Varnish Cache Server Setup Using Ansible

By: Er. Vikas Nehra (M. Tech, B. Tech), Experience: 15 + Years

Session - 46 Agenda:

Varnish Cache Server Setup Using Ansible.

Varnish Cache is a free open source, modern and high-performance web application accelerator. It is a fast reverse HTTP proxy that caches content to speed up your web server performance, by storing web content in server memory – in a cache. It is configured to run in front of an origin server such as Apache (HTTPD) webserver.

When a client requests for content, Varnish accepts the HTTP request, sends the request to the origin server, caches the returned objects, and replies to the client request. The next time the client requests for the same content, Varnish will serve it from the cache. This way, it reduces the response time and network bandwidth consumption on future equivalent requests. Varnish also works as an HTTP request router, web application firewall, load balancer, and more. It is configured using the flexible Varnish Configuration Language (VCL) which is extensible using Varnish Modules (also known as VMODs), supports for Edge Side Includes (ESL), Gzip compression and decompression, and much more.

Let's create an Ansible playbook to setup the varnish cache server at the managed node(s).

$ vim varnish-server.yml

```
---
- name: Varnish Cache For Apache HTTP Server Configuration Playbook
  hosts: node1
  become: true
  tasks:
    - name: Setting up the static hostname in the server machine.
      hostname:
        name: node1.nehraclasses.com
        use: systemd

    - name: Making entries in the /etc/hosts file for the server hostname & IP Address
      lineinfile:
        dest: /etc/hosts
        line: 192.168.229.129 www.nehraclasses.com node1
        insertafter: EOF

    - name: Installing Apache & SELinux Policy packages in the machine.
      dnf:
        name:
          - httpd
          - policycoreutils-python-utils
          - varnish
        state: latest

    - name: Creating /nehraclassesweb/ directory for the document root.
      file:
        path: /nehraclassesweb/
        mode: '0755'
        state: directory
        owner: apache
        group: apache
```

Varnish Cache Server Setup Using Ansible

By: Er. Vikas Nehra (M. Tech, B. Tech), Experience: 15 + Years

```
- name: Copying the image file to the /nehraclassesweb/ directory.
  ansible.builtin.copy:
    src: /home/vikasnehra/NehraClassesLogo.png
    dest: /nehraclassesweb/NehraClassesLogo.png
    mode: '0644'

- name: Creating the website index file in the /nehraclassesweb/ directory.
  copy:
    dest: "/nehraclassesweb/index.html"
    content: |
      <img src="/NehraClassesLogo.png" width="500" height="500" />
      <h1>Nehra Classes Are Awesome.</h1>
      <i>This page is hosted on node1 using name based apache virtual hosting.</i>

- name: Creating HTTPD configuration file in the /etc/httpd/conf.d/ directory.
  copy:
   dest: "/etc/httpd/conf.d/httpd.conf"
   content: |
      <VirtualHost 192.168.229.229:8080>
      ServerAdmin root@node1.nehraclasses.com
      ServerAlias nehraclasses.com node1
      DocumentRoot /nehraclassesweb/
      ServerName www.nehraclasses.com
      ErrorLog logs/www.nehraclasses.com-error_log
      CustomLog logs/www.nehraclasses.com-access_log common
      <Directory /nehraclassesweb/>
      Options Indexes FollowSymLinks
      AllowOverride None
      Require all granted
      </Directory>
      </VirtualHost>

- name: Copying /etc/httpd/conf/httpd.conf file using ansible jinja template.
  template:
    src: httpd3.conf.j2
    dest: /etc/httpd/conf/httpd.conf
    force: true

  - name: Allowing Apache to modify the files in the /nehraclassesweb/ directory. (SELinux Context)
    community.general.sefcontext:
      target: '/nehraclassesweb(/.*)?'
      setype: httpd_sys_content_t
      state: present

- name: Applying new SELinux file context to filesystem /nehraclassesweb/
  command: restorecon -Rv /nehraclassesweb/

- name: Allowing HTTP traffic in the firewall.
```

Varnish Cache Server Setup Using Ansible

By: Er. Vikas Nehra (M. Tech, B. Tech), Experience: 15 + Years

```
    firewalld:
      service: http
      zone: public
      permanent: true
      immediate: true
      state: enabled

  - name: Starting & enabling the httpd service.
    service:
      name: httpd
      state: started
      enabled: yes

  - name: Making changes in the varnish cache /etc/varnish/default.vcl file.
    replace:
      dest: /etc/varnish/default.vcl
      regexp: '^.port ='
      replace: '.port = "8080";'

  - name: Creating systemd unit file for varnish service.
    template:
      src: varnish.service.j2
      dest: /etc/systemd/system/varnish.service

  - name: Reloading the Systemd to re-read the configuration.
    systemd:
      daemon-reload: yes

  - name: Starting & enabling the varnish service.
    service:
      name: varnish
      state: started
      enabled: yes
...
```

We would require the ansible.posix collection which we can install from Ansible Galaxy.
$ ansible-galaxy collection install ansible.posix

We would also require the community.general collection which we can install from Ansible Galaxy.
$ ansible-galaxy collection install community.general

Now, we can execute the ansible playbook to setup the Varnish Cache server at the managed node(s).
$ ansible-playbook varnish-server.yml

We can run the following curl command to verify it.
$ curl -I http:///192.168.229.129
OR
$ curl -I http:///www.nehraclasses.com

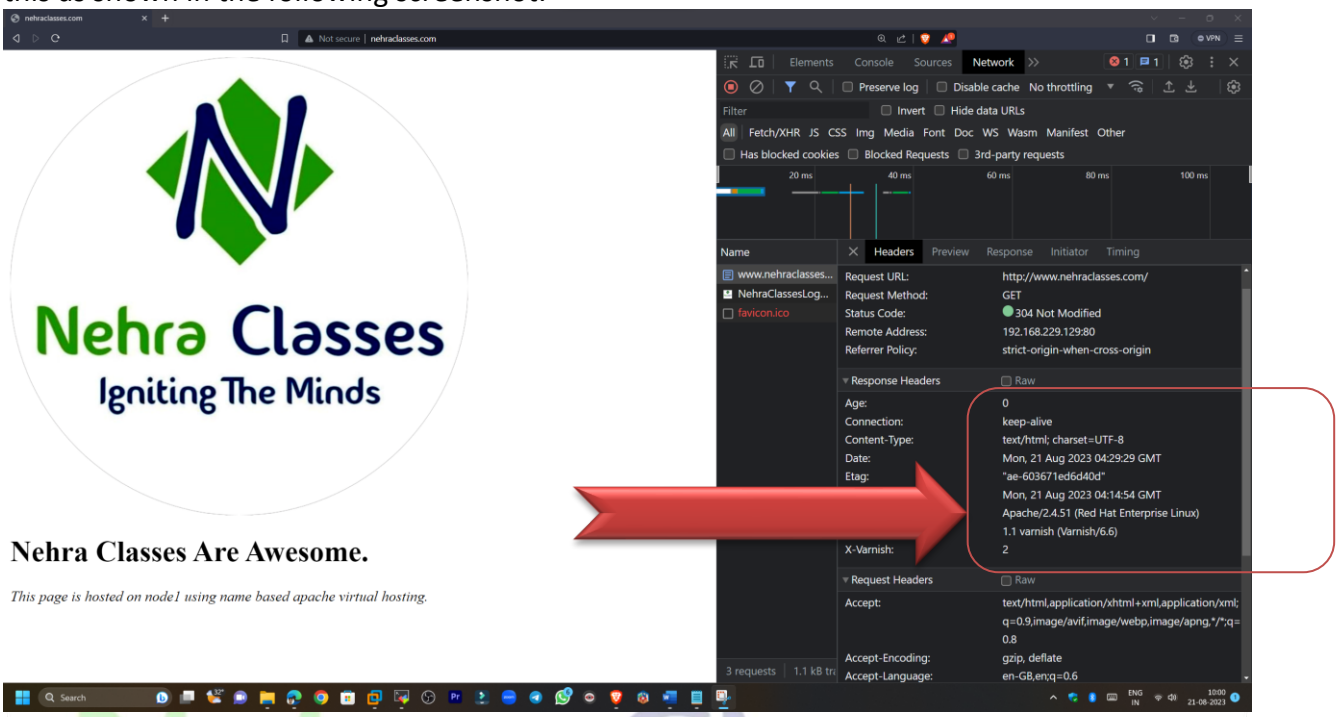
Varnish Cache Server Setup Using Ansible

By: Er. Vikas Nehra (M. Tech, B. Tech), Experience: 15 + Years

Alternatively, we can verify from the web browser as well. To test the Varnish Cache-HTTPD setup, open a web browser, and navigate using the server IP or FQDN as shown in the following screenshot.

http://www.nehraclasses.com

Then check if web pages are being served via Varnish Cache as follows. Check the HTTP headers by right-clicking on the displayed web page, select Inspect to open the developer tools, then click the Network tab, and reload the page. Then select a request to view the HTTP headers to confirm this as shown in the following screenshot.

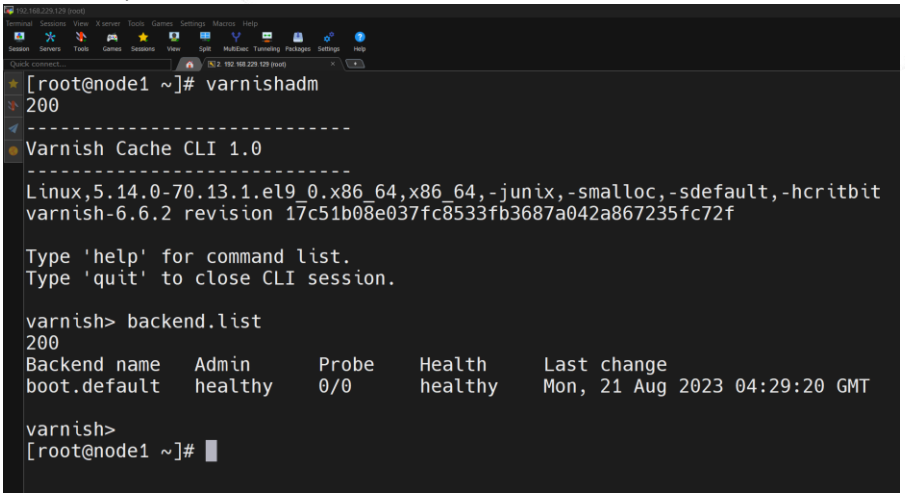


Useful Varnish Cache Utility Programs:

Let's end this guide by looking at some of the useful programs that the Varnish Cache distribution comes with. They include utilities for varnish cache administration, displaying detailed log records, and view varnish performance statistics as described below.

Varnishadm:

The first is varnishadm which is used to administer a running Varnish instance. It establishes a command-line interface connection to varnishd. It can affect a running instance of Varnish by starting and stopping varnishd, changing configuration parameters, reloading the VCL, listing backends, and more.

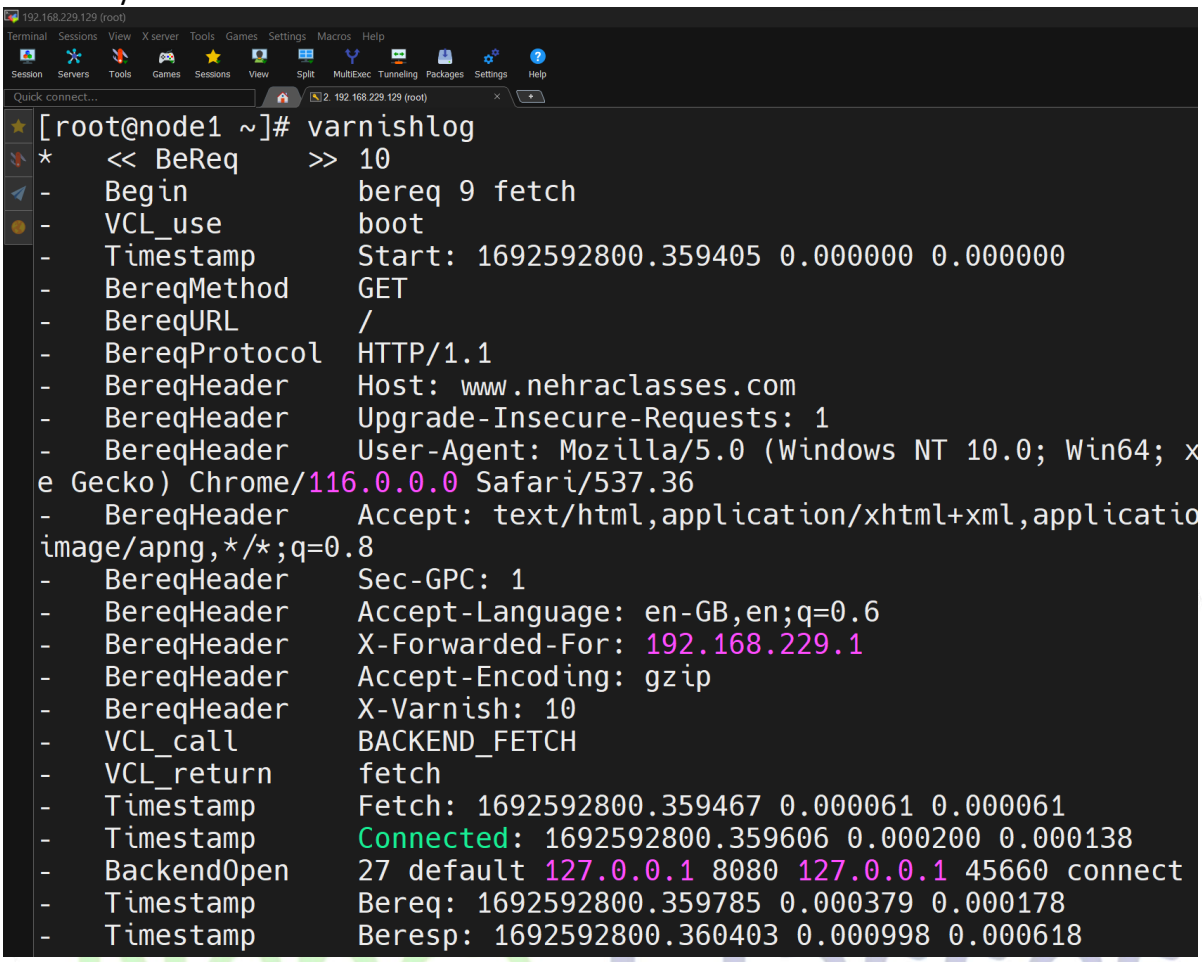


Varnish Cache Server Setup Using Ansible

By: Er. Vikas Nehra (M. Tech, B. Tech), Experience: 15 + Years

Varnishlog:

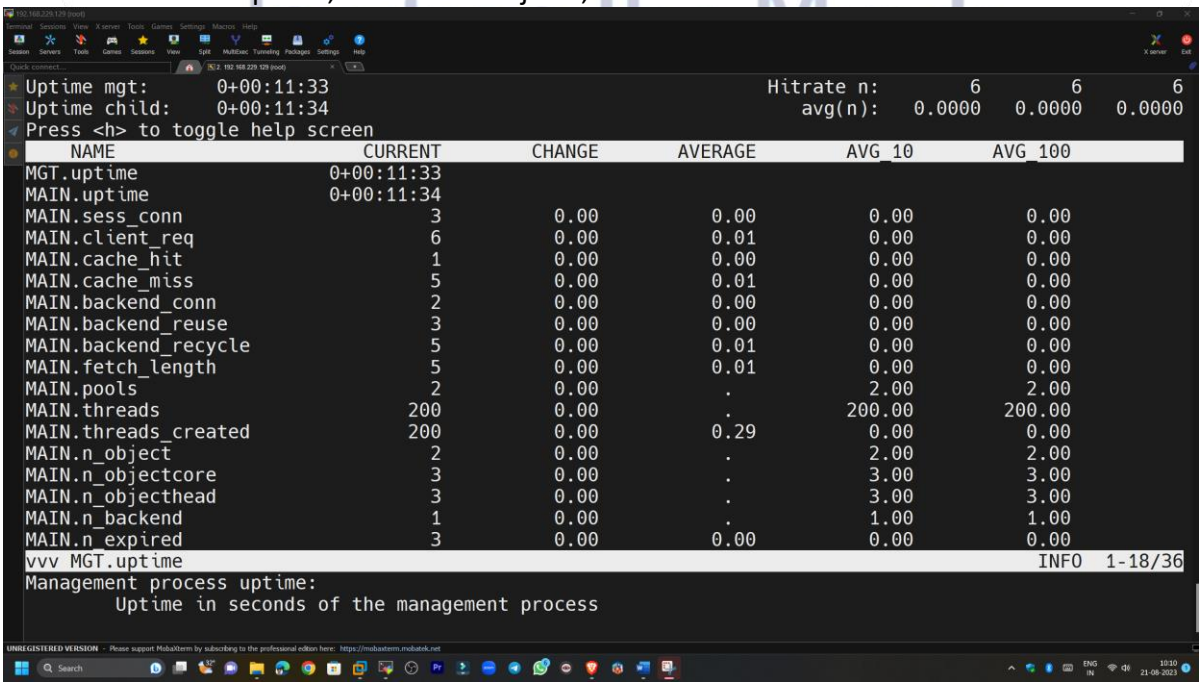
The next program is varnishlog which is used to access request-specific data (i.e information about specific clients and requests). It provides large amounts of information; thus, it is usually necessary to filter it.



```
[root@node1 ~]# varnishlog
*   << BeReq    >> 10
-   Begin          bereq 9 fetch
-   VCL_use        boot
-   Timestamp      Start: 1692592800.359405 0.000000 0.000000
-   BereqMethod    GET
-   BereqURL       /
-   BereqProtocol  HTTP/1.1
-   BereqHeader    Host: www.nehraclasses.com
-   BereqHeader    Upgrade-Insecure-Requests: 1
-   BereqHeader    User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x
e Gecko) Chrome/116.0.0.0 Safari/537.36
-   BereqHeader    Accept: text/html,application/xhtml+xml,applicatio
image/apng,*/*;q=0.8
-   BereqHeader    Sec-GPC: 1
-   BereqHeader    Accept-Language: en-GB,en;q=0.6
-   BereqHeader    X-Forwarded-For: 192.168.229.1
-   BereqHeader    Accept-Encoding: gzip
-   BereqHeader    X-Varnish: 10
-   VCL_call       BACKEND_FETCH
-   VCL_return     fetch
-   Timestamp      Fetch: 1692592800.359467 0.000061 0.000061
-   Timestamp      Connected: 1692592800.359606 0.000200 0.000138
-   BackendOpen    27 default 127.0.0.1 8080 127.0.0.1 45660 connect
-   Timestamp      Bereq: 1692592800.359785 0.000379 0.000178
-   Timestamp      Beresp: 1692592800.360403 0.000998 0.000618
```

Varnishstat:

We also have varnishstat (varnish statistics) which is used to access overall statistics such as the number of total requests, number of objects, and more.



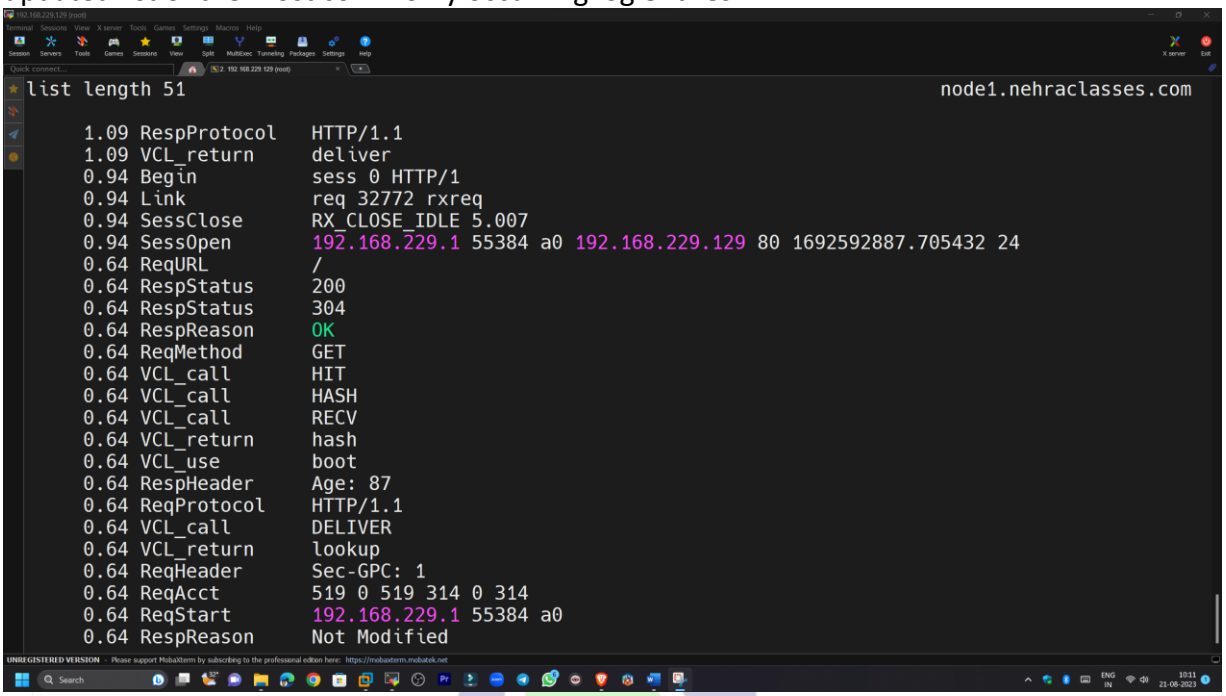
```
Uptime mgt:       0+00:11:33                          Hitrate n:       6       6       6
Uptime child:     0+00:11:34                            avg(n):  0.0000  0.0000  0.0000
Press <h> to toggle help screen
    NAME                      CURRENT        CHANGE       AVERAGE      AVG_10      AVG_100
MGT.uptime                 0+00:11:33
MAIN.uptime                0+00:11:34
MAIN.sess_conn                      3          0.00          0.00        0.00         0.00
MAIN.client_req                     6          0.00          0.01        0.00         0.00
MAIN.cache_hit                      1          0.00          0.00        0.00         0.00
MAIN.cache_miss                     5          0.00          0.01        0.00         0.00
MAIN.backend_conn                   2          0.00          0.00        0.00         0.00
MAIN.backend_reuse                  3          0.00          0.00        0.00         0.00
MAIN.backend_recycle                5          0.00          0.01        0.00         0.00
MAIN.fetch_length                   5          0.00          0.01        0.00         0.00
MAIN.pools                          2          0.00             .        2.00         2.00
MAIN.threads                      200          0.00             .      200.00       200.00
MAIN.threads_created              200          0.00          0.29        0.00         0.00
MAIN.n_object                       2          0.00             .        2.00         2.00
MAIN.n_objectcore                   3          0.00             .        3.00         3.00
MAIN.n_objecthead                   3          0.00             .        3.00         3.00
MAIN.n_backend                      1          0.00             .        1.00         1.00
MAIN.n_expired                      3          0.00          0.00        0.00         0.00
vvv MGT.uptime                                                                    INFO  1-18/36
Management process uptime:
        Uptime in seconds of the management process
```

Varnish Cache Server Setup Using Ansible

By: Er. Vikas Nehra (M. Tech, B. Tech), Experience: 15 + Years

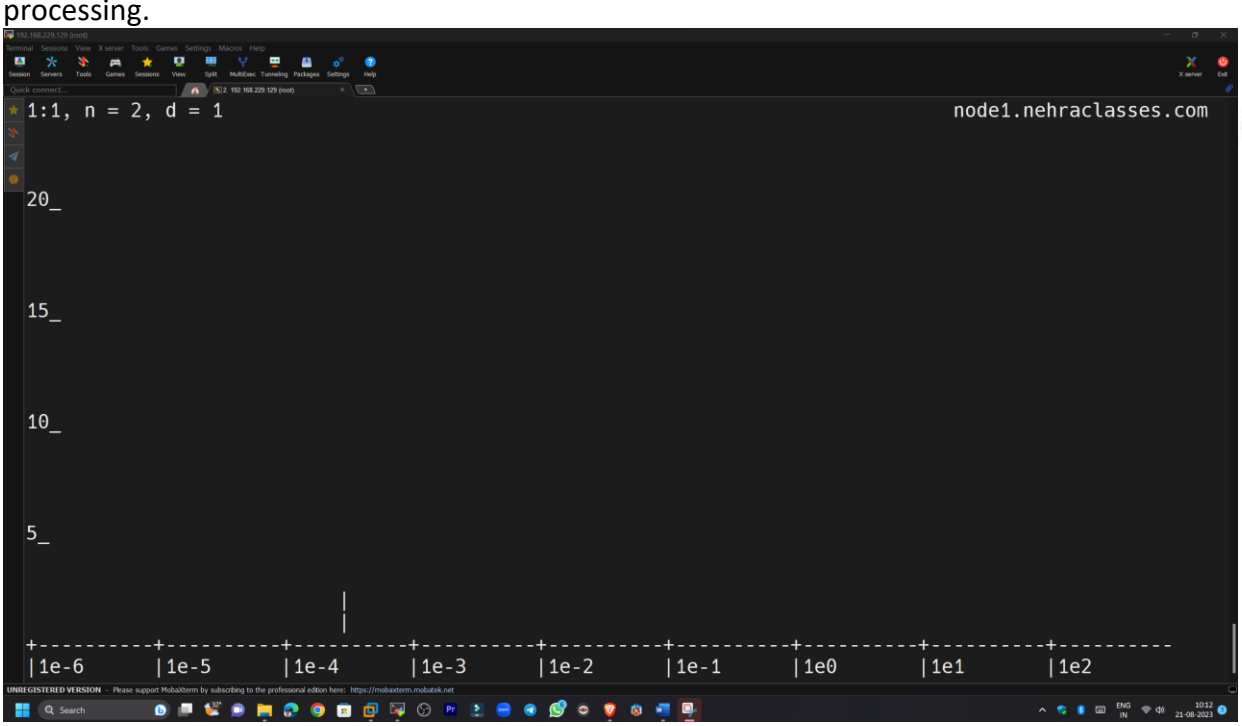
Varnishtop:

Then we have varnishtop which a utility that reads the Varnish log and presents a continuously updated list of the most commonly occurring log entries.



Varnishhist:

Another useful utility is varnishhist (varnish history) utility reads Varnish logs and presents a continuously updated histogram showing the distribution of the last N requests by their processing.



Varnish Cache server is working as expected.

Thank You