Ceph Install / Deployment in a Production Environment

Ceph is a file system primarily created for object based storage. It features block storage as well and is currently in beta testing for file storage functionality. The following are instructions for using Ceph-Deploy to install a ceph cluster to a bunch of different servers. If you are trying to setup a dev environment, you will probably want to read over their documentation on the Ceph site as this is mainly for people wanting to know what goes where during the install. These directions will still work for you but some may be extraneous. Just for clarification, a **node** is a **server instance**, whether it is physical or virtual. Ceph architecture consists of OSD nodes and monitoring nodes. Monitoring nodes ensure the state of the cluster and keep the storage balanced(they do more, RTFM if you want more). The first monitoring node is considered the Admin node! OSD nodes are where the storage is at, either raid storage or JBOD. Read the documentation if you want more on them. What I have come to find out is that they only really recommend up to 24 OSDs per physical host. Now an OSD can be a raid array of disks or individual disks, so keep that mind. They also recommend you put the journal for each OSD on an SSD. Since I am setting storage servers in a JBOD setup with 30+ spinning disks, I am putting the journal on each disk. Refer to the ceph documentations wiki for instructions on placing the journal on different disks(yes there is a command that does it that is different from my create OSD command). The following instructions were created for the emperor release, you may need to modify the wget/package repository for a newer release. Also, shout out to the #ceph IRC channel that is normally full of very helpful people, though you may have to wait for assistance depending on what time it is. IRC info is listed on the ceph documentation site. Warning, be careful copying commands from this document.

1. Install Ubuntu server (do the following install for all nodes till step 10)
2. Update Ubuntu
    a. Sudo apt-get updates
    b. Sudo apt-get dist-upgrade
3. Install ssh (this is a selectable package that can be done while installing Ubuntu)
    a. Sudo apt-get install ssh
4. Configure SSH
    a. sudo pico /etc/ssh/sshd_config
    b. set PermitRootLogin yes
    c. set PermitEmptyPasswords yes
5. Install ntp (this is also a selectable package under manual package selection)
    a. Sudo apt-get install ntp
6. Configure NTP on main server
    a. Sudo pico /etc/ntp.conf
    b. Add iburst to the end of the first entry on the main time server
    c. On the secondary servers, comment out all time servers and add:
        i. Server main.server.name.here iburst
    d. Restart the services
        i. “sudo service ntp restart”
        ii. Verify time is working properly (ntpq – p)

7. Update hosts file
    a. Sudo pico /etc/hosts
        i. Add each server ip and corresponding names
            1. IE: 192.168.1.100 tpixary1 tpixary1.ceph.osd.mon.lc
            2. Make sure to have the short and long names
8. Add ceph user (username is **ceph** but can get be changed, change bold to reflect)
    a. sudo useradd -d /home/ceph -m **ceph**
    b. sudo passwd **ceph** (set password)
9. Add root priv to ceph user
    a. echo "**ceph** ALL = (root) NOPASSWD:ALL" | sudo tee /etc/sudoers.d/**ceph**
    b. sudo chmod 0440 /etc/sudoers.d/**ceph**
    c. **VERIFY it took**: sudo more /etc/sudoers.d/**ceph**
    d. Restart ssh: sudo service ssh restart
10. Create the ssh key so no password is required **(rest of the steps are performed on the admin monitoring node only!!) Reminder that the first monitoring node is also considered the Admin node as it contains the admin keyring needed to issue commands and authenticate servers.**
    a. Change to that user using SU
    b. Su ceph (enter password at prompt)
    c. ssh-keygen (keep pressing enter till its done)
11. Copy the ssh key to other nodes
    a. Ssh-copy-id ceph@node2
    b. Ssh-copy-id ceph@node3
12. Add and edit the local config for the ceph user
    a. Cd ~/.ssh/
    b. Pico config
        i. Add
            1. Host *
            2. User ceph
13. SSH over to another host to verify everything is working right
    a. “ssh <nodenamehere>” it shouldn’t prompt for anything except maybe to learn the ssh key of the host, no passwords though.
14. Go back to main user “exit” at prompt
15. Install ceph
    a. wget -q -O- 'https://ceph.com/git/?p=ceph.git;a=blob_plain;f=keys/release.asc' | sudo apt-key add -
    b. echo deb http://ceph.com/debian-emperor/ $(lsb_release -sc) main | sudo tee /etc/apt/sources.list.d/ceph.list
    c. sudo apt-get update && sudo apt-get install ceph-deploy
16. I would recommend changing back to ceph user login “su ceph” at this point to roll out the install to ensure fully working environment.

    a. NOTE: You can also specify a user in the ceph-deploy command using "ceph-deploy --username USERNAMEHERE" I wouldn't do this unless you are an expert on ceph!
17. Create a subdirectory under /etc call it whatever you want, probably best to use "ceph" or the cluster name….
18. Change directory into that folder, then proceed to next step..
    a. Make sure the ceph user has write permissions to the newly created directory!
        i. Under root, "sudo chmod –R 777 /etc/<directory> <username>"
            1. I also set the directory to be ceph owned "sudo chown –R <username> /etc/<directory>"
19. As the ceph user on the admin monitoring node(make sure you are in the newly created directory)!
    a. Reminder, the first monitoring node IS the admin monitoring node.
    b. Create cluster(do NOT use sudo to run commands, if command errors verify permissions on directory created in step 18!!!)
        i. ceph-deploy new <adminMONnodenamehere> <otherMONnode1> <otherMONnode2>
            1. Only list your monitoring nodes here
            2. The reason I specify two additional nodes is because you need to have a minimum of 3 monitoring nodes in a cluster in order to maintain a quorum so they don't fight when a debate about system status arises.
    c. Install ceph (make sure to list all your server names after the command with spaces)
        i. Ceph-deploy install <adminMONnodenamehere> <othenode1> <othernode2> etc…
            1. List all your nodes here(OSD nodes and monitor nodes)
            2. If you get an error string in the first few seconds from this command, rerun the permissions on the ceph directory again and restart ssh
                a. Its stupid and annoying as to why this doesn't seem to take right away, but if it still doesn't work, wait 5 minutes and try again.
    d. Add monitor
        i. ceph-deploy mon create <firstMONnodenamehere> <MONnode2namehere>
            1. Only your monitoring nodes should be listed
        ii. if you get a command run error, update the permissions on the ceph directory again
    e. Gather keys (wait a minute and rerun if you get an error)
        i. ceph-deploy gatherkeys <adminservernamehere>
        ii. if you get a command run error, update the permissions on the ceph directory again
        iii. To setup another server as an admin node: ceph-deploy admin <servername>
            1. This might require the –overwrite-conf switch to force copy to another server.
            2. Should make at least one other server an admin node.

f. Do a directory (ls –aslh) to verify all the keyrings(total of 3 unless mon then 4) are there.
g. (OPTIONAL) Add a MDS server (dosent have to be on admin server, if not then do this after you add the monitor server you want to put it on): "ceph-deploy --overwrite-conf mds create <monservername>"
    i. if you get a command run error, update the permissions on the ceph directory again

20. Create OSDs (These are the storage devices) (I am assuming you have whole hard disks to add here, you can also create a directory and use that)
    a. ceph-deploy osd prepare <hostname>:/dev/<drivehere> (IE: /dev/sdb)
        i. You might see an error here if the drive has never been provisioned before, just ignore it.
    b. Sudo ceph-deploy osd activate <hostname>:/dev/<drivehere>
    c. Note: You can do the first two steps with one command:
        i. ceph-deploy osd create <hostname>:/dev/<drivehere>
        ii. Example: "ceph-deploy osd create tpixary1:/dev/sdf"
        iii. You can also do multiple drives, just put a space at the end of that command followed by "ceph-deploy osd create tpixary1:/dev/sdf tpixary1:/dev/sdg tpixary1:/dev/sdh"
    d. NOTE: the cluster automatically adds and updates all the nodes when you add an OSD, no need to do anything other than run the commands to add each disk(or raid).
    e. Note: And you can also do multiple hosts:
        i. ceph-deploy osd create <hostname1>:/dev/<drivehere> <hostname2>:/dev/<drivehere> <hostname3>:/dev/<drivehere>
        ii. I recommend one host at a time to deal with any issues.
    f. NOTE: if you cannot add a disk because it is already partitioned or was partitioned by the OSD creation process and needs to be cleared, do the following:
        i. Discover the /dev/sd<letter>
        ii. Use the "parted"command to partition the disk
        iii. New Way
            1. Try Number 5 below first, then use 2-5 if that doesn't work.
            2. "parted /dev/sd<letter>"
            3. "Unit GB"
            4. "mklabel gpt"
            5. Run "ceph-deploy disk zap osdh:/dev/sd<letter>"
        iv. Old Way
            1. Print list to discover the available devices
                a. It may ask if the partitions are GPT, just say yes/confirm.
            2. Select /dev/sd<letter> to select the disk
            3. Print list to show the partitions
            4. rm partition# to remove the partitions

    a. Once all the partitions are cleared, you can now rerun the OSD creation command and it should work on that drive

21. Adding a node (after the fact)
    a. Ceph-deploy install <newnodenamehere>
22. Adding a Monitor (after the fact, be sure to run step 21 first)
    a. Edit the ceph.conf file and add (if its not already there)
    b. "public network = subnet/subnetmask" ie: 192.168.1.0/24
        i. Update the cluster(only if you modify the ceph.conf file): "ceph-deploy --overwrite-conf config push <node1> <node2><restofnodes>"
    c. NOTE: if you want to verify your monitors, you can look at the MON map:
        i. "ceph mon dump"
    d. REF: http://ceph.com/docs/master/rados/deployment/ceph-deploy-mon/
        i. Also covers removing monitors
    e. NOTE: the cluster automatically adds and updates all the nodes when you add a monitor, no need to do anything other than run the commands to add it.
23. Adding OSDs after the fact is the same as step 20, just make sure to run step 21 first.
24. Adding physical Disks to a running server and bringing them in without rebooting the server.
    a. First make sure scsi tools is installed "sudo apt-get install scsitools"
    b. Then run "sudo /usr/share/doc/sg3-utils/examples/archive/rescan-scsi-bus.sh"
    c. It should tell you there is a secondary drive now (or other drive you installed)
25. Remove OSDs(this is a manual process still):
    a. http://ceph.com/docs/master/rados/operations/add-or-rm-osds/#removing-osds-manual
26. Destroying a cluster
    a. Remove OSDs: "ceph-deploy disk zap <nodename>:sdb <nodename2>:sdb"
    b. Ceph-deploy purgedata <clustername>
    c. Ceph-deploy purge <clustername>
27. Shutting down the cluster (probably the most important information but not documented anywhere I could find!)
    a. Shutdown the monitor nodes first to ensure all the client connections are properly terminated.
        i. If you have a RADOS gateway, this needs to be shutdown first before the monitors.
    b. Shutdown the OSD storage nodes

NOTE: all keyrings are kept in each deamons directory, admin nodes have all keys, other nodes have only the keys they need.

Enabling CEPHX (enabled by default in emperor)(connection encryption, only for connection creation, the connection itself is not encypted… or in other words, the data being transferred is not encrypted in transit):
http://ceph.com/docs/master/rados/operations/authentication/#the-client-admin-key

Editing the CRUSH map (has everything about the cluster in it) Don't do this lightly unless you are just looking!
"http://ceph.com/docs/master/rados/operations/crush-map/"

Information on the CEPH networking system:
http://ceph.com/docs/master/rados/configuration/network-config-ref/

Infrastructure explanation of CEPH (very good!):
https://www.usenix.org/legacy/publications/login/2010-08/openpdfs/maltzahn.pdf

Scenarios(notes on various things):

1. Failure Detection
    a. Run this: "ceph health" , if you get something other than a timer being off or ok status, run this: "ceph health detail"
    b. Repair a page
        i. ceph pg repair <idhere>
    c. If you are using XFS and you lose your log, there is no current way to recover the OSD, you must remove the OSD and recreate it.
2. Disk failure – how to handle
    a. http://ceph.com/w/index.php?title=Replacing_a_failed_disk/OSD&redirect=no
    b. My current procedure
        i. unmount drive
        ii. replace drive
        iii. mount drive
        iv. ceph osd crush remove osd._
        v. ceph -s
        vi. ceph auth del osd._
        vii. ceph osd rm osd._
            1. wait till cluster is health OK and not backfilling/recovering.
        viii. ceph osd lost _
        ix. ceph-deploy disk list osdh_
        x. ceph-deploy disk zap osdh_:sd_
        xi. ceph-deploy --overwrite-conf osd prepare node14:sdi
        xii. ceph osd tree
    c. If you lost a bunch of drives:
        i. for i in {2..20}; do ceph crush remove osd.$i; done
            1. Be careful with this, don't use it for rm osd.x command unless you are sure your cluster can handle it

d. Count the number of online OSDs on a host
    i. ps aux |grep ceph |grep osd |wc –l
e. Start osd manually with debug enabled.
    i. sudo ceph-osd -i <idhere> -d -f --debug_ms 1 --debug_osd 20 --debug_filestore 20
f. XFS checks
    i. xfs_check OR xfs_repair /dev/sd<driveletterhere>1 (must unmount /dev/sd first)
    ii. Must download and install tool first, Ubuntu prompts for it.

3. Host Failure – how to handle
    a. If you cannot bring the machine back online to a running state in which it previously existed, IE the disk is bad, best to just remove it from the cluster on the admin node as well as all of its associated objects.
4. Full Cluster – how to handle
    a. Call a consultant, you are in deep if this a production environment.
        i. Off the wall suggestion would be to power down the monitor nodes, shutdown your OSD nodes, fix whatever got messed up, then bring the OSD nodes back online and once they are happy, then bring the monitor nodes back online. I take no responsibility for this!! Most likely a LOT of replication will occur after this depending on how messed up things got, you might want to wait a while to provide access to the data, keep checking ceph –w
5. Info on failures:
    a. http://eu.ceph.com/docs/wip-3060/ops/manage/failures
6. Monitoring the cluster:
    a. http://ceph.com/docs/master/rados/operations/monitoring/
7. Upgrading
    a. Using Ceph-Deploy via the ceph user
        i. Update the packages using step 15, just make sure to change the website directory in step 15.2 to match the version… IE the old one was emperor, the new one is firefly, so change the primary directory to firefly
            1. echo deb http://ceph.com/debian-firefly/ $(lsb_release -sc) main | sudo tee /etc/apt/sources.list.d/ceph.list
            2. Make sure to start on a monitor server, I recommend the adminserver or first server setup in the cluster.
        ii. Run “ceph-deploy install <adminservername>” and you will see it update the local client and then at the end show version it upgraded to. Run “ceph –v” to verify the version running, no need to restart the monitor.
            1. Unfortunately, as of .80.5, ceph-deploy doesn’t restart the OSDs, so the only way is to go to each host, login as a plain user, run:
                a. “Sudo restart ceph-osd-all” (warning!!)

i. Only execute this on 1 osd host at a time, if an OSD fails, you will need the online replica to fix the bad OSD, which means recreating the bad OSD, letting the system rebalance the data lost and then moving on to the next OSD host.
    1. This is annoying cause an update killed a bunch of my OSD journals and thus the disks and it was across many osd hosts.
b. Can also do “sudo stop ceph-all” or “sudo start ceph-all” or “sudo stop ceph-osd-all” which is monitor specific. Or OSD specific can be “stop ceph-osd id=idofosd”

iii. Repeat last step for each monitor server, then proceed with the OSD hosts and finally the radosgw if you have one. If you have an MDS server, that would after everything(or last on the list to upgrade).
    1. Verify osd version from admin server using this command:
        a. “ceph tell osd.X version”

8. Speed Testing (block level at the cluster using RBD) using ceph user
    a. Sudo rbd create test –size 20000
        i. Test is the datastore name
    b. Sudo rbd map test
    c. sudo dd if=/dev/zero of=/dev/rbd1 bs=1024k count=1000 oflag=direct
        i. This writes a large file and then provides the stats.
    d. sudo dd if=/dev/rbd1 of=/dev/null bs=1024k count=1000
        i. This reads the newly created large file and provides the stats
    e. ceph osd tell osd.N bench
9. Speed Testing (using the rados Gateway server) user sudo under local user
    a. Writes testing
        i. rados -p data bench -b 4194304 60 write -t 1 --no-cleanup
            1. data is the pool name
    b. Sequential testing
        i. rados -p data bench -b 4194304 60 seq -t 1 --no-cleanup

Setting up RADOS gateway

This section covers the setup of RADOS gateway on a separate node. The purpose of the gateway is to allow Amazon S3 like access to the Ceph storage object storage system. I will also be covering some test utilities and basic user setup as well as access. Best to have at least two of these in a production environment with a load balancer up front, I was told dreamhost was using a total of 5 gateway nodes for a few petabytes of info, so note that they can handle a lot. These steps are following these steps: http://ceph.com/docs/next/radosgw/config/ but I am essentially translating what they heck they are doing and filling in the gaps for my own documentation.

1. On the Admin node, edit the ceph.conf file and add the following(after the main statement):

```
[client.radosgw.<nodenamehere>]
    host = <nodename>
    keyring = /etc/ceph/keyring.radosgw.<nodenamehere>
    rgw socket path = /tmp/radosgw.sock
    log file = /var/log/ceph/radosgw.log
    rgw dns name = <nodename.domainname.ext>
```

   a. Node name is what you called the RADOS gateway server and have listed in the /etc/hosts file on all the ceph nodes.
   b. For the "rgw dns name", this one is a little bit more complicated. The domain name will be what you are using to reference the RADOS gateway from external servers. So if you own "awesomedude.com" and you plan to host the files via "awesomedude.com", then the domainname.ext would be "awesomedude.com". The more complicated piece of this is the nodename. If you are going to expose the outside world to your nodename, then just leave it as your RADOS gateway node name. If you are going to create a mask, say "storage.awesomedude.com" then your rgw dns name entry would be "storage.awesomedude.com". This is useful if you plan to load balance a single IP. In the load balancer config, you can use an ACL to point to the storage.awesomedude.com internal IP when external people reference it.
   c. The final piece of this is the fact that when you access buckets on the RADOS gateway, you will need a catch all entry. The way amazon works is by using <bucketname>.<nodename>.<domainname>.<ext>. So if you are using "awesomedude.com" and your bucket was named "foo", when you access the bucket "foo", the URL will look like this: "foo.storage.awesomedude.com". Obviously if you have more than one bucket, your entry could be secondbucket.storage.awesomedude.com, so its important to know if you are going to use more than one bucket. To allow unlimited buckets, you will need that catch-all DNS entry for the storage.awesomedude.com (this is also helpful for the fact that you will be creating a catch-all for just "storage.awesomedude.com" instead of "awesomedude.com" which could affect other named prefixes you might want to use later. Hopefully this makes sense.

i. If you are only going to use your RADOS gateway internally, then when you edit your internal DNS servers entries, this problem is solved in the same manner.

d. This part is very important, so make sure to know what you need. I think my method is the best setup, but if you know what you are doing, feel free to do whatever.

2. If you have added the RADOS gateway node after the initial install, meaning it was never deployed to during the install process in steps 1-20, you can use steps 1-9 and then 21 to install Ceph to the RADOS node. If it was part of the original install and you completed steps 1-9 and 19c on this server, then there is nothing to do here.
3. Push the configuration out to all the Ceph nodes: "ceph-deploy --overwrite-conf config push <node1> <node2> <nodes3>" (this is done from the admin node)
4. The following steps are completed on the RADOS gateway node:
    a. Create a directory: "sudo mkdir -p /var/lib/ceph/radosgw/ceph-radosgw.gateway"
    b. Install Apache with special apps RADOS needs:
        i. sudo echo "deb http://gitbuilder.ceph.com/libapache-mod-fastcgi-deb-precise-x86_64-basic/ref/master precise main" >> /etc/apt/source.list.d/ceph.list
        ii. sudo apt-get update && sudo apt-get install apache2 libapache2-mod-fastcgi
        iii. Enable modules in apache(run the following commands):
            1. a2enmod rewrite
            2. a2enmod fastcgi
    c. Install RADOS gateway package: "sudo apt-get install radosgw"
    d. Configure Apache
        i. In /etc/apache2/sites-available create a file called "rgw.conf"
        ii. Fill it with the following("sudo nano rgw.conf"):

```
FastCgiExternalServer /var/www/s3gw.fcgi -socket /tmp/radosgw.sock
        <VirtualHost *:80>
            ServerName <nodename>.<domainname>.<ext>
            ServerAlias s3.<domainname>.<ext>
            ServerAlias *.<domainname>.<ext>
            ServerAlias storage.<domainname>.<ext> #please see notes about using the storage.domainname.ext entry
            ServerAdmin <email.address>
            DocumentRoot /var/www
            RewriteEngine On
            RewriteRule ^/([a-zA-Z0-9-_.]*)([/]?.*) /s3gw.fcgi?page=$1&params=$2&%{QUERY_STRING} [E=HTTP_AUTHORIZATION:%{HTTP:Authorization},L]
            <IfModule mod_fastcgi.c>
                <Directory /var/www>
                    Options +ExecCGI
                    AllowOverride All
                    SetHandler fastcgi-script
                    Order allow,deny
                    Allow from all
                    AuthBasicAuthoritative Off
                </Directory>
            </IfModule>
            AllowEncodedSlashes On
            ErrorLog /var/log/apache2/error.log
            CustomLog /var/log/apache2/access.log combined
```

```
        ServerSignature Off
</VirtualHost>
```

1. Make sure the Rewrite rule is all on the same line.

iii. Enable the site: "sudo a2ensite rgw.conf"

iv. Disable the default site: "sudo a2dissite default"

e. Create FastCGI script referenced in the rgw.conf file:

i. Cd /var/www

ii. Nano s3gw.fcgi

iii. Paste this into the file(note that the node name is the simple host name of the RADOS server(IE: rados1.awesomedude.com would be "rados1" so the below entry would read "client.radosgw.rados1"):

```
#!/bin/sh
exec /usr/bin/radosgw -c /etc/ceph/ceph.conf -n client.radosgw.<nodename>
```

iv. Make the file executable: "sudo chmod +x s3gw.fcgi"

5. Next is the fun part, the creation of the access keyring for the RADOS gateway node, the following steps take place on the Ceph cluster Admin Node

a. Change directory to the /etc/<clustername> folder where all your keyrings are at. Most likely your cluster name is ceph, so the folder would be the ceph folder. This folder was created above in step 17.

b. Create a keyring for your gateway(change gateway to your gateways simple hostname):

i. "sudo ceph-authtool --create-keyring keyring.radosgw.<gateway>"

ii. Add read to it: "sudo chmod +r keyring.radosgw.<gateway>"

c. Generate a key so the gateway can authenticate with the cluster

i. "sudo ceph-authtool keyring.radosgw.<gateway> -n client.radosgw.<gateway> --gen-key"

ii. "sudo ceph-authtool -n client.radosgw. <gateway> --cap osd 'allow rwx' --cap mon 'allow rw' keyring.radosgw.<gateway>"

iii. ceph auth caps client.radosgw.<gateway> mon 'allow rw' osd 'allow rwx'

1. writing permissions to auth list a second time

a. "ceph auth list" to verify your rados gateway server is listed with permissions

d. Add keyring entries to ceph storage admin keyring

i. "sudo ceph -k ceph.client.admin.keyring auth add client.radosgw.<gateway> -I keyring.radosgw.<gateway>"

e. Copy the keyring to your RADOS gateway node to the:

i. "scp keyring.radosgw.<gateway> <username>@<radosnode>:/etc/<ceph>"

1. Note that the /etc/<ceph> directory should be the same name you set in step 17.

6. On the RADOS gateway node, restart the necessary services:

a. "sudo service ceph restart"

b. “sudo service apache2 restart”
c. “sudo /etc/init.d/radosgw start”

7. Take a look at the RADOS log to verify all is well:
    a. “more /var/log/ceph/radosgw.log”
        i. You might see something about buckets being created because they aren’t there, that is fine, emperor release sets them up now automatically.
        ii. If you see any authentication errors, make sure the keys on both sides are readable by the services (IE: chmod –R 666 /etc/ceph)
        iii. I also set the keyring.radosgw.<gateway> file to be owned by www-data (“chown www-data /etc/ceph/keyring.radosgw.<gateway>”)
        iv. This part is very tricky, so if the keyrings were not created properly, the RADOS service will never connect. The DNS must also be setup properly so as to ensure the apache server will answer.
8. Make sure to edit the /etc/hosts file on the RADOS node and add “storage.<domainname>.<ext> if you are going to use that method.
9. On your DNS server(s), you need to take authority for your outside domain name that you are using internally for the Ceph gateway setup.
    a. Edit your named.conf.default-zones and add an entry for your outside domain name(we will use awesomedude.com).
        i. Files are in /etc/bind/

```
zone "awesomedude.com" {
        type master;
          file "/etc/bind/db.awesomedude.com";
};
```

        ii. Save the file.
    b. Create a file for your zone that you just referenced above (db.awesomedude.com)
    c. “nano db.awesomedude.com”
    d. Fill it with the following:

```
@ 86400 IN SOA awesomedude.com. root. awesomedude.com. (
                20091028 ; serial yyyy-mm-dd
                10800 ; refresh every 15 min
                3600 ; retry every hour
                3600000 ; expire after 1 month +
                86400 ); min ttl of 1 day
@ 86400 IN NS awesomedude.com.
@ 86400 IN A x.x.x.1
@ 86400 IN A x.x.x.2
* 86400 IN CNAME @
```

    e. Exchange awesomedude.com with your domain name
    f. Exchange x.x.x.x with the local IP of your RADOS gateway node
        i. You can add multiple entries to round robin the connections.
    g. Make sure not to delete any of the trailing periods

h. Restart bind service (service bind9 restart)

i. Make sure all your servers/clients are pointed to your DNS server(s)

10. Next we will create a RADOS gateway user on the RADOS node

```
sudo radosgw-admin user create --uid="{username}" --display-name="{Display Name}"
```

a. IE: sudo radosgw-admin user create --uid="test" --display-name="Test User"

b. It will then spit out information you need and MUST save immediately somewhere else you can later reference!!!!!!


```
{ "user_id": "test",
            "rados_uid": 0,
            "display_name": "Test User",
            "email": "test@example.com",
            "suspended": 0,
            "subusers": [],
            "keys": [
              { "user": "test",
                "access_key": "QFAMEDSJP5DEKJO0DDXY",
                "secret_key": "iaSFLDVvDdQt6lkNzHyW4fPLZugBAI1g17LO0+87"}],
  "swift_keys": []}
```


c. You need to save the uid, the access_key and the secret_key !! DO NOT LOSE THIS INFO!

11. Next we will install the s3cmd tools to test out our new RADOS gateway node:

a. "Sudo apt-get install s3cmd"

12. Time to test the gateway (from the gateway node using the s3cmd client).

a. First, open a browser from any machine pointed to the modified DNS server(s), go to storage.awesomedude.com (replace with your info) and you should see a something like this:

```
This XML file does not appear to have any style information associated with it. The document tree is shown below.
<ListAllMyBucketsResult xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <Owner>
   <ID>anonymous</ID>
   <DisplayName/>
  </Owner>
  <Buckets/>
</ListAllMyBucketsResult>
```

b. If you see this, then your gateway is working!

c. Now, back to the RADOS node, on the command line, run the following:

i. "s3cmd --configure"

1. This will ask you for the information you saved above in step 10 of the RADOS install.
2. Good Reference: http://s3tools.org/s3cmd

ii. Next you need to edit the newly created file:
    1. "nano ~/.s3cfg"
    2. Change the host_base and host_bucket entries to reflect your domain information.. IE:
        a. host_base = awesomedude.com
        b. host_bucket = %(bucket)s.awesomedude.com
            i. Note that I didn't include the storage prefix!
    3. Save the file
iii. Now at a command prompt, run "s3cmd ls"
    1. You shouldn't get anything back because you haven't created any buckets but you shouldn't get an error either!
iv. Create a bucket
    1. "s3cmd mb s3://foo"
        a. It should say the bucket "foo" was created.
v. Next run the list command again: "s3cmd ls"
    1. Should display the bucket you just created!
vi. See the reference page I listed above for commands you can run.

d. So now that you have a bucket, to reference it in your connection tools, you would be referencing "foo.storage.awesomedude.com".

13. FIN (holy crap that was a lot, I hope they find a way to shorten this)

Adding a second RADOS Gateway (or more)

1. Clone your existing gateway if it's a virtual or reinstall following steps 1-5
    a. For a clone:
        i. Do step 5
        ii. Edit /etc/apache2/sites-available/rgw.conf
            1. Change the servername to the new servers hostname and save
        i. Edit / var/www/s3gw.fcgi
            2. Change the client.radosgw.<nodename> to match the new hostname and save the file
2. Push configuration out to host from main admin node
3. On the new RADOS gateway node, restart the necessary services:
    a. "sudo service ceph restart"
    b. "sudo service apache2 restart"
    c. "sudo /etc/init.d/radosgw start"
4. "more /var/log/ceph/radosgw.log" to verify no problems exist

Current Configuration

```
[global]
fsid = <generated>
mon initial members = mon1, mon2, mon3
mon host = mon1, mon2, mon3
auth cluster required = cephx
auth service required = cephx
auth client required = cephx
filestore xattr use omap = true
public network = 10.10.0.0/20
cluster network = 10.10.33.0/20
#mon warn on legacy crush tunables = false
osd pool default size = 3  #Write an object 3 times( default now ).
osd pool default min size = 1 #Allow writing one copy in a degraded state.
osd pool default pg num = 4096
osd pool default pgp num = 4096
#Disabled optimizations ( defaults in newest ceph versions are better now )
#filestore op thread suicide timeout = 360
#filestore op thread timeout = 180
#filestore max sync interval = 25
#filestore min sync interval = 5

[osd]
#osd mkfs options xfs = "-f -i size=2048"
osd mount options xfs = "rw,noatime,inode64,allocsize=4M"
osd journal size = 20000
#disabled Optimizations ( defaults in newest ceph versions are better now )
#osd op threads = 4
#osd max backfills = 2
#osd recovery max active = 4
#debug osd = 20
#debug filestore = 20
#debug ms = 1

[client.radosgw.rados1]
host = rados1
keyring = /etc/ceph/ceph.client.radosgw.keyring
rgw_socket_path = /tmp/radosgw.sock
log_file = /var/log/ceph/radosgw.log
rgw dns name = storage.awesomedude.com
rgw thread pool size = 512
```

rgw cache lru size = 50000 # the default is 10000

[client.radosgw.rados2]
host = rados2
keyring = /etc/ceph/ceph.client.radosgw.keyring
rgw_socket_path = /tmp/radosgw.sock
log_file = /var/log/ceph/radosgw.log
rgw dns name = storage.awesomedude.com
rgw thread pool size = 512
rgw cache lru size = 50000 # the default is 10000

[client.radosgw.rados3]
host = rados3
keyring = /etc/ceph/ceph.client.radosgw.keyring
rgw_socket_path = /tmp/radosgw.sock
log_file = /var/log/ceph/radosgw.log
rgw dns name = storage.awesomedude.com
rgw thread pool size = 512
rgw cache lru size = 50000 # the default is 10000

Created by Brent at http://blog.scsorlando.com with the help of **MANY** people online both paid and unpaid!

Updated 1/4/15