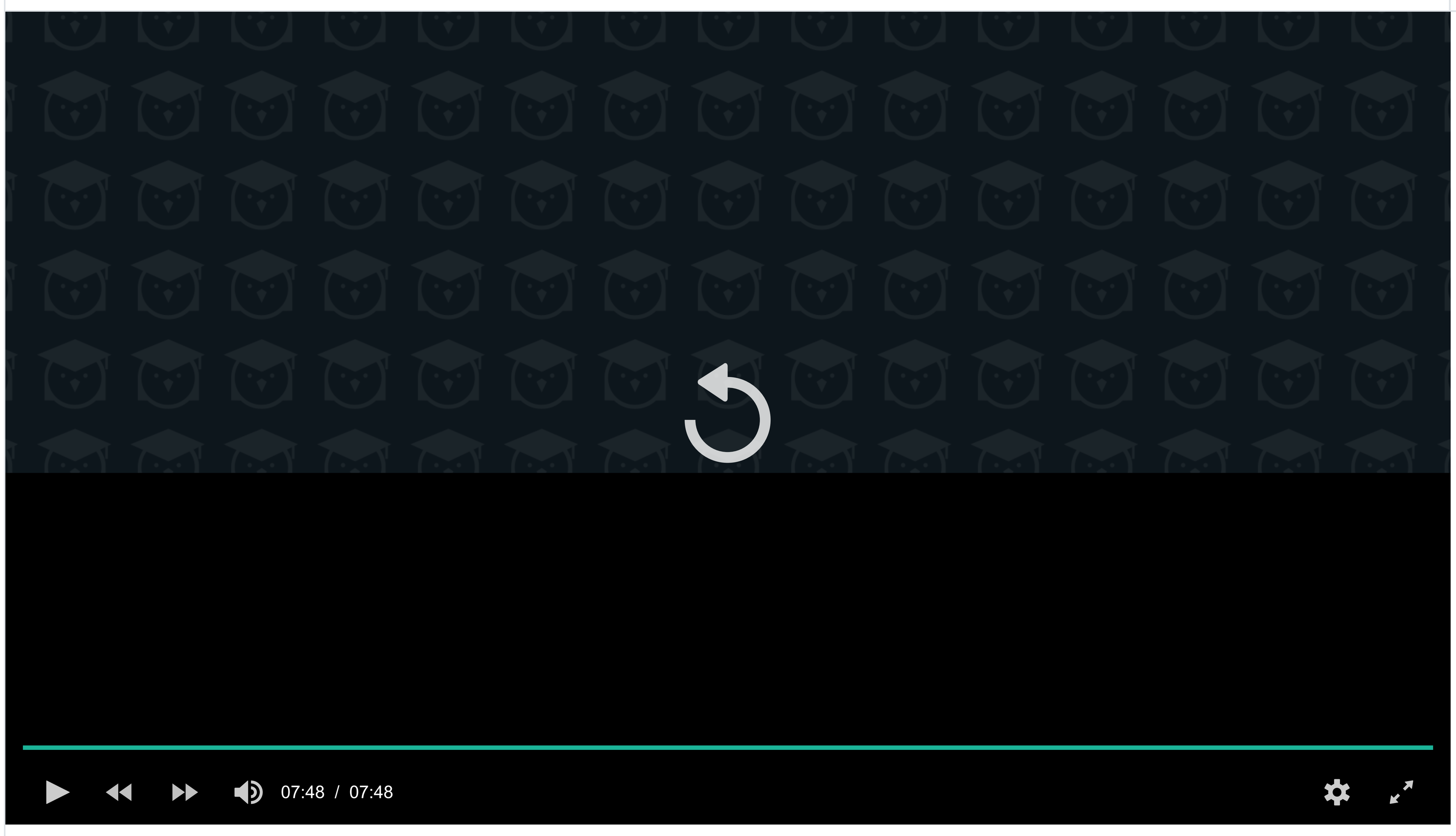
← NGINX Web Server Deep Dive

Lecture: Load Balancing to Multiple Servers



Flash Cards | Notes | Transcript | Downloads

Looks like you haven't created any flash cards for this deck yet.

Create My First Flash Card

Popular Decks | Popular Cards

Nginx Web Server Deep Dive

NGINX Web Server Deep Dive

NGINX Web Server Deep Dive

In this lesson, we look at how we can utilize NGINX as a load balancer to distribute traffic between multiple instances of the same application.

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

Documentation For This Video

- NGINX `http_upstream` module
- NGINX `upstream` directive
- NGINX `server` directive from the `upstream` module

Running Multiple Instances

To demonstrate load balancing, we need to run more than one instance of an application. We're going to create some duplicate services for `photos.example.com`, where the separate web-clients all run on different ports. To do this, we'll need to duplicate the `web-client.service` file a few times and add an additional `PORT` environment variable.

```
[root] $ cp /etc/systemd/system/web-client{,2}.service
[root] $ cp /etc/systemd/system/web-client{,3}.service
```

Now we set the `PORT` value in the new files:

/etc/systemd/system/web-client2.service

```
[Unit]
Description=S3 Photo App Node.js service
After=network.target photo-filter.target photo-storage.target

[Service]
Restart=always
User=nobody
Group=nobody
Environment=NODE_ENV=production
Environment=AWS_ACCESS_KEY_ID=YOUR_AWS_KEY_ID
Environment=AWS_SECRET_ACCESS_KEY=YOUR_AWS_SECRET_KEY
Environment=PORT=3100
ExecStart=/srv/www/s3photoapp/apps/web-client/bin/www


[Install]
WantedBy=multi-user.target
```

And for the third service:

/etc/systemd/system/web-client3.service

```
[Unit]
Description=S3 Photo App Node.js service
After=network.target photo-filter.target photo-storage.target

[Service]
Restart=always
User=nobody
Group=nobody
Environment=NODE_ENV=production
Environment=AWS_ACCESS_KEY_ID=YOUR_AWS_KEY_ID
Environment=AWS_SECRET_ACCESS_KEY=YOUR_AWS_SECRET_KEY
Environment=PORT=3101
ExecStart=/srv/www/s3photoapp/apps/web-client/bin/www


[Install]
WantedBy=multi-user.target
```

Lastly, we need to start these services:

```
[root] $ systemctl start web-client2
[root] $ systemctl start web-client3
```

Creating a Server Group

To load balance the traffic between our three identical services with NGINX we'll need to use the `http_upstream` module and the `upstream` directive. This directive creates a new context where we configure all of the load balancing behavior. Let's create our server group now:

/etc/nginx/conf.d/photos.example.com

```
upstream photos {
    server 127.0.0.1:3000;
    server 127.0.0.1:3100;
    server 127.0.0.1:3101;
}

server {
    listen 80;
    server_name photos.example.com;

client_max_body_size 5m;

location / {
    proxy_pass http://photos;
    proxy_http_version 1.1;
    proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
    proxy_set_header X-Real-IP  $remote_addr;
    proxy_set_header Upgrade $http_upgrade;
    proxy_set_header Connection "upgrade";
}

location ~* \.(js|css|png|jpe?g|gif) {
    root /var/www/photos.example.com;
}

}
```

Our `upstream` context is named `photos` and defines three servers. These servers us the `server` directive provided by the upstream module and their definition is similar to what we would provide to `proxy_pass` except that we will leave off the `http` or `https` scheme. Besides setting up the `upstream` context, we also needed to change our `proxy_pass` to proxy traffic to the server group instead of a specific server, and we did this by replacing `127.0.0.1:3000` with the name of our server group `photos`.

If we reload NGINX and make 4 or 5 requests to `photos.example.com` we can check stdout from each service to see that they each received at least 1 request. Each will look something like this:

```
[root] $ systemctl status web-client3
? web-client3.service - S3 Photo App Node.js service
   Loaded: loaded (/etc/systemd/system/web-client3.service; disabled; vendor preset: disabled)
   Active: active (running) since Sun 2018-03-18 12:59:03 UTC; 15min ago
 Main PID: 1511 (node)
   CGroup: /system.slice/web-client3.service
           ??1511 node /srv/www/s3photoapp/apps/web-client/bin/www

Mar 18 12:59:03 keiththomps3.mylabserver.com systemd[1]: Started S3 Photo App Node.js service.
Mar 18 12:59:03 keiththomps3.mylabserver.com systemd[1]: Starting S3 Photo App Node.js service...
Mar 18 12:59:07 keiththomps3.mylabserver.com www[1511]: Listening on port 3101
Mar 18 13:10:34 keiththomps3.mylabserver.com www[1511]: GET /favicon.ico 404 118.663 ms - 150
Mar 18 13:14:05 keiththomps3.mylabserver.com www[1511]: GET / 200 413.474 ms - 1745
```

Exceeded my Expectations | Room for Improvement

Mark Complete & Start Next Lesson