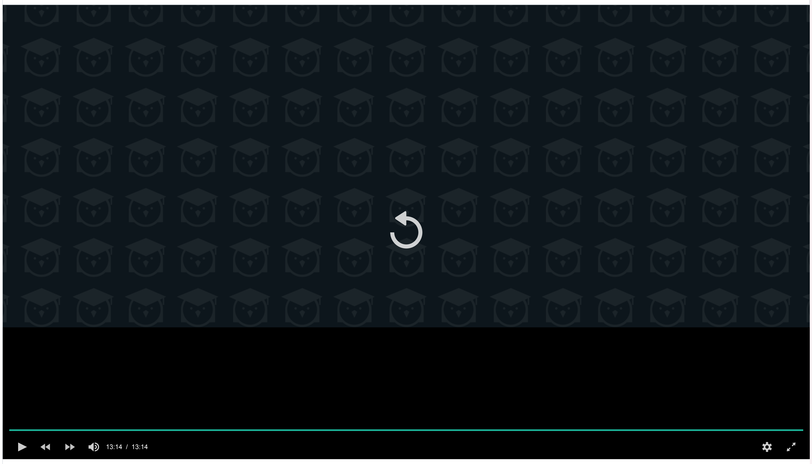
Lecture: Configuring Logging



In this lesson, we'll look into some of the options that we have when configuring logging with NGINX.

Note: the commands in this video are run as the `root` user.

Documentation For This Video

- NGINX `http_log` module
- NGINX `log_format` directive
- NGINX `access_log` directive
- NGINX `error_log` directive

Configuring Access Logs

The two main types of logging that we'll worry about with NGINX are going to be logging of requests ("access logging") and also error logging. Configuring access logs is more involved than configuring error logging, so let's investigate that more by looking at what we already have configured in `/etc/nginx/nginx.conf`:

/etc/nginx/nginx.conf (partial)

```
    log_format  main  '$remote_addr - $remote_user [$time_local] "$request" '
                      '$status $body_bytes_sent "$http_referer" '
                      '"$http_user_agent" "$http_x_forwarded_for"';
```

```
access_log  /var/log/nginx/access.log  main;</code></pre>
```

The `log_format` line here sets the name of the log format to main and then sets the string to be logged by combining many of the available NGINX variables. By itself, creating this format doesn't do anything until it's used as part of the `access_log` line in combination with the path to the log file.

It's fairly common to read NGINX configuration for a log that looks something like this:

```
    access_log /var/log/nginx/example.log combined;
```

The combined format is a pre-defined log format type.

Configuring Error Logs

Error logging is simpler than access logging because there is no format to set, but we do need to understand the logging levels that exist:

1. debug
2. info
3. notice
4. warn
5. error
6. crit
7. alert
8. emerg

With that list in mind, as the number gets larger the "severity" increases, and every level of a lower severity contains content logged at the severities higher than itself. The error log that is set up by default for NGINX looks like this:

/etc/nginx/nginx.conf (partial)

```
error_log  /var/log/nginx/error.log warn;
```

With the warn level set, anything logged at a notice, info, or debug level will not be written to the log file.

Utilizing Syslog

It's common to want to log using syslog instead of a file so that logs can be aggregated with a logging server or service like PaperTrail. NGINX supports utilizing syslog through the `error_log` and `access_log` directives that we've already looked at. Let's change our blog and notes servers to log to a locally running syslog server by adding the following line to each file's server block:

```
access_log syslog:/dev/log combined;
```

We're using the default combined format and logging to the default `syslog` socket at /dev/log. The server attribute could also be set to a remote address using a domain or IP address instead of a local unix: socket. Now we can refresh our NGINX configuration, make a few requests and see what is logged:

```
[root] $ systemctl reload nginx
[root] $ curl --header "Host: blog.example.com" localhost
...
[root] $ curl --header "Host: notes.example.com" localhost
...
```

Depending on the system that you're running on the file that we will read from will be a little different. On CentOS it will be /var/log/messages, but on debian based systems it will be /var/log/syslog.

Here's how we can check the last 2 logged lines in the file:

```
[root] $ tail -n 2 /var/log/messages
Mar 19 22:02:25 keiththomps3 journal: keiththomps3.mylabserver.com nginx: 127.0.0.1 - - [19/Mar/2018:22:02:25 +0000] "GET / HTTP/1.1" 200 53332 "-" "curl/7.29.0"
Mar 19 22:02:33 keiththomps3 journal: keiththomps3.mylabserver.com nginx: 127.0.0.1 - - [19/Mar/2018:22:02:33 +0000] "GET / HTTP/1.1" 200 1863 "-" "curl/7.29.0"
```

Custom Log Format

The log messages that main provides are not very helpful because we don't know which virtual host was requested. Let's create our own custom log_format that will include this information that we can use for our virtual hosts. Since we want to use this in many files, we'll add it to the top level http context within /etc/nginx/nginx.conf:

/etc/nginx/nginx.conf (partial)

```
    log_format vhost '$host $remote_addr - $remote_user [$time_local] "$request" '
                     '$status $body_bytes_sent "$http_referer" '
                     '"$http_user_agent" "$http_x_forwarded_for"';
```

The format is the same as main except we added the $host value. Now the line that we used in our virtual hosts can be changed to this:

```
  access_log syslog:/dev/log vhost;
```

After a reload and a few more requests we should be able to tell what server received each request:

```
[root] $ systemctl reload nginx
[root] $ curl --header "Host: notes.example.com" localhost 2&>1
[root] $ curl --header "Host: blog.example.com" localhost 2&>1
[root] $ tail -n 2 /var/log/messages
Mar 19 22:10:18 keiththomps3 journal: keiththomps3.mylabserver.com nginx: notes.example.com 127.0.0.1 - - [19/Mar/2018:22:10:18 +0000] "GET / HTTP/1.1" 200 1863 "-" "curl/7.29.0"
"-"
Mar 19 22:10:24 keiththomps3 journal: keiththomps3.mylabserver.com nginx: blog.example.com 127.0.0.1 - - [19/Mar/2018:22:10:24 +0000] "GET / HTTP/1.1" 200 53324 "-" "curl/7.29.0"
"-"
```