**curl -s http://public-dns.info/nameserver/br.csv | cut -d, -f1 | shuf | tail -n 50 | xargs -i timeout 1 ping -c1 -w 1 {} | grep "time=" | awk '{print substr($7, 6, length($7)) " " substr($4, 1, length($4) -1)}' | sort -n | awk '{print $2 " " $1 "ms"}' | head -n 10**

**curl -s http://public-dns.info/nameserver/br.csv:** Client URL transfers data to or from server. It basically transfers data to the terminal unless it is redirected.

-s is used for doing it silently without showing any progress meter or tracker. It makes this operation mute.

-o is used to redirect it to a file.

This command displays content of br.csv onto the terminal.

**|-** denotes a pipe. It feeds the output of the command to it's left, to the command to it's right.

**cut -d, -f1:** It removes some part of lines from a specified file. It is mostly used to display selected fields from each line of file.

This command has two options:-

-d: Specifies a delimiter, and ',' is the delimiter in this command.

-f: Specifies a field or set of fields, and 1 is the field number we have provided.

Here we get only the IP column data.

**shuf**: It shuffles the input lines and prints it to the output terminal.

-o: Writes output to a file instead of terminal.

-r: Lines in output can be repeated.

IP values are shuffled to a random order.

**tail -n 50:** tail command prints last 10 lines of input file.

-n: This prints the last n lines of input file.

Here we get the last 50 IP values.

**xargs -i timeout 1 ping -c1 -w 1 {}:** xargs executes the commands from standard input.

timeout enables to execute a command within a specified time limit. Here 1s is the timelimit which is imposed on this command.

ping -c1 -w1 will not stop after sending a packet and doesn't timeout until count packets are received.

-w is the deadline option.

-c is the count option.

PING command is used to check the network connectivity between host and server. This command takes the IP address or the URL as an input and sends a data packet to the specified address with the message “PING” and gets a response from the server. This time is recorded which is called latency.

ping -c -W will set time to wait for response.

ping -w 5 will stop pinging after 5 seconds.

ping -c 5 will stop pinging after sending 5 packets.

{} will take all the IP's present in the input sequentially.

**grep "time=":** This command will return all the lines in the input which has "time=" string.

Here the output only has the lines which have "time=" string.

**awk '{print substr($7, 6, length($7)) " " substr($4, 1, length($4) -1)}':**

In awk, every string can be represented with a $1,$2,$3,$4..., where each string is seperated by a blank space.

substr() can be used using awk command. It basically extracts substring from a string.

substr(a,b,c): a is the string, b is the starting position, and c is the length of portion to extract.

Here, we extract, just the latency and the respective IP.

**sort -n:**

sort: It sorts the lines of input.

-n: This is an option for numeric-sort.

-d: This sorts based on the dictionary order(only alphanumeric characters are considered).

-r: This is reverse sort.

Here sorting is done based on the latency of an IP.

**awk '{print $2 " " $1 "ms"}':**

awk is used again to reverse the printing order by specifying the column numbers.

**head -n 10:**

-n: This takes number of lines to be displayed.

The first 10 lines of output are printed.

**Output:-**

The command will return the **Ten** IP addresses which have the least latency(time taken for ping).

Here is the attached screenshot of the output