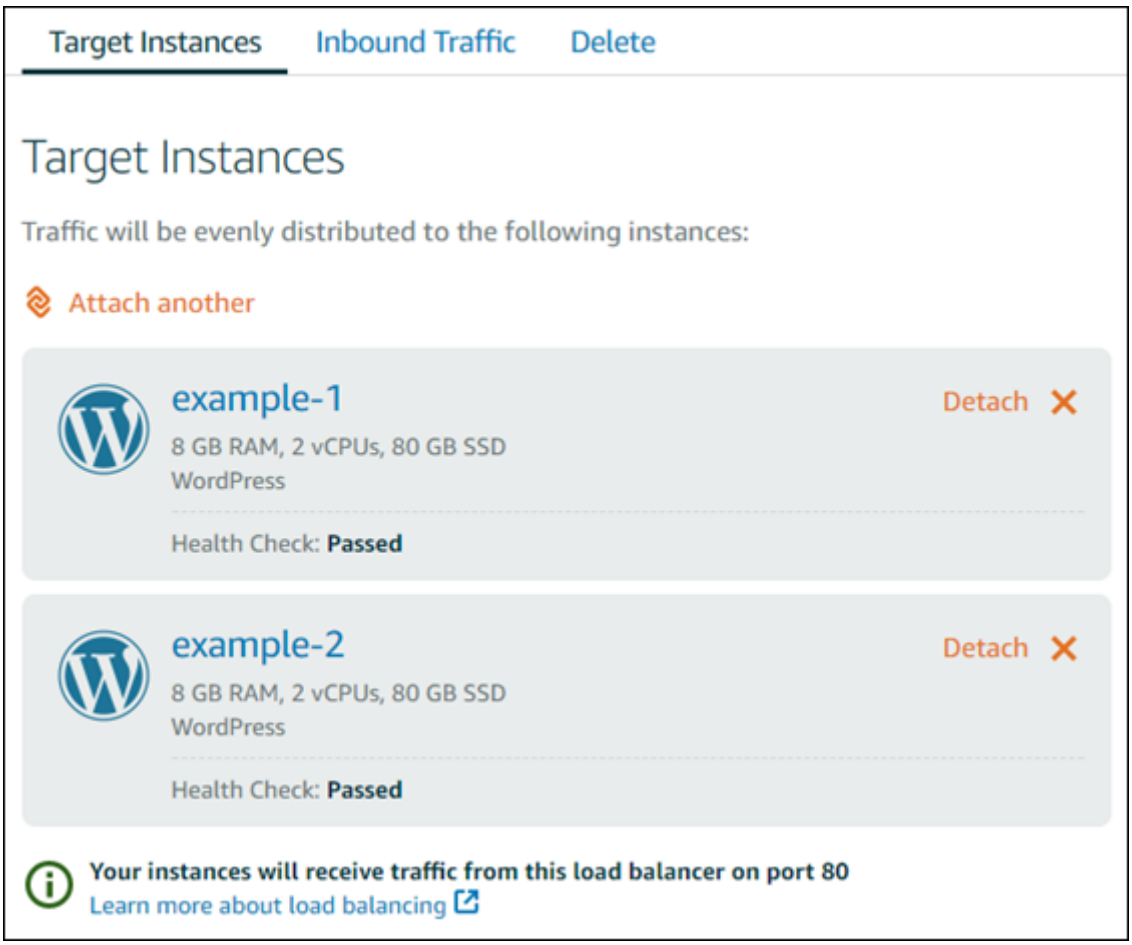
Lightsail load balancer health checking

Note
By default, Lightsail performs health checks on your instances at the root ("/") of your web application. The health checks are used to monitor the health of the target instances so that the load balancer can send requests only to the healthy instances. You'll receive a **Passed** or **Failed** message on each instance attached to the load balancer.

Last updated: October 14, 2019

Health checking starts as soon as you attach your Lightsail instances to your load balancer, and it occurs every 30 seconds thereafter. You can see the health check status on the load balancer management page.

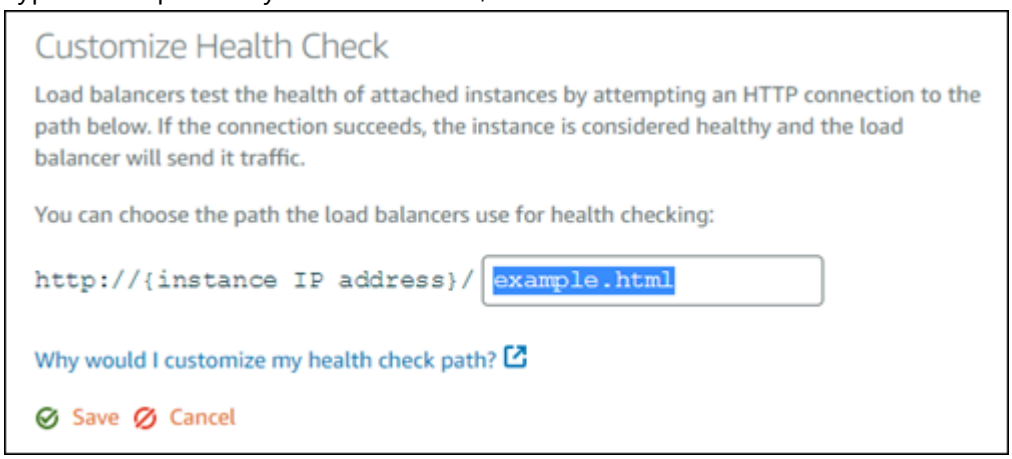


Customize your health check path

You might want to customize your health check path. For example, if your home page loads slowly or has a lot of images on it, you can configure Lightsail to check a different page that loads faster.

1. On the Lightsail home page, choose **Networking**.
2. Choose your load balancer to manage it.
3. On the **Target instances** tab, choose **Customize health checking**.

4. Type a valid path for your health check, and then choose **Save**.



Health check metrics

The following metrics can help you diagnose health check problems. Use the AWS Command Line Interface or the Lightsail API to return information about the specific health check metric.

- `ClientTLSNegotiationErrorCount` - The number of TLS connections initiated by the client that did not establish a session with the load balancer. Possible causes include a mismatch of ciphers or protocols.

  `Statistics`: The most useful statistic is `Sum`.

- `HealthyHostCount` - The number of target instances that are considered healthy.

  `Statistics`: The most useful statistic are `Average`, `Minimum`, and `Maximum`.

- `UnhealthyHostCount` - The number of target instances that are considered unhealthy.

  `Statistics`: The most useful statistic are `Average`, `Minimum`, and `Maximum`.

- `HTTPCode_LB_4XX_Count` - The number of HTTP 4XX client error codes that originate from the load balancer. Client errors are generated when requests are malformed or incomplete. These requests have not been received by the target instance. This count does not include any response codes generated by the target instances.

  `Statistics`: The most useful statistic is `Sum`. Note that `Minimum`, `Maximum`, and `Average` all return `1`.

- `HTTPCode_LB_5XX_Count` - The number of HTTP 5XX server error codes that originate from the load balancer. This count does not include any response codes generated by the target instances.

  `Statistics`: The most useful statistic is `Sum`. Note that `Minimum`, `Maximum`, and `Average` all return `1`. Note that `Minimum`, `Maximum`, and `Average` all return `1`.

- `HTTPCode_Instance_2XX_Count` - The number of HTTP response codes generated by the target instances. This does not include any response codes generated by the load balancer.

  `Statistics`: The most useful statistic is `Sum`. Note that `Minimum`, `Maximum`, and `Average` all return `1`.

- `HTTPCode_Instance_3XX_Count` - The number of HTTP response codes generated by the target instances. This does not include any response codes generated by the load balancer.

`Statistics`: The most useful statistic is `Sum`. Note that `Minimum`, `Maximum`, and `Average` all return `1`.

- **`HTTPCode_Instance_4XX_Count`** - The number of HTTP response codes generated by the target instances. This does not include any response codes generated by the load balancer.

  `Statistics`: The most useful statistic is `Sum`. Note that `Minimum`, `Maximum`, and `Average` all return `1`.

- **`HTTPCode_Instance_5XX_Count`** - The number of HTTP response codes generated by the target instances. This does not include any response codes generated by the load balancer.

  `Statistics`: The most useful statistic is `Sum`. Note that `Minimum`, `Maximum`, and `Average` all return `1`.

- **`InstanceResponseTime`** - The time elapsed, in seconds, after the request leaves the load balancer until a response from the target instance is received.

  `Statistics`: The most useful statistic is `Average`.

- **`RejectedConnectionCount`** - The number of connections that were rejected because the load balancer had reached its maximum number of connections.

  `Statistics`: The most useful statistic is `Sum`.

- **`RequestCount`** - The number of requests processed over IPv4. This count includes only the requests with a response generated by a target instance of the load balancer.

  `Statistics`: The most useful statistic is `Sum`. Note that `Minimum`, `Maximum`, and `Average` all return `1`.