

# Introduction

This workshop takes you through the steps to get the 3scale configured Nginx Gateway running on Openshift. It largely follows the steps on this [How-To](https://support.3scale.net/docs/deployment-options/apicast-openshift) on the 3scale Support site.

# Pre-Requisites

You should have completed the first workshop in the series *3scaleWorkshop-BasicConfig.docx*.

# Goals

Integrate this simple ***unmanaged*** API Endpoint into 3scale and deploy the previously configured Nginx gateway for this API, on Openshift.  
<https://tc-apis.herokuapp.com/flights/intl/flights>   
(\*\*Note, this is a test API on Heroku. The first time it is hit every hour is slow)

# Steps

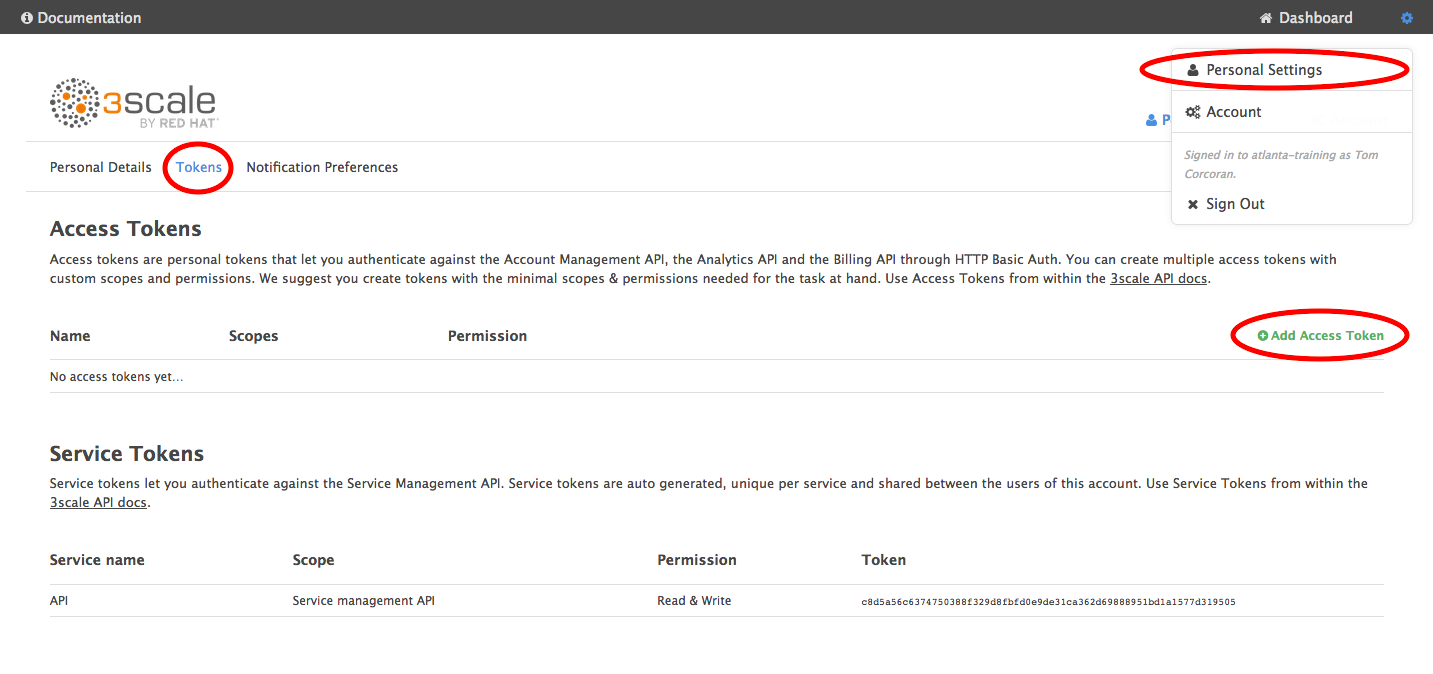
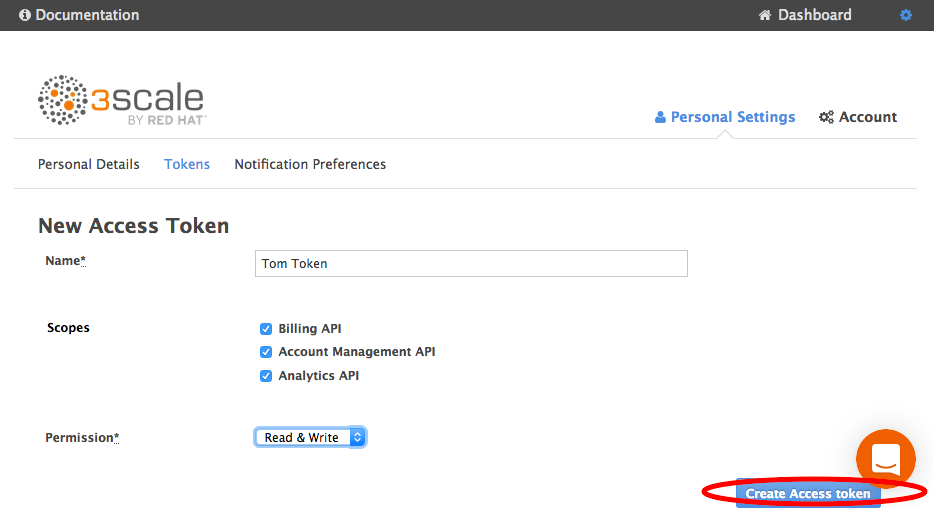
Here the URLs assume you have installed Openshift on AWS. Correct as appropriate.

1. Setup Openshift on a RHEL or other appropriate Linux machine. You will probably have this already. If not, see Appendix 1 for the steps to do a very basic Openshift installation on AWS.
2. Install Client tools if not already on your local machine. See Appendix 1 if not.
3. Start Docker on your RHEL box:   
   sudo systemctl start docker
4. Fire up Openshift:  
   sudo /home/ec2-user/oc-cli-tools/oc cluster up --public-hostname***=<your RHEL host>*** --routing-suffix=***<RHEL box IP>***.xip.io
5. Login to Openshift web console (credentials ***developer/developer*** work by default)  
   https://***<your RHEL host>***.compute-1.amazonaws.com:8443
6. On your local machine or on the machine that runs openshift (ssh in) :
   * oc login https://***<your RHEL host>***.compute-1.amazonaws.com:8443

Note: login as developer/developer???

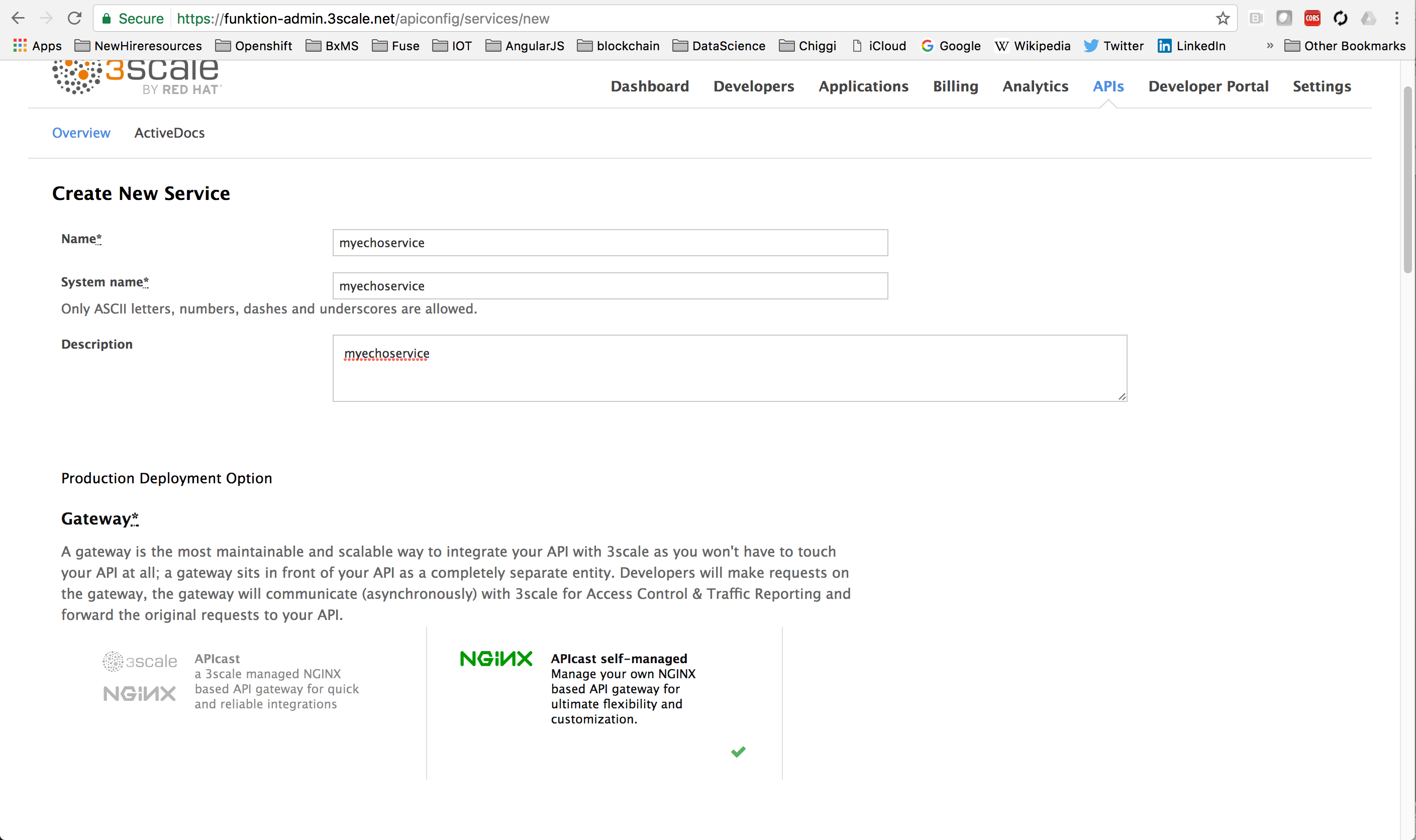
* + oc new-project "3scalegateway" --display-name="gateway" --description="3scale gateway demo"

1. On 3scale:
   * get an Access Token. Follow these: Gear Icon -> Personal Settings -> Tokens -> Add Access Token.

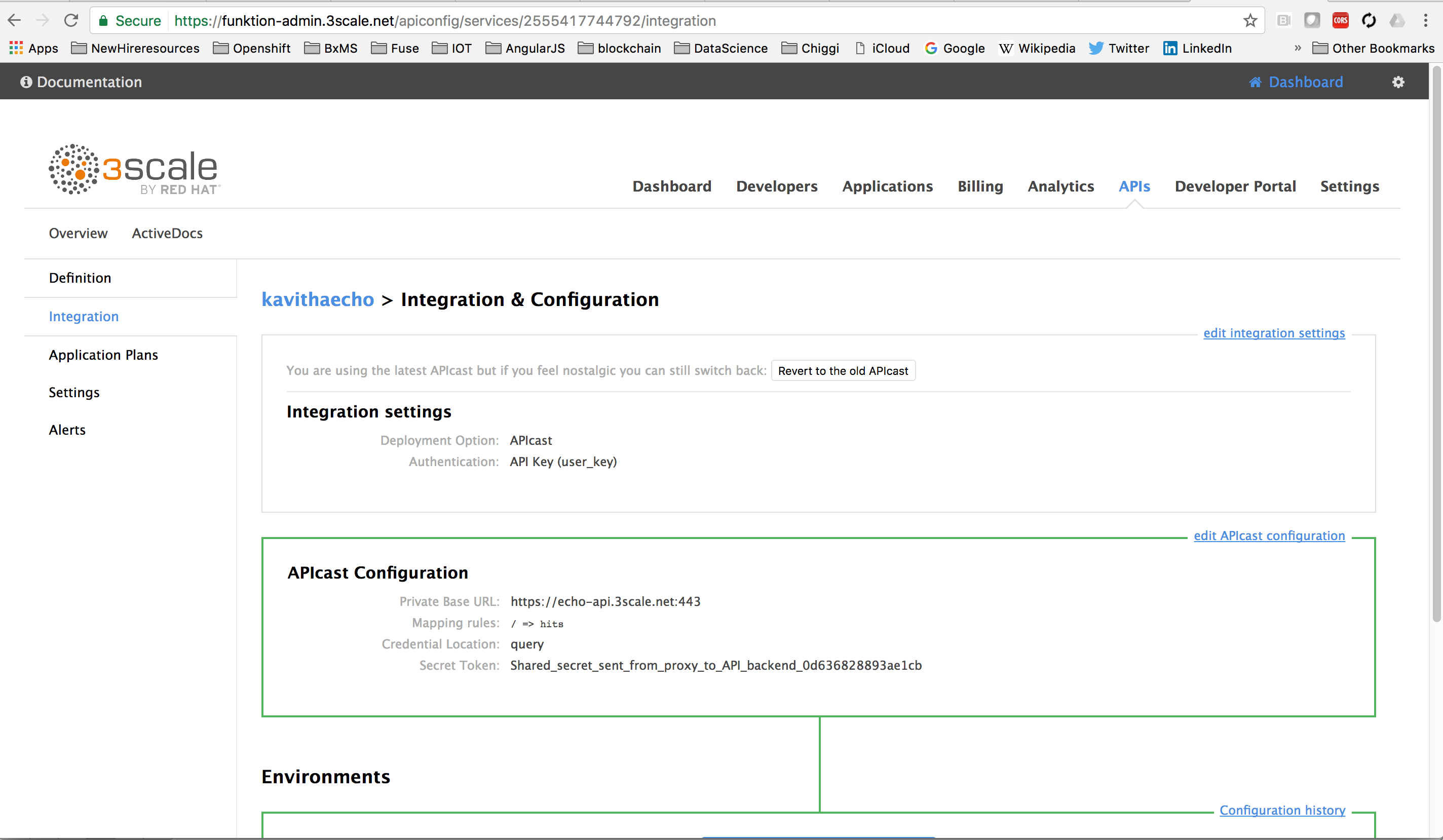
  
  
Name it, give it Read Write Access to everything, Create it and copy it somewhere (you won’t be able to get it again)  


Note: The Public Base URL in 3scale with the Openshift Route you will create shortly:

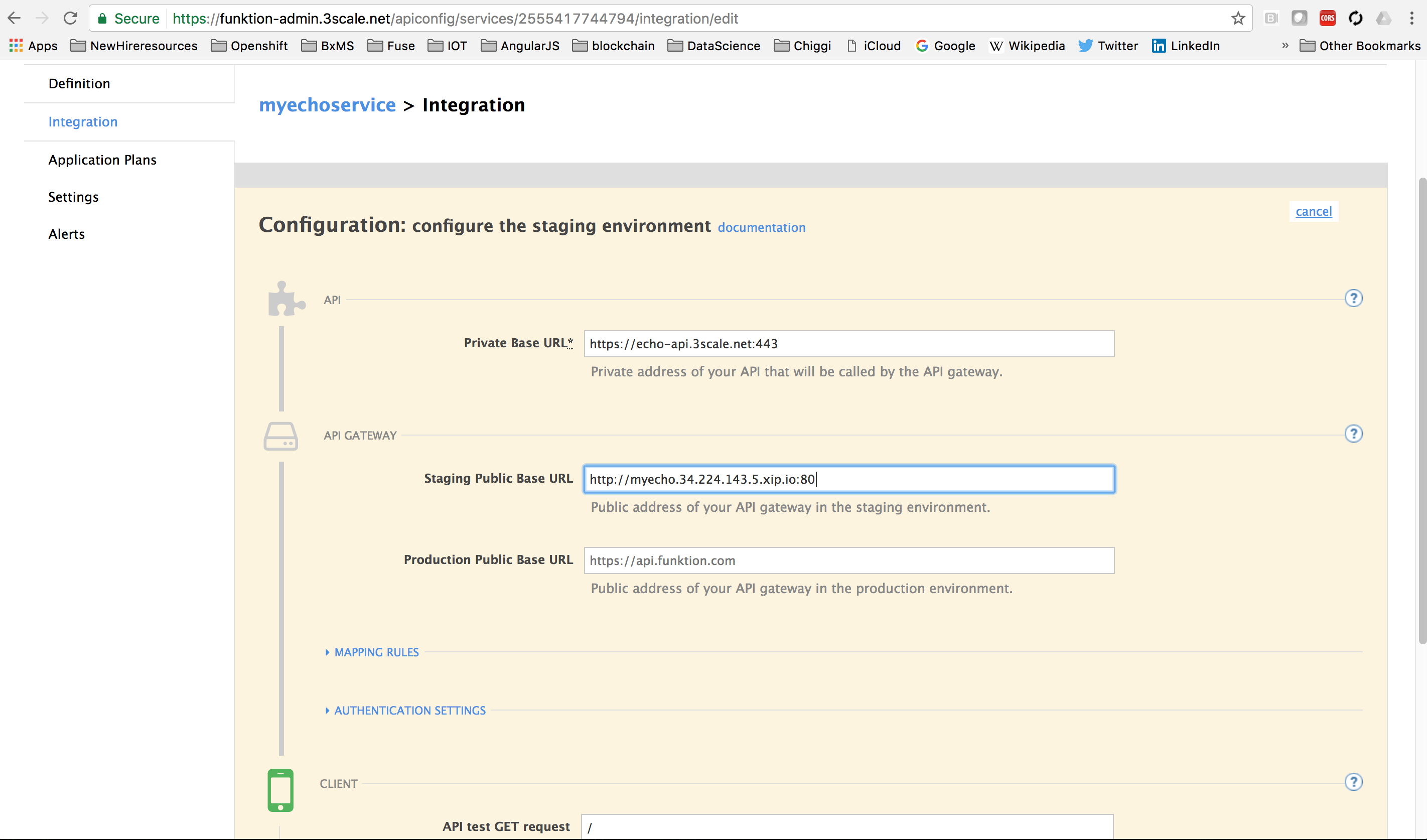
* + Go To APIs>create service



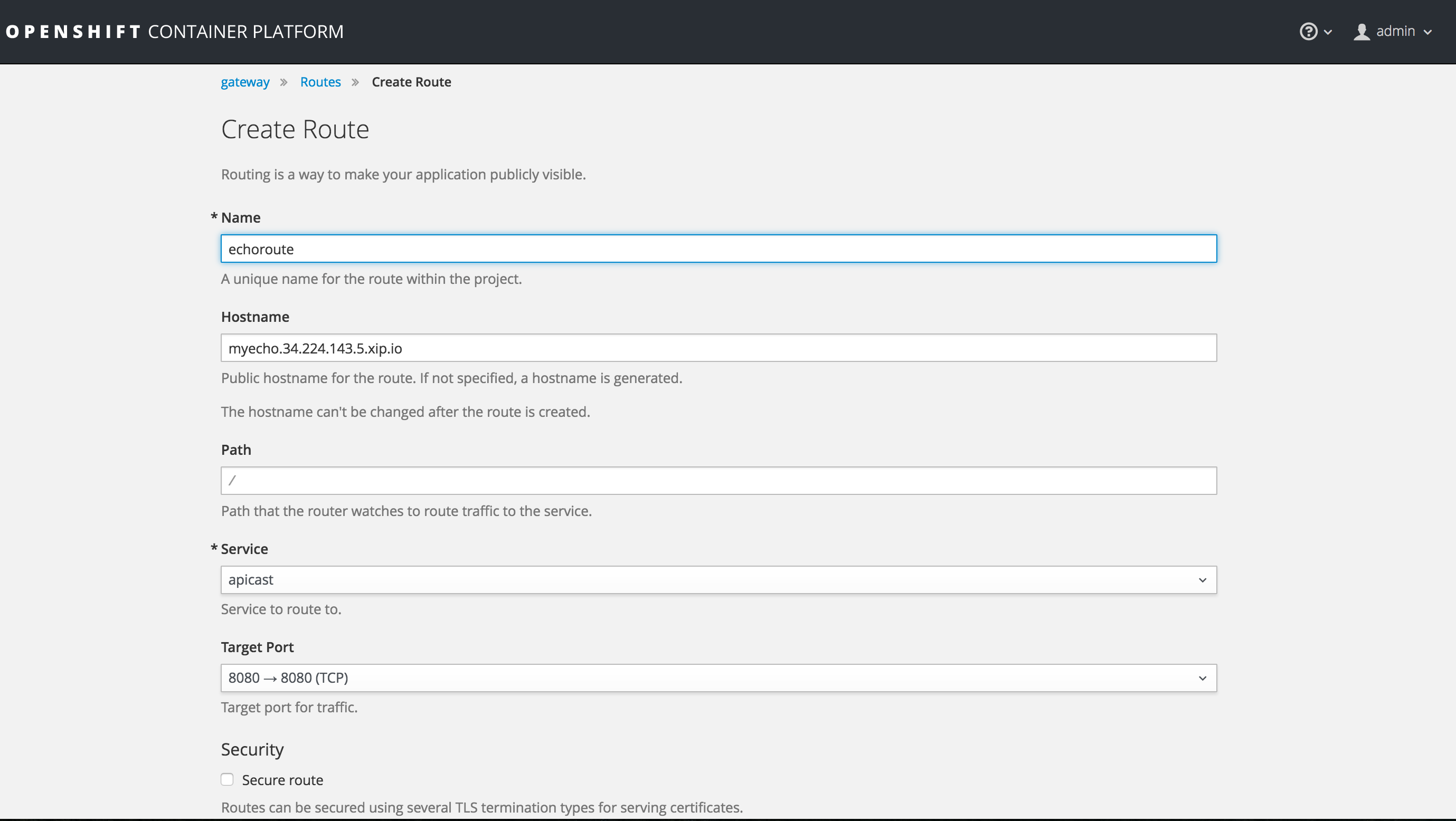
* + Scroll down and click ‘Create Service’
  + Now lets edit the service. Go to APIs>myechoserviceAPI(or any other name you have defined)>Integration>edit APICast configuration:



* + Click Update Service at the bottom.
  + Scroll to the bottom of the Integration Page and enter this in the *Public Base URL* then *Update Production Configuration*:  
    http://flights.<your RHEL box IP>.xip.io:80

This is the public base URL you will use to access this service.  


**Note**: the Staging Public base URL will match the Route host we will define in Openshift. The name doesn’t matter but a format of <logicalservicename>.<OpenshiftIP>.xip.io is recommended.

1. On Openshift, login as developer, create a Route corresponding to the Public Base URL you just created on 3scale (without the protocol and port).  
   

1. On your local machine
   * oc login (as developer)
   * change project if necessary: oc project 3scalegateway
   * create a secret:  
     oc secret new-basicauth **apicast-configuration-url-secret** --password=https://***<access token created above>***@***<your 3scale host>***  
     (***<your 3scale host>***  is in the format: xyz-admin.3scale.net)
   * create the app:  
     oc new-app -f <https://raw.githubusercontent.com/3scale/3scale-amp-openshift-templates/2.0.0.GA-redhat-2/apicast-gateway/apicast.yml>

This deploys the Gateway on Openshift.

Note: For every change made in the 3scale UI, a re-deployment is required.

1. Test the API:  
   http://myecho.34.224.143.5.xip.io/?user\_key=<***your user key>***  
   You’ll find ***<your user key>*** on the Integration screen in 3scale where you set your Public Base URL above.  
   Check your API calls are registered by going to the Analytics screen on 3scale. See previous workshop, *3scaleWorkshop-BasicConfig.docx*, for details on that.

Note, if you make any changes to the Integration page on 3scale, you’ll normally have to re-deploy on Openshift to pull down the new config to Openshift.

# Appendix 1 – Very basic Installation of Openshift on RHEL Follow the following steps:

sudo yum repolist all

sudo yum-config-manager --enable rhui-REGION-rhel-server-extras

sudo yum install docker docker-registry

cd /etc/sysconfig/

sudo vi docker

\*\*change the line

# INSECURE\_REGISTRY='--insecure-registry'

to

INSECURE\_REGISTRY='--insecure-registry 172.30.0.0/16'

DON'T FORGET TO UN-COMMENT

and save

sudo systemctl start docker

sudo systemctl status docker

\*\*\*Install Client tools - followed "Installing and Running an All-in-One Server" here: https://docs.openshift.org/latest/getting\_started/administrators.html

cd ~

sudo yum install wget

sudo wget https://github.com/openshift/origin/releases/download/v1.5.0-alpha.2/openshift-origin-client-tools-v1.5.0-alpha.2-e4b43ee-linux-64bit.tar.gz

sudo tar xzvf openshift-origin-client-tools-v1.5.0-alpha.2-e4b43ee-linux-64bit.tar.gz

sudo mv openshift-origin-client-tools-v1.5.0-alpha.2+e4b43ee-linux-64bit oc-cli-tools

sudo vi ~/.bash\_profile

\*\*\*make the changes save and exit

PATH=$PATH:/home/ec2-user/oc-cli-tools

source ~/.bash\_profile

\*\*\* check new location is on the path

echo $PATH

\*\*\*Start Openshift – replacing with your hostname and IP

sudo /home/ec2-user/oc-cli-tools/oc cluster up --public-hostname=***<your RHEL host>*** --routing-suffix==***<your RHEL IP>***.xip.io

\*\*\*Should see something like:  
 You are logged in as:

User: developer

Password: developer

\*\*\*Web console. Can login as developer/developer.

https://***<your host>***:8443