Service Level Agreement Contract

Technical Document

PIA Technical Review  
12/02/2018

Contents

[2. Introduction 3](#_Toc506369681)

[3. Ubuntu Server 17.10 Installation 3](#_Toc506369682)

[3.1 Update your server 3](#_Toc506369683)

[3.2 Configure IP address as static and check the gateway information 4](#_Toc506369684)

[3.3 Install Ubuntu desktop 5](#_Toc506369685)

[3.4 PIA –Back Installation 6](#_Toc506369686)

[3.5 Installation of POSTGRESQL 6](#_Toc506369687)

[3.5.1 Check version postgres 6](#_Toc506369688)

[3.5.2 check client version 7](#_Toc506369689)

[3.5.3 Install version postgres 7](#_Toc506369690)

[3.5.4 Check install postgres 7](#_Toc506369691)

[3.5.5 Install posgresql with Password (best practice) 8](#_Toc506369692)

[3.6 Roles creation 8](#_Toc506369693)

[3.6.1 Prod User & Role Creation (useradmin) 8](#_Toc506369694)

[4. RAILS 5.0.x - RUBY 2.3.x - mode.js 9](#_Toc506369695)

[4.1 GNUP Key 9](#_Toc506369696)

[4.2 RUBY 2.3.1 9](#_Toc506369697)

[4.3 RAILS 5.0.6 10](#_Toc506369698)

[4.4 Node.js 11](#_Toc506369699)

[4.4.1 Check the versions 11](#_Toc506369700)

[5. piaback : Installation 11](#_Toc506369701)

[5.1 Fill out the DB information 12](#_Toc506369702)

[5.2 Install all dependencies 12](#_Toc506369703)

[5.3 Fill out the secret keybase 12](#_Toc506369704)

[5.4 piaback : Configuration 12](#_Toc506369705)

[5.4.1 DEV (Same for test just change the Environment name) 12](#_Toc506369706)

[5.4.2 Prod 13](#_Toc506369707)

[5.5 piaback : Test the application 13](#_Toc506369708)

[6. PIA Front : Installation 14](#_Toc506369709)

[6.1 Install pre-requisite 14](#_Toc506369710)

[6.2 Build the Production Environment 14](#_Toc506369711)

[6.3 Run the Front Server 15](#_Toc506369712)

[7. Configure PIA tool 15](#_Toc506369713)

# Introduction

PIA Application is a suite of front and back end servers using Ruby on Rails and Angular JS.  
This application is an open source application that you can find with all relevant information in the below link: <https://github.com/LINCnil>

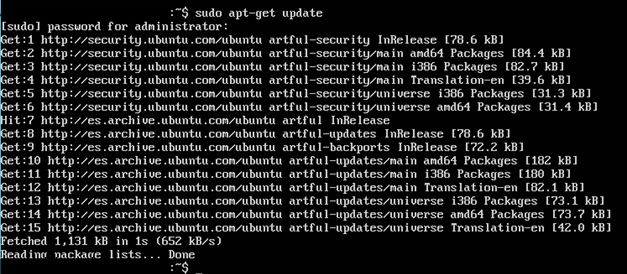
# Ubuntu Server 17.10 Installation

## Update your server



Update the package with the command

Sudo Apt-get update



## Configure IP address as static and check the gateway information

ifconfig -a | grep eth

To verify the IP address configuration of *eth0*, you can use the *ifconfig* command in the following manner.

ifconfig eth0

eth0 Link encap:Ethernet HWaddr 00:15:c5:4a:16:5a

inet addr:10.0.0.100 Bcast:10.0.0.255 Mask:255.255.255.0

inet6 addr: fe80::215:c5ff:fe4a:165a/64 Scope:Link

UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1

RX packets:466475604 errors:0 dropped:0 overruns:0 frame:0

TX packets:403172654 errors:0 dropped:0 overruns:0 carrier:0

collisions:0 txqueuelen:1000

RX bytes:2574778386 (2.5 GB) TX bytes:1618367329 (1.6 GB)

Interrupt:16

To configure a default gateway, you can use the *route* command in the following manner. Modify the default gateway address to match your network requirements.

sudo route add default gw 10.0.0.1 eth0

To verify your default gateway configuration, you can use the *route* command in the following manner.

route -n

Kernel IP routing table

Destination Gateway Genmask Flags Metric Ref Use Iface

10.0.0.0 0.0.0.0 255.255.255.0 U 1 0 0 eth0

0.0.0.0 10.0.0.1 0.0.0.0 UG 0 0 0 eth0

If you require DNS for your temporary network configuration, you can add DNS server IP addresses in the file /etc/resolv.conf. In general, editing /etc/resolv.conf directly is not recommanded, but this is a temporary and non-persistent configuration. The example below shows how to enter two DNS servers to /etc/resolv.conf, which should be changed to servers appropriate for your network. A more lengthy description of the proper persistent way to do DNS client configuration is in a following section.

nameserver 10.34.1X.X

nameserver 10.34.1X.X

If you no longer need this configuration and wish to purge all IP configuration from an interface, you can use the *ip* command with the flush option as shown below.

ip addr flush eth0

## Install Ubuntu desktop

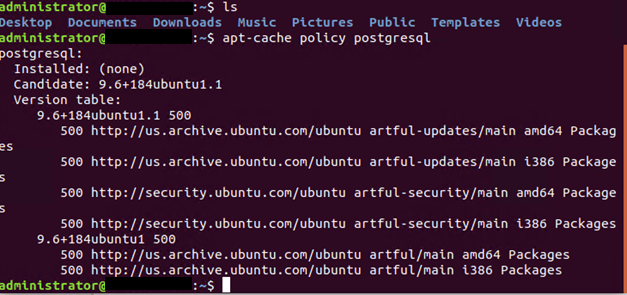
sudo apt-get install ubuntu-desktop

Restart the server



## PIA –Back Installation

Check where you are in the directory and check the postgresql install that you have



Useful Links

**# How-to : install pia-back**

<https://github.com/LINCnil/pia-back>

**# How-to : install postgres**

<https://help.ubuntu.com/community/PostgreSQL>

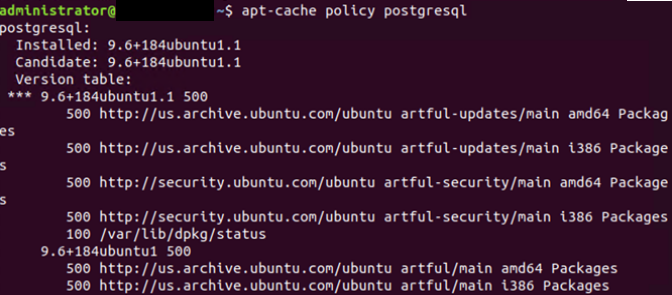
**# How-to : install rail 5.1 et ruby 2.3.1**

<http://rubyonrails.org/>

## Installation of POSTGRESQL

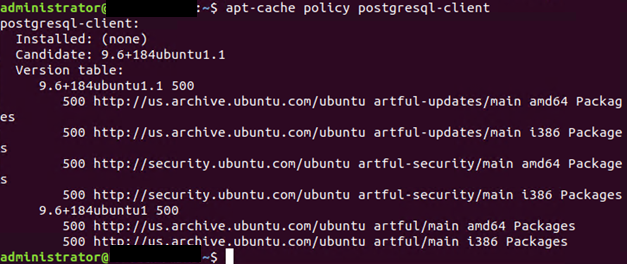
### Check version postgres

apt-cache policy postgresql



### check client version

apt-cache policy postgresql-client



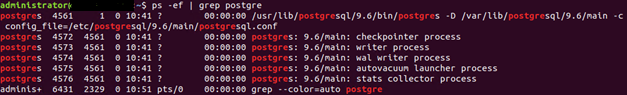
### Install version postgres

Sudo apt-get install postgresql postgresql-client –y

Sudo apt-get update && apt-get install -y procps

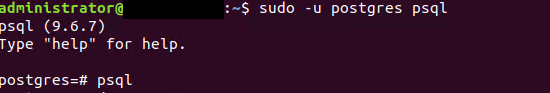
### Check install postgres

ps -ef | grep postgre



### Install posgresql with Password (best practice)

su – postgres



To exit the psql console just use the command \q

## Roles creation

Care : Depending of the installation you want to make (Dev/Test or Prod)

I recommend to create the Prod environment only on prod server and without Dev Test to avoid miss configuration.

For Dev/Test, I recommend to create it in dev test server without the production environment.

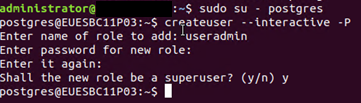
To make sure those environment are created properly, you need to adapt the database.yml that you will configure with the DB access you will create below

* Each users need to have DB creation access for each correspondent base « prod » « dev » and « test »

### Prod User & Role Creation (useradmin)

Sudo su - postgres

createuser --interactive –P



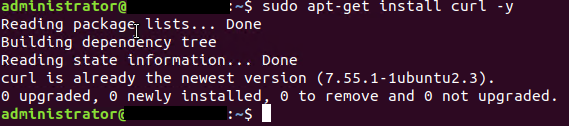
Super-user ? (y/n) : Y

Ctrl D to go out from this account

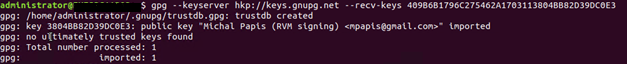
# **RAILS 5.0.x - RUBY 2.3.x - mode.js**

## GNUP Key

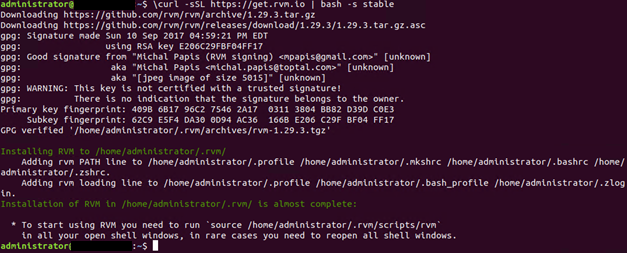
apt-get install curl –y



gpg --keyserver hkp://keys.gnupg.net --recv-keys409B6B1796C275462A1703113804BB82D39DC0E3

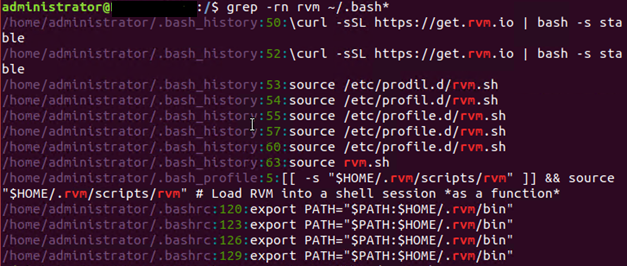


\curl -sSL https://get.rvm.io | bash -s stable



## **RUBY 2.3.1**

source /etc/profile.d/rvm.sh (if doesn’t work … check the source )



rvm install 2.3.1

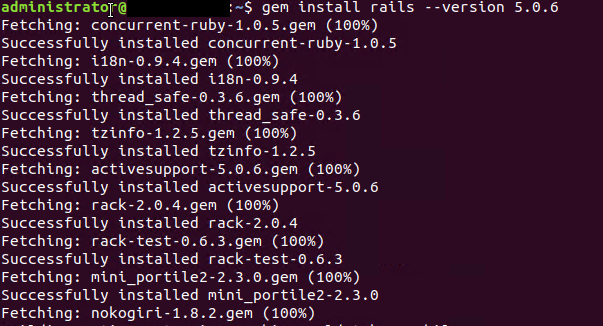
rvm use 2.3.1 --default



## RAILS 5.0.6

The last version is 5.1.4 but requierement is 5.0.x ...

gem install rails --version 5.0.6

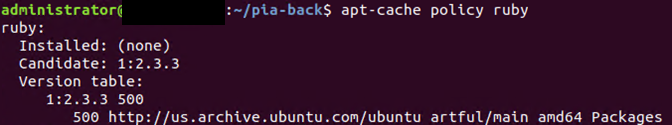


## Node.js

apt-get install nodejs -y

### Check the versions

apt-cache policy ruby



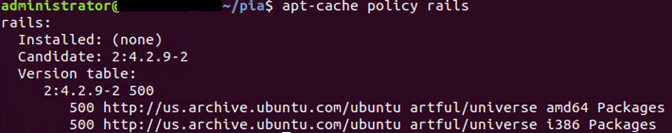
whereis ruby



ruby -v



apt-cache policy rails



whereis rails



# piaback : Installation

Run the following command as admin or with sudo

Sudo apt-get install git -y

git clone https://github.com/LINCnil/pia-back.git

cd pia-back

cp config/database.example.yml config/database.yml

## Fill out the DB information

*if you only have Prod env, just fill out Prod, Dev and test need to be fillout the same way*

vi /root/pia-back/config/database.yml

## Install all dependencies

cd piaback

Sudo apt-get install libpq-dev

Sudo bundle install

## Fill out the secret keybase

You need to get your secret key baserake to copy it inside application config

Sudo echo "SECRET\_KEY\_BASE: $(rake secret)" | sudo tee /root/pia-back/config/

application.yml

You can also putting the key manually as soon as we use the Ubuntu desktop. To get the key use the following command and copy past on the application.yml (pia-back/config)

RAILS\_ENV=production rake secret

## piaback : Configuration

### DEV (Same for test just change the Environment name)

#### Create database

bin/rake db:create

#### Create tables

bin/rake db:migrate

#### Create Environment

RAILS\_ENV=development bin/rake db:migrate

#### Run the application (test)

bin/rails s

#### Run the application (dev)

RAILS\_ENV=development bin/rails s

### Prod

#### Create database

RAILS\_ENV=production bin/rake db:create

#### Create tables

RAILS\_ENV=production bin/rake db:migrate

#### Run the application

RAILS\_ENV=production bin/rails s

## piaback : Test the application

Bin/rails s -b 10.34.10.253 -p 8080

If you use the URL you should see a PIA Icon like below :



# PIA Front : Installation

On home directory

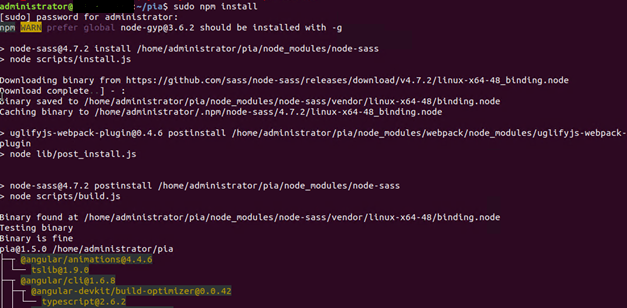
git clone https://github.com/LINCnil/pia.git

## **Install pre-requisite**

cd pia

Sudo apt install npm

Sudo npm install -g @angular/cli

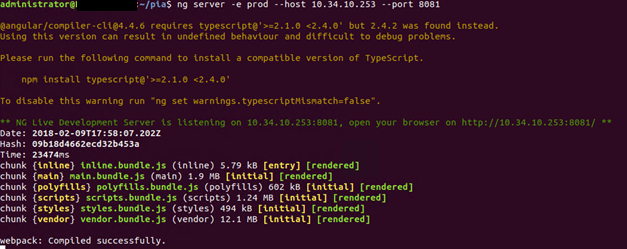


## Build the Production Environment

sudo ng build prod  

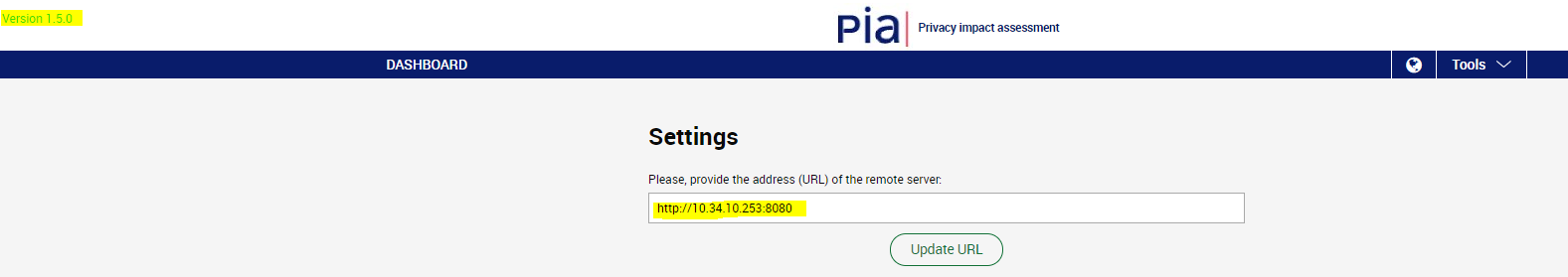

## Run the Front Server

Ng serve –e prod –host 10.34.10.XXX –port XXXX



# Configure PIA tool

Use the URL and Port you configure on the Front server, here it’s 10.34.10.XXX:XXXX



If you are running a prod version you should see like below the version, if not it should mention “Development” or “Tests”  
  
Put on Tools >> Settings the PIA Back URL that you configure on step 5.5.

If you have stand alone user of Pia App, you need to do the same to have every user using the Back End Database.

If you have any question please check the PIA Github.