CHTC Student Handbook

Edition 0.1

Dakota Chambers

May 18, 2016

Contents

1 Student Work 101

1.1 Tasks and Workflow

As a student worker for the CHTC you will be responsible for a large variety of tasks to help keep the CHTC Cluster up and running, but there a few basic things you will find yourself doing most often. Our primary job is to fix individual machines in the cluster that go down. Most often machines can be brought down by kernel panics, networking issues, or hardware issues. To fix a machine, we take a simple approach to diagnose the issue and come up with a solution.

1. Diagnose the problem. You might need to plug into the machine with a console, or SSH into it if it is still online. If a machine kernel panics it will often print a memory stack trace to the console and freeze up. You will need to physically reboot the machine to proceed.
2. Find a solution. Often times rebooting the machine will clear up minor problems such as kernel panics or networking issues.
    - Kernel Panic (crash) - reboot, check server for errors.
    - Networking issue - reboot machine or restart networking, then go to advanced networking troubleshooting if still needed.
    - Hardware issue - diagnose the hardware issue and fix it. (Replace a bad disk, make note of a bad RAID card, etc)
3. Fix the problem and confirm that condor is running jobs again.
    - his might mean rebooting the machine, rebuilding the machine, replacing a bad disk and rebuilding, or something else entirely. Fix it if possible. If not - make note of the issue in the ticket and await further instructions.

Some other tasks you might be assigned include, but are not limited to: archiving condor releases to DVD, building new execute machines, working with Dell/Cisco Tech Support to get replacement parts, moving servers, a nd other random tasks assigned by full time CHTC staff.

1.2 RT Ticket System

We use the Request Tracker (RT) Ticket System for managing work at the CHTC. crt.cs.wisc.edu is the web address to access our Request Tracker.This will be your home for managing work tickets. The basic flow is: You are assigned a ticket, click on the ticket name to open it. By default, your 10 highest priority, or newest, tickets will be listed on your home page. You can view all of your tickets by clicking the “10 Highest Priority Tickets I Own” link (counter-intuitive, I know).

Once you have selected a ticket, you can read all correspondence related to that particular issue. Under the “actions” button (top right) the most common things you will do are “reply” and “resolve.” Whenever you make changes to a machine, make sure to log what you did in the ticket. When the machine is fixed or the task is completed, you can resolve the ticket.

1.3 Monitoring Nodes

You can monitor the cluster on monitor0.chtc.wisc.edu. This has links to Icinga, our host monitoring application, Grafana, and Ganglia. All of these are extremely powerful tools that can be used to monitor the status of machines in the cluster and diagnose problems with individual nodes.

1.4 How to Use This Guide

- Any line or section beginning with “$” is a command line operation. Enter it directly as it appears, without the preceding “$”.
- Any bracketed item (eg. [System Name]) needs to be replaced with the corresponding entity. (eg. change [System Name] $\rightarrow$ e1000.chtc.wisc.edu).
- Following the step-by-step walkthroughs to complete most basic tasks and use this as a reference manual.
- The appendix includes a complete list of node locations, useful for location a machine for physical changes (hard reboot, HDD swap, etc).

2 Rebuilding a Node

Rebuilding nodes is one of the primary responsibilities of student CHTC employees. Here's a step-by-step walkthrough for you.

1. Enable netboot in cobbler.
   (a) Via Command Line:
      i. SSH into wid-service-1.chtc.wisc.edu
      ii. `$ sudo cobbler system edit --name=[server name] --netboot-enabled=True`
      iii. Check that netboot is enabled → `$ sudo cobbler system report --name=[server name]`
      iv. Sync Cobbler → `$ sudo cobbler sync`
   (b) Via Web Interface
      i. Open Cobbler in your web browser
      ii. Click on "Systems" under the Configuration tab on the left side.
      iii. Find the node you want to rebuild and click on it's name in the list to open the node configuration tool.
      iv. Alternatively, navigate directly to wid-service-1.chtc.wisc.edu/cobbler_web/system/edit/[node name]
      v. Click on the "general" drop down button to reveal more options
      vi. Check the "enable netboot" option
      vii. Hit "save" on the bottom of the page
      viii. When you are redirected to the cobbler system list page, hit "Sync" Under the Actions tab on the left side.
      ix. Once you get a popup notification on the top-right of the screen, the sync has complete. This may take a few seconds.
   (c) Make sure you have the correct profile enabled in Cobbler as well. If you are doing a standard rebuild you probably won't have to change it, but make sure if it's a multi-disk execute node that it is set to the correct SL66 Exec profile.
2. Reboot the machine and netboot it.
   (a) Most machines are set to netboot by default, meaning if you reboot them they will search for a netboot entry and if they find it, they will netboot automatically. If a machine is not netbooting automatically, you may need to press a button on the keyboard (Often F12) when it POSTS in order to force it to netboot. If you have tried these and it still won't boot from the network, go into the BIOS and change the boot order to set netboot as boot priority #1.
   (b) If done correctly, it should launch the Scientific Linux installer. Once you see this is happening, move on to the next step.

3. Run Puppet.
   (a) We are going to need to run puppet once the machine is rebuilt in order to configure it.
   (b) While the machine is rebuilding, SSH into wid-service-1.chtc.wisc.edu
   (c) Run this command: `$ sudo puppetca -c [node name]` on wid-service-1.
   (d) If you do this before the machine finishes rebuilding, it may run puppet automatically when it rebuilds. To see if it is doing this: When the machine is booting Scientific Linux, hit an arrow key $\leftarrow$ or $\rightarrow$ on the console keyboard to view the boot log. If the log is paused on "`Starting: anamon... [OK]` " for a while, that mean's it's running puppet. Good job! Once it finishes booting now, you should be able to log in with your username.
   (e) If the machine does not automatically run puppet, connect a console to the machine and log in as a root user (ask Admin for root login)
   (f) After clearing the puppet files from wid-service-1 in the previous steps, run `$ sudo rm -rvf /var/lib/pup/ssl` on the target node. DO NOT RUN THIS ON wid-service-1.
   (g) Then run `$ sudo puppetd -tv --configtimeout=1000`
   (h) Note: Puppet will not run if networking is broken (try to restart networking or reboot the machine if this is the case) or the system clock is broken. To set the clock run: `$ rdate -s ntp.doit.wisc.edu`
4. Log in and confirm condor is running.
   (a) Once puppet finishes running, you should be able to log in with your own username
   (b) Confirm condor is running: `$ condor_status $HOSTNAME`
   (c) You should also see Condor running with `$ ps aux`

3 Building a New Node

TODO

4 Decommissioning a Node

TODO

5 Creating Tickets

TODO

6 Checking for Errors

6.1 Monitoring with Icinga

6.1.1 Icinga Basics

6.1.2 Web Interface

6.1.3 Icinga CLI

6.2 Error Logs

6.3 Hardware Errors

A Node Locations