Ceph Storage :: Next Big Thing

Ceph is a free software defined storage platform designed to present object, block, and file storage from a single distributed computer cluster. Ceph's main goals are to be completely distributed without a single point of failure, scalable to the exabyte level, highly relible anc freely-available

TUESDAY, SEPTEMBER 23, 2014

Ceph Calamari : The Survival Guide

CEPH CALAMARI : STEP-BY-STEP

Calamari is a management and monitoring system for Ceph storage cluster. It provides a beautiful Dashboard User Interface that makes Ceph cluster monitoring amazingly simple and handy. Calamari was initially a part of Inktank's Ceph Enterprise product offering and it has been open sourced few months back by Red Hat. Hence Calamari is a matured enterprise level software which is now Open Source.

If you are wondering how you can deploy calamari for your Ceph cluster , this blog will navigate you with the step by step instruction. As of now precompiled packages for calamari are not available , you would require to compile from source and get calamari up and running for your environment.

I would like to express my thanks to Gregory Meno and Dan Mick from Inktank (RedHat) for their support during troubleshooting.

INTRODUCTION

Calamari is the management and monitoring service for Ceph, exposing a high level REST API.

Calamari consists of two pieces (and two separate repos, now in the Ceph Github project):

1) Backend (Calamari Server) — the Calamari backend is written in Python 2.6+, using Saltstack, ZeroRPC, gevent, Django, django-rest-framework, graphite and instantiates a new REST API for integration with other systems. This is an important distinction because earlier versions of Calamari were based on the Ceph REST API. The new Calamari REST API is designed to be much more comprehensive and should be the basis for new development efforts that wish to interact with a Ceph cluster. Source Code :https://github.com/ceph/calamari

2) Frontend (Calamari Clients) — the Calamari frontend is a web browser UI implemented primarily in Javascript that

NAVIGATION
► 2013 (11)
▼ 2014 (7)
► January (4)
► March (1)
► April (1)
▼ September (1)
Ceph Calamari : The Survival Guide
► 2015 (2)

POPULAR POSTS
Zero To Hero Guide : CEPH CLUSTER PLANNING
Ceph Installation :: Part-2
How Data Is Stored I CEPH Cluster
Admin Guide :: Replacing a Failed in a Ceph Cluster
Ceph Installation :: Part-1

uses the Calamari REST API. Source Code : https://github.com/ceph/calamari-clients
Calamari Server side components include : Apache, salt-master , supervisord , cthulhu , carbon-cache
Calamari Client side components include : salt-minion , diamond

BUILDING CALAMARI FROM SOURCE

Unlike Ceph , Calamari does not comes with prebuilt RPM / DEB packages. If you are interested in calamari you need to build your own packages based on your distributions. To deploy Calamari , you need to build packages for Calamari Server as well as for Calamari clients.

Building Calamari Server packages

These instruction will guide you to build RPM packages that will be used with RHEL based distributions. For Ubuntu DEB packages , most of the steps are same. I would try to highlight extra steps if required.
1. Install **GIT** and **vagrant** on your local workstation.

For example i am using a MAC workstation and will connect to my Ceph cluster and other servers over SSH. So i would install GIT and vagrant locally on my machine. On this workstation we will create some vagrant instances and inside those vagrant instances we will build RPM packages for Calamari server and client . Once packages are built , we will transfer the packages from the workstation to the Server machine that you designate for Calamari server and calamari clients.

2. Clone calamari server and diamond repositories.

```
$ mkdir /tmp/calamari-repo
$ cd /tmp/calamari-repo

$ git clone https://github.com/ceph/calamari.git
Cloning into 'calamari'...
remote: Counting objects: 10253, done.
remote: Compressing objects: 100% (4433/4433), done.
remote: Total 10253 (delta 5434), reused 10188 (delta 5387)
Receiving objects: 100% (10253/10253), 20.53 MiB | 3.55 MiB/s, done.
Resolving deltas: 100% (5434/5434), done.
Checking connectivity... done.
$

$ git clone https://github.com/ceph/Diamond.git --branch=calamari
Cloning into 'Diamond'...
remote: Counting objects: 16225, done.
remote: Compressing objects: 100% (9229/9229), done.
remote: Total 16225 (delta 6170), reused 16225 (delta 6170)
Receiving objects: 100% (16225/16225), 3.79 MiB | 1.19 MiB/s, done.
Resolving deltas: 100% (6170/6170), done.
Checking connectivity... done.
$
```

3. Check calamari contents , under vagrant directory , you would find various distributions . Select your distribution and perform vagrant up.
Ubuntu users : In this step select the vagrant directory *precise-build* for ubuntu DEB packages.

```
$ cd calamari/
$ ls -la
total 168
drwxr-xr-x  32 ksingh  wheel   1088 Sep 17 10:34 .
drwxr-xr-x   4 ksingh  wheel    136 Sep 17 10:36 ..
drwxr-xr-x  13 ksingh  wheel    442 Sep 17 10:34 .git
-rw-r--r--   1 ksingh  wheel    292 Sep 17 10:34 .gitignore
-rw-r--r--   1 ksingh  wheel   1310 Sep 17 10:34 .travis.yml
-rw-r--r--   1 ksingh  wheel    251 Sep 17 10:34 COPYING
-rw-r--r--   1 ksingh  wheel  26436 Sep 17 10:34 COPYING-LGPL2.1
-rw-r--r--   1 ksingh  wheel   8977 Sep 17 10:34 Makefile
-rw-r--r--   1 ksingh  wheel   2183 Sep 17 10:34 README.rst
-rwxr-xr-x   1 ksingh  wheel   2961 Sep 17 10:34 adduser.py
drwxr-xr-x   6 ksingh  wheel    204 Sep 17 10:34 alembic
-rwxr-xr-x   1 ksingh  wheel    979 Sep 17 10:34 build-rpm.sh
```

```
drwxr-xr-x   5 ksingh  wheel    170 Sep 17 10:34 calamari-common
drwxr-xr-x   4 ksingh  wheel    136 Sep 17 10:34 calamari-web
-rw-r--r--   1 ksingh  wheel   3785 Sep 17 10:34 calamari.spec
drwxr-xr-x  11 ksingh  wheel    374 Sep 17 10:34 conf
drwxr-xr-x   5 ksingh  wheel    170 Sep 17 10:34 cthulhu
drwxr-xr-x  13 ksingh  wheel    442 Sep 17 10:34 debian
drwxr-xr-x  11 ksingh  wheel    374 Sep 17 10:34 dev
drwxr-xr-x  13 ksingh  wheel    442 Sep 17 10:34 doc
-rwxr-xr-x   1 ksingh  wheel    659 Sep 17 10:34 get-flavor.sh
-rwxr-xr-x   1 ksingh  wheel    874 Sep 17 10:34 get-versions.sh
drwxr-xr-x   5 ksingh  wheel    170 Sep 17 10:34 minion-sim
-rwxr-xr-x   1 ksingh  wheel    326 Sep 17 10:34 pre-commit.py
drwxr-xr-x   4 ksingh  wheel    136 Sep 17 10:34 repobuild
drwxr-xr-x   4 ksingh  wheel    136 Sep 17 10:34 requirements
drwxr-xr-x   6 ksingh  wheel    204 Sep 17 10:34 rest-api
drwxr-xr-x   4 ksingh  wheel    136 Sep 17 10:34 salt
drwxr-xr-x  17 ksingh  wheel    578 Sep 17 10:34 tests
-rw-r--r--   1 ksingh  wheel    792 Sep 17 10:34 tox.ini
drwxr-xr-x  11 ksingh  wheel    374 Sep 17 10:34 vagrant
drwxr-xr-x   3 ksingh  wheel    102 Sep 17 10:34 webapp
$
$ cd vagrant/
$ ls -la
total 8
drwxr-xr-x  11 ksingh  wheel   374 Sep 17 10:34 .
drwxr-xr-x  32 ksingh  wheel  1088 Sep 17 10:34 ..
drwxr-xr-x   4 ksingh  wheel   136 Sep 17 10:34 centos-build
drwxr-xr-x   5 ksingh  wheel   170 Sep 17 10:34 devmode
drwxr-xr-x   4 ksingh  wheel   136 Sep 17 10:34 precise-build
drwxr-xr-x   4 ksingh  wheel   136 Sep 17 10:34 production
drwxr-xr-x   4 ksingh  wheel   136 Sep 17 10:34 rhel-build
drwxr-xr-x   4 ksingh  wheel   136 Sep 17 10:34 rhel7-build
drwxr-xr-x   4 ksingh  wheel   136 Sep 17 10:34 trusty-build
-rwxr-xr-x   1 ksingh  wheel   344 Sep 17 10:34 urllib-bootstrap-salt.sh
drwxr-xr-x   4 ksingh  wheel   136 Sep 17 10:34 wheezy-build
$
$ cd centos-build/
$ ls -la
total 8
drwxr-xr-x   4 ksingh  wheel   136 Sep 17 10:34 .
drwxr-xr-x  11 ksingh  wheel   374 Sep 17 10:34 ..
-rw-r--r--   1 ksingh  wheel  1112 Sep 17 10:34 Vagrantfile
drwxr-xr-x   4 ksingh  wheel   136 Sep 17 10:34 salt
$
```

4. Finally perform **vagrantup** for Centos distribution, this will create a virtual environment with all the prerequisite necessary for calamari server package building process.

```
$ vagrant up
Bringing machine 'default' up with 'virtualbox' provider...
==> default: Importing base box 'centos6'...
==> default: Matching MAC address for NAT networking...
==> default: Setting the name of the VM: centos-build_default_1410939808331_97817
==> default: Clearing any previously set network interfaces...
==> default: Preparing network interfaces based on configuration...
    default: Adapter 1: nat
==> default: Forwarding ports...
    default: 22 => 2201 (adapter 1)
==> default: Running 'pre-boot' VM customizations...
==> default: Booting VM...
==> default: Waiting for machine to boot. This may take a few minutes...
    default: SSH address: 127.0.0.1:2201
    default: SSH username: vagrant
    default: SSH auth method: private key
```

```
==> default: Machine booted and ready!
==> default: Checking for guest additions in VM...
==> default: Mounting shared folders...
    default: /git => /private/tmp/calamari-repo
    default: /vagrant => /private/tmp/calamari-repo/calamari/vagrant/centos-build
    default: /srv/salt => /private/tmp/calamari-repo/calamari/vagrant/centos-build/salt/roots
==> default: Running provisioner: salt...
Copying salt minion config to vm.
Checking if salt-minion is installed
salt-minion was not found.
Checking if salt-call is installed
salt-call was not found.
Bootstrapping Salt... (this may take a while)
Salt successfully configured and installed!
run_overstate set to false. Not running state.overstate.
run_highstate set to false. Not running state.highstate.
$
```

5. On this new CentOS vagrant VM , you would notice some filesystem were mounted , actually they calamari source directory that we have cloned from github and has been shared and mounted as file system on CentOS vagrant VM. Now for building packages , we will use the same directory hierarchy and the resultant RPM package will also be stored in this shared directory.SSH into vagrant CentOS instance

```
$ vagrant ssh

[vagrant@vagrant-centos64 ~]$
[vagrant@vagrant-centos64 ~]$ df -h
Filesystem            Size  Used Avail Use% Mounted on
/dev/sda1             7.3G 1017M  6.0G  15% /
tmpfs                 246M     0  246M   0% /dev/shm
git                   233G  214G   20G  92% /git
vagrant               233G  214G   20G  92% /vagrant
srv_salt              233G  214G   20G  92% /srv/salt
[vagrant@vagrant-centos64 ~]$
[vagrant@vagrant-centos64 ~]$ cd /git
[vagrant@vagrant-centos64 git]$ ll
total 0
drwxr-xr-x 1 vagrant vagrant 1088 Sep 17 07:34 calamari
drwxr-xr-x 1 vagrant vagrant  952 Sep 17 07:35 Diamond
[vagrant@vagrant-centos64 git]$
```

6. Build you Calamari server RPM packages , with the below salt command.

```
$ sudo salt-call state.highstate
```

```
[vagrant@vagrant-centos64 ~]$ sudo salt-call state.highstate
[INFO    ] Loading fresh modules for state activity
[INFO    ] Creating module dir '/var/cache/salt/minion/extmods/modules'
[INFO    ] Syncing modules for environment 'base'
[INFO    ] Loading cache from salt://_modules, for base)

............... Output truncated ................


----------
          ID: cp-artifacts-to-share Diamond/dist/diamond-*.noarch.rpm
    Function: cmd.run
        Name: cp Diamond/dist/diamond-*.noarch.rpm /git/
      Result: True
     Comment: Command "cp Diamond/dist/diamond-*.noarch.rpm /git/" run
```

```
    Changes:
             ----------
             pid:
                 28292
             retcode:
                 0
             stderr:

             stdout:


Summary
-------------
Succeeded: 11
Failed:     0
-------------
Total:     11
[vagrant@vagrant-centos64 ~]$
```

For brevity the output of above command has been truncated . You can follow the output here http://paste.ubuntu.com/8363649/

7. The output summary confirms that there are 11 Successful and 0 Failed operations . List the directories to find your Calamar Server RPM package. Don't worry this should be copied to the base directory of your workstation , as the same has been shared and mounted with vagrant instance.

```
[vagrant@vagrant-centos64 ~]$ ll
total 14964
drwxr-xr-x 19 vagrant vagrant     4096 Sep 17 08:00 calamari
-rw-r--r--  1 vagrant vagrant 15303265 Sep 17 08:00 calamari-server_1.2.1.tar.gz
drwxr-xr-x 12 vagrant vagrant     4096 Sep 17 07:59 Diamond
drwxr-xr-x  8 vagrant vagrant     4096 Sep 17 08:00 rpmbuild
[vagrant@vagrant-centos64 ~]$
[vagrant@vagrant-centos64 ~]$ cd rpmbuild/
[vagrant@vagrant-centos64 rpmbuild]$
[vagrant@vagrant-centos64 rpmbuild]$
[vagrant@vagrant-centos64 rpmbuild]$ find . -name *.rpm
./BUILD/calamari-server-1.2.1/repobuild/el6/sshpass-1.05-1.el6.x86_64.rpm
./BUILD/calamari-server-1.2.1/repobuild/el6/openpgm-5.1.118-3.el6.x86_64.rpm
./BUILD/calamari-server-1.2.1/repobuild/el6/python-msgpack-0.1.13-3.el6.x86_64.rpm
./BUILD/calamari-server-1.2.1/repobuild/el6/salt-2014.1.10-4.el6.noarch.rpm
./BUILD/calamari-server-1.2.1/repobuild/el6/PyYAML-3.10-3.el6.x86_64.rpm
./BUILD/calamari-server-1.2.1/repobuild/el6/python-babel-0.9.4-5.1.el6.noarch.rpm
./BUILD/calamari-server-1.2.1/repobuild/el6/python-zmq-2.2.0.1-1.el6.x86_64.rpm
./BUILD/calamari-server-1.2.1/repobuild/el6/salt-minion-2014.1.10-4.el6.noarch.rpm
./BUILD/calamari-server-1.2.1/repobuild/el6/libyaml-0.1.6-1.el6.x86_64.rpm
./BUILD/calamari-server-1.2.1/repobuild/el6/python-crypto-2.0.1-22.el6.x86_64.rpm
./BUILD/calamari-server-1.2.1/repobuild/el6/zeromq3-3.2.4-1.el6.x86_64.rpm
./BUILD/calamari-server-1.2.1/repobuild/el6/m2crypto-0.20.2-9.el6.x86_64.rpm
./BUILD/calamari-server-1.2.1/repobuild/el6/diamond-3.4.67-0.noarch.rpm
./BUILD/calamari-server-1.2.1/repobuild/el6/python-jinja2-2.2.1-2.el6_5.x86_64.rpm
./RPMS/x86_64/calamari-server-1.2.1-29_g9d02dc3.el6.x86_64.rpm
[vagrant@vagrant-centos64 rpmbuild]$
```

8. You should now exit from the centos vagrant machine and on your local machine list the directory you would find rpm packages for calamari server and diamond.

```
teeri:calamari-repo ksingh$
teeri:calamari-repo ksingh$ ls -la
total 61960
drwxr-xr-x   7 ksingh  wheel       238 Sep 17 11:18 .
drwxrwxrwt  10 root    wheel       340 Sep 17 11:46 ..
```

```
drwxr-xr-x  28 ksingh  wheel       952 Sep 17 10:35 Diamond
drwxr-xr-x  32 ksingh  wheel      1088 Sep 17 10:34 calamari
-rw-r--r--   1 ksingh  wheel   7038999 Sep 17 11:18 calamari-repo-el6.tar.gz
-rw-r--r--   1 ksingh  wheel  24082060 Sep 17 11:18 calamari-server-1.2.1-29_g9d02dc3.el6.x86_64.rpm
-rw-r--r--   1 ksingh  wheel    595548 Sep 17 11:18 diamond-3.4.67-0.noarch.rpm
teeri:calamari-repo ksingh$
```

9 . At this point you are ready with Calamari server RPM packages. The next step is to build calamari client packages.

Building Calamari client packages

The calamari client packages provides the GUI to calamari dashboard. As calamari build process is not greatly matured , we need to use ubuntu vagrant to build client packages. This will generate a tarball of client packages , that you can use with centos machines.

CentOS / RHEL users : Note , this will not generate any RPM package and you would need to user tarball.
Ubuntu Users: Cheers , you will get DEB packages that you can use directly.

1. Clone calamari client from github on to your workstation (the same directory as you were using before) , and perform vagrant for ubuntu precise VM

CentOS / RHEL users : Don't panic , you also have to user ubuntu precise vagrant instance in order to build Calamari client tarball.
Ubuntu Users : No comment for you , please proceed :-)

```
$ cd /tmp/calamari-repo/
$
$ git clone https://github.com/ceph/calamari-clients.git
Cloning into 'calamari-clients'...
remote: Counting objects: 16261, done.
remote: Total 16261 (delta 0), reused 0 (delta 0)
Receiving objects: 100% (16261/16261), 28.81 MiB | 4.01 MiB/s, done.
Resolving deltas: 100% (9441/9441), done.
Checking connectivity... done.
```

```
$ ls -la
total 61960
drwxr-xr-x   8 ksingh  wheel       272 Sep 17 12:03 .
drwxrwxrwt  10 root    wheel       340 Sep 17 11:46 ..
drwxr-xr-x  28 ksingh  wheel       952 Sep 17 10:35 Diamond
drwxr-xr-x  32 ksingh  wheel      1088 Sep 17 10:34 calamari
drwxr-xr-x  22 ksingh  wheel       748 Sep 17 12:04 calamari-clients
-rw-r--r--   1 ksingh  wheel   7038999 Sep 17 11:18 calamari-repo-el6.tar.gz
-rw-r--r--   1 ksingh  wheel  24082060 Sep 17 11:18 calamari-server-1.2.1-29_g9d02dc3.el6.x86_64.rpm
-rw-r--r--   1 ksingh  wheel    595548 Sep 17 11:18 diamond-3.4.67-0.noarch.rpm
```

```
$ cd calamari-clients/vagrant/precise-build/
$
$ ls -la
total 16
drwxr-xr-x   4 ksingh  wheel  136 Sep 17 12:04 .
drwxr-xr-x  10 ksingh  wheel  340 Sep 17 12:04 ..
-rw-r--r--   1 ksingh  wheel  789 Sep 17 12:04 README.rst
-rw-r--r--   1 ksingh  wheel  882 Sep 17 12:04 Vagrantfile
$

$ vagrant up
Bringing machine 'default' up with 'virtualbox' provider...
==> default: Importing base box 'precise64'...
==> default: Matching MAC address for NAT networking...
==> default: Setting the name of the VM: precise-build_default_1410947713821_66264
==> default: Clearing any previously set network interfaces...
==> default: Preparing network interfaces based on configuration...
    default: Adapter 1: nat
```

```
==> default: Forwarding ports...
    default: 22 => 2202 (adapter 1)
==> default: Running 'pre-boot' VM customizations...
==> default: Booting VM...
==> default: Waiting for machine to boot. This may take a few minutes...
    default: SSH address: 127.0.0.1:2202
    default: SSH username: vagrant
    default: SSH auth method: private key
==> default: Machine booted and ready!
==> default: Checking for guest additions in VM...
    default: The guest additions on this VM do not match the installed version of
    default: VirtualBox! In most cases this is fine, but in rare cases it can
    default: prevent things such as shared folders from working properly. If you see
    default: shared folder errors, please make sure the guest additions within the
    default: virtual machine match the version of VirtualBox you have installed on
    default: your host and reload your VM.
    default:
    default: Guest Additions Version: 4.2.0
    default: VirtualBox Version: 4.3
==> default: Mounting shared folders...
    default: /git => /private/tmp/calamari-repo
    default: /vagrant => /private/tmp/calamari-repo/calamari-clients/vagrant/precise-build
    default: /srv/salt => /private/tmp/calamari-repo/calamari-clients/vagrant/salt/roots
==> default: Running provisioner: salt...
Copying salt minion config to vm.
Checking if salt-minion is installed
salt-minion was not found.
Checking if salt-call is installed
salt-call was not found.
Bootstrapping Salt... (this may take a while)
Salt successfully configured and installed!
run_overstate set to false. Not running state.overstate.
run_highstate set to false. Not running state.highstate.
$
```

```
$ vagrant ssh
Welcome to Ubuntu 12.04 LTS (GNU/Linux 3.2.0-23-generic x86_64)

 * Documentation:  https://help.ubuntu.com/
New release '14.04.1 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Welcome to your Vagrant-built virtual machine.
Last login: Fri Sep 14 06:23:18 2012 from 10.0.2.2
vagrant@precise64:~$
vagrant@precise64:~$
vagrant@precise64:~$ df -h
Filesystem                  Size  Used Avail Use% Mounted on
/dev/mapper/precise64-root   79G  2.3G   73G   4% /
udev                        489M  4.0K  489M   1% /dev
tmpfs                       200M  276K  199M   1% /run
none                        5.0M     0  5.0M   0% /run/lock
none                        498M     0  498M   0% /run/shm
/dev/sda1                   228M   25M  192M  12% /boot
git                         233G  216G   17G  93% /git
vagrant                     233G  216G   17G  93% /vagrant
srv_salt                    233G  216G   17G  93% /srv/salt
vagrant@precise64:~$
```

2. Install the following dependencies using either aptitude to apt-get

```
vagrant@precise64:~$ sudo apt-get install ruby1.9.1 ruby1.9.1-dev python-software-properties g++ make
git debhelper build-essential devscripts
```

3. Install node js from PPA

```
$ sudo apt-add-repository http://ppa.launchpad.net/chris-lea/node.js/ubuntu
$ sudo apt-get update
$ sudo apt-get install nodejs
$ sudo npm install -g bower@1.3.8
$ sudo npm install -g grunt-cli
$ sudo gem install compass
```

4. Start the build process for Calamari client DEB packages / tarball (centos / rhel)

```
vagrant@precise64:~$ sudo salt-call state.highstate
[INFO    ] Loading fresh modules for state activity
[INFO    ] Creating module dir '/var/cache/salt/minion/extmods/modules'
[INFO    ] Syncing modules for environment 'base'
[INFO    ] Loading cache from salt://_modules, for base)
[INFO    ] Caching directory '_modules/' for environment 'base'
[INFO    ] Creating module dir '/var/cache/salt/minion/extmods/states'
[INFO    ] Syncing states for environment 'base'
[INFO    ] Loading cache from salt://_states, for base)

.................. OUTPUT TRUNCATED .........................

----------
          ID: copyout_build_product
    Function: cmd.run
        Name: cp calamari-clients*tar.gz /git/
      Result: True
     Comment: Command "cp calamari-clients*tar.gz /git/" run
     Changes:
              ----------
              pid:
                  25090
              retcode:
                  0
              stderr:

              stdout:


Summary
-------------
Succeeded: 13
Failed:     0
-------------
Total:     13
vagrant@precise64:~$
```

You can follow the entire output at http://paste.ubuntu.com/8364367/

5. Check the shared filesystem and read my comments against the output.

```
vagrant@precise64:~$ df -h
Filesystem                  Size  Used Avail Use% Mounted on
/dev/mapper/precise64-root   79G  3.8G   72G   6% /
udev                        489M  4.0K  489M   1% /dev
tmpfs                       200M  276K  199M   1% /run
none                        5.0M     0  5.0M   0% /run/lock
none                        498M     0  498M   0% /run/shm
/dev/sda1                   228M   25M  192M  12% /boot
git                         233G  219G   15G  94% /git
vagrant                     233G  219G   15G  94% /vagrant
```

```
srv_salt                   233G  219G   15G  94% /srv/salt
vagrant@precise64:~$
vagrant@precise64:~$ cd /git/
vagrant@precise64:/git$ ll
total 34324
drwxr-xr-x  1 vagrant vagrant      340 Sep 17 10:37 ./
drwxr-xr-x 25 root    root        4096 Sep 17 09:55 ../
drwxr-xr-x  1 vagrant vagrant     1088 Sep 17 07:34 calamari/
# git clone of calamari server
drwxr-xr-x  1 vagrant vagrant      748 Sep 17 09:04 calamari-clients/
# git clone of calamari clients
-rw-r--r--  1 vagrant vagrant  1705158 Sep 17 10:37 calamari-clients_1.2.1-27-g9eb09d5_all.deb
# package for installation on calamari server if OS is ubuntu
-rw-r--r--  1 vagrant vagrant  1711183 Sep 17 10:37 calamari-clients-build-output.tar.gz  #
unpackaged build of calamari-clients , to be added on calamari server if the OS is CentOS
-rw-r--r--  1 vagrant vagrant  7038999 Sep 17 08:18 calamari-repo-el6.tar.gz
-rw-r--r--  1 vagrant vagrant 24082060 Sep 17 08:18 calamari-server-1.2.1-29_g9d02dc3.el6.x86_64.rpm
# package for installation on calamari server
drwxr-xr-x  1 vagrant vagrant      952 Sep 17 07:35 Diamond/
# git clone of diamond
-rw-r--r--  1 vagrant vagrant   595548 Sep 17 08:18 diamond-3.4.67-0.noarch.rpm
# diamond package for use on ceph nodes
vagrant@precise64:/git$
```

6. AT THIS STAGE YOU HAVE CALAMARI CLIENT DEB PACKAGES AND TARBALL CALAMARI-CLIENTS-BUILD-OUTPUT.TAR.GZ

DEPLOYING CALAMARI

Deploying calamari is again a multi step process.

#1 Install Calamari server packages
#2 Add calamari client web app contents to Calamari server
#3 Install diamond and salt-minion on Ceph nodes , that you wish to monitor using Calamari dashboard.

Installing Calamari Server

Copy the following packages that we have generated in the earlier steps from the workstation to the server that you intend to configure as calamari master.

calamari-server-1.2.1-29_g9d02dc3.el6.x86_64.rpm
diamond-3.4.67-0.noarch.rpm
calamari-clients-build-output.tar.gz

Ubuntu Users : Copy the DEB packages of calamari server and client.

1. Instal the packages on the node that will be your Calamari master.

```
$ sudo yum install -y calamari-server-1.2.1-37_g6f353e6.el6.x86_64.rpm
$ sudo yum install -y diamond-3.4.67-0.noarch.rpm
```

2. Uncompress , untar the files and move it to /opt/calamari/webapp of calamari master . These files provides GUI facility to calamari webapp

```
$ sudo gunzip calamari-clients-build-output.tar.gz
$ sudo tar -xvf calamari-clients-build-output.tar
```

```
$ cd /tmp/opt/calamari/webapp
$ sudo cp -rp content/ /opt/calamari/webapp/content
$ sudo  ls -l /opt/calamari/webapp/content
total 16
drwxr-xr-x. 7 1000 1000 4096 Sep 23 09:28 admin
drwxr-xr-x. 7 1000 1000 4096 Sep 23 09:28 dashboard
```

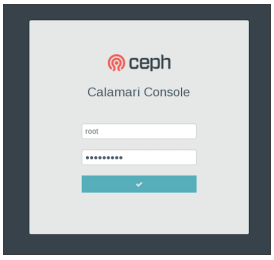
```
drwxr-xr-x. 7 1000 1000 4096 Sep 23 09:28 login
drwxr-xr-x. 8 1000 1000 4096 Sep 23 09:28 manage
```

3. At this point your calamari server is ready , you just need to initialize calamari by running
$ sudo calamari-ctl initialize

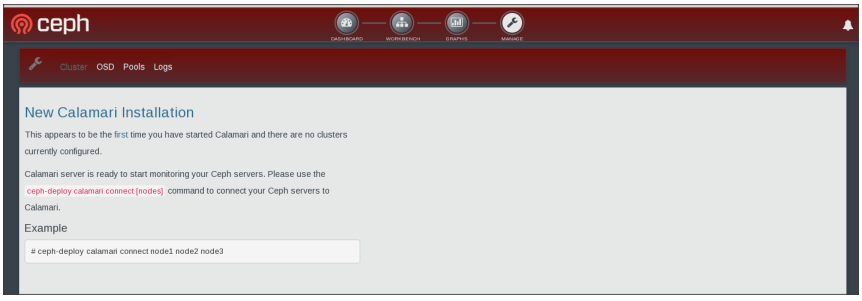
```
[ceph@ceph-node4 calamari]$ sudo calamari-ctl initialize
[INFO] Loading configuration..
[INFO] Starting/enabling salt...
[INFO] Starting/enabling postgres...
[INFO] Initializing database...
[INFO] Initializing web interface...
[INFO] You will now be prompted for login details for the administrative user account.  This is the
account you will use to log into the web interface once setup is complete.
Username (leave blank to use 'root'):
Email address: karan.singh@csc.fi
Password:
Password (again):
Superuser created successfully.
[INFO] Starting/enabling services...
[INFO] Restarting services...

................................. Output Truncated ...............
```

This username and password would be required to login to Calamari dashboard.

4. Openup browser and navigate to the IP address of your calamari server. You would see calamari dashboard login page , provide the login details that we have set in the last step.



5 Calamari would not be able to find the Ceph cluster and will ask to add a cluster , for this we need to add Ceph clients to dashboard by installing salt-minion and diamond packages on them.



Adding your Ceph cluster nodes to Calamari Dashboard

Perform the following steps on all your Ceph cluster nodes, inorder to add them to Calamari dashboard.

1. Install diamond packages that we have build already (previous steps)
$ sudo yum install -y diamond-3.4.67-0.noarch.rpm

2. Create a default diamond configuration file.
$ sudo mv /etc/diamond/diamond.conf.example /etc/diamond/diamond.conf

3. Install salt-minion packages
$ sudo yum install -y salt-minion
$ sudo chkconfig salt-minion on

Ubuntu Users : Use ubuntu package manager for installing salt-minion

4. Configure salt-minion , so that it can reach to Calamari master.
Edit /etc/salt/minion and add your calamari server hostname.

```
master: ceph-node4
```

5. Restart services
$ sudo service salt-minion restart
$ sudo service diamond restart

6. Repeat these steps for all the Ceph cluster nodes

Authorize Ceph cluster to add in calamari

Once you configure all the Ceph cluster nodes with salt-minion and diamond, you need to accept salt-keys from calamari master

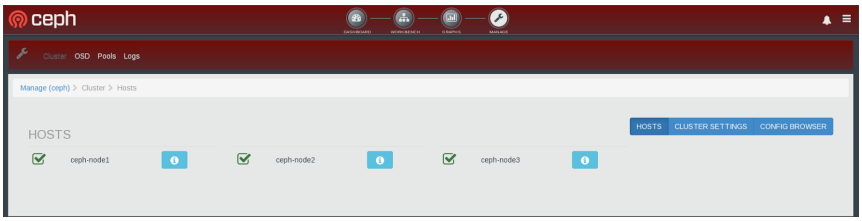
1. From calamari master check for unaccepted salt-keys.
$ sudo salt-key -L

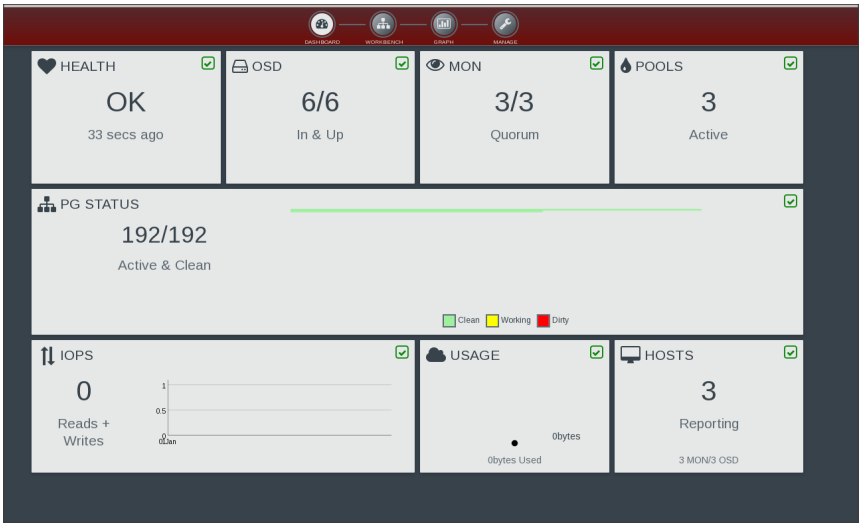
```
$ sudo salt-key -L
Accepted Keys:
Unaccepted Keys:
ceph-node1
ceph-node2
ceph-node3
Rejected Keys:
$
```

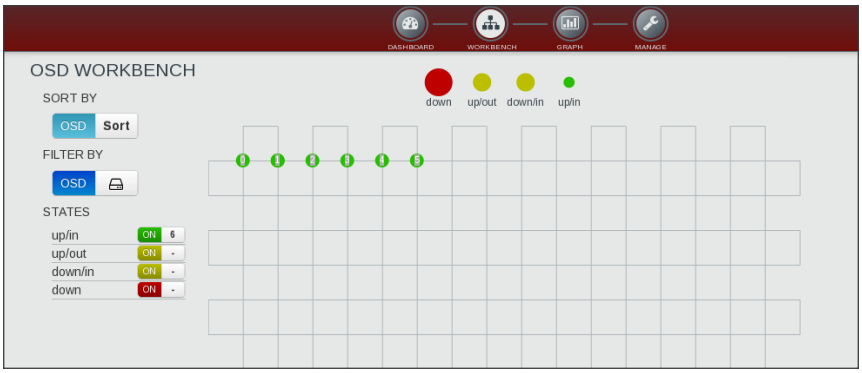
2. Accept salt-keys

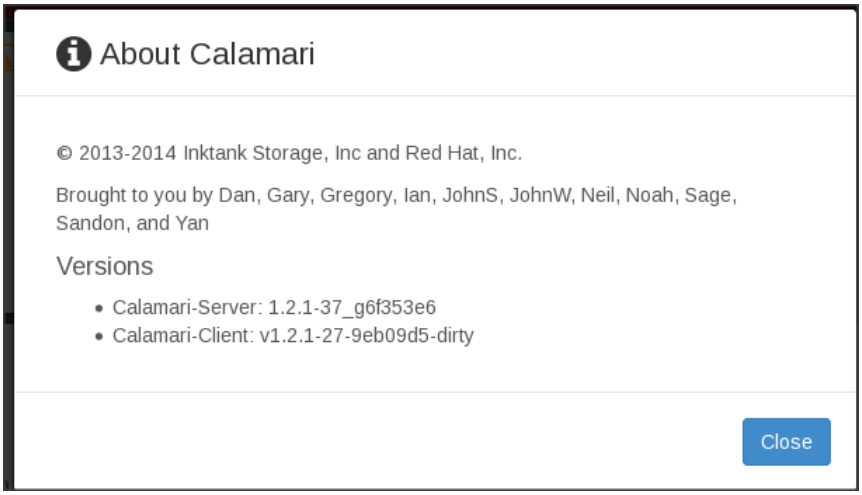
```
$ sudo salt-key -A
The following keys are going to be accepted:
Unaccepted Keys:
ceph-node1
ceph-node2
ceph-node3
Proceed? [n/Y] y
Key for minion ceph-node1 accepted.
Key for minion ceph-node2 accepted.
Key for minion ceph-node3 accepted.

$ sudo salt-key -L
Accepted Keys:
ceph-node1
ceph-node2
ceph-node3
Unaccepted Keys:
Rejected Keys:
$
```

3. Finally check calamari dashboard , you should be able to view the Ceph cluster.









Posted by Karan Singh at 8:13 AM 0 Comments

Labels: calamari, ceph, ceph calamari, ceph dashboard, ceph gui, ceph monitoring, ceph step by step, karan singh, Karan Singh :: Ceph

Blogs

Location: Espoo, Finland

26 comments

Add a comment

Top comments

Ceph 1 year ago - Shared publicly

RT @scuttlemonkey: Don't miss @karansingh010 's step-by-step walkthrough of @Ceph Calamari. http://ow.ly/BSbWf

+6 1 · Reply

Dmitry Nosachev 1 year ago

Ceph в сто раз лучше любой политоты.

Translate

Karan Singh 1 year ago - Shared publicly

Ceph Calamari : The Survival Guide

Ceph Calamari : step-by-step Calamari is a management and monitoring system for Ceph storage cluster. It provides a beautiful Dashboard User Interface that makes Ceph cluster monitoring amazingly simple and handy. Calamari was initially a part of Inktank'...

+2 1 · Reply

Karan Singh via Google+ 1 year ago - Shared publicly

Step by step guide to Ceph Calamari

+2 1 · Reply

View all 4 replies

Jesús Chávez Argüelles 10 months ago

Can anybody tell why I have 500 internal error in mu web browser when I try hit localhost address to get access to Calamari? Everytime job went succesful I dont know that else to do :(

Karan Singh 10 months ago

+Jesús Chávez Argüelles you can give a try by reinitializing calamari-ctl again.

Robert Sander via Google+ 1 year ago - Shared publicly

> **Ceph** originally shared this
> RT @scuttlemonkey: Don't miss @karansingh010 's step-by-step walkthrough of @Ceph Calamari. http://ow.ly/BSbWf

1 · Reply

Mike Plugnikov 1 year ago - CephRussian (Статьи)

> **Ceph** originally shared this
> RT @scuttlemonkey: Don't miss @karansingh010 's step-by-step walkthrough of @Ceph Calamari. http://ow.ly/BSbWf

+2 1

Jean-Philippe Pagadoy 1 week ago - Shared publicly

Hy thanks for sharing. I would to install calamari on fedora. And with squid. Is that the same steps? Fedora has rpm packages. I suppose to use centos in the directory? Thanks in advance for your reply. Best wishes

1 · Reply

Steffen Winther Sørensen 11 months ago - Shared publicly

Would it be possible to run UI plus other parts on a non-Ceph node, like a separate webapps+httpd, calamari, salt-master server, which then could talk to diamond/salt-minion on Ceph cluster nodes?

1 · Reply

Karan Singh 11 months ago

I have read somewhere , it should be doable, as long as the UI node can talk to ceph cluster.

Fourat Ferchichi 1 year ago - Shared publicly

I am following the installation tutorial
http://karan-mj.blogspot.fi/2014/09/ceph-calamari-survival-guide.html,
I successfully built the server packages, but after installing the required packages on the precise64 VM in order to build the client packages,
Read more (444 lines)

1 · Reply

View all 9 replies

Karan Singh 11 months ago

+Fourat Ferchichi Hello

Are you still stucked at the same error . You error lines says that "fatal: unable to connect to github.com:" do you have connectivity to outside world from this machine. ?

DIO PA 11 months ago (edited)

I solved
The problem was in the git download version.
yo this is a modified version 1.2.3.1 or 1.3rc version.
But get down in 1.2.2 and "git clone https://github.com/ceph/calamari-clients.git".
So perfect when you install the 1.2.3 version that is available through the installation .

Johan Kooijman 1 year ago - Shared publicly

Thx, good tutorial. After following I see 12 nodes in calamari (9 OSD + 3 mon), but still this message. Any hint on what to do? This is a working ceph cluster already!

12 Ceph servers are connected to Calamari, but no Ceph cluster has been created yet. Please use ceph-deploy to create a cluster; please see the Inktank Ceph Enterprise documentation for more details.

1 · Reply

Karan Singh 1 year ago

I would suggest restarting calamari server services using supervisorctl restart all . Or if possible restart the node and check.

Claws Sahlström 1 year ago - Shared publicly

Thanks, I was looking for something like this to have a dashboard to keep an eye on my small 3-node Ceph-cluster at home.

+1 1 · Reply

Post a Comment

Newer Post Home Older Post

Subscribe to: Post Comments (Atom)

GOOGLE+ FOLLOWERS

Karan Singh

Add to circles

193 have me in circles View all