Gmail Mahammad Azeem <amazeeeeem6@gmail.com>

important unix basics

3 messages

Azeem Mahammad <amazeeeeem6@gmail.com> Fri, Jun 2, 2017 at 3:12 PM
To: "amazeeeeem6@gmail.com" <Mahammadazeem14@gmail.com>, MA00465237@techmahindra.com

Command To List Number Of Open File Descriptors

Use the following command command to display maximum number of open file descriptors:
cat /proc/sys/fs/file-max

To see the hard and soft values, issue the command as follows:
ulimit -Hn
ulimit -Sn
To see the hard and soft values for httpd or oracle user, issue the command as follows:
su - username
In this example, su to oracle user, enter:
# su - oracle
$ ulimit -Hn
$ ulimit -Sn

System-wide File Descriptors (FD) Limits

The number of concurrently open file descriptors throughout the system can be changed via /etc/sysctl.conf file under Linux operating systems.

The Number Of Maximum Files Was Reached, How Do I Fix This Problem?

Many application such as Oracle database or Apache web server needs this range quite higher. So you can increase the maximum number of open files by setting a new value in kernel variable /proc/sys/fs/file-max as follows (login as the root):
sysctl -w fs.file-max=100000
Above command forces the limit to 100000 files. You need to edit /etc/sysctl.conf file and put following line so that after reboot the setting will remain as it is:
vi /etc/sysctl.conf
Append a config directive as follows:
fs.file-max = 100000
Save and close the file. Users need to log out and log back in again to changes take effect or just type the following command:
sysctl -p
Verify your settings with command:
cat /proc/sys/fs/file-max
OR
sysctl fs.file-max

User Level FD Limits

The above procedure sets system-wide file descriptors (FD) limits. However, you can limit httpd (or any other users) user to specific limits by editing /etc/security/limits.conf file, enter:
vi /etc/security/limits.conf
Set httpd user soft and hard limits as follows:
httpd soft nofile 4096
httpd hard nofile 10240
Save and close the file. To see limits, enter:
# su - httpd
$ ulimit -Hn
$ ulimit -Sn

A note about RHEL/CentOS/Fedora/Scientific Linux users

Edit /etc/pam.d/login file and add/modify the following line (make sure you get pam_limts.so):

session required pam_limits.so

Save and close the file.

==========

Set the Time Zone Manually on a Linux System

Find the appropriate zone file in /usr/share/zoneinfo/ and link that file to /etc/localtime. See the examples below for possibilities:

Universal Coordinated Time:

ln -sf /usr/share/zoneinfo/UTC /etc/localtime

==========

Configure the /etc/hosts File

The /etc/hosts file provides a list of IP addresses with corresponding hostnames. This allows you to specify hostnames for an IP address in one place on the local machine, and then have multiple applications connect to external resources via their hostnames. The system of host files precedes DNS, and hosts files are *always* checked before DNS is queried. As a result, /etc/hosts can be useful for maintaining small “internal” networks, for development purposes, and for managing clusters.

Some applications require that the machine properly identify itself in the /etc/hosts file. As a result, we recommend configuring the /etc/hosts file shortly after deployment. Here is an example file:

/etc/hosts

1127.0.0.1 localhost.localdomain localhost

2103.0.113.12 username.example.com username

You can specify a number of hostnames on each line separated by spaces. Every line must begin with one and only one IP address. In the above example, replace 103.0.113.12 with your machine’s IP address. Consider a few additional /etc/hosts entries:

/etc/hosts

1198.51.100.30 example.com

2192.168.1.1 stick.example.com

In this example, all requests for the example.com hostname or domain will resolve to the IP address 198.51.100.30, which bypasses the DNS records for example.com and returns an alternate website.

The second entry tells the system to look to 192.168.1.1 for the domain stick.example.com. These kinds of host entries are useful for using “private” or “back channel” networks to access other servers in a cluster without needing to send

traffic on the public network.

==========

The mtr Command

The mtr command, like the traceroute tool, provides information about the route that internet traffic takes between the local system and a remote host. However, mtr provides additional information about the round trip time for the packet. In a way, you can think of mtr as a combination of traceroute and ping.

Like the ping command, mtr tracks the speed of the connection in real time until you exit the program with **CONTROL+C**. To have mtr stop automatically and generate a report after ten packets, use the --report flag:

```
mtr --report
```

==========

Check Current Memory Usage

To see how much memory your system is currently using:

```
free -m
```

On a Linode 2GB under moderate use, the output should resemble the following:

```
1             total       used       free     shared    buffers     cached
2Mem:          1999        954       1044        105         34        703
3-/+ buffers/cache:        216       1782
4Swap:          255          0        255
```

This output takes a bit of careful reading to interpret. Out of a total 1999 megabytes of memory (RAM), the system is using 954 megabytes and has 1044 megabytes free. *However*, the system also has 703 megabytes of “stale” data buffered and stored in cache. The operating system will “drop” the caches if it needs the space, but retains the cache if there is no other need for the space. It is normal for a Linux system to leave old data in RAM until the space is needed, so don’t be alarmed if only a small amount of memory is “free.”

In the above example, there are 1782 megabytes of memory that are actually *free*. This means 1782 megabytes are available to your system when you start an additional process or a running application needs more memory.

==========

Monitor I/O Usage with vmstat

The vmstat tool provides information about memory, swap utilization, I/O wait, and system activity. It is particularly useful for diagnosing I/O-related issues.

If you think you’re having an I/O issue then run the following command:

```
1vmstat 1 20
```

This runs a vmstat every second, twenty times, giving a sample of the current state of the system. The output generated resembles the following:

```
1  procs -----------memory---------- ---swap-- -----io---- -system-- ----cpu----
2   r  b   swpd   free   buff  cache   si   so    bi    bo   in   cs us sy id wa
3   0  0      4  32652  47888 110824    0    0 0     2   15   15  0  0 100  0
4   0  0      4  32644  47888 110896    0    0 0     4  106  123  0  0 100  0
5   0  0      4  32644  47888 110912    0    0 0     0   70  112  0  0 100  0
6   0  0      4  32644  47888 110912    0    0 0     0   92  121  0  0 100  0
7   0  0      4  32644  47888 110912    0    0 0    36   97  136  0  0 100  0
8   0  0      4  32644  47888 110912    0    0 0     0   96  119  0  0 100  0
9   0  0      4  32892  47888 110912    0    0 0     4   96  125  0  0 100  0
10  0  0      4  32892  47888 110912    0    0 0     0   70  105  0  0 100  0
11  0  0      4  32892  47888 110912    0    0 0     0   97  119  0  0 100  0
12  0  0      4  32892  47888 110912    0    0 0    32   95  135  0  0 100  0
13  0  0      4  33016  47888 110912    0    0 0     0   75  107  0  0 100  0
14  0  0      4  33512  47888 110912    0    0 0    24  113  134  0  0 100  0
15  0  0      4  33512  47888 110912    0    0 0     0  175  244  0  0 100  0
16  0  0      4  33512  47888 110912    0    0 0     0   92  148  0  0 100  0
17  0  0      4  33512  47888 110912    0    0 0     0  114  162  0  0 100  0
18  0  0      4  33512  47888 110912    0    0 0    36  100  157  0  0 100  0
19  0  0      4  33388  47888 110912    0    0 0     0  116  166  0  0 100  0
20  0  0      4  33388  47888 110912    0    0 0     0   97  157  0  0 100  0
21  0  0      4  33388  47888 110912    0    0 0     0   89  144  0  0 100  0
22  0  0      4  33380  47888 110912    0    0 0     0  181  185  0  0 99  0
```

The memory and swap columns provide the same kind of information provided by the “free -m” command, albeit in a slightly harder to understand format. The most relevant information produced by this command is the wa column, which is the final column in most implementations. This field displays the amount of time the CPU spends waiting for I/O operations to complete.

If this number is consistently and considerably higher than 0, you might consider taking measures to address your IO usage. However, if the vmstat output resembles the above, you can be sure in the knowledge that you’re not experiencing an IO-related issues.

If you are experiencing an intermittent issue, you will need to run vmstat *when* you experience the issue in order to properly diagnose or rule out an I/O issue. vmstat output can sometimes help support diagnose problems.

==========

Symbolic Links

Symbolic linking, colloquially “symlinking”, allows you to create an object in your filesystem that points to another object on your filesystem. This is useful when you need to provide users and applications access to specific files and directories without reorganizing your folders. This way you can provide restricted users access to your web-accessible directories without moving your DocumentRoot into their home directories.

To create a symbolic link, issue a command in the following format:

ln -s /home/username/config-git/etc-hosts /etc/hosts

This creates a link of the file etc-hosts at the location of the system's /etc/hosts file. More generically:

ln -s [/path/to/target/file] [/path/to/location/of/sym/link]

Note the following features of the link command:

- The final term, the location of the link, is optional. If you omit the link destination, a link will be created in the current directory with the same name as the file you're linking to.
- When specifying the location of the link, ensure that path does not have a final trailing slash. You can create a sym link that *targets* a directory, but sym links cannot terminate with slashes.
- You may remove a symbolic link without affecting the target file.
- You can use relative or absolute paths when creating a link.

==========

Finding Packages Installed on Your System

For Arch Linux systems:

pacman -Q

==========

SED == Stream EDitor

==========

Web Servers and HTTP Issues

Linodes do not come with a web server installed by default. You must install and configure your web server. This allows you to configure your web server in a way that makes sense for your application or website. Linode Guides & Tutorials contains a number of documents regarding the installation and maintenance of various web servers.

This section covers a number of basic web serving tasks and functions, as well as some guidance for users new to the world of web servers.

Serve Websites

Web servers work by listening on a TCP port, typically port 80 for HTTP and port 443 for HTTPS. When a visitor makes a request for content, the servers respond by delivering the resource requested. Typically, resources are specified with a URL that contains the protocol, http or https; a colon and two slashes, ://; hostname or domain, www.example.com or username.example.com; and the path to a file, /images/avatar.jpg, or index.html. A full URL would resemble http://www.example.com/images/avatar.jpg.

In order to provide these resources to users, your Linode needs to be running a web server. There are many different HTTP servers and countless configurations to provide support for various web development frameworks. The three most popular general use web servers are the Apache HTTP server, Lighttpd ("Lighty"), and nginx ("Engine X"). Each server has its strengths and weaknesses, and your choice depends largely on your experience and your needs.

Once you've chosen a web server, you need to decide what (if any) scripting support you need to install. Scripting support allows you to run dynamic content with your web server and program server side scripts in languages such as Python, PHP, Ruby, and Perl.

If you need a full web application stack, we encourage you to consider one of our more full-featured LAMP stack guides. If you need support for a specific web development framework, consult our tutorials for installing and using specific web application frameworks.

How to Choose a Web Server

In most situations, end users are unaware of which web server you use. As a result, choosing a web server is often a personal decision based on the comfort of the administrator and the requirements of the deployment in question. This can be a challenge for the new systems administrator. This section offers some guidance by providing some background and information on the most popular web servers.

The Apache HTTP Server is considered by some to be the *de facto* standard web server. It is the most widely deployed open-source web server, its configuration interface has been stable for many years, and its modular architecture allows it to function in many different types of deployments. Apache forms the foundation of the LAMP stack, and supports the integration of dynamic server-side applications into the web server.

By contrast, web servers like Lighttpd and nginx are optimized for efficiently serving static content. If you have a deployment where server resources are limited and are facing a great deal of demand, consider one of these servers. They are functional and stable with minimal system resources. Lighttpd and nginx can be more difficult to configure when integrating dynamic content interpreters.

Your choice of web servers is based on your needs. Specific choices depend on factors like the type of content you want to serve, the demand for that content, and your comfort with that software as an administrator.

Apache Logs

When there is something wrong with Apache, it can be difficult to determine what the cause of the error is from the behavior of the web server. There are a number of common issues with which you might begin your troubleshooting efforts. When more complex issues arise, you may need to review the Apache error logs.

By default, error logs are located in the /var/log/apache2/error.log file (on Debian-based distributions). You can track or “tail” this log with the following command:

tail -F /var/log/apache2/error.log

In the default virtual host configurations suggested in our Apache installation and LAMP guides, we suggest adding a custom log setting:

Apache Virtual Host Configuration

ErrorLog /var/www//html/example.com/logs/error.log CustomLog /var/www/html/example.com/logs/access.log combined

Where example.com represents the name of your virtual host and the location of its resources. These directives make Apache create two log files that contain logging information specific to that virtual host. This allows you to easily troubleshoot errors on specific virtual hosts. To track or tail the error log:

tail -F /var/www/html/example.com/logs/error.log

This will allow you to see new error messages as they appear. Problems can be diagnosed by using specific parts of an error message from an Apache log as a term in web search. Common errors to look for include:

- Missing files, or errors in file names
- Permissions errors
- Configuration errors
- Dynamic code execution or interpretation errors

Mahammad Azeem <amazeeeeem6@gmail.com> Wed, Jan 19, 2022 at 6:36 PM
To: md asif Mm <mdasif.mm@gmail.com>

[Quoted text hidden]
--
Regards,
Azeem
+6590507284

Mahammad Azeem <amazeeeeem6@gmail.com> Wed, Oct 19, 2022 at 12:59 AM
To: shifakorti@gmail.com

Regards,
Azeem
+6590507284

---------- Forwarded message ---------
From: **Azeem Mahammad** <amazeeeeem6@gmail.com>
Date: Fri, Jun 2, 2017, 3:12 PM
Subject: important unix basics
To: amazeeeeem6@gmail.com <Mahammadazeem14@gmail.com>, <MA00465237@techmahindra.com>

[Quoted text hidden]