

Network Load Balancer Access

- Parallels Remote Application Server 18.3
- Parallels Remote Application Server 19.1

The Network Load Balancers access section is intended for deployment scenarios where third-party front-end load balancers such as Amazon Web Services (AWS) Elastic Load Balancers (ELBs) are used. It allows you to configure an alternate hostname and port number to be used by the Network Load Balancer (NLB). This is needed to separate hostnames and ports on which TCP and HTTPS communications are carried out because AWS load balancers don't support both specific protocols over the same port.

The following options are available.

- **Use alternate hostname:** Select this option and specify an alternate hostname. When the alternate hostname is enabled, all platform-specific Parallels Clients will use this hostname to connect to the RAS farm or site.
- **Use alternate port:** Select this option and specify an alternate port number. The port must not be used by any other component in the RAS farm or site. To reset the port number to the default value, click **Default**. When the alternate port is enabled, all platform-specific Parallels Clients will use this port to connect to the RAS farm or site. Note that RDP sessions in HTML5 Client will still be connecting to the standard SSL port (443).

In addition, the AWS Application Load Balancer (ALB), which handles HTTPs traffic required by the Parallels HTML5 Client, only supports specific cookies that are usually automatically generated. When a load balancer first receives a request from a client, it routes the request to a target and generates a cookie named **AWSALB**, which encodes information about the selected target. The load balancer then encrypts the cookie and includes it in the response to the client. When sticky sessions are enabled, the load balancer uses the cookie received from the client to route the traffic to the same target, assuming the target is registered successfully and is considered healthy. By default, Parallels RAS uses its own ASP.NET cookie named *_SessionId*. However, in this case you must customize the cookie specifying the mentioned AWS cookie for sticky sessions. This can be configured using the **Web cookie** field on the **Web Requests** tab.