writing). To install:
To install NodeJS, you can add it using a PPA repository:
$ sudo add-apt-repository ppa:chris-lea/node.js

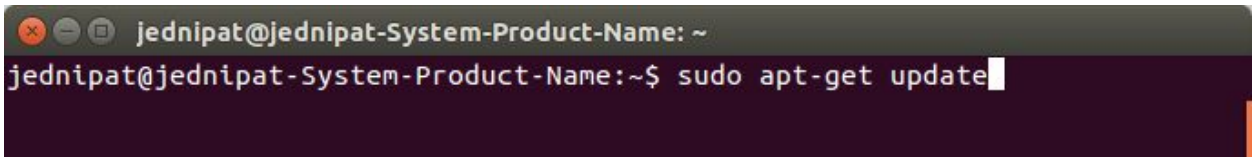




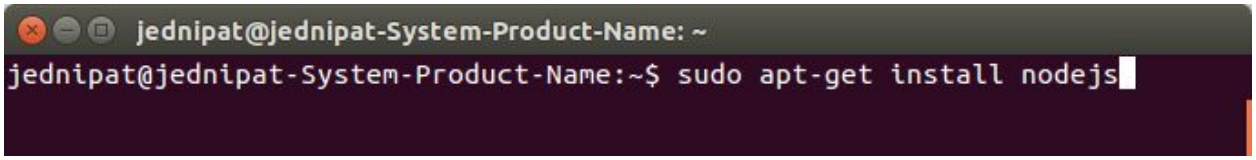
Press an enter button to proceed to next step



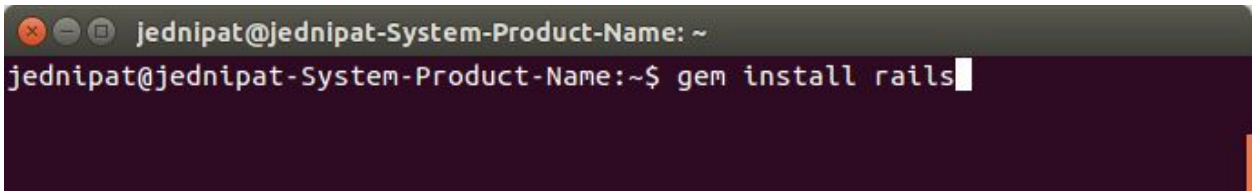
$ sudo apt-get update

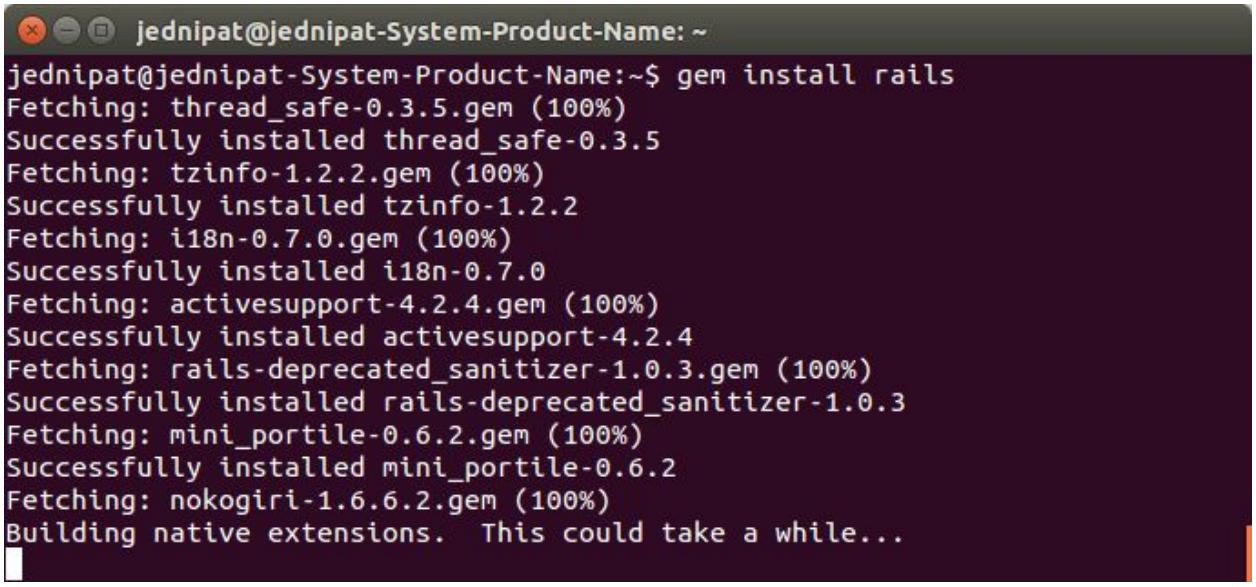


$ sudo apt-get install nodejs

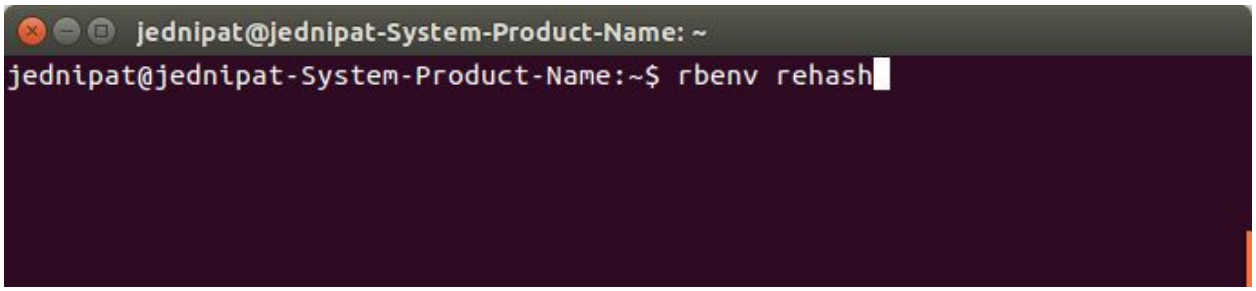


$ gem install rails

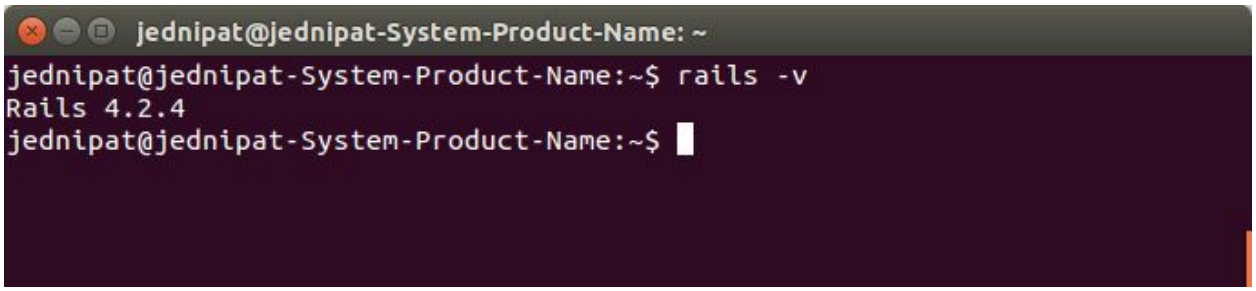




You'll need to run the following command to make the rails executable available:
$ rbenv rehash



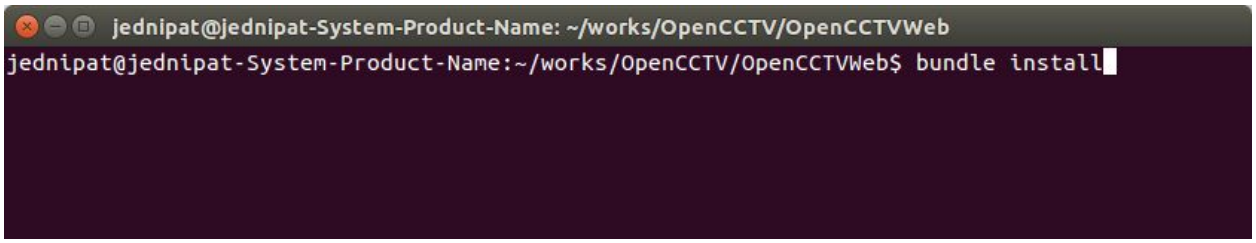
Now that you've installed Rails. You can check the version as follows:
$ rails -v

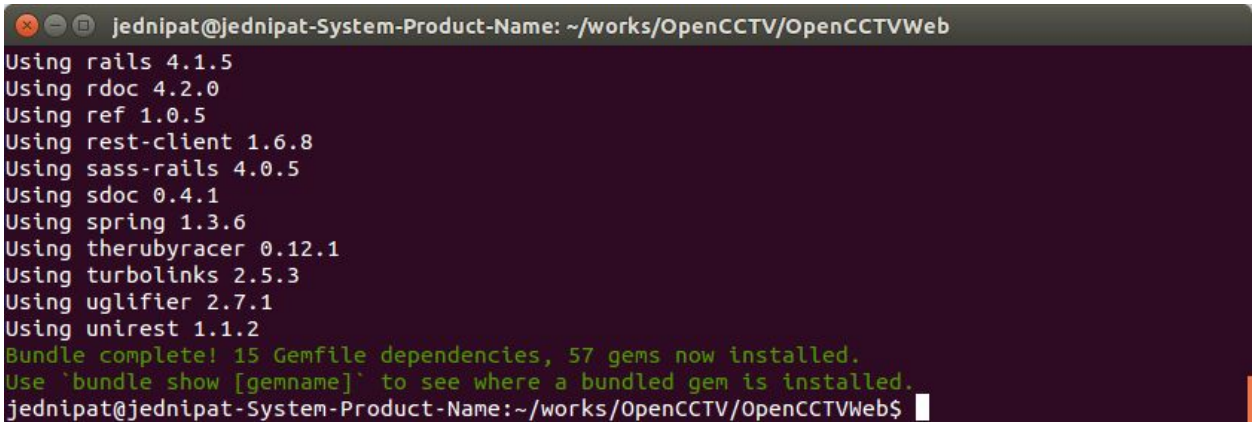


Step 3: OpenCCTV Web application configurations

User has to execute the following commands:

1) cd OpenCCTVWeb/
2) bundle install





3) If your computer doesn't have MySQL, you have to install MySQL

    sudo apt-get install mysql-server

    While installing MySQL, you will be asked to set username and password for administrator (root).
    Then, you have to create MySQL account for using with Ruby On Rails Web application by the following commands:

    mysql -u root -p

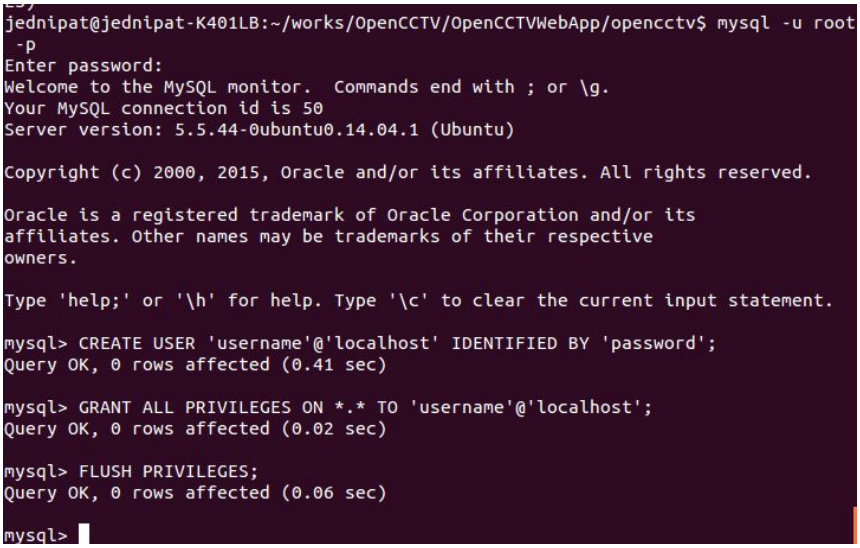
    After executing this command, you will be asked to enter a password. You have to fill the same password which is set during MySQL installation. Next, you have to enter the following commands for creating a new username and password, and setting privileges.

    CREATE USER 'username'@'localhost' IDENTIFIED BY 'password';

    GRANT ALL PRIVILEGES ON *.* TO 'username'@'localhost';

FLUSH PRIVILEGES;

NOTE: Please do not copy these commands and paste into terminal because a single quote character (‘) will create a problem. You have to type in a command by yourself.



If you can execute all three commands without error messages, you have to press Ctrl+d (hold a Ctrl button and press ‘d’ button) to exit mysql command line. In this example, a new user called “username” is created for “localhost”. A *username’*s password is “password”. After that, you have to restart mysql service by executing this command:

sudo service mysql restart

Before making changes in database.yml, you need to run a command

gem install mysql2

at opencctv folder.

At this step, user needs to understand database configuration. If user opens a database.yml file in OpenCCTV/OpenCCTVWeb/config, the following lines will be shown up:

MySQL. Versions 5.0+ are recommended.
#
Install the MYSQL driver
gem install mysql2
#
Ensure the MySQL gem is defined in your Gemfile

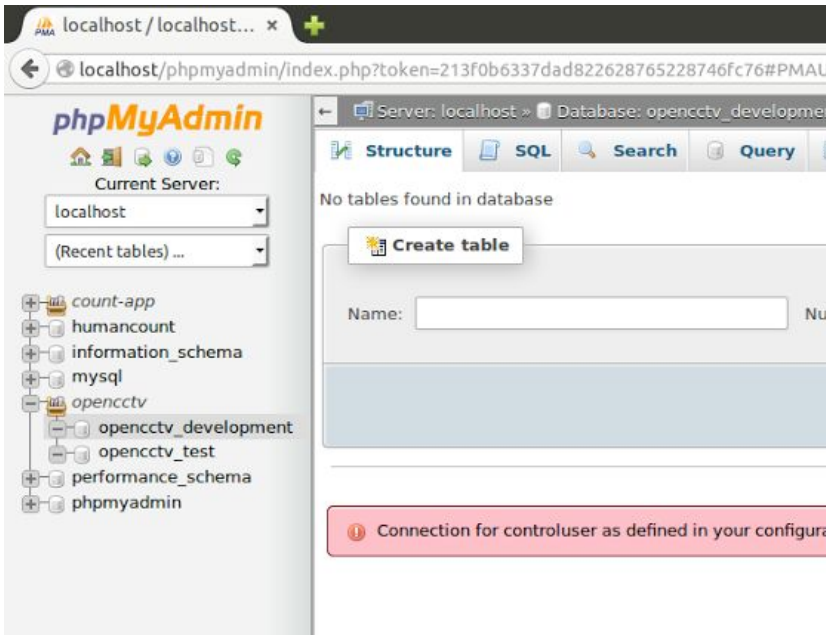
```
#   gem 'mysql2'
#
# And be sure to use new-style password hashing:
#   http://dev.mysql.com/doc/refman/5.0/en/old-client.html
#
default: &default
  adapter: mysql2
  encoding: utf8
  pool: 5
  username: username
  password: password
  host: host
   socket: /var/run/mysqld/mysqld.sock
```

Since a user called “username” is created for localhost, you have to change host name from “host” to “localhost”. If you would like to change Web application configurations, please make sure that you create a new username and password, and assign to the same hostname used in database.yml.

If everything is fine and there is no error message, we are now ready for creating a database which will be used by Ruby on Rails Web application. To create a database for Ruby on Rails, at opencctv folder, you have to execute a command

```
rake db:create
```

After executing this command, you should get database as shown in figure below.



NOTE: This Web GUI is PHPMyAdmin. User can install it by executing a

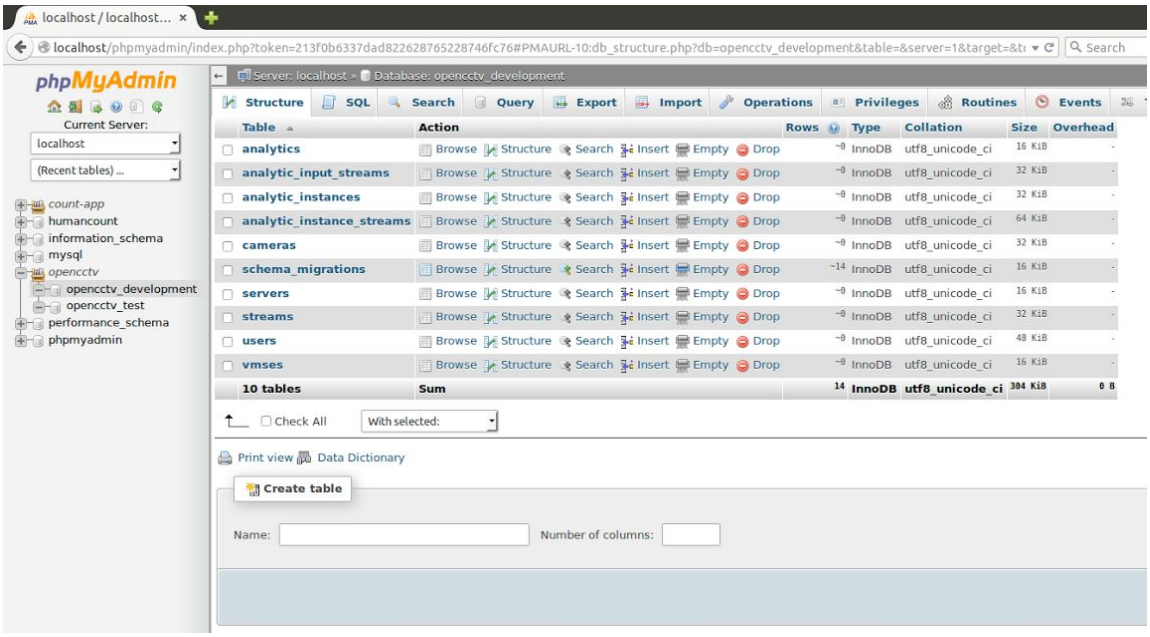
command: sudo apt-get install phpmyadmin

The rake db:create will just create only a database named “opencctv” and two tables: opencctv_development, opencctv_test.

Next, user has to run a command

rake db:migrate

to create all tables required for Ruby on Rails Web application as shown below.



Step 4: OpenCCTV installation

We are almost ready for OpenCCTV installation. Please make sure that you have installed the required libraries as follows:

- Boost C++ Libraries
    - Installation guide to Boost: http://particlephysicsandcode.com/2013/03/11/installing-boost-1-52-ubuntu-12-04-fedora/
- OpenCV
    - Installation guide to OpenCV: http://docs.opencv.org/trunk/doc/tutorials/introduction/linux_install/linux_install.html
- libVLC
    - sudo apt-get install x264 libvlccore-dev libvlc-dev
- libsodium
    - Download libsodium source code from https://download.libsodium.org/libsodium/releases/
    - Extract tar ball file
    - Execute the following command
        - ./configure
        - make
        - sudo make install
- Google Protocol Buffer
    - https://developers.google.com/protocol-buffers/?hl=en
- libzmq
    - Download libzeromq source code from http://zeromq.org/area:download
    - Extract tar ball file
    - Execute the following command
        - ./configure
        - make
        - sudo make install

- zmq header file
    - Download zmq.hpp from https://github.com/zeromq/cppzmq
    - Copy zmq.hpp to /usr/local/include/

- libmysqlconn-dev
    - sudo apt-get install libmysqlcppconn-dev

- Update recently installed libraries by executing a command
    - sudo ldconfig

NOTE: User can check about commands for installing required libraries in opencctv-dependencies folder.

There is one more step before we can compile the source code and install the OpenCCTV. User has to compile two .proto files (image.proto and analytic_result.proto) by using the command below.

protoc -I=/home/jednipat/works/OpenCCTV/OpenCCTVServer/src/opencctv/util/serialization/gpb --cpp_out=/home/jednipat/works/OpenCCTV/OpenCCTVServer/src/opencctv/util/serialization/gpb /home/jednipat/works/OpenCCTV/OpenCCTVServer/src/opencctv/util/serialization/gpb/image.proto

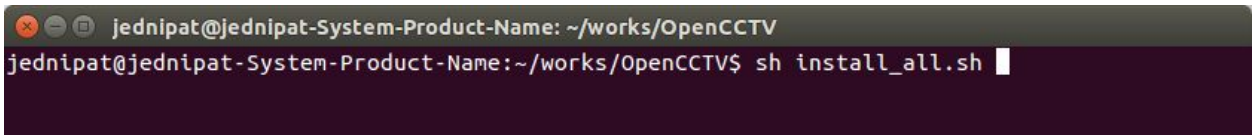
and

protoc -I=/home/jednipat/works/OpenCCTV/OpenCCTVServer/src/opencctv/util/serialization/gpb --cpp_out=/home/jednipat/works/OpenCCTV/OpenCCTVServer/src/opencctv/util/serialization/gpb /home/jednipat/works/OpenCCTV/OpenCCTVServer/src/opencctv/util/serialization/gpb/analytic_result.proto

User has to change a path to folder and file to match with your system environment. If there is no error, we can now proceed to final step of installation.

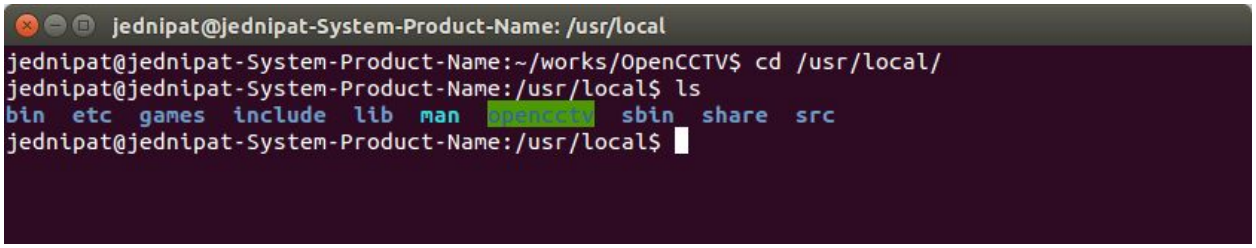
To install the OpenCCTV, user has to execute a command

sh install_all.sh

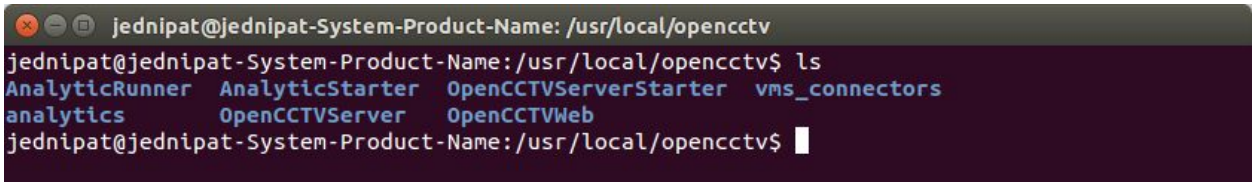


Executing this shell script will ask you to enter root's password. After that, you have to read the messages in the terminal carefully to make sure that all modules are compiled successfully without any error messages.

If the installation is completed, you will see a folder named "opencctv" in /usr/local directory.



The opencctv in /usr/local contains the following sub-folders.


jednipat@jednipat-System-Product-Name: /usr/local/opencctv
jednipat@jednipat-System-Product-Name:/usr/local/opencctv$ ls
AnalyticRunner AnalyticStarter OpenCCTVServerStarter vms_connectors
analytics   OpenCCTVServer   OpenCCTVWeb
jednipat@jednipat-System-Product-Name:/usr/local/opencctv$

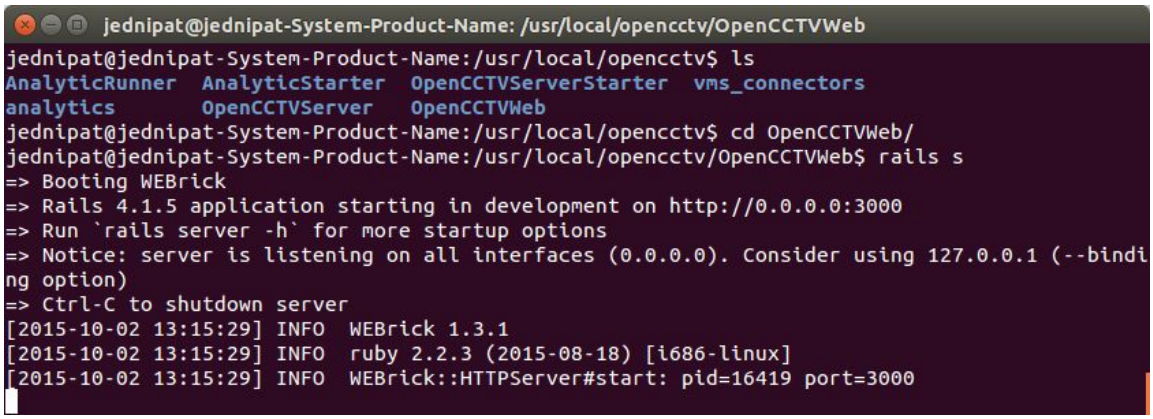
Step 5: Start OpenCCTV services

Once the installation is completed, user will get opencctv folder in /usr/local. We need to start three services which are

1) **OpenCCTVWeb**

   As we explain earlier, the OpenCCTVWeb is a main web application which is used for controlling the whole system. To start this module, we have to execute the following command:

   rails s



   *User **cannot close** this terminal window or **kill** this process. Otherwise, the web application will stop working. If you want to stop web application, you can hold a control button and press 'c' button on your keyboard (Ctrl+C). This is a shortcut key to kill process via terminal in Ubuntu OS.*

2) **AnalyticStarter**

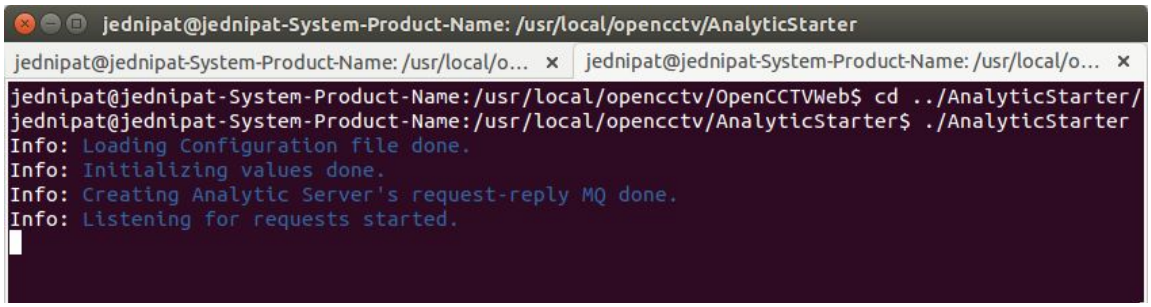
   AnalyticStarter is a module to handle analytic plugin for retrieving video stream from OpenCCTV server and passing a video frame to analytic module. To open a new terminal window without stopping the running process, user can press a hold ctrl and shift button, and the press 't' button on your keyboard (Ctrl+Shift+T). This shortcut key will open a new tab of terminal window within the same window. You will see a terminal window like this.

Once we open a new terminal under the same window, we have to change our current directory to AnalyticStarter directory.

```
cd ../AnalyticStarter
```

Then, user has to execute the AnalyticStarter using a command

```
./AnalyticStarter
```

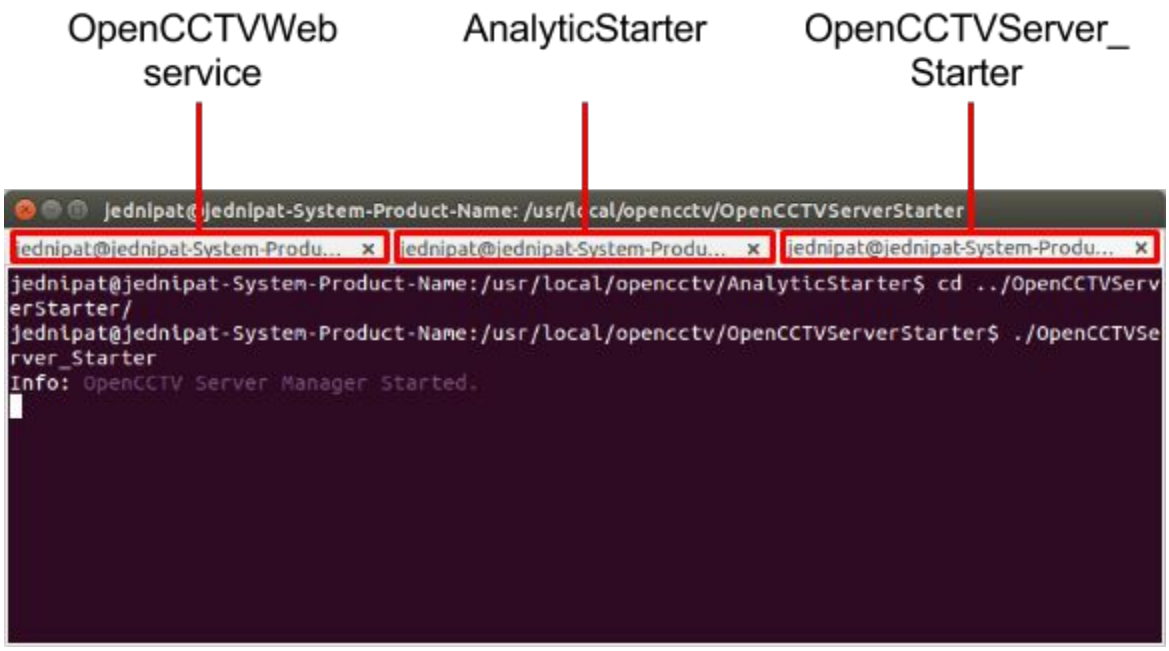


*As previously mentioned in OpenCCTVWeb, we **cannot close** this terminal or **kill** this process.*

3) **OpenCCTVServerStarter**

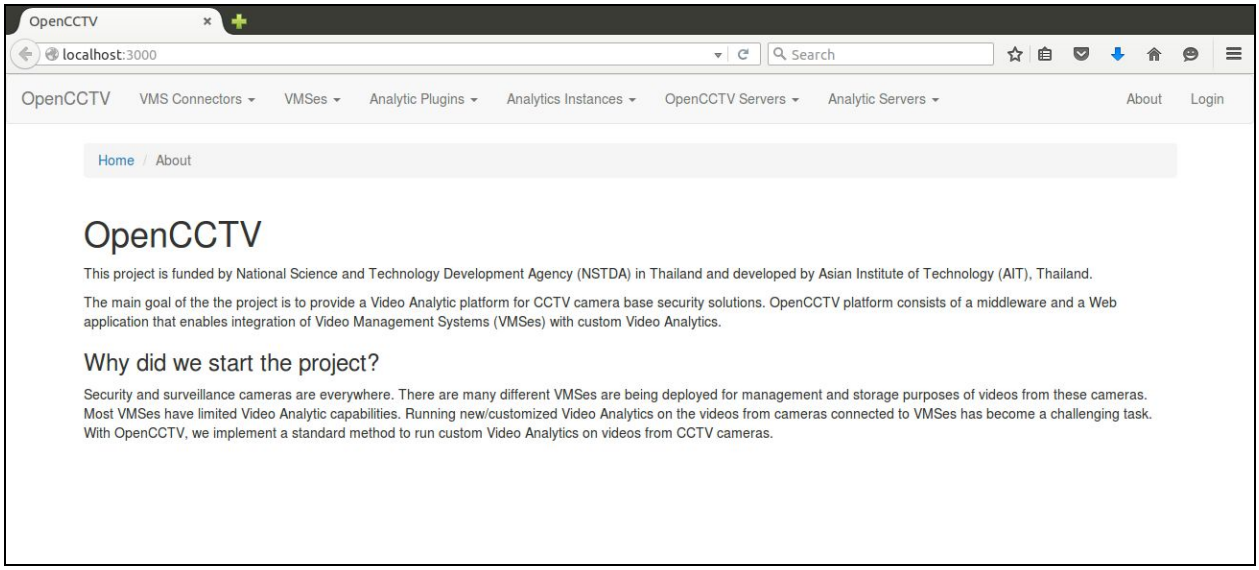
OpenCCTVServerStarter is used to start/stop OpenCCTVServer. To start this service, we will do the same process as we did for AnalyticStarter. User has to open a new terminal, go to OpenCCTVServerStarter directory, and then execute the following command:

```
./OpenCCTVServer_Starter
```



*As previously mentioned in OpenCCTVWeb, we **cannot close** this terminal or **kill** this process.*

Now, we can check our web application on web browser. User has to enter http://localhost:3000 in web browser and you should see the main web page of OpenCCTV.

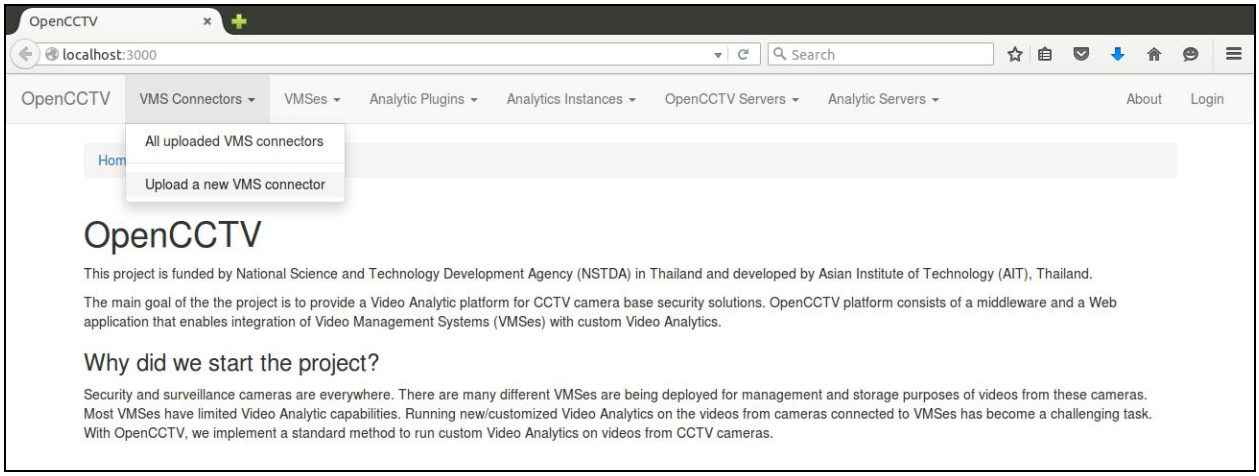


If you see this main web page, it means that your OpenCCTV is ready.

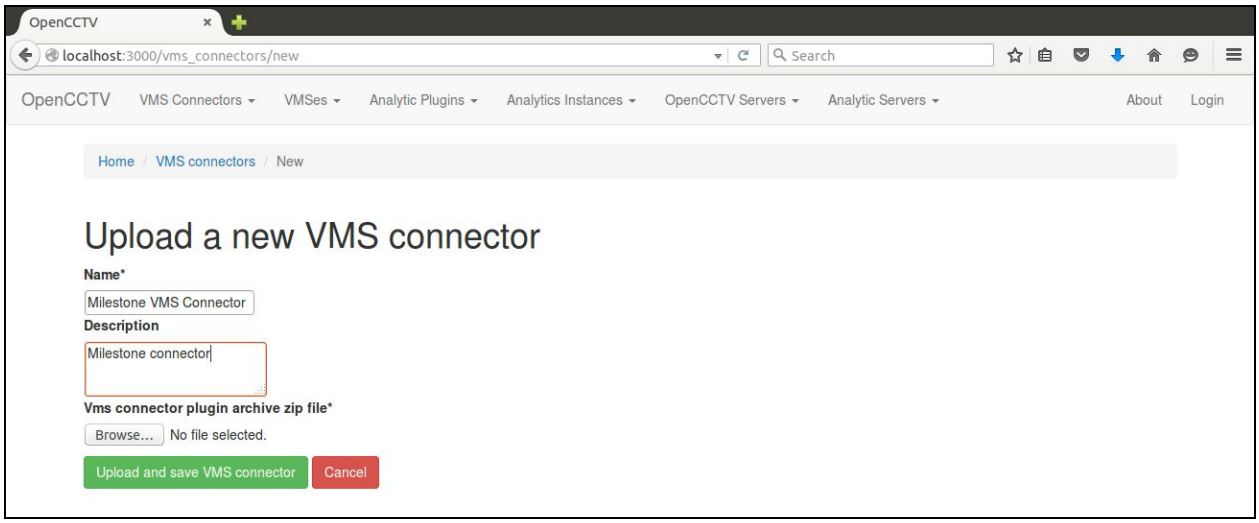
OpenCCTV Platform User Manual

1. Upload VMS Connector

1.1) Click at VMS Connectors->Upload a new VMS Connector



1.2) Fill a connector name and description

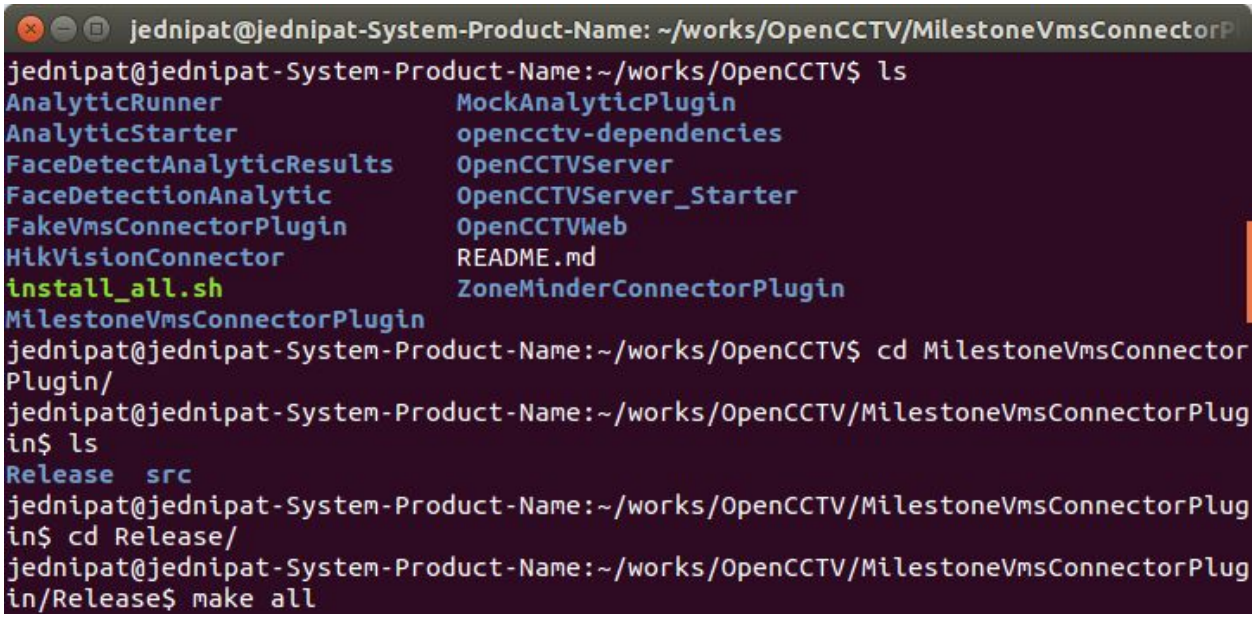


1.3) Compile and upload Milestone VMS connector

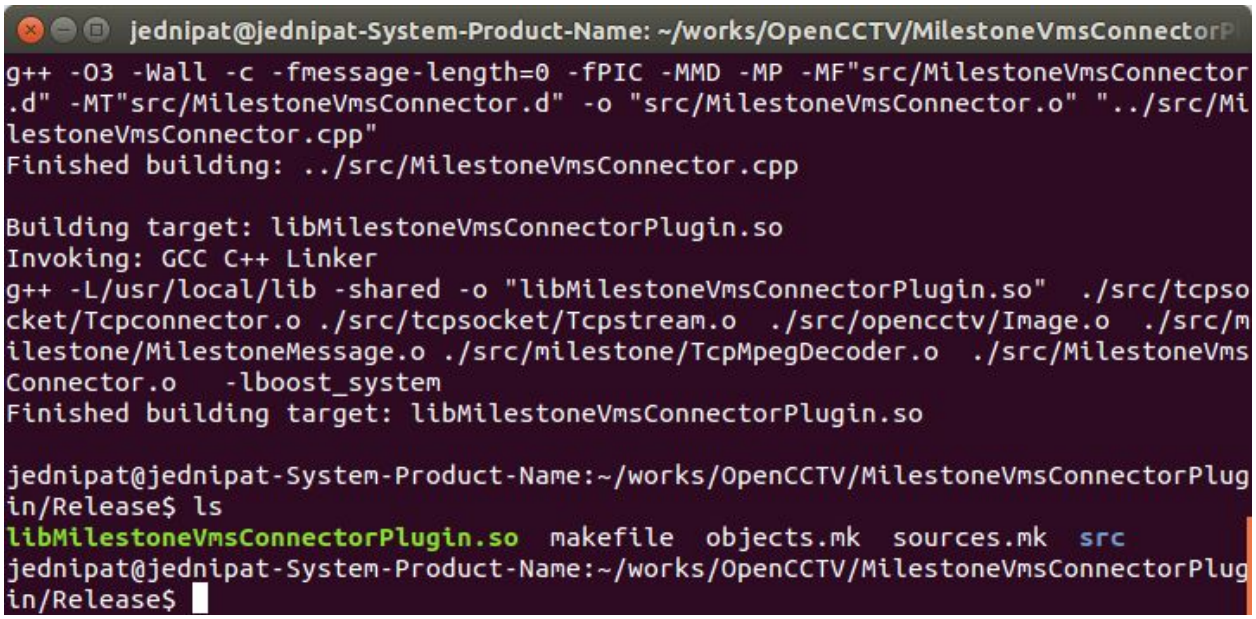
1.3.1) Compile Milestone VMS connector

To compile Milestone VMS connector, user has to go to a folder which is cloned from OpenCCTV Github repository and follow the steps below:

```
cd path_to_your_OpenCCTV
cd MilestoneVmsConnectorPlugin
cd Release
make all
```

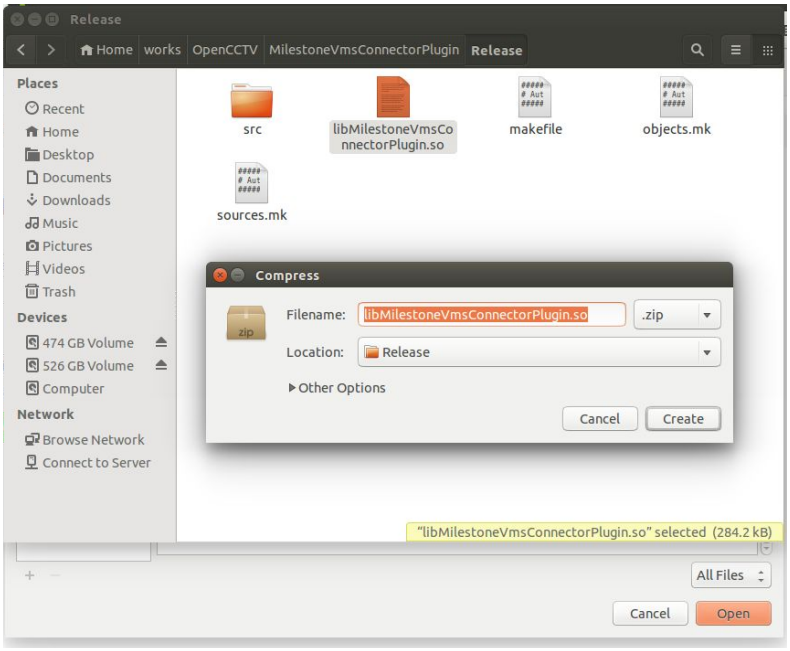


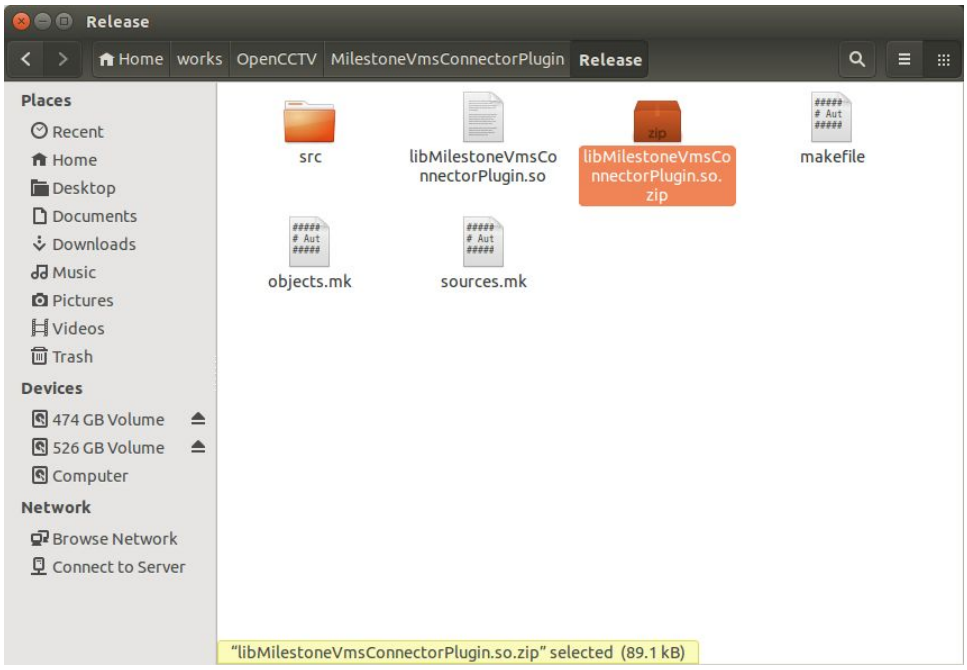
If the source code is compiled successfully, user will get **libMilestoneVmsConnectorPlugin.so** file.



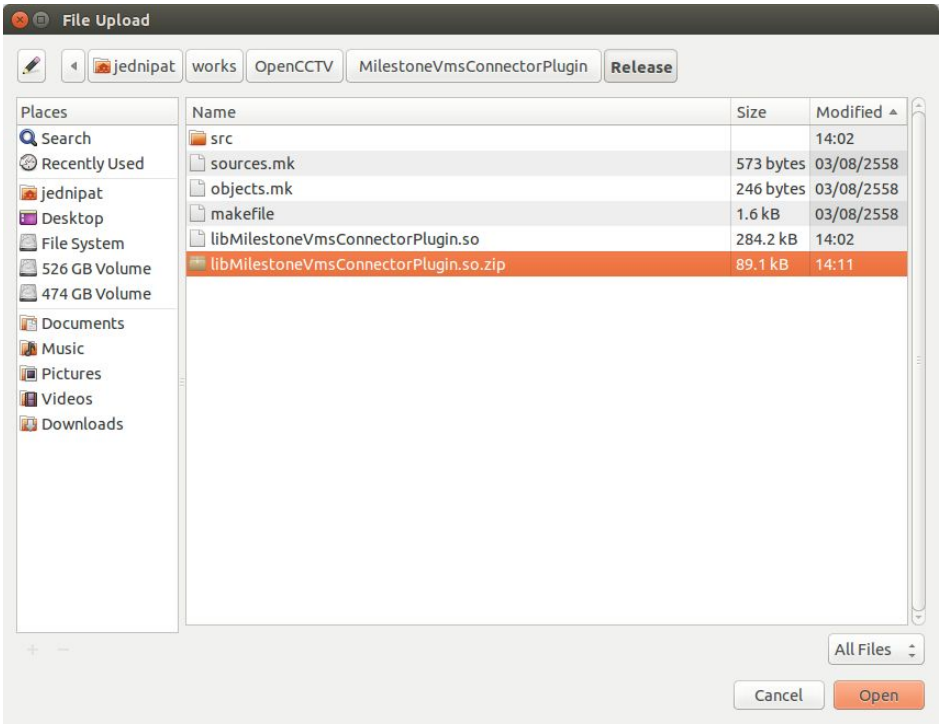
1.3.2) Create a zip file of VMS connector

Once we get libMilestoneVmsConnectorPlugin.so, user has to create a zip file of this file for uploading.

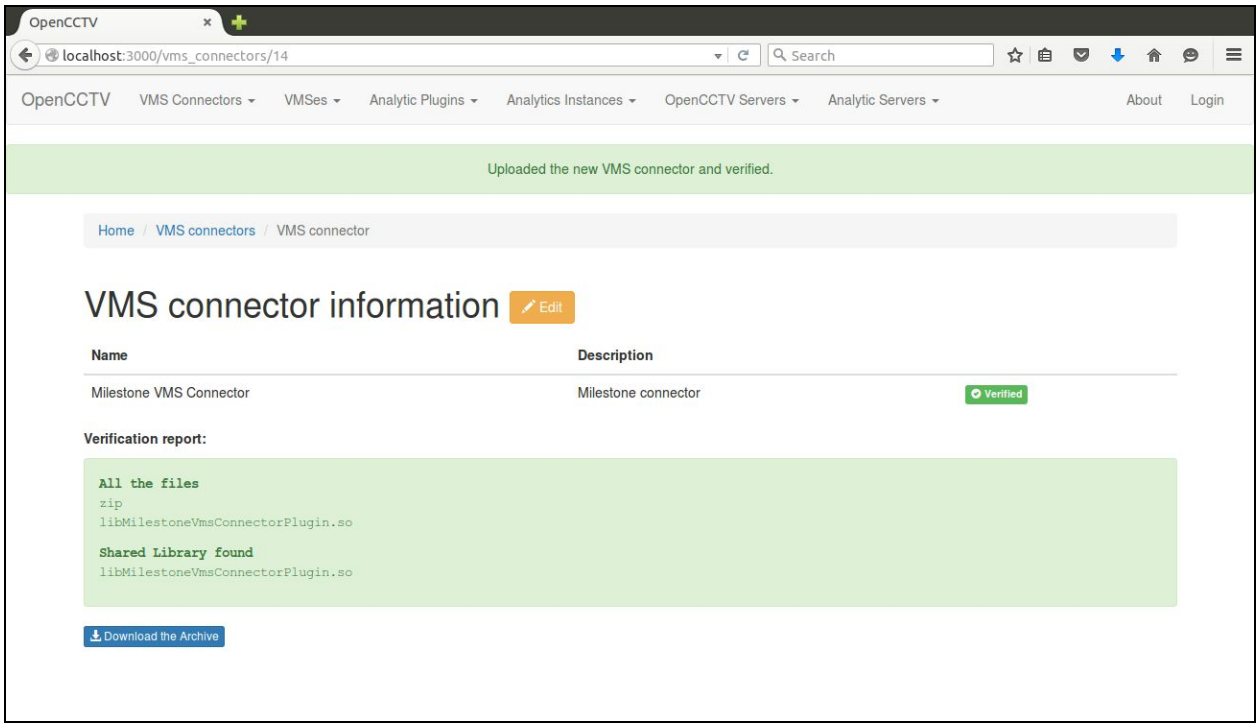




1.3.3) Select a file for uploading



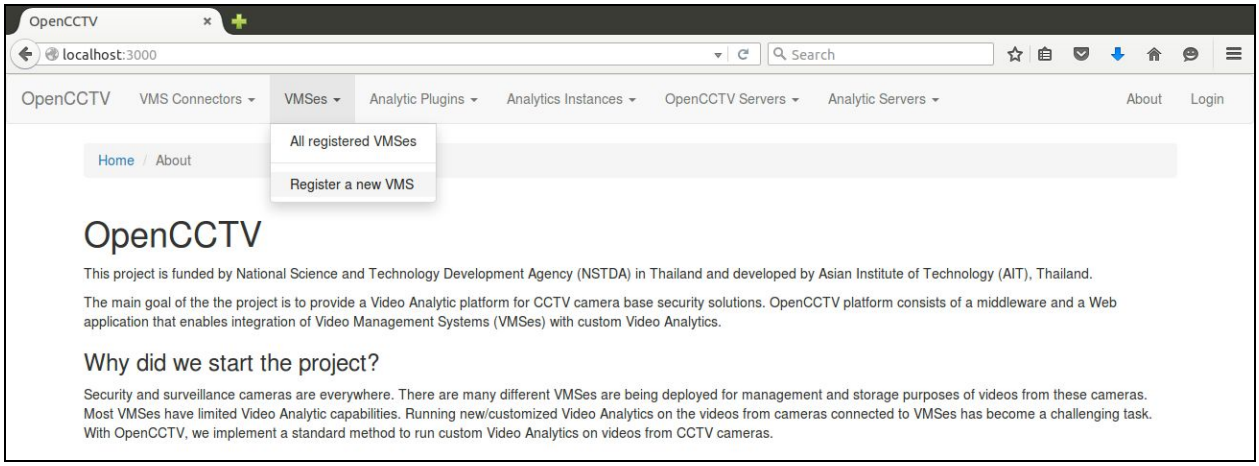
1.3.4) Upload and save VMS connector to server



After uploading, user should see a verification report. If a VMS connector is implemented correctly, the system will show a status “Verified”. This means that the uploaded VMS connector is ready for using.

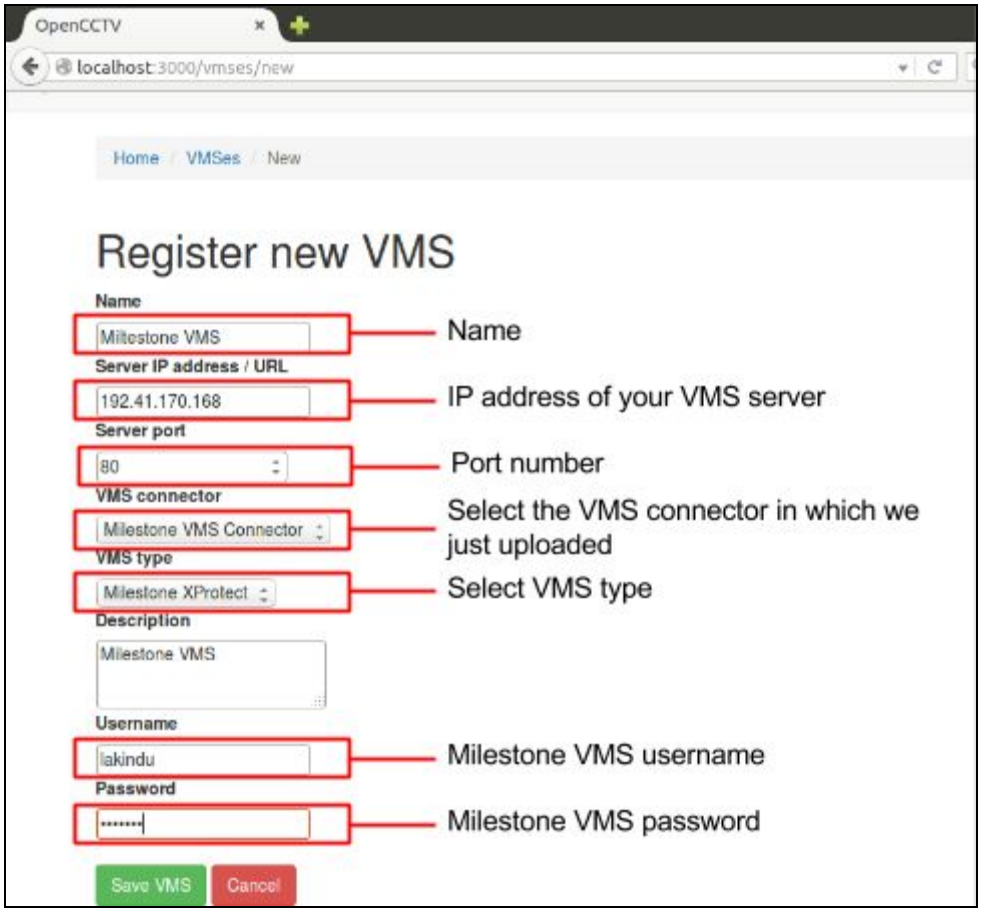
2. Register a VMS

2.1) Click at VMSes->Register a new VMS

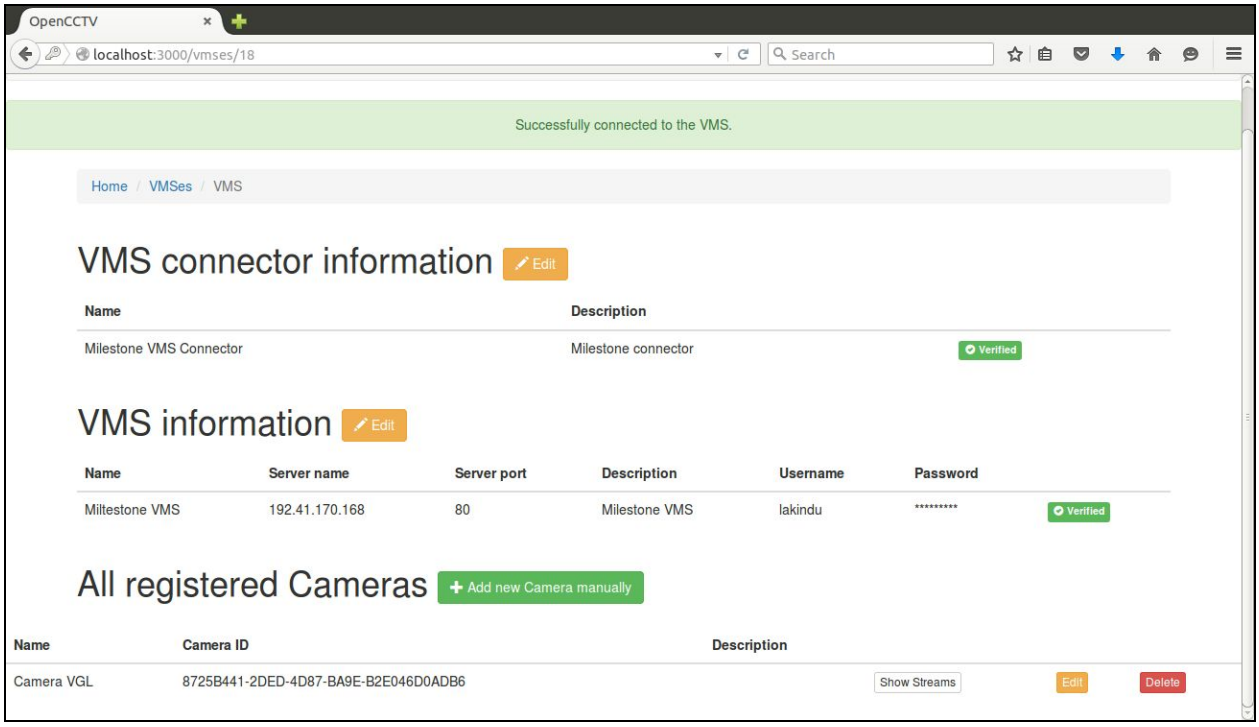


2.2) Register a new VMS

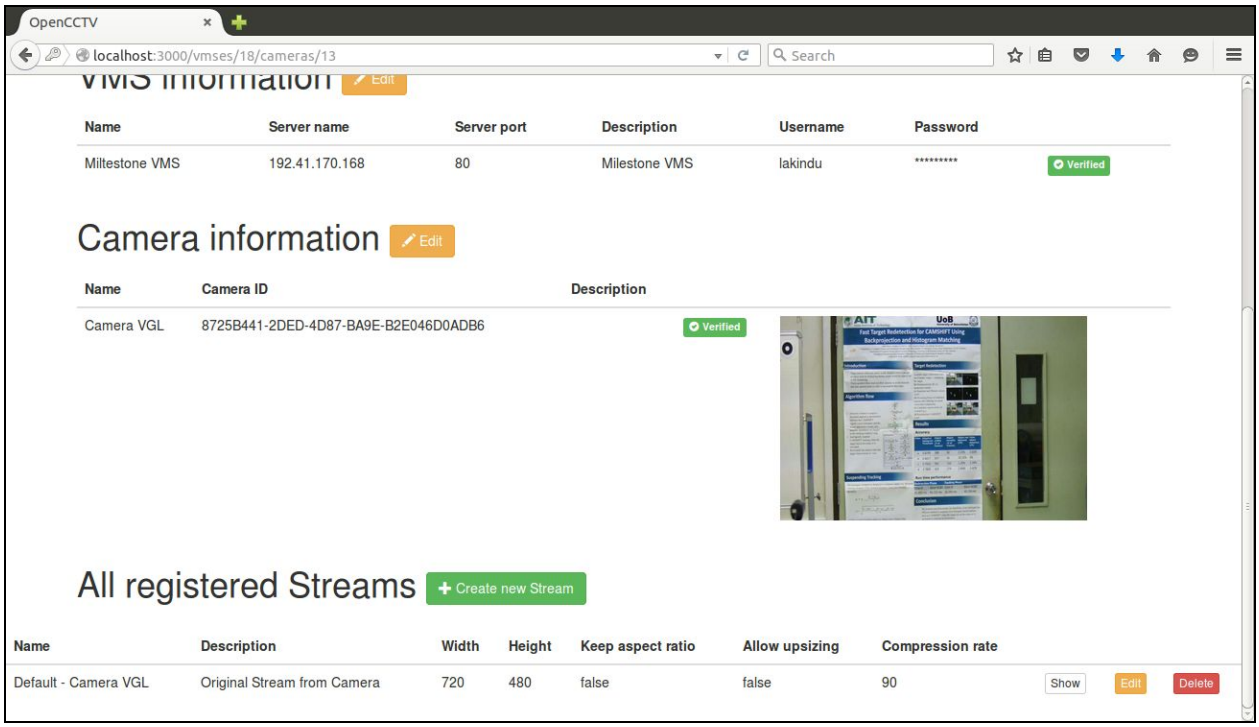
Before starting this process, please make sure that you have information about your VMS. In this example, we will use Milestone VMS as an example because we just uploaded the Milestone VMS connector in the previous section.



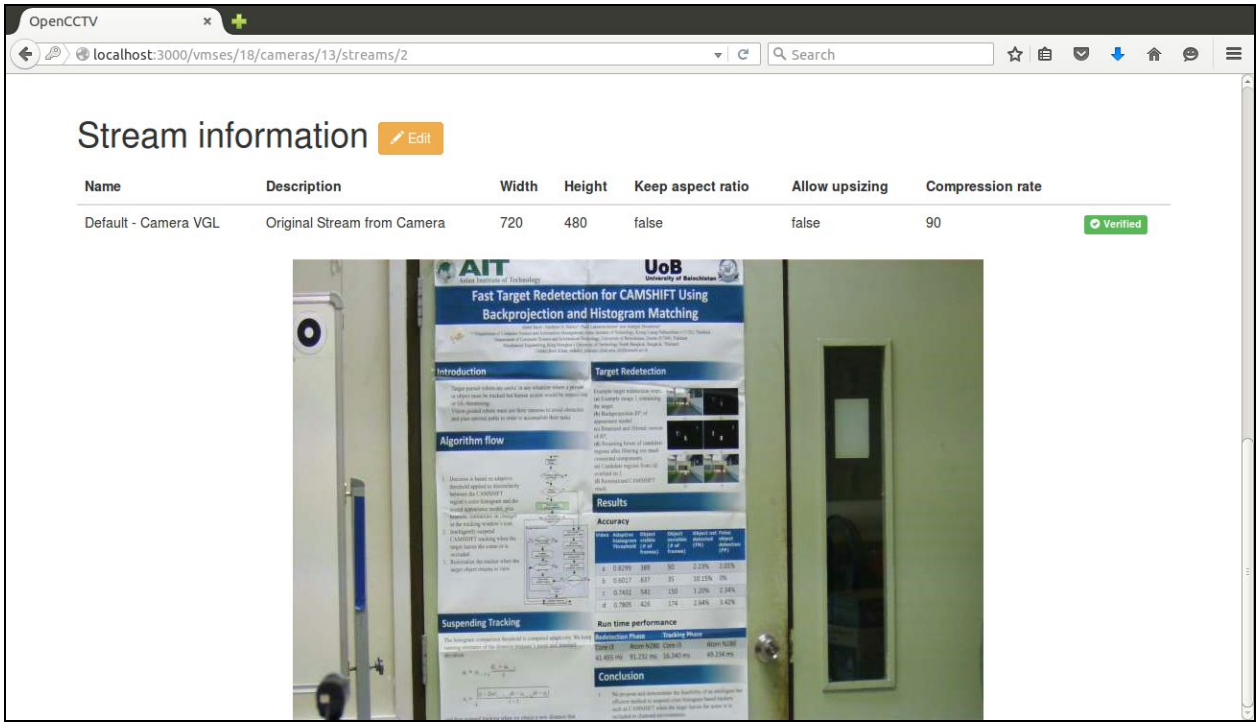
2.3) Check the VMS connector information



As shown in the figure above, after user registered a new VMS, the system will show the result of VMS connection and all registered cameras on the VMS. In this example, we connect only one IP camera to the Milestone VMS. Therefore, there is only one camera in the registered cameras list. User can click at “Show streams” to view the image from the camera.



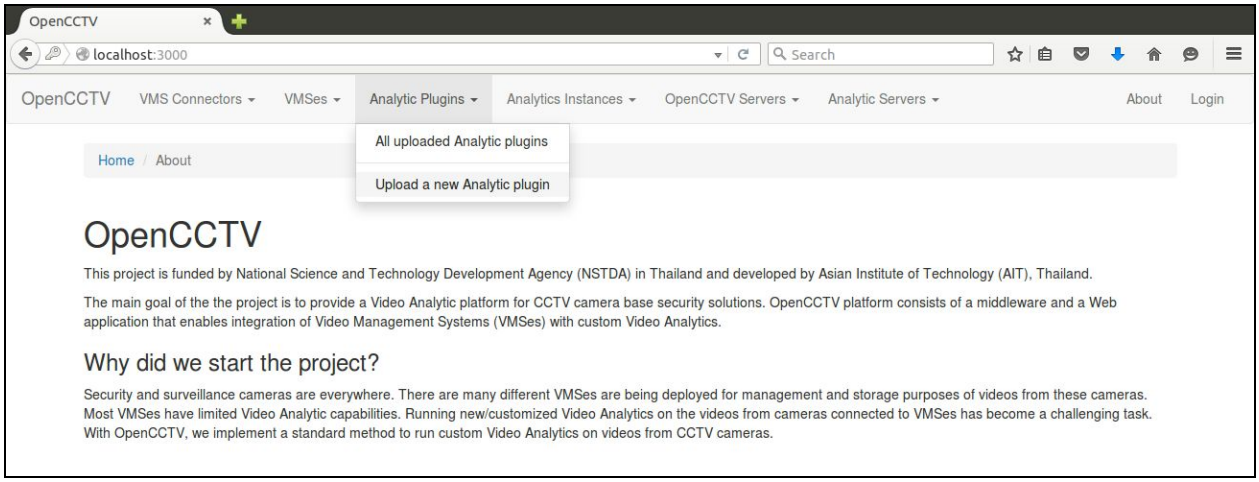
If the video stream is accessible from OpenCCTV, when user clicks at “Show” button in “All registered Streams”, user should see a single frame from the video stream as shown in figure below.



If you can see a single frame in stream information, you can proceed to next step.

3) Upload Analytic plugin

3.1) Click at Analytic Plugins->Upload a new Analytic plugin

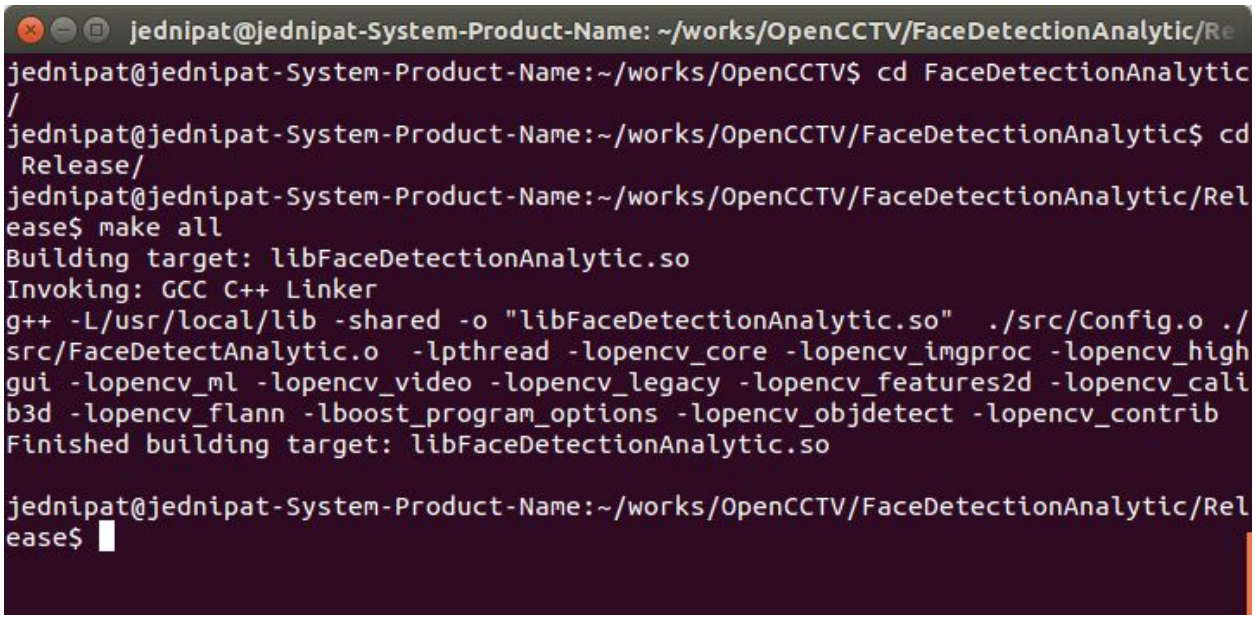


3.2) Compile and upload analytic plugin

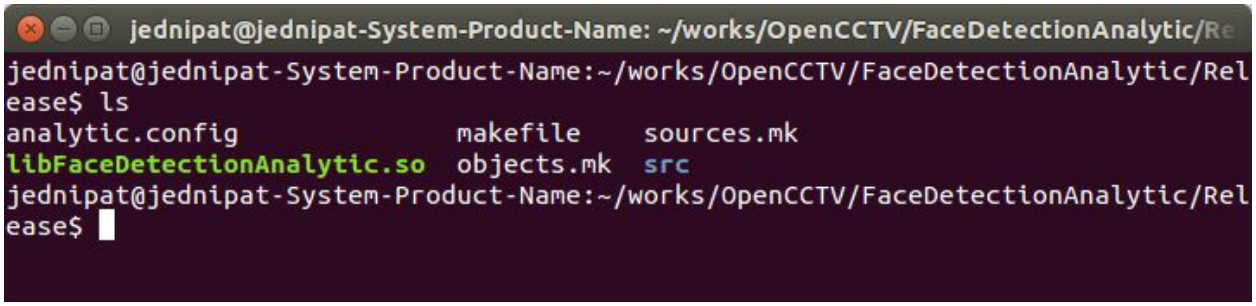
3.2.1) Compile analytic plugin

In this example, we will use a face detection analytic as an example. User has to go to a folder which is cloned from OpenCCTV Github repository and follow the steps below:

```
cd path_to_your_face_detection_analytic_plugin
cd Release
make all
```

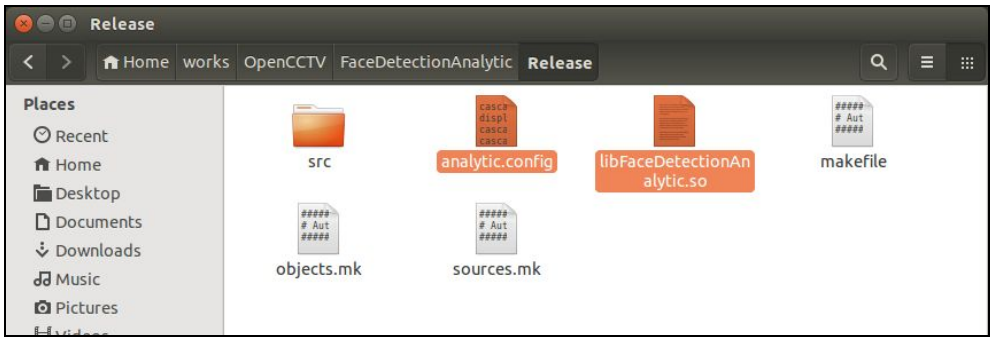


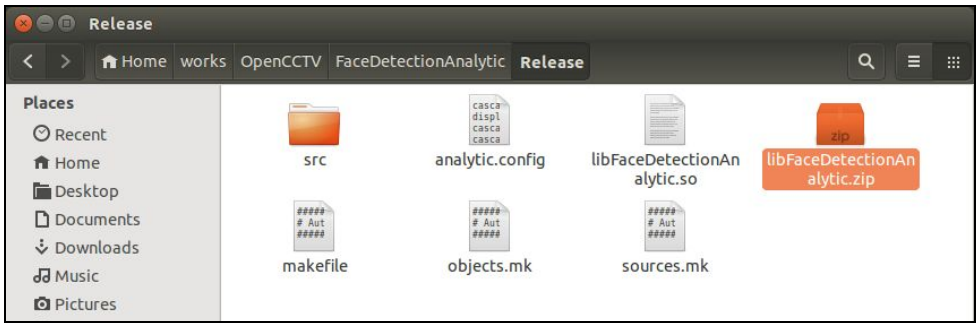
After compiling face detection analytic plugin, you will get libFaceDetectionAnalytic.so.



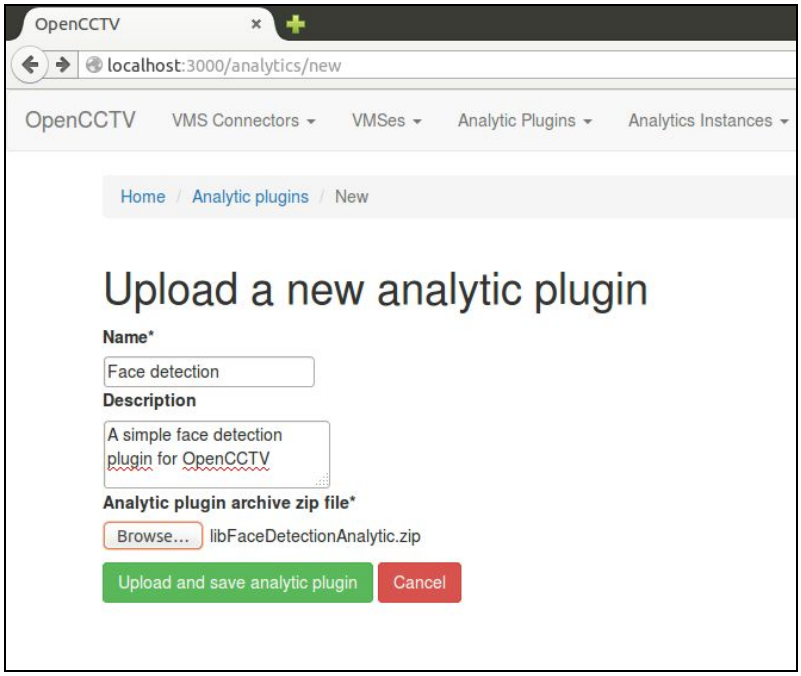
3.2.2) Create a zip file of libFaceDetectionAnalytic.so and upload to server

To create a zip file for face detection analytic, we have to include analytic.config into the zip file too. This analytic.config contains configuration data of the face detection. Therefore, we have to select analytic.config and libFaceDetectionAnalytic.so for creating a zip file.

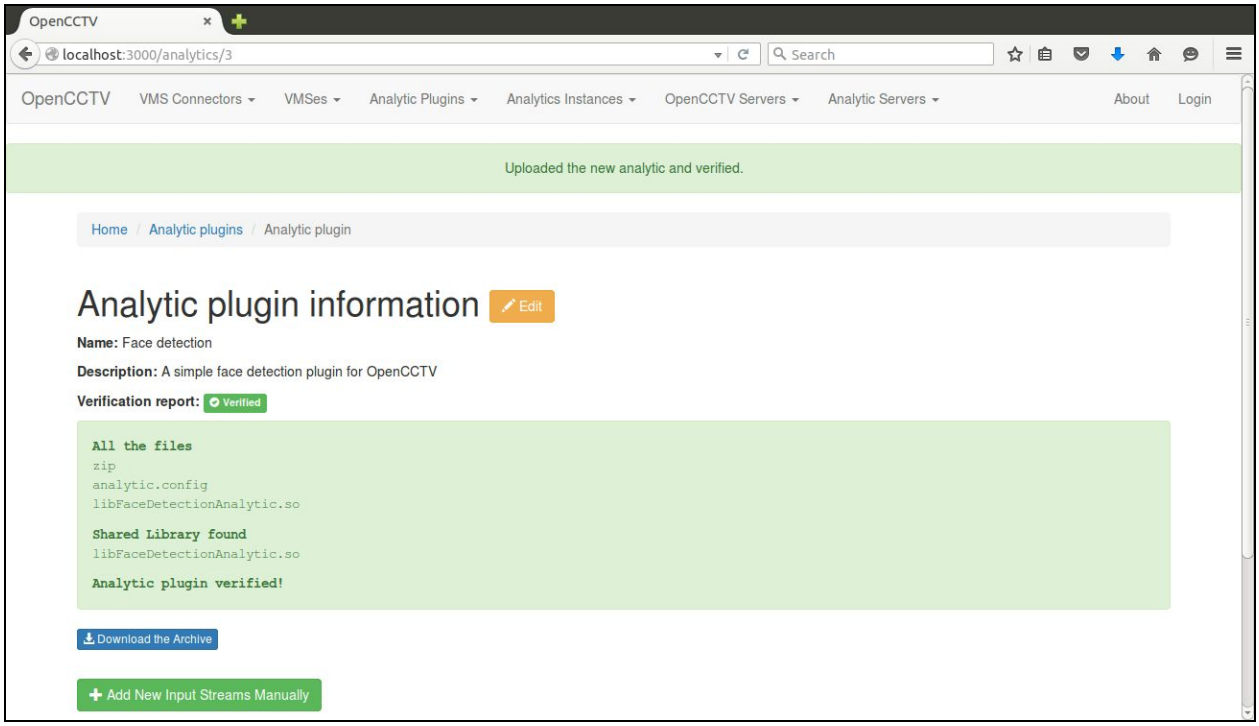




The libFaceDetectionAnalytic.zip will be used for uploading to the server.



3.3) Check the status of analytic plugin

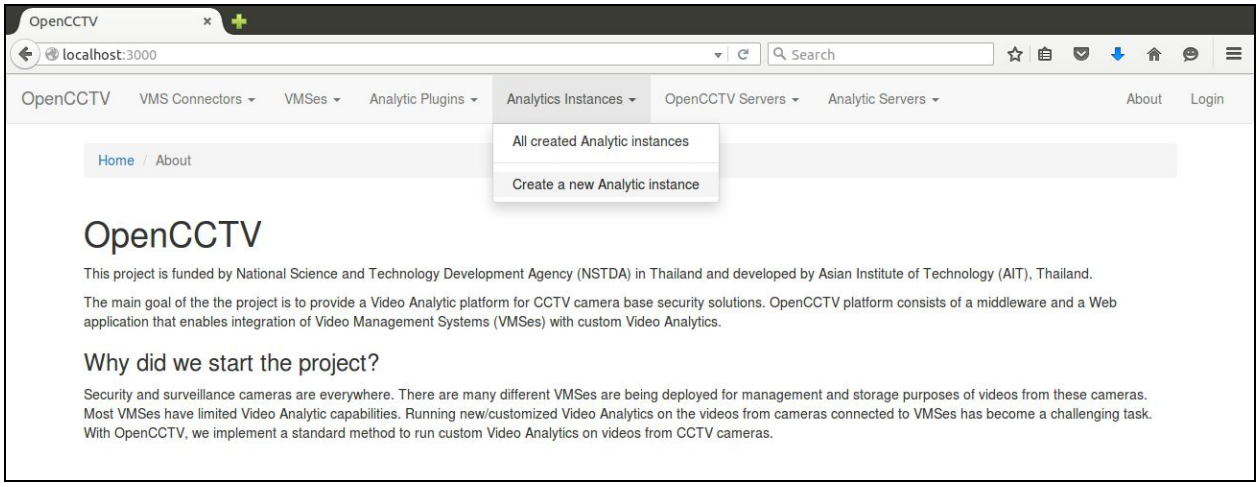


If the verification report status is “verified”, your analytic plugin is ready to use. User can proceed to next step.

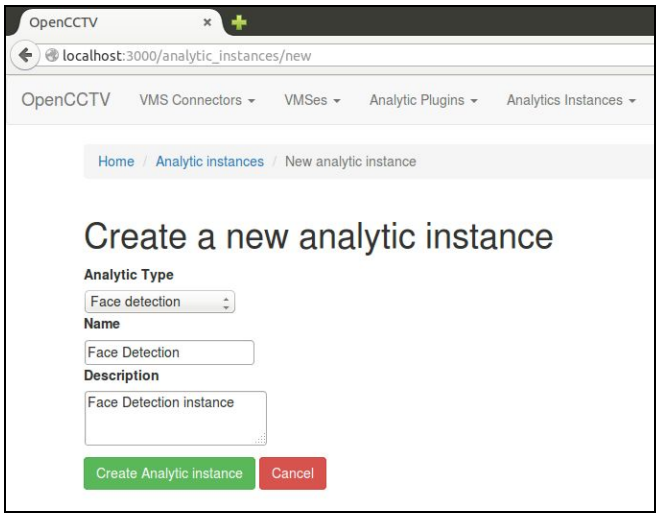
4) Create a new analytic instance

In the previous steps, we uploaded a Milestone VMS connector and a simple face detection analytic plugin. Now, we will create a new analytic instance.

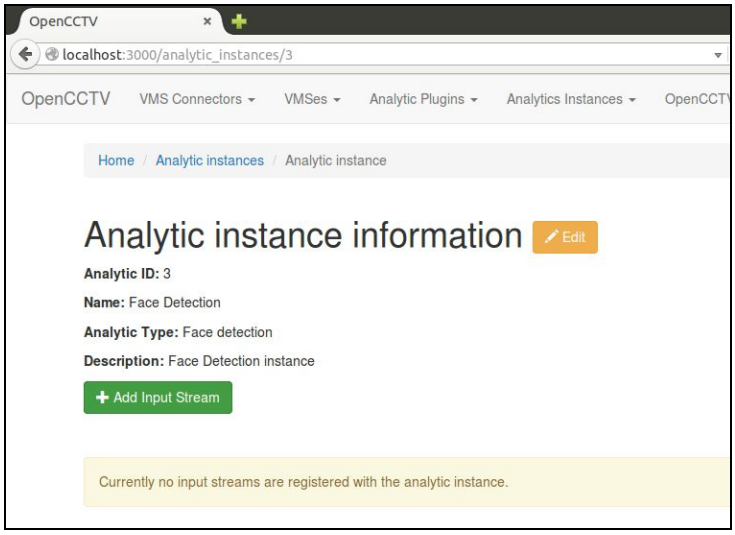
4.1) Click at Analytics Instances->Create a new Analytic instance



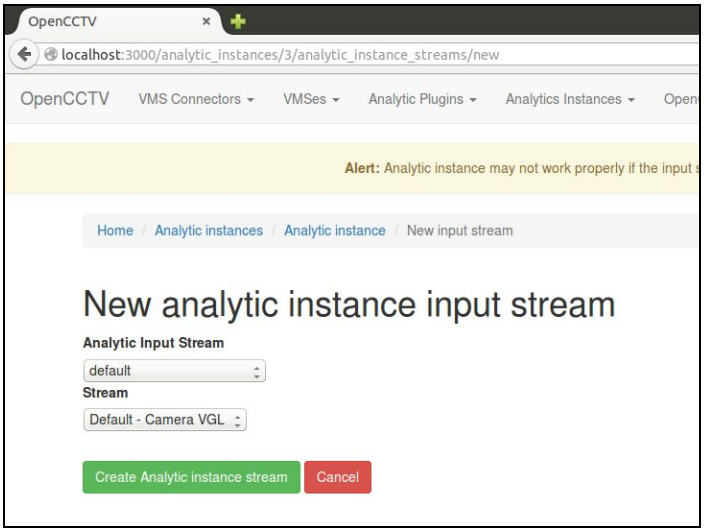
4.2) Select analytic plugin and create an analytic instance



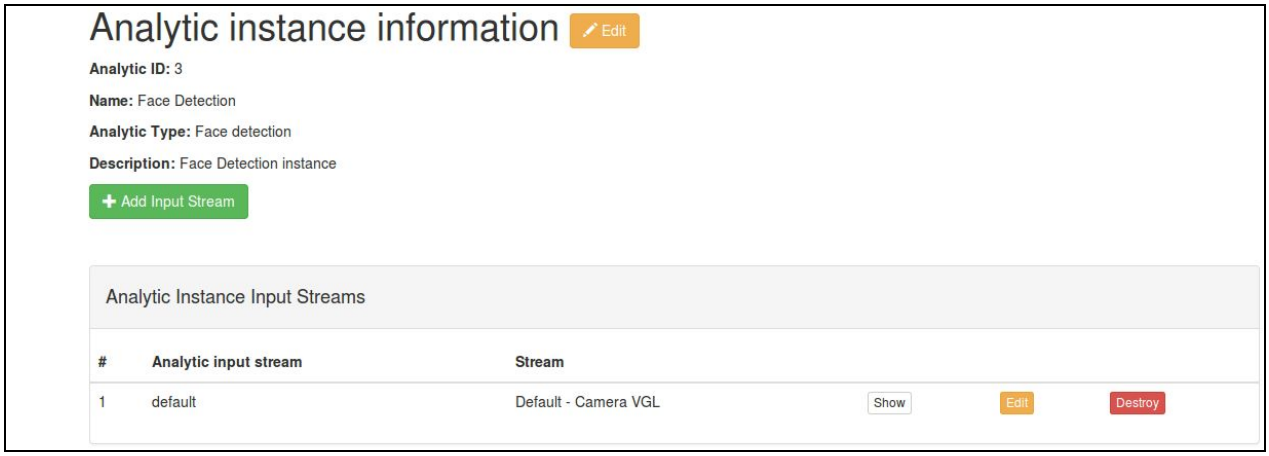
4.3) Add input stream for a new analytic instance by clicking at “Add Input Stream”



User has to select analytic input stream and input stream, then click at “Create Analytic instance stream” button.

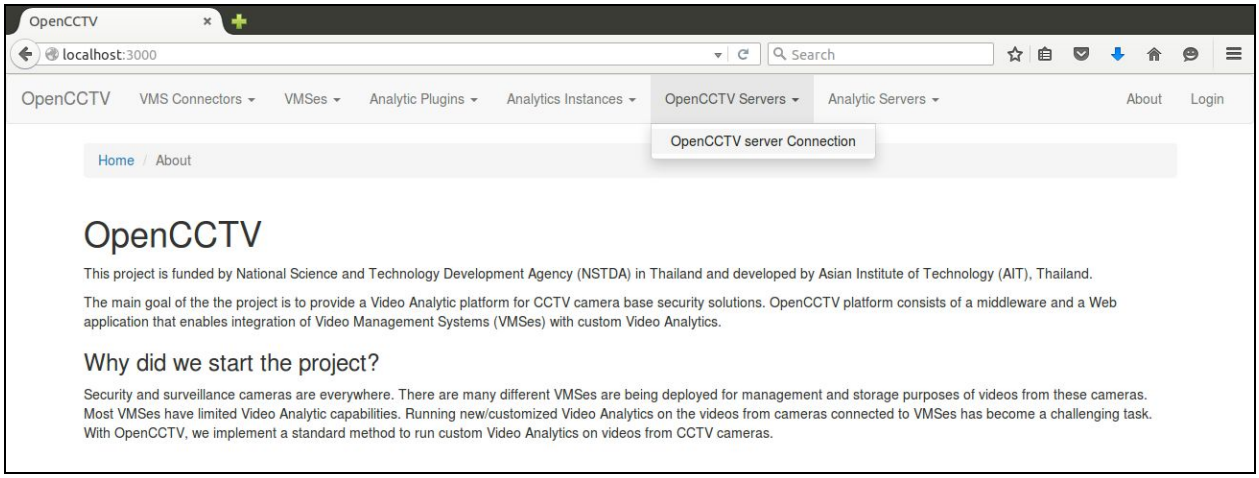


If you see the figure below, it means that we can create a new analytic instance and bind it to the specified video stream already. You can proceed to next step.

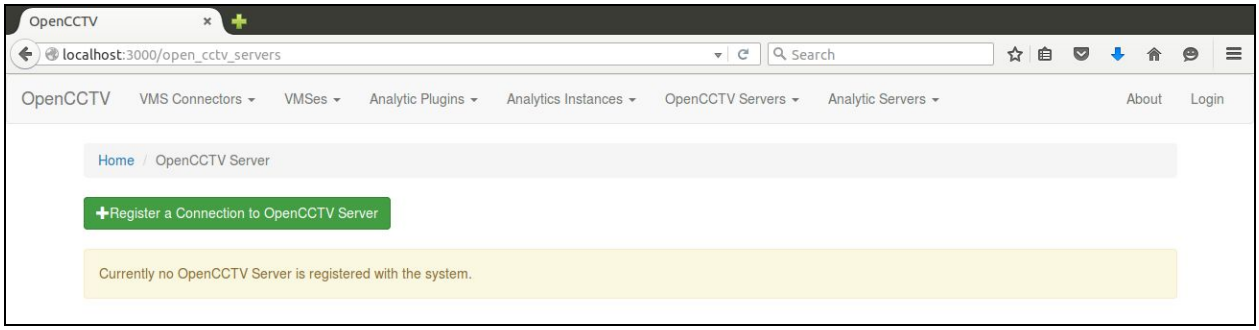


5) Create OpenCCTV Server

5.1) Click at OpenCCTV Servers->OpenCCTV server Connection

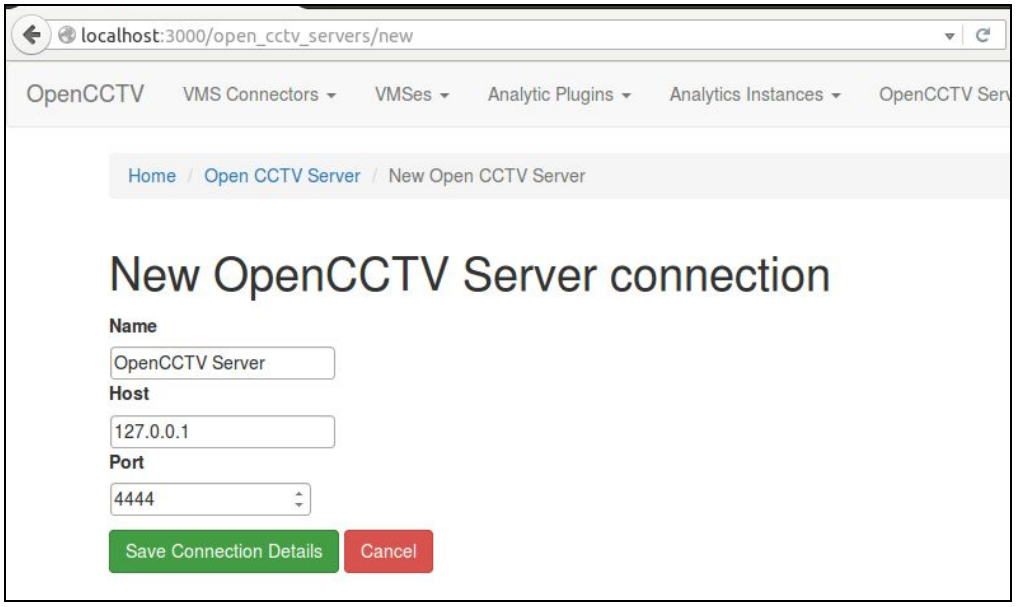


5.2) Click at “Register a Connection to OpenCCTVServer”

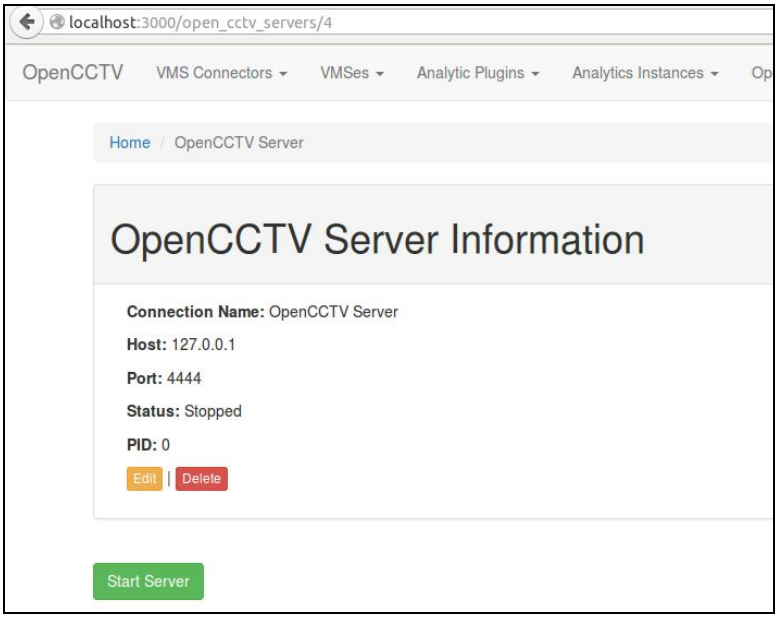


5.3) Fill the information about connection

User can use default value for creating this server connection, and then click at “Save Connection Details”.



5.4) Start OpenCCTV server by clicking at “Start Server”.



Once user clicks on a “Start Server” button, user should see one window showing a video stream from camera. This video stream window is displayed from our face detection analytic. In a real deployment, we can disable this video stream window. This video stream is fetched from Milestone VMS and then our analytic performs face detection on this video stream. The result (output) is written to the database. User may need another web application for displaying the output.