



Redirect HTTP to HTTPS

Avi Technical Reference (v18.2)

Copyright © 2018

Redirect HTTP to HTTPS

[view online](#)

Overview

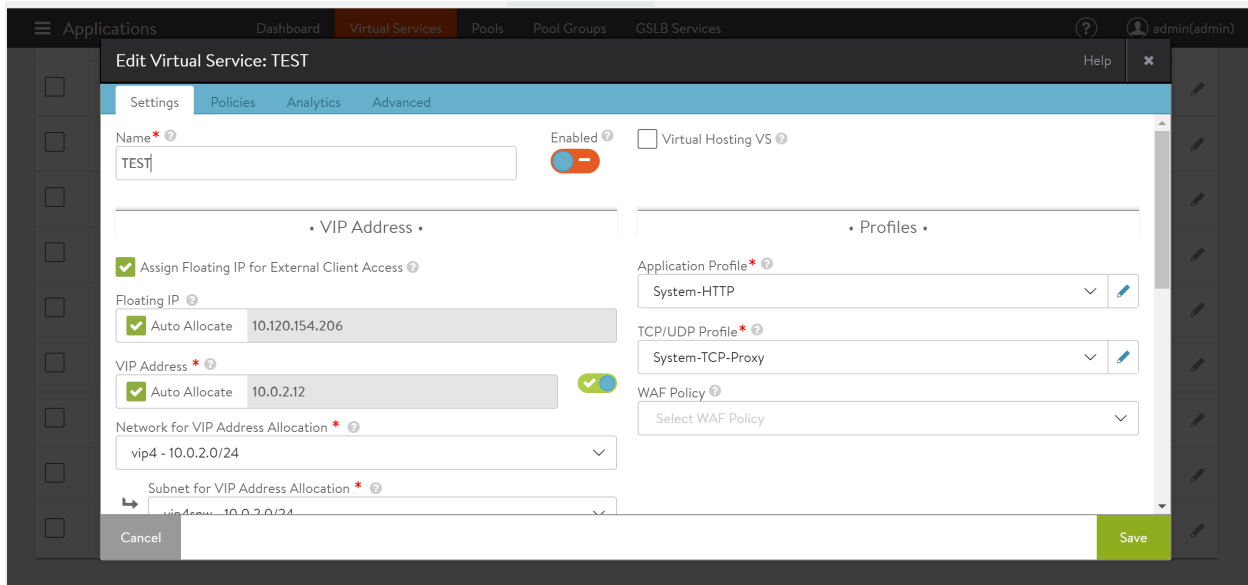
For security, an industry best practice is to ensure all HTTP traffic is SSL-encrypted as HTTPS. Since typical end-users do not specify the HTTPS protocol when entering URLs for requests, the initial requests arrive over HTTP. Because Avi Vantage can provide SSL termination services, it also must handle redirecting of HTTP users to HTTPS. You can enable HTTP-to-HTTPS redirect in any of the following ways. The methods are presented in order from simplest (with fewest options) to most advanced.

Configuration

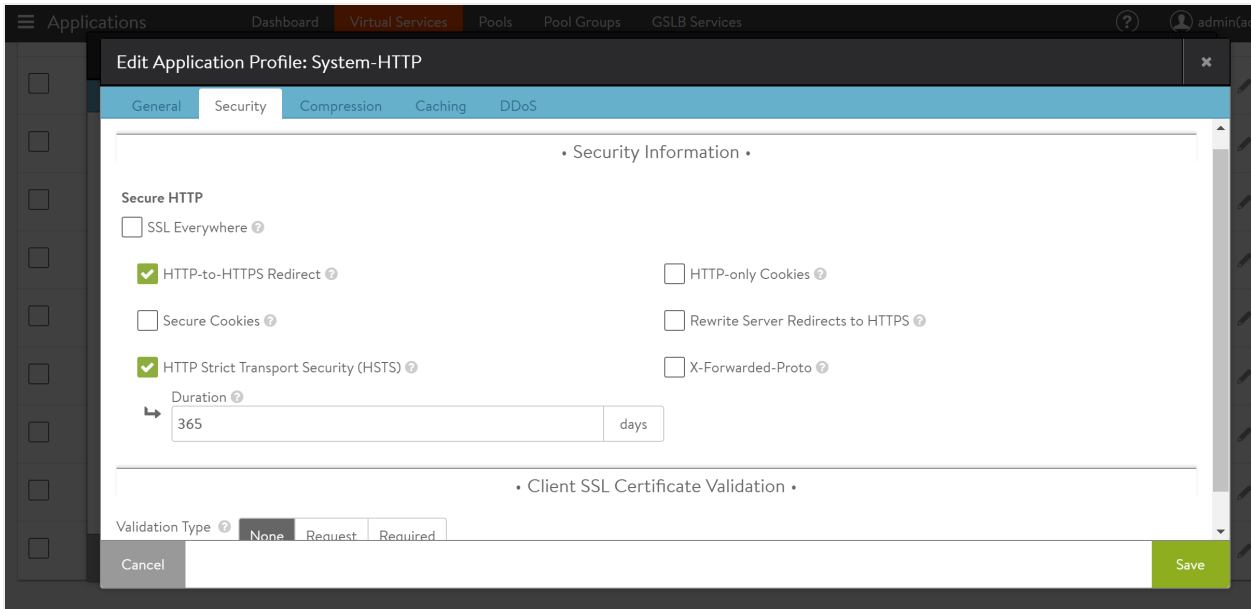
Using Application Profile

Option 1

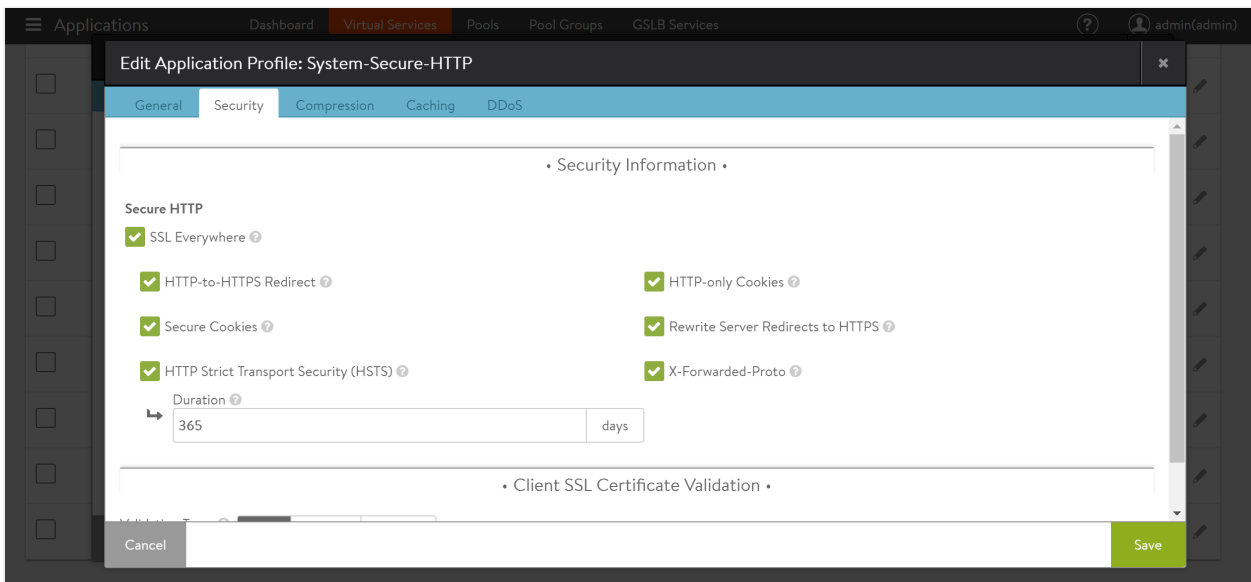
If the virtual service is configured for both HTTP (usually port 80) and HTTPS (usually SSL on port 443), enable HTTP-to-HTTPS redirect via the attached HTTP application profile. Navigate to Applications > Virtual Services, select the desired virtual service, click on the edit icon on the right side, and navigate to the Profiles section.



Select the edit option for the attached Application Profile (System HTTP profile), and navigate to the Security tab. In the SSL Everywhere section of this tab, select the HTTP to HTTPS Redirect checkbox.

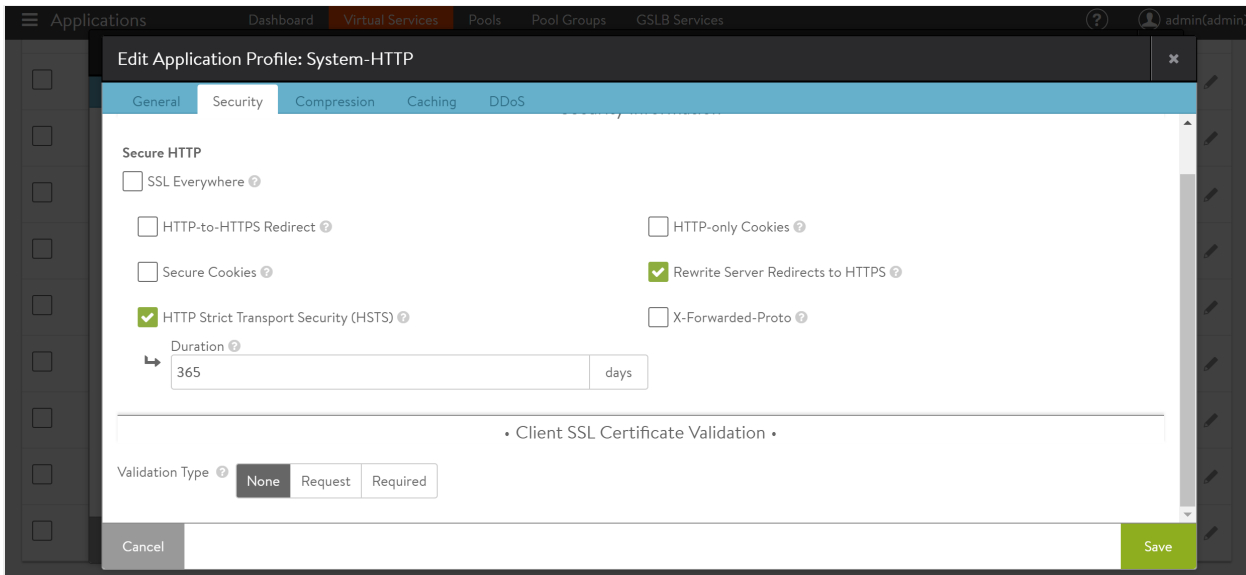


Avi Vantage also has the option for the System-Secure-HTTP profile in the drop-down list for the Application Profile. This profile is identical to the System-HTTP profile with the exception that the SSL Everywhere checkbox, which includes the HTTP to HTTPS Redirect option, is already enabled.



Option 2

Rewrite Server Redirects to HTTPS option is available within the Security tab of the Application Profile option. This option will change the Location header of redirects from HTTP to HTTPS, and will also remove any hardcoded ports. The following example shows a Location header sent from a server:



```
http://www.test.com:5000/index.htm
```

Avi Vantage will rewrite the Location header, sending the following to the client:

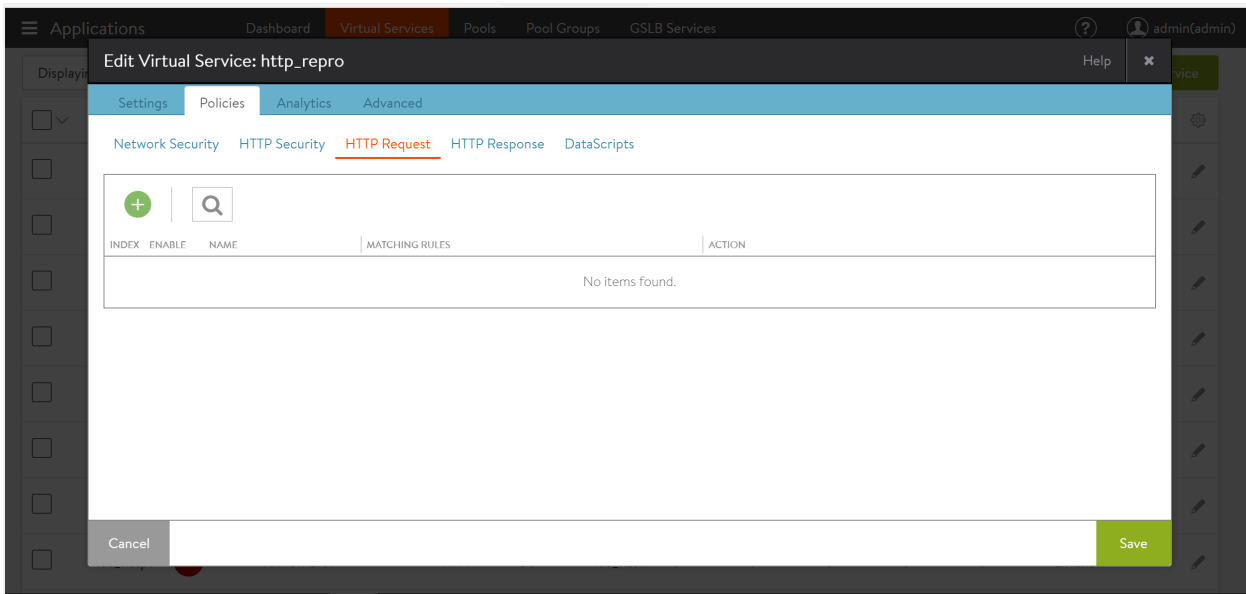
```
https://www.test.com/index.htm
```

Notes: * Relative redirects are not altered, only absolute. Therefore it is encouraged to have both checkboxes enabled.

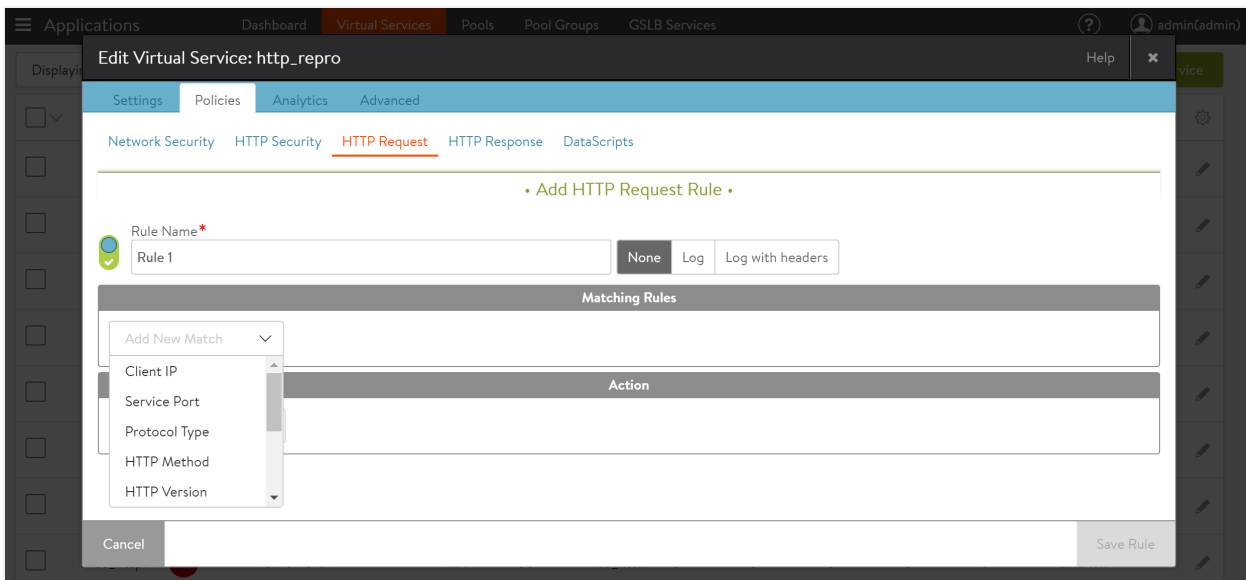
- This profile setting will have no impact for virtual services if the VS does not have HTTPS configured.

Using HTTP Request Policy

For more granularity, use an HTTP Request Policy. Navigate to Applications > Virtual Services, and select the edit option. Navigate to the Policies option, select the HTTP Request tab and click on the Create option (the plus icon).



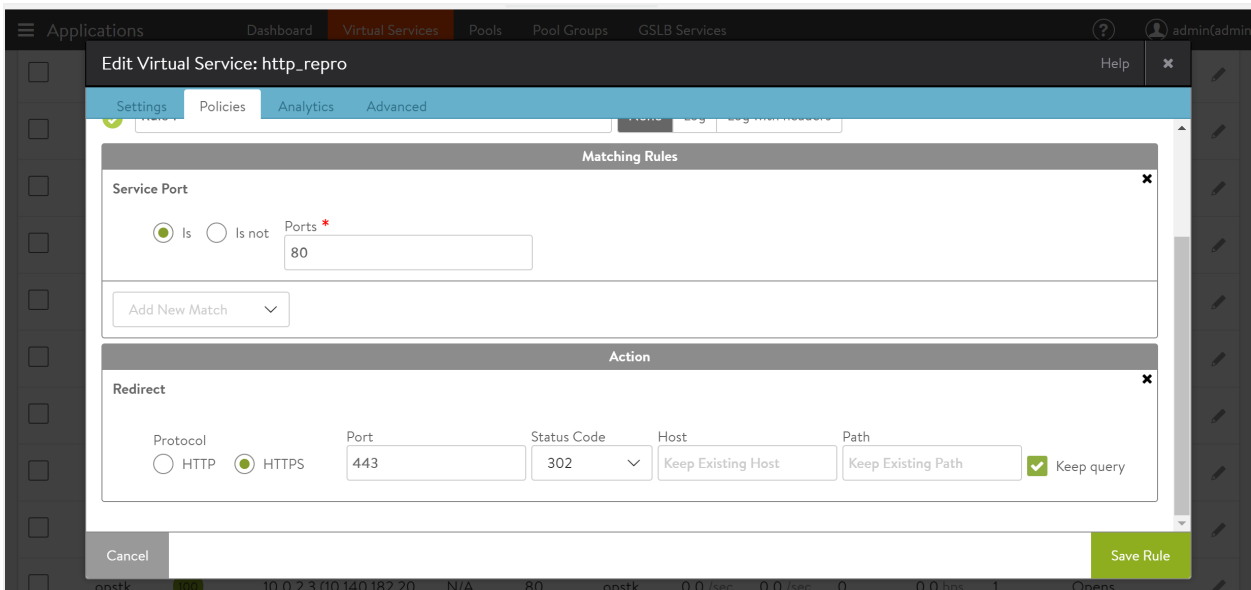
Provide the desired name to the new rule, select Service Port from the drop-down option for Matching Rules, and provide 80 as the value for Ports option.



Optionally, the required criteria can be added to determine when to perform the redirect.

Note: When redirecting to the same virtual service, you must specify a match criteria to prevent a redirect loop.

In the Action section, select Redirect from the drop-down menu. Then set the protocol to HTTPS. This will set the redirect port to 443 and the redirect response code to 302 (temporary redirect).



HTTP Request Policies are quick and easy to set up, and impact only a single virtual service at a time.

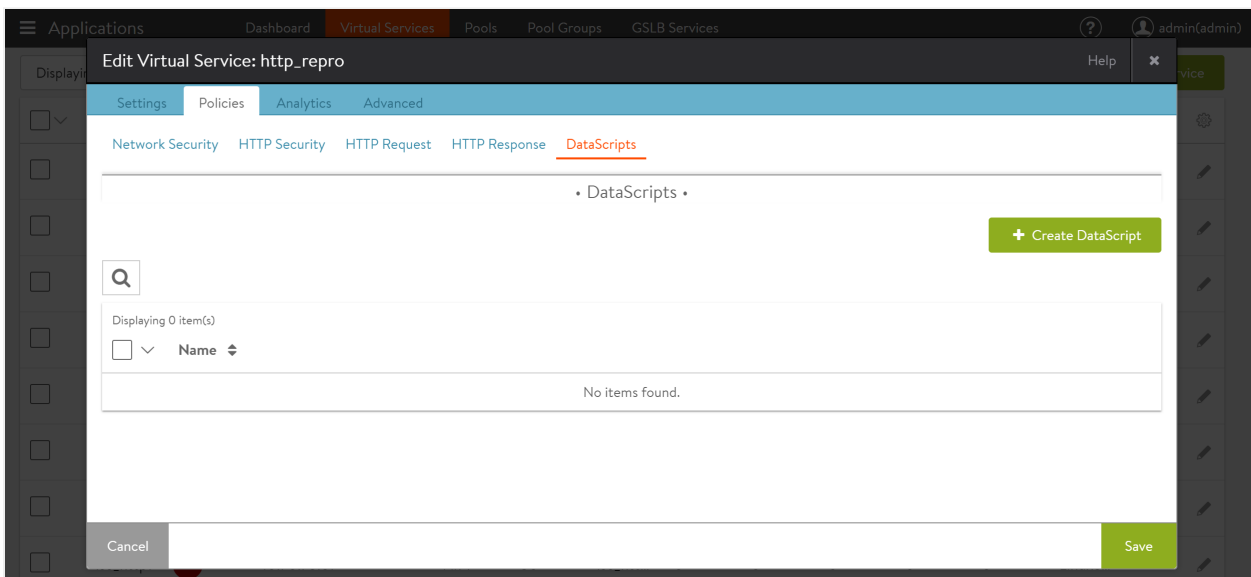
For more information on the usage of HTTP request policy, refer to [HTTP Request Policy](#)

Using DataScript

For maximum granularity and reusability, use a DataScript to specify the redirect behavior. While using DataScript may be overkill for basic requirements, it's always good to have for complex or granular requirements.

Navigate to Applications > Virtual Service, select the desired virtual service, and click on the edit option.

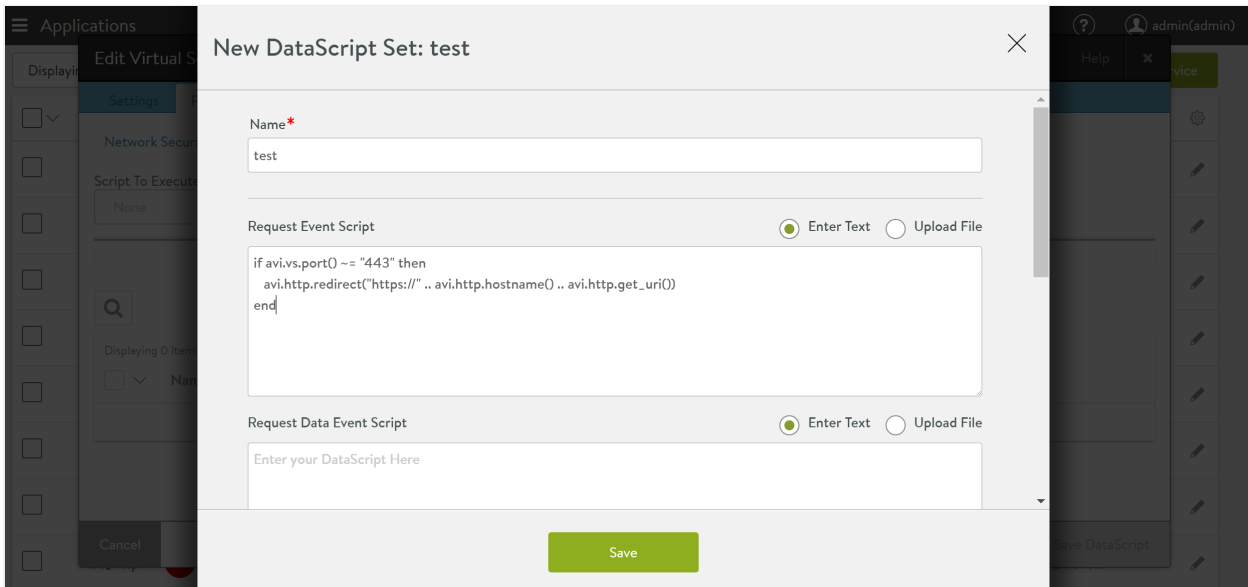
Select the Policies tab, and navigate to the DataScript tab, and click on the Create DataScript option to create a new DataScript policy.



Provide a name for the script, then paste the following text into the Request Event Script box and save:

```
if avi.vs.port() ~= "443" then
  avi.http.redirect("https://" .. avi.http.hostname() .. avi.http.get_uri())
end
```

Below is the screenshot from the Avi UI for reference:



For more information on using DataScript for redirecting HTTP to HTTPS, refer to [DataScript for HTTP Redirect](#)