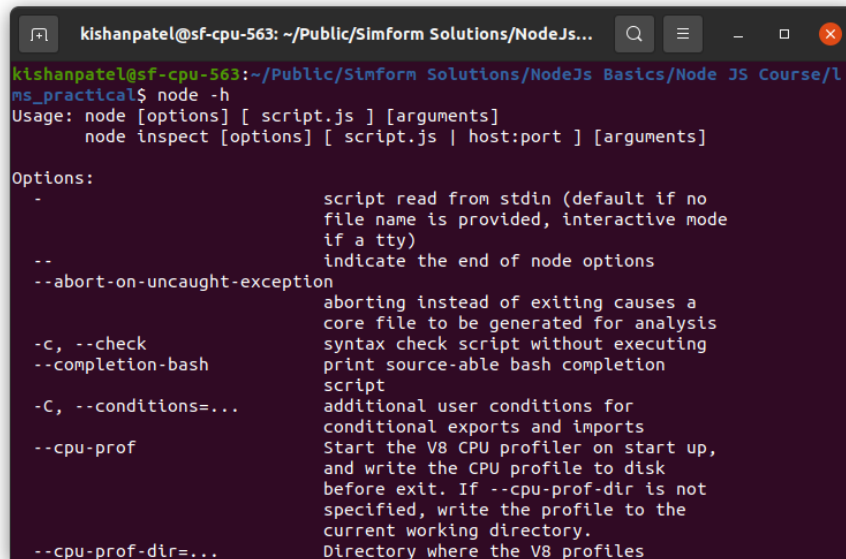


NodeJs Commands

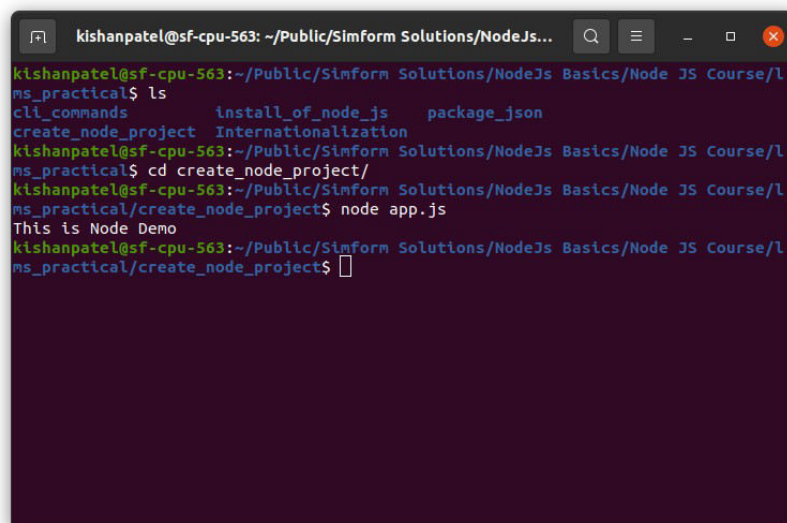
1. Get help: node -h



```
kishanpatel@sf-cpu-563: ~/Public/Simform Solutions/NodeJs...
kishanpatel@sf-cpu-563:~/Public/Simform Solutions/NodeJs Basics/Node JS Course/l
ms_practical$ node -h
Usage: node [options] [ script.js ] [arguments]
       node inspect [options] [ script.js | host:port ] [arguments]

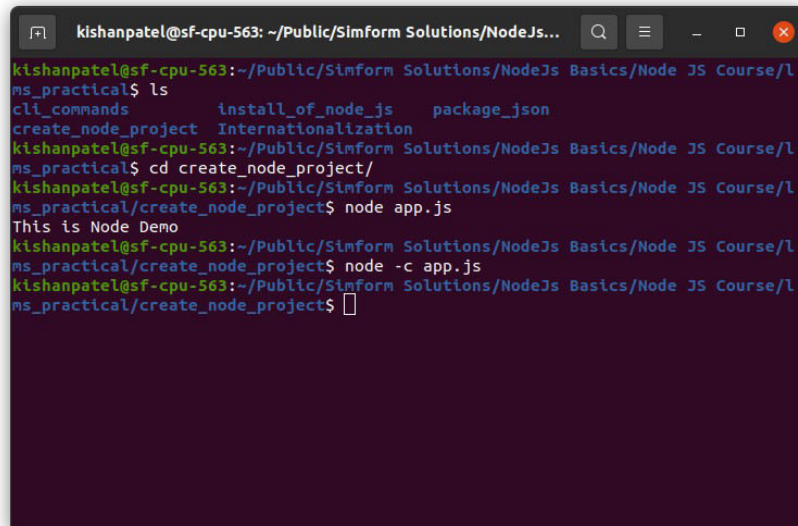
Options:
-          script read from stdin (default if no
           file name is provided, interactive mode
           if a tty)
--        indicate the end of node options
--abort-on-uncaught-exception aborting instead of exiting causes a
                               core file to be generated for analysis
-c, --check syntax check script without executing
--completion-bash print source-able bash completion
                  script
-C, --conditions=... additional user conditions for
                       conditional exports and imports
--cpu-prof Start the V8 CPU profiler on start up,
            and write the CPU profile to disk
            before exit. If --cpu-prof-dir is not
            specified, write the profile to the
            current working directory.
--cpu-prof-dir=... Directory where the V8 profiles
```

2 Run script: node script_name.js



```
kishanpatel@sf-cpu-563: ~/Public/Simform Solutions/NodeJs...
kishanpatel@sf-cpu-563:~/Public/Simform Solutions/NodeJs Basics/Node JS Course/l
ms_practical$ ls
cli_commands      install_of_node_js  package_json
create_node_project Internationalization
kishanpatel@sf-cpu-563:~/Public/Simform Solutions/NodeJs Basics/Node JS Course/l
ms_practical$ cd create_node_project/
kishanpatel@sf-cpu-563:~/Public/Simform Solutions/NodeJs Basics/Node JS Course/l
ms_practical/create_node_project$ node app.js
This is Node Demo
kishanpatel@sf-cpu-563:~/Public/Simform Solutions/NodeJs Basics/Node JS Course/l
ms_practical/create_node_project$
```

3. Check syntax without executing script: `node -c app.js`

A terminal window with a dark background and light-colored text. The window title is "kishanpatel@sf-cpu-563: ~/Public/Simform Solutions/NodeJs...". The terminal shows a series of commands and their outputs. The user is in a directory structure: ~/Public/Simform Solutions/NodeJs Basics/Node JS Course/l. They run 'ls' and see a list of files: cli_commands, install_of_node_js, package_json, create_node_project, and Internationalization. Then they run 'cd create_node_project/'. Next, they run 'node app.js' and see the output "This is Node Demo". Finally, they run 'node -c app.js' and see no output, indicating the syntax is correct. The prompt changes from 'ms_practical\$' to 'ms_practical/create_node_project\$' as they navigate the directory structure.

```
kishanpatel@sf-cpu-563: ~/Public/Simform Solutions/NodeJs Basics/Node JS Course/l
ms_practical$ ls
cli_commands      install_of_node_js  package_json
create_node_project  Internationalization
kishanpatel@sf-cpu-563:~/Public/Simform Solutions/NodeJs Basics/Node JS Course/l
ms_practical$ cd create_node_project/
kishanpatel@sf-cpu-563:~/Public/Simform Solutions/NodeJs Basics/Node JS Course/l
ms_practical/create_node_project$ node app.js
This is Node Demo
kishanpatel@sf-cpu-563:~/Public/Simform Solutions/NodeJs Basics/Node JS Course/l
ms_practical/create_node_project$ node -c app.js
kishanpatel@sf-cpu-563:~/Public/Simform Solutions/NodeJs Basics/Node JS Course/l
ms_practical/create_node_project$
```

4. print V8 options : `node --v8-options`

5: print node version: `node -v`