DAVID ARNOLD

daverno@au1.ibm.com

Abstract

Leveraging the ACE micro services 1,2 and 3 on RHOS and testing them with IBM (Rational) Integration Tester. The IBM (Rational) Service Virtualization Tester will also be used for mocking and stubbing ACE MS2 and an MQ backend application

IBM Middleware Tested on RHOS with IBM (Rational) Integration Tester

Testing ICP4i (ACE and MQ) custom images on RHOS 4.2 with Rational Integration Tester

Contents

[Using IBM (Rational) Integration Tester with ACE custom images on OPENSHIFT 3](#_Toc36540254)

[Introduction 3](#_Toc36540255)

[Testing Environment 3](#_Toc36540256)

[Overview Diagram 4](#_Toc36540257)

[IBM (Rational) Integration Tester – downloads 4](#_Toc36540258)

[IBM Integration Tester 4](#_Toc36540259)

[IBM installation manager for the IBM Integration Tester install 5](#_Toc36540260)

[Download/clone the IBM (Rational) Integration Tester project 5](#_Toc36540261)

[Open IBM Integration Tester and direct to workspace 5](#_Toc36540262)

[Navigating the IBM (Rational) Integration Tester 7](#_Toc36540263)

[Architecture School - Schema library 7](#_Toc36540264)

[Architecture School - Physical View 7](#_Toc36540265)

[Architecture School - Logical View 9](#_Toc36540266)

[Test Factory - Tests 11](#_Toc36540267)

[Test Factory – Stubs 13](#_Toc36540268)

[Explore the Test Lab 14](#_Toc36540269)

[Testing ACE Micro Service 2 Stub on IBM (Rational) Integration Tester – local client 17](#_Toc36540270)

[Run the ACE Micro Service 2 stub 17](#_Toc36540271)

[Test ACE Micro Service 2 stub with a local REST client 17](#_Toc36540272)

[Testing ACE Micro Service 2 Stub on IBM (Rational) Integration Tester – Public IP 22](#_Toc36540273)

[IBM Secure Gateway Service – Server side 22](#_Toc36540274)

[IBM Secure Gateway Client side 23](#_Toc36540275)

[Windows firewall 26](#_Toc36540276)

[Test calling the RSVT ACE MS2 stub via public IP 26](#_Toc36540277)

[Calling ACE MS1 via IBM Rational Integration Tester – ACE MS1 calls ACE MS2 stub 29](#_Toc36540278)

[Get the ACE MS1 route details from RH OpenShift console 30](#_Toc36540279)

[Start the IBM Rational Integration Tester ACE MS2 Stub 31](#_Toc36540280)

[Invoke the REST client to call ACE MS1 31](#_Toc36540281)

[Observe results in ACE MS 2 stub console 32](#_Toc36540282)

[Invoke ACE MS1 on RH OpenShift from a IBM Rational Integration Tester 33](#_Toc36540283)

[Observe the results in the stub 35](#_Toc36540284)

[Observe results in the test console 35](#_Toc36540285)

[Using ACE Micro Service 3 (MS3) with a remote MQ Queue Manager 38](#_Toc36540286)

[Configure Queue Manager MyAPPQMGR on the IBM Integration Tester Machine 38](#_Toc36540287)

[MQ SVRCONN Channel creation for Linux based queue manager 38](#_Toc36540288)

[MQ SVRCONN Channel creation for windows based queue manager 38](#_Toc36540289)

[Set up IBM Secure Gateway Service for Public IP to Queue Manager MyAPPQMGR 39](#_Toc36540290)

[IBM Cloud server side 39](#_Toc36540291)

[IBM Secure Gateway Service – Client (Laptop end) 42](#_Toc36540292)

[Windows firewall 42](#_Toc36540293)

[IBM Secure Gateway Service client side 42](#_Toc36540294)

[Using/Matching Secure Gateway parameters on ACE MS 3 MQOutput configuration 43](#_Toc36540295)

[Testing MQ via IBM Public IP – using RFHUTILC (MQ client test) 46](#_Toc36540296)

[Check the result on the target queue manager 47](#_Toc36540297)

[Testing ACE Microservice 3 to Put to MQ via IBM Public IP – Using a REST Client 48](#_Toc36540298)

[Obtain the RH OpenShift Route for ACE MS3 48](#_Toc36540299)

[URL and test data for calling ACE MS3 service 48](#_Toc36540300)

[Test with REST Client 49](#_Toc36540301)

[Check Test Results on MQ 50](#_Toc36540302)

[Using the MQ Stub on IBM Rational Integration Tester as a backend application 51](#_Toc36540303)

[Review stub parameters in IBM Rational Integration Tester 51](#_Toc36540304)

[Stub uses IVT.SVRCONN channel definition in the MyAPPQMGR 53](#_Toc36540305)

[Start IBM Rational Integration Tester Stub to retrieve messages placed on MQ by ACE MS3 54](#_Toc36540306)

[IBM Rational Integration Tester Stub receives message from ACE MS3 55](#_Toc36540307)

[Review MyAPPQMGR queue FROM.BACKEND.Q contents 55](#_Toc36540308)

# Using IBM (Rational) Integration Tester with ACE custom images on OPENSHIFT

## Introduction

Assuming the role of the “disconnected” integration developer we will explore using components of the IBM (Rational) Test Workbench to

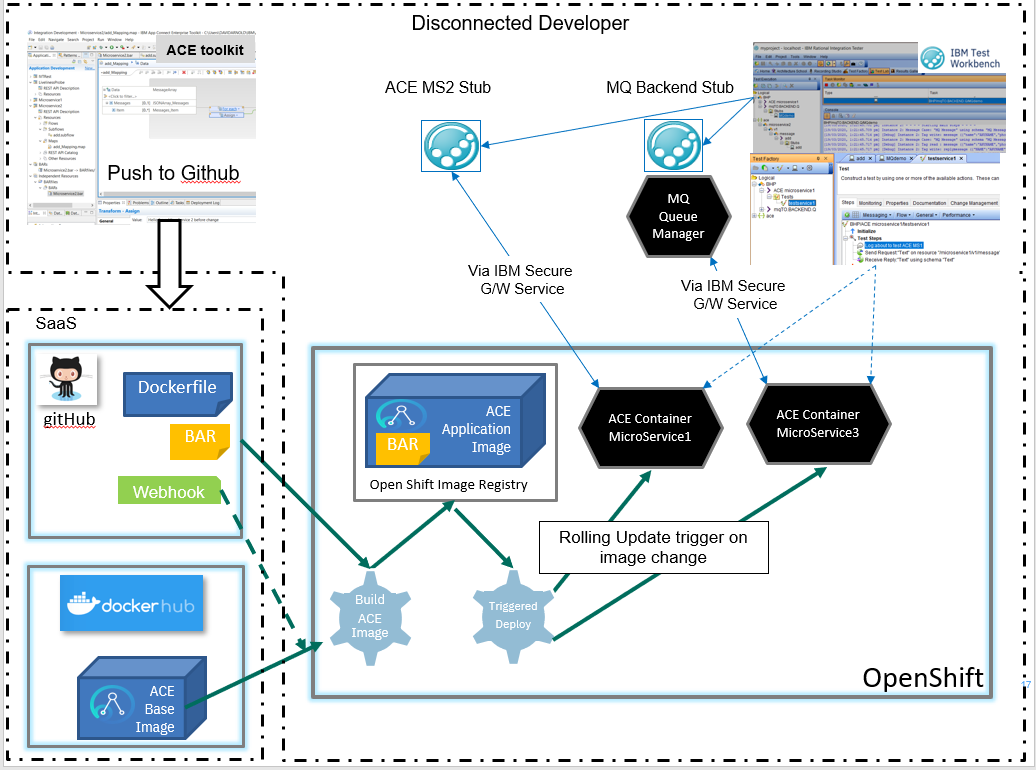
1. Test ACE microservices deployed on RH OpenShift 4.2
   1. ACE microservice 1 that will be configured call the “Stub” implementation of ACE microservice 2
   2. ACE microservice 3 that will place a message on MQ for consumption by a “stub”
2. Run “Stubs”
   1. ACE microservice 2 stub
   2. MQ backend application stub

## Testing Environment

In the examples on this document.

* IBM ACE Toolkit is on a Windows T480 laptip
* IBM (Rational) Integration Tester (with Virtualization Test Server) on Windows T480 laptop
* RH OpenShift 4.2 on IBM Public Cloud
* IBM Cloud Secure Gateway Service is used to offer public IP addresses to the T480 laptop
  + Note: CodeReady, Minishift or similar could have be used locally on the laptop
  + The Secure Gateway service would not have been required in that case

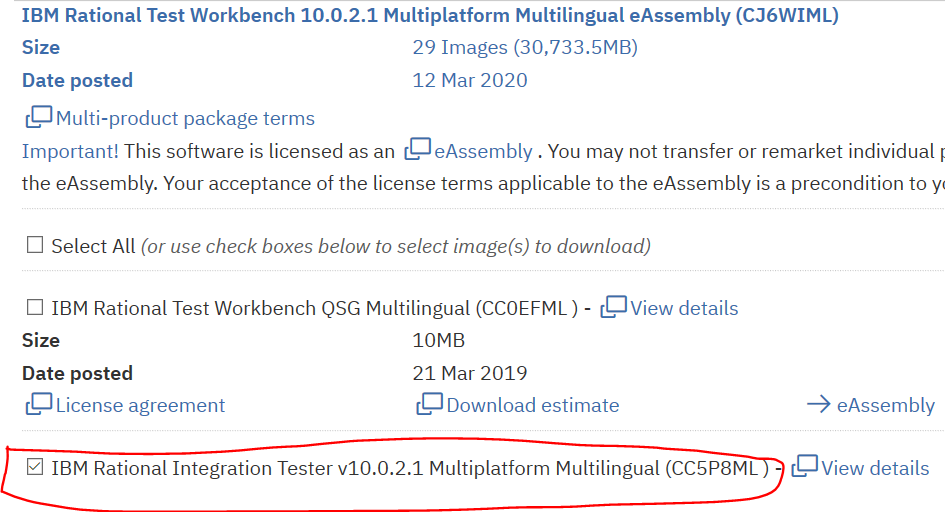
## Overview Diagram



## IBM (Rational) Integration Tester – downloads

Download IBM Rational Integration Tester v10.0.2.1

### IBM Integration Tester



### IBM installation manager for the IBM Integration Tester install

Download and install the IBM Installation manager

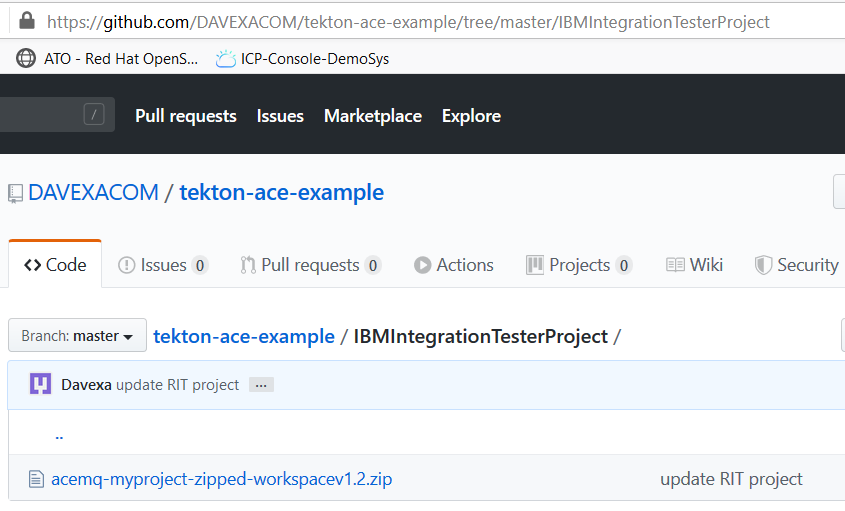
<https://jazz.net/downloads/ibm-installation-manager/releases/1.9.1.1/agent.installer.win32.win32.x86_64_1.9.1001.20191112_1525.zip>

Use the installation manager to install IBM Integration Tester

## Download/clone the IBM (Rational) Integration Tester project

Available from github - https://github.com/DAVEXACOM/tekton-ace-example

<https://github.com/DAVEXACOM/tekton-ace-example/tree/master/IBMIntegrationTesterProject>

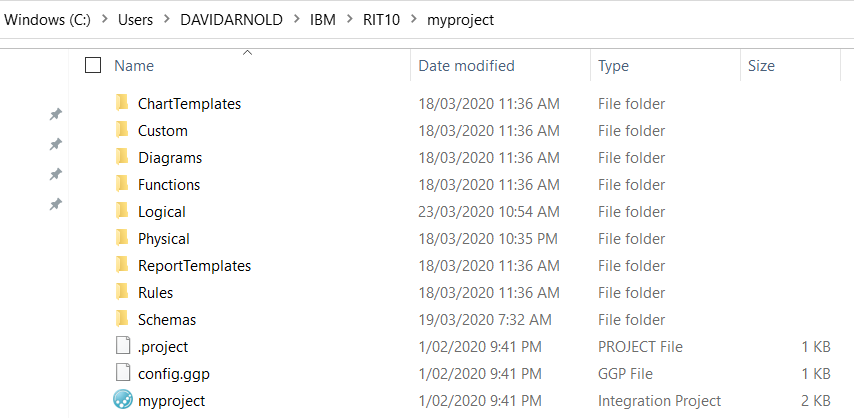


This is a ZIP of the IBM Integration Tester workspace for the ACE and MQ artifacts.

### Open IBM Integration Tester and direct to workspace

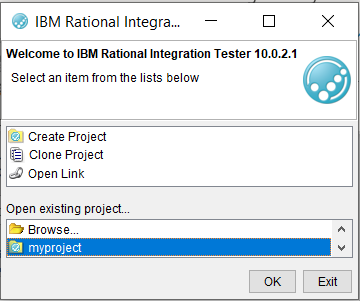
Unzip the IBM Integration Tester workspace

Example: c:\users\username\IBM\RIT10



Open the IBM Integration Tester.

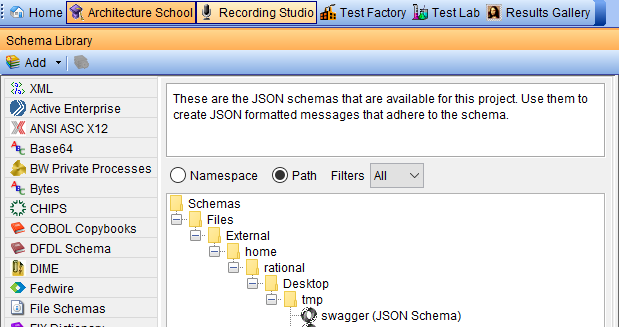
The first time in you will Browse to open an existing project. Subsequent you can open and existing project.



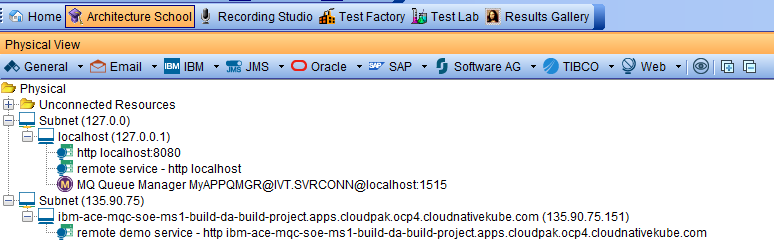
## Navigating the IBM (Rational) Integration Tester

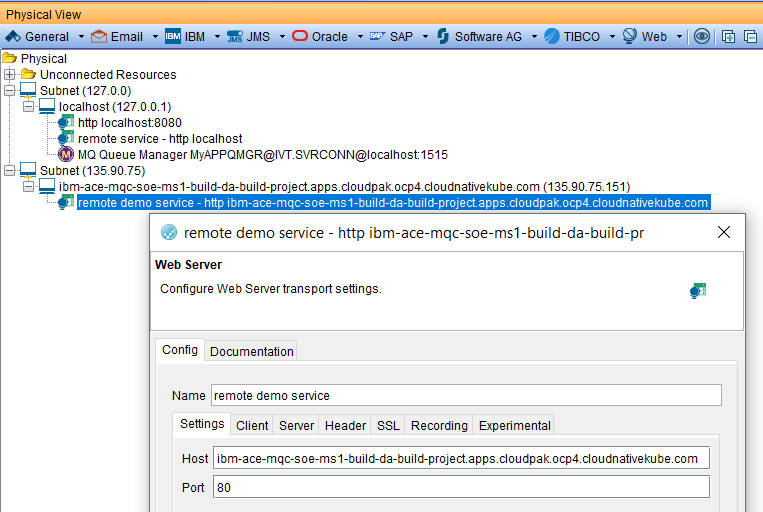
### Architecture School - Schema library

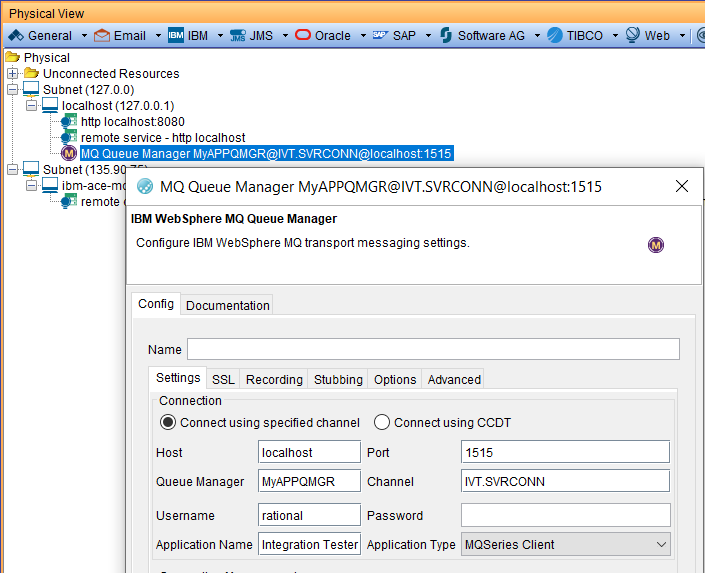
ACE Microservice open api documents imported here



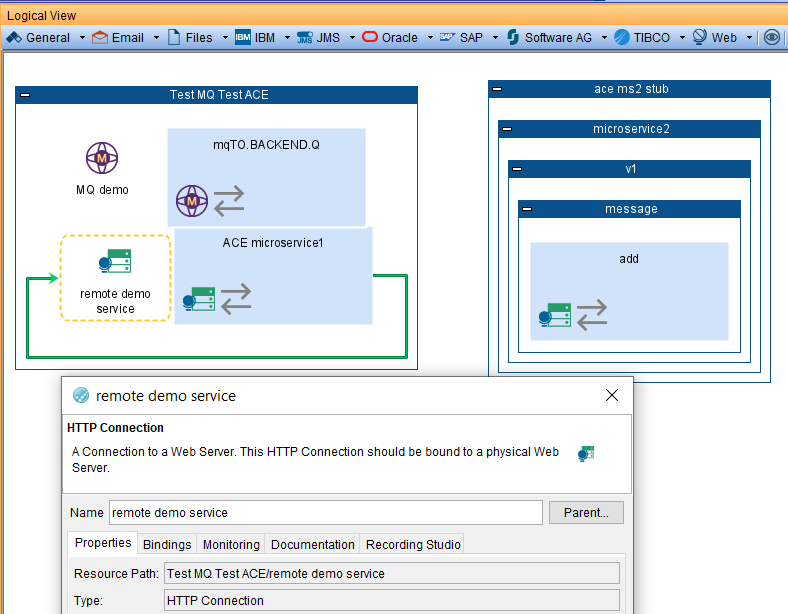
### Architecture School - Physical View



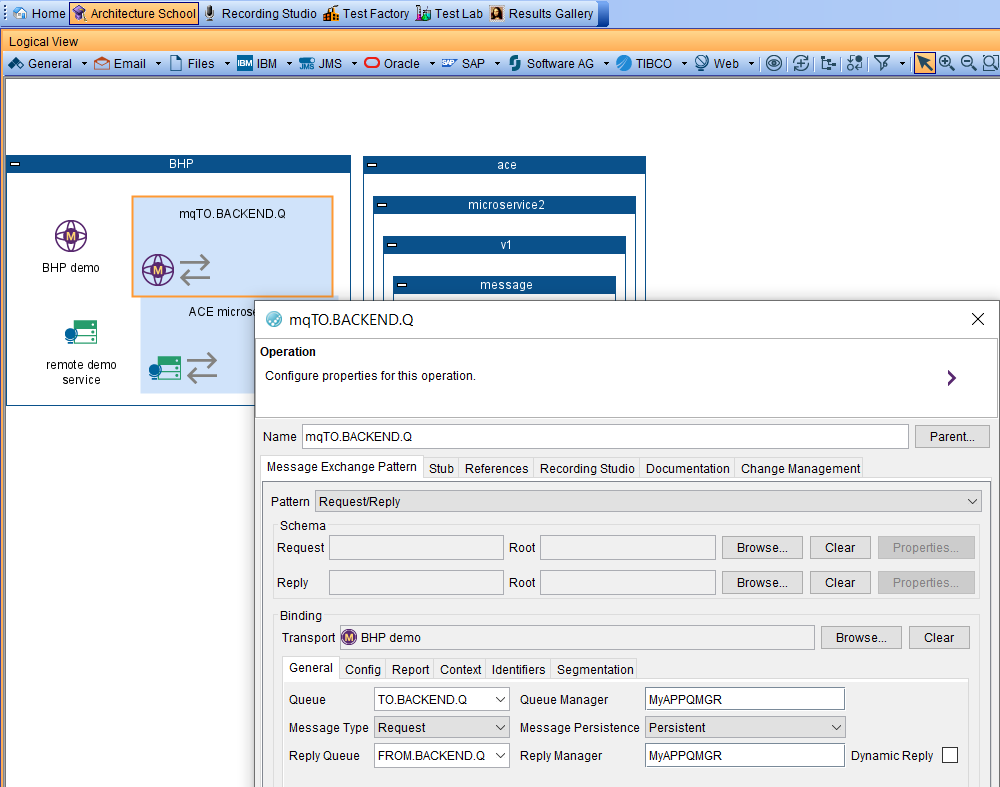




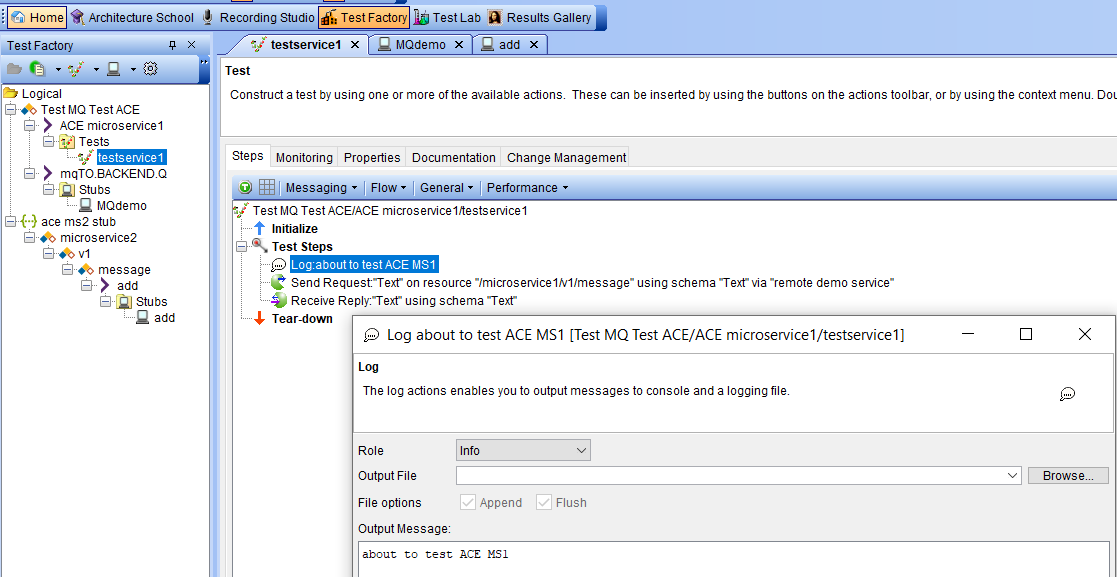
### Architecture School - Logical View

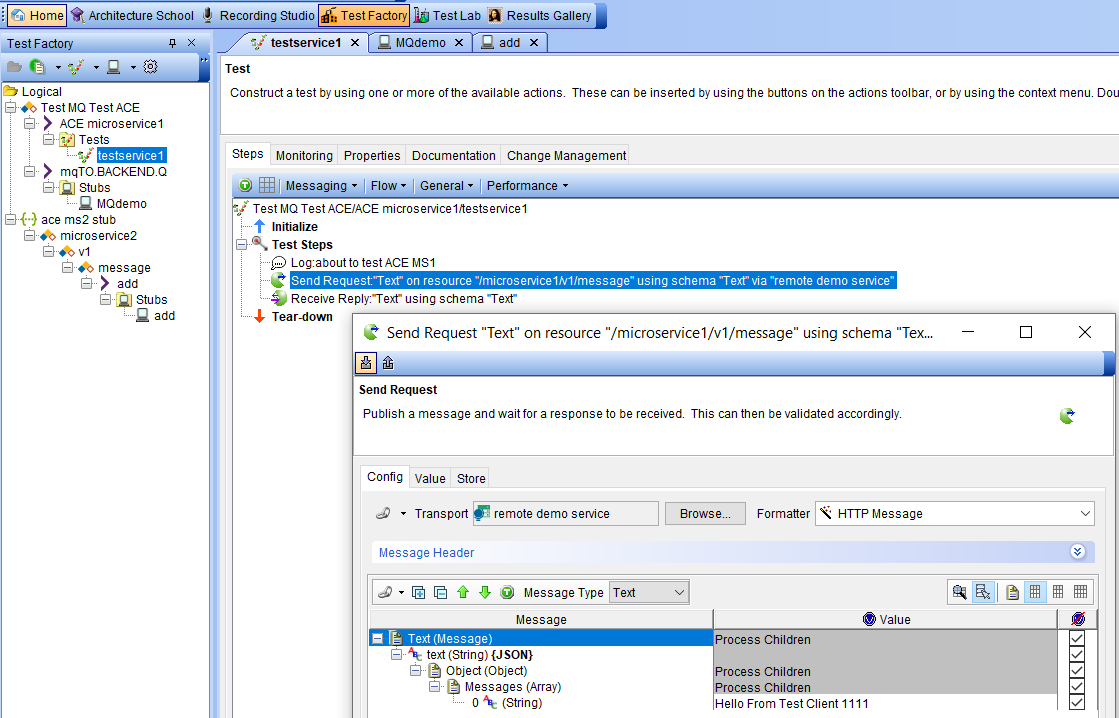


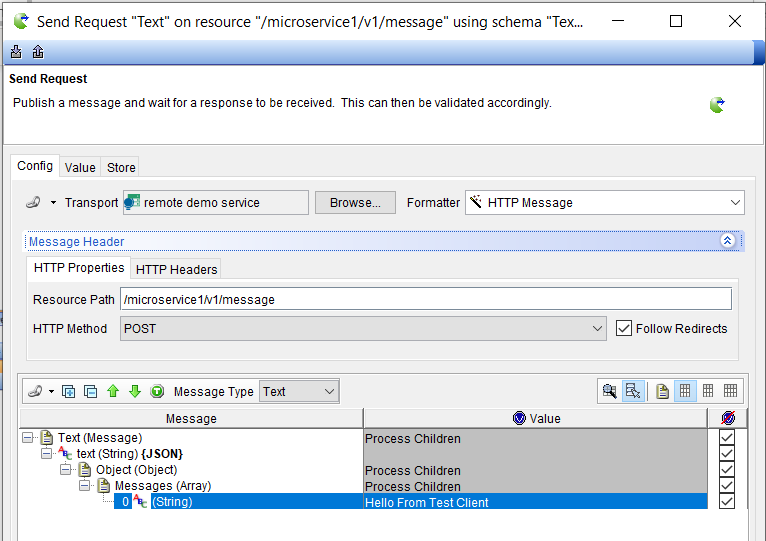




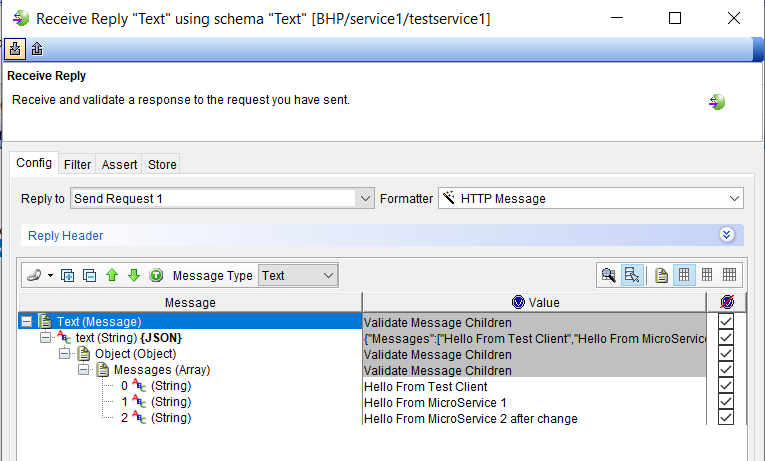
### Test Factory - Tests





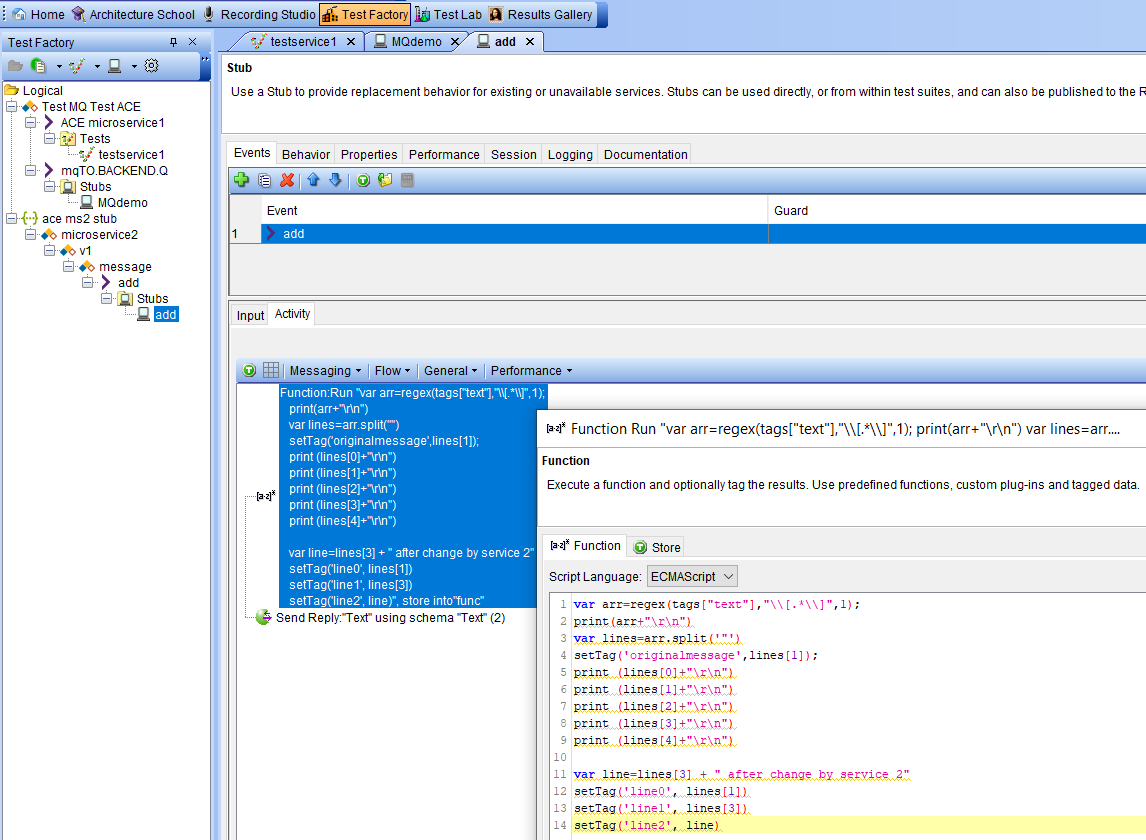


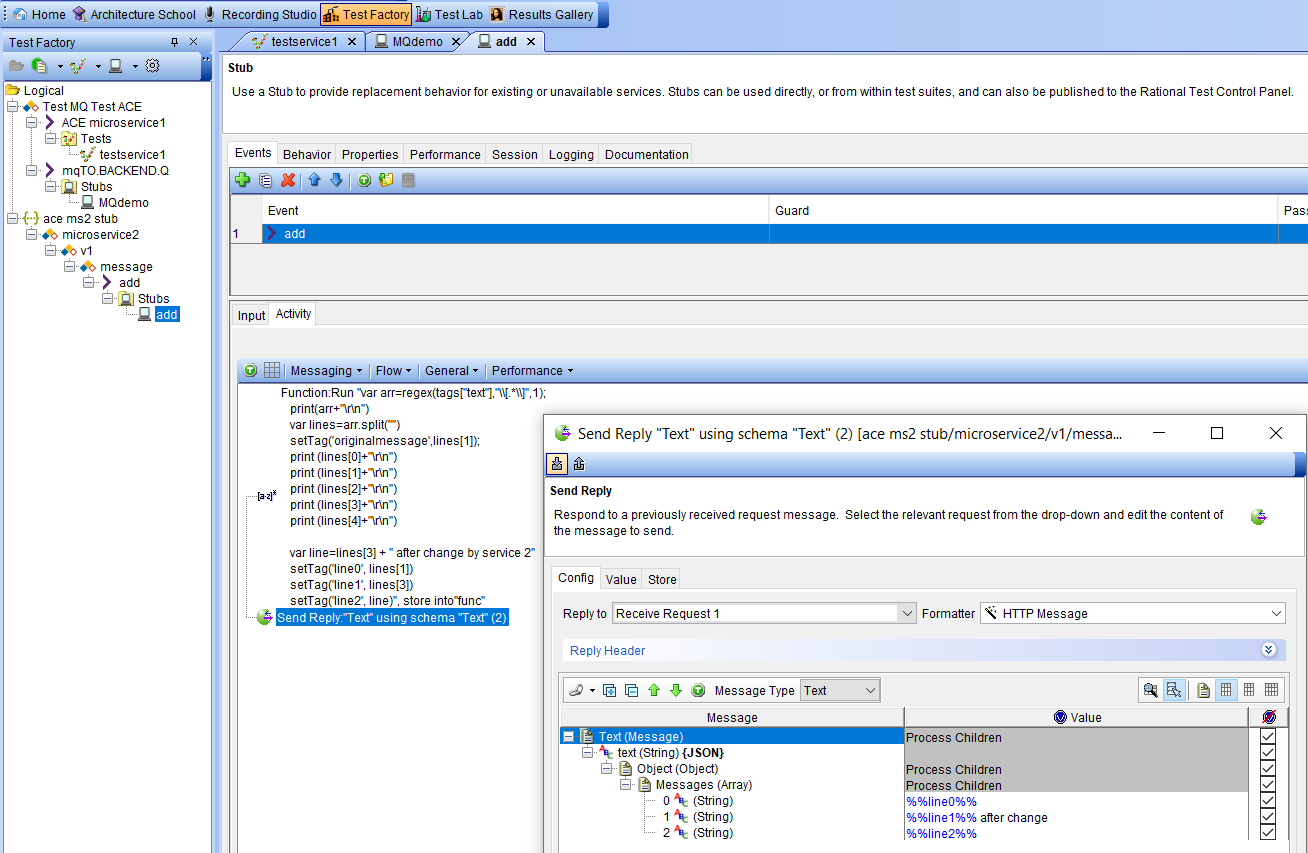
Defines the expected result – for comparison purposes



### Test Factory – Stubs

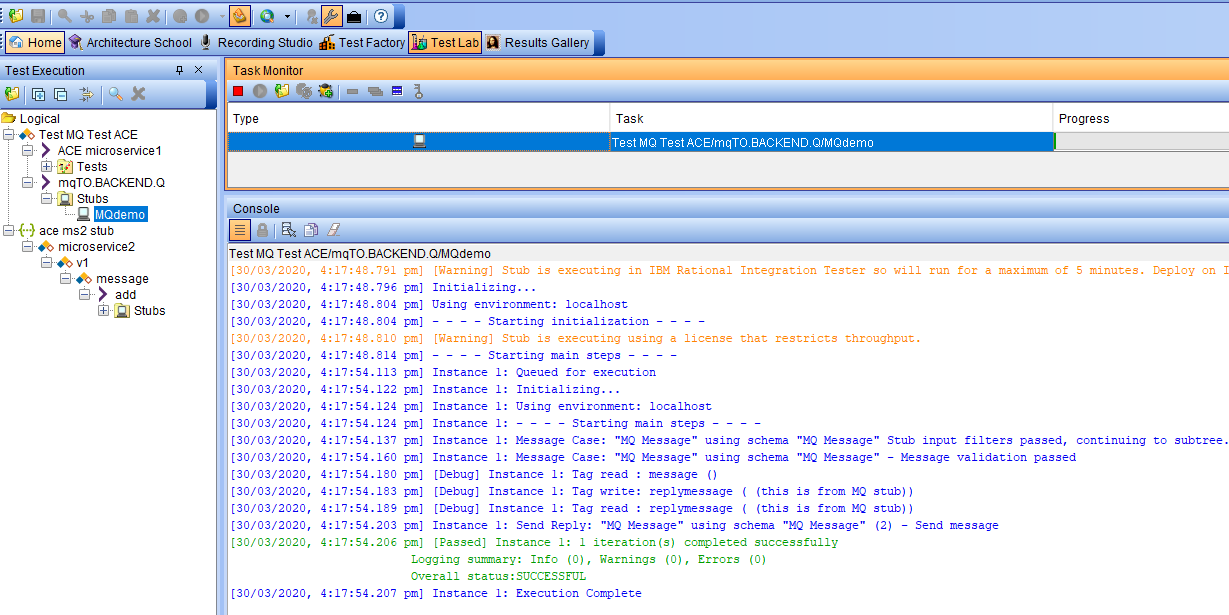
#### Explore the ACE MS2 Stub



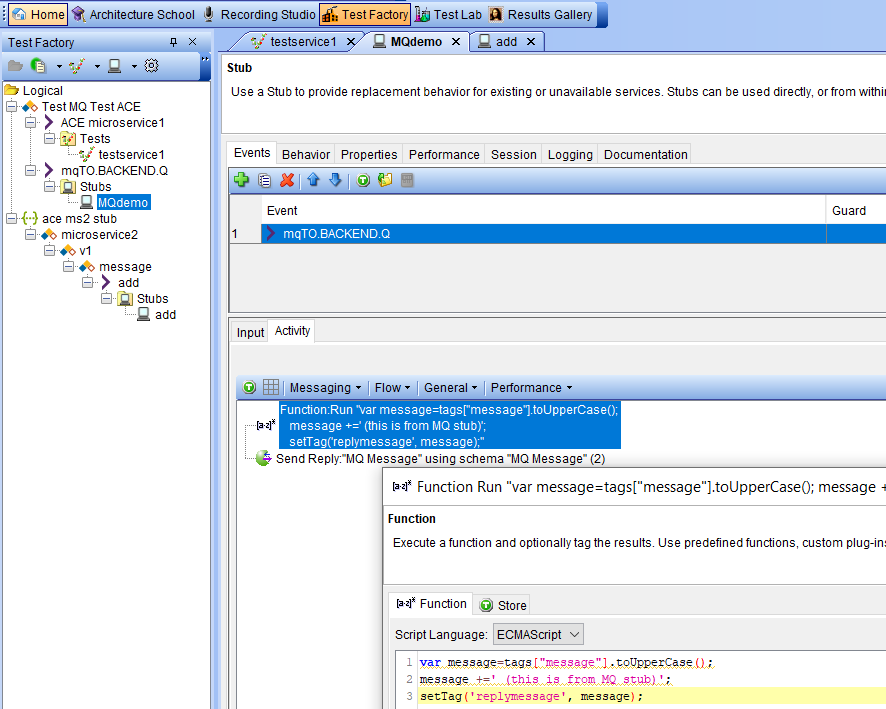


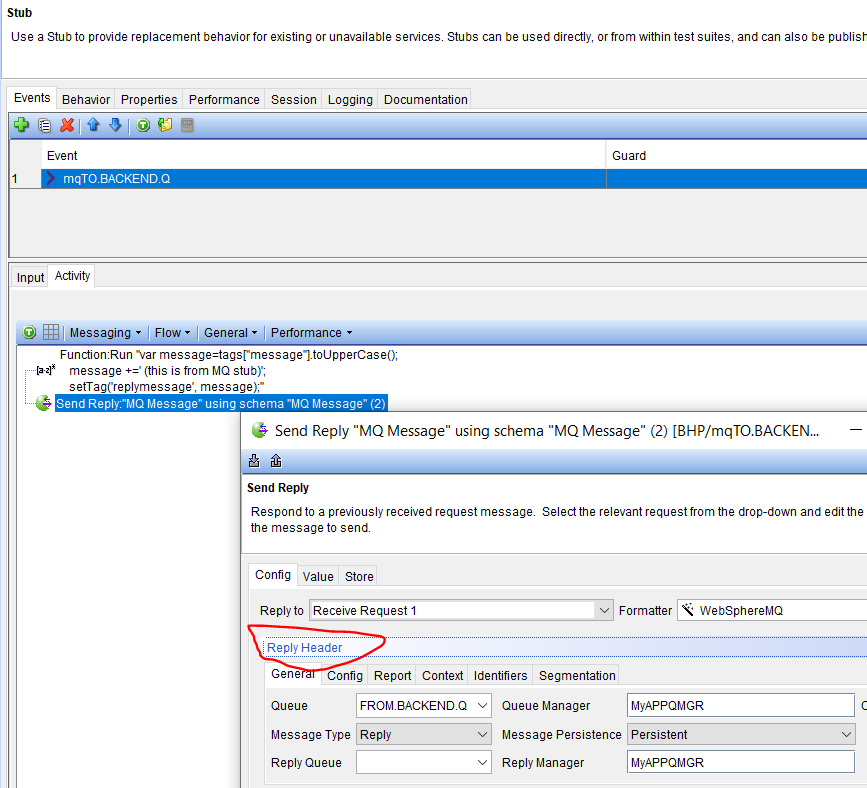
### Explore the Test Lab

In the Test Lab you can observer running tests and stubs



#### Explore the MQ Stub

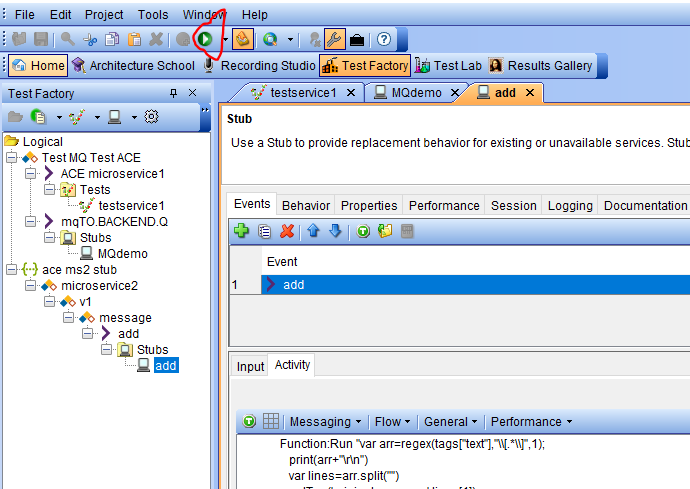


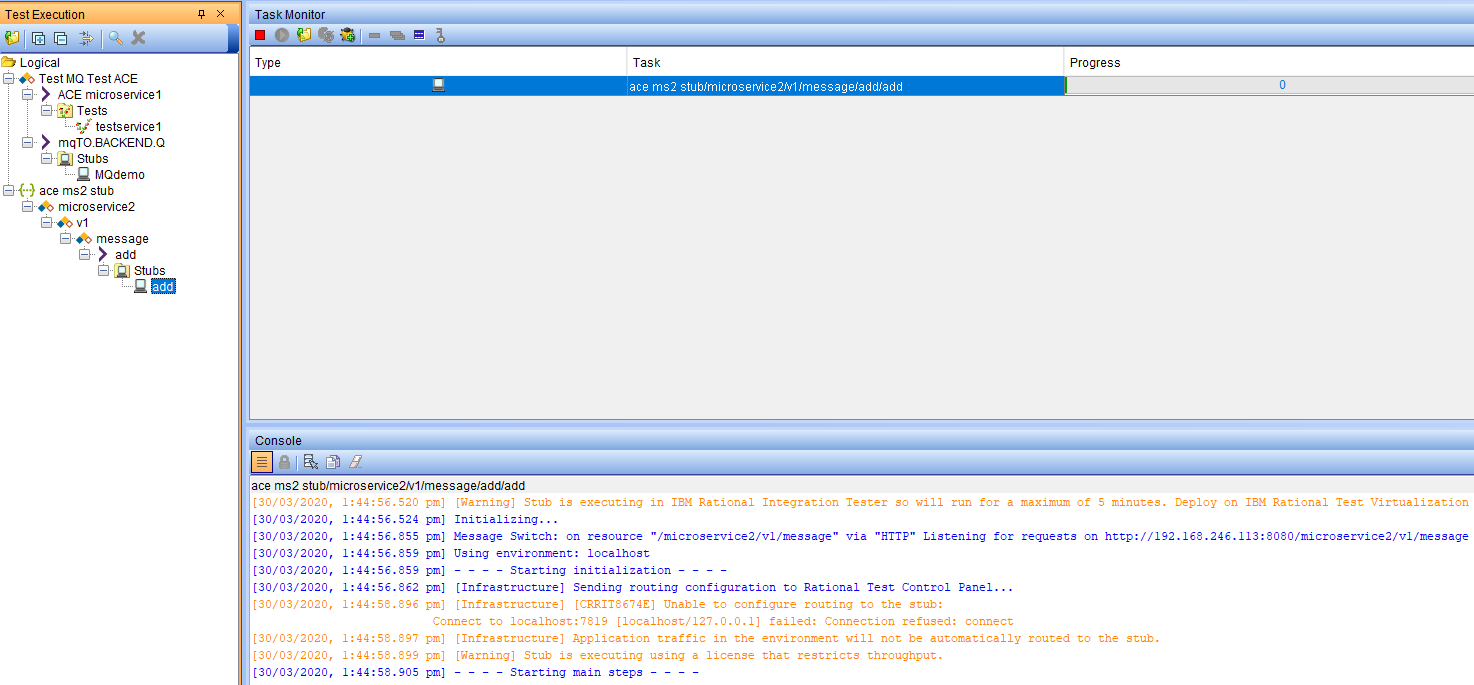


## Testing ACE Micro Service 2 Stub on IBM (Rational) Integration Tester – local client

### Run the ACE Micro Service 2 stub

The stub will stay active for 5 minutes





### Test ACE Micro Service 2 stub with a local REST client

POST

http://localhost:8080/microservice2/v1/message

with data

{"Messages":["Hello From Test Client 1111"]}

should return

{

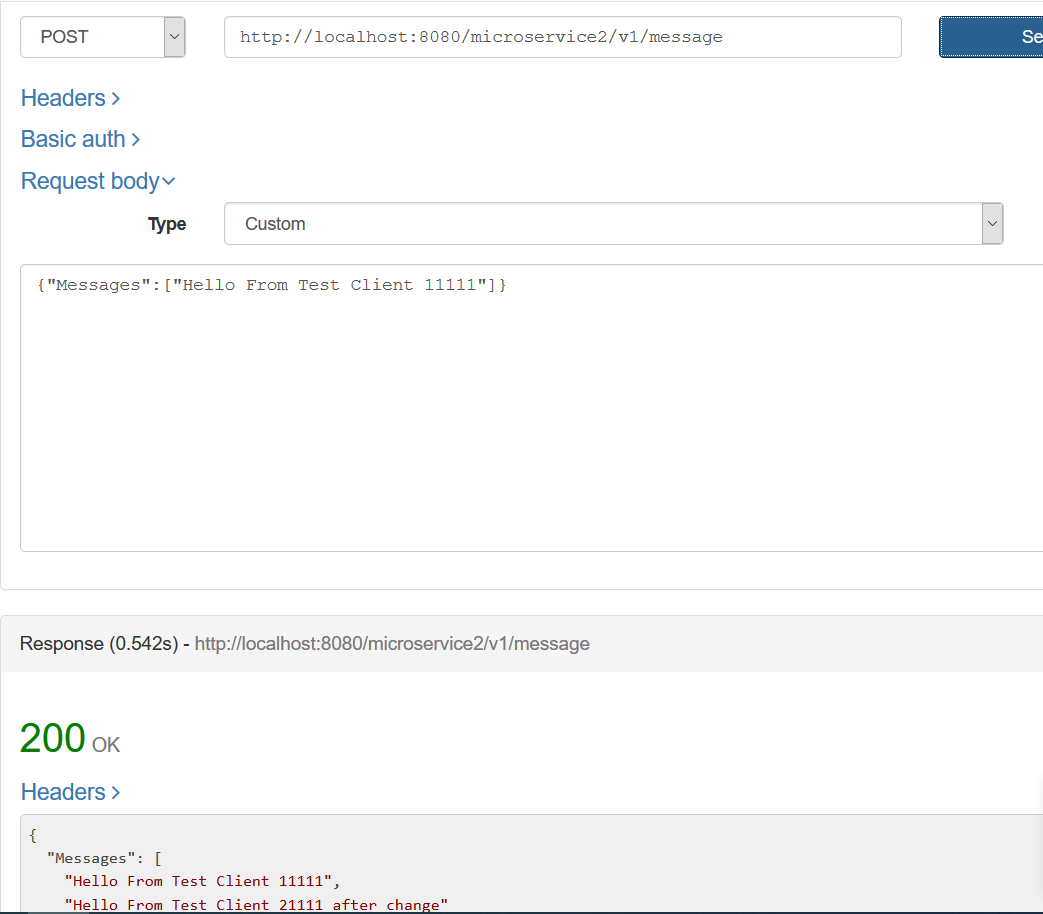
"Messages": [

"Hello From Test Client 1111",

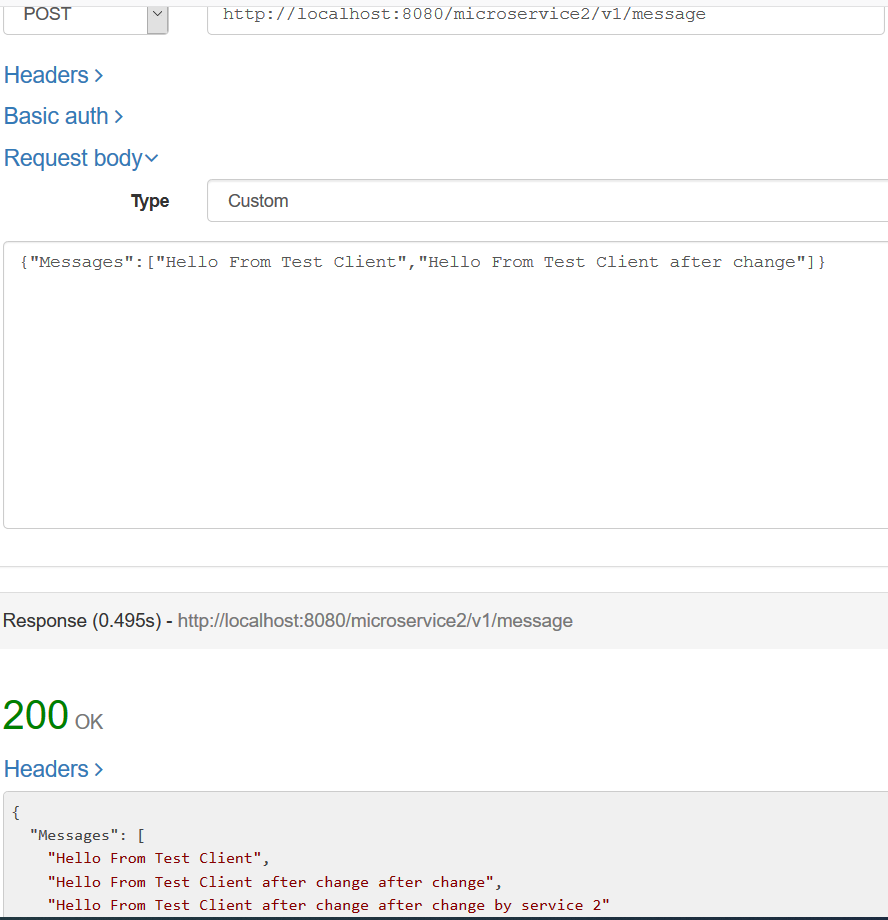
"Hello From Test Client 2111 after change"

]

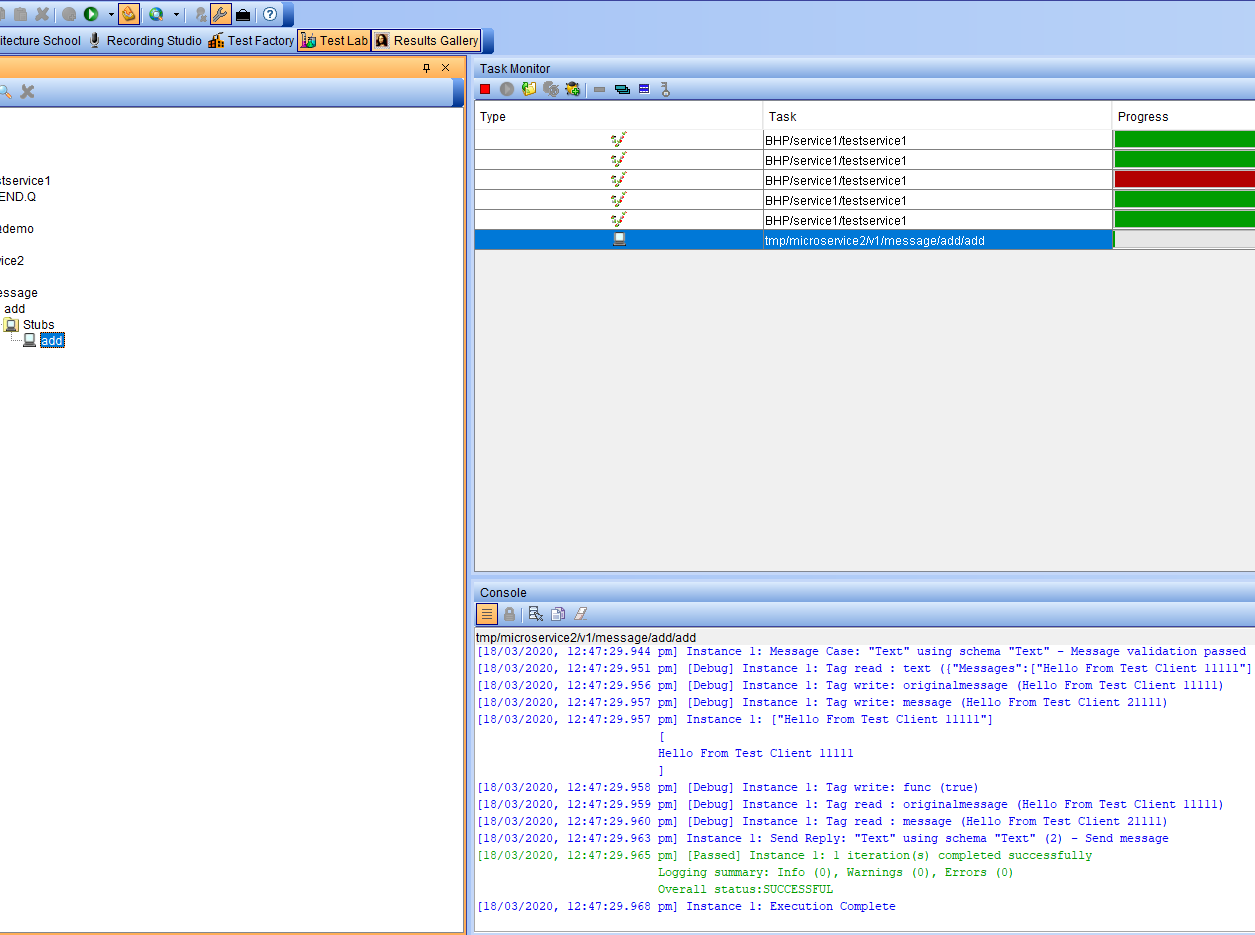
}



A second example

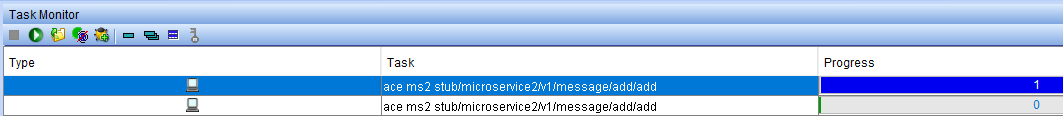


Observer the console output

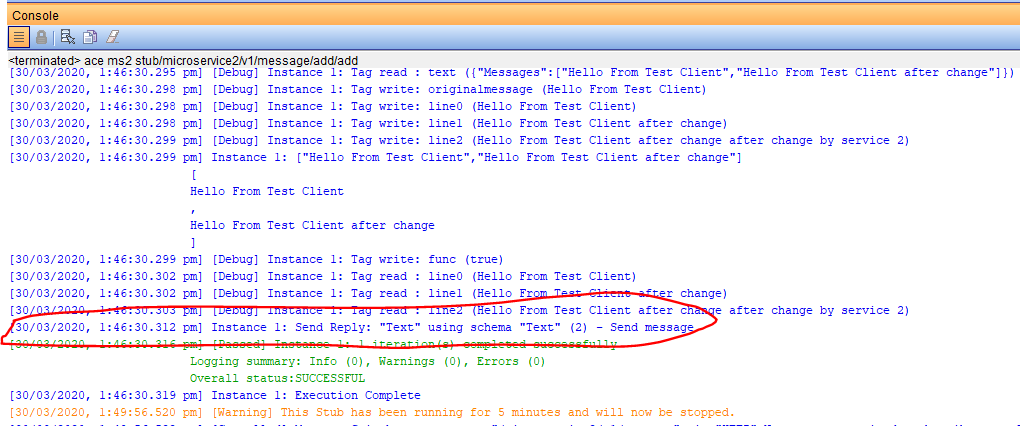


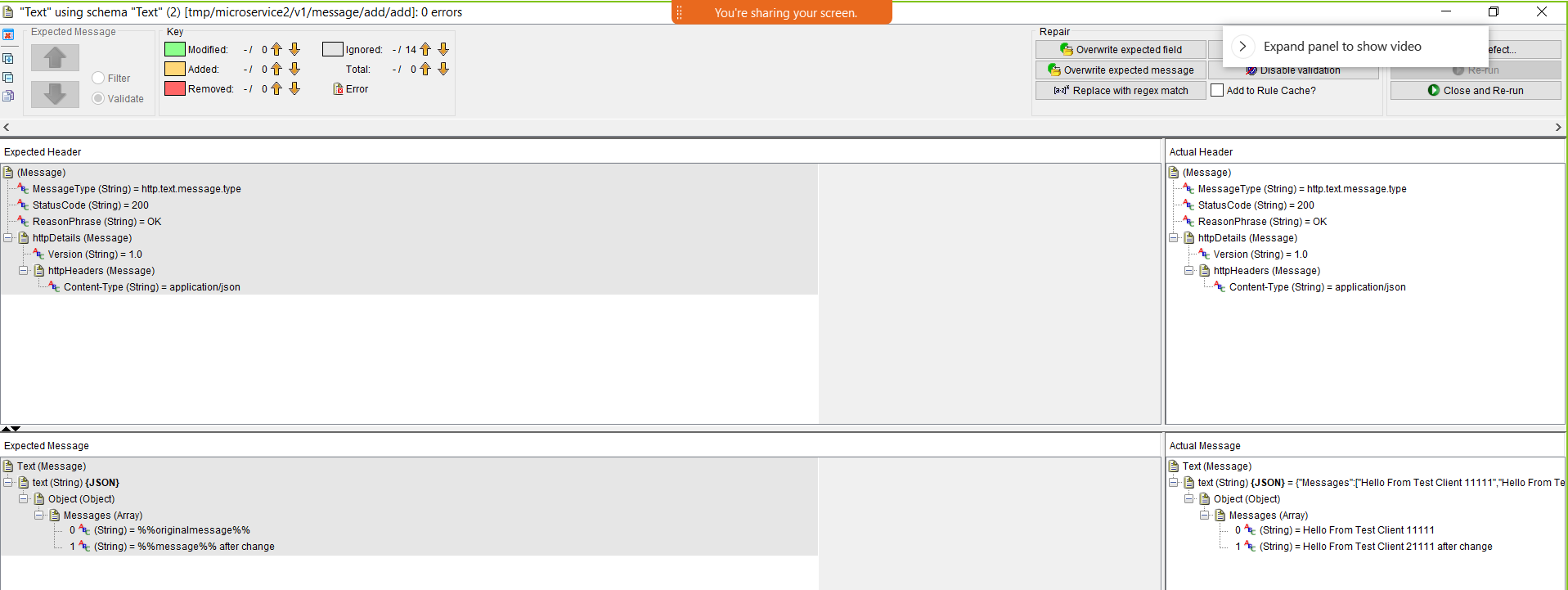
#### Check the results

Select the Task for the test in the Task monitor

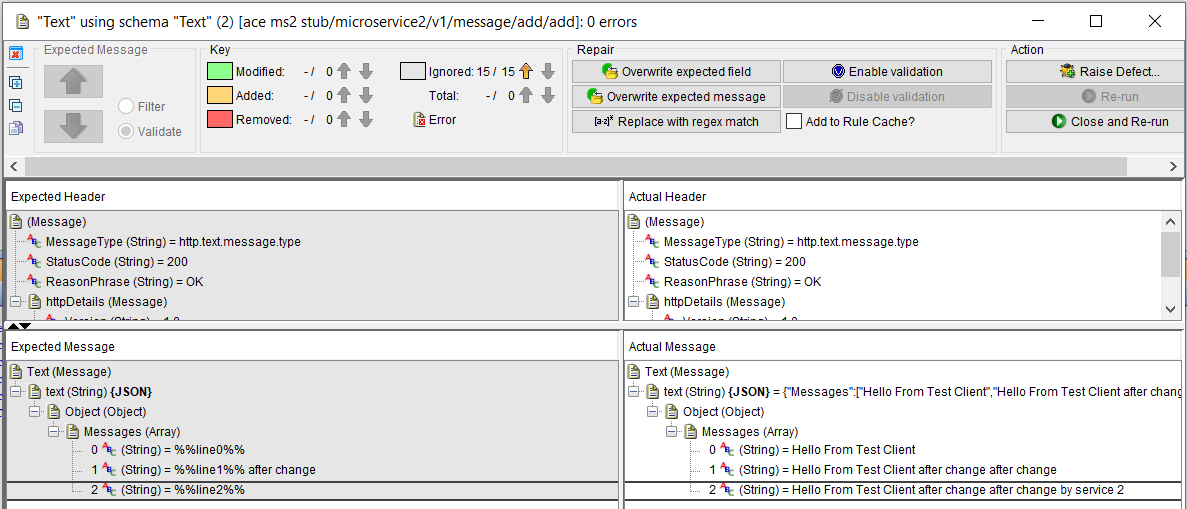


Click on the send reply line in the console



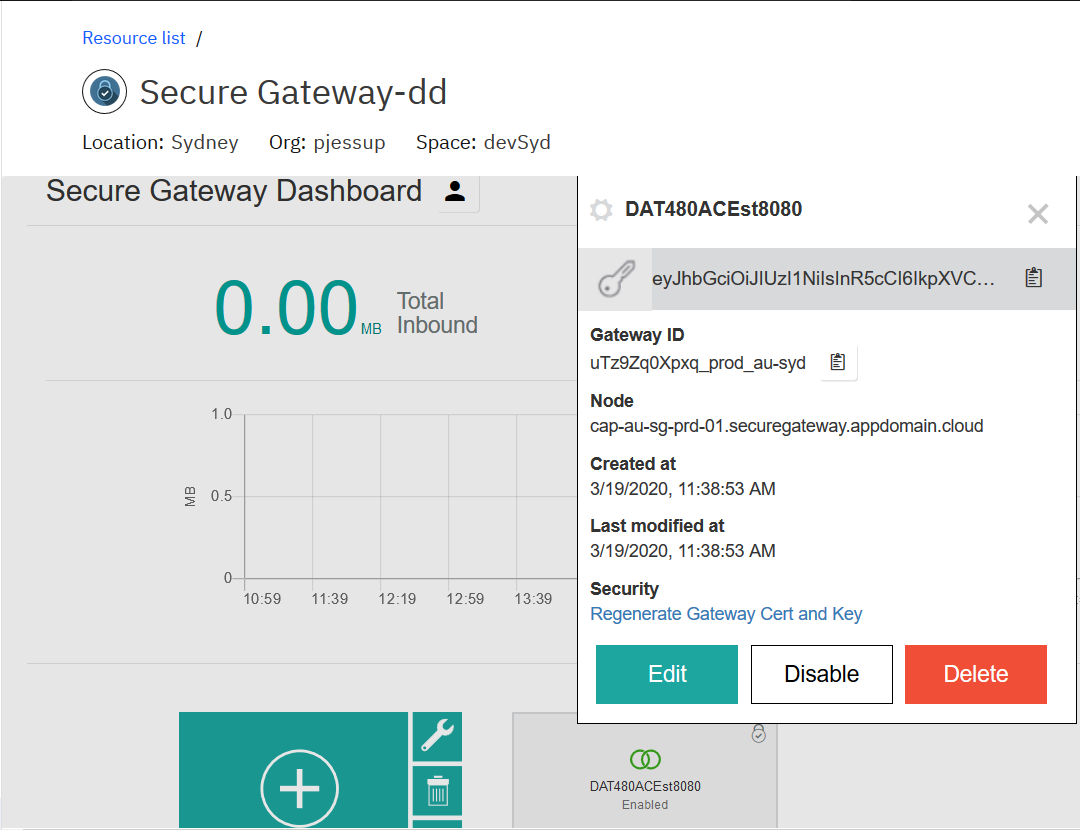


Example 2 results



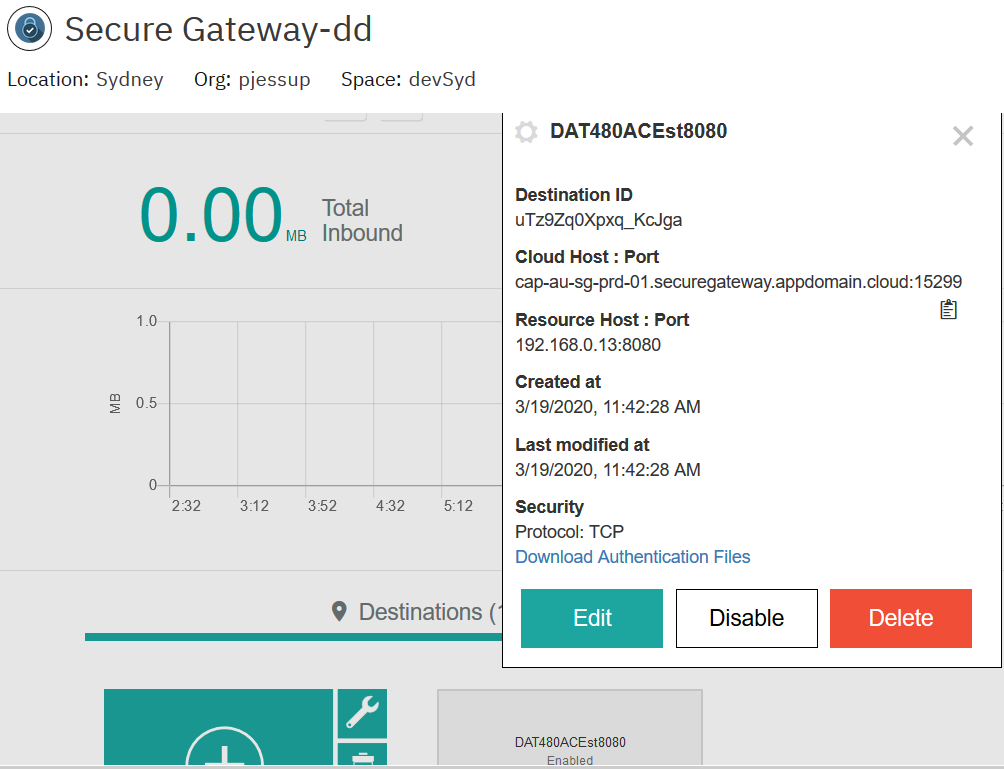
## Testing ACE Micro Service 2 Stub on IBM (Rational) Integration Tester – Public IP

### IBM Secure Gateway Service – Server side



Copy the gateway ID and token for use in the client side set up

Open the destination and copy the public IP address and port number



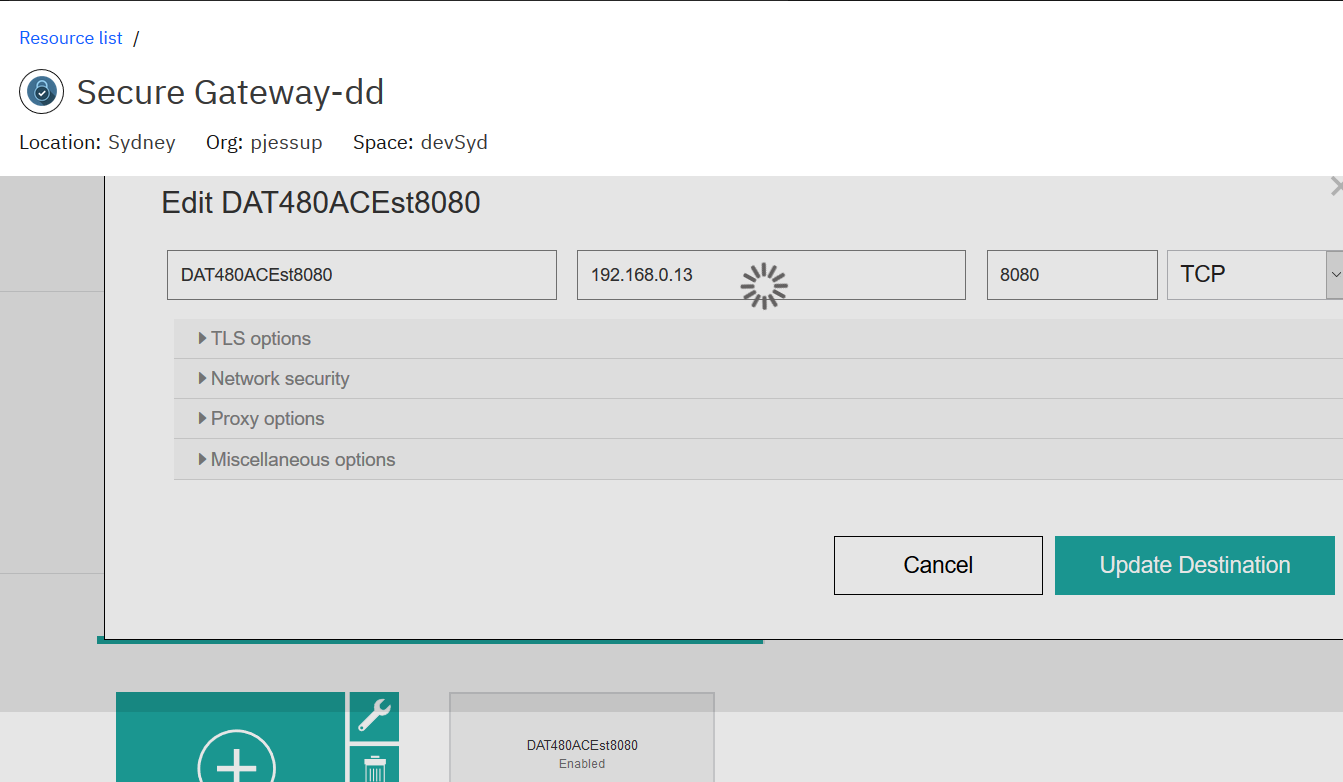
cap-au-sg-prd-01.securegateway.appdomain.cloud:15299

### IBM Secure Gateway Client side

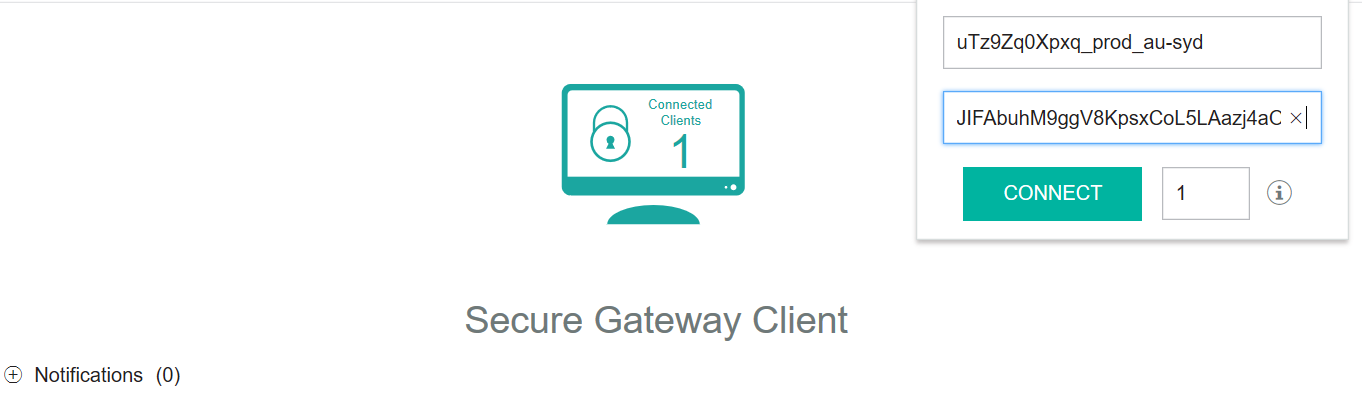
You need the IP address that your machine is using at the time could be wifi adapter, Bluetooth or IBM 9. Address – depends + plus the port. In this example my queue manager is listening on 1515 and my RIT RSVT stub for ACE MS2 is listening on 8080

192…. Is my wifi adapter when at home.

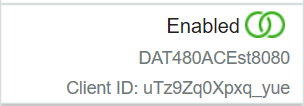
172 … I my Bluetooth when connected on the phone



Use the copied gateway Id and token

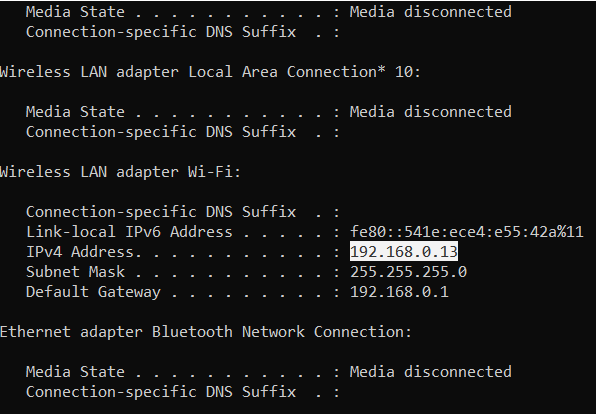


Click connect

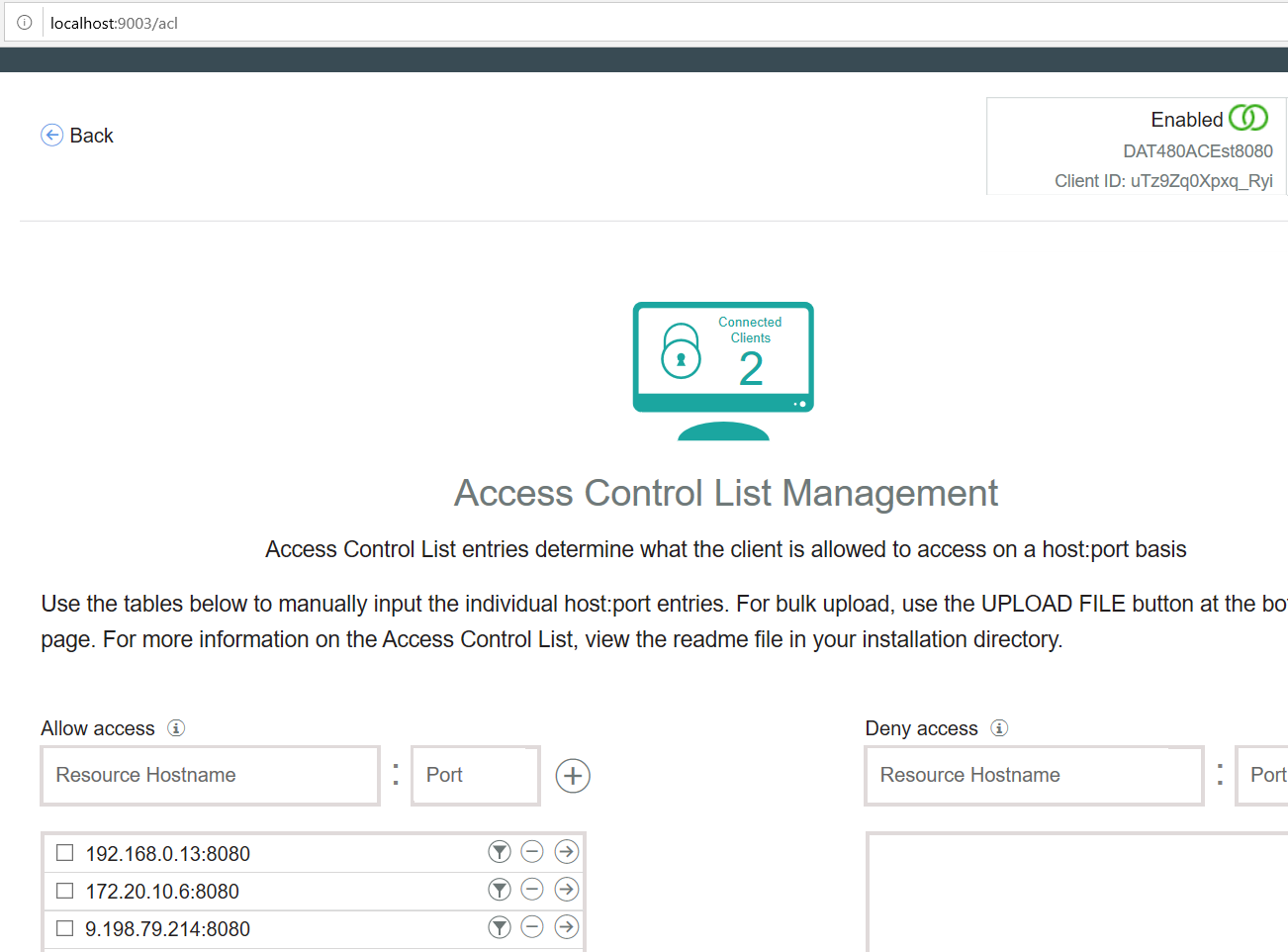


Set up the ACLs for inbound connection

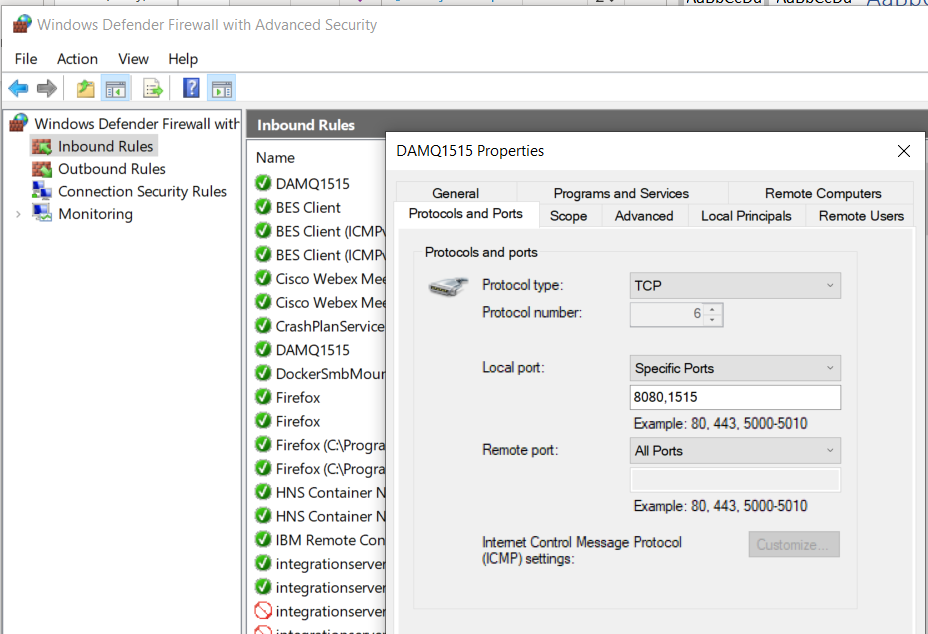
Type IP config on your machine/laptop to find the IP address of the currently connected adapter



Config the ACL in the Secure Gateway Client console

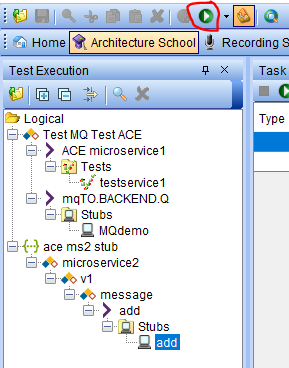


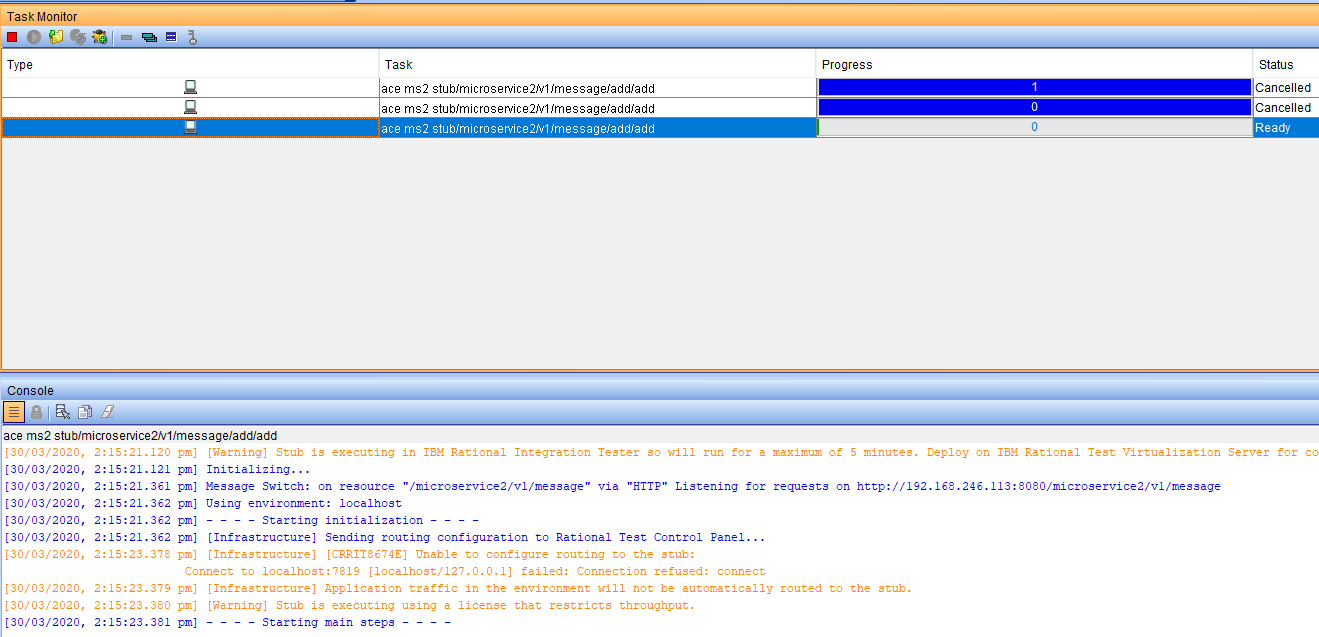
### Windows firewall



### Test calling the RSVT ACE MS2 stub via public IP

Start the ACE MS 2 stub in IBM Integration Tester





Using a REST Client POST to

http:// cap-au-sg-prd-01.securegateway.appdomain.cloud:15299/microservice2/v1/message

with data

{"Messages":["Hello From Test Client 1111"],["Hello From Test Client 2222"]}

should return

{

"Messages": [

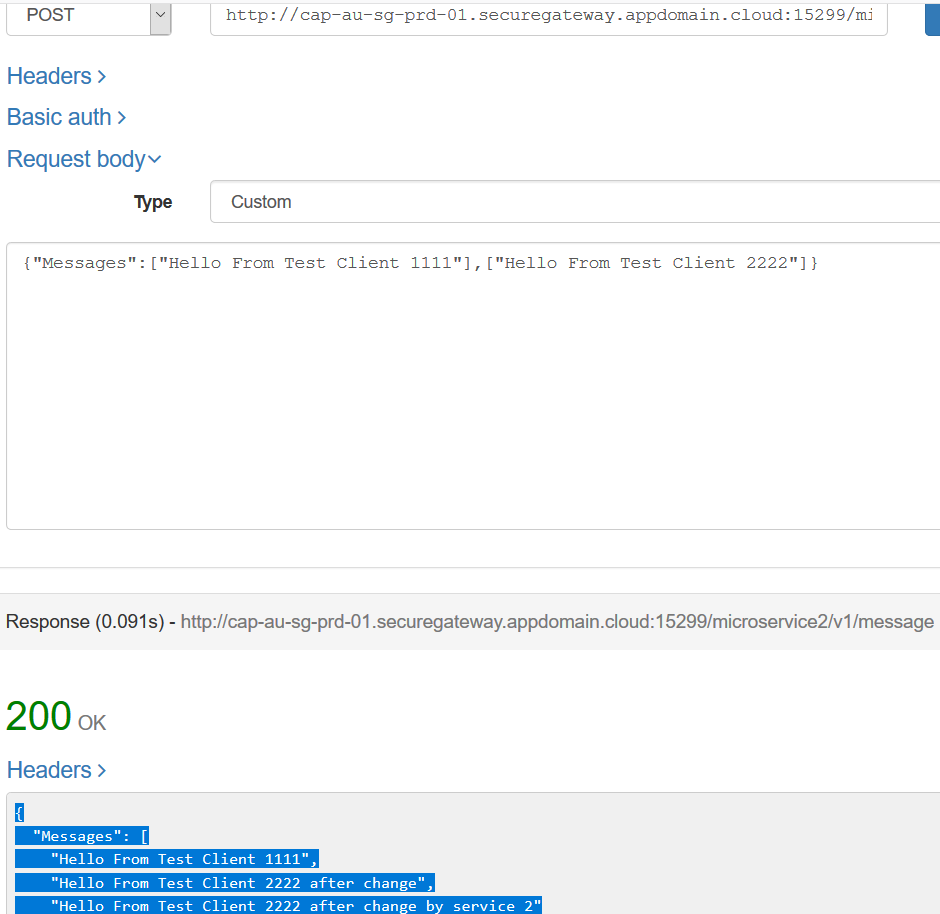
"Hello From Test Client 1111",

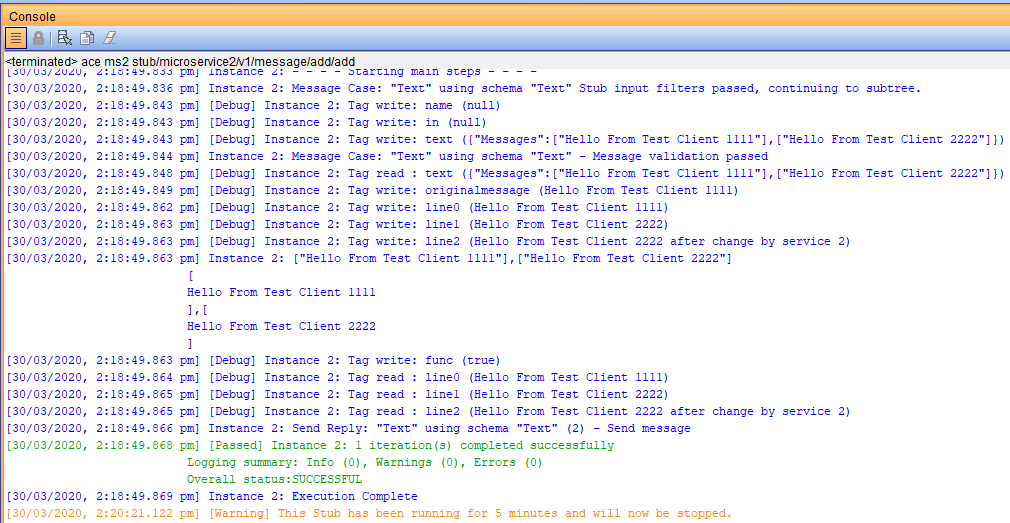
"Hello From Test Client 2222 after change",

"Hello From Test Client 2222 after change by service 2"

]

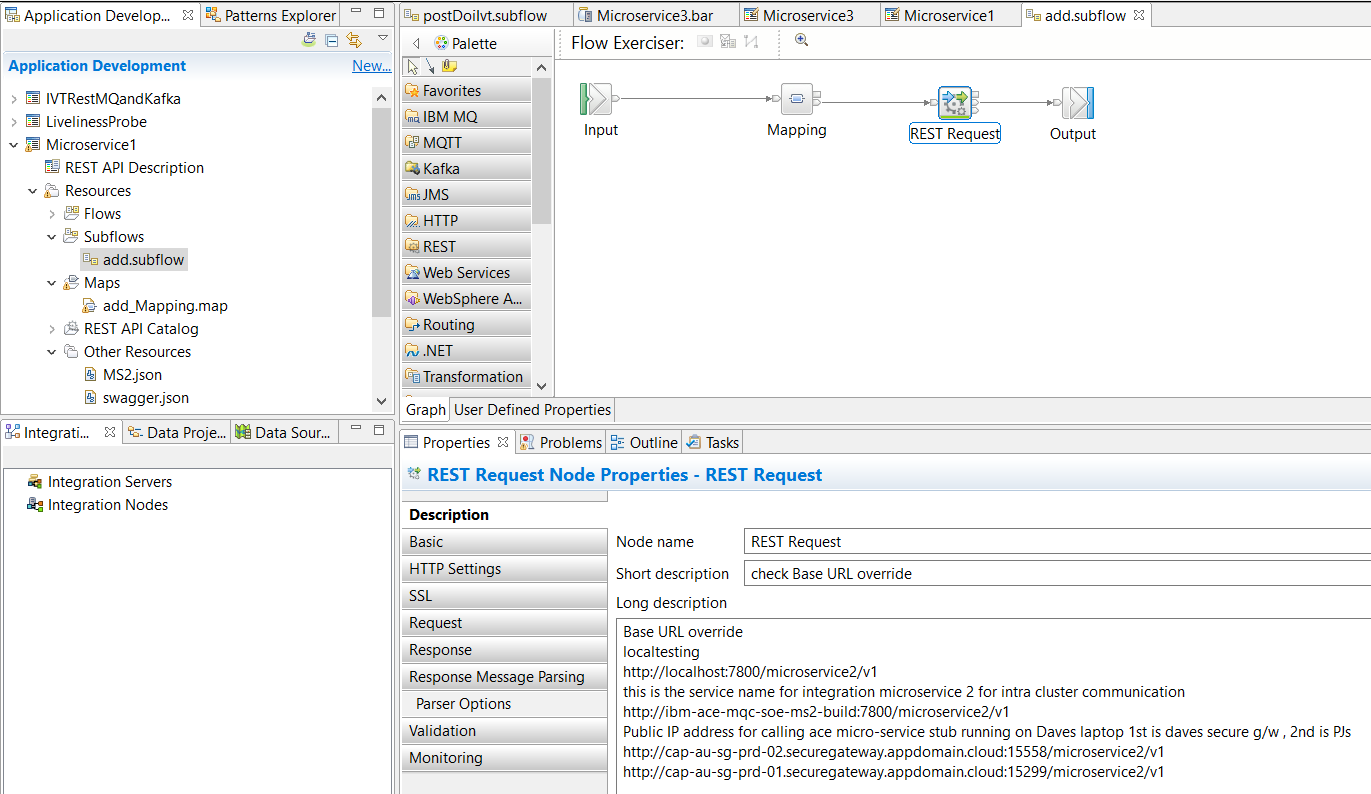
}





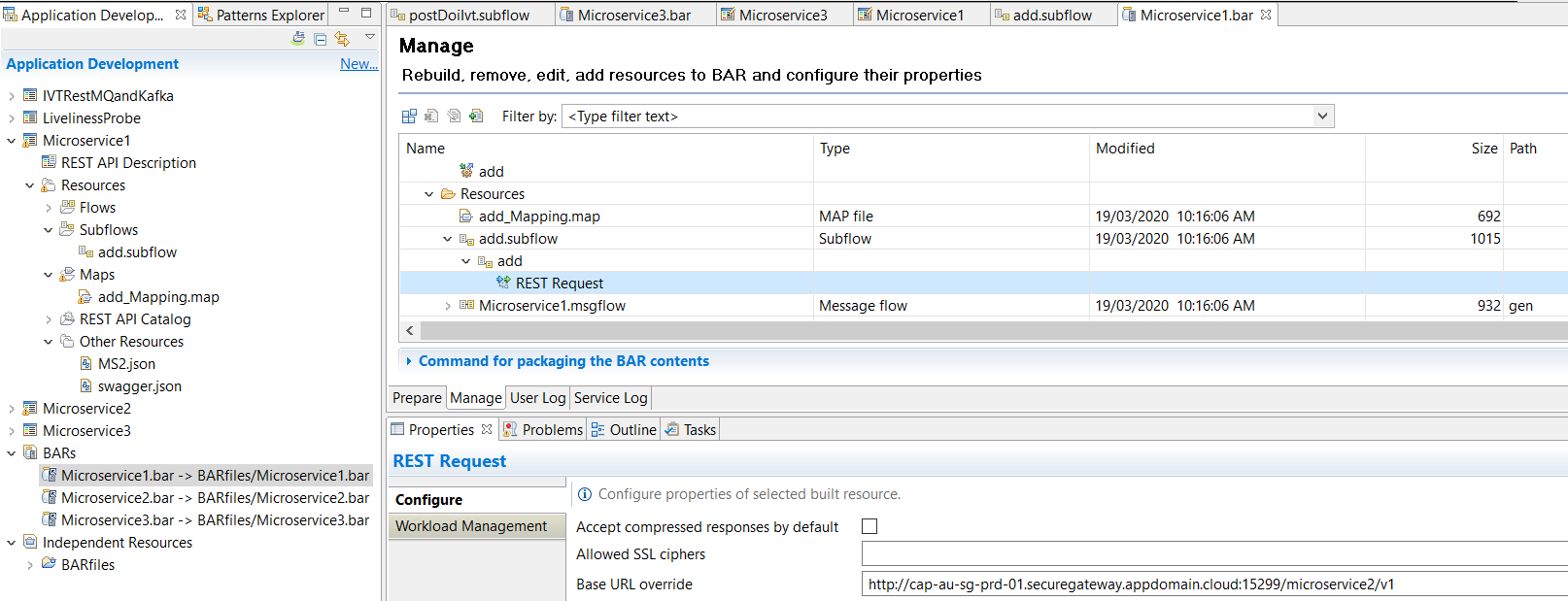
## Calling ACE MS1 via IBM Rational Integration Tester – ACE MS1 calls ACE MS2 stub

For this test ACE MS1 RestRequest node need its URL to call ACE MS2 updated to target the ACE MS2 stub on IBM Rational Integration Tester via the public IP address offered by the IBM Secure Gateway service.



http://cap-au-sg-prd-01.securegateway.appdomain.cloud:15299/microservice2/v1

Rather than changing the source. User the BAR override parms on the BAR file itself



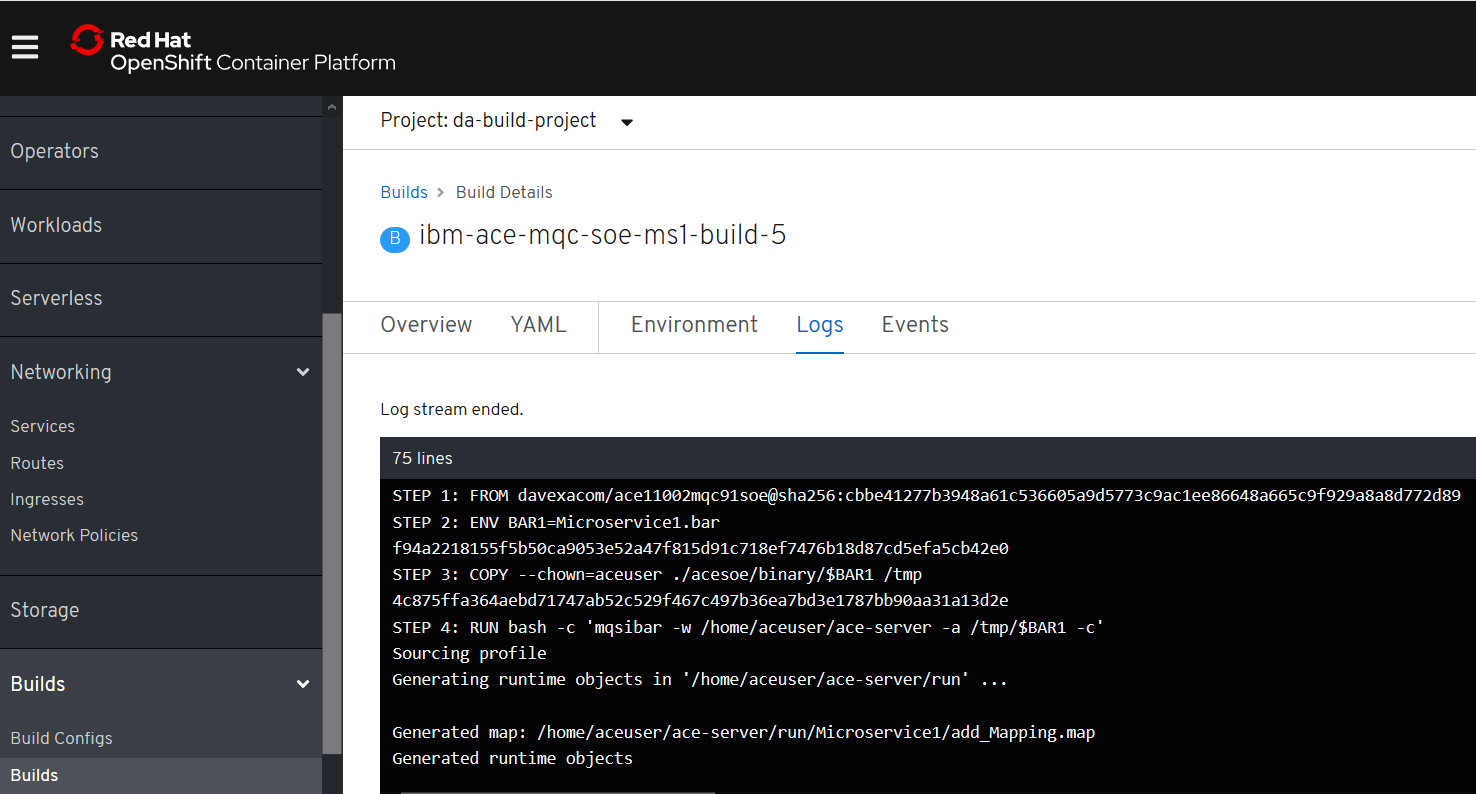
Next, the image on RH OpenShift will need to be replace with a new version containing the updated BAR. The document “**1.Developer Experience for ACE MQ with RHOS Tools and Tekton v1.1**”

Section: **Build and Run ACE Microservice 1 from SoE on RH OpenShift**

Page : **49**

Describes the steps if you have not already set up ACE MS1. The steps are

1. Save the BAR file
2. Push it to the GIT repos
3. Webhook fires and RHOS will build and deploy a new container with the updated BAR file



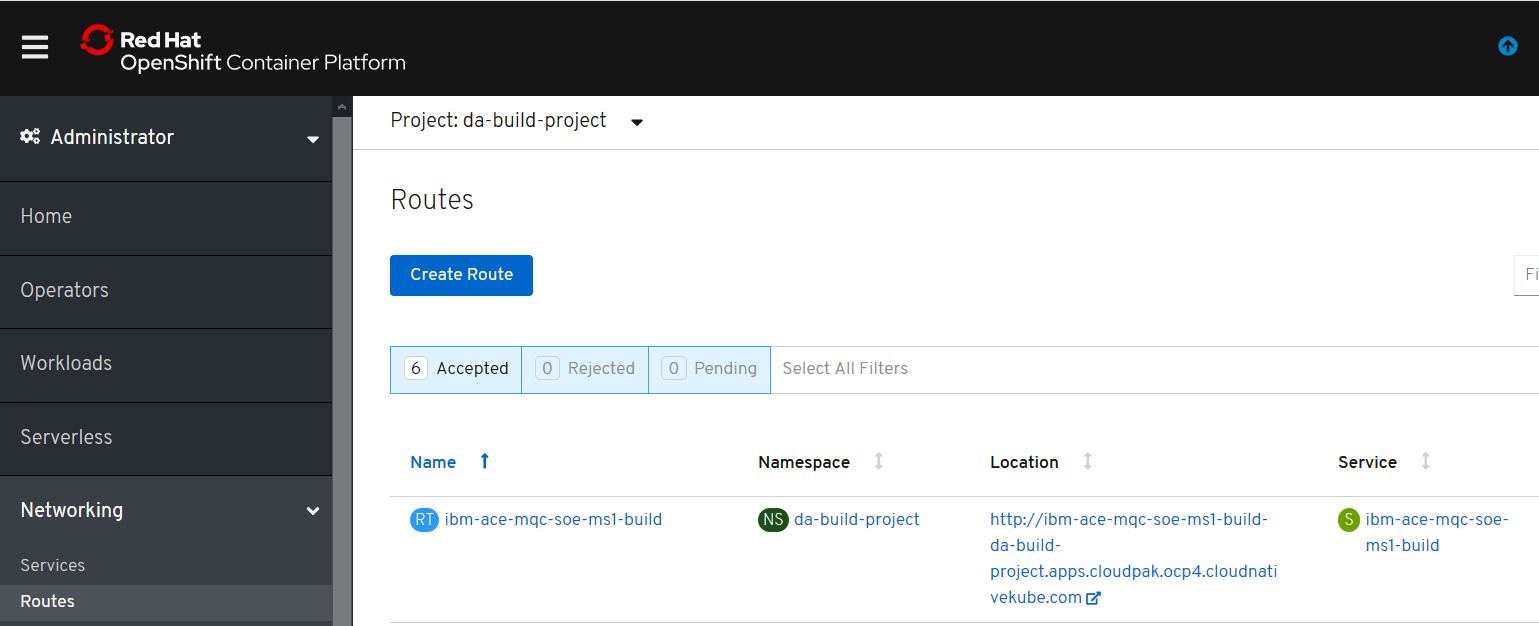
Check the logs on the triggered build/rebuild

RH OpenShift console->Builds->builds->ibm-ace-mqcsoe-ms1-build-n

ACE MS1 on RH OpenShift is now pointing (via the secure gateway) to the stub of ACE MS2 on IBM Rational Integration Tester.

### Get the ACE MS1 route details from RH OpenShift console

RH OpenShift Console->Networking->routes->ibm-ace-mqc-soe-m1-build

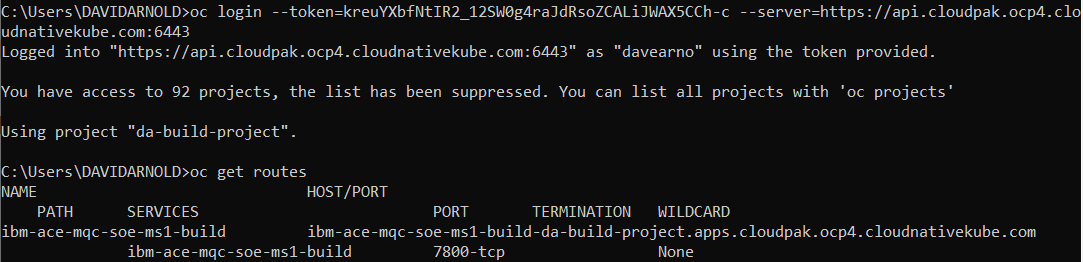


Or use

Oc get routes

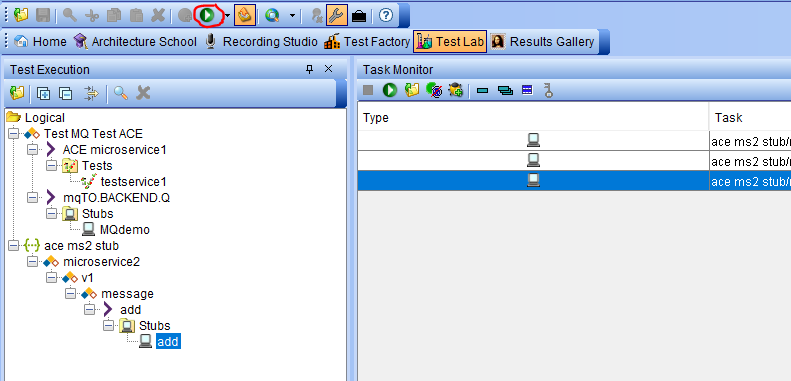
To retrieve the target hostname:

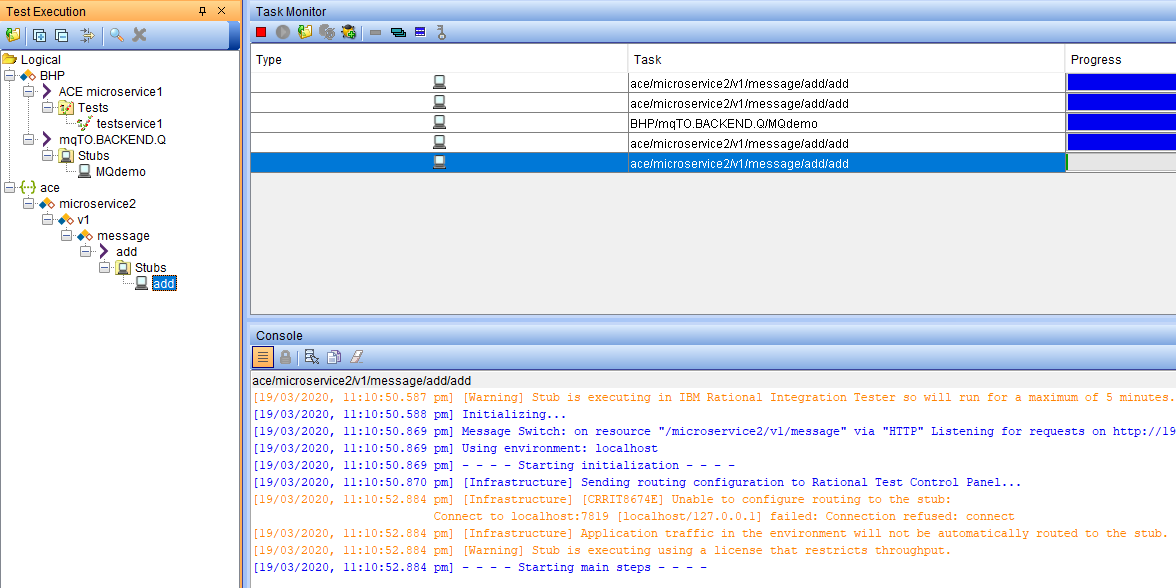
ibm-ace-mqc-soe-ms1-build-da-build-project.apps.cloudpak.ocp4.cloudnativekube.com



Add the ACE MS 1 base URL to the hostname and then using a REST Client to ensure you have connectivity to ACE MS 1 and the ACE MS2 stub it is set up to call.

### Start the IBM Rational Integration Tester ACE MS2 Stub





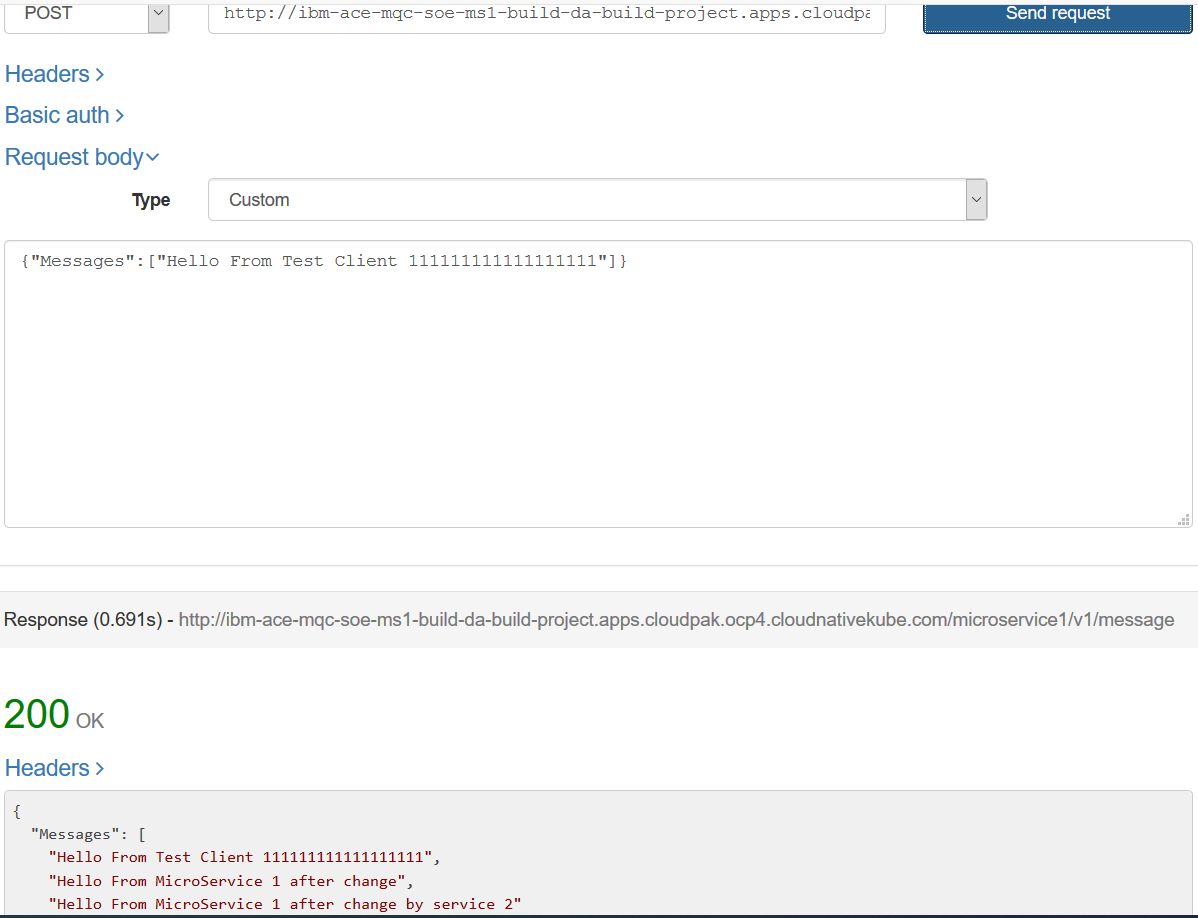
### Invoke the REST client to call ACE MS1

Use a REST Client to POST

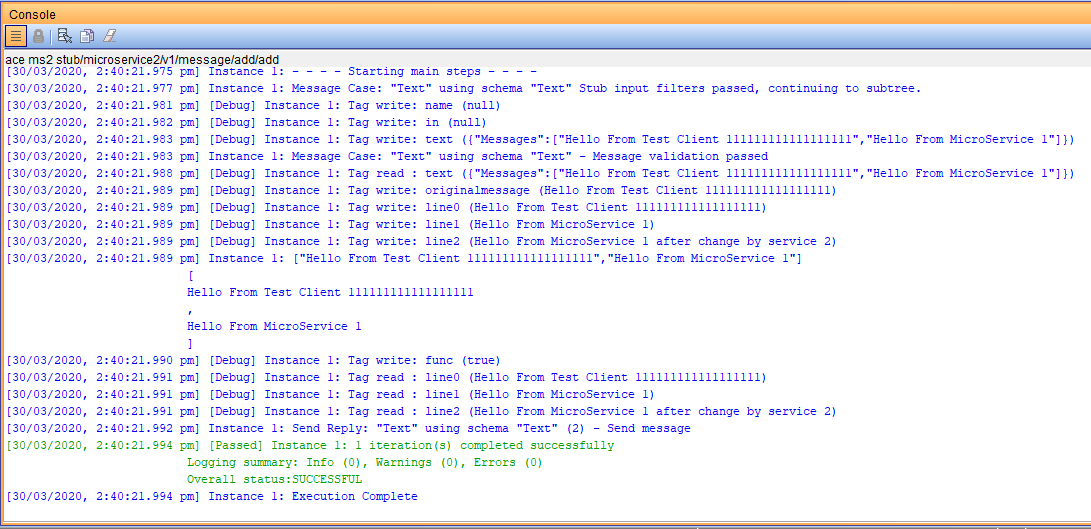
<http://ibm-ace-mqc-soe-ms1-build-da-build-project.apps.cloudpak.ocp4.cloudnativekube.com/microservice1/v1/message>

with Data

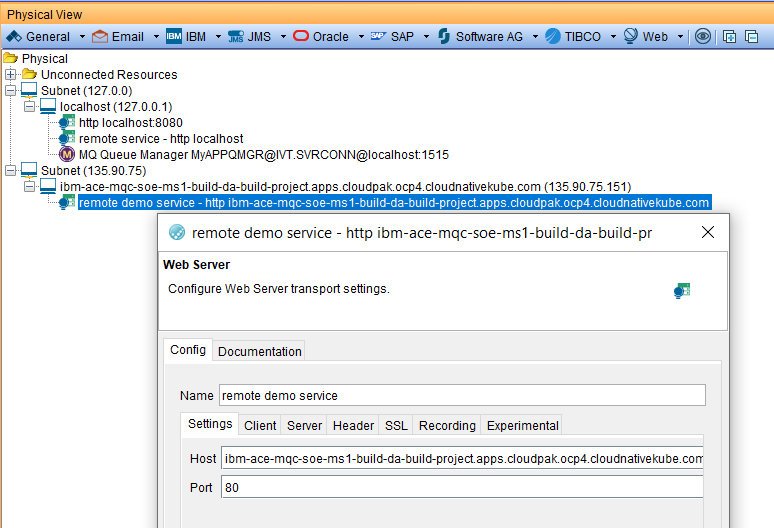
{"Messages":["Hello From Test Client 111111111111111111"]}



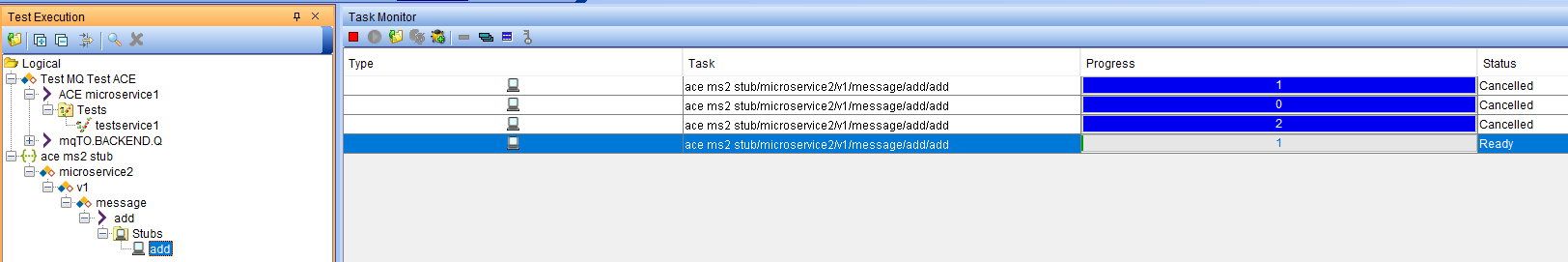
### Observe results in ACE MS 2 stub console



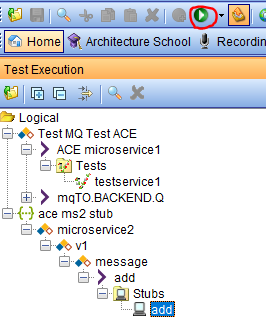
### Invoke ACE MS1 on RH OpenShift from a IBM Rational Integration Tester



Ensure the ACE MS 2 stub is still running

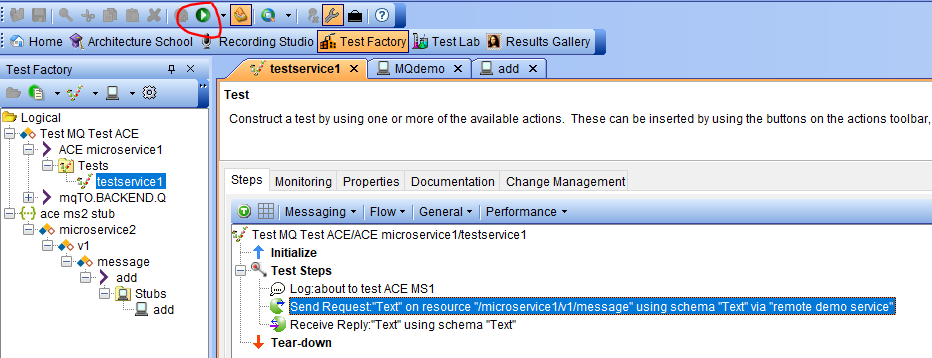


If not start a new test run

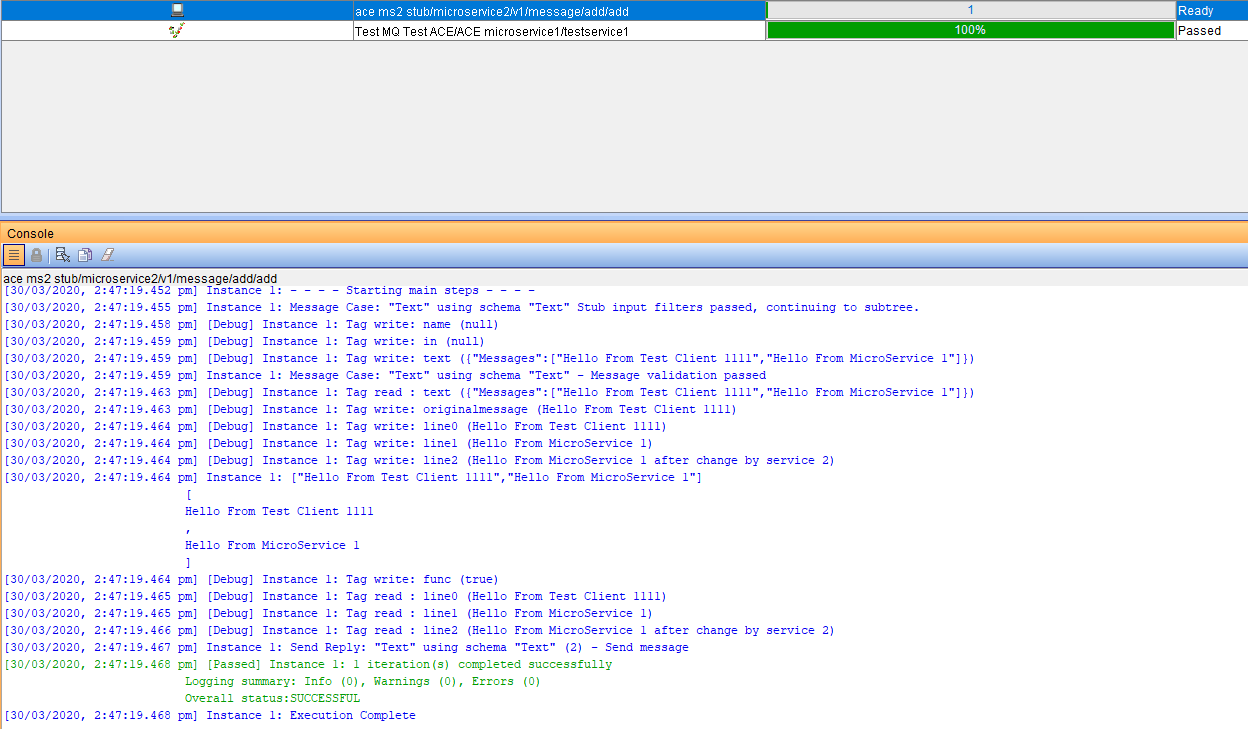


In the test factory select Logical->”Test MQ Test ACE”->ACE microservice1->tests->testservice1

And hit the Run button

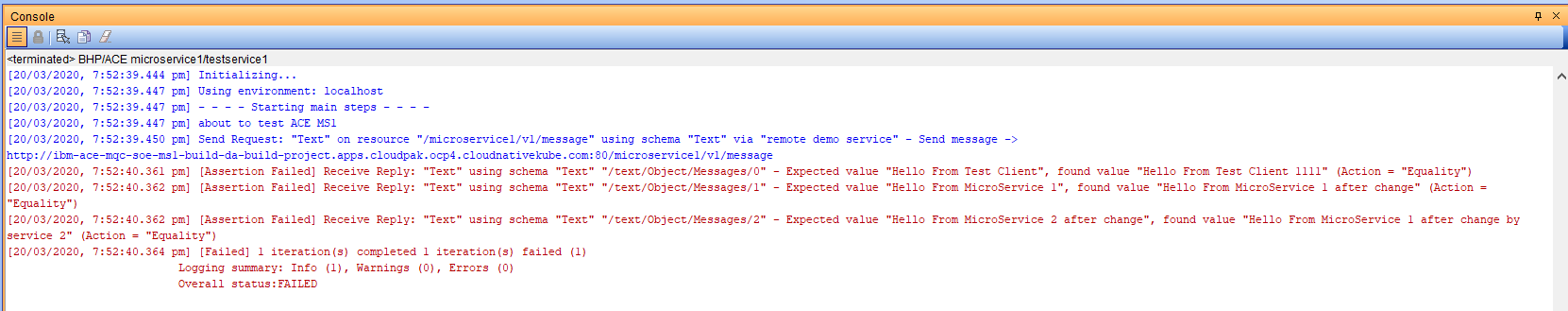


### Observe the results in the stub

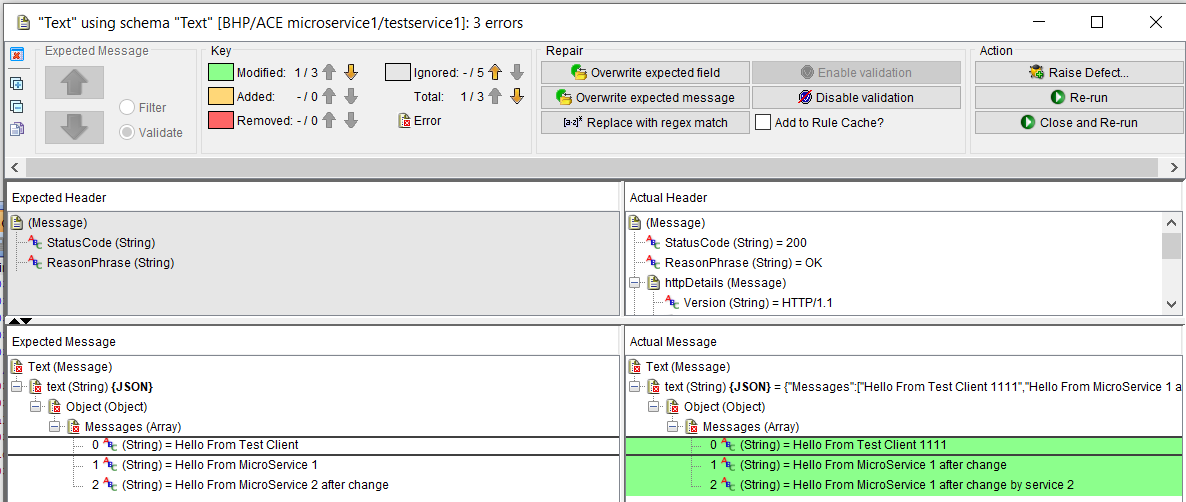


### Observe results in the test console

#### An example of a working test with incorrect data (unmatching data)

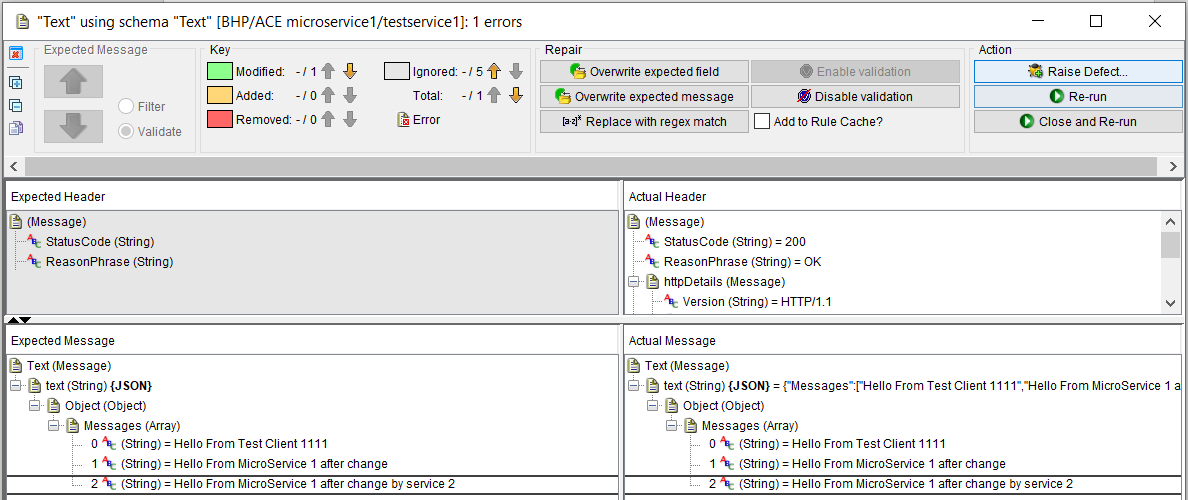


The test actually worked but we can compare the actual and expected

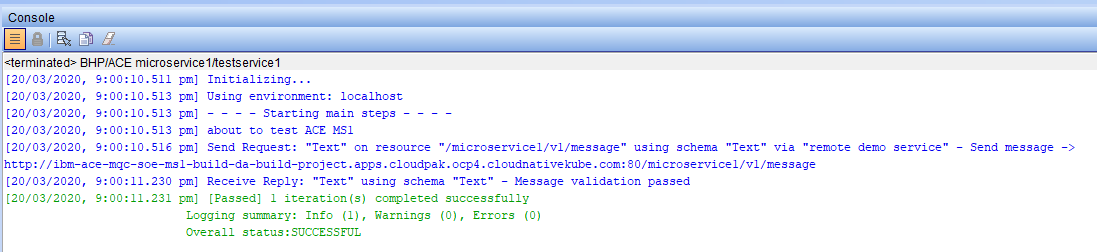


Note the semantics are correct but the data is different

Right click on the right and select overwrite expected

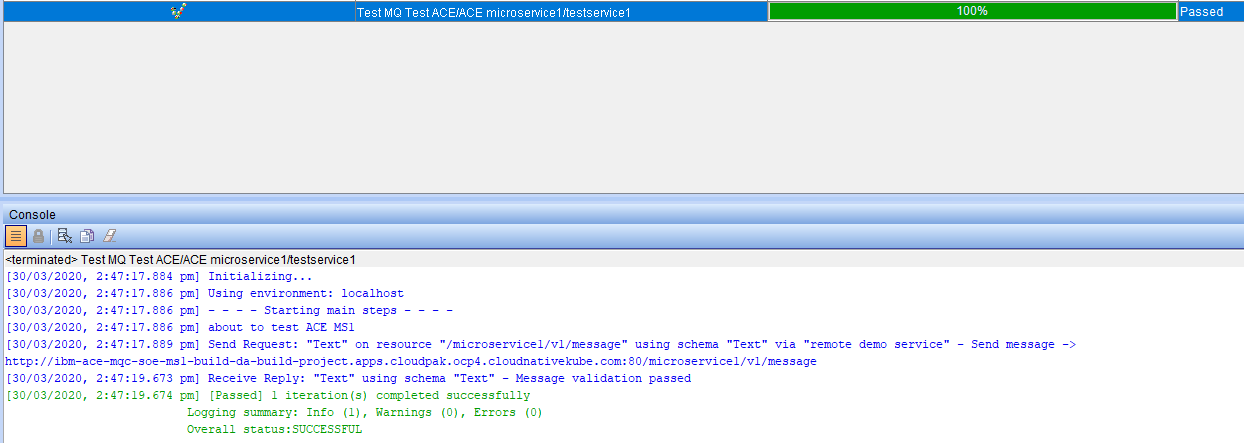


Re-run the test





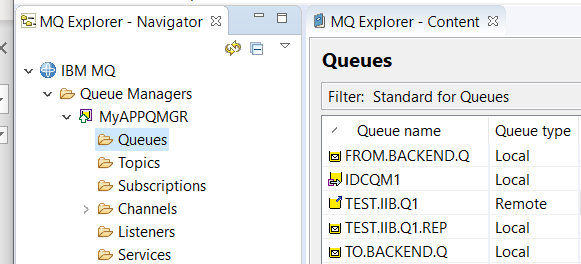
#### An example of a working test with correct data (matching data)



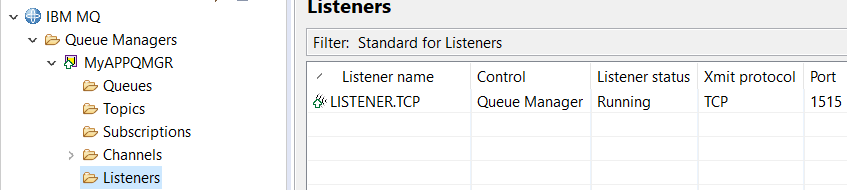
## Using ACE Micro Service 3 (MS3) with a remote MQ Queue Manager

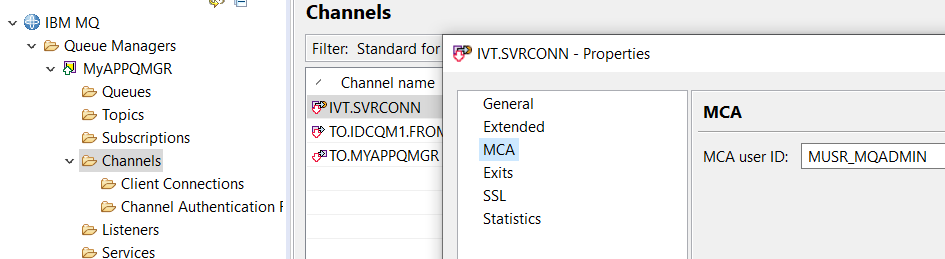
### Configure Queue Manager MyAPPQMGR on the IBM Integration Tester Machine

#### Queue Manager and Queues



#### MQ Listener





### MQ SVRCONN Channel creation for Linux based queue manager

DEFINE CHANNEL(IVT.SVRCONN) CHLTYPE(SVRCONN) REPLACE

SET CHLAUTH(IVT.SVRCONN) TYPE(BLOCKUSER) USERLIST(nobody)

ALTER AUTHINFO(SYSTEM.DEFAULT.AUTHINFO.IDPWOS) AUTHTYPE(IDPWOS) CHCKCLNT(NONE) ADOPTCTX(YES)

SET CHLAUTH(IVT.SVRCONN) TYPE (ADDRESSMAP) ADDRESS(\*) MCAUSER('mqm')

REFRESH SECURITY TYPE(CONNAUTH)

### MQ SVRCONN Channel creation for windows based queue manager

DEFINE CHANNEL(IVT.SVRCONN) CHLTYPE(SVRCONN) REPLACE

SET CHLAUTH(IVT.SVRCONN) TYPE(BLOCKUSER) USERLIST(nobody)

ALTER AUTHINFO(SYSTEM.DEFAULT.AUTHINFO.IDPWOS) AUTHTYPE(IDPWOS) CHCKCLNT(NONE) ADOPTCTX(YES)

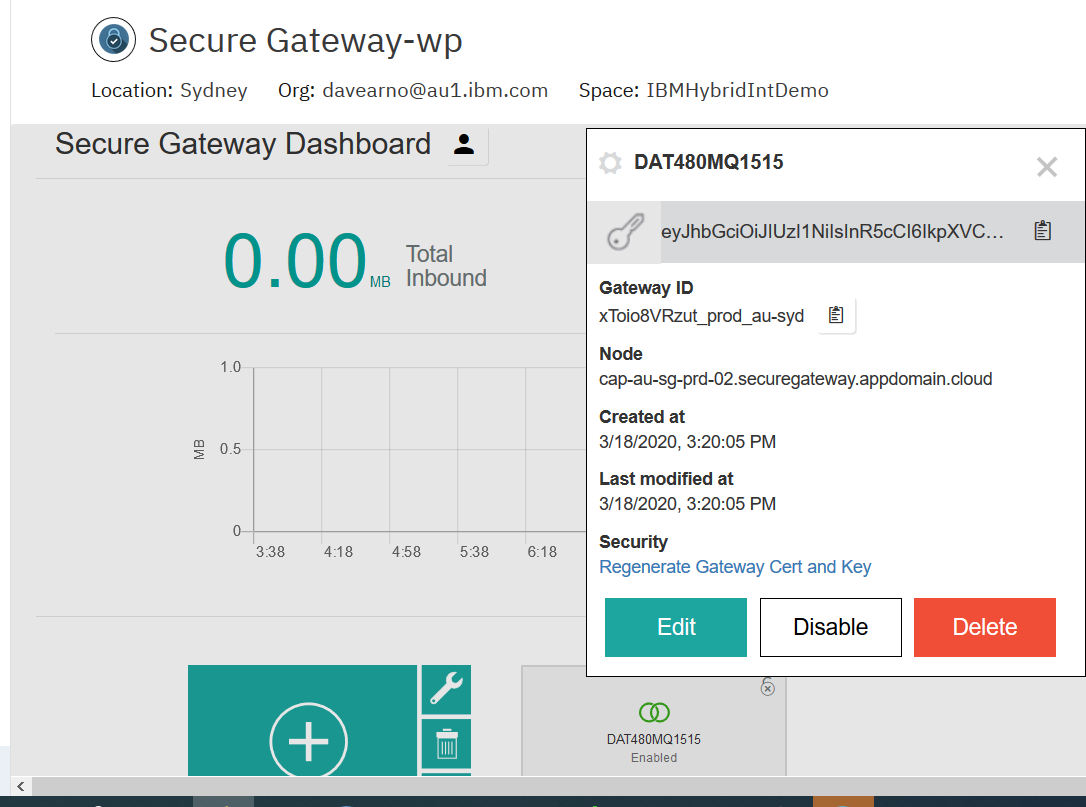
SET CHLAUTH(IVT.SVRCONN) TYPE (ADDRESSMAP) ADDRESS(\*) MCAUSER('MUSR\_MQADMIN')

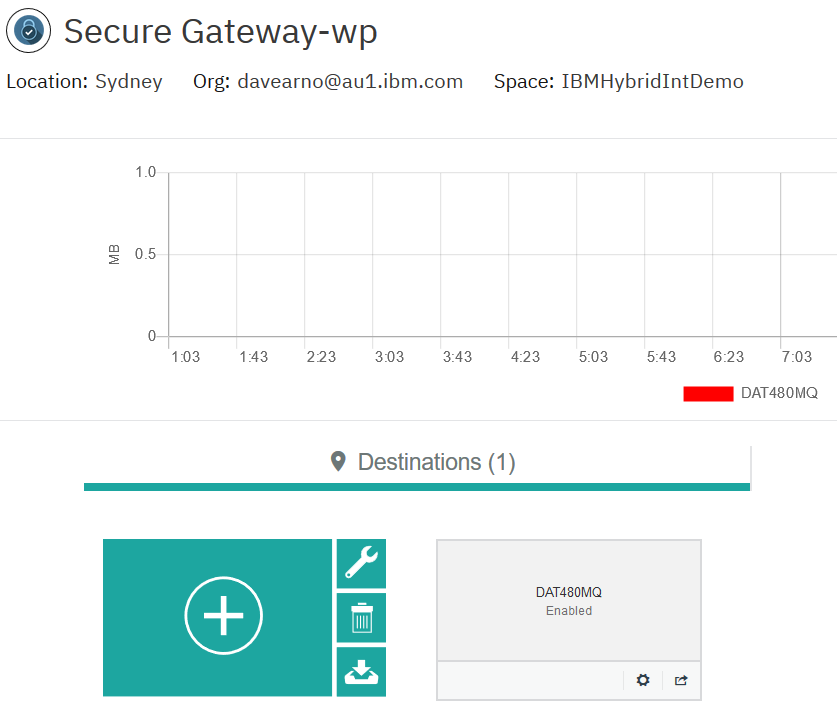
REFRESH SECURITY TYPE(CONNAUTH)

## Set up IBM Secure Gateway Service for Public IP to Queue Manager MyAPPQMGR

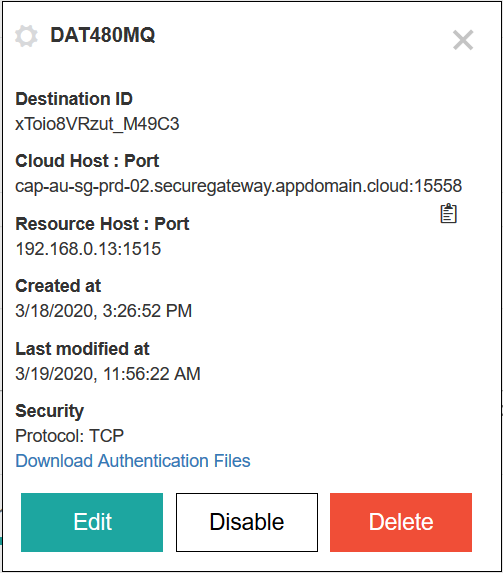
### IBM Cloud server side

Copy the Gateway ID and the Token for use in the client side set up



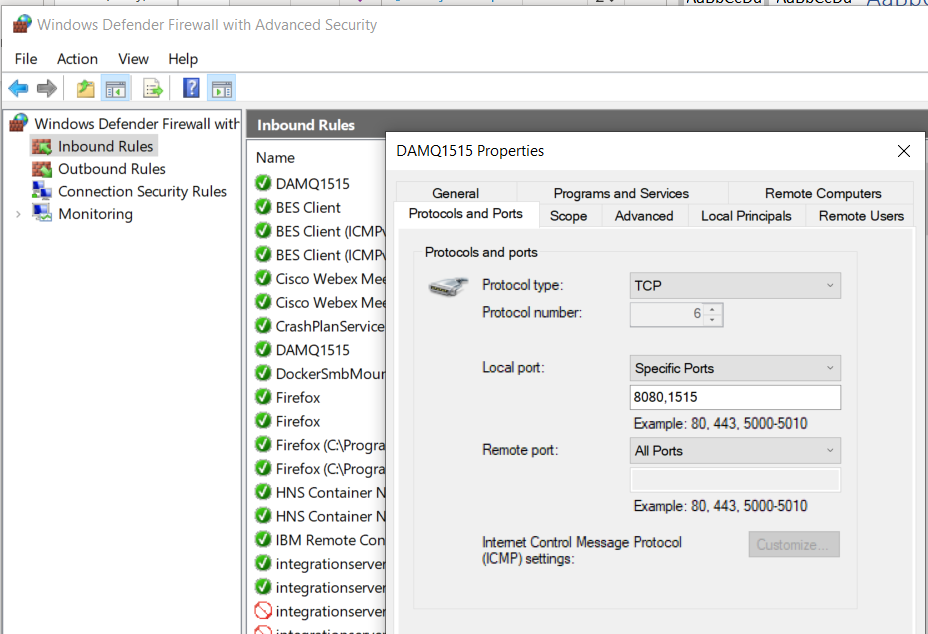


Copy the Cloud:Host Port for use on the ACE MS3 BAR override



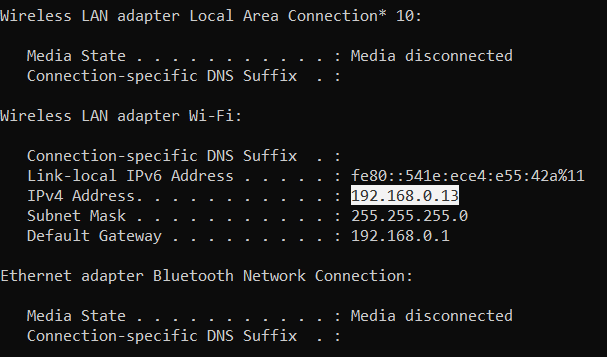
## IBM Secure Gateway Service – Client (Laptop end)

### Windows firewall



### IBM Secure Gateway Service client side

Use ipconfig command to find the IP address of a connected network adapter on your machine/lapop



Add it to the ACLs in the secure gateway client side console.



## Using/Matching Secure Gateway parameters on ACE MS 3 MQOutput configuration

Depending on whether you have complete Part 1 of these materials you will need to check and potentially change the configuration of ACE MS3 running on RH OpenShift and trigger a build/rebuild.

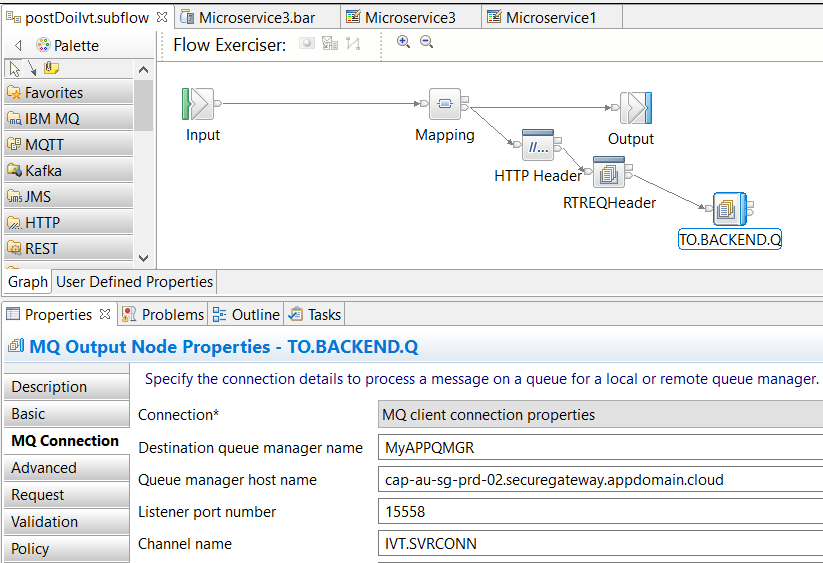
Document: ”1.Developer Experience for ACE MQ with RHOS Tools and Tekton v1.1”

Section: ACE MS 3 – REST Service exposing off RH OpenShift cluster MQ QMGR

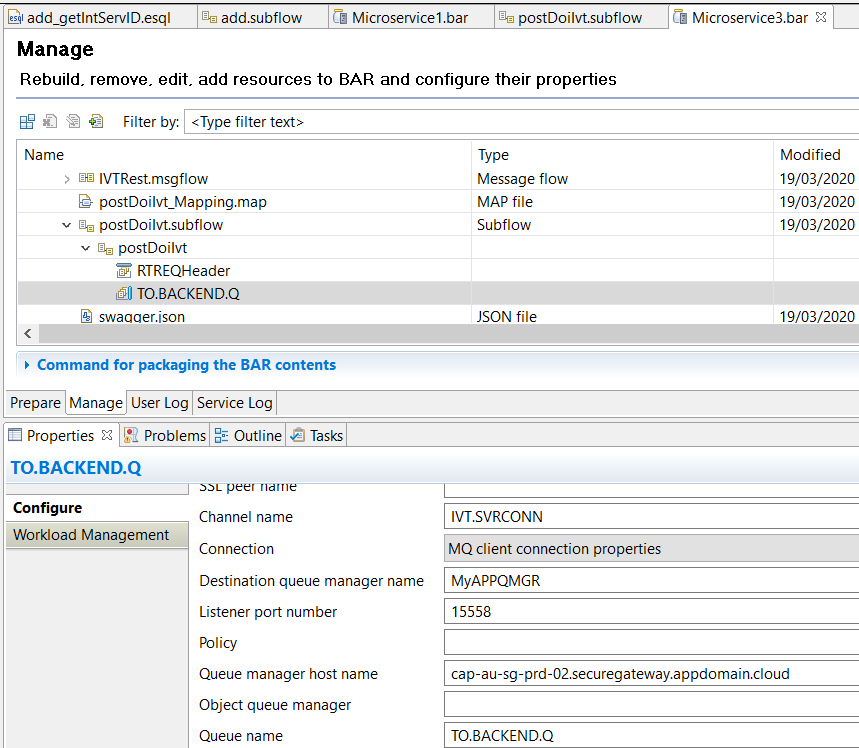
Page: 78

There are two options for configuring the MQOutput node in Microservice 3 to use the secure gateway connection to resolve connectivity to the MyAPPQMGR on your machine/laptop.

#### Option 1:MQ configuration update in the source message flow



#### Option 2:MQ configuration update in the BAR override



With the change made you will need to:

1. Push the BAR to GitHub
2. Webhook trigger or manual RH OpenShift Console rebuild
3. Wait on successful build and deployment before continuing

## Testing MQ via IBM Public IP – using RFHUTILC (MQ client test)

RFHUTIL is a popular MQ test tool shipped as part of supportpac MS03.

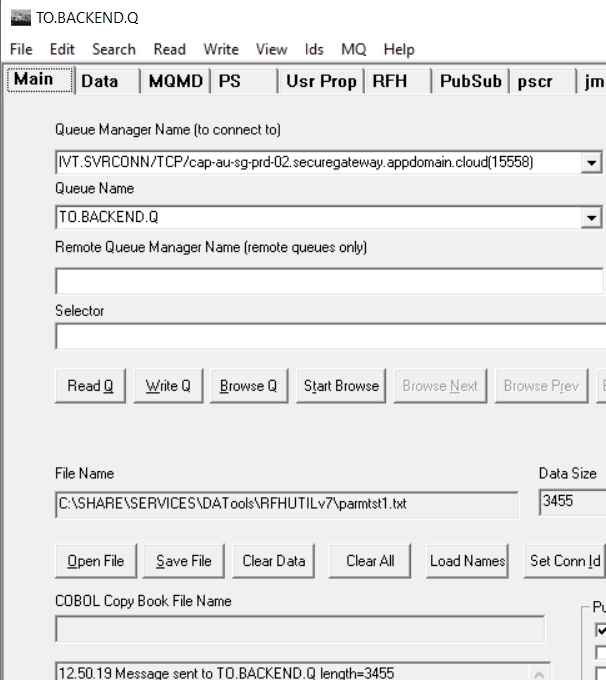
RFHUTILC is the client version and using the following format in the “Queue Manager Name”

IVT.SVRCONN/TCP/cap-au-sg-prd-02.securegateway.appdomain.cloud(15558)

Set the Queue name to TO.BACKEND.Q

Open a file with some data to act as the message

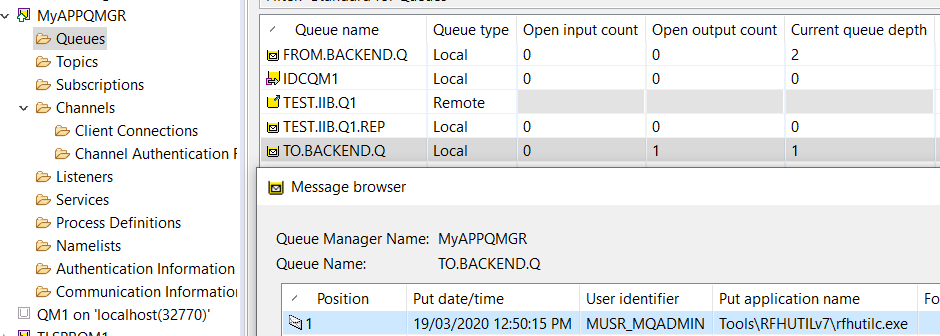
Hit writeQ



Check the secure gateway client side logs and you should see RFHUTILs connection to the MyAPPQMGR via the public IP address.



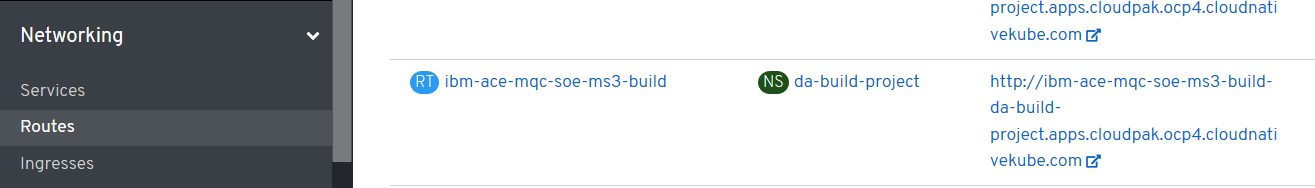
### Check the result on the target queue manager



## Testing ACE Microservice 3 to Put to MQ via IBM Public IP – Using a REST Client

### Obtain the RH OpenShift Route for ACE MS3

RH OpenShift Console->Networking->Routes->ibm-ace-mqc-soe-ms3-build



Or use the command:

Oc get routes



### URL and test data for calling ACE MS3 service

Add the ACE MS 3 base URL to the ipaddress of ACE MS3 retrieved from the route.

Using a REST Client POST to UEL

<http://ibm-ace-mqc-soe-ms3-build-da-build-project.apps.cloudpak.ocp4.cloudnativekube.com/ivtrest/v1/doiIvt>

using data:

{

"name":"anyname",

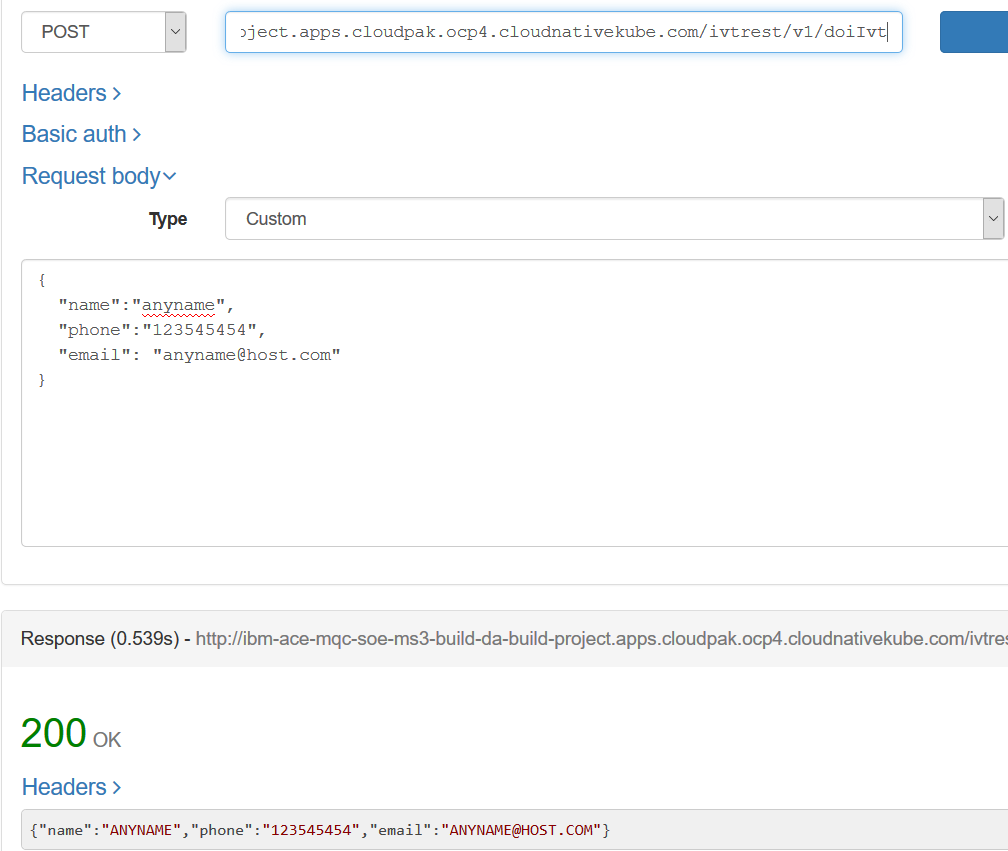
"phone":"123545454",

"email": "anyname@host.com"

}

You should receive an “Echo” style response from ACE MS3 with the data UPPERCASED

### Test with REST Client

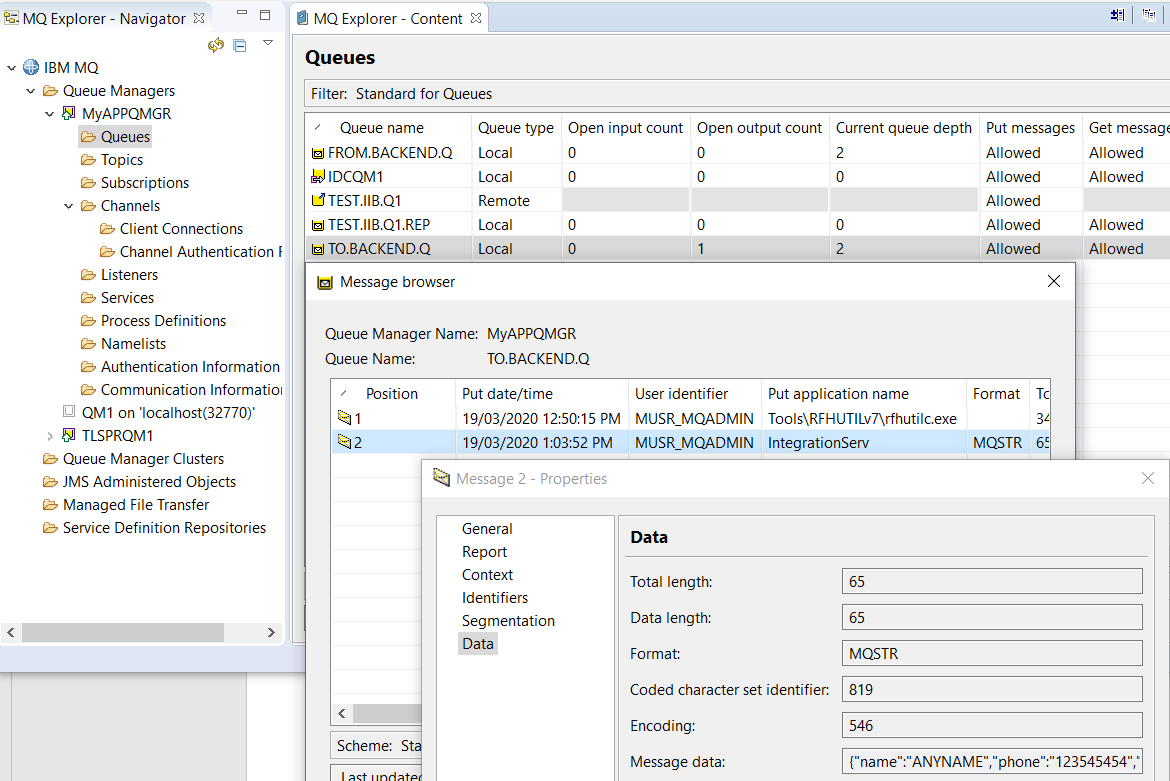


### Check Test Results on MQ

On Queue Manager: MyAPPQMGR

On Queue: TO.BACKEND.Q

Data: JSON payload with the values UPPERCASED

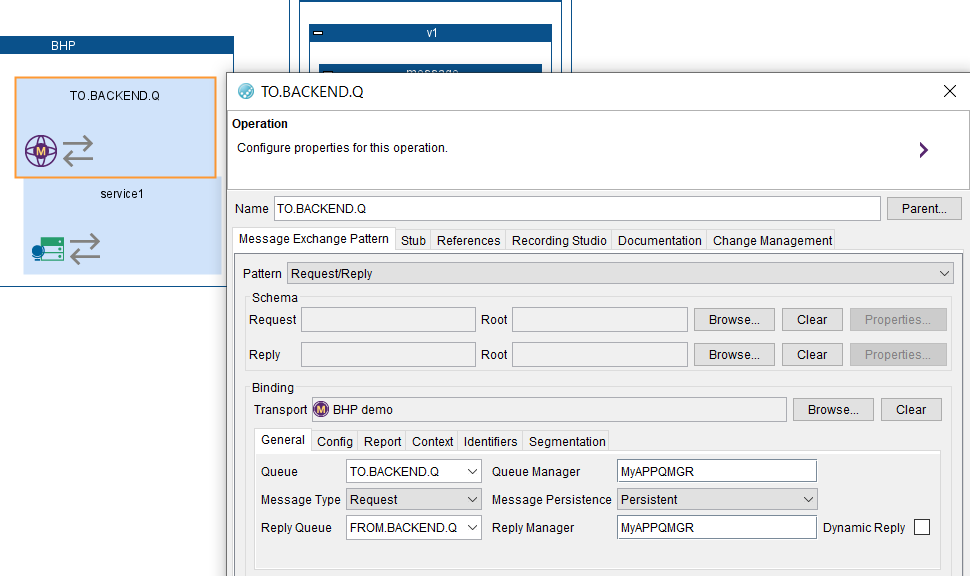


## Using the MQ Stub on IBM Rational Integration Tester as a backend application

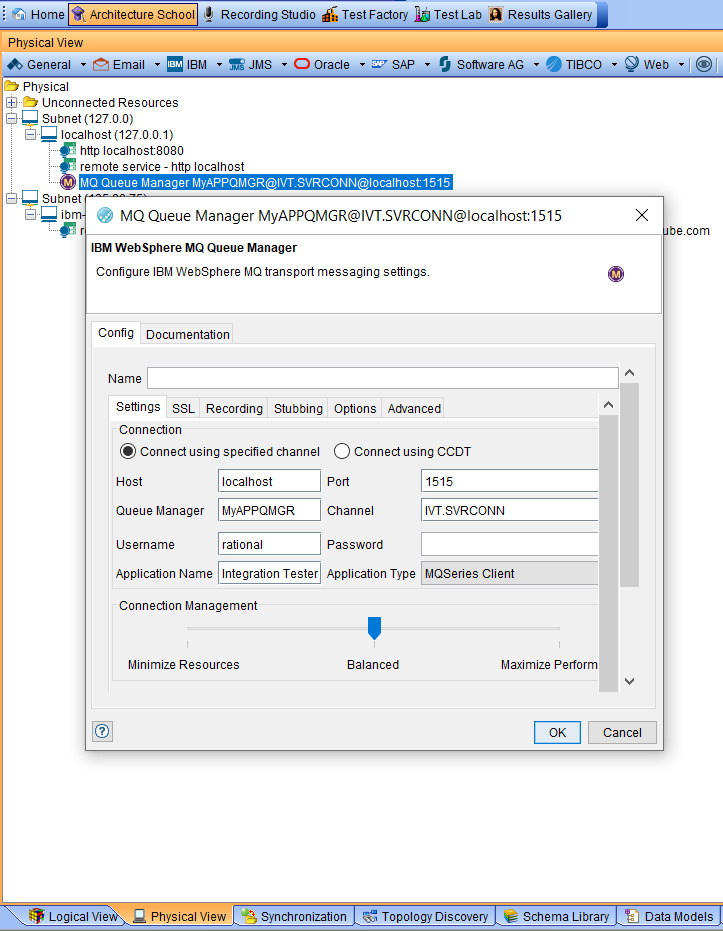
### Review stub parameters in IBM Rational Integration Tester

Observe that the IBM (Rational) Integration Tester stub is set up to Get from TO.BACKEND.Q and Put to FROM.BACKEND.Q.

Depending on the inbound data payload the stub will either append the message “This is from MQ stub” to the original message or just place “This is from MQ stub” on FROM.BACKEND.Q









### Stub uses IVT.SVRCONN channel definition in the MyAPPQMGR

The IBM (Rational) Integration Tester stub will use a SVRCONN channel to connect to the MyAPPQMGR. The channel definition is as follows:

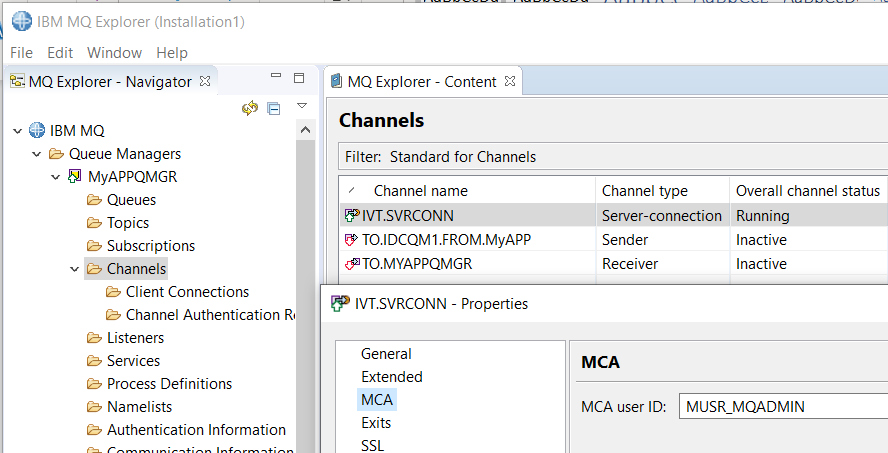
DEFINE CHANNEL(IVT.SVRCONN) CHLTYPE(SVRCONN) REPLACE

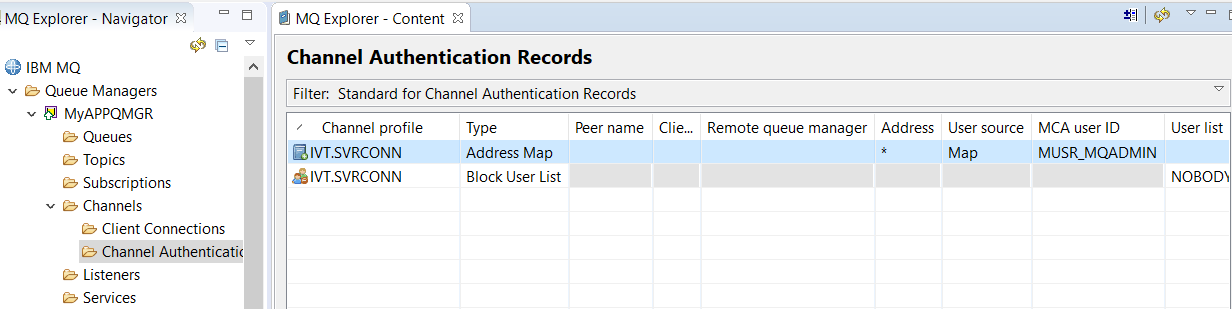
SET CHLAUTH(IVT.SVRCONN) TYPE(BLOCKUSER) USERLIST(nobody)

ALTER AUTHINFO(SYSTEM.DEFAULT.AUTHINFO.IDPWOS) AUTHTYPE(IDPWOS) CHCKCLNT(NONE) ADOPTCTX(YES)

SET CHLAUTH(IVT.SVRCONN) TYPE (ADDRESSMAP) ADDRESS(\*) MCAUSER('')

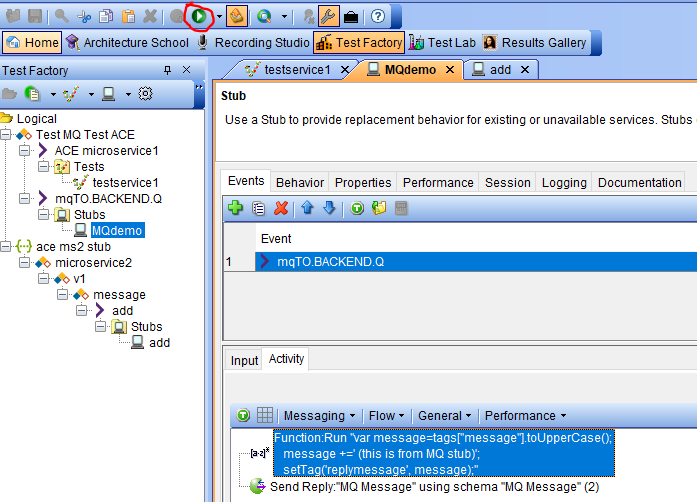
REFRESH SECURITY TYPE(CONNAUTH)





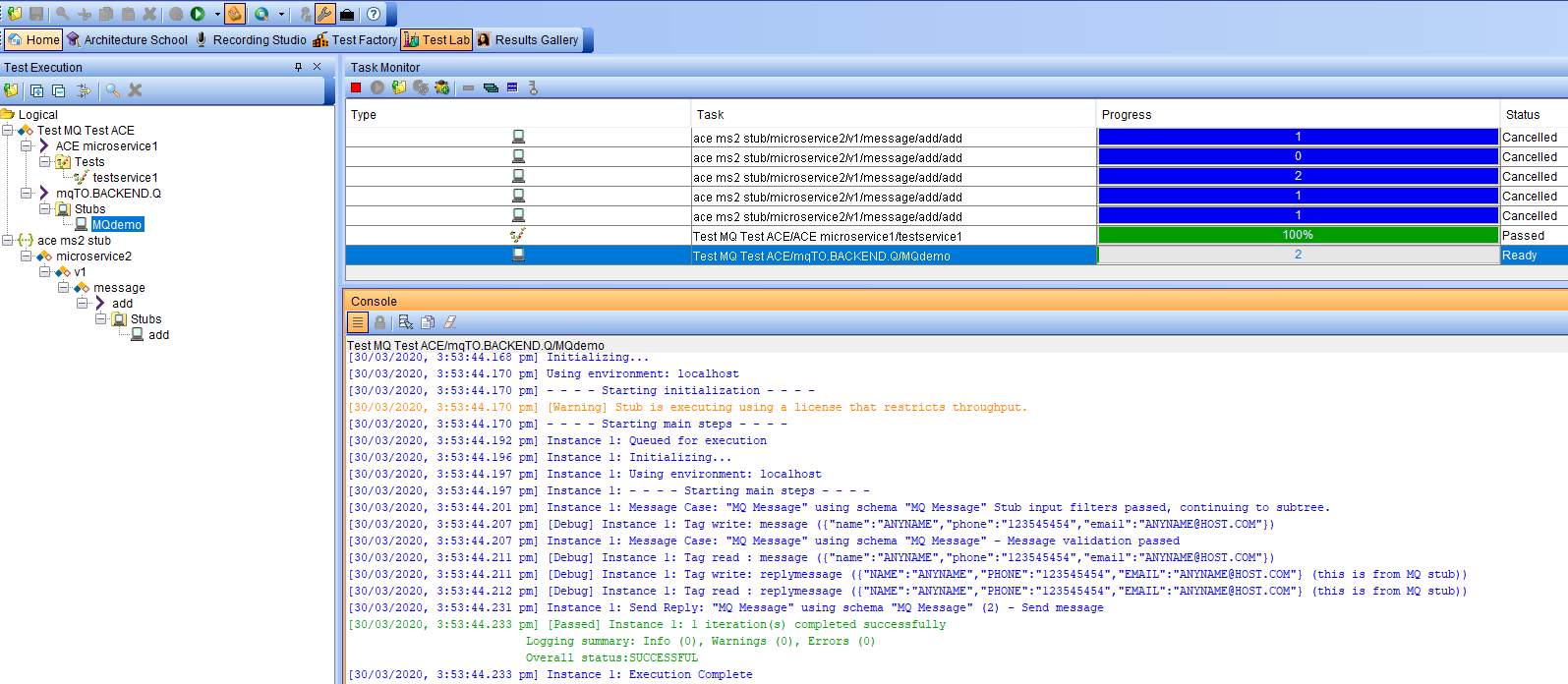
### Start IBM Rational Integration Tester Stub to retrieve messages placed on MQ by ACE MS3

Select the MQ demo stub and hit the Run button



### IBM Rational Integration Tester Stub receives message from ACE MS3

In the Task monitor console in the Test Lab tab observe the Stub reading and writing messages successfully



### Review MyAPPQMGR queue FROM.BACKEND.Q contents

Depending on the test you have run with RHFUTIL and the actual ACE Micro service 3 you will have a mixture of messages on the FROM.BACKEND.Q

Type 1: (this is from MQ stub)

Type 2:{"name":"ANYNAME","phone":"123545454","email":"ANYNAME@HOST.COM"} (this is from MQ stub)

