Ceph Install / Deployment in a Production Environment (Ubuntu/Debian)

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 update
    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(no longer needed in Ubuntu 14.04, skip step if using)
    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
        i. I also add time.nist.gov (or add another one of your favorite internet time servers) so ensure there is a longer list available to the server as a bad actor could knock out syncing for a bit.
    c. On the secondary servers, comment out all time servers and add:
        i. Server main.monitor.server.name.here iburst

d. Restart the services
    i. “sudo service ntp restart”
    ii. Verify time is working properly (ntpq – p)
e. Side Notes
    i. If you are running the monitor servers on virtual machines, make sure the host server is synced so that when the virtuals boot, their system time is synced.
        1. You could always set the host server to sync with the main monitor server.
        2. VMware workstation 11 doesn’t sync with host by default
    ii. If you are running the monitor servers on physical machines, make sure they sync their time with the servers bios.
    iii. Always check the cluster for issues when you reboot machines, clock skew is usually quick to resolve but it’s a high priority issue.
    iv. If the ntp services keep rejecting your primary server, you can add “fudge main.monitor.server.name.here stratum 1” to the ntp.conf under the iburst line to ensure it stops rejecting your server and then restart ntp service. It may take a few times before sync occurs.

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 node deployed from is the Admin node as it contains the admin keyring needed to issue commands and authenticate servers(you can push this keyring to other nodes to make them admin nodes).** Also note that when you are deploying configurations or anything that pushes configurations(like installs), you need to be under the ceph user and have that user currently in the /etc/ceph(may be different if you changed it) folder when running the command.
    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. Removed this steps instructions as they are no longer necessary(11/15), leaving in due to other references in the doc.
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 (only install from the official ceph repository, not official Ubuntu ones, because they don’t update them as they release newer versions of Ubuntu).
    a. If updating, remove repositories for ceph
        i. cd /etc/apt/sources.list.d/
        ii. sudo rm -rf ceph.list ceph-deploy.list
    b. wget -q -O- 'https://download.ceph.com/keys/release.asc' | apt-key add - | sudo apt-key add -
    c. echo deb http://download.ceph.com/debian-hammer/ $(lsb_release -sc) main | sudo tee /etc/apt/sources.list.d/ceph.list
    d. sudo apt-get update && sudo apt-get install ceph-deploy
        i. If you need an older (or newer) version change the release name, “hammer” in my case above, perhaps to “Jewel”. You cannot downgrade after install.
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> ”
            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 if you deploy from there or wherever you deployed ceph from using ceph-deploy.
    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.
3. After running the command, do a directory and you will see the newly created keyrings, log and configuration file.
4. You can run hostname{1,2,3} if your hostnames are the same but numbered, which will allow you to save typing during these commands

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.
      3. To install specific release of ceph, ceph-deploy install --release <releasenamehere> <adminMONnodenamehere>
         a. Release name IE: giant or firefly or dumpling
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
   ii. make sure to run a "ceph –s" to verify the cluster is healthy. You may have to address clock skew (see notes section for clock skew handling).

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) Please note that will want to review a

standard ceph.conf file and modify any settings you want to use on the OSDs in that file so that when you run the commands below OSDs are created with those settings(very important). Make sure to push the new configuration to all the ceph nodes once its been updated so that when you create the OSDs, they are setup with the required parameters and defaults. (you will need to bounce the monitors so they pull in the new configuration, prior to the push so that when pools are created they are using your new defaults)

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. 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
g. NOTE: Add osd with separate journal file or drive: "ceph-deploy osd create <hostname4>:/dev/<drive>:/dev/<drive or folder>"

i. This is the command you will be using if you need to create osds with their journal files on SSDs (or in my case, partitions on SSDs).
ii. You may need to overwrite the config file to create, simply add "--overwrite-conf" after ceph-deploy (IE: ceph-deploy --overwrite-conf osd create)

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/Powering down the cluster (probably the most important information but not documented anywhere I could find!) Usually for moving equipment…
    a. Shutdown
        i. Put the cluster in maintenance mode ("ceph osd set noout") (only if health is listed as OK, otherwise fix or let the cluster resolve issues before doing this)
        ii. If you have a RADOS gateway, these need to be shutdown first
        iii. Shutdown the monitor nodes first to ensure all the client connections are properly terminated.
            1. Stops all IO, important to ensure nothing is moving around.
        iv. Shutdown the OSD storage nodes
    b. Start up
        i. Start the MON nodes
            1. Verify all monitors are online and there is no clock skew

ii. Start the OSD nodes
iii. Check ceph with ceph –s (fix things if need be) (“ceph health detail” for details if anything shows or ceph –w for continuous feed)
iv. Start RADOS gateway nodes
v. Take the cluster out of maintenance mode (“ceph osd unset noout”)

28. Maintenance Mode
    a. If you need to restart an OSD or a host or need to affect something in the cluster without invoking a recovery…
        i. Run a “ceph osd set noout”
            1. To undo this when done: “ceph osd unset noout”

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.
    d. If a disk is offline, but your raid tools show no problems with the drive, you can attempt to bring the drive back online

        i. On the OSD server, run “start ceph-osd id=x”
2. OSD failure – how to handle
    a. http://ceph.com/w/index.php?title=Replacing_a_failed_disk/OSD&redirect=no
    b. My current procedure (we are using individual drives as OSDs, hopefully you use RAID, so a disk replacement is as simple as swapping a drive). Always try to restart the OSD(start ceph-osd id=# on osd host server) first before proceeding(unless the drive is blinking red, then you know its dead).
        i. Hardware (if using individual drives as OSDs or a complete RAID failure case).
            1. General Steps
                a. unmount drive
                b. replace drive
                c. mount drive
            2. Step with my hardware on current dell hardware using MegaCli (applicable to many linux installs using an LSI raid card).
                a. Before beginning, few commands
                    i. “df –h” to check to see what drive is the problem (ceph only shows it in general commands as OSD #). Running the ceph osd tree command lets you determine which OSD it is and on what host. Now that you are on the host, ready to fix the drive, you can use the df –h command output to determine which logical drive letter it is that went bad on the host(assuming you didn’t document which OSD is what sd on the host).
                    ii. If the drive is still mounted (this is the case most of the time), you will want to make sure the ceph osd daemon is not running “ ps –ef |grep ceph”, assuming the OSD is 7, then you should not see a “/usr/bin/ceph-osd --cluster=ceph -i 7 –f” in the list. You can also use ps aux(it’s the same character for me and –ef is a habit from my AIX days).
                    iii. Next umount the drive, df –h gives you the output (assuming osd is #7): /var/lib/ceph/osd/ceph-7 then “umount /var/lib/ceph/osd/ceph-7” and it should unmount the drive. If you get an error, then you must kill any processes that are holding it in use. Usually its another SSH session, just kill any SSH sessions on the host that are not yours! Run the who command and then “pkill –KILL –t pts/#” any sessions that are not yours. Try umount again. If successful, then you can move on!

1. Note: if this doesn't work, you can proceed with removing the drive and then before making it live again, run the umount command again to see if it works(it should since the drive was removed)).

iv. I don't use alias for anything normally, so I didn't look into why this is, but I must add the MegaCli command back each time I log in. I know you can add it to the shell, but with multiple servers and limited time, this just hasn't been a priority.
    1. If you haven't added it to the shell, you must add the alias before running the MegaCli commands(unless you want to pass the location of the bins each time you run the command).
        a. So here is the alias command: alias MegaCli="/opt/MegaRAID/MegaCli64"
            i. Note that this path is where MegaCli installs on Ubuntu 12/14 by default.
            ii. If you haven't installed it(I have on all servers), then you will need to find instructions and download from LSI(assuming you have an LSI raid card).
    2. Next is the command to check the status of all the drives, I find this very useful, especially when figuring out which slot died(since its not in the list when run)
        a. "MegaCli –PDList –aALL |more"
            i. I also tend to grep it to see if there is any offline, ie: MegaCli –PDList –aALL |grep "Offline" (or "Online" for whats online), you could also play around and add more lines. Or use "slot" to see what the slot list is instead of going

through the whole command output.

b. Hopefully you documented which serial is in which slot, because otherwise you have to find it first if the disk is not dead(light is red). If its red light dead, then skip down to iii.
    i. If you need to locate(light isnt red, but you know what slot, just not where the slot is) then here you go:
        1. MegaCli -pdlocate -start -physdrv[<controllerIDhere>:<IDHERE>] -a0
            a. (IE: MegaCli -pdlocate -start -physdrv[12:3] -a0) in this case the raid controller is 12 and the slot is 3.
                i. Note, our current servers start with slot 0 being the bottom right drive.
        2. OR use “cat /dev/sdb >/dev/null” to create activity on the disk so you can see which light is blinking furiously.
    ii. Since the drive is not noted as dead(red light), you must then tell the raid system you intend to destroy the drive with the following commands(once located).
        1. Tell the raid system that slot 3 on raid controller ID 12 is being taken offline (just continuing with the example IDs above)
            a. MegaCli -pdoffline -physdrv[12:3] -a0
        2. Mark the drive missing
            a. MegaCli -pdmarkmissing -physdrv[12:3] -aAll
        3. Prepare the disk for removal
            a. MegaCli -pdprprmv -physdrv[12:3] -a0
    iii. Now you can pull the drive out, replace with a new drive and then put back in the slot.
    iv. Next check dev to see if there is a dead entry for the old disk: ls –l /dev/sd* (it wont be colored), if there is simply delete it to ensure when the new drive is activated, it gets setup properly (rm /dev/sd<letterhere>).

1. You can force a disk rescan via this command:
    a. for i in /sys/class/scsi_host/*; do echo "- - -" > $i/scan; done

v. Once the drive is swapped out and the drive light is green, you can then run the following command to add the drive back (and online).
    1. Continuing with the example above where the raid controller ID is 12 and the slot is 3
        a. MegaCli -cfgldadd -r0 [12:3] -a0
            i. Please note that this adds a Raid 0 drive back to slot 3 (if you are not using raid 0 or a setup like mine, you would need to alter this command to include the other drives and a different raid type).
    2. Run the pdlist command to verify the drive is back and online.
        a. MegaCli –pdlist –aALL |more
        b. Or MegaCli –pdlist –aALL | grep slot (to see a list of slots)
            i. Use grep –C 3 to include 3 lines above and 3 lines below the filter word. Use –B for before only and –A for after only

ii. Ceph Cluster
    1. Set noout (“ceph osd set noout”) and noscrub (“ceph osd set nodeep-scrub “) on the cluster
        a. Just a note, setting noout prevents additional osd’s from dropping if they become unresponsive during the backfilling. Having a separate network usually keeps this from being an issue, so its more of precautionary measure than anything else. I mainly worry about a cluster collapse(partial or full) during a rebuilds, but that hasn’t happened with newer hardware. As for disabling deep scrub, this actually has a function to prevent an intensive process from occurring while backfilling is done. Since the resources are not wasted on this function during a backfill situation(we don’t need to verify integrity while we are

trying to restore functionality), the backfilling should finish much quicker.

2. ceph osd crush remove osd._
3. ceph –s
    a. wait till the backfilling/recovery finishes and degraded status drops off. Hopefully you are back to Health OK.
4. ceph auth del osd._
5. ceph osd rm osd._
    a. check to make sure cluster is not backfilling/recovering.
6. ceph osd lost _
    a. Should tell you its not there, but run it anyway.
7. Change over to the ceph user created when the cluster was created and change directory to the /etc/ceph folder. Run these commands as that user(or any time you see ceph-deploy command).
    a. ceph-deploy disk list osdh_
    b. ceph-deploy disk zap osdh_:sd_
    c. ceph-deploy --overwrite-conf osd create <osdhost#>:<diskSD> (IE: osdhost4:sdb)
        i. If you are using SSD's for journal drives, then the command is expanded to include the SSD or partition on SSD
            1. IE: ceph-deploy –overwrite-conf osd create <osdhost#>:<diskSD>:/dev/<journaldiskSD> (IE: osdhost4:sdh:/dev/sdm6)
                a. This is assuming you partitioned the SSD and are placing the journal on different partitions.
                    i. Note: max of 5 osds per SSD is recommended.
            2. Since older versions of ceph-deploy don't output which disk they are using for the journal, you had to either document it or run the following command on the OSD host to hopefully list them:
                a. ls -alh /var/lib/ceph/osd/ceph-* |egrep -wi --color 'ceph|journal'
                    i. This provides the OSD number(ceph-#) and the journal directory output which lists the /dev/sd info on the next line.

ii. Run a df –h to fully correlate what is where.
iii. If any directories don't have a correlating journal drive, they may be old failed osd folders that you should delete(after verifying they are indeed empty).
iv. This command sometimes gives me the UUID of the journal drive instead of the friendly name, so you then must go do a ls –l /dev/disk/by-partuuid to correlate it to the SSD partition.

8. ceph osd tree (escape out of ceph user and run this).
9. Ceph –s to see the rebuilding begin, I don't recommend any action until you are back to health ok
10. Now unset the noout and nodeep-scrub flags to return the cluster to normal operation.

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. Clock Skew
    a. Restart the ntp services on the host with the skewed clock, then run the ntpq –p command until you see the offset drop below +/- 10 seconds. The clock skew message should clear once everything is relatively close.
    b. If after syncing the clocks and the offset is below 1 second, you may have to restart the monitor server. There is a bug that sometimes causes the clock skew report(still a problem as of 2/7/2016).
    c. Just a note, the cluster checks for clock skew every 5 minutes, so once you have everything in sync, the skew message should go away within 10 minutes. If not, see step b above.
    d. If you get a situation where the other servers wont talk to the main ntp server you setup, then reboot the ones that are saying connection refused. This relates the issue noted in 5b above.
6. Info on failures:
    a. http://eu.ceph.com/docs/wip-3060/ops/manage/failures
7. Monitoring the cluster:
    a. http://ceph.com/docs/master/rados/operations/monitoring/
8. 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. If updating, remove repositories for ceph
                a. cd /etc/apt/sources.list.d/
                b. sudo rm -rf ceph.list ceph-deploy.list
            2. wget -q -O - 'https://.ceph.com/git/?p=ceph.git;a=blob_plain;f=keys/release.asc' | sudo apt-key add -
            3. echo deb http://download.ceph.com/debian-giant/ $(lsb_release -sc) main | sudo tee /etc/apt/sources.list.d/ceph.list
            4. sudo apt-get update && sudo apt-get install ceph-deploy
            5. 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"

9. 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
10. 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
11. Repairing a pg
    a. If you have an error stating that pgs are inconsistent, then you need to run a ceph health detail to get the pg numbers (note that the numbers before the period is the OSD number)
        i. Then run "ceph pg repair <pg_num_id>" for each one that is having an issue.

1. If you notice that you are repairing a lot of inconsistent pages from one osd, then I recommend setting the “ceph osd set noout” flag on the cluster and restarting the osd daemon.
    a. To restart the daemon, run a “ceph osd tree”, note which host the osd is on, then go to the host and run “stop ceph-osd id=<osd_num>”, then once its complete, just run the “start ceph-osd id=<osd_num>” to kick it back on.
    b. Next go back to the management server and run the “ceph osd tree” command until you see the osd noted as “up” again. Then you may want to run a ceph –s until you see the repair process finish. Once the cluster is back to normal(probably still has inconsistent pages, which is fine), then run the “ceph osd unset noout” command to take the cluster out of maintenance mode. Now run the “ceph pg repair <pg_num_id>” command on the inconsistent page groups.

ii. An easier way, if you have a bunch or are just lazy is to use this command: “ ceph pg dump | grep -i incons | cut -f 1 | while read i; do ceph pg repair ${i} ; done “

12. Drives filling up
    a. Seems even with the auto balancing that is supposed to take place, there are times when drives start to get full ahead of others and you have reweight individual drives.
        i. Try first to run the “ceph osd reweight-by-utilization” command, then you can try “ceph osd reweight-by-utilization 110” 110 can be increased as needed, but if that doesn’t work out you can target the OSDs individually.
        ii. Log on to each osd host and run a “df –h” to see which drives are above the 85% mark, which is the default near full status mark.
        iii. Once you have identified the OSD that needs to be reweighted
            1. “ceph osd reweight 16 0.9084”
            2. So essentially 16 is the OSD number and the 0.9084 is the reweight I want applied, which is between 0 and 1, you want to decrement this slowly, but I have found that this number gets me down to where I want an OSD the first time.
            3. Also, I tend to set noout before running the reweight just incase something bad happens while this process is going on, which the time varies by drive size. 4TB takes a few hours per drive. I also recommend only doing one at a time. So best to hit near full drives as they come up. Or ask your dev team to delete their crap.
    b. Sometimes you get a drive nearing full that will not allow for backfilling, this is problematic because you cannot finish the reweight. To fix this TEMPORARILY, inject some temp variables then once you are down, inject them back to default

i. First up to fix the near full " ceph tell osd.* injectargs '--mon_osd_nearfull_ratio 0.87' " , note the commas, default is 0.85 and that's what you are going to set them back to once you are done.
ii. Next fix the backfill reserve (only if it shows the backfill blocked): "ceph tell osd.* injectargs '--osd_backfill_full_ratio 0.87' " , again note the commas, default is 0.85
1. Note that if you inject an argument that you want to stick, you need to update the ceph.conf and send it out to all the servers so that if they are reset, they don't lose the change.

c. Lastly, when doing these actions, along with setting the noout setting, its also recommended that you disable deep scrubbing:
i. "ceph osd set nodeep-scrub " and then "ceph osd unset nodeep-scrub " to re-enable it.

d. Useful commands for this
i. "ceph df" see pool sizes
ii. "ceph osd df" see all osd sizes and info
iii.

13. Stuck, unclean, incomplete pgs (rare issue)
a. There could be an issue where you have pgs that are listed at stuck, unclean and incomplete where the repair doesn't work. Usually there is the same number listed for each
i. IE: HEALTH_WARN 64 pgs incomplete; 64 pgs stuck inactive; 64 pgs stuck unclean

b. Run a "ceph health detail" and note which osd in each line item [x,x,x] repeats. Sometimes all three repeat (or two or more, depends on your cluster replication count).
c. Set the "ceph osd set noout" flag on the cluster
d. Next go to the osd host server and restart the noted osd number
i. "restart ceph-osd id=x"

e. Go back to the monitor and wait for the cluster to clear up. You should see the number reduce evenly. If it doesn't go back to completely clean, then you must repeat the previous step until you get the whole cluster. Remember to wait until the cluster cleans up.
i. If you see the same three osd's and restarting the first osd doesn't fix the issue, restart the second and then the third (assuming you have a replication factor of 3).

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(with the right hardware and config tuning. The steps they have on the ceph website are much better than when I original typed this up, so I have adjusted the section below(as of 11/15) to work with the updated version(as of the same date). Please note that I am using Ubuntu 14.04 and apache 2.4.7 which in Ubuntu per their notes does not support the socket connection. There are still some gaps in the live docs, but its mainly concerning apache2 service parts that people who don't know apache wouldn't know. Please note that as of this writing, if you are using apache 2.4.9, then you can use the socket config on their site. That configuration has some differences in ceph configuration and configuration on the rados gateway(which are noted in the ceph rados gateway doc).

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

```
[client.radosgw.<GatewayNodeHostnamehere>]
host = <GatewayNodeHostnamehere>
keyring = /etc/ceph/ceph.client.radosgw.keyring
rgw socket path = ""
rgw frontends = fastcgi socket_port=9000 socket_host=0.0.0.0
log file = /var/log/radosgw/client.radosgw. <GatewayNodeHostnamehere>.log
rgw print continue = false
rgw dns name = storage.awesomedude.com
```

a.
    i. Gateway Node Hostname 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. Keeping for reference, skip to ii
            1. No longer needed(keeping in directions for posterity): Apache now has fastcgi as a mod and radosgw is installable now from apt
                a. 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 update && sudo apt install apache2 libapache2-mod-fastcgi
        iii. Enable modules in apache(run the following commands):
            1. a2enmod rewrite
            2. a2enmod fastcgi
            3. a2enmod proxy_fcgi
    c. Install RADOS gateway package: “sudo apt 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”):

1.

```
<VirtualHost *:80>
ServerName localhost
DocumentRoot /var/www/html
        ErrorLog ${APACHE_LOG_DIR}/error.log
        CustomLog ${APACHE_LOG_DIR}/access.log
combined
# LogLevel debug
RewriteEngine On
RewriteRule .* -
[E=HTTP_AUTHORIZATION:%{HTTP:Authorization},L]
SetEnv proxy-nokeepalive 1
ProxyPass / fcgi://localhost:9000/
#ProxyPass /
unix:///var/run/ceph/ceph.radosgw.<nodenamehere
>.fastcgi.sock|fcgi://localhost:9000/
</VirtualHost>
```

iii. Enable the site: "sudo a2ensite rgw.conf"
iv. Disable the default site: "sudo a2dissite default"
    1. If you get an error, do "sudo a2dissite 000-default" instead
        a. They changed the name depending on version of apache you have installed.

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 ceph.client.radosgw.keyring”
    ii. Add read to it: “sudo chmod +r ceph.client.radosgw.keyring ”
c. Generate a key so the gateway can authenticate with the cluster
    i. “sudo ceph-authtool ceph.client.radosgw.keyring -n client.radosgw.<gateway> --gen-key”
    ii. “sudo ceph-authtool -n client.radosgw. <gateway> --cap osd 'allow rwx' --cap mon 'allow rwx' ceph.client.radosgw.keyring ”
    iii. “sudo ceph -k ceph.client.admin.keyring auth add client.radosgw.<gateway> -i ceph.client.radosgw.keyring”
        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 /etc/ceph/ceph.client.admin.keyring auth add client.radosgw.<radosnodename> -i /etc/ceph/ceph.client.radosgw.keyring”
e. Copy the keyring to your RADOS gateway node to the:
    i. “scp ceph.client.radosgw.keyring <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 restart”
7. Take a look at the RADOS log to verify all is well:
    a. “more /var/log/radosgw/client.radosgw. <GatewayNodeHostnamehere>.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. For new versions of BIND9 (9.4.6 and up), you need to modify the configuration to allow recursion. There are ways to do this securely, so if you need that, browse the web for the add-on settings to make this happen, otherwise, here is what needs to be added to the configuration file “named.conf.options” in the same folder (/etc/bind):

i. “recursion yes;”

ii. “allow-recursion { any; };”

iii. “allow-recursion-on { any; };”

i. Restart bind service (service bind9 restart)

j. 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!

d. NOTE: If you get a secret key with " \/ " in it, you need to remove the " \ " from the line. Its an escape character they put in, so it screws up the actual information you need.

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 using the new python scripts(s3cmd has an issue)
    a. Install Python
    b. Install python-pip (apt install python-pip)
    c. Install pip-boto (pip install –U bot)
    d. Create a script file with the contents below

```
import boto
import boto.s3.connection
access_key = ''
secret_key = ''
conn = boto.connect_s3(
      aws_access_key_id = access_key,
      aws_secret_access_key = secret_key,
      host = 'ukradosgw2',
      is_secure=False,
      calling_format = boto.s3.connection.OrdinaryCallingFormat(),
      )

#Creates Bucket
#bucket = conn.create_bucket('userimages')

#Deletes Bucket
#bucket = conn.delete_bucket('userimages')

#Lists Buckets
for bucket in conn.get_all_buckets():
      print "{name}\t{created}".format(
            name = bucket.name,
            created = bucket.creation_date,
```

)

e. Save this file and chmod it 777 to make executable.
f. Edit the access key and secret key so that it matches the user you created
g. Edit the bucket name in create bucket to whatever you want and then remove the # sign at the beginning of the line to make the command active. Make sure to put it back when you are done.
h. Run the script you created with "./scriptname" minus the quotes. It should create and the list the bucket. Change the bucket name to add another, make sure to comment out the create bucket line so you can just list buckets.

13. 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
            3. Don't bother testing as you need to change the host base and bucket info
        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 = <hostname>.awesomedude.com
                b. host_bucket = %(bucket)s.<radoshostname>.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.
        b. I should note that there is an issue with the current ceph or ceph gateway software, you must use uppercase for the first letter of bucket names. At this time, S3cmd throws an error message about it but the python script they show works without the case requirement.
    v. Next run the list command again: “s3cmd ls”
        1. Should display the bucket you just created!
    vi. Create a text file (touch text.txt) and upload it
        1. “s3cmd put test.txt s3://FOO”
        2. Delete the local file
    vii. Pull the file back from storage
        1. “s3cmd get s3://FOO/test.txt”
    viii. 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”.
14. 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 3/3/16