Assignment for Week 4: Web service and basic load-balancing

Assignment for Week 4: Web service and basic load-balancing

Put up a web page on your system in its document root. Send me the URLs to that page (`http://ip-x-x-x-x.aws.amazon.com/foo.html` and I'll try to retrieve it.

In addition, find log messages showing accesses to a web page in your web server document root, and send me samples of those log messages.

Provide the domain name associated with your load balancer so I can test web server access to it.

Describe the results of testing your load balancer when you shut down one, then both, back-end instances, and when you restore them.

After shutting down one instance:

After shutting down both:

After restoring them one by one:

Provide excerpts from web server logs from each of your web server instances showing load balancer health checks and accesses to your web content.

Indicate the Puppet code you use to replicate content between your web server instances.

Put up a web page on your system in its document root. Send me the URLs to that page (`http://ip-x-x-x-x.aws.amazon.com/foo.html` and I'll try to retrieve it.

Currently it is "http://ec2-34-221-75-250.us-west-2.compute.amazonaws.com/index.html"

In addition, find log messages showing accesses to a web page in your web server document root, and send me samples of those log messages.

```
ubuntu@ip-10-0-6-163:~$ tail -f /var/log/nginx/access.log
128.223.222.50 - - [17/Jul/2019:00:26:32 +0000] "GET / HTTP/1.1" 200 293 "-"
"Mozilla/5.0 (Macintosh; Intel Mac OS X 10_14_5) AppleWebKit/537.36 (KHTML,
like Gecko) Chrome/75.0.3770.100 Safari/537.36"
128.223.222.50 - - [17/Jul/2019:00:26:32 +0000] "GET /index.css HTTP/1.1" 200
125 "http://ec2-54-201-34-210.us-west-2.compute.amazonaws.com/" "Mozilla/5.0
(Macintosh; Intel Mac OS X 10_14_5) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/75.0.3770.100 Safari/537.36"
128.223.222.50 - - [17/Jul/2019:00:29:29 +0000] "GET / HTTP/1.1" 304 0 "-"
"Mozilla/5.0 (Macintosh; Intel Mac OS X 10_14_5) AppleWebKit/537.36 (KHTML,
like Gecko) Chrome/75.0.3770.100 Safari/537.36"
128.223.222.50 - - [17/Jul/2019:00:29:29 +0000] "GET /index.css HTTP/1.1" 304
0 "http://ec2-54-201-34-210.us-west-2.compute.amazonaws.com/" "Mozilla/5.0
(Macintosh; Intel Mac OS X 10_14_5) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/75.0.3770.100 Safari/537.36"
```

Provide the domain name associated with your load balancer so I can test web server access to it.

```
http://maxitestinglb-1927444259.us-west-2.elb.amazonaws.com/
```

Describe the results of testing your load balancer when you shut down one, then both, back-end instances, and when you restore them.

After shutting down one instance:

- In the Instances tab of Load Balancer page on AWS console, the instance I turned off has a status "OutOfService."
- Checking the access log of the only instance left, when accessing the domain name associated with the load balancer, I could see my access record.

After shutting down both:

- In the Instances tab of Load Balancer page on AWS console, both instances have a status "OutOfService."
- When accessing the domain name associated with the load balancer, a 503 error is returned.

After restoring them one by one:

- Able to access load balancer's domain name, though both of two instances public domain names have changed.

Provide excerpts from web server logs from each of your web server instances showing load balancer health checks and accesses to your web content.

```
ubuntu@ip-10-0-6-163:~$ tail -f /var/log/nginx/access.log
10.0.6.253 - - [17/Jul/2019:23:30:05 +0000] "GET /index.html HTTP/1.1" 200 388
"-" "ELB-HealthChecker/1.0"
10.0.6.195 - - [17/Jul/2019:23:30:06 +0000] "GET /index.html HTTP/1.1" 200 388
"-" "ELB-HealthChecker/1.0"
10.0.6.253 - - [17/Jul/2019:23:30:35 +0000] "GET /index.html HTTP/1.1" 200 388
"-" "ELB-HealthChecker/1.0"
10.0.6.195 - - [17/Jul/2019:23:30:36 +0000] "GET /index.html HTTP/1.1" 200 388
"-" "ELB-HealthChecker/1.0"
10.0.6.253 - - [17/Jul/2019:23:31:05 +0000] "GET /index.html HTTP/1.1" 200 388
"-" "ELB-HealthChecker/1.0"
10.0.6.195 - - [17/Jul/2019:23:31:06 +0000] "GET /index.html HTTP/1.1" 200 388
"-" "ELB-HealthChecker/1.0"
10.0.6.253 - - [17/Jul/2019:23:31:35 +0000] "GET /index.html HTTP/1.1" 200 388
"-" "ELB-HealthChecker/1.0"
10.0.6.195 - - [17/Jul/2019:23:31:36 +0000] "GET /index.html HTTP/1.1" 200 388
"-" "ELB-HealthChecker/1.0"
10.0.6.253 - - [17/Jul/2019:23:32:05 +0000] "GET /index.html HTTP/1.1" 200 388
"-" "ELB-HealthChecker/1.0"
10.0.6.195 - - [17/Jul/2019:23:32:06 +0000] "GET /index.html HTTP/1.1" 200 388
"-" "ELB-HealthChecker/1.0"
```

```
ubuntu@ip-10-0-6-235:~$ tail -f /var/log/nginx/access.log
10.0.6.195 - - [17/Jul/2019:23:30:51 +0000] "GET /index.html HTTP/1.1" 200 388
"-" "ELB-HealthChecker/1.0"
10.0.6.253 - - [17/Jul/2019:23:30:51 +0000] "GET /index.html HTTP/1.1" 200 388
"-" "ELB-HealthChecker/1.0"
10.0.6.195 - - [17/Jul/2019:23:31:21 +0000] "GET /index.html HTTP/1.1" 200 388
"-" "ELB-HealthChecker/1.0"
10.0.6.253 - - [17/Jul/2019:23:31:21 +0000] "GET /index.html HTTP/1.1" 200 388
"-" "ELB-HealthChecker/1.0"
10.0.6.195 - - [17/Jul/2019:23:31:51 +0000] "GET /index.html HTTP/1.1" 200 388
"-" "ELB-HealthChecker/1.0"
10.0.6.253 - - [17/Jul/2019:23:31:51 +0000] "GET /index.html HTTP/1.1" 200 388
"-" "ELB-HealthChecker/1.0"
10.0.6.195 - - [17/Jul/2019:23:32:21 +0000] "GET /index.html HTTP/1.1" 200 388
"-" "ELB-HealthChecker/1.0"
10.0.6.253 - - [17/Jul/2019:23:32:21 +0000] "GET /index.html HTTP/1.1" 200 388
"-" "ELB-HealthChecker/1.0"
10.0.6.195 - - [17/Jul/2019:23:32:51 +0000] "GET /index.html HTTP/1.1" 200 388
"-" "ELB-HealthChecker/1.0"
10.0.6.253 - - [17/Jul/2019:23:32:51 +0000] "GET /index.html HTTP/1.1" 200 388
"-" "ELB-HealthChecker/1.0"
```

Indicate the Puppet code you use to replicate content between your web server instances.

In my repository, the module "pageserver_jerryx" was used to deploy the web page server.