Spike Arrest Policy:

The Spike Arrest policy protects against traffic spikes. It throttles the number of requests processed by an API proxy and sent to a backend, protecting against performance logs and downtimes.

where:

you can attach this policy anywhere in the flow, we recommend that you attach it in the following location so that it can provide spike protection at the immediate entry point of your API proxy i.e PreFlow.

Following are elements and attributes you can configure on this policy.

```
<SpikeArrest async="false" continueOnError="false" enabled="true" name="Spike-Arrest-1">
   <DisplayName>Custom label used in UI</DisplayName>
   <Rate>30ps</Rate>
   <Identifier ref="request.header.some-header-name"/>
   <MessageWeight  ref="request.header.weight"/>
</SpikeArrest>
```

<Rate> element

Specifies the rate at which to limit traffic spikes (or bursts). Specify a number of requests that are allowed in per minute or per second intervals.

Ex:
<Rate>10ps</Rate>
<Rate>30pm</Rate>

{int}ps (number of requests per second, smoothed into intervals of milliseconds)
{int}pm (number of requests per minute, smoothed into intervals of seconds)

How spike arrest works

- Generally protect against traffic spikes rather than as a way to limit traffic to a specific number of requests.
- The runtime Spike Arrest behavior differs from what you might expect to see from the literal per-minute or per-second values you enter.

Per-minute rates get smoothed into full requests allowed in intervals of **seconds**.

For example, 30pm gets smoothed like this:

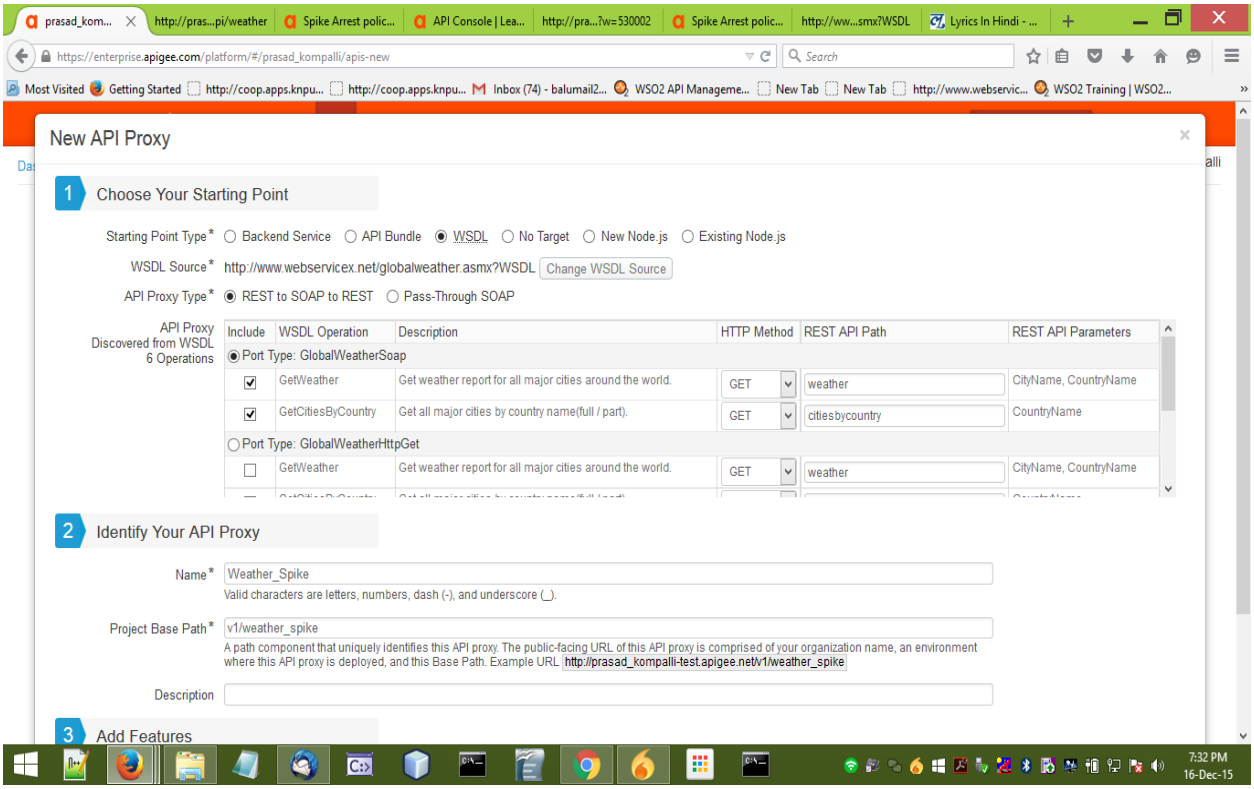
60 seconds (1 minute) / 30pm = 2-second intervals, or 1 request allowed every 2 seconds. A second request inside of 2 seconds will fail. Also, a 31st request within a minute will fail.

Per-second rates get smoothed into full requests allowed in intervals of **milliseconds**.

For example, 10ps gets smoothed like this:
1000 milliseconds (1 second) / 10ps = 100-millisecond intervals, or 1 request allowed every 100 milliseconds. A second request inside of 100ms will fail. Also, an 11th request within a second will fail.

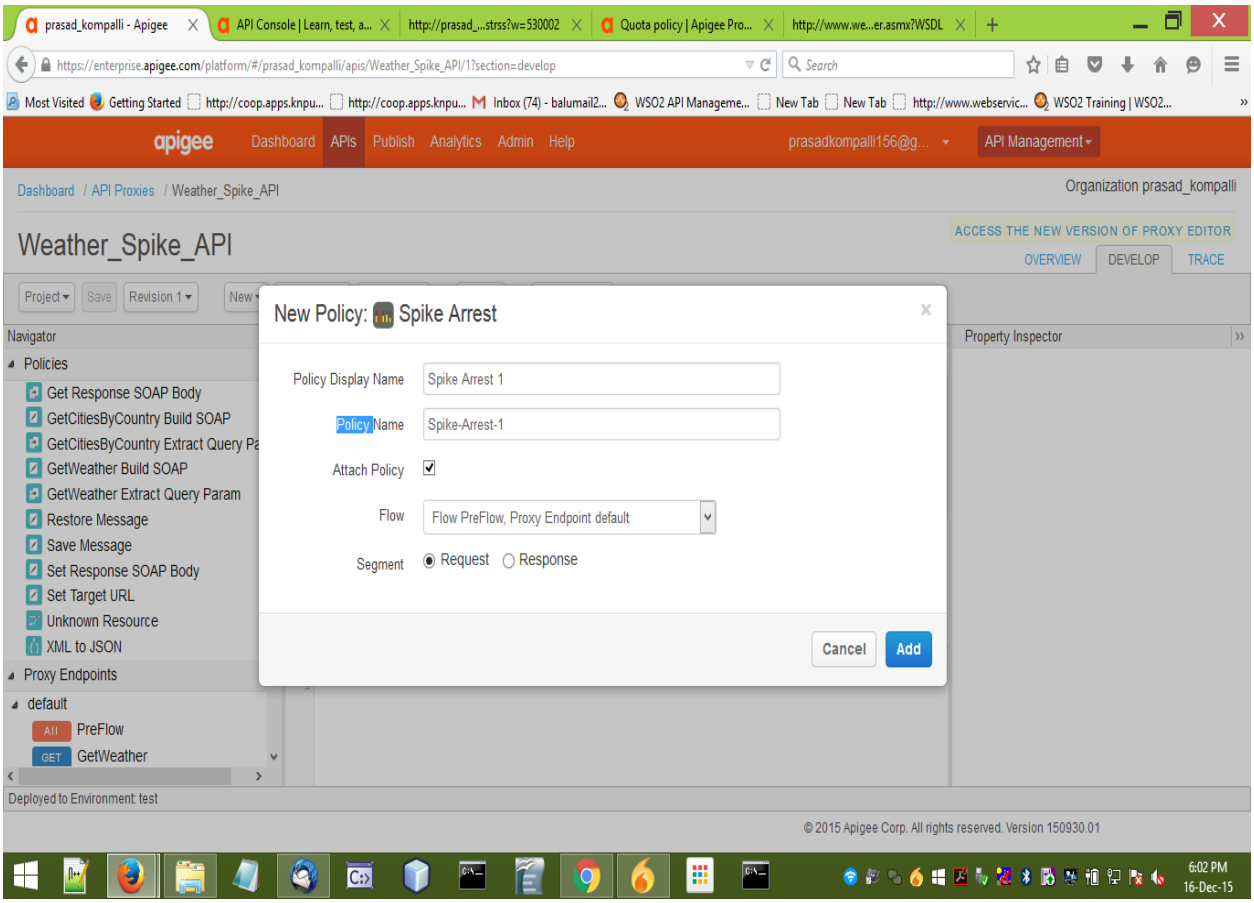
Step 1:

Initially, create an API Proxy by providing WSDL and Project Base path.



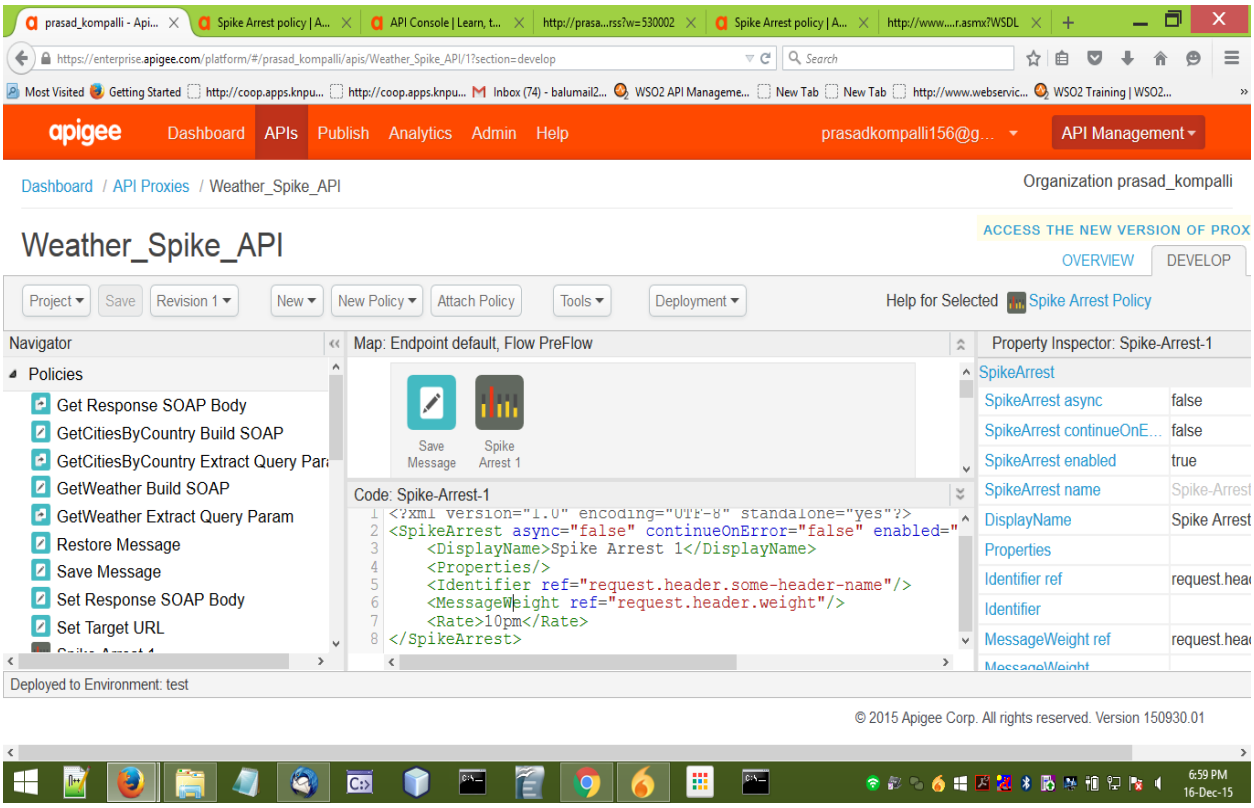
Step 2 :

In order to protects against traffic spikes and user based on the hits, select Spike Arrest Policy in New Policy tab.



Step 3:

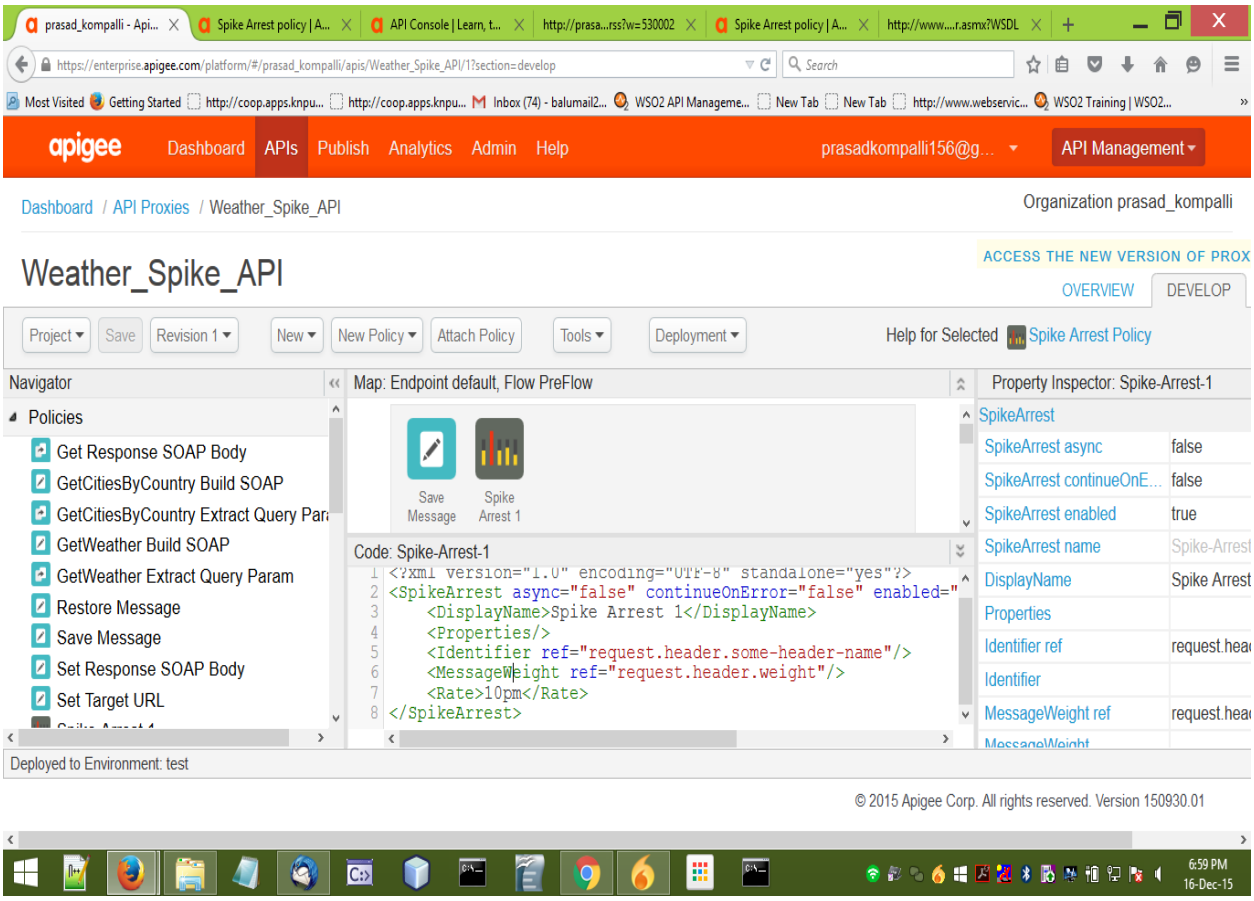
Here select the *Spike Arrest Policy* and edit the code and specify the *Rate* based on our requirement.



Step 4 :

Change the *Rate* based on our requirement.

In this example, it is changed to 10 requests per minute. i.e It spikes 1 request per 6 seconds.



Step 5:

In overview tab hit the backend URL for viewing the response.

Step 6:

After hitting the URL for 10 times, it will show an error as follows. And within 6 seconds it avoids second request.

After completion of 1 minute the data again show and follow the spike arrest policy.