# Lab 1: Proxy an Existing API

[Lab 1: Proxy an Existing API](#_unhd2nhxtatb)

[Overview](#_fs8583o0cdf6)

[Step 1: Create an Anypoint Platform account](#_ni3afidknyyt)

[Step 2: Define the Proxy](#_8sncezb9pjww)

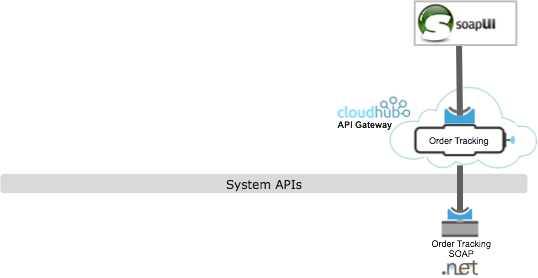
[Step 3: Test the Proxy in SoapUI](#_gbzi5w1obk6k)

[Summary](#_1ci18dljnnrx)

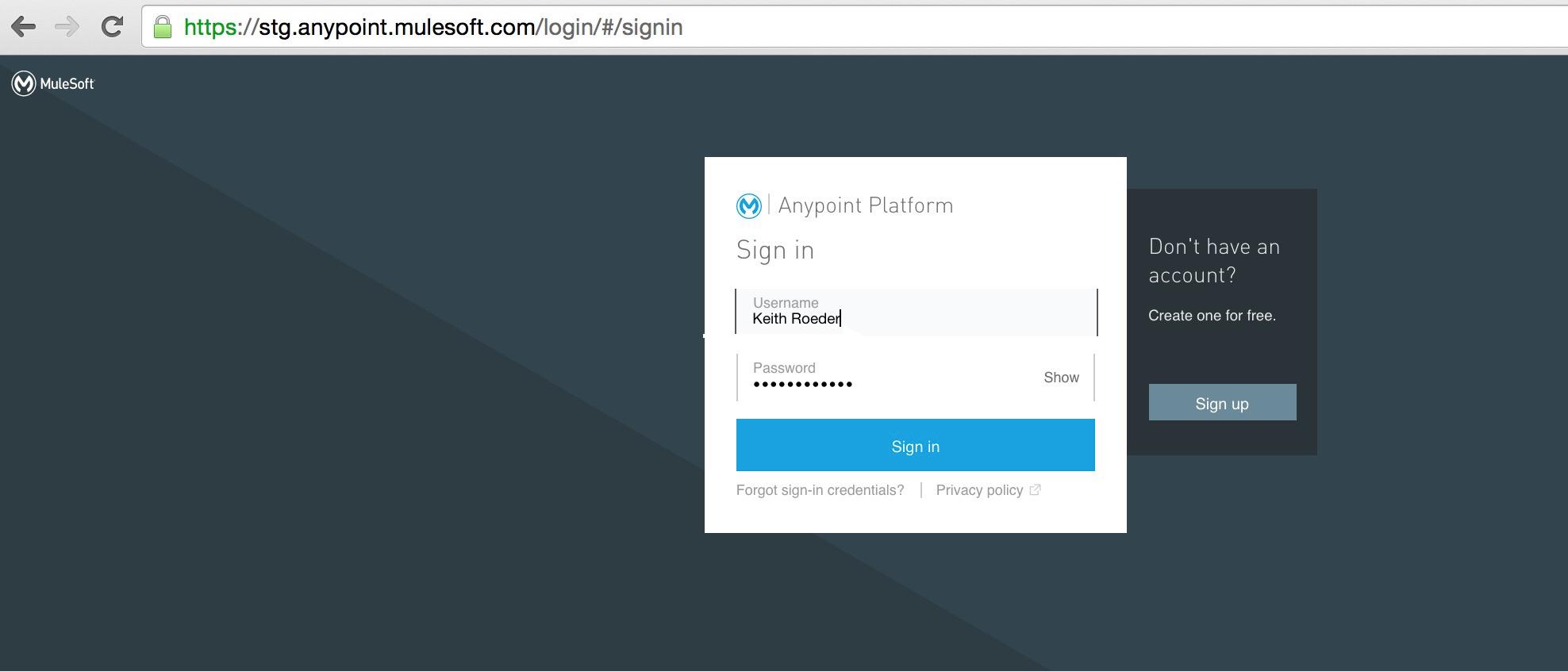
## Overview

To get started with the API Led Connectivity approach, let’s take a look at how to bring existing APIs to **Mulesoft’s** **Anypoint Platform**. We will leverage the API Gateway for running our APIs either on-premises or on the cloud and later on, manage the APIs with the API Manager, monitor it with API Analytics and provide access through the API Portal.

In this lab, we will proxy an existing SOAP-based system API used for order tracking. The proxy API will be deployed to the API Gateway on **CloudHub**, MuleSoft’s integration-Platform-as-a-Service (iPaaS) as shown in the diagram below. The gateway will host the API proxy application and will route requests to the existing .NET SOAP based **Order Tracking Service**. We will later use this proxy API in our mobile order entry use-case.



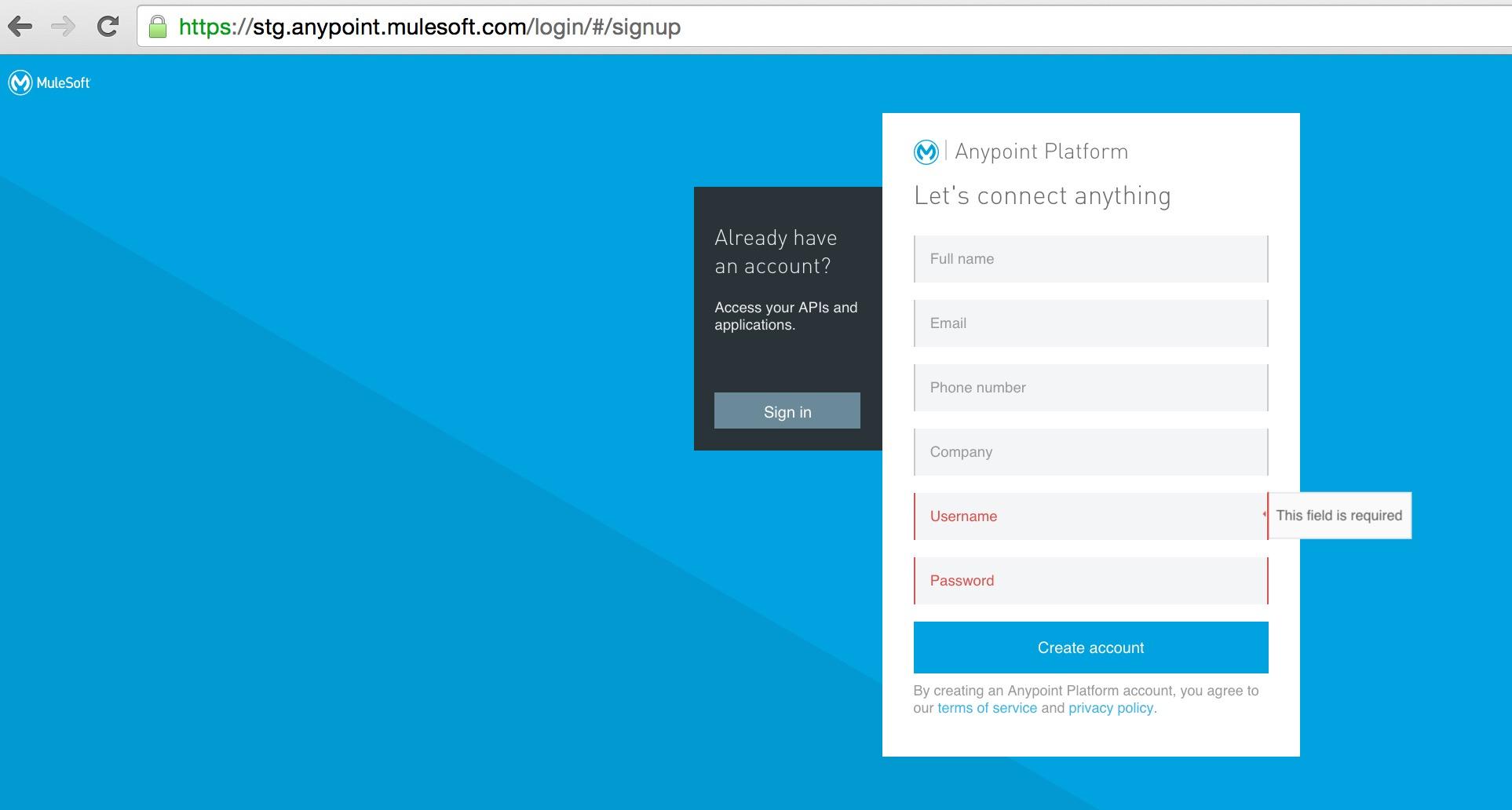
## Step 1: Create an Anypoint Platform account



To get started, you will create an Anypoint Platform account.

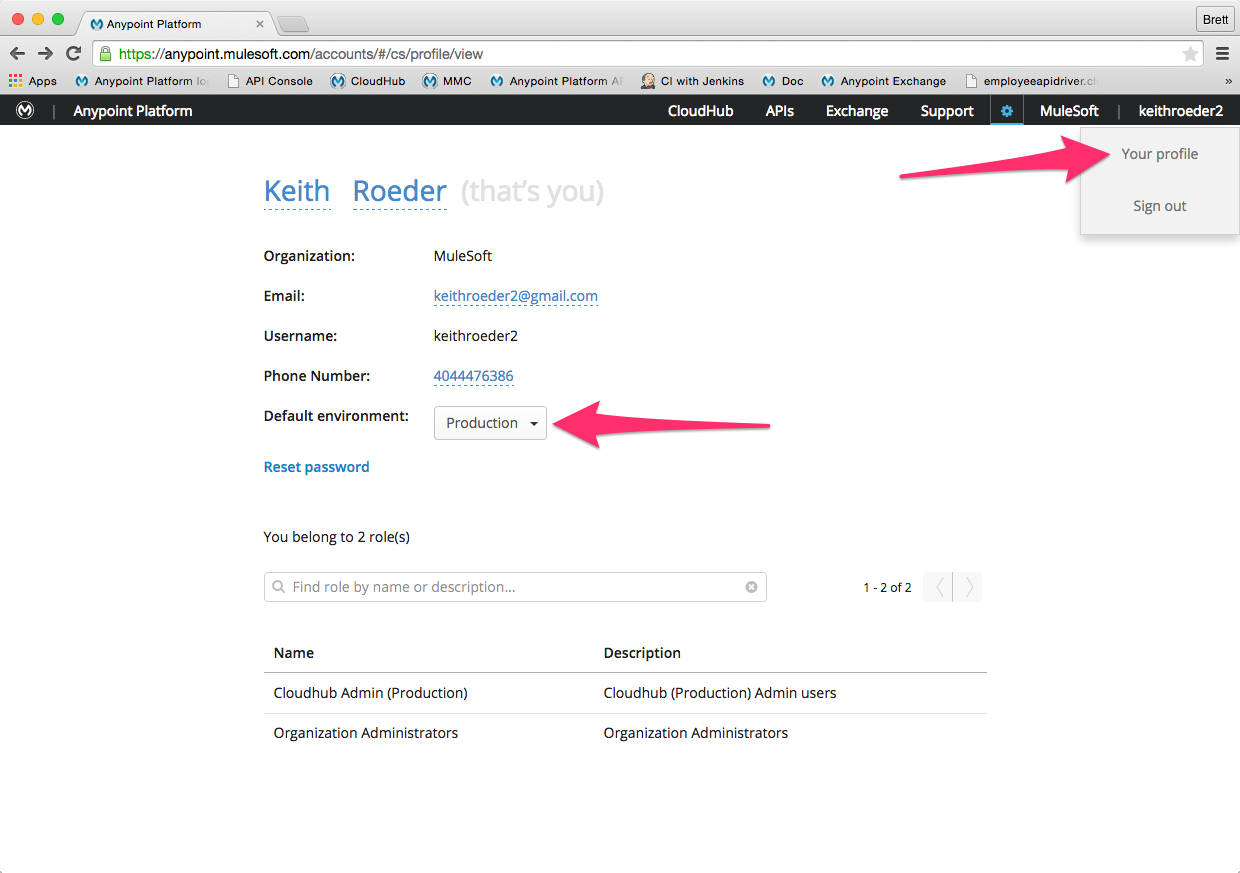
1. Go to<http://anypoint.mulesoft.com>
2. Click **Sign up** to create a new Anypoint Platform account.

|  |
| --- |
| **NOTE**: If you already have an account under your enterprise **DO NOT** use it for this workshop! Create a new account not linked to your organization. |



1. Enter your information
2. Click **Create account** a new API Platform Account .
3. Login with your newly created account.

### Step 2.1: Set your default environment



Before you can deploy any of the projects you will build you will need to set your default environment.

1. Click on your account name in the top right.
2. Click **Your profile**
3. Set your **Default environment** to *Production*
4. Close the confirmation message that appears.

Now you can deploy applications.

1. Click on **APIs** in the task bar to go to the main API page.

## 

## Step 2: Define the Proxy

Let’s take a look at how you can generate a proxy for an existing SOAP web service.

In your workshop instance You have an **Order Tracking** SOAP Service in .NET that you’d like to manage with your API Gateway. The WSDL for this service is accessible from this URL:

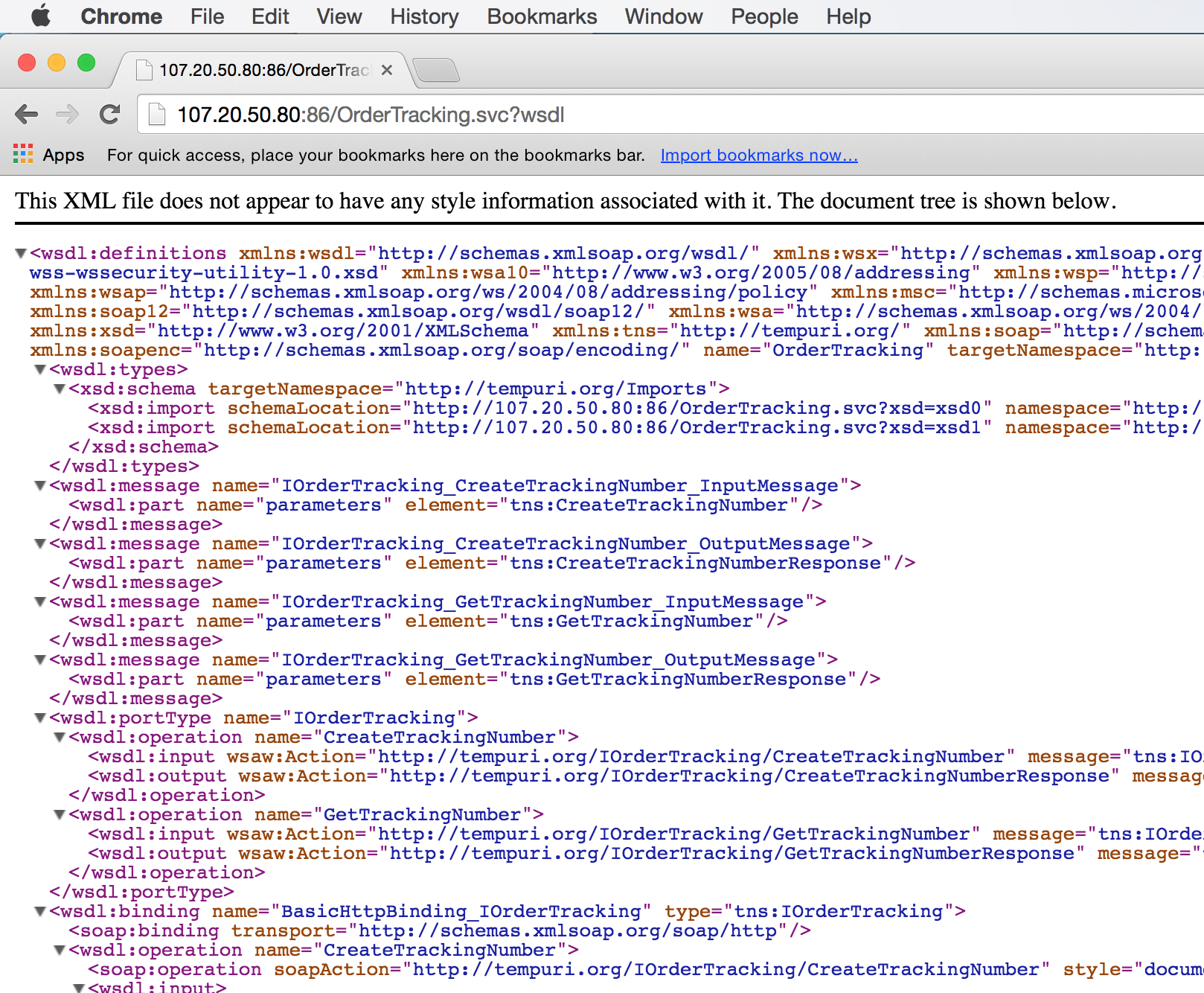
1. Copy the URL ([**http://<Your IP Address>:86/OrderTracking.svc?wsdl**](http://23.23.200.52:86/OrderTracking.svc?wsdl)) to the clipboard

|  |
| --- |
| **NOTE**: The Order Tracking SOAP Web Service is running on your workshop instance so you need to make sure that you access it using the Public IP address that was assigned to you.  **For example:** [**http://107.20.50.80:86/OrderTracking.svc?wsdl**](http://107.20.50.80:86/OrderTracking.svc?wsdl)  **Make sure you retain port 86** to the IP address. |

|  |
| --- |
| **DOUBLE CHECK**: Did you retain port 86 in the URL? |

1. **Open a new tab** to verify access to the WSDL URL
2. Hit enter to load the WSDL

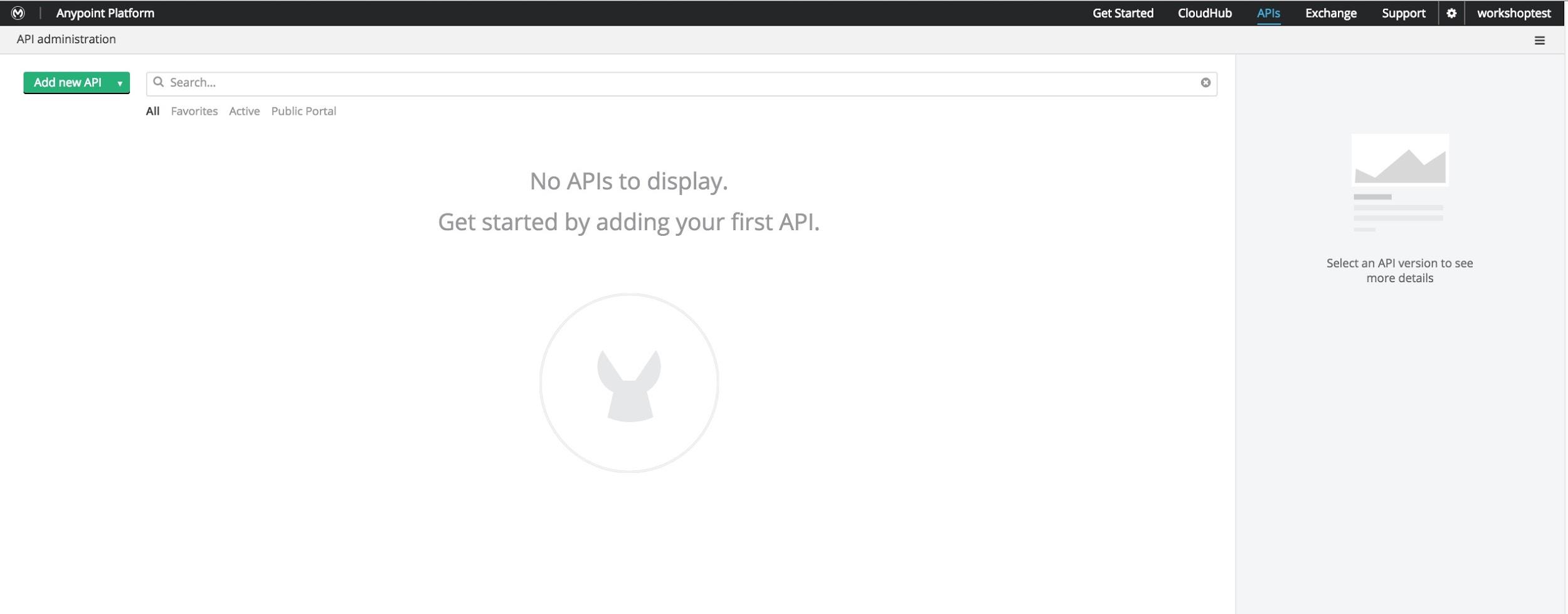
You should see the following



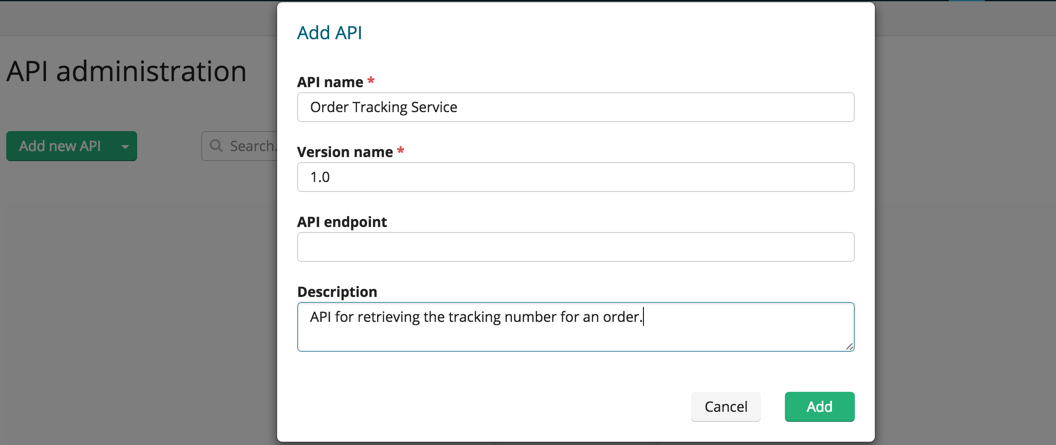
If you do not see the above:

* Check the IP address and verify it matches your assigned IP address
* Did you retain port 86?

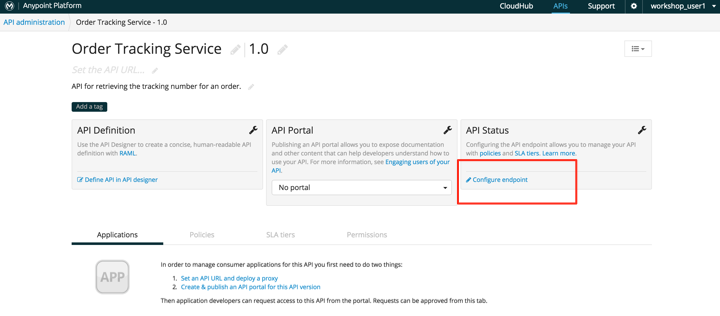
1. After you verify access to the WSDL, copy the URL to the clipboard
2. **Go back** to the main API administration page



1. Select **Add new API**



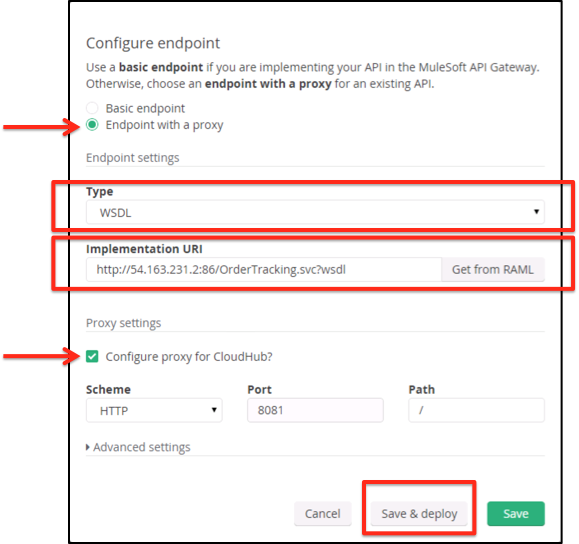
1. Enter the **API name** of **Order Tracking Service**.
2. Enter **Version name** of **1.0**
3. Leave the the **API Endpoint** blank for now.
4. Enter a **description**.
5. Click **Add**



You should see the **API administration** page for your newly created API. From this page you can **Define API in API designer** using RAML, publish a portal for the API, and **Configure endpoint** for the API.

For this lab we are only going to proxy the API. To do this:

1. Click on **Configure endpoint** under **API Status***.*

**

1. Select **Endpoint with a proxy**
2. Set **Type** to **WSDL** !!!! Make sure to set this to WSDL !!!!
3. Set the **Implementation URI**:

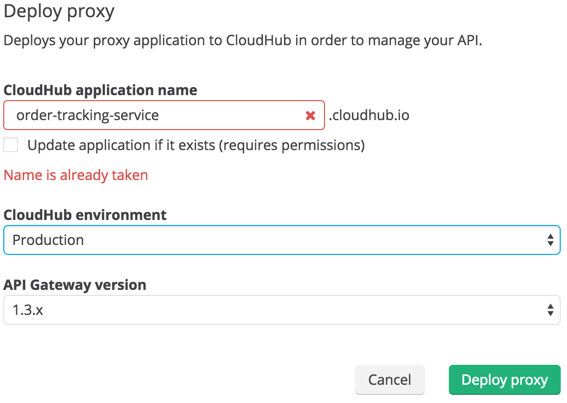
The SOAP web service is running on your workshop instance.

Use the **Public IP Address** that has been assigned to you.

[**http://<Your Public IP Address>:86/OrderTracking.svc?wsdl**](http://23.23.200.52:86/OrderTracking.svc?wsdl)

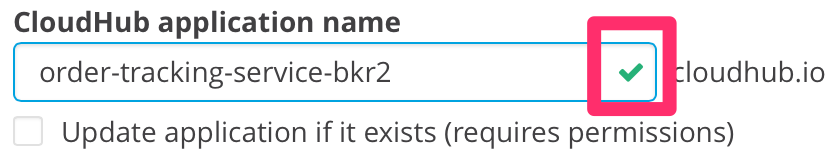
**For example:** [**http://54.163.231.2:86/OrderTracking.svc?wsdl**](http://54.163.231.2:86/OrderTracking.svc?wsdl)

1. Check **Configure proxy for CloudHub**
2. Click **Save & Deploy**

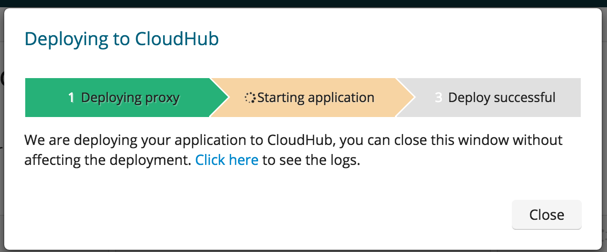


Since your API has not been deployed before you will be presented with a dialog to configure the run-time proxy.

1. Change the **CloudHub application name** to be unique by adding your name, initial or any random number until you see a Green Check Mark.

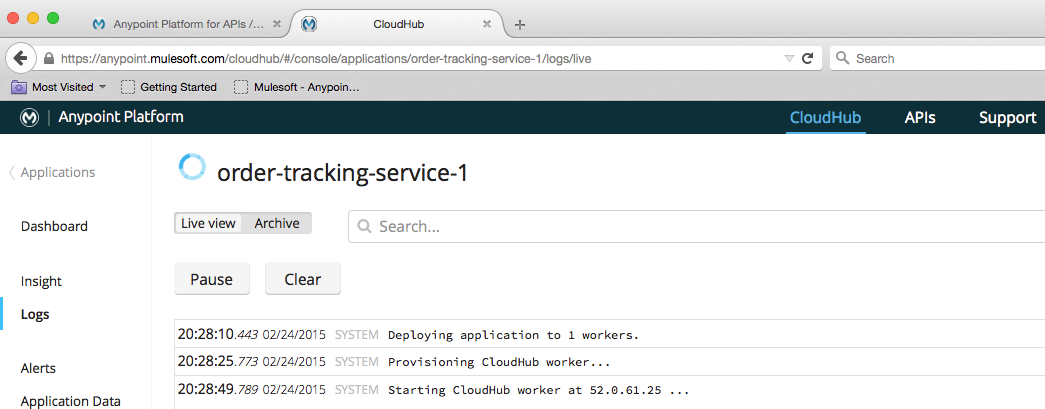


1. Ensure your **CloudHub environment** is set to Production
2. Set the **API Gateway version** to 1.3.x
3. Click **Deploy Proxy.**

****

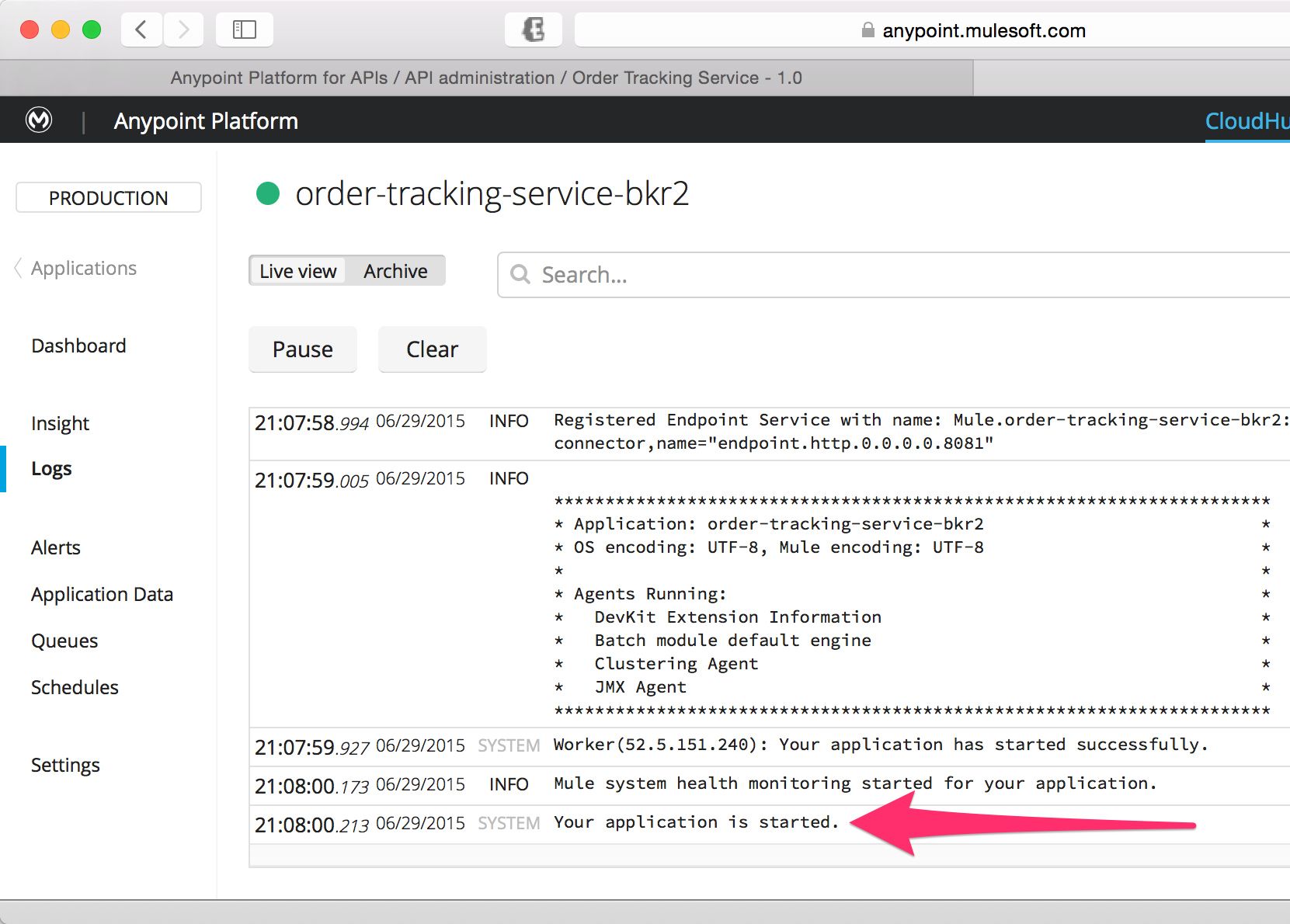
You will see the deploy status indicating “**Starting application**”.

1. Click on **Click Here** to see the log data and monitor the progress. A new browser tab will open that will show the CloudHub log for this application.

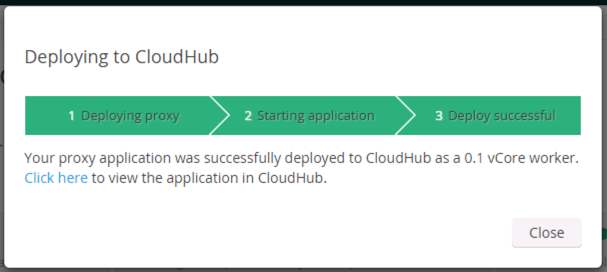


You’ll see a blue circle next to the application name indicating the application is deploying. CloudHub is allocating a worker with 0.1 vCore to host your new proxy application. You will know it’s complete when a green icon appears. In the meantime, feel free to look around (view the Logs to see what’s going on).

|  |
| --- |
| **NOTE**: It will take about three minutes to deploy the proxy. Good time to grab some coffee or explore the [documentation](http://www.mulesoft.org/documentation/display/current/Home).  Check out more information about [worker sizing](https://developer.mulesoft.com/docs/display/current/Deploying+a+CloudHub+Application). |



When you see ***Your application is started*** you can continue.



Or when you see all three green bars here.

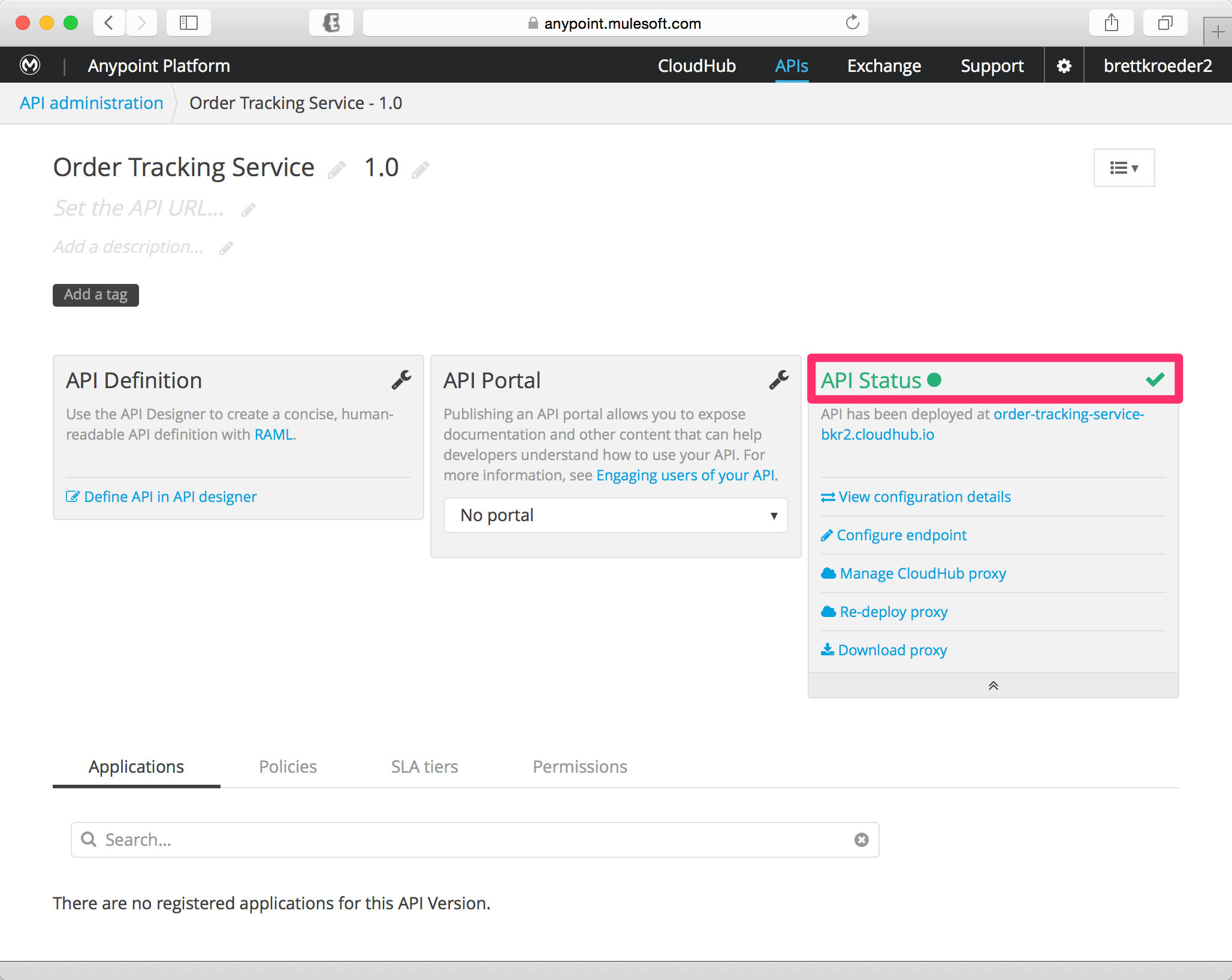
|  |
| --- |
| **NOTE**: CloudHub allows you to choose the appropriate size worker for your application. By default, the auto deploy chooses a 0.1 vCore worker. Check out the [*Deploying a CloudHub Application*](http://www.mulesoft.org/documentation/display/current/Deploying+a+CloudHub+Application) documentation for details about *Worker Sizing*. |

When the proxy has been successfully started

1. Go back to the previous browser tab where you were configuring the API.

You should see the green Deploy Successful in the status bar.

1. Click **Close**.

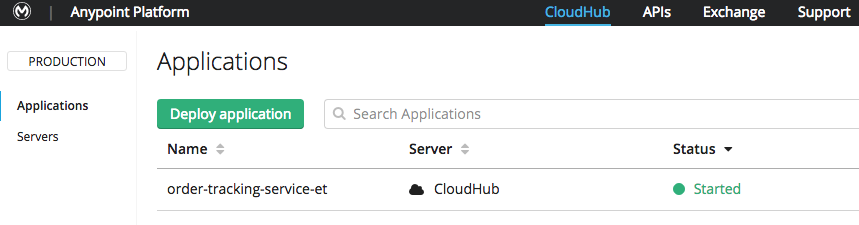
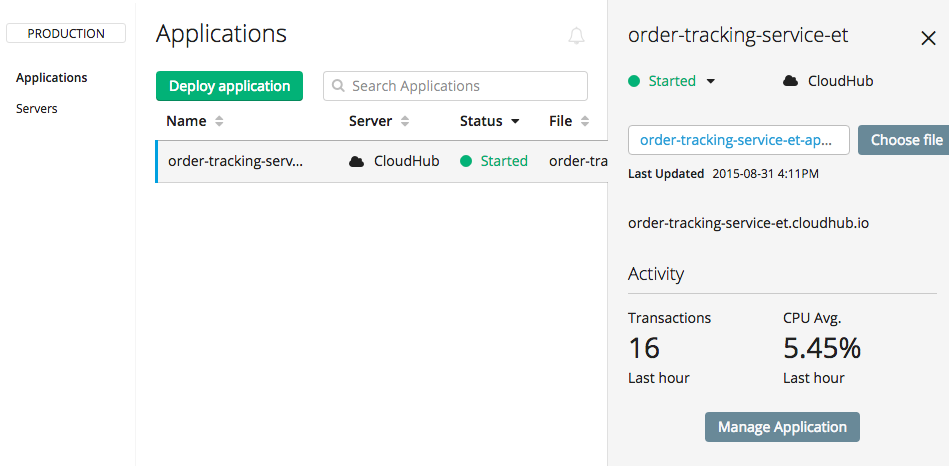
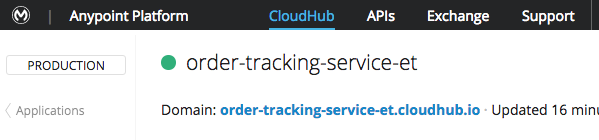


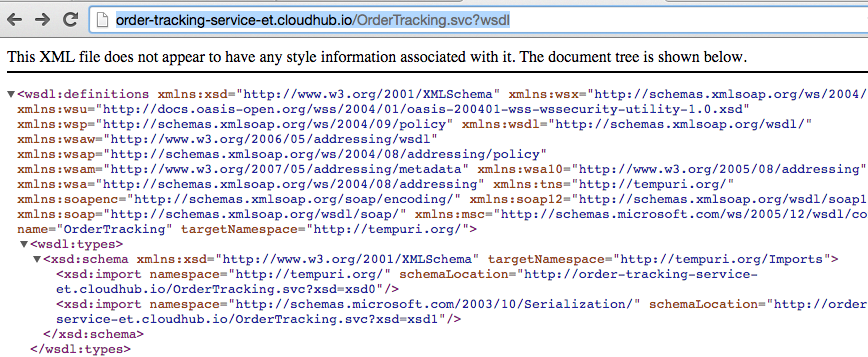
You should now see the **API Status** turn **green** with a **green** ball next to it, as shown above. This indicates that your API was successfully deployed and is now being managed. You should also see more options under the **API Status** section.

|  |
| --- |
| **NOTES**:   * The proxy you just created is a Mule application/project. If you want to see what is going on under the covers just download the proxy and import it into Anypoint Studio. * If needed, you could augment the generated flow with additional logic. * While you deployed this proxy to an API Gateway running on CloudHub, you could also deploy this proxy application to an on-premise **API Gateway**. |

## Step 3: Test the Proxy in SoapUI

Your proxy API is now accessible through CloudHub.

1. Let's review how to get the exact URL of your hosted proxy in Cloudhub. Go to CloudHub → Applications → Click on your deployed proxy application  
     
   
2. Click on **Manage Application  
   **
3. You will see the exact URL of your application ****
4. Try accessing its WSDL through your browser using the CloudHub domain name you specified adding **/OrderTracking.svc?wsdl** at the end.  
     
   For example: <http://order-tracking-service-et.cloudhub.io/OrderTracking.svc?wsdl>



You can test the SOAP service by using SoapUI.

1. Open SoapUI from the Desktop or from the Windows Task Bar.



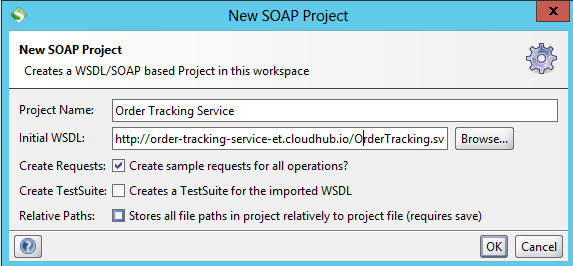


1. Create a new project with the following parameters:

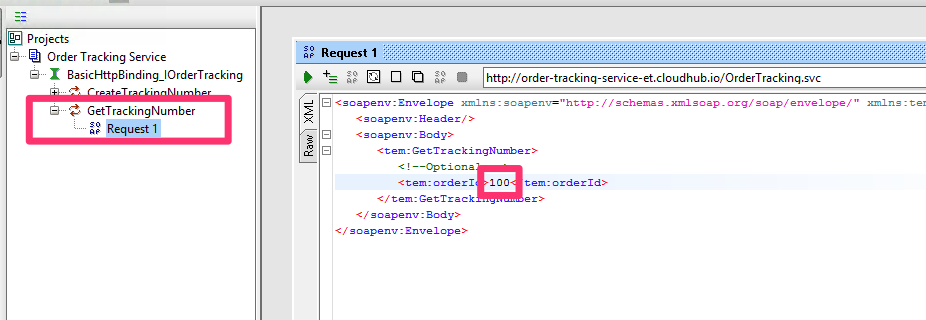
**Name**: Order Tracking Service

**Initial WSDL**: *URL you used in previous step*

**For example:** [**http://order-tracking-service-et.cloudhub.io/OrderTracking.svc?wsdl**](http://order-tracking-service-et.cloudhub.io/OrderTracking.svc?wsdl)



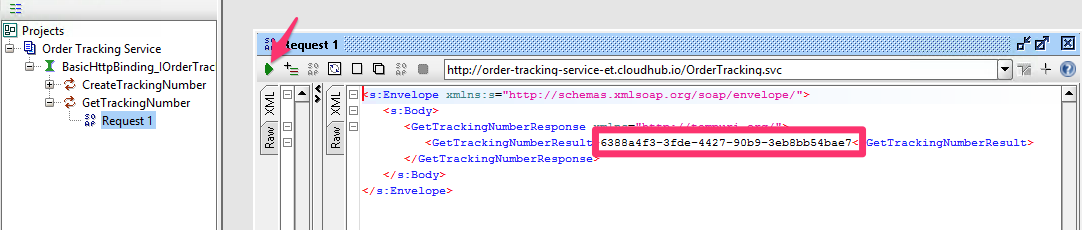
Once the WSDL has been introspected SOAP-UI will open the project.



1. Expand the **GetTrackingNumber** operation

|  |
| --- |
| **DOUBLE CHECK!** Are you sure you selected the GetTrackingNumber operation? |

1. Double-click on **Request 1** to open it.
2. Enter an **orderID** of **100** in **Request 1**
3. Click the Play icon to test.



You should see a response with a tracking number.

## Summary

In this lab, we completed the following steps:

[Step 1: Create an Anypoint Platform account](#_ni3afidknyyt)

[Step 2: Define the Proxy](#_8sncezb9pjww)

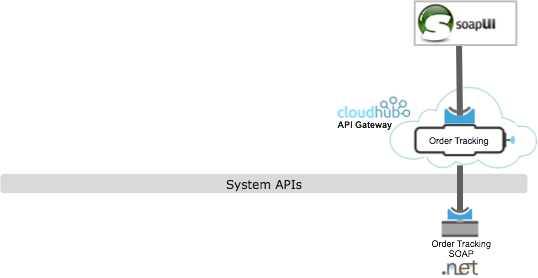
[Step 3: Test the Proxy in SoapUI](#_gbzi5w1obk6k)

We saw how easy it is to proxy existing web services and deploy it on a CloudHub-based API Gateway. This provides a **low friction** approach to manage existing APIs across your environment. You are able to leverage a **hybrid** API Gateway service offering which can be on-premise or on the cloud. In this lab, we saw how we can deploy our API to CloudHub which can significantly **speed up your deployment** without having to manage and maintain infrastructure.

For further reading on deploying an API gateway and proxying your API, please refer to the following documentation:

* [Configuring an API Gateway](http://www.mulesoft.org/documentation/display/current/Configuring+an+API+Gateway)

* [Proxying Your API](http://www.mulesoft.org/documentation/display/current/Proxying+Your+API)



Congratulations! You have finished Lab 1.

Please update the spreadsheet indicating you have completed Lab 1.