# Parallels RAS Front-End Load Balancing using AWS Elastic Load Balancing (NLB and ALB)

- Parallels Remote Application Server

The below guide is a step-by-step configuration guide for deploying AWS Elastic load balancing (ELB) to front-end and load-balance Parallels RAS Environment.

## Prerequisites and Assumptions

It is assumed that reader has a basic understanding of both AWS ELB solutions (Application Load Balancer (ALB)/Network Load Balancer (NLB)) and Parallels RAS. This guide will focus on the configuration of AWS ELB and Parallels Secure Client Gateways load balancing. It is assumed that Parallels RAS environment have already been deployed and configured on EC2 instances with 443 Inbound rule on the Parallels Secure Client Gateways security groups.

***Note:*** *Steps 1-5 focusing on AWS NLB configuration that will allow connectivity from native Parallels Clients. Should you need to configure load balancing for HTML5 clients only or in addition to the native Parallels Clients, please also review the configuration at Step 6.*

*More information AWS Elastic Load Balancing available here: https://docs.aws.amazon.com/AmazonECS/latest/developerguide/service-load-balancing.html*

## Document process flow

The process that will be discussed in more detail is as illustrated below:

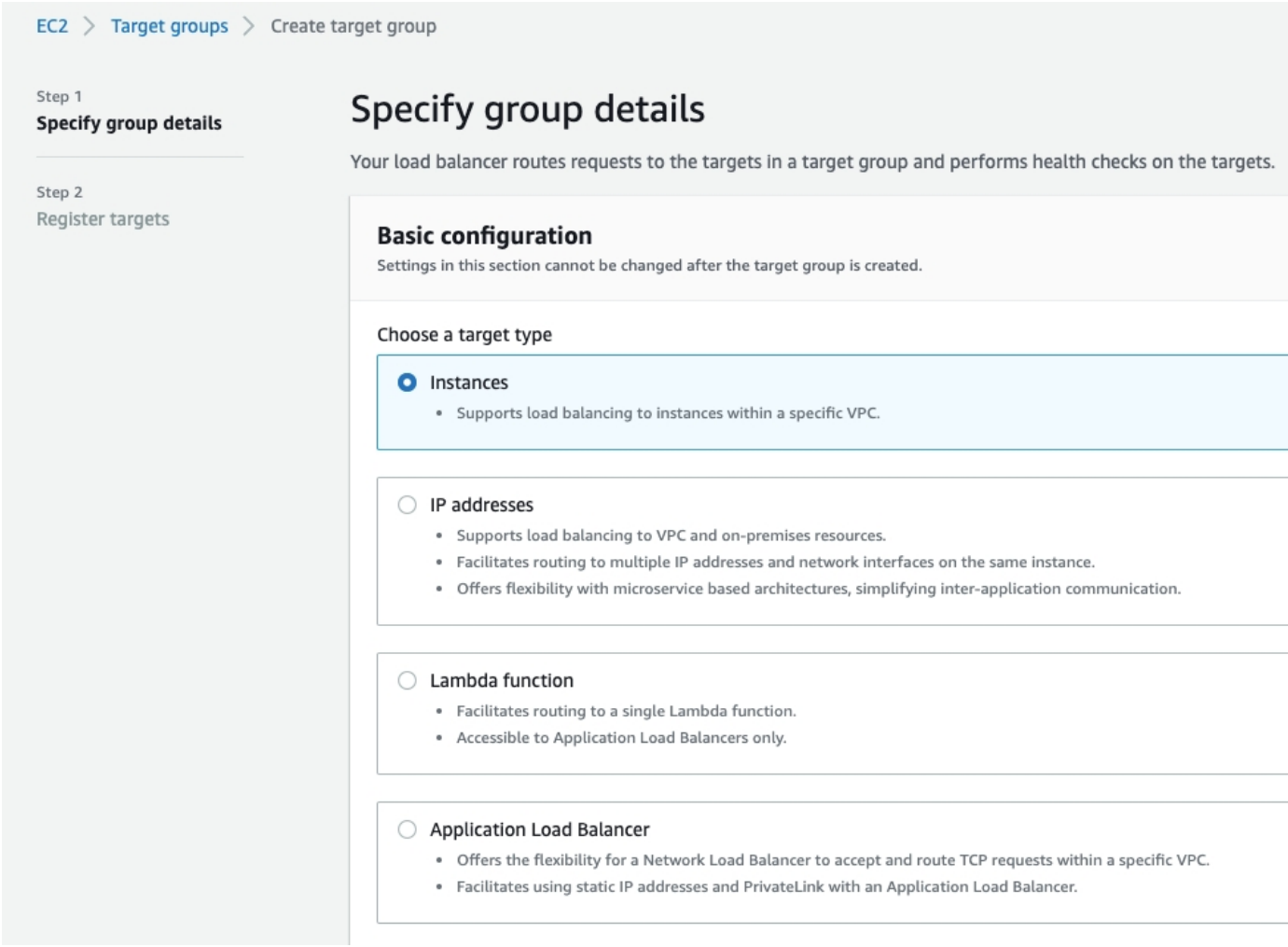
1. Configure your Target Group.
2. Configure Target Group attributes.
3. Choose the Load Balancer type.
4. Configure Load Balancer and Listener.
5. Test and Evaluate Load Balancing.
6. Using Network Load Balancer Access feature

## Step 1: Configure your Target Group

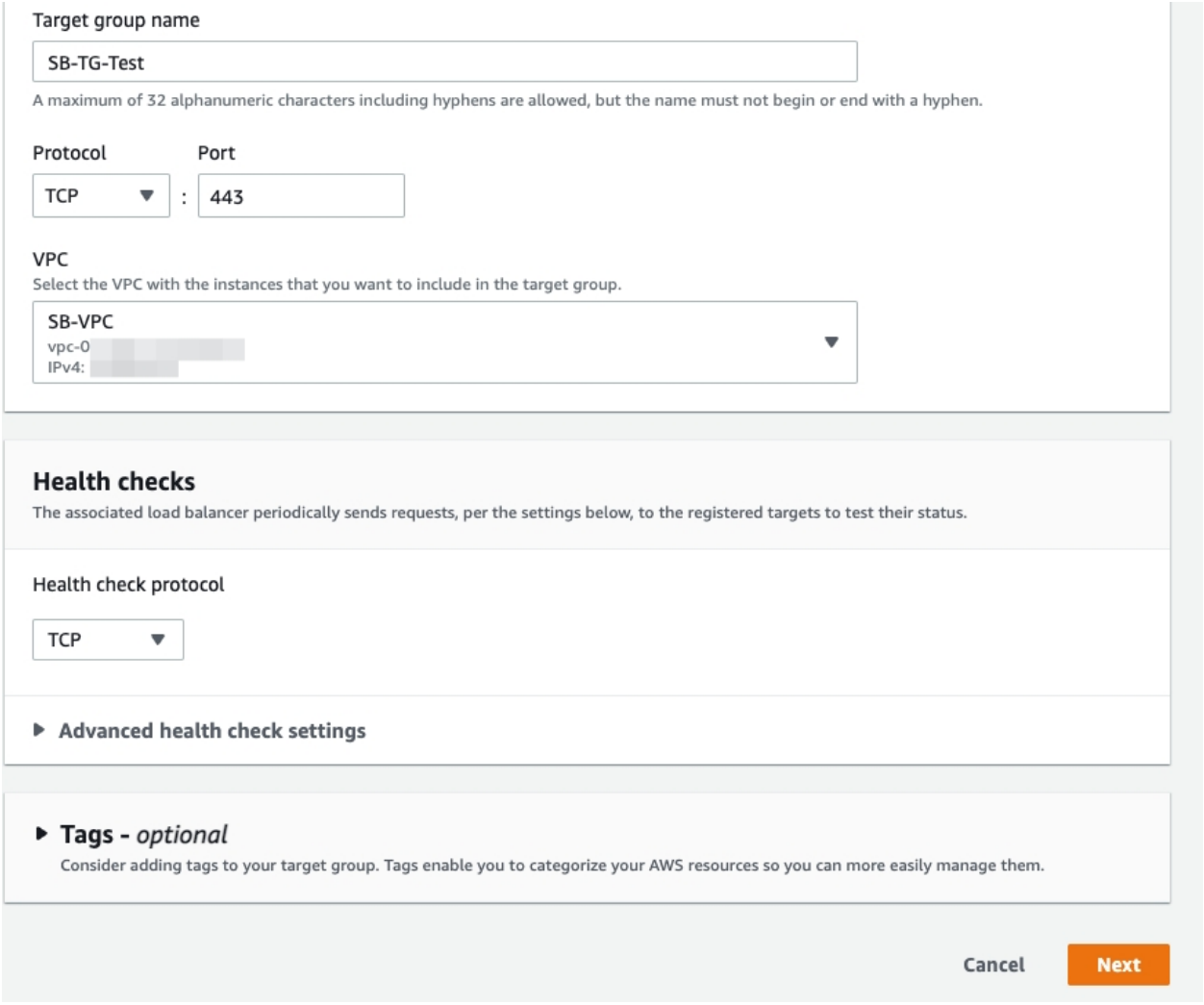
Create a target group, which is used in request routing. The rule for your listener routes requests to the registered targets in this target group. The load balancer checks the health of targets in this target group using the health check settings defined for the target group.

**To configure your target group**

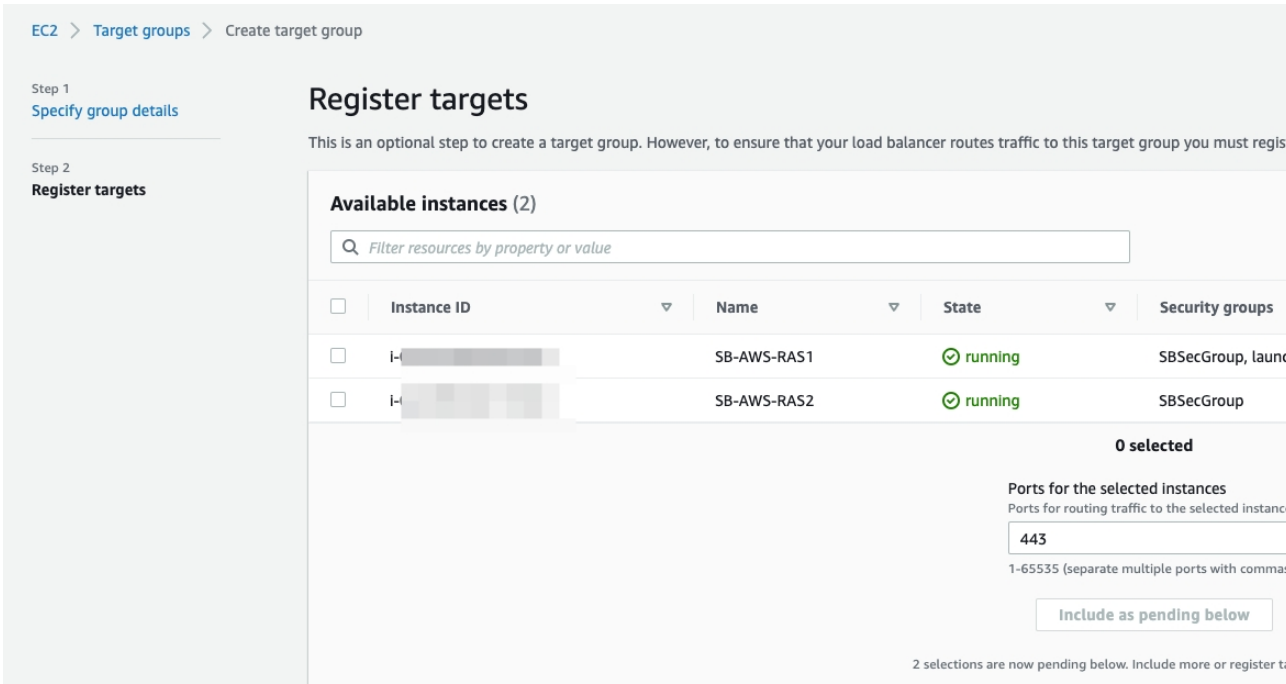
1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, under Load Balancing, choose Target Groups.
3. Choose Create target group.
4. Keep Target type as instance.



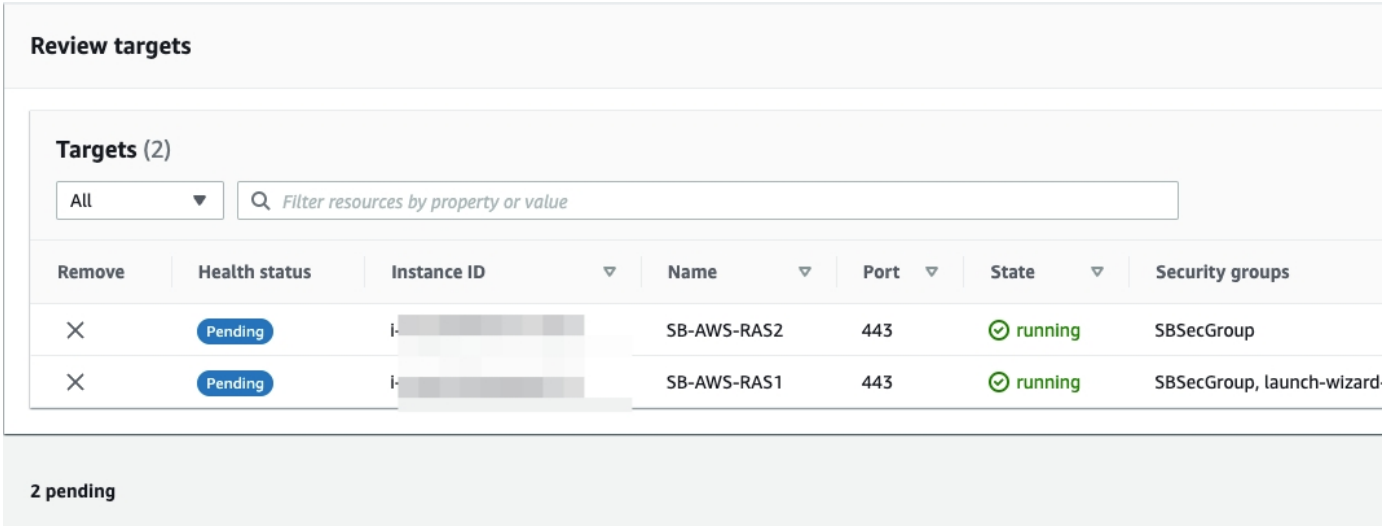

5. For **Target group name**, enter a name for the new target group.
6. Set Protocol as **TCP**, and Port as **443**.
7. Select the **VPC** containing your instances.
8. For **Health checks**, keep the default settings.
9. Choose **Next**.




10. On the **Register targets** page, complete the following steps. This is an optional step to create a target group. However, you must register your targets if you want to test your load balancer and ensure that it is routing traffic to your targets.
    1. For **Available instances**, select one or more instances.
    2. Keep the default port **443**, and choose **Include as pending below**.

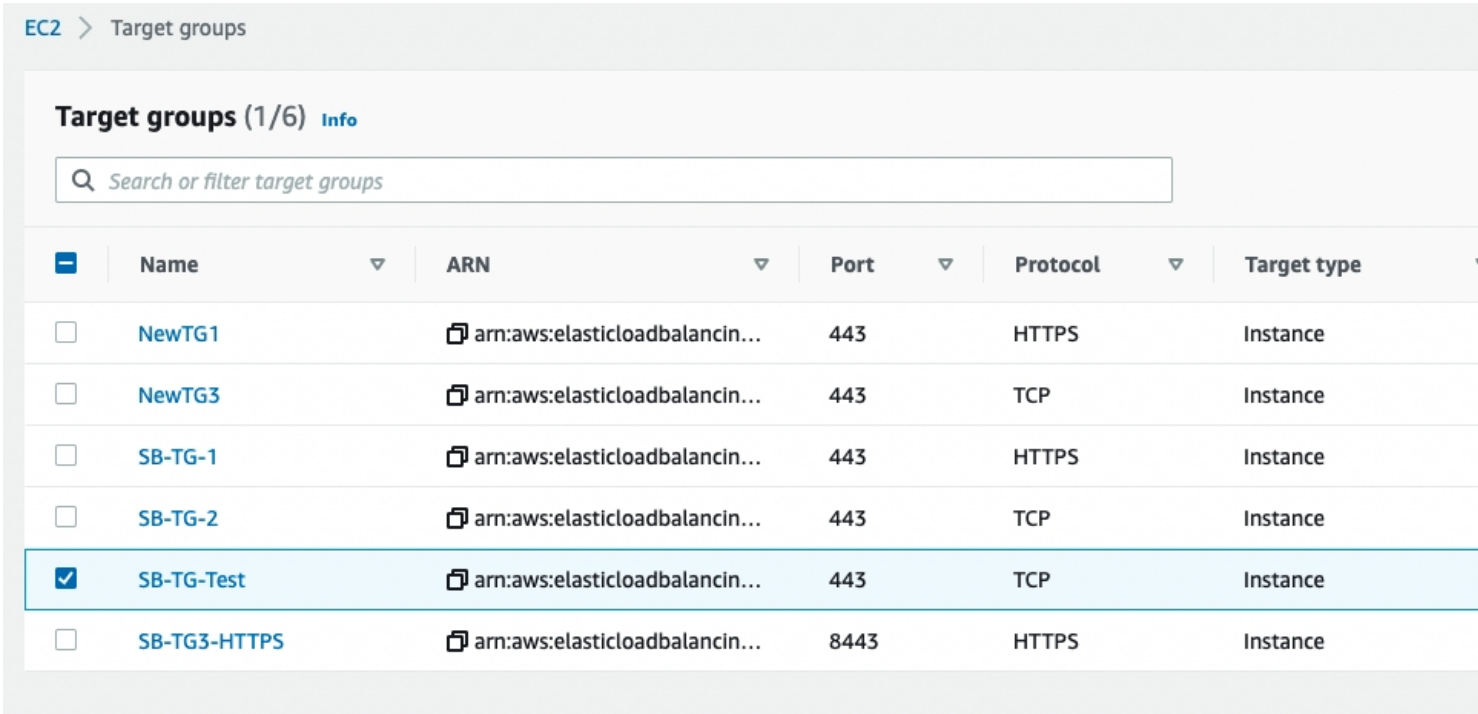



11. Click on **Create target group**.

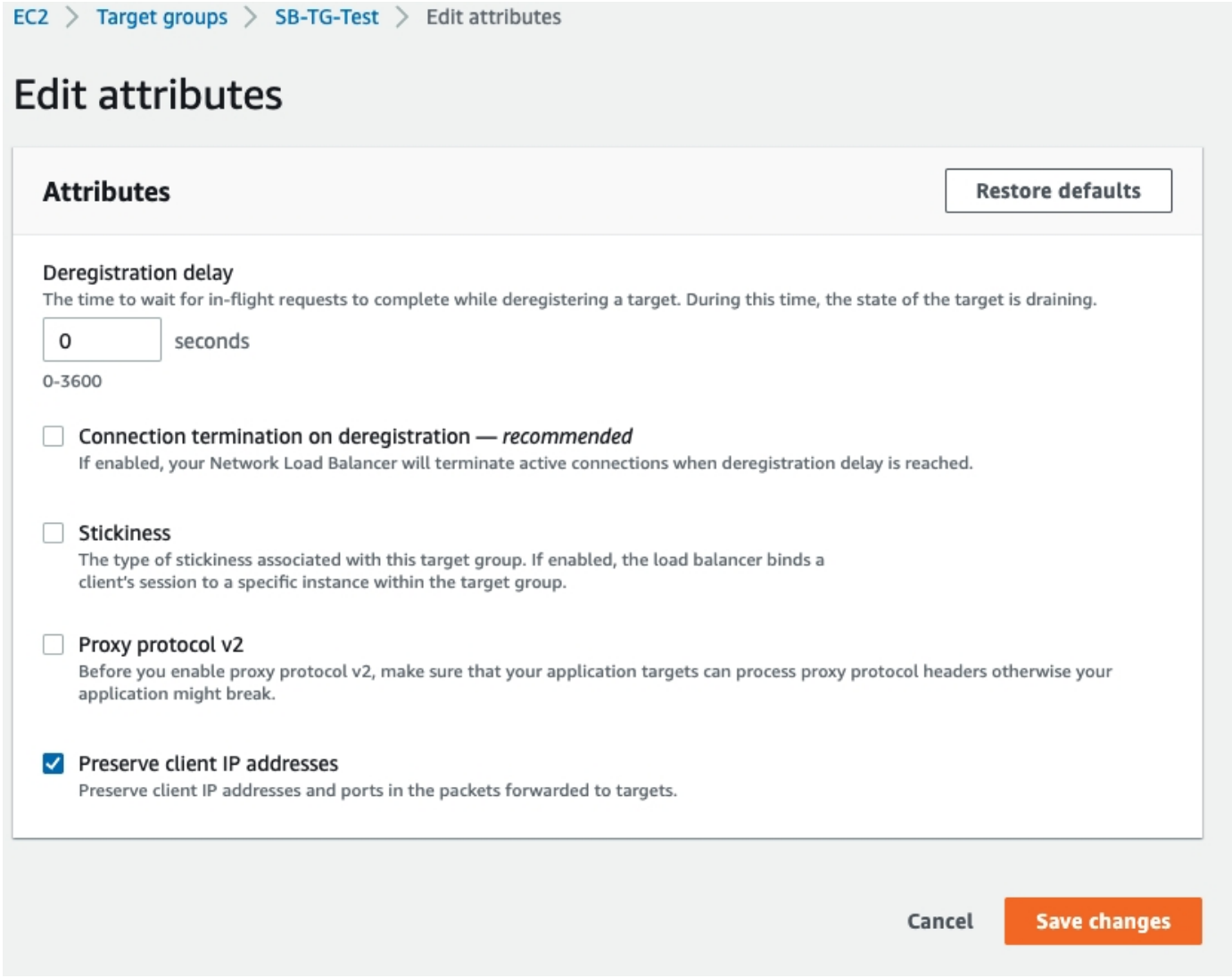



# Step 2: Configure Target Group attributes

Once target group created successfully, open AWS navigation pane and go to Target groups. Choose the Target group created in Step 1 above > **Actions > Edit attributes.**




Edit the **Deregistration delay** from default 300 to 0 and click **Save Changes**.



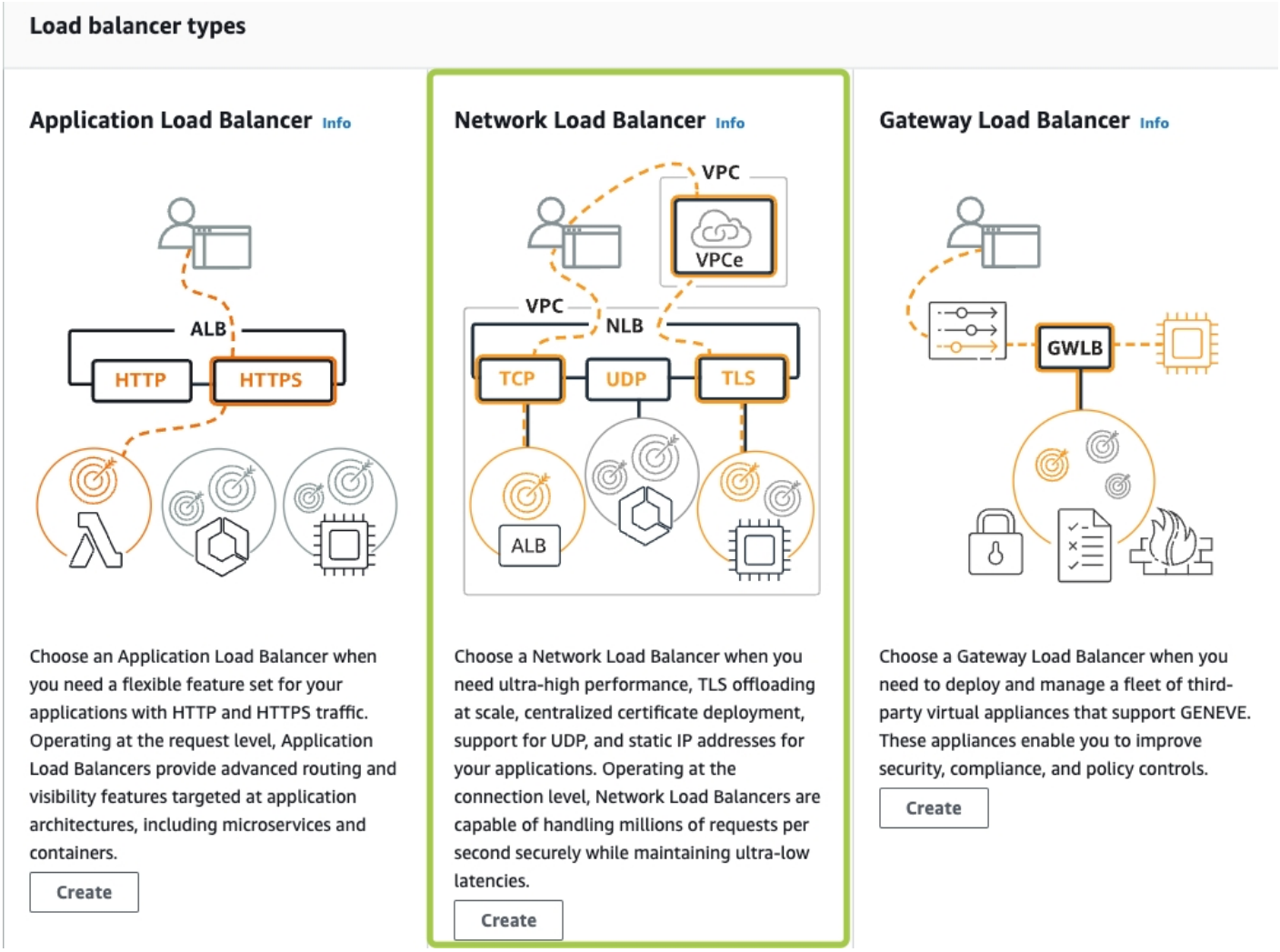

Please see here for more information regarding deregistration delay.

## Step 3: Choosing the Load Balancer Type

A load balancer serves as the single point of contact for clients. The load balancer distributes incoming traffic across multiple targets, such as Amazon EC2 instances acting as Parallels RAS Secure Client Gateways.

Elastic Load Balancing supports the following types of load balancers: Application Load Balancers, Network Load Balancers, Gateway Load Balancers and Classic Load Balancers (will be retired on August 15th, 2022).

For native Parallels Clients connectivity the **Network Load Balancer** can be used.




For more information on AWS Network Load Balancer please check this article

# Step 4: Configure Load Balancer and Listener

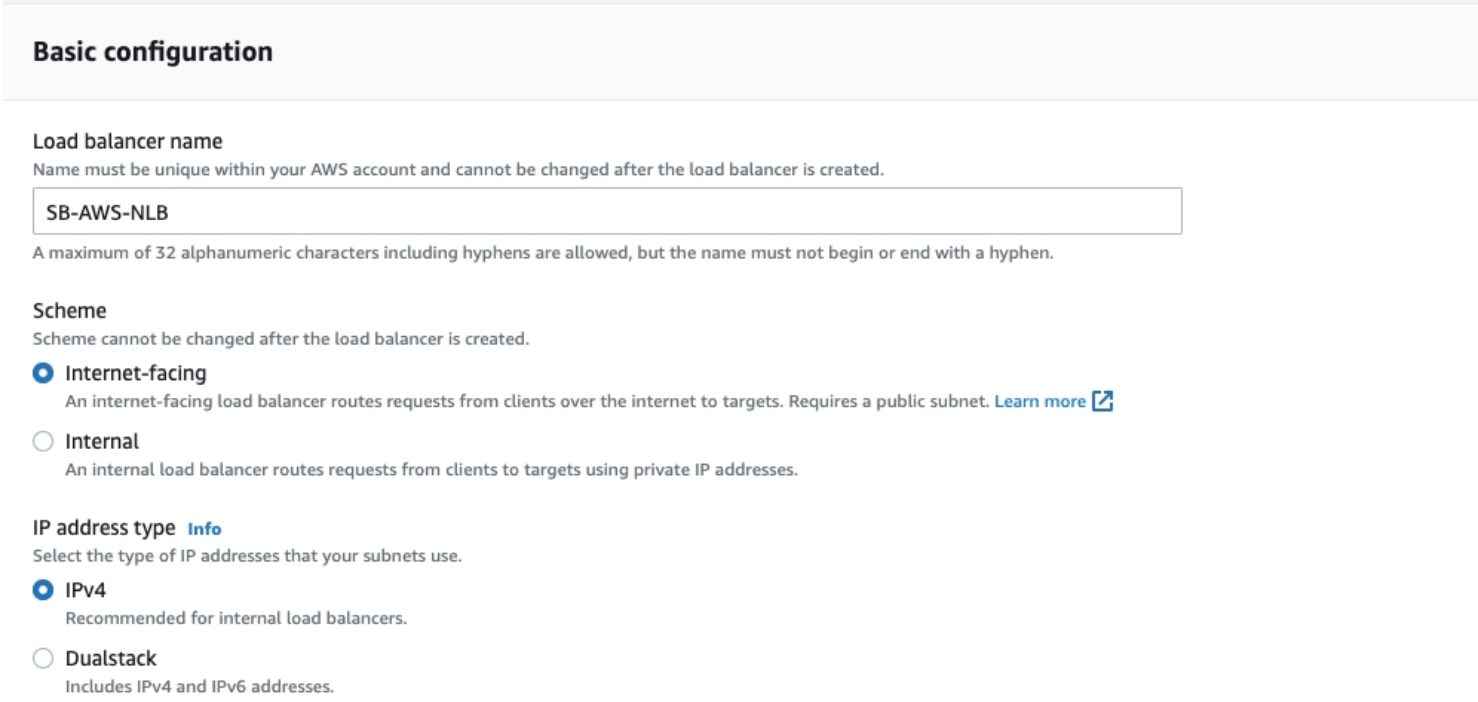
| | | |
|---|---|---|
| **1** | **Basic Configuration** | |
| | Name | *Provide a significant name for your load balancer* |
| | Scheme | Internet-facing |
| **2** | **Network Mappings** | |
| | Select the appropriate VPC and choose the availability zones where your instances reside in | |
| **3** | **Listeners and routing** | |
| | Protocol | TCP |
| | Port | 443 (or an alternate port in case AWS ALB used for HTML5 clients. See **Step 6** below) |
| | Default action | Select the target group created and registered in Step 1 |

1. For **Load balancer name**, enter a name for your load balancer. For example, **MY-AWS-NLB**

2. For **Scheme** and **IP address** type, keep the default values.
3. For **Network mappings**, select the **VPC** that you used for your EC2 instances. For each Availability Zone that you used to launch your EC2 instances, select the Availability Zone and then select one public subnet for that Availability Zone.

By default, AWS assigns an IPv4 address to each load balancer node from the subnet for its Availability Zone. Alternatively, when you create an internet-facing load balancer, you can select an Elastic IP address for each Availability Zone. This provides your load balancer with static IP addresses.

4. For Listeners and routing, keep the default, which is a listener that accepts TCP traffic on port 443.
5. For Default action, select the target group that you created and registered in step 1.
6. (Optional) Add a tag to categorize your load balancer. Tag keys must be unique for each load balancer.
7. Review your configuration, and choose Create load balancer. A few default attributes are applied to your load balancer during creation. You can view and edit them after creating the load balancer. For more information, see Load balancer attributes.



## Network mapping Info

The load balancer routes traffic to targets in the selected subnets, and in accordance with your IP address settings.

VPC

Select the virtual private cloud (VPC) for your targets. Only VPCs with an internet gateway are enabled for selection. The selected VPC cannot be changed after the load balancer is
confirm the VPC for your targets, view your target groups.

SB-VPC
vpc-071d6e1781f75d822
IPv4: 10.0.0.0/16

Mappings

Select at least two Availability Zones and one subnet per zone. The load balancer routes traffic to targets in these Availability Zones only. Availability Zones that are not supported
balancer or the VPC are not available for selection. Subnets cannot be removed after the load balancer is created, but additional subnets can be added.

- [x] us-east-1a

- [x] us-east-1e

Subnet

subnet-0ebd06f34a93f3331 MySubnet

**IPv4 settings**

IPv4 address

Assigned by AWS

## Listeners and routing Info

A listener is a process that checks for connection requests, using the protocol and port you configure. Traffic received by the listener is then routed per your specification.

Listener **TCP:443** Rem

Protocol: TCP

Port: 443

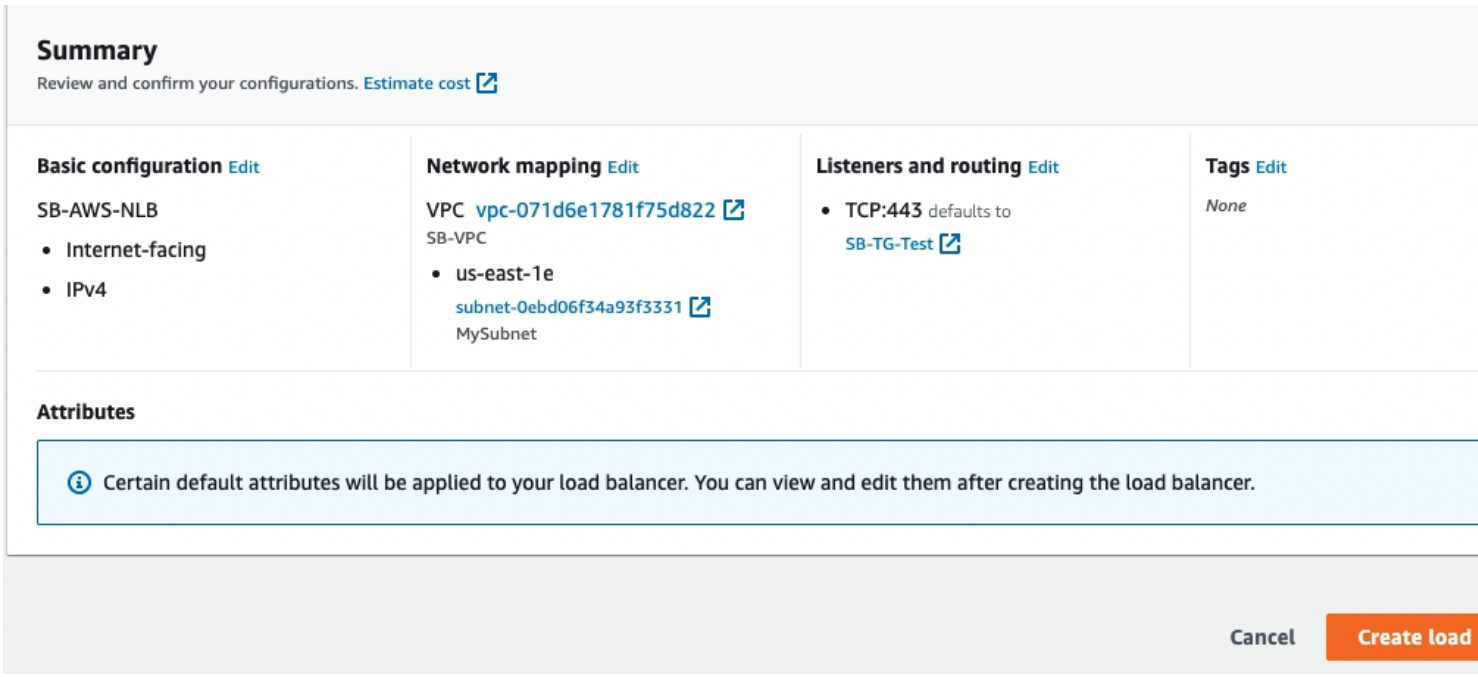
1-65535

Default action Info

Forward to: SB-TG-Test, TCP
Target type: Instance, IPv4

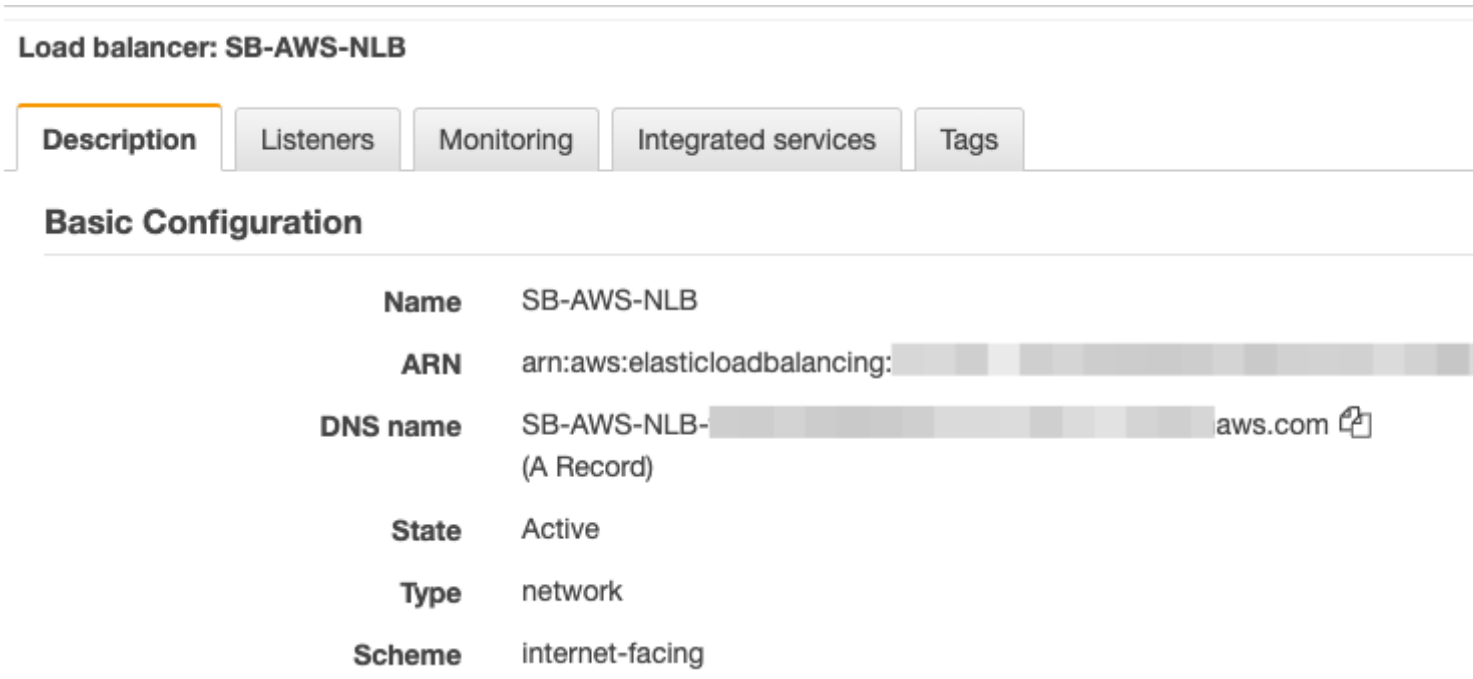
Create target group

Add listener

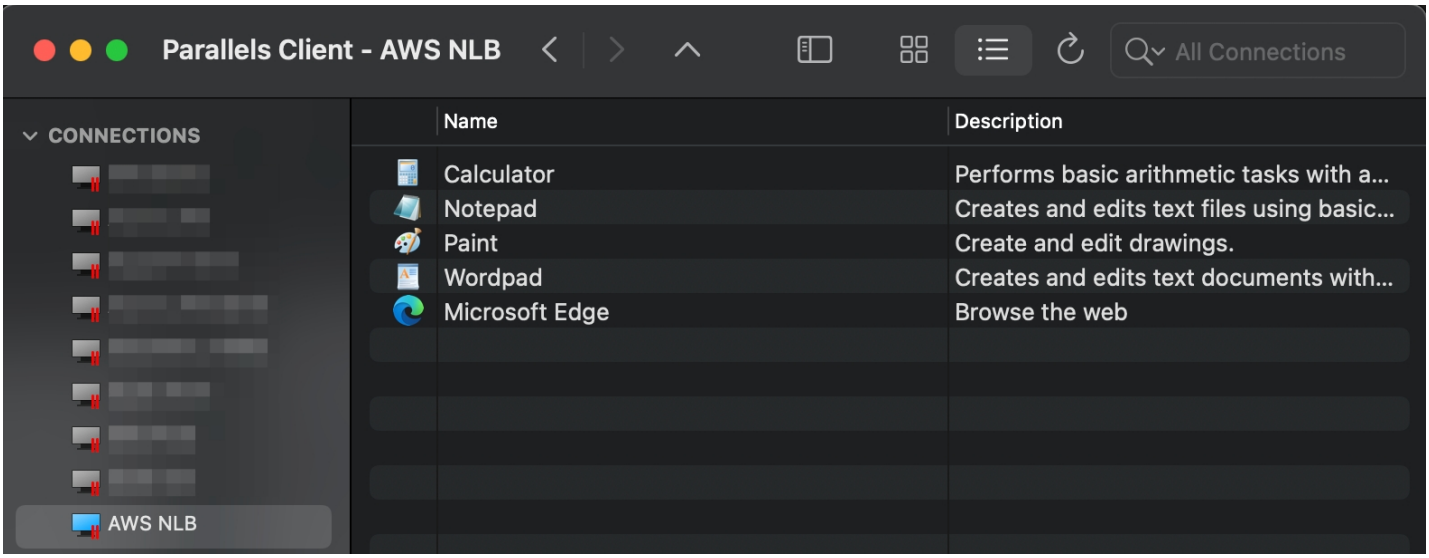



# Step 5: Test and Evaluate Load Balancing

You can test Load balancer configuration by taking note and copying the DNS name given to the Load balancer as shown from Load balancers – Description – Basic configuration




Log on from Parallels Client and confirm application launching:




# Step 6: Using "Network Load Balancer Access" feature

The aforementioned configuration enables support for native Parallels Clients, but connections over HTML5 client using a web browser will fail as TCP does not support stickiness. The Network Load Balancers access feature 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.

When one need to utilize both, native Parallels and HTML5 connections, in addition to AWS NLB, AWS ALB needs to be deployed.

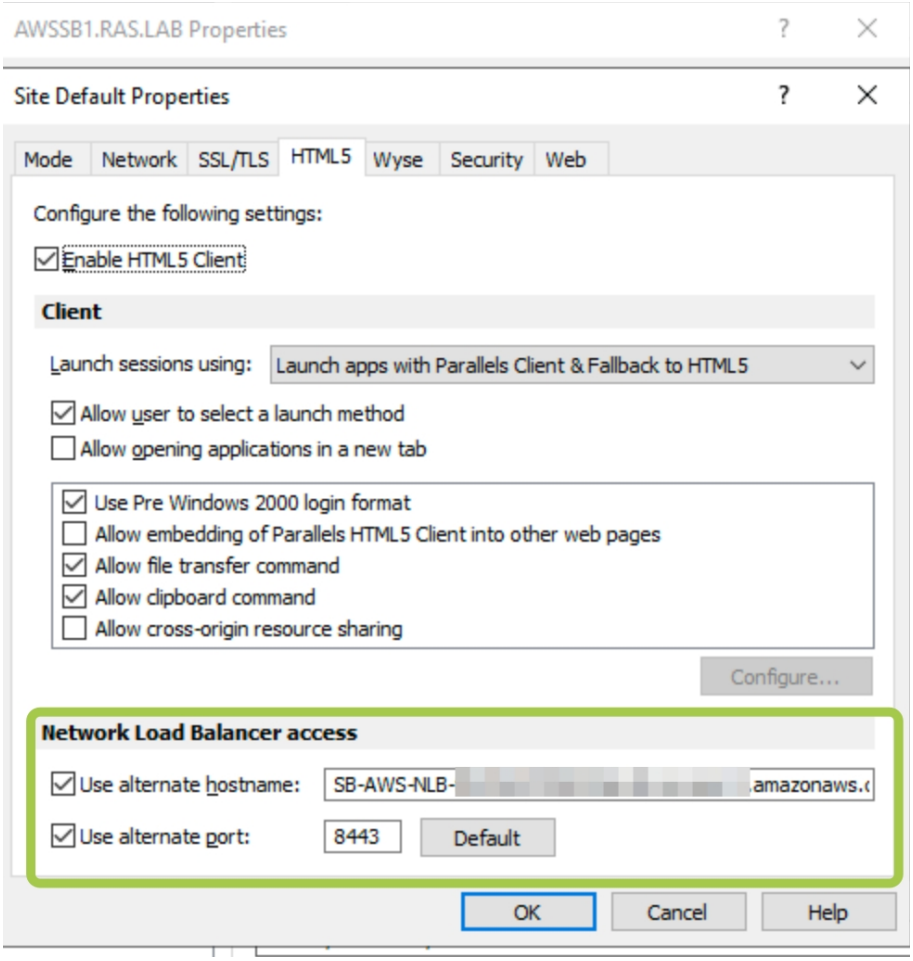
In this case, below please find the recommended configuration:

## Parallels RAS Console

1. In RAS Console **> Gateways >** right-click on the required Gateway Agent **> Properties > HTML5 tab** (one can also apply this configuration to all RAS gateways within the site by modifying the **Site defaults**):

2. In the **Network load balancer** section do the following:

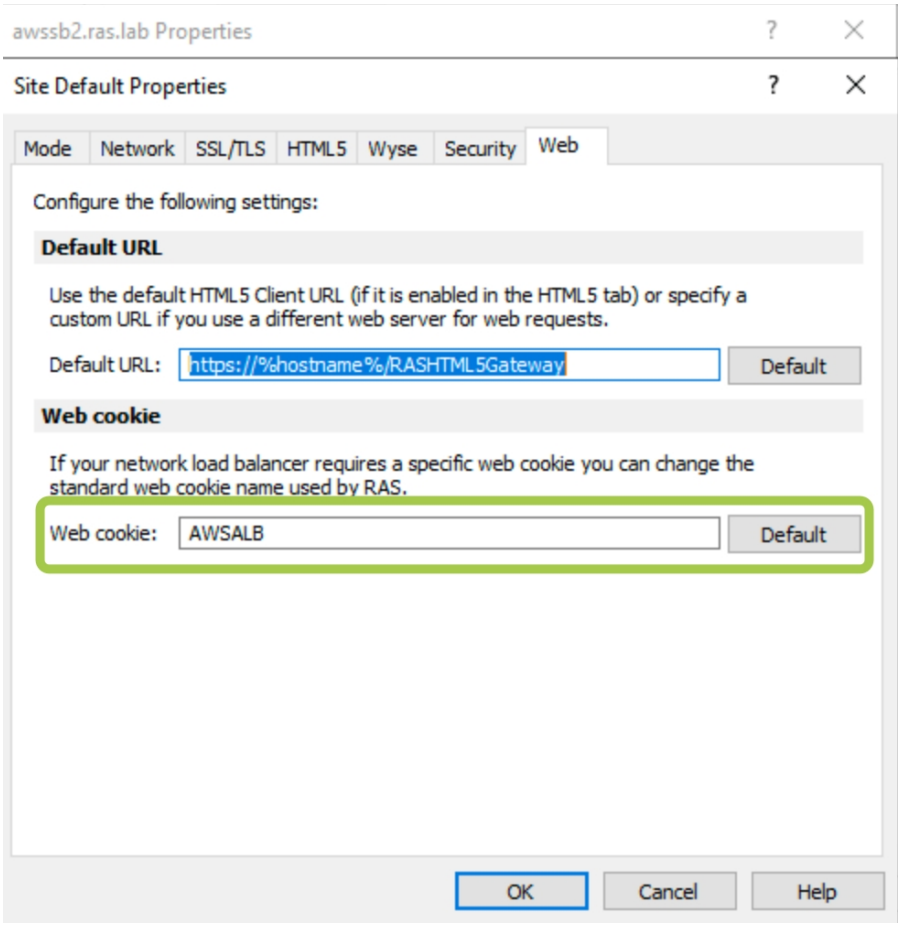
- Check **Use alternate port** and specify an alternate port number (in our example, port 8443). The port must not be used by any other component in the RAS Farm or Site. When the alternate port is enabled, all native Parallels Clients will use this port to connect to the RAS Farm or Site.
- Check **Use alternate hostname** and specify the hostname of your AWS NLB (When the alternate hostname is enabled, all platform-specific Parallels Clients will use this hostname to connect to the RAS Farm or Site)




Check this article for more information

***Note:*** *using an alternate host or port is not suitable in a multi-tenant environment as Tenant Broker RAS Secure Client Gateways are shared between Tenants, which would require different configurations.*

3. Switch to the **Web tab** and set the web cookie as **AWSALB**




***Note:*** *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. Please note that this functionality is available in Parallels RAS 17.1 or newer.*

## AWS Application Load Balancer

Configure AWS Application Load Balancer (ALB) as described here: https://docs.aws.amazon.com/elasticloadbalancing/latest/application/application-load-balancer-getting-started.html

For AWS ALB Listener we suggest the following configuration:

Protocol: **HTTPS**
Port: **443**

Add the required target group(s), specify the security settings and click **Save Changes**

EC2 > Load balancers > SB-AWS-ALB : Edit listener

# Edit listener

arn:aws:elasticloadbalancing:us-east-

## Listener details

A listener is a process that checks for connection requests, using the protocol and port you configure. Traffic received by the listener is then routed per your specification. You can specify multiple rules and multiple certificates per listener after the load balancer is created.

| Protocol | Port |
|---|---|
| HTTPS | 443 |
| | 1-65535 |

### Default actions Info

Specify the default actions for traffic on this listener. Default actions apply to traffic that does not meet the conditions of rules on your listener. Rules can be configured after the listener is created.

▼ 1. Forward to Info — Remove

| Target group | Weight (0-999) |
|---|---|
| SB-TG-1 HTTPS<br>Target type: Instance, IPv4 | 1 |
| Traffic distribution: | 100% |
| Select a target group | 0 |

Create target group

Enable group-level stickiness Info

If you enable stickiness for your target group, requests routed to it remain in the same group for the duration you specify.

## Secure listener settings Info

### Security policy

Your load balancer uses a Secure Socket Layer (SSL) negotiation configuration, known as a security policy, to negotiate SSL connections with clients.

ELBSecurityPolicy-2016-08

Compare security policies

### Default SSL certificate

The certificate used if a client connects without SNI protocol, or if there are no matching certificates. You can add more certificates after you create the load balancer.

From ACM — AWSSB1.RAS.LAB

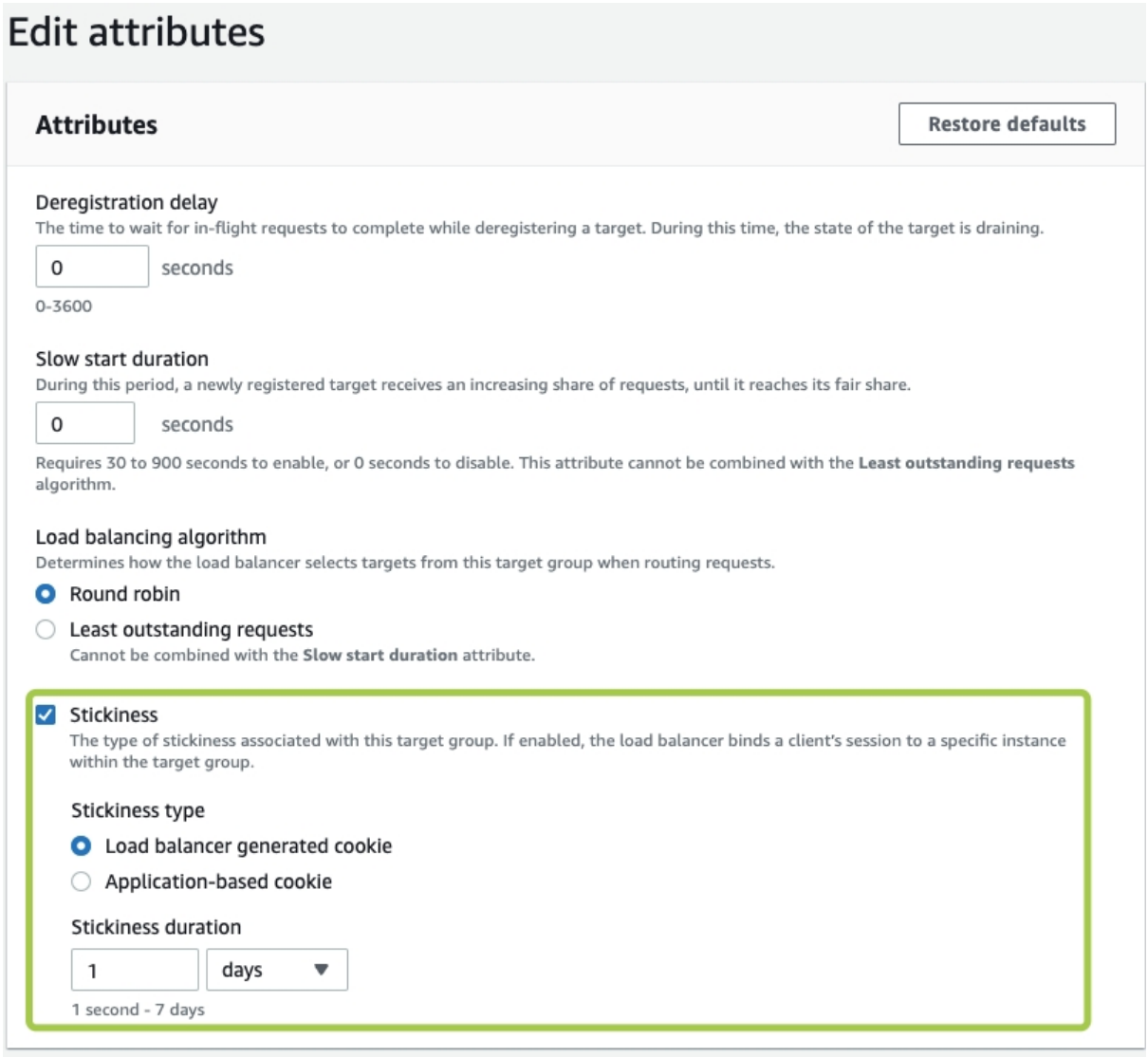
Request new ACM certificate

Cancel | Save changes

Target Group settings for AWS Application Load Balancer:

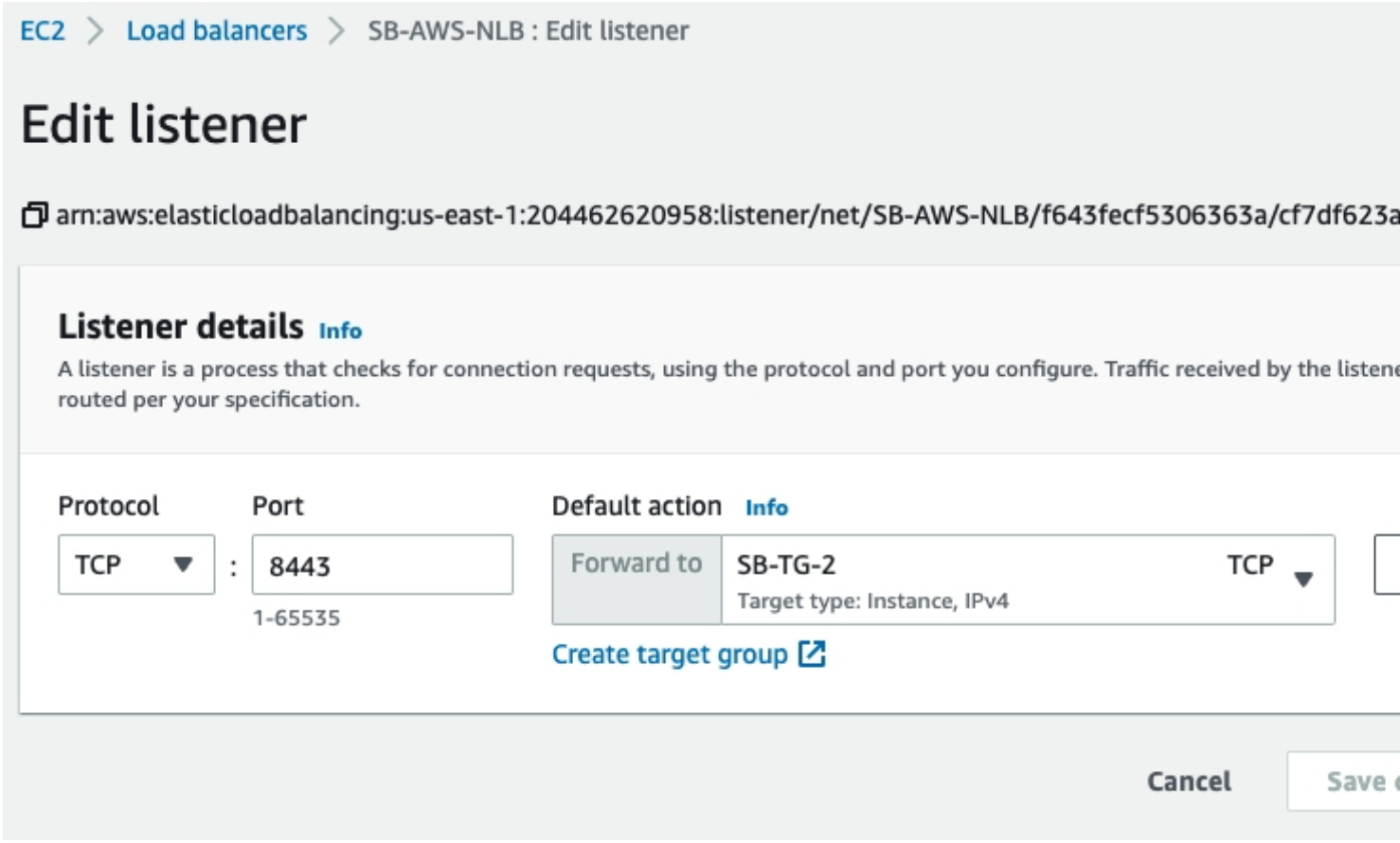
- Target type: **Instance**
- Protocol **HTTPS**
- Port **443**

Ensure you have enabled session stickiness for the target group associated with your AWS ALB:




## AWS Network Load Balancer

AWS NLB Listener port needs to be changed to an alternate one that we configured in Parallels RAS Console above (in our example, **8443**)

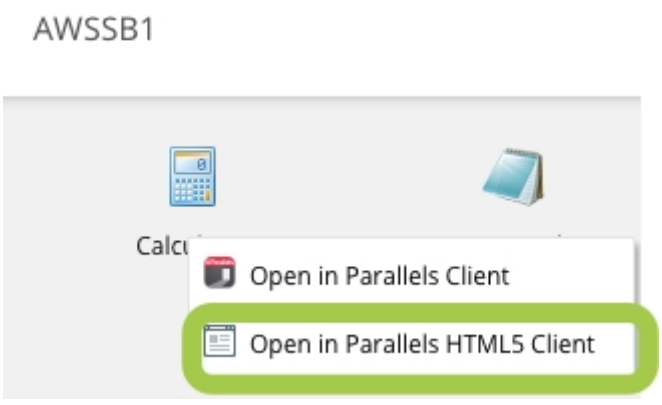


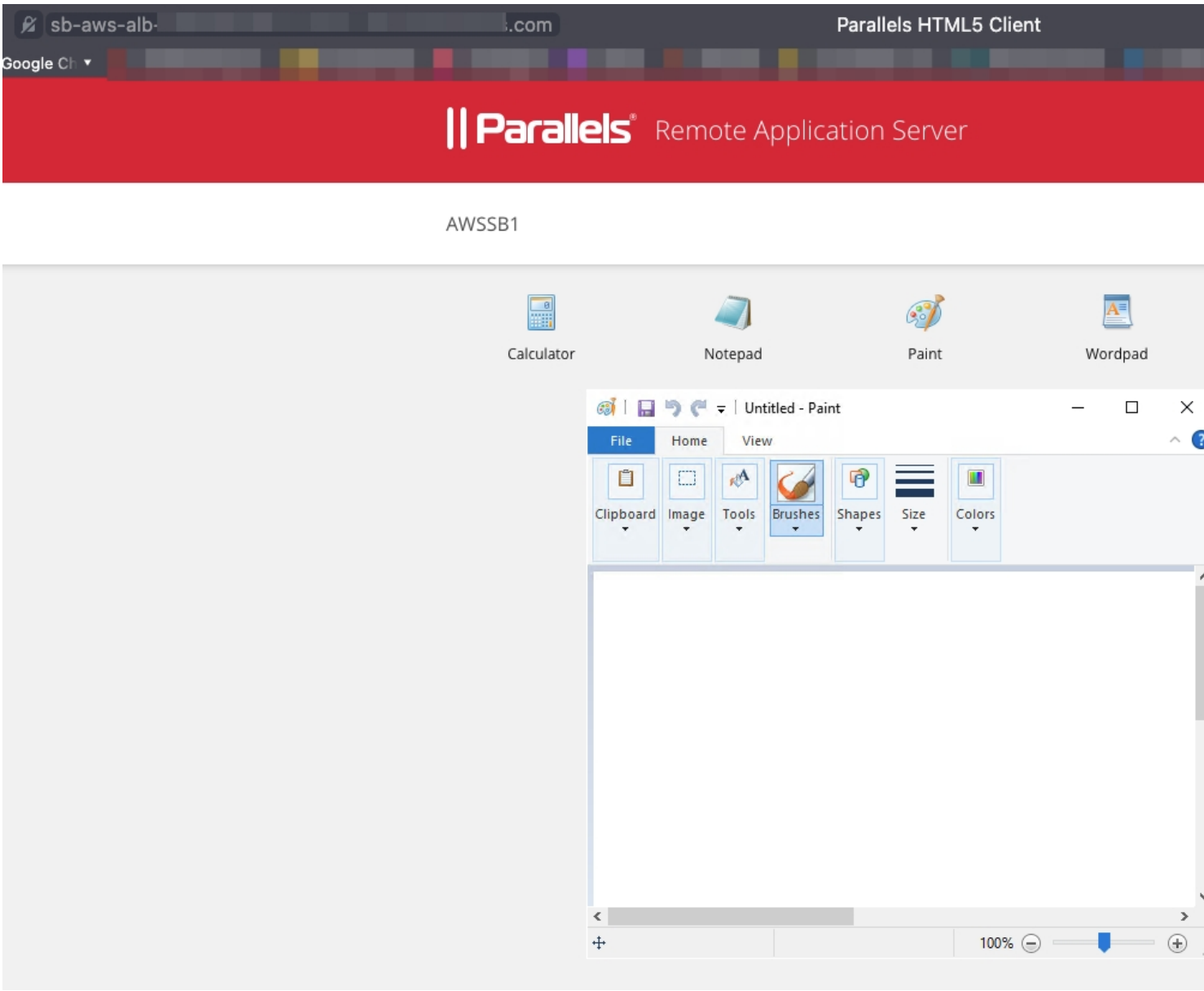

## Testing Load Balancing using Parallels HTML5 Client

Copy the DNS hostname of AWS ALB in AWS Console (e.g. SB-AWS-ALB-########.us-east-1.elb.amazonaws.com) and try accessing this using the web browser.

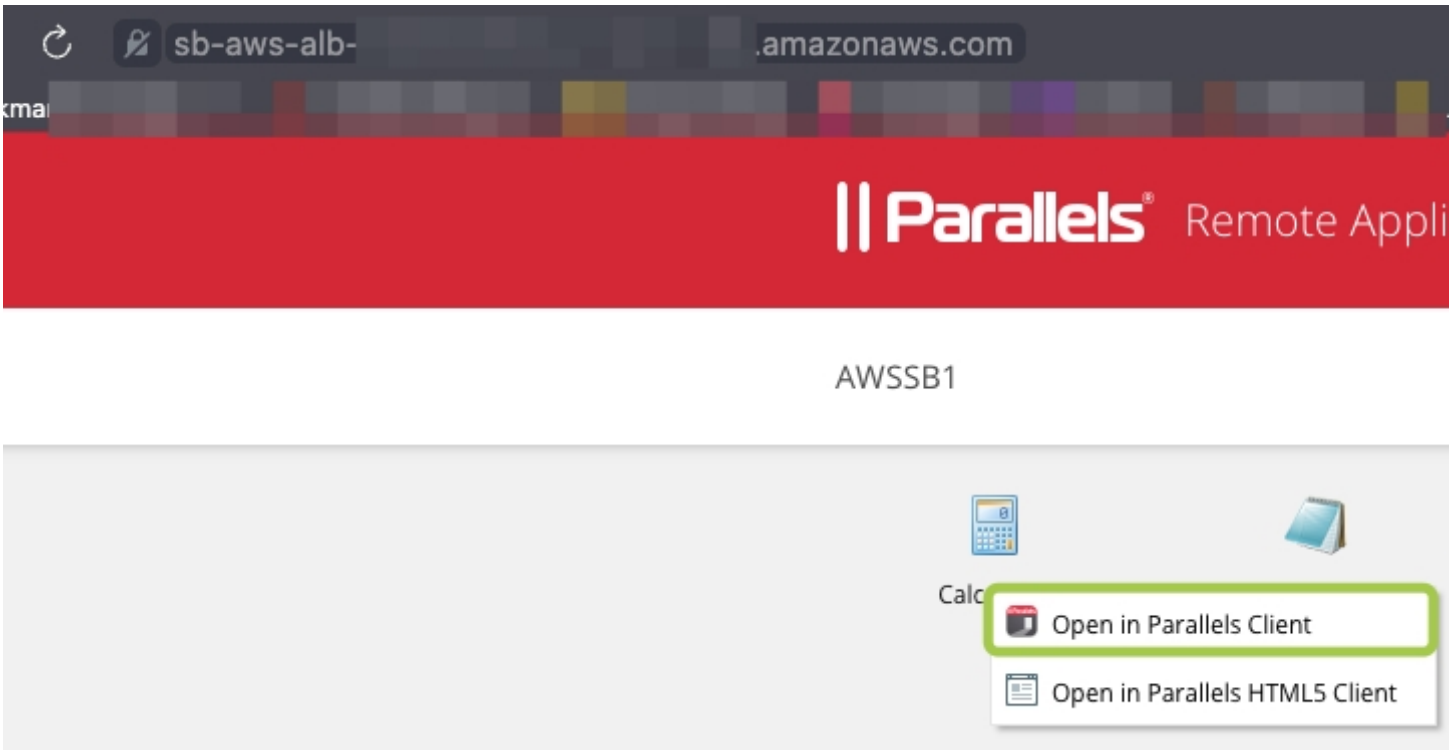
To test the work of AWS ALB (connection is being done over HTTPS):

Right-click on an app **> Open in Parallels HTML5 Client**

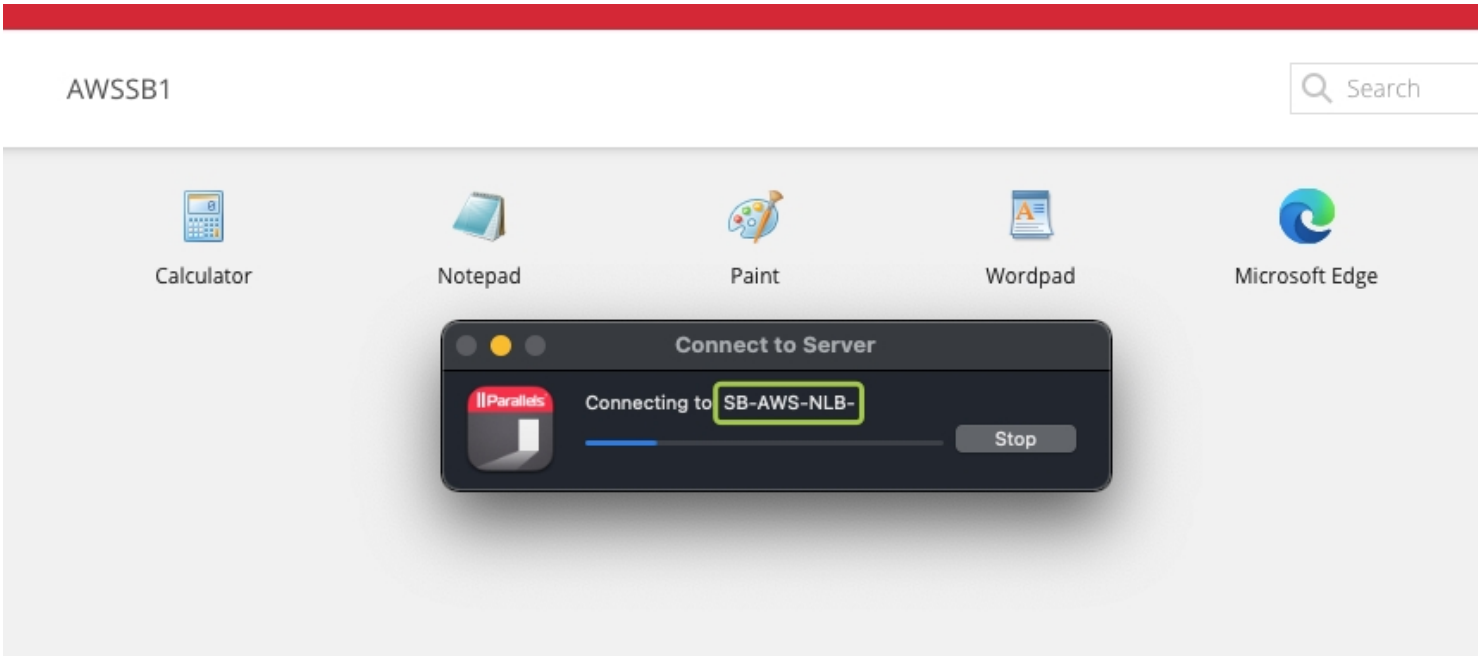




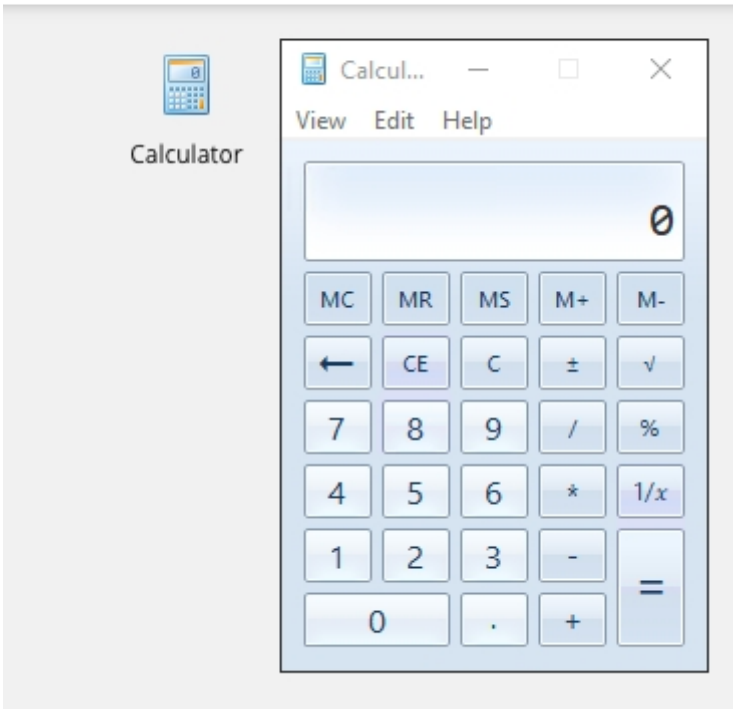

To check the load balancing over TCP, in the same Parallels HTML5 portal, right-click on an app **> Open in Parallels Client**




You will notice that the native Parallels Client is establishing connection using the alternate hostname:

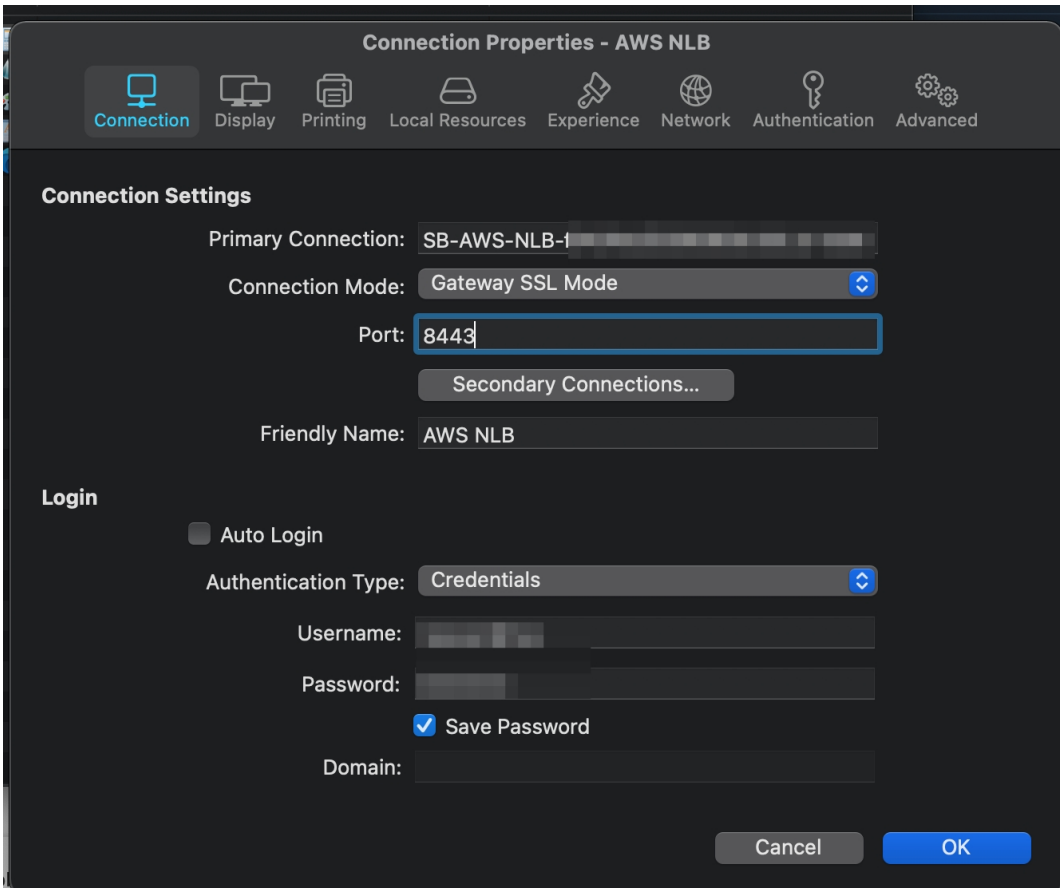


AWSSB1

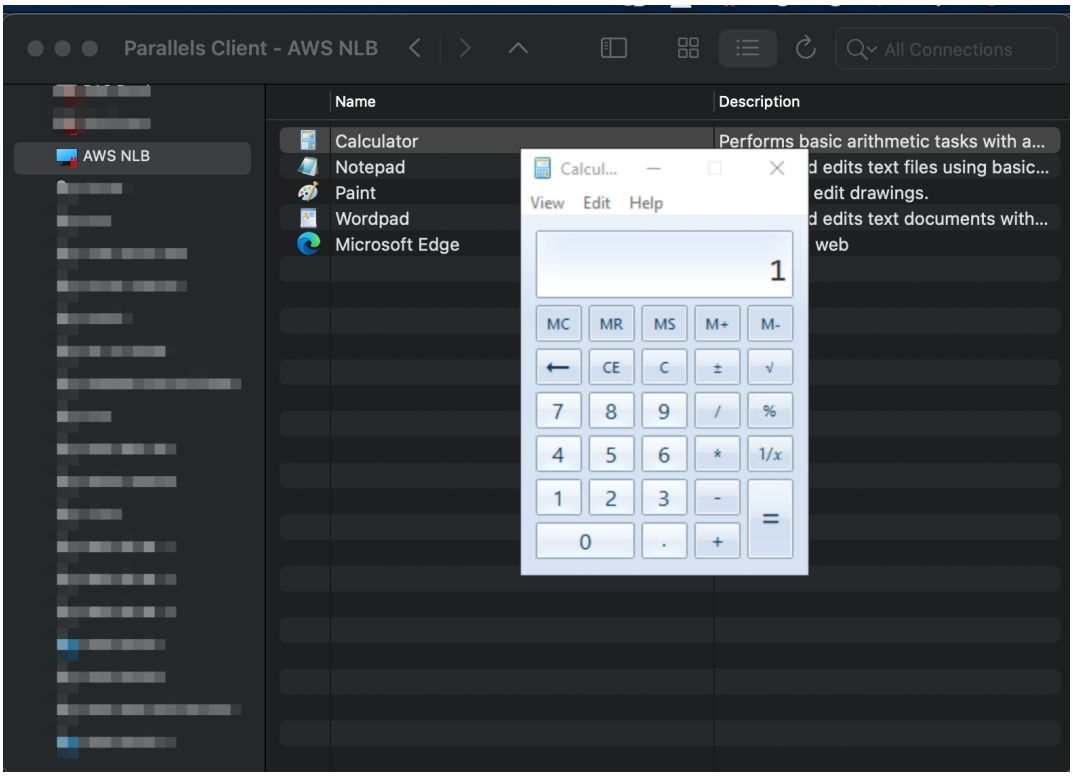



## Testing Load Balancing using Parallels Client

Since recent modification, at Parallels Client we need to specify the alternate port that was set in Parallels RAS Console and AWS NLB Listener.




Connect and launch a published app




# Links for reference

For more information regarding Parallels RAS please see here: https://www.parallels.com/products/ras/remote-application-server/

For more information regarding AWS ELB please see
here: https://docs.aws.amazon.com/elasticloadbalancing/index.html

---