**Commands CCDC Winter 2016**

Debian Based

Check all Current Services

**service --status-all**

Check All open Ports

**nestat -tulpn**

Check all Users

**cut -d: -f1 /etc/passwd**

**getent /etc/password**

RHEL Based

Check all current services

Check all open ports

Check all users

Arch

Check all current services

Check all open ports

Check all users

Gentoo

Check all current services

Check all open ports

Check all users

**Quick Commands**

Check who is running a service

**$ ps -ef |grep apache**

rmp -qa |grep httpd

**dpkg -l |grep apache**

find if apache was installed through apt manager

ls -la /usr/local/src

where locally compiled sources would be in

**Find out information about the operating system**

cat /etc/issue

cat /etc/\*-release

**Find out where the root is of the server along where the configuration**

**file is located.**

apachectl -v

**Manual way of finding configuration file on the server**

RHEL

find / |grep "httpd\.conf"

DEB

find / grep "apache2\.conf"

**Search for exact keywords in files for the files in current directory**

grep -Ri DocumentRoot .

**Check for syntax error in apache configuration**

apachectl -t

gives all virtual host IP, name, configuration location

apachectl -t -D DUMP\_VHOSTS

check to see which modules are enabled.

apachectl -t D DUMP\_MODULES

Find which apache modules are enabled in debian

apache2/mods-available

apache2/mods-enablaed

Enable or disable apaceh modudels

enable a module a2enmod

disable a module

a2dismod

Finding Error Logs. if " ${APACHE\_LOG\_DIR} " is returned, it means that it is an enviromental variable and "export APACHE\_LOG\_DIR" should be searched instead

grep -Ri ErrorLog /etc/apache2

grep -R "export APACHE\_LOG\_DIR" /etc/apache2

to see log files in real time

tail -f logfile

####################PHPMYADMIN Notes

Finding All users on a system

cut -d: -f1 /etc/passwd

or

getent passwd

# Default Configuration file for phpmyadmin on Debian

# in this file, the default localhost/phpmyadmin can be changed to whater

# localhost/ganoo

/etc/apache2/conf-available/phpmyadmin.conf

# Default installation folder for phpmyadmin on Debian

# In this file there could be a .htaccess file that adds

# an extra layer of login security anytime someone tries to breaks in

/usr/share/phpmyadmin

/usr/share/phpmyadmin/.htaccess

# Inside .htaccess the AuthUserFile will display where the .htpasswd

# file is located, this file is used for a second layer authenticaion

/etc/phpmyadmin/.htpasswd

####

Note: in order for .htaccess files to work, a few steps need to be done

1) inside /etc/apache2/conf-available/phpmyadmin.conf

inside <Directory /usr/share/phpmyadmin

add AllowOverride All

2) Create .htaccess file

sudo nano /usr/share/phpmyadmin/.htaccess

enter the following lines

AuthType Basic

AuthName "Restricted Files"

AuthUserFile /etc/phpmyadmin/.htpasswd

Require valid-user

3) make sure apache2-utils is installed

this is required to generate the htpasswd file

sudo apt-get install htpasswd

4) create the htpasswd file

this file should be the same as the file in AuthUserFile

sudo htpasswd -c /etc/phpmyadmin/.htpasswd ANYUSERNAME

this will prompt you to enter a passwd

restart server and you are done!

https://www.howtoforge.com/setting-changing-resetting-mysql-root-passwords

####

**Splunk**

Note: port 8000 for apache and 8089 for management

Download with wget

wget -O splunk-6.3.2-aaff59bb082c-Linux-i686.tgz '<http://www.splunk.com/bin/splunk/DownloadActivityServlet?architecture=x86&platform=linux&version=6.3.2&product=splunk&filename=splunk-6.3.2-aaff59bb082c-Linux-i686.tgz&wget=true>'

tar -xvpf

cp -R splunk /opt/splunk

cd /opt/splunk/bin

./splunk start --accept-licence

**Installing Nagios on UBUNTU BOTNET**

<https://assets.nagios.com/downloads/nagioscore/docs/nagioscore/3/en/quickstart-ubuntu.html>

**Introduction**

This guide is intended to provide you with simple instructions on how to install Nagios from source (code) on Ubuntu and have it monitoring your local machine inside of 20 minutes. No advanced installation options are discussed here - just the basics that will work for 95% of users who want to get started.

These instructions were written based on an **Ubuntu 6.10** (desktop) installation. They should work for an **Ubuntu 7.10** install as well.

**What You'll End Up With**

If you follow these instructions, here's what you'll end up with:

* Nagios and the plugins will be installed underneath /usr/local/nagios
* Nagios will be configured to monitor a few aspects of your local system (CPU load, disk usage, etc.)
* The Nagios web interface will be accessible at http://localhost/nagios/

**Required Packages**

Make sure you've installed the following packages on your Ubuntu installation before continuing.

* Apache 2
* PHP
* GCC compiler and development libraries
* GD development libraries

You can use *apt-get* to install these packages by running the following commands:

sudo apt-get install apache2  
sudo apt-get install libapache2-mod-php5  
sudo apt-get install build-essential

With Ubuntu 6.10, install the gd2 library with this command:

sudo apt-get install libgd2-dev

With Ubuntu 7.10, the gd2 library name has changed, so you'll need to use the following:

sudo apt-get install libgd2-xpm-dev

**1) Create Account Information**

Become the root user.

sudo -s

Create a new *nagios* user account and give it a password.

/usr/sbin/useradd -m -s /bin/bash nagios  
passwd nagios

On older Ubuntu server editions (6.01 and earlier), you will need to also add a *nagios* group (it's not created by default). You should be able to skip this step on desktop, or newer server editions of Ubuntu.

/usr/sbin/groupadd nagios  
/usr/sbin/usermod -G nagios nagios

Create a new *nagcmd* group for allowing external commands to be submitted through the web interface. Add both the nagios user and the apache user to the group.

/usr/sbin/groupadd nagcmd  
/usr/sbin/usermod -a -G nagcmd nagios  
/usr/sbin/usermod -a -G nagcmd www-data

**2) Download Nagios and the Plugins**

Create a directory for storing the downloads.

mkdir ~/downloads  
cd ~/downloads

Download the source code tarballs of both Nagios and the Nagios plugins (visit<https://www.nagios.org/download/> for links to the latest versions). These directions were tested with Nagios 3.1.1 and Nagios Plugins 1.4.11.

wget http://prdownloads.sourceforge.net/sourceforge/nagios/nagios-3.2.3.tar.gz  
http://www.nagios-plugins.org/download/nagios-plugins-2.1.1.tar.gz

**3) Compile and Install Nagios**

Extract the Nagios source code tarball.

cd ~/downloads  
tar xzf nagios-3.2.3.tar.gz  
cd nagios-3.2.3

Run the Nagios configure script, passing the name of the group you created earlier like so:

./configure --with-command-group=nagcmd

Compile the Nagios source code.

make all

Install binaries, init script, sample config files and set permissions on the external command directory.

make install  
make install-init  
make install-config  
make install-commandmode

Don't start Nagios yet - there's still more that needs to be done...

**4) Customize Configuration**

Sample [configuration files](https://assets.nagios.com/downloads/nagioscore/docs/nagioscore/3/en/config.html) have now been installed in the */usr/local/nagios/etc* directory. These sample files should work fine for getting started with Nagios. You'll need to make just one change before you proceed...

Edit the */usr/local/nagios/etc/objects/contacts.cfg* config file with your favorite editor and change the email address associated with the *nagiosadmin* contact definition to the address you'd like to use for receiving alerts.

vi /usr/local/nagios/etc/objects/contacts.cfg

**5) Configure the Web Interface**

Install the Nagios web config file in the Apache conf.d directory.

make install-webconf

Create a *nagiosadmin* account for logging into the Nagios web interface. Remember the password you assign to this account - you'll need it later.

htpasswd -c /usr/local/nagios/etc/htpasswd.users nagiosadmin

Restart Apache to make the new settings take effect.

/etc/init.d/apache2 reload

Note Note: Consider implementing the ehanced CGI security measures described [here](https://assets.nagios.com/downloads/nagioscore/docs/nagioscore/3/en/cgisecurity.html) to ensure that your web authentication credentials are not compromised.

**6) Compile and Install the Nagios Plugins**

Extract the Nagios plugins source code tarball.

cd ~/downloads  
tar xzf nagios-plugins-1.4.11.tar.gz  
cd nagios-plugins-1.4.11

Compile and install the plugins.

./configure --with-nagios-user=nagios --with-nagios-group=nagios  
make  
make install

**7) Start Nagios**

Configure Nagios to automatically start when the system boots.

ln -s /etc/init.d/nagios /etc/rcS.d/S99nagios

Verify the sample Nagios configuration files.

/usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg

If there are no errors, start Nagios.

/etc/init.d/nagios start

**8) Login to the Web Interface**

You should now be able to access the Nagios web interface at the URL below. You'll be prompted for the username (*nagiosadmin*) and password you specified earlier.

http://localhost/nagios/

Click on the "Service Detail" navbar link to see details of what's being monitored on your local machine. It will take a few minutes for Nagios to check all the services associated with your machine, as the checks are spread out over time.

**9) Other Modifications**

If you want to receive email notifications for Nagios alerts, you need to install the mailx (Postfix) package.

sudo apt-get install mailx  
sudo apt-get install postfix

You'll have to edit the Nagios email notification commands found in */usr/local/nagios/etc/objects/commands.cfg* and change any '/bin/mail' references to '/usr/bin/mail'. Once you do that you'll need to restart Nagios to make the configuration changes live.

sudo /etc/init.d/nagios restart

Configuring email notifications is outside the scope of this documentation. Refer to your system documentation, search the web, or look to the [Nagios Support Portal](https://support.nagios.com) or [Nagios Community Wiki](http://wiki.nagios.org) for specific instructions on configuring your Ubuntu system to send email messages to external addresses.