SADMIN

Documentation

Table of content

Getting the SADMIN software

You can get the software in two differents ways ;

1. Cloning the sadmin git repository
2. Download the software file from sadmin.ca

Cloning the 'sadmin' repository.

- cd /opt
- git clone https://jadupl2@bitbucket.org/jadupl2/sadmin.git

Download the software

- Go to the download page http://wsadmin.maison.ca/www/download.php
- Download the latest version, copy it to /opt directory.
- cd /opt
- mkdir sadmin
- cd sadmin
- tar -xzf SADMIN_XX.XX_XX.tgz

Installing SADMIN client software

Easy installation steps

All the components of the SADMIN tools reside in one directory, we recommend using '**/opt/sadmin**' but you can install it in the directory of your choice. At least 1Gb of free space is suggested for the client installation.

The instructions below is assuming you have chosen to install it in '/opt/sadmin' directory. The directory you choose can either be a directory or a filesystem. The SADMIN **client** have been tested to work on Redhat, Fedora, CentOS, Debian, Ubuntu, Raspbian and Aix.

Create a filesystem or directory where you want SADMIN to reside

```
$ mkdir /opt/sadmin
```

Copy the latest version of SADMIN file you have downloaded in the chosen directory.

```
$ cp sadmin_xx.xx_YYYYMMDD.tgz /opt/sadmin
```

Change directory to the name you choose

```
$ cd /opt/sadmin
```

Untar the file

```
$ tar -xvzf sadmin_XX.XX_YYYYMMDD.tgz
```

Run the setup program and answer the questions.

```
$ sudo /opt/sadmin/setup/setup.sh
```

Log off and log back in

You are now ready to use the SADMIN Tools, follow instructions on the screen.

Running the setup script

```
# sudo /opt/sadmin/setup/setup.sh
```

- The setup program will ask some questions regarding your environment and will store your answers in the SADMIN configuration file ($SADMIN/cfg/sadmin.cfg). You can modify it later on if you want to.

- If there are missing packages on your system, the setup program will try to install them for you.

- An environment variable named 'SADMIN' containing the installation directory you have chosen, is critical for all the tools to work. To survive a reboot, a script (/etc/profile.d/sadmin.sh) and a file (/etc/environment) will be created or modify to make sure the environment variable 'SADMIN' is defined. The directory '$SADMIN/bin' and $SADMIN/usr/bin will be added to your PATH to ease the use of the SADMIN tools.

Setup script will create/modify these files to make sure that SADMIN environment is available and properly set. If you decided to install 'SADMIN'in /opt/sadmin these two files should look like the one below.

```
# cat /etc/environment
SADMIN=/opt/sadmin
```

```
# cat /etc/profile.d/sadmin.sh
#! /bin/sh
################################################################################
# Name      :     sadmin.sh
# Version   :     2.0
# Author    :     J.Duplessis
# Date      :     2018-04-01
# Requires  :     bash shell
# Synopsis  :     Executed when users login system wide & when system start
#                 Do not modify this, it could be overwritten by new release.
################################################################################
# SADMIN root Directory
SADMIN=/opt/sadmin
export SADMIN
#
# Add SADMIN bin and usr/bin to PATH
export PATH=$PATH:${SADMIN}/bin:${SADMIN}/usr/bin
```

Your comment are welcome

If you ever ran into problem while installing or running the SADMIN tools, please run the 'sadm_support_request.sh', attach the resulting log to an email with a description of your problem or question and sent to webadmin@sadmin.ca. We will get back to you as soon as possible.

The setup script explained

The setup process is divided in two main scripts. The first one verify some requirement and the second ask you some questions and setup your environment so that you have a pleasing experience using the SADMIN tools. No need to run the second script, when the first one finish, it will automatically start the second one.

When you start the first setup script, it will first verified if the command 'lsb_release"and "python3" is available. If not it will try to install them. The "lsb_release" is used to determine the Linux distribution and version you are running the setup script with. The "python3" package is needed because some of SADMIN scripts are written in Python3.

Example of first part of setup process output

```
SADMIN Pre-Installation Verification Version 1.5
---------------------------------------------------------------------------
Checking if 'lsb_release' is installed ...  Done
Checking if 'python3' is installed ...  Done
All Vérifications Pass ...
```

Sample of setup process second part

```
SADMIN Setup V2.8
---------------------------------------------------------------------------
SADMIN Environment variable is now set to /opt/sadmin
  - The line below is now in /etc/profile.d/sadmin.sh and in /etc/environment
  - SADMIN=/opt/sadmin
  - This will make sure 'SADMIN' environment variable is set upon reboot
  - Initial SADMIN configuration file (/opt/sadmin/cfg/sadmin.cfg) in place



----------
[SADM_HOST_TYPE]
Specify if this host is to become a SADMIN [C]lient or [S]erver.
Host will be a SADMIN [S]erver or a [C]lient [C] : c
[SADM_HOST_TYPE] set to 'C' in /opt/sadmin/cfg/sadmin.cfg

----------
[SADM_CIE_NAME]
This name that will appear in the heading of the web interface
and on some report and email that SADMIN produce.
Enter your Company Name : Batcave
[SADM_CIE_NAME] set to 'Batcave' in /opt/sadmin/cfg/sadmin.cfg

----------
```

```
[SADM_MAIL_ADDR]
This field specify the email address of the system administrator.
It is used by SADMIN Tools to send various email and alert.
Enter System Administrator Email : batman@batcave.com
[SADM_MAIL_ADDR] set to 'batman@batcave.com' in /opt/sadmin/cfg/sadmin.cfg

----------
[SADM_MAIL_TYPE]
Default option for sending email after a script is finish.
Can be overridden by changing SADM_MAIL_TYPE in SADMIN section of your script.
[0] = Never send email.
[1] = Only send the log when script finish with error (Default)
[2] = Only send the log when script finish with success.
[3] = Always send the script log when script finish.
Enter default email type [1] : 1
[SADM_MAIL_TYPE] set to '1' in /opt/sadmin/cfg/sadmin.cfg

----------
[SADM_DOMAIN]
Enter the default Domain Name to use when you will be adding a new server.
Default domain name [batcave.com] : batcave.com
[SADM_DOMAIN] set to 'maison.ca' in /opt/sadmin/cfg/sadmin.cfg

----------
[SADM_SERVER]
Enter the name of the SADMIN Server - (MUST be a fully qualified domain name)
This name can't be an alias in the DNS, it must be the real name of the server.
It MUST be the result of the 'hostname' command on the SADMIN server.
Enter SADMIN (FQDN) server name : batserver.batcave.com
Validating server name ...
[SADM_SERVER] set to batserver.batcave.com' in /opt/sadmin/cfg/sadmin.cfg

----------
[SADM_MAX_LOGLINE]
Every time a script (python or bash script) that use SADMIN Tools is run,
it produce a log, in the dedicated directory ${SADMIN}/log.

      - The name of the Log file is "[HOSTNAME]_[NAME_OF_SCRIPT].log".
      - Log file are cumulative by default, but it can be overriden if you want.
      - This number indicate the maximum of lines you want to keep in your log.
      - This help keeping our log to a reasonnable size.

Maximum number of lines in LOG file [1000] : Press [ENTER] for 1000 lines
[SADM_MAX_LOGLINE] set to '1000' in /opt/sadmin/cfg/sadmin.cfg
```

```
[SADM_MAX_RCHLINE]
When a (python or shell) script using the SADMIN tools start and ends,
it record the date/time and the ending status of your script in what we call a
RCH file ([R]eturn [C]ode [H]istory] file).

 - The RCH file name are "[HOSTNAME]_[NAME_OF_SCRIPT].rch"
 - They are located in the directory ${SADM_BASE_DIR}/dat/rch.
 - This number represent the maximum number of lines that each RCH file can contain.

A value of 100 lines is recommended.
This will keep an history of 50 days and is the default value.
Maximum number of lines in RCH file [100] : Press [ENTER] for 100 lines
[SADM_MAX_RCHLINE] set to '100' in /opt/sadmin/cfg/sadmin.cfg

----------
[SADM_GROUP]
This is the primary user group that have access to all SADMIN directories.
      - All files in the $SADMIN directories are own by this group.
      - If others users need to access or used scripts located in $SADMIN,
      they must be part of this group.
      - The SADMIN default user group is 'sadmin'.

Enter SADMIN User Group [sadmin] : Press [ENTER] for using 'sadmin'
Group sadmin is an existing group
[SADM_GROUP] set to 'sadmin' in /opt/sadmin/cfg/sadmin.cfg

----------
[SADM_USER]
Main SADMIN user that have access to all SADMIN directories and files.
      - This user will be assign to the SADMIN user group you just specified.
      - The SADMIN default user name is 'sadmin'.

Enter the default user name [sadmin] : Press [ENTER] for using 'sadmin'
User sadmin is an existing user
Add user sadmin to group sadmin
[SADM_USER] set to 'sadmin' in /opt/sadmin/cfg/sadmin.cfg
```

This section of the setup script verify the requirement for running the SADMIN client. If one of them is missing it will install it.

- If you're installing on Debian, Raspbian, Ubuntu there should be not problem all of these requirements can be downloaded from their standard repository.
- For Redhat, CentOS and Fedora we may need to access the EPEL repository to install some packages, so it will be configure automatically.

```
--------------------
Checking SADMIN Client Package requirement

Running apt-get update... Done
Checking for nmon ...  Ok
Checking for sudo ...  Ok
Checking for bc ...  Ok
Checking for lshw ...  Ok
Checking for parted ...  Ok
Checking for gawk ...  Ok
Checking for libdatetime-perl libwww-perl ...  Ok
Checking for util-linux ...  Ok
Checking for dmidecode ...  Ok
Checking for ethtool ...  Ok
Checking for openssh-client ...  Ok
Checking for util-linux ...  Ok
Checking for facter ...  Ok
Checking for net-tools ...  Ok
Checking for lsb-release ...  Ok
Checking for perl-base ...  Ok
Checking for mailutils ...  Ok
Checking for python pip3 command ... Done
Installing python3 PyMySQL module (pip3 install PyMySQL) ...  Done

--------------------
```

These two automatic steps, will ensure that the user 'sadmin' (or the one you selected earlier), will have 'sudo' right. The crontab file will make sure that the System Monitor (**${SADMIN}/bin/sadm_sysmon.pl**) run every 6 minutes and that the end of day tasks are executed (**/opt/sadmin/bin/sadm_client_sunset.sh**).

```

      Creating 'sadmin' user sudo file
       - Creating SADMIN sudo file (/etc/sudoers.d/033_sadmin-nopasswd)
       - Permission on sudo file changed successfully
       - Ownership of sudo file changed successfully

      --------------------
      Creating SADMIN client crontab file (/etc/cron.d/sadm_client)
       - Client Crontab Permission changed successfully
       - Ownership of client crontab changed successfully
```

Running client daily script for the first time

```
--------------------
Run SADM scripts to feed Database and Web Interface

Running '/opt/sadmin/bin/sadm_create_sysinfo.sh' script ...  Done
Running '/opt/sadmin/bin/sadm_client_housekeeping.sh' script ...  Done
Running '/opt/sadmin/bin/sadm_dr_savefs.sh' script ...  Done
Running '/opt/sadmin/bin/sadm_cfg2html.sh' script ...  Done
Running '/opt/sadmin/bin/sadm_sysmon.pl' script ...  Done
```

End of the setup script

```
SADMIN TOOLS - VERSION sadmin_0.86_20180624 - Successfully Installed
============================================================================
You need to logout and log back in before using SADMIN Tools,
or type the following command (The dot and the space are important)
. /etc/profile.d/sadmin.sh
This will make sure SADMIN environment variable is define.
============================================================================

CREATE YOUR OWN SCRIPT USING SADMIN LIBRARIES
To create your own script using the SADMIN tools, you may want to take a look
at the templates, run them and view their code.
  - bash shell script      : /opt/sadmin/bin/sadm_template.sh
  - python script          : /opt/sadmin/bin/sadm_template.py

For example, to create your own shell script, copy '/opt/sadmin/bin/sadm_template.sh'
to a name of your choice 'your_script.sh', modify it to your need, run it and see the
results.
============================================================================

VIEW SADMIN FUNCTIONS IN ACTION AND LEARN HOW TO USE THEM BY RUNNING :
  - /opt/sadmin/bin/sadmlib_std_demo.py
  - /opt/sadmin/bin/sadmlib_std_demo.sh

============================================================================
ENJOY !!
```

Installing SADMIN Server

```
Script started on Thu 28 Jun 2018 09:58:36 AM EDT
root@idebian9:/opt# ls -l
total 20536
-rw-r--r-- 1 root root 21027593 Jun 28 09:56 sadmin_0.86_20180628.tgz
root@idebian9:/opt#
root@idebian9:/opt#
root@idebian9:/opt# mkdir sadmin

root@idebian9:/opt# cd sadmin

root@idebian9:/opt/sadmin# tar -zvxf ../sadmin_0.86_20180628.tgz

root@idebian9:/opt/sadmin# ls -l
total 88
drwxrwxr-x  2 601 601  4096 Jun 24 12:54 bin
drwxrwxr-x  2 601 601  4096 Jun 28 09:54 cfg
drwxrwxr-x  8 601 601  4096 Jun 28 09:54 dat
drwxrwxr-x  3 601 601  4096 Jun  5 11:13 doc
drwxrwxr-x  3 601 601  4096 Jun  6 11:28 lib
-r--r--r--  1 601 601 35141 Jun 28 09:54 LICENSE
drwxrwxr-x  2 601 601  4096 Jun 28 09:54 log
drwxrwxr-x  6 601 601  4096 Jun 24 11:00 pkg
-rw-rw-r--  1 601 601  3292 Jun 28 09:54 README.md
drwxrwxr-x  6 601 601  4096 Jun 28 09:54 setup
drwxrwxr-x  2 601 601  4096 Jun 23 09:06 sys
drwxrwxr-x  2 601 601  4096 Jun 28 09:54 tmp
drwxrwxr-x  6 601 601  4096 Jun 28 09:54 usr
drwxrwxr-x 13 601 601  4096 Jun 28 09:54 www

root@idebian9:/opt/sadmin# cd setup

root@idebian9:/opt/sadmin/setup# ./setup.sh


 ____    _    ____  __  __ ___ _   _
/ ___|  / \  |  _ \|  \/  |_ _| \ | |
\___ \ / _ \ | | | | |\/| || ||  \| |
 ___) / ___ \| |_| | |  | || || |\  |
|____/_/   \_\____/|_|  |_|___|_| \_|

SADMIN Pre-Installation Verification Version 1.5
---------------------------------------------------------------------------
Checking if 'lsb_release' is installed ...  Done
Checking if 'python3' is installed ...  Done
All Verifications Pass ...


SADMIN Setup V2.9
---------------------------------------------------------------------------
```

```
Enter directory path where your install SADMIN : /opt/sadmin
SADMIN Environment variable is now set to /opt/sadmin
  - The line below is now in /etc/profile.d/sadmin.sh and in /etc/environment
  - SADMIN=/opt/sadmin
  - This will make sure 'SADMIN' environment variable is set upon reboot
  - Initial SADMIN configuration file (/opt/sadmin/cfg/sadmin.cfg) in place


----------
[SADM_HOST_TYPE]
Specify if this host is to become a SADMIN [C]lient or [S]erver.
Host will be a SADMIN [S]erver or a [C]lient [C] : s
[SADM_HOST_TYPE] set to 'S' in /opt/sadmin/cfg/sadmin.cfg

----------
```

```
SADMIN TOOLS - VERSION sadmin_0.86_20180628 - Successfully Installed
===========================================================================
You need to logout and log back in before using SADMIN Tools,
or type the following command (The dot and the space are important)
. /etc/profile.d/sadmin.sh
This will make sure SADMIN environment variable is define.
===========================================================================

USE THE WEB INTERFACE TO ADMINISTRATE YOUR LINUX SERVER FARM

The Web interface is available at :
http://sadmin.maison.ca or http://idebian9.maison.ca
  - For http://sadmin.maison.ca to work, 'sadmin.maison.ca' must be define in your DNS
or /etc/hosts file.
  - Use it to add, update and delete server in your server farm.
  - View performance graph of your servers up to two years in the past.
  - If you want, you can automatically update your server O/S at the time and day you
scheduled.
  - Have server configuration on hand, usefull in case of a Disaster Recovery.
  - View your servers farm subnet utilization and see what IP are free to use.
  - There's still a lot more to come.
===========================================================================

CREATE YOUR OWN SCRIPT USING SADMIN LIBRARIES

To create your own script using the SADMIN tools, you may want to take a look
at the templates, run them and view their code.
  - bash shell script   : /opt/sadmin/bin/sadm_template.sh
  - python script       : /opt/sadmin/bin/sadm_template.py

For example, to create your own shell script :
  # copy /opt/sadmin/bin/sadm_template.sh /opt/sadmin/usr/bin/newscript.sh
modify it to your need, run it and see the results.
===========================================================================

VIEW SADMIN FUNCTIONS IN ACTION AND LEARN HOW TO USE THEM BY RUNNING :

  - /opt/sadmin/bin/sadmlib_std_demo.sh
  - /opt/sadmin/bin/sadmlib_std_demo.py.
===========================================================================

USE THE SADMIN WRAPPER TO RUN YOUR EXISTING SCRIPT

  - # $SADMIN/bin/sadm_wrapper.sh $SADMIN/usr/bin/yourscript.sh

===========================================================================
ENJOY !!
```

Updating SADMIN software

The SADMIN can be updated to the latest version using the “sadm_update_version.sh” script. This script will update part of your SADMIN installation directory that you should not have to changed.

The update process will not changed anything within theses directories :

- usr (Your scripts and data)
- sys (Startup/Shutdown scripts) directories.
- dat (system data files)
- log
- tmp

The update process may changed content of these directories :

- bin
- lib
- cfg (Only dot prefix files)
- doc
- pkg
- setup
- www

Starting the update script :

```
$ $SADMIN/bin/sadm_update_version.sh
```

When a new version of updater is present

This first thing the SADMIN updater does, is to verify if a new version of the updater is present in the new release file (tgz). If there is an update, the updater will get upgraded and restarted using this new version. When that situation happen the output will look similar to the one below.

```
SADMIN Updater - sadm_updater.sh - Version 2.4

Rollback Directory is /opt/sadmin/setup/update/2018_06_24

Enter location of the new SADMIN version tgz file : /opt/sadmin_0.86_20180624.tgz
Creating working directory /tmp/sadmin_20180624
Untar /opt/sadmin_0.86_20180624.tgz into /tmp/sadmin_20180624

There is a new version of the updater ...
We will install the new version and restart it.
Please wait while the script is restarting ...
```

Running with some scripts to update

This is an example of an update when a new release of SADMIN bash template (sadmin_template.sh) is release and when a script have been accidently deleted from the system (sadm_template.py).
Notice that is this example, instead of entering the name of new release file name interactively, we use the switch ‘-f’ to specified the new SADMIN release file.

```
SADMIN Updater - sadm_updater.sh - Version 2.5

Rollback Directory is /opt/sadmin/setup/update/2018_06_24

Enter location of the new SADMIN version tgz file : /opt/sadmin_0.86_20180624.tgz
Creating working directory /tmp/sadmin_20180624
Untar /opt/sadmin_0.86_20180624.tgz into /tmp/sadmin_20180624

Your current version is  : sadmin_0.86_20180623
Your updating to version : sadmin_0.86_20180624

Update Information
 - Anything under /opt/sadmin/usr, /opt/sadmin/sys will not be touch.
 - Your configuration files in /opt/sadmin/cfg will not be modify.
 - Scripts or files you have created won't be modified or deleted.
 - For SADMIN scripts (sadm*.py, sadm*.sh, sadm*.php, ...) :
       - If you haven't change them, they may be updated, if needed.
       - If you have change them, you will be asked what to do for each of them.
 - Only files in /opt/sadmin will be changed, not elsewhere.
 - Before proceeding, you should have a backup of /opt/sadmin

Proceed with the update [y,n] ? y
```

In the example below, the new version update trigger 3 events :

- The new version included a new version of “sadm_template.sh”
- On the SADMIN client the user modified “sadm_backup.sh”
- The new version include a new script called “demo2.sh”

```
================================================================================
Installing new file /opt/sadmin/bin/demo2.sh ...

================================================================================

File : "bin/sadm_backup.sh"
One of these conditions happened since last update of this file.
   1) Change were made to the file.
   2) You didn't update the file at the last update.
   3) The file was deleted from the system.
   4) This is a new file (or have moved in dir. structure).
Current file will be moved to RollBack Directory before updating it.

Want to [S]kip or [U]pdate this file [U] ? u
Saving /opt/sadmin/bin/sadm_backup.sh to Rollback Directory.
Updating file /opt/sadmin/bin/sadm_backup.sh ...

================================================================================
Saving /opt/sadmin/bin/sadm_template.sh to Rollback Directory.
Updating file /opt/sadmin/bin/sadm_template.sh ...

---------------
Updating MD5 Version files
Update completed Sun Jun 24 11:50:43 EDT 2018
```

File changed are kept in the Release Rollback directory

- Every files replace in $SADMIN by the updater and copy to this directory before the update take place.
- The rollback directory can be use to recover a file that was updated by mistake.
- This directory is under the subdirectory "pkg/sadm_update" and as the date at which the updater was ran.

```
root@raspi3/opt/sadmin/setup/update# ls -lR
.:
total 4
drwxrwxr-x 3 sadmin sadmin 4096 Jun 24 11:42 2018_06_24

./2018_06_24:
total 4
drwxrwxr-x 2 sadmin sadmin 4096 Jun 24 11:50 bin

./2018_06_24/bin:
total 64
-rwxr-xr-- 1 root root 43842 Jun 24 11:58 sadm_backup.sh
-rwxr-xr-- 1 root root 18916 Jun 24 11:58 sadm_template.sh
root@raspi3/opt/sadmin/setup/update#
```

Running update in batch mode

```
 root@nomad /opt/sadmin/bin# sadm_updater.sh -a -f /opt/sadmin_0.86_20180624.tgz

SADMIN Updater - sadm_updater.sh - Version 2.5

Rollback Directory is /opt/sadmin/setup/update/2018_06_24
Creating working directory /tmp/sadmin_20180624
Untar /opt/sadmin_0.86_20180624.tgz into /tmp/sadmin_20180624

Your current version is  : sadmin_0.86_20180624
Your updating to version : sadmin_0.86_20180624

================================================================================
Installing new file /opt/sadmin/bin/demo3.sh ...

================================================================================
Saving /opt/sadmin/bin/sadm_backup.sh to Rollback Directory.
Updating file /opt/sadmin/bin/sadm_backup.sh ...

================================================================================
Saving /opt/sadmin/bin/sadm_template.sh to Rollback Directory.
Updating file /opt/sadmin/bin/sadm_template.sh ...

---------------
Updating MD5 Version files
Update completed Sun Jun 24 11:58:58 EDT 2018
root@nomad /opt/sadmin/bin#
```

Adding a SADMIN client

Add client to web interface

Automating SSH from server to client

Every day SADMIN client produce performance data (via nmon), information that may be used for disaster recovery situation, monitoring reports, start scheduled O/S update, scripts results (log and rch files) that inform us about the status of our system. To accomplish this, the SADMIN server need to have root access to client via ssh.
To automate the ssh access and to do it in a safely and secure manner we will use the 'public-key authentication'. So this automated access will be only be possible from the SADMIN server to the client. So if any systems tries to access one of your SADMIN client via ' root' , they will get the '`Permission denied`' message.
In the example below, we have a SADMIN server called ' holmes.maison.ca' and we just installed SADMIN client on a system called "raspi5.maison.ca". We will now automate the ssh login from 'holmes.maison.ca' to 'raspi5.maison.ca' .

Before we change anything let's try to logon to system 'raspi5' using ssh. Since this is the first time we are trying to access that server from 'holmes.maison.ca' , it ask us a confirmation. After answering 'yes' the 'raspi5' server key is added to the user (root) known hosts file (/root/.ssh/known_hosts). So you can see that we can't logon this server as the 'root' user.

```
root@holmes~# ssh root@raspi5
The authenticity of host 'raspi5 (192.168.1.25)' can't be established.
ECDSA key fingerprint is SHA256:v1d0mK15pA9NtrhqbzFIu4boQoot99UxCi+aFcMs394.
ECDSA key fingerprint is MD5:99:4e:d6:3a:65:e1:bb:40:ec:ce:da:3b:52:63:ee:f1.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'raspi5,192.168.1.25' (ECDSA) to the list of known hosts.
root@raspi5's password:
Permission denied, please try again.
root@raspi5's password:
Permission denied, please try again.
root@raspi5's password:
Permission denied (publickey,password).
root@holmes~#
```

So let's start configuring the client to accept the automated 'root' login from 'holmes' system.

In the 'root' user HOME directory, check if you have a directory called '.ssh' and that it contains a private key (id_rsa) and a public key (id_rsa.pub)

```
root@holmes~# ls -l /root/.ssh
total 48
-rw-r-----  1 root root  1187 Oct  3  2017 authorized_keys
-rw-------  1 root root  1675 Feb 23  2016 id_rsa
-rw-r--r--  1 root root   403 Feb 23  2016 id_rsa.pub
-rw-r-----  1 root root 26291 Jul 17 09:27 known_hosts
root@holmes~#
```

Generating the ‘root’ user SSH key on the SADMIN server

- If you already have these files then proceed with the next step.
- If you don’t, run the command below to generate them.
    - When ask for a password just press [ENTER] to have a blank password.

```
root@holmes~/.ssh# ssh-keygen -C "SADMIN server"
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_rsa.
Your public key has been saved in /root/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:3dd5vZTTv3i8Qa0osnOmp5d0wVKh3Dl2ZziNpvwp3To SADMIN server
The key's randomart image is:
+---[RSA 2048]----+
|     ..    |
|         . o.. + |
|          oo= * +|
|         ..+o= =*|
|        S ..+..**|
|          . .=o+=|
|        ...oo *oo|
|        .o*. .E+o|
|        +O   .o+.|
+----[SHA256]-----+

root@holmes~/.ssh# ls -l id*
-rw------- 1 root root 1675 Jul 17 09:53 id_rsa
-rw-r--r-- 1 root root  395 Jul 17 09:53 id_rsa.pub
```

Temporarily allow ssh ‘root’ access to client

- Add this line to the SSH service configuration file, to allow access to ‘root’ user.

```
      root@raspi5:~# echo "PermitRootLogin yes" >> /etc/ssh/sshd_config
```

- Restart the SSH service

```
On Ubuntu,Raspbian and other Debian like O/S
root@raspi5:~# systemctl restart ssh

On RedHat, CentOS, Fedora
root@raspi5:~# systemctl restart sshd
```

Copying SADMIN server ssh public key to the client

```
root@holmes~# ssh-copy-id root@raspi5.maison.ca

/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/root/.ssh/id_rsa.pub"
/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any
that are already installed
/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it
is to install the new keys
root@raspi5.maison.ca's password: XXXXXX

Number of key(s) added: 1

Now try logging into the machine, with:   "ssh 'root@raspi5.maison.ca'"
and check to make sure that only the key(s) you wanted were added.
root@holmes~#
```

Put back original ssh configuration on client

- Confirm the last line in the file is the one we added :

```
root@raspi5:~# tail -1 /etc/ssh/sshd_config
PermitRootLogin yes
```

- Delete the last line in the file

```
root@raspi5:~# sed -i '$d' /etc/ssh/sshd_config
```

- Confirm that the delete worked

```
root@raspi5:~# tail -1 /etc/ssh/sshd_config
#    ForceCommand cvs server
root@raspi5:~#
```

- Restart the SSH service

```
On Ubuntu,Raspbian and other Debian like O/S
root@raspi5:~# systemctl restart ssh

On RedHat, CentOS, Fedora
root@raspi5:~# systemctl restart sshd
```