ACEv11.0.0.2 On IBM Cloud Private 3.1

Devops driven integration to micro services principles

V1.1 Draft

Dave Arnold

Do Nguyen

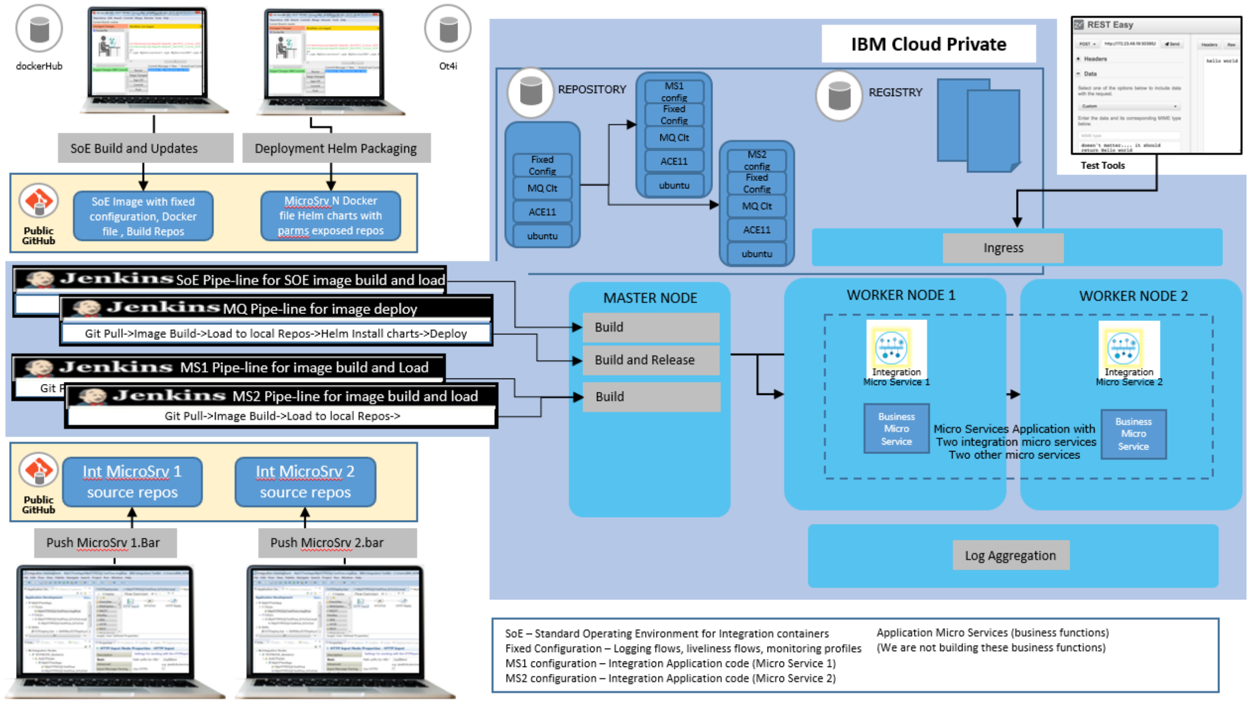
Peter Jessup

February 11, 2019

Table of Contents

Introduction

Scenario Overview Diagram


dockerHub
Ot4i
SoE Build and Updates
Deployment Helm Packaging
Public GitHub
SoE Image with fixed configuration, Docker file , Build Repos
MicroSrv N Docker file Helm charts with parms exposed repos
Jenkins SoE Pipe-line for SOE image build and load
Jenkins MQ Pipe-line for image deploy
Git Pull->Image Build->Load to local Repos->Helm Install charts->Deploy
Jenkins MS1 Pipe-line for image build and Load
Git P
Jenkins MS2 Pipe-line for image build and load
Git Pull->Image Build->Load to local Repos->
Public GitHub
Int MicroSrv 1 source repos
Int MicroSrv 2 source repos
Push MicroSrv 1.Bar
Push MicroSrv 2.bar
IBM Cloud Private
REPOSITORY
Fixed Config
MQ Clt
ACE11
ubuntu
MS1 config
Fixed Config
MQ Clt
ACE11
ubuntu
MS2 config
Fixed Config
MQ Clt
ACE11
ubuntu
REGISTRY
REST Easy
Headers
Data
hello world
Test Tools
Ingress
MASTER NODE
Build
Build and Release
Build
WORKER NODE 1
WORKER NODE 2
Integration Micro Service 1
Integration Micro Service 2
Business Micro Service
Business Micro Service
Micro Services Application with Two integration micro services Two other micro services
Log Aggregation

SoE – Standard Operating Environment for Integration containers
Fixed Configuration – Logging flows, liveliness flows, monitoring profiles
MS1 configuration – Integration Application code (Micro Service 1)
MS2 configuration – Integration Application code (Micro Service 2)

Application Micro Services (business functions)
(We are not building these business functions)

Overview Description

The Scenario

The ICP Cloud represents an environment that exists in the standard Software Delivery Life Cycle (could be SIT, QA, Perf, Prod (or potentially one of those environments in an ICP cluster namespace)
We are delivering a Micro Services application consisting of multiple micro services two of which have requirements that are integration centric and ACE has been chosen as the runtime for these functions

Our fictitious organization maintains a standard operating environment (container image) for micro service integration with a fixed configuration component on top of which the integration micro services are added and then the images deployed and run as a helm release.

The primary mechanism for deployment will be Jenkins but we'll use Microclimate as it sorts out security on the ICP instance for us and standup the Jenkins toolchains.

The Integration Micro Services

The Integration Micro Service 1 will call Integration Micro Service 2
Integration Micro Service 1 RESTInput(HTTP) ->Mapping Node ->RESTRequest (call Integration Microservice 2)->RESTReply(HTTP)
Integration Micro Service 2 RESTInput(HTTP)->Mapping Node Payload+"Hello from Integration Microservice 2"->RESTReply(HTTP)
Integration Micro Service 2 can be called directly

We have set up the integration in this way in order to position for using Istio as a service mesh to route between integration micro services. However, in this initial example I will make use of the ICP DNS service that uses services names to resolve host names and port numbers. So the base URL on the RESTRequest Node in Micro Service 1 will call Micro Service 2 via its service name.

The docker images

The ACE Standard Operating Environment image is:

Ubuntu
ACEv11.0.0.2
MQ v9.1 Client
Fixed configuration - a bar with RESTful Service, a custom Liveliness Listener

The ACE micro services images build FROM this (base) Standard Operating Environment

The Personas

Developer 1 and Developer 2 have their own ACE Toolkits and workspaces and are “just” developers. They write ACE services and flows and build BAR files.

Builder(s) are responsible for maintaining the SoE image and building integration microservices images FROM the SoE based on the BAR files created by the Developers
Deployer(s) are responsible for packaging and deployment of micro services application that can include integration micro services. They use Helm and the ICP Devops tooling and pipelines to deploy.

Customized Jenkins Pipelines

Microclimate you have to customize the jenkins build scripts (micro climate is really set up for Java and NodeJS).

We made changes to the Jenkins pipeline scripts in order to get greater flexibility in the Image Build, Deploy Only and Image Build and Deploy. This allowed us greater control in demoing.

https://github.com/cloudnativedemo/icp-notes/blob/master/microclimate_notes.md

Reference Section

Summary of Github Repositories

ACE Developer Project source for a custom Liveliness probe that will deploy into the Standard Operating Image on which all other images are based
https://github.com/DAVEXACOM/ACEonICPIntMicSoE

ACE Developer Project source for Integration Micro Service 1
https://github.com/DAVEXACOM/ACEonICPIntMicSrv1

ACE Developer Project source for Integration Micro Service 2
https://github.com/DAVEXACOM/ACEonICPIntMicSrv2

ACE Standard Operating Environment Image Build - The base image build with Liveliness probe that other images are build FROM
https://github.com/DAVEXACOM/ACEonICPIntStdOpImg

ACE Micro Service 1 Image Build
https://github.com/DAVEXACOM/ACEonICPIntMicSrv1Img

ACE Micro Service 2 Image Build
https://github.com/DAVEXACOM/ACEonICPIntMicSrv2Img

Helm Charts that can deploy ACE Micro Service 1 or 2 as an individual for testing (modify Jenkins file and Values.yaml)
https://github.com/DAVEXACOM/ACEonICPIntMicSrvHelm

Helm Charts that deploy both ACE Micro Service 1 and 2 as a micro services application

https://github.com/DAVEXACOM/ACEonICPIntMicSrvApplHelm

Summary of Dockerhub Images

ACE Developer Project source for a custom Liveliness probe that will deploy into the Standard Operating Image on which all other images are based

https://cloud.docker.com/u/davexacom/repository/docker/davexacom/ace11002mqc91soe

ACE Developer Project source for Integration Micro Service 1

https://cloud.docker.com/u/davexacom/repository/docker/davexacom/ace11002mqc91intms1

ACE Developer Project source for Integration Micro Service 2

https://cloud.docker.com/u/davexacom/repository/docker/davexacom/ace11002mqc91intms2

You can use the dockerHub images with ICP rather than the on board repository OR pull them to you workstations and play around with them in a docker only (non K8s) environment.

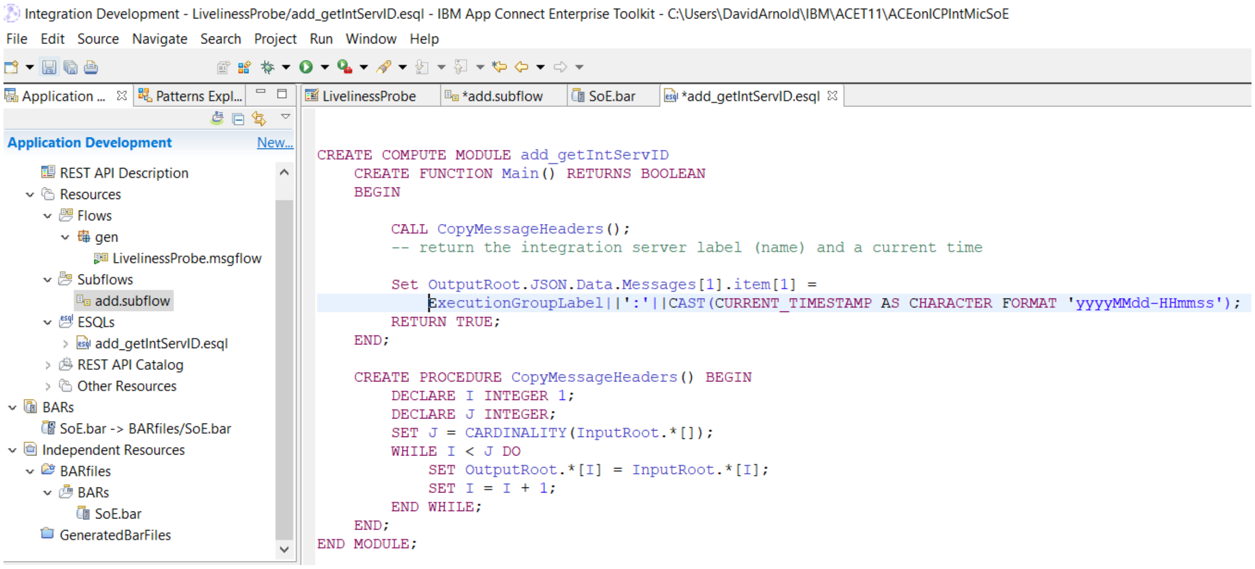
ACE Integration Liveliness Probe

Github Source Repos for ACE Liveliness Probe (The SoE ACE project)

https://github.com/DAVEXACOM/ACEonICPIntMicSoE

Description

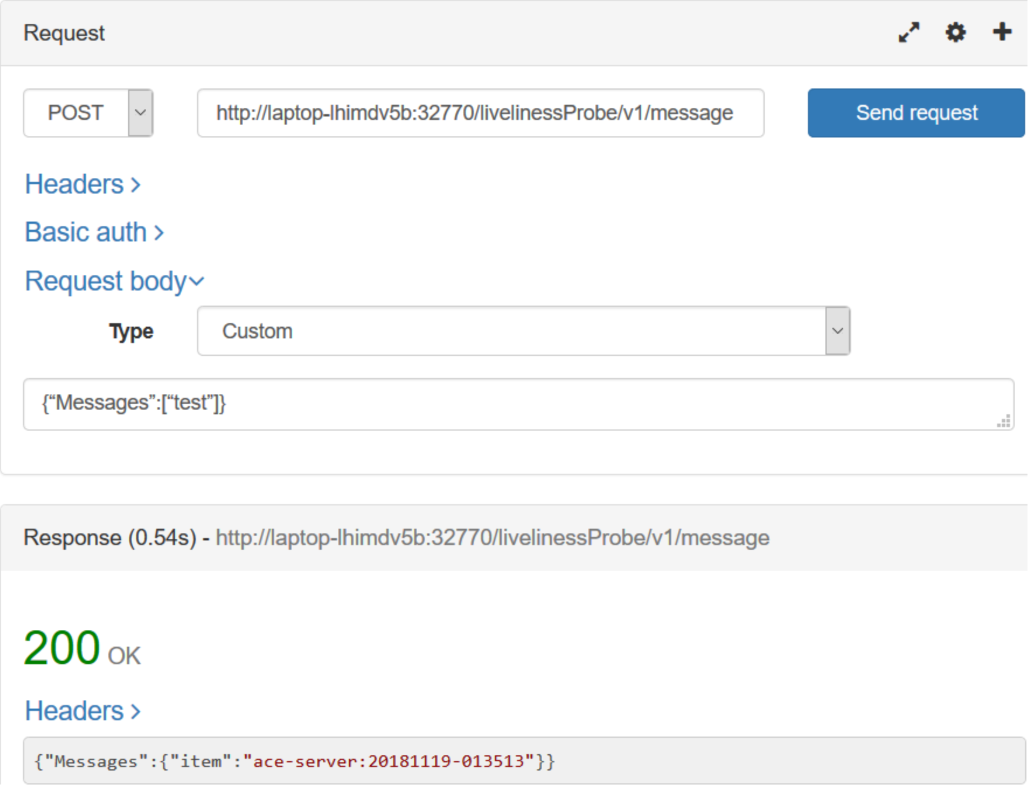
The Liveliness Probe Service is an other restful service that we deploy into the ACE Standard operating environment. The base image from which Micro Service 1 and Micro Service 2 images are built. So it appears in all ACE Integration Micro Services. (this is not the service baked into the cloud paks we turn those off to demonstrate having a customer centric base standard image. (an SoE).



```
CREATE COMPUTE MODULE add_getIntServID
    CREATE FUNCTION Main() RETURNS BOOLEAN
    BEGIN

        CALL CopyMessageHeaders();
        -- return the integration server label (name) and a current time

        Set OutputRoot.JSON.Data.Messages[1].item[1] =
            ExecutionGroupLabel||':'||CAST(CURRENT_TIMESTAMP AS CHARACTER FORMAT 'yyyyMMdd-HHmmss');
        RETURN TRUE;
    END;

    CREATE PROCEDURE CopyMessageHeaders() BEGIN
        DECLARE I INTEGER 1;
        DECLARE J INTEGER;
        SET J = CARDINALITY(InputRoot.*[]);
        WHILE I < J DO
            SET OutputRoot.*[I] = InputRoot.*[I];
            SET I = I + 1;
        END WHILE;
    END;
END MODULE;
```

Testing Liveliness Probe

Note it returns the integration server name plus a current timestamp for input: {“Messages”:[“test”]}



POST http://laptop-lhimdv5b:32770/livelinessProbe/v1/message

Type: Custom

```
{"Messages":["test"]}
```

Response (0.54s) - http://laptop-lhimdv5b:32770/livelinessProbe/v1/message

200 OK

```
{"Messages":{"item":"ace-server:20181119-013513"}}
```

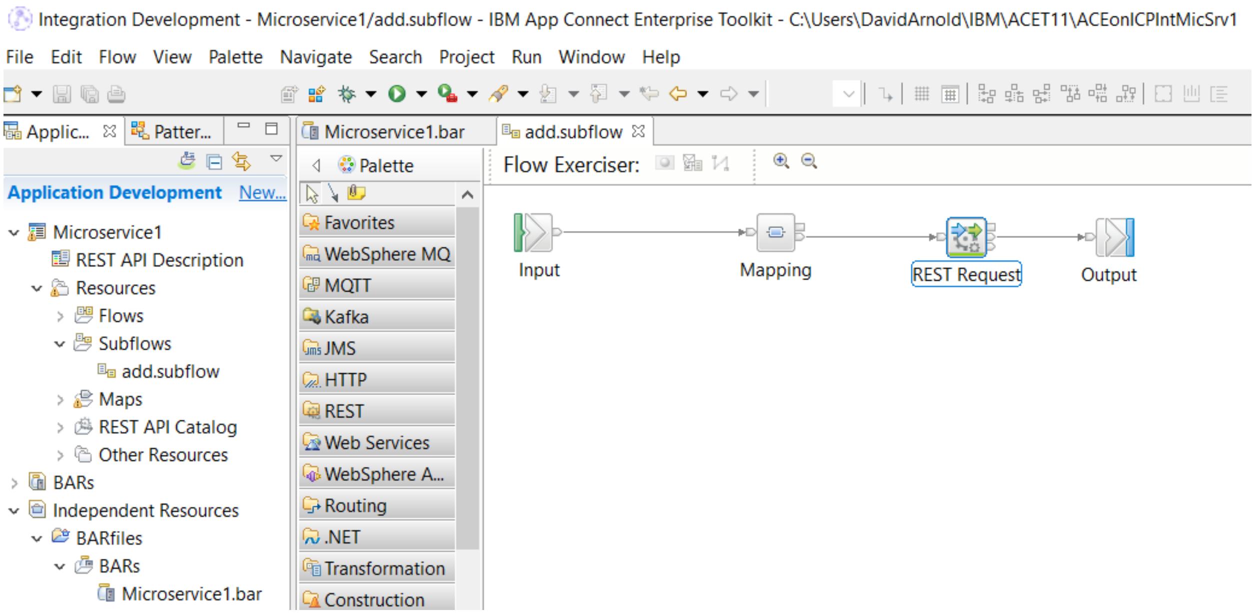
ACE Integration Micro Service 1

Github Source Repos for ACE Micro Service 1

https://github.com/DAVEXACOM/ACEonICPIntMicSrv1

Description

Integration Micro Service 1 calls Integration Micro Service 2 via Rest request



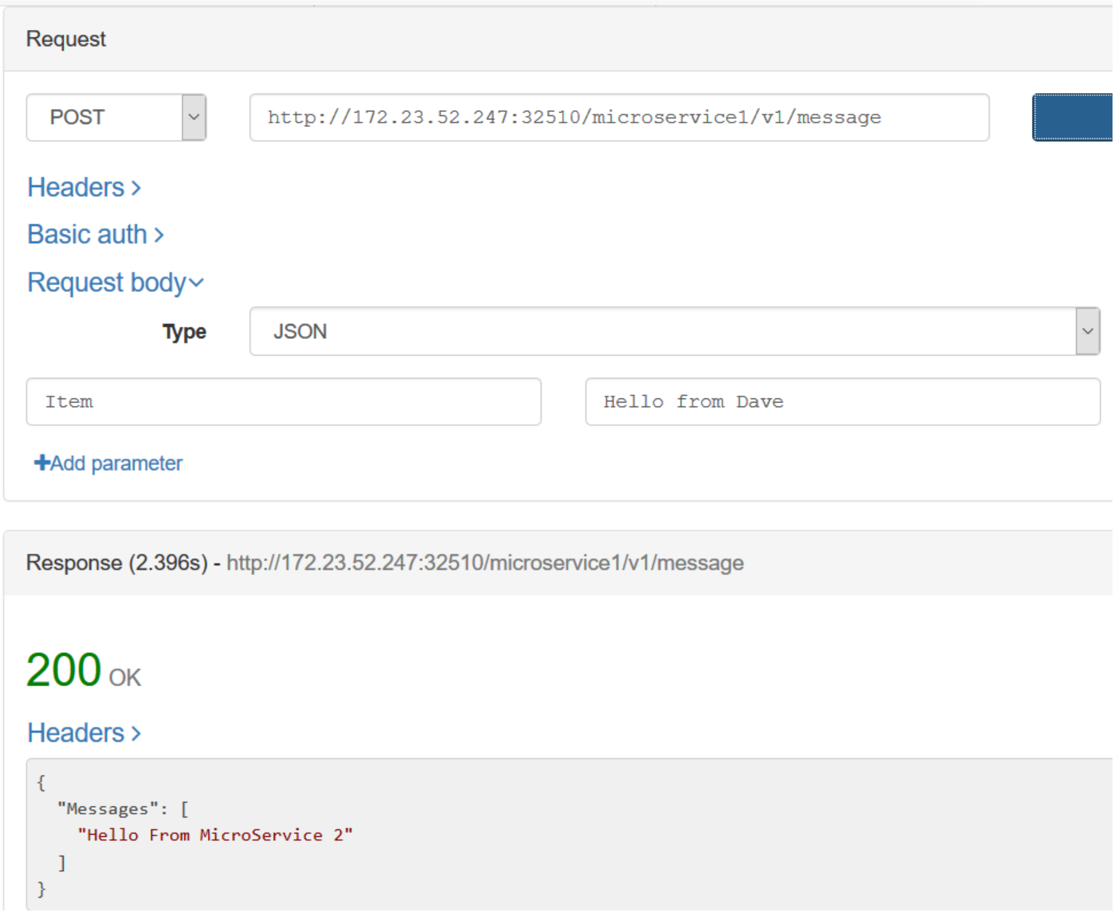
The base URL on the REST Request Node leverages the ICP DNS service naming convention to call MS2 via its service name. This will need to be replaced with Istio service mesh routing.

http://ace11002mqc91intmsall-intmicsrvtwo:7800/microservice2/v1

Testing Micro Service 1

Micro service 1 if deployed and tested standalone will fail.

Micro service 1 if deployed and tested with micro service 2 will return “hello from Micro Service 2”



Request

POST

http://172.23.52.247:32510/microservice1/v1/message

Type JSON

Item | Hello from Dave

Response (2.396s) - http://172.23.52.247:32510/microservice1/v1/message

200 OK

```
{
  "Messages": [
    "Hello From MicroService 2"
  ]
}
```

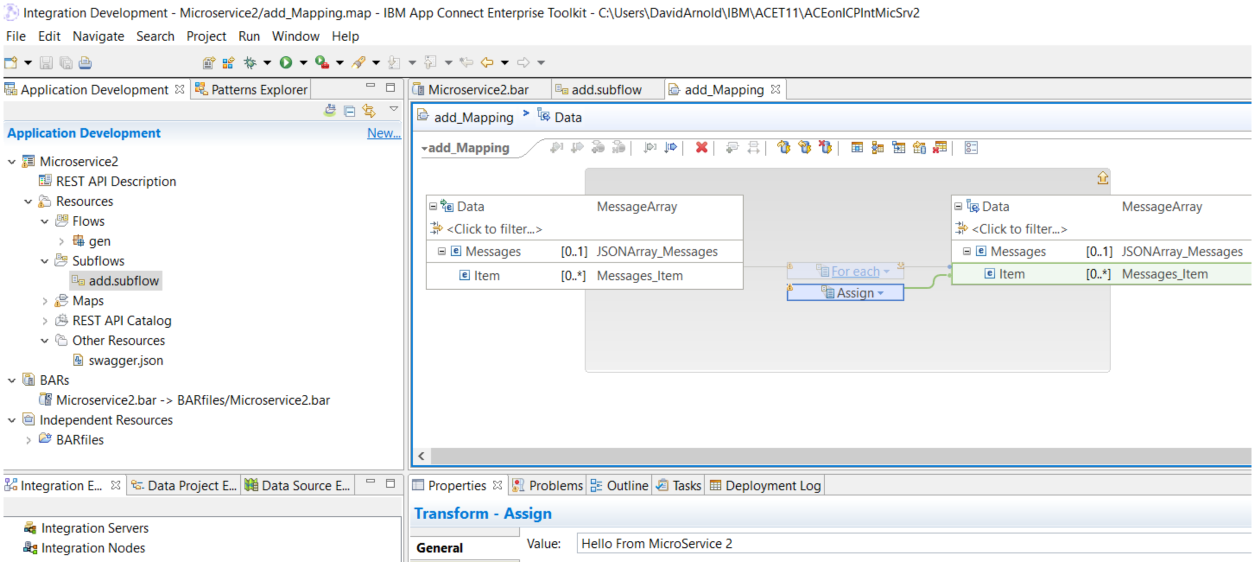
ACE Integration Micro Service 2

Github Source Repos for ACE Micro Service 2

https://github.com/DAVEXACOM/ACEonICPIntMicSrv2

Description

Integration Micro Service 2 can be called directly as a rest service or access by calling Micro Service 1 (which in turn calls it)

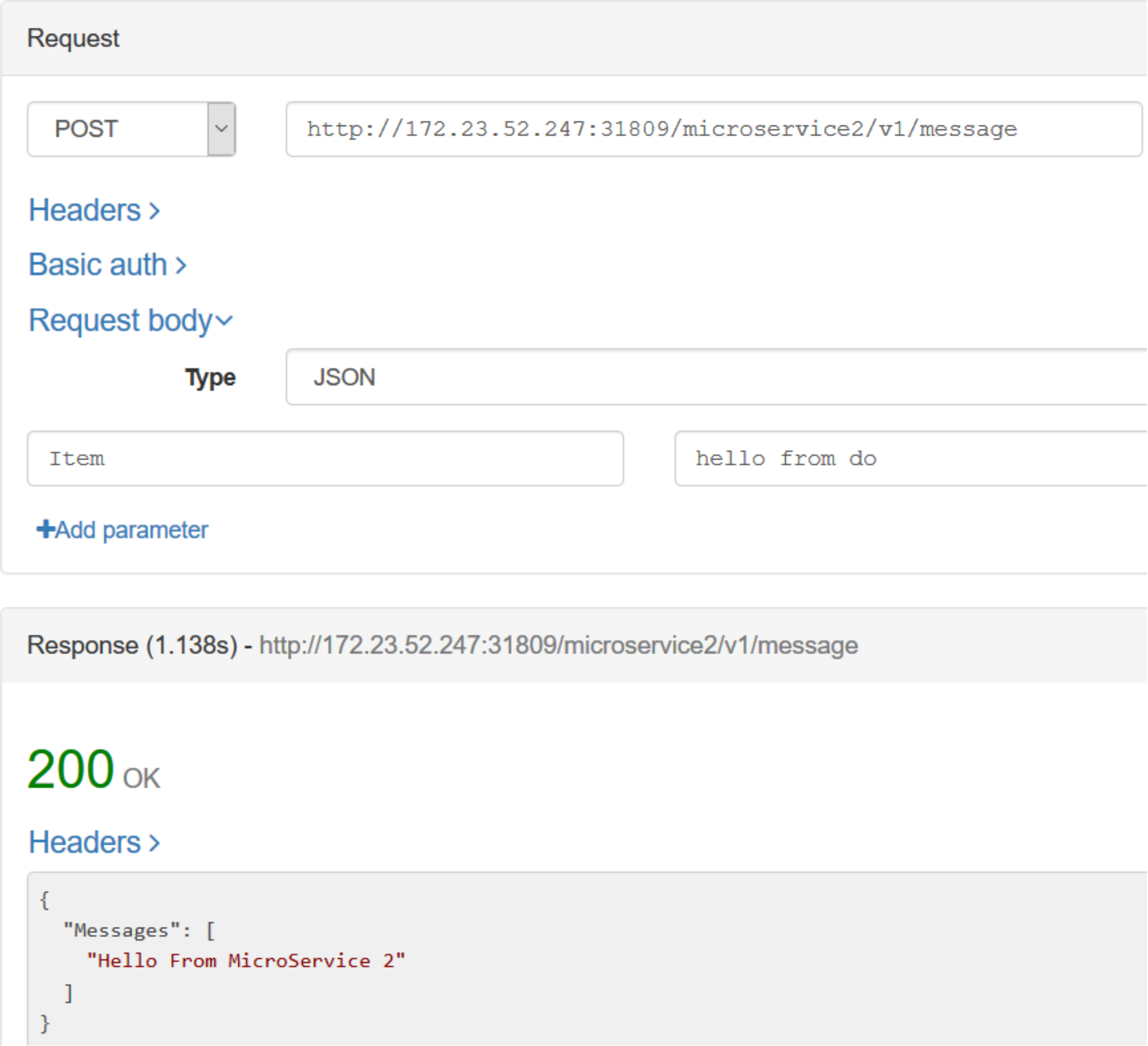


Integration Development - Microservice2/add_Mapping.map - IBM App Connect Enterprise Toolkit - C:\Users\DavidArnold\IBM\ACET11\ACEonICPIntMicSrv2

Value: Hello From MicroService 2

Testing Micro Service 2

Note it returns hello from Micro Service 2



Request

POST http://172.23.52.247:31809/microservice2/v1/message

Type: JSON

Item: hello from do

Response (1.138s) - http://172.23.52.247:31809/microservice2/v1/message

200 OK

```
{
  "Messages": [
    "Hello From MicroService 2"
  ]
}
```

Building the Environment on ICP

Pre-Req work – Customization of Microclimate/Jenkins build scripts on ICP

Documentation is kept up to date on this process here
https://github.com/cloudnativedemo/icp-notes/blob/master/microclimate_notes.md

1. Create pipeline deployment namespace

```
kubectl create namespace microclimate-pipeline-deployments
```

2. Edit ClusterImagePolicy

```
kubectl edit clusterimagepolicies ibmcloud-default-cluster-image-policy
```

To add the following:

```
  - name: mycluster.icp:8500:*
  - name: docker.io/maven:*
  - name: docker.io/lachlanevenson/k8s-helm:*
  - name: docker.io/jenkins/*
```

3. Create Docker Registry secret to microclimate namespace

```
kubectl create secret docker-registry microclimate-registry-secret \
  --docker-server=mycluster.icp:8500 \
  --docker-username=admin \
  --docker-password=admin \
  --docker-email=null
```

4. Initialise Helm and login

```
helm init --client-only --skip-refresh
cloudctl login -a https://mycluster.icp:8443 -u admin -p admin -c id-mycluster-account -n
default --skip-ssl-validation
```

5. Create Helm secret

```
export HELM_HOME=$HOME/.helm
kubectl create secret generic microclimate-helm-secret --from-
file=cert.pem=$HELM_HOME/cert.pem --from-file=ca.pem=$HELM_HOME/ca.pem --from-
file=key.pem=$HELM_HOME/key.pem
```

6. Create Docker Registry secret for microclimate-pipeline-deployments namespace

```
kubectl create secret docker-registry microclimate-pipeline-secret \
  --docker-server=mycluster.icp:8500 \
  --docker-username=admin \
  --docker-password=admin \
  --docker-email=null \
  --namespace=microclimate-pipeline-deployments
```

7. Update ImagePullSecret for microclimate-pipeline-deployments namespace

```
kubectl patch serviceaccount default --namespace microclimate-pipeline-deployments -p
'{"imagePullSecrets": [{"name": "microclimate-pipeline-secret"}]}'
```

8. Customise Jenkins library

By default, the Jenkins library parameter is pointing to https://github.com/microclimate-dev2ops/jenkins-library This Jenkins library was a part of the Microclimate DevOps process. When a pipeline is created within a project in Microclimate, microclimate will create a Jenkins pipeline. The pipeline uses this library to .. 1. Pull the code from github repo . .. 2. Build a Docker image based on a Dockerfile found in the repo . .. 3. Authenticate and push the image into ICP's private registry . .. 4.

Notify Microclimate to move to the next stage (e.g. deploy) . .. 5. Microclimate 'helm deploy' the helm chart found in the repo (by default it's under the /chart directory) .

Unfortunately, Microclimate only deploy it's supported project types e.g. Swift, NodeJS, Java/Liberty or Springboot. The easiest way to address this limitation is to fork and update the Jenkins library and inject the 'helm deploy' scriptlet onto step 4 (line 400 of microserviceBuilderPipeline.groovy)

```
            container ('helm') {
            echo "Attempting to deploy the test release"
            def deployCommand = "helm install ${realChartFolder} --values pipeline.yaml --namespace ${namespace} --name ${helmRelease}"
            if (fileExists("chart/overrides.yaml")) {
              deployCommand += " --values chart/overrides.yaml"
            }
            if (helmSecret) {
              echo "Adding --tls to your deploy command"
              deployCommand += helmTlsOptions
            }
            testDeployAttempt = sh(script: "${deployCommand} > deploy_attempt.txt", returnStatus: true)
            if (testDeployAttempt != 0) {
              echo "Warning, did not deploy the test release into the test namespace successfully, error code is: ${testDeployAttempt}"
              echo "This build will be marked as a failure: halting after the deletion of the test namespace."
            }
            printFromFile("deploy_attempt.txt")
         }
```

- ***Note:*** in my deployCommand, I've created one new variable `${helmRelease}`. The variable is defined on the top of the script (line 56 of the microserviceBuilderPipeline.groovy). Alternatively, you can just reuse `${image}` as your helm release name .

```
  def helmRelease = (config.releaseName ?: config.image ?: "").trim()
```

- My forked updated Jenkins library repo can be found here .

9. Deploy Microclimate helm chart

Via Helm command line

- **Add ibm-charts Helm repo**

```
helm repo add ibm-charts https://raw.githubusercontent.com/IBM/charts/master/repo/stable/
```

- **Deploy microclimate Helm chart**

```
helm install --name microclimate --namespace <target namespace> --set global.rbac.serviceAccountName=micro-sa,jenkins.rbac.serviceAccountName=pipeline-sa,global.ingressDomain=172.23.52.247.nip.io,jenkins.Pipeline.Template.RepositoryUrl=https://github.com/cloudnativedemo/jenkins-library.git,jenkins.Pipeline.Template.Version=master ibm-charts/ibm-microclimate --tls
```

Note: Replace <172.23.52.247> with your <PROXY_IP>

Via ICP catalog

- Select ibm-microclimate from ICP catalog > click `Configure`
- Provide values for the following parameters:
  - `Helm release name`: your-microclimate-release-name
  - `Namespace`: default (or your preferred namespace)
  - `Microclimate hostname`: microclimate.172.23.52.247.nip.io (replace with your <microclimate.PROXY_IP.nip.io> or your own hostname)
  - Ensure that you've already created Persistent Volumes for Microclimate and Jenkins
  - `Service account name for Portal`: micro-sa
  - `Jenkins library repository`: https://github.com/cloudnativedemo/jenkins-library.git
  - `Jenkins hostname`: jenkins.172.23.52.247.nip.io (replace with your <jenkins.PROXY_IP.nip.io> or your own hostname)
  - `Service account name`: pipeline-sa
  - Click `deploy`

Create a project in Microclimate and Deploy

Once the Microclimate helm deployment completed, you can start to deploy your custom project

1. Make sure that your project contains a Dockerfile, Jenkinsfile and a chart directory (for helm chart)
2. Launch Microclimate (https://microclimate.172.23.52.247.nip.io - replace with your own microclimate hostname) and accept licensing agreement (for first launch only)
3. Select Projects > Click `New Project`
4. Select `Java` project type and provide a `project name` > click `Next`
5. Select `Microprofile/J2EE` and keep default value for `Context root` > click `Create`
6. Once the project is created, select `Pipeline` on the left menu
7. Click `Create pipeline`, and provide name and github repo of the pipeline > click `Create pipeline` to create a Jenkins pipeline
8. Switch to Jenkins (https://jenkins.172.23.52.247.nip.io - replace with your Jenkins hostname) to see if the pipeline has been created and built (refer to the troubleshooting section below if you have to wait for too long)

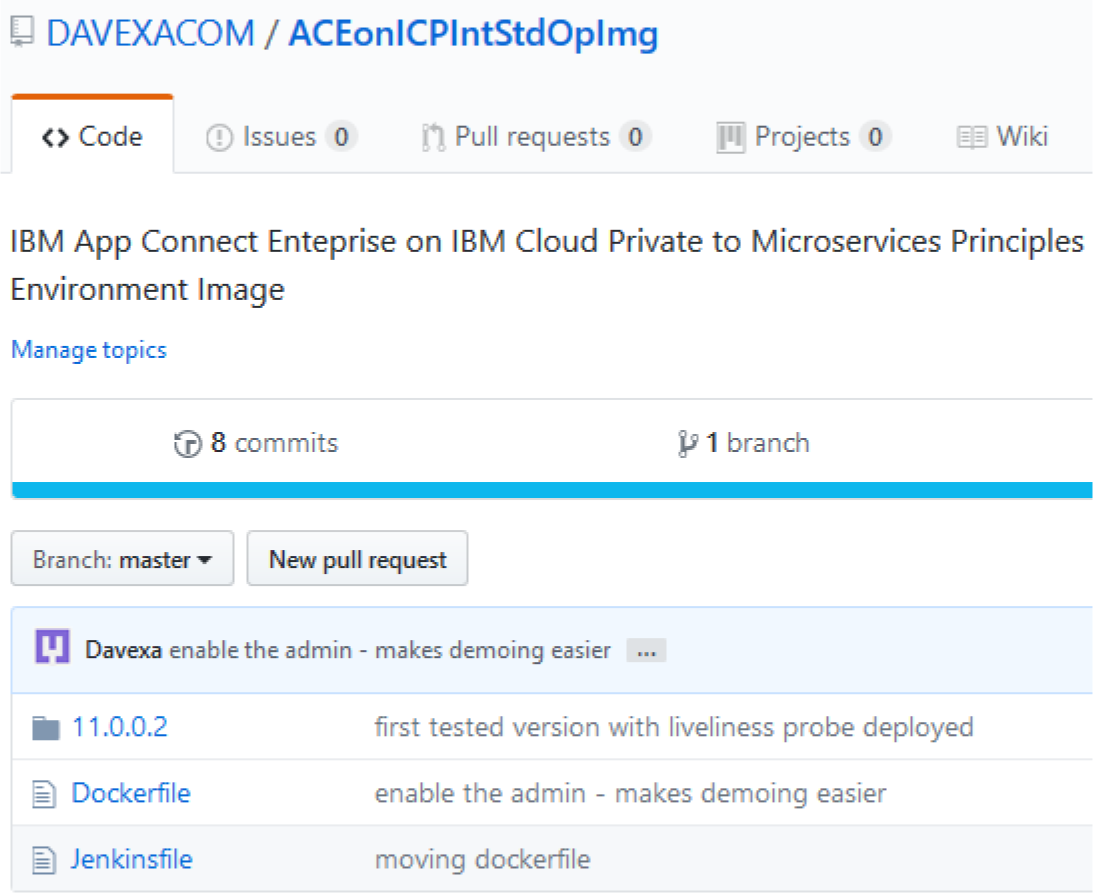
Troubleshooting

- When your Jenkins pipeline keeps looking for an executor for too long, there's probably an error occurred within your Jenkins containers. To identify the issue:
  - Identify the Jenkins pod name: `kubectl get pods -n <NAME_SPACE_WHERE_JENKINS_INSTALLED> | grep jenkins`

- View the log: `kubectl log -n <NAME_SPACE_WHERE_JENKINS_INSTALLED> <JENKINS_POD_NAME> -f`
- Most of the case I found caused by cluster image policy is not defined, you might need to update the default `clusterimagepolicy`
  - `kubectl edit clusterimagepolicies ibmcloud-default-cluster-image-policy`

Creating the Microclimate projects and Jenkins pipelines

ACE on ICP Integration Standard Operating Environment Image

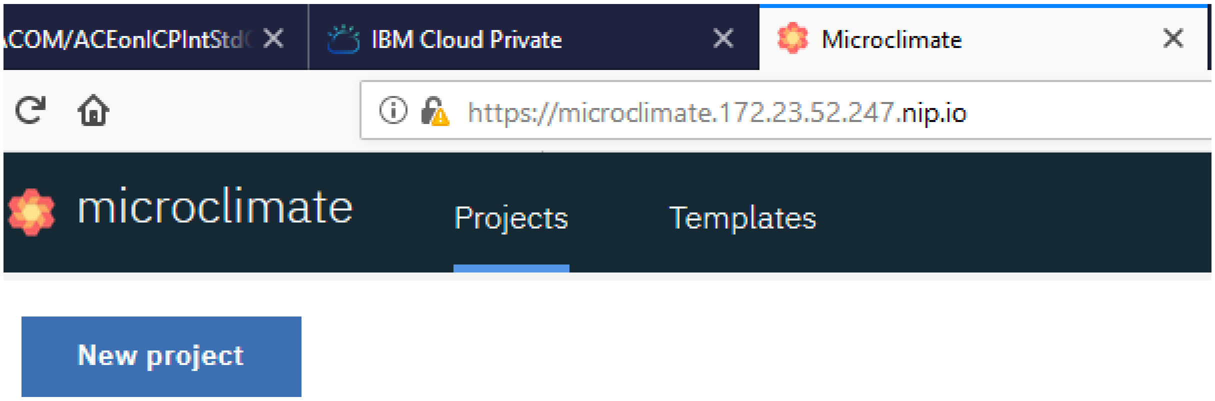


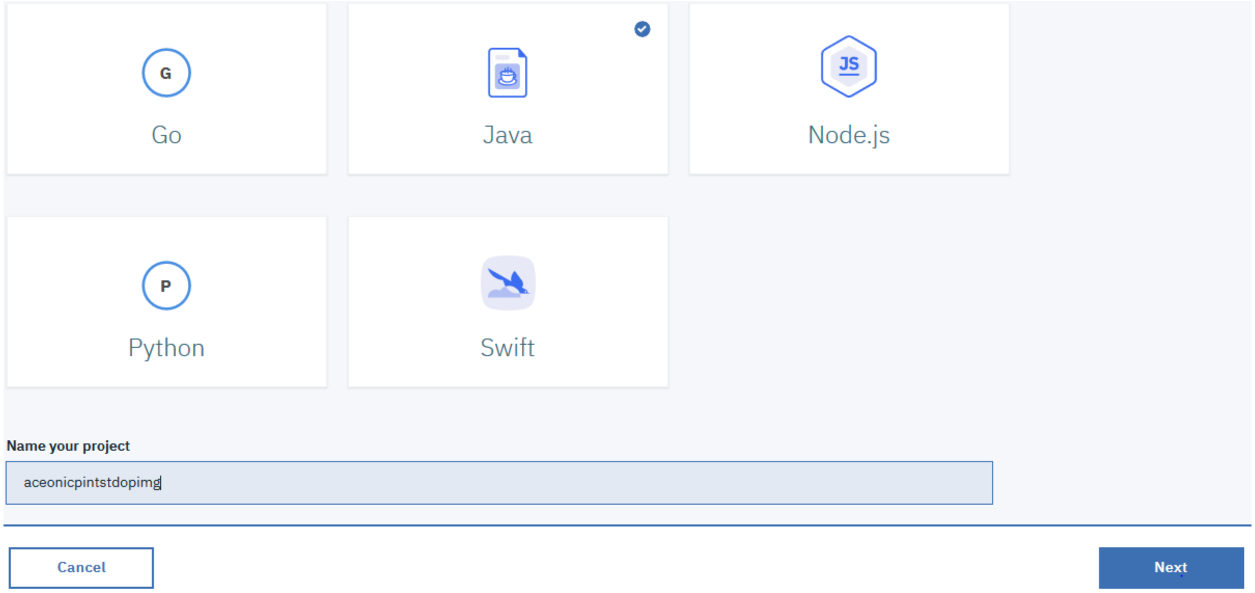
DAVEXACOM / ACEonICPIntStdOpImg

IBM App Connect Enteprise on IBM Cloud Private to Microservices Principles Environment Image

Davexa enable the admin - makes demoing easier ...

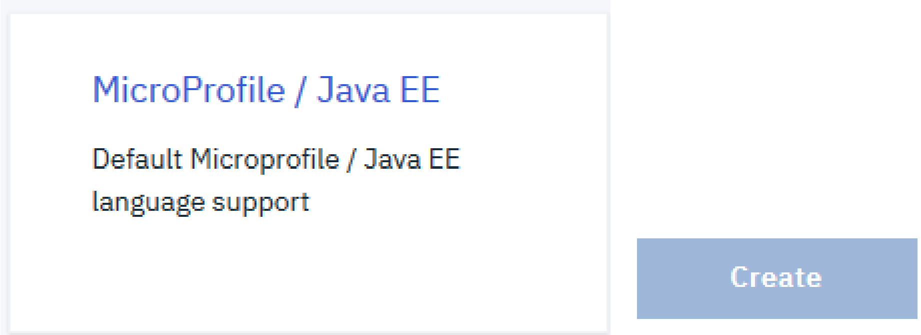
11.0.0.2	first tested version with liveliness probe deployed
Dockerfile	enable the admin - makes demoing easier
Jenkinsfile	moving dockerfile





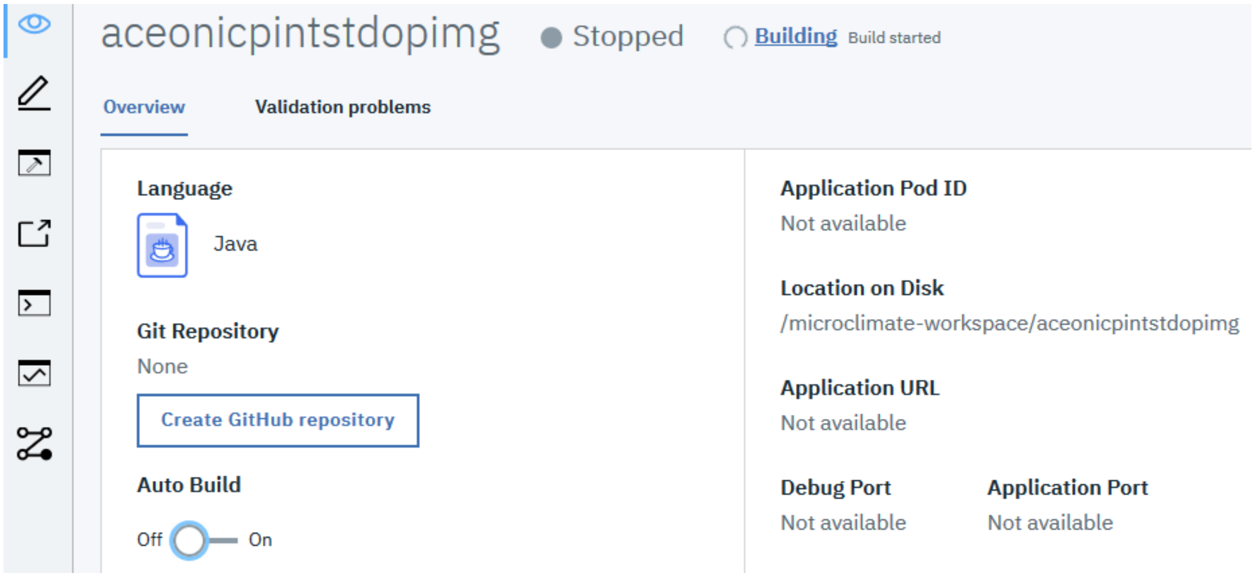
Name your project

aceonicpintstdopimg



MicroProfile / Java EE

Default Microprofile / Java EE language support



aceonicpintstdopimg ● Stopped ◯ Building Build started

Language

Java

Git Repository

None

Auto Build

Off ○ On

Application Pod ID

Not available

Location on Disk

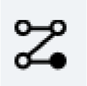
/microclimate-workspace/aceonicpintstdopimg

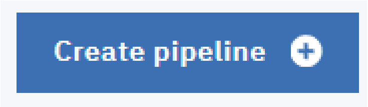
Application URL

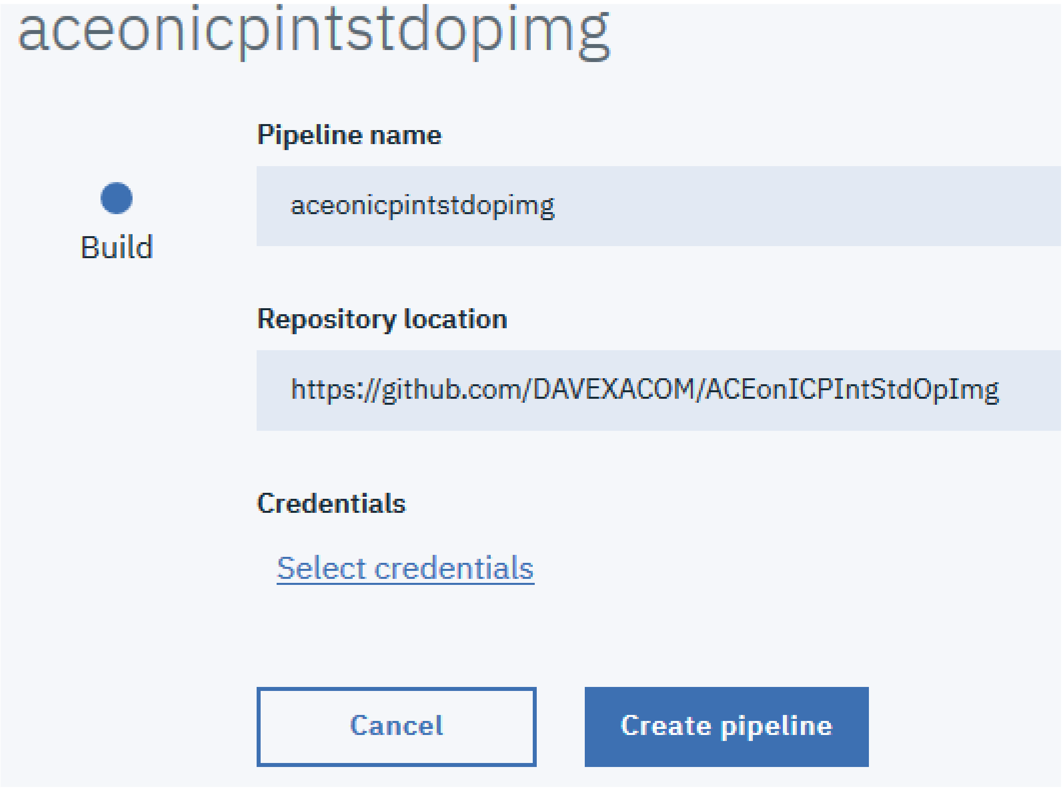
Not available

Debug Port	Application Port
Not available	Not available

Click on the Pipeline icon






aceonicpintstdopimg

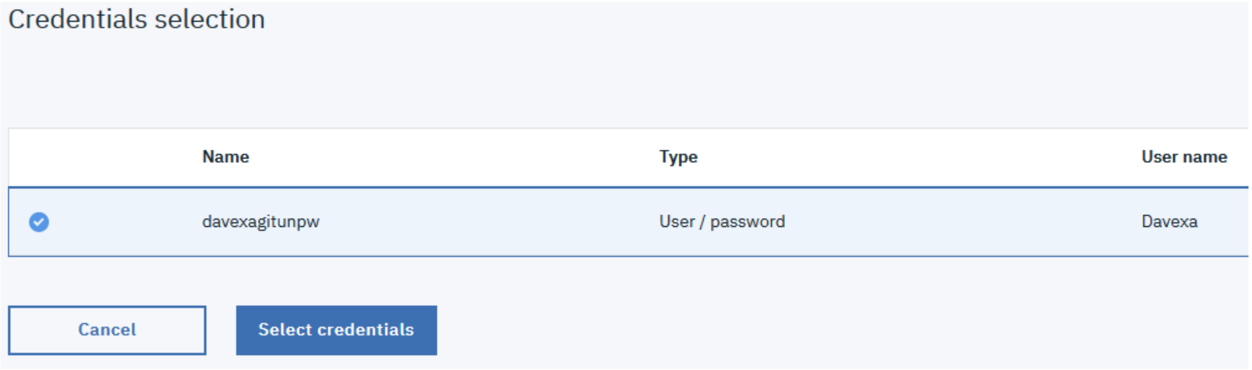
Pipeline name: aceonicpintstdopimg

Repository location: https://github.com/DAVEXACOM/ACEonICPIntStdOpImg

Credentials

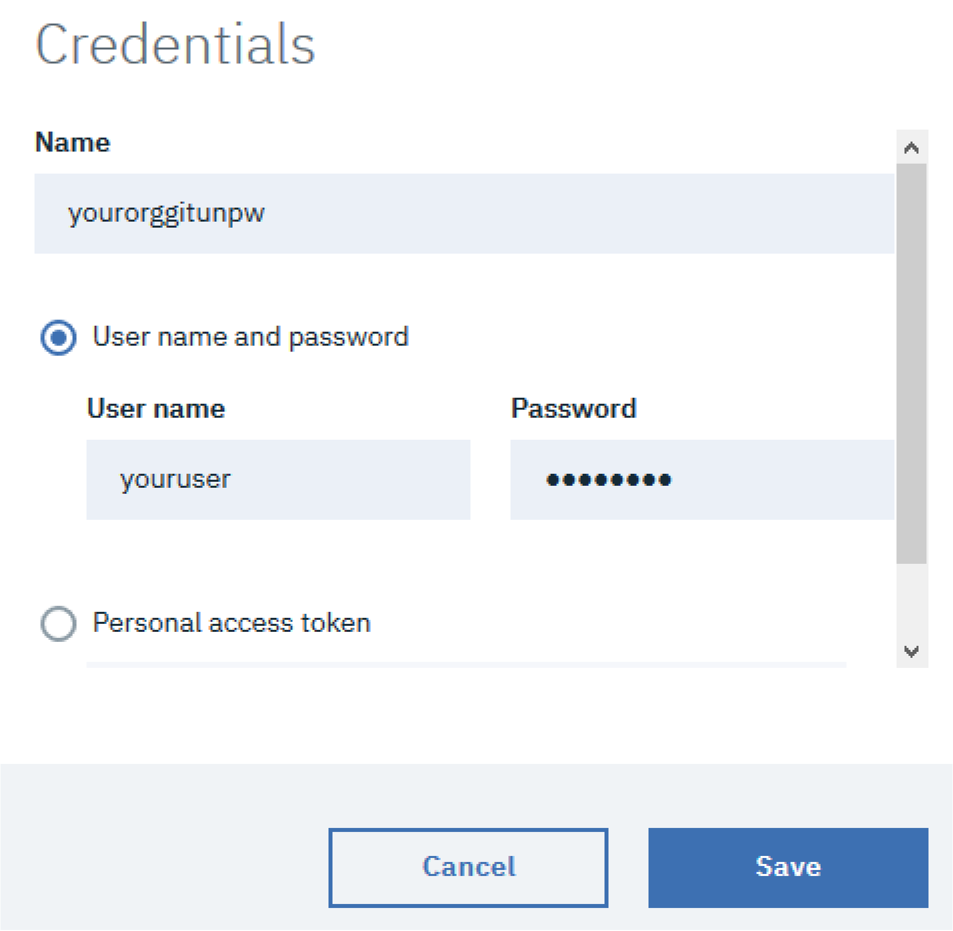
Credentials

If you already have credentials to your Github organization created use them


Credentials selection

	Name	Type	User name
●	davexagitunpw	User / password	Davexa

If not click the Add and enter your credentials and save



Credentials

Name: yourorggitunpw

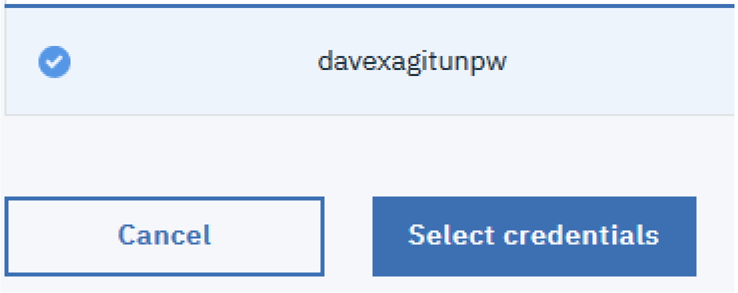
● User name and password

User name: youruser

Password: ●●●●●●●●●

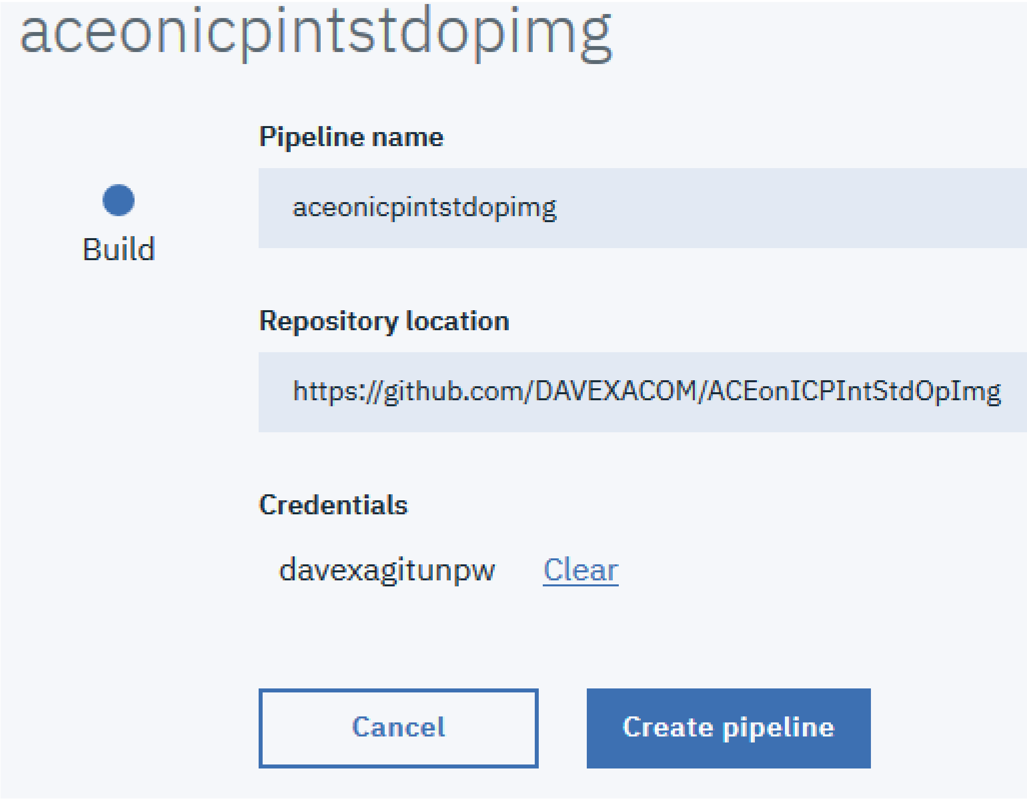
○ Personal access token

Highlight the credentials and hit select



✓ davexagitunpw

Next create the Jenkins pipeline



aceonicpintstdopimg

Pipeline name

aceonicpintstdopimg

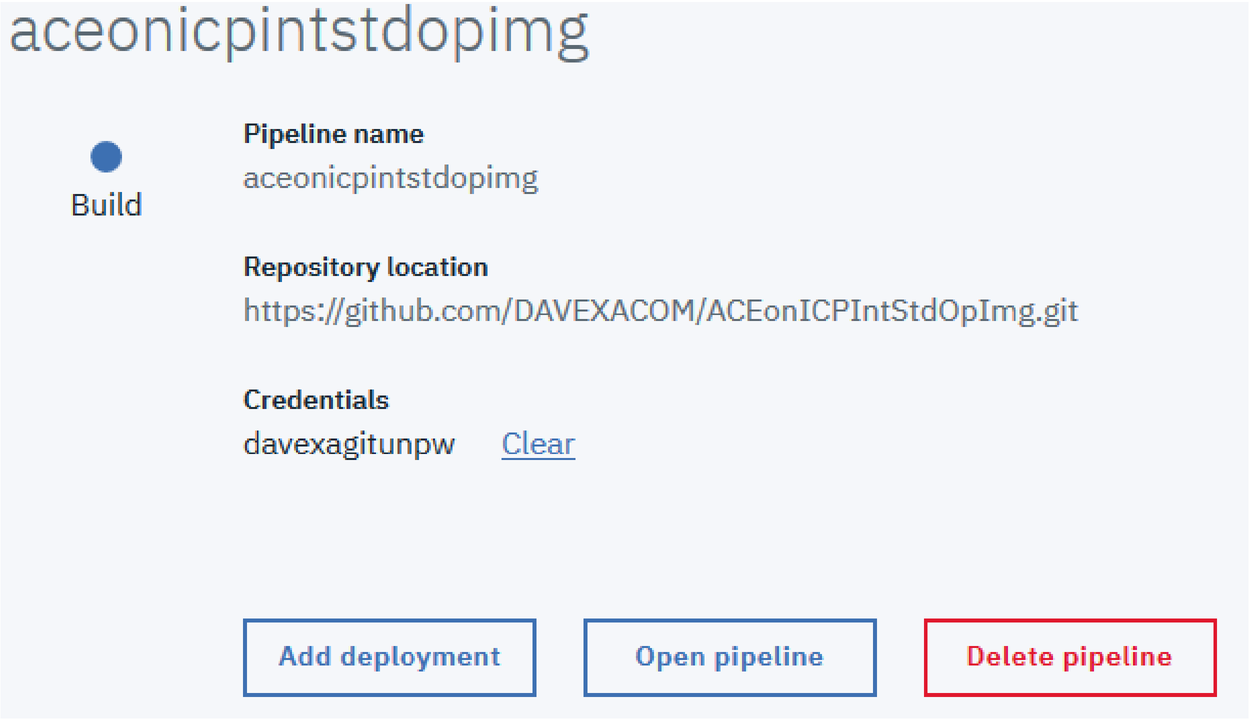
Repository location

https://github.com/DAVEXACOM/ACEonICPIntStdOpImg

Credentials

davexagitunpw

And then Open the pipeline



aceonicpintstdopimg

Pipeline name

aceonicpintstdopimg

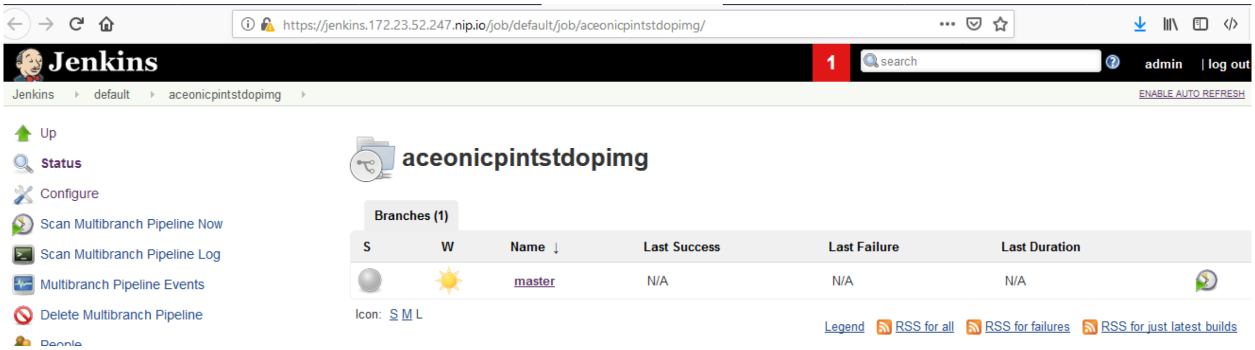
Repository location

https://github.com/DAVEXACOM/ACEonICPIntStdOpImg.git

Credentials

davexagitunpw

Jenkins will open


https://jenkins.172.23.52.247.nip.io/job/default/job/aceonicpintstdopimg/

Jenkins
1
search
admin
| log out
Jenkins › default › aceonicpintstdopimg ›
ENABLE AUTO REFRESH
Up
Status
Configure
Scan Multibranch Pipeline Now
Scan Multibranch Pipeline Log
Multibranch Pipeline Events
Delete Multibranch Pipeline
People

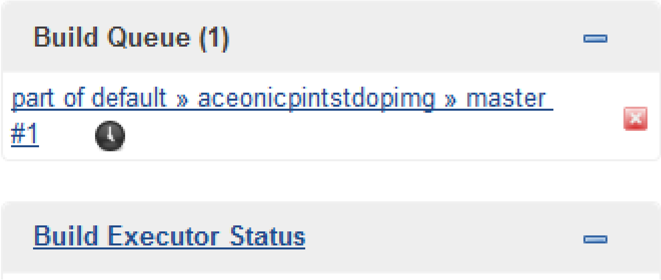
aceonicpintstdopimg

Branches (1)

S	W	Name ↓	Last Success	Last Failure	Last Duration
●	☀	master	N/A	N/A	N/A

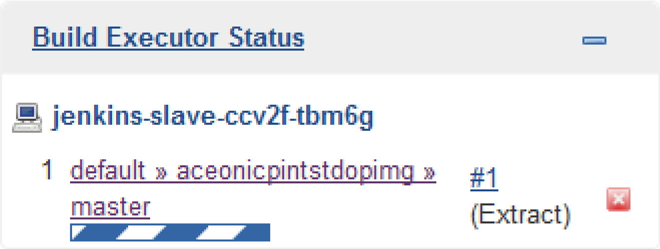
Icon: S M L
Legend
RSS for all
RSS for failures
RSS for just latest builds

And you’ll see an initial build is initiated


Build Queue (1)

part of default » aceonicpintstdopimg » master #1

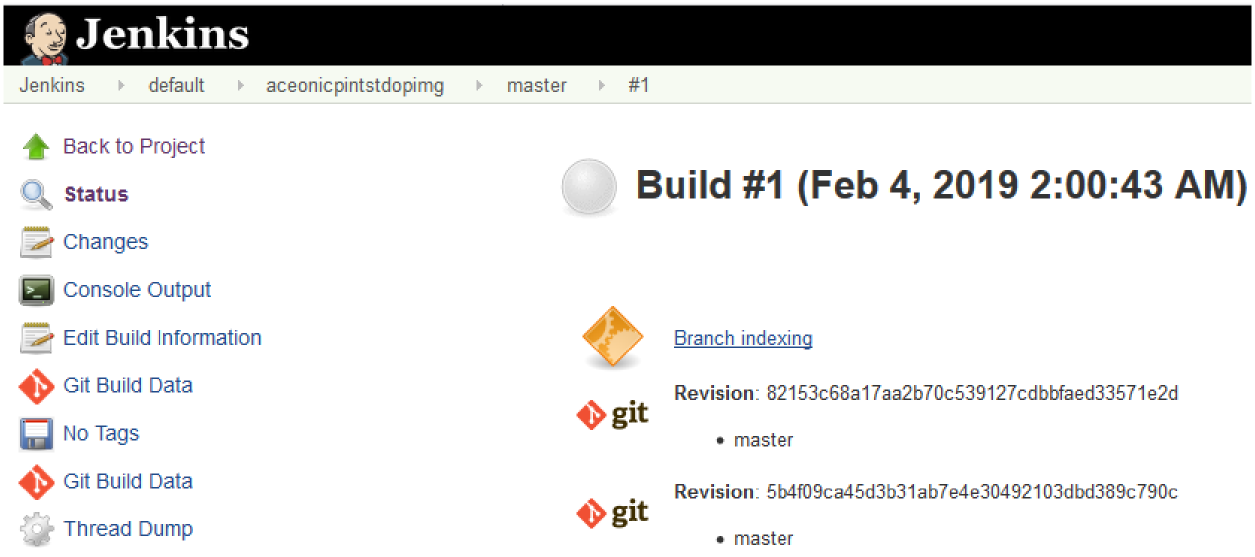
Click on #1



jenkins-slave-ccv2f-tbm6g

1 default » aceonicpintstdopimg » master #1 (Extract)

And review the console log to see all is underway and working



Build #1 (Feb 4, 2019 2:00:43 AM)

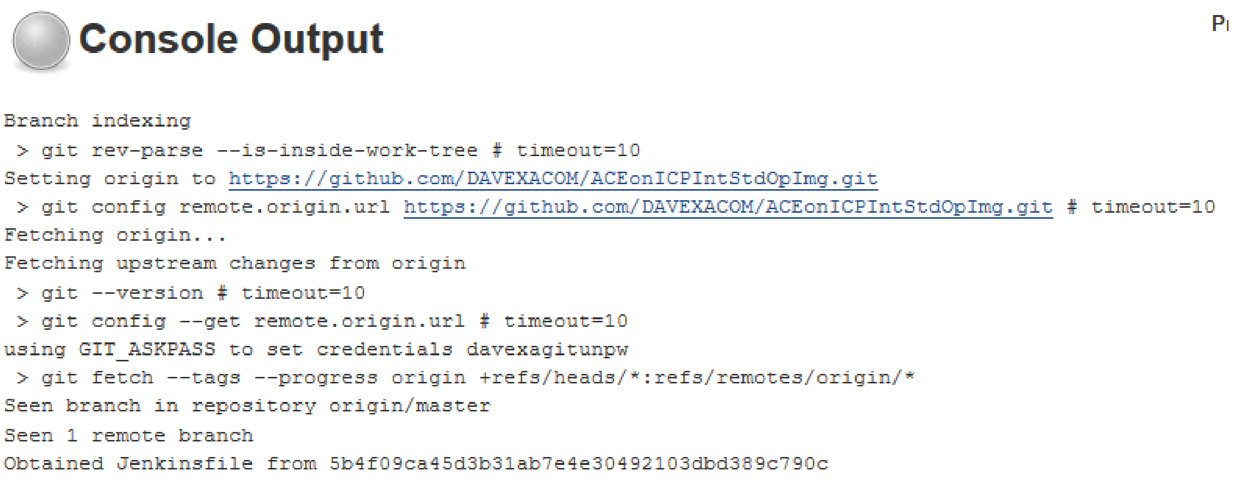
Branch indexing

Revision: 82153c68a17aa2b70c539127cdbbfaed33571e2d

- master

Revision: 5b4f09ca45d3b31ab7e4e30492103dbd389c790c

- master



```
Branch indexing
 > git rev-parse --is-inside-work-tree # timeout=10
Setting origin to https://github.com/DAVEXACOM/ACEonICPIntStdOpImg.git
 > git config remote.origin.url https://github.com/DAVEXACOM/ACEonICPIntStdOpImg.git # timeout=10
Fetching origin...
Fetching upstream changes from origin
 > git --version # timeout=10
 > git config --get remote.origin.url # timeout=10
using GIT_ASKPASS to set credentials davexagitunpw
 > git fetch --tags --progress origin +refs/heads/*:refs/remotes/origin/*
Seen branch in repository origin/master
Seen 1 remote branch
Obtained Jenkinsfile from 5b4f09ca45d3b31ab7e4e30492103dbd389c790c
```

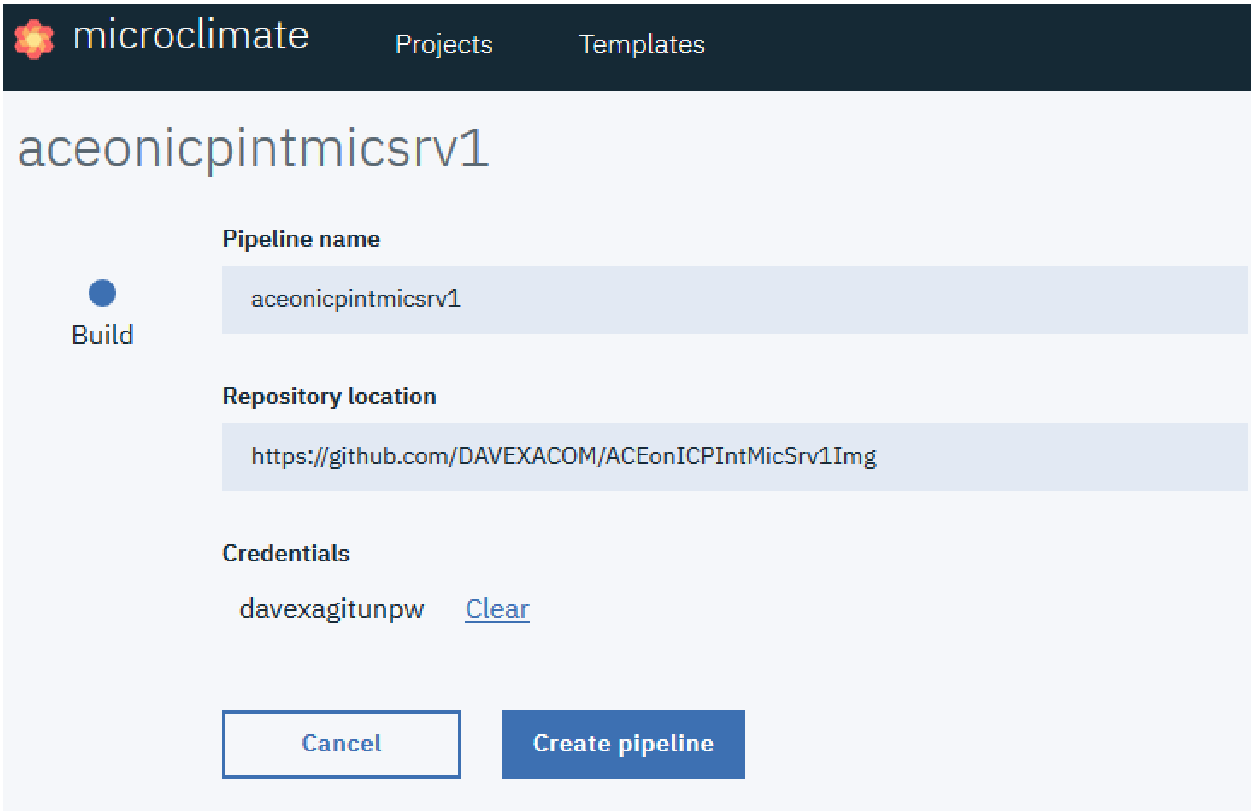
ACE on ICP Integration Micro Service 1

Now you must repeat the process for the integration micro service 1 image using the following:

Naming convention : aceonicpintmicsrv1

Github repository: https://github.com/DAVEXACOM/ACEonICPIntMicSrv1Img

This is the image build pipeline for Integration Microservice 1.



Pipeline name

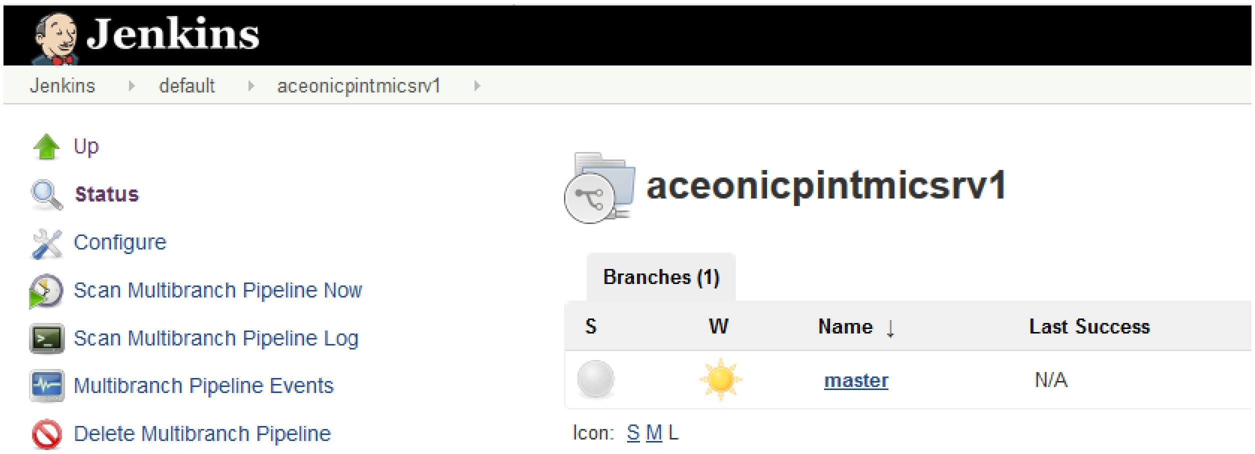
aceonicpintmicsrv1

Repository location

https://github.com/DAVEXACOM/ACEonICPIntMicSrv1Img

Credentials

davexagitunpw


Jenkins
Jenkins
default
aceonicpintmicsrv1
Up
Status
Configure
Scan Multibranch Pipeline Now
Scan Multibranch Pipeline Log
Multibranch Pipeline Events
Delete Multibranch Pipeline
aceonicpintmicsrv1
Branches (1)
Icon: S M L

S	W	Name ↓	Last Success
○	☀	master	N/A

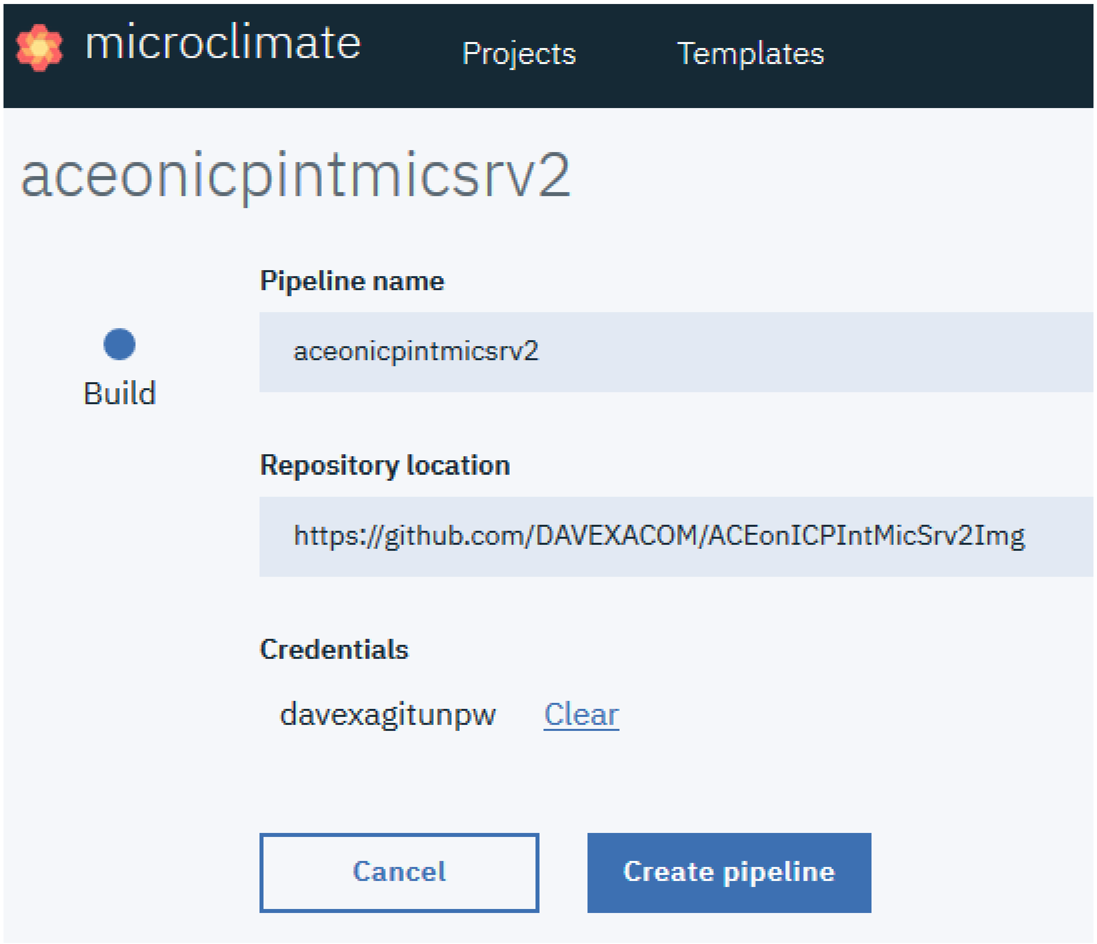
ACE on ICP Integration Micro Service 2

Now you must repeat the process for the integration micro service 2 image using the following:

Naming convention : aceonicpintmicsrv2

Github repository: https://github.com/DAVEXACOM/ACEonICPIntMicSrv2Img

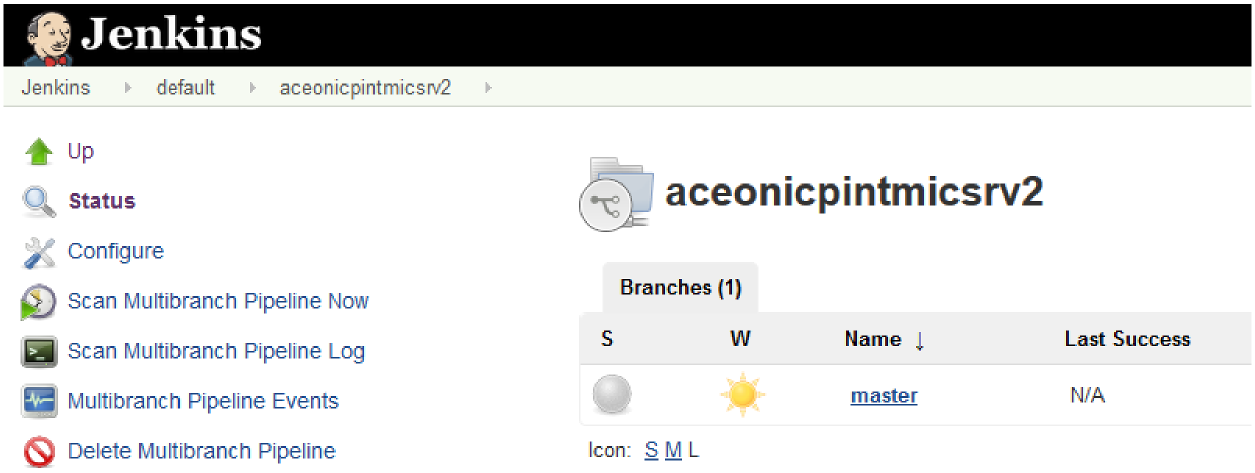
This is the image build pipeline for Integration Microservice 2.



Pipeline name: aceonicpintmicsrv2

Repository location: https://github.com/DAVEXACOM/ACEonICPIntMicSrv2Img

Credentials: davexagitunpw


Jenkins
Jenkins
default
aceonicpintmicsrv2
Up
Status
Configure
Scan Multibranch Pipeline Now
Scan Multibranch Pipeline Log
Multibranch Pipeline Events
Delete Multibranch Pipeline
Branches (1)
Icon: S M L

aceonicpintmicsrv2

S	W	Name ↓	Last Success
○	☀	master	N/A

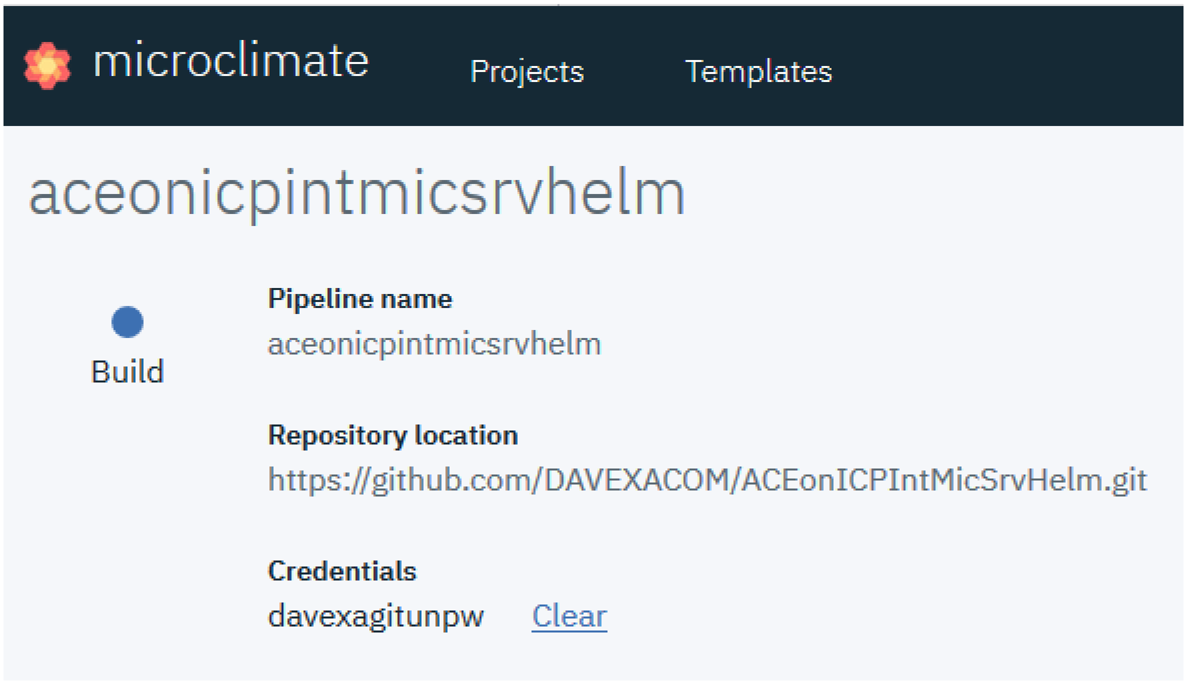
ACE on ICP Integration Micro Service Helm Release – Individual services

Now you must repeat the process for the integration micro service helm release pipeline for individual services using the following:

Naming convention : aceonicpintmicsrvhelm

Github repository: https://github.com/DAVEXACOM/ACEonICPIntMicSrvHelm

This is the Helm Release pipeline that will allow you to deploy either Integration Microservice 1 OR Integration Microservice 2 individually for unit test purposes.

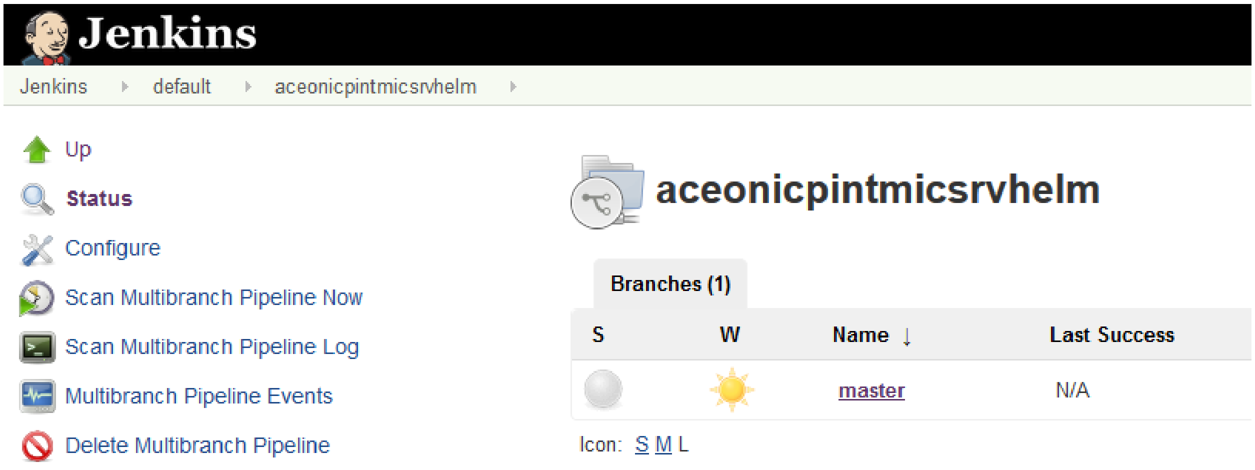


aceonicpintmicsrvhelm

Pipeline name
aceonicpintmicsrvhelm

Repository location
https://github.com/DAVEXACOM/ACEonICPIntMicSrvHelm.git

Credentials
davexagitunpw


Jenkins
Jenkins › default › aceonicpintmicsrvhelm ›
Up
Status
Configure
Scan Multibranch Pipeline Now
Scan Multibranch Pipeline Log
Multibranch Pipeline Events
Delete Multibranch Pipeline
Icon: S M L

aceonicpintmicsrvhelm

Branches (1)

S	W	Name ↓	Last Success
○	☀	master	N/A

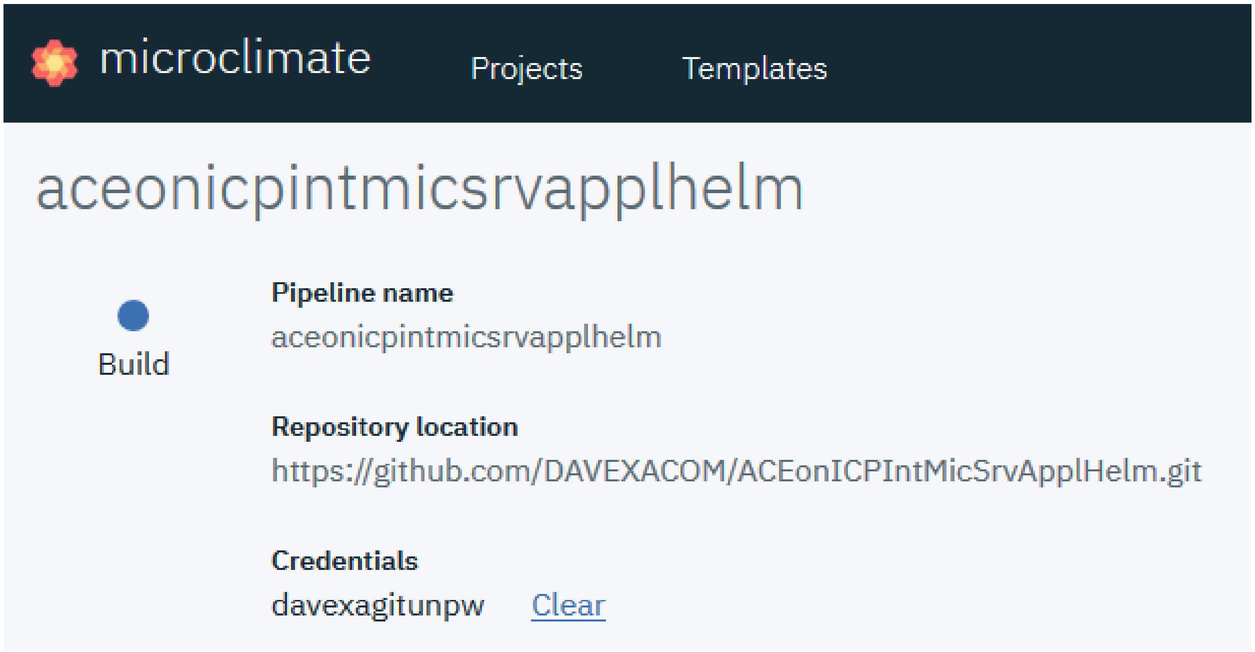
ACE on ICP Integration Micro Services Application Helm Release – Application Release

Now you must repeat the process for the integration micro services application helm release pipeline to deploy all services as an application using the following:

Naming convention : aceonicpintmicsrvapplhelm

Github repository: https://github.com/DAVEXACOM/ACEonICPIntMicSrvApplHelm

This is the Helm Release pipeline that will allow you to deploy both Integration Microservice 1 AND Integration Microservice 2 as an application.

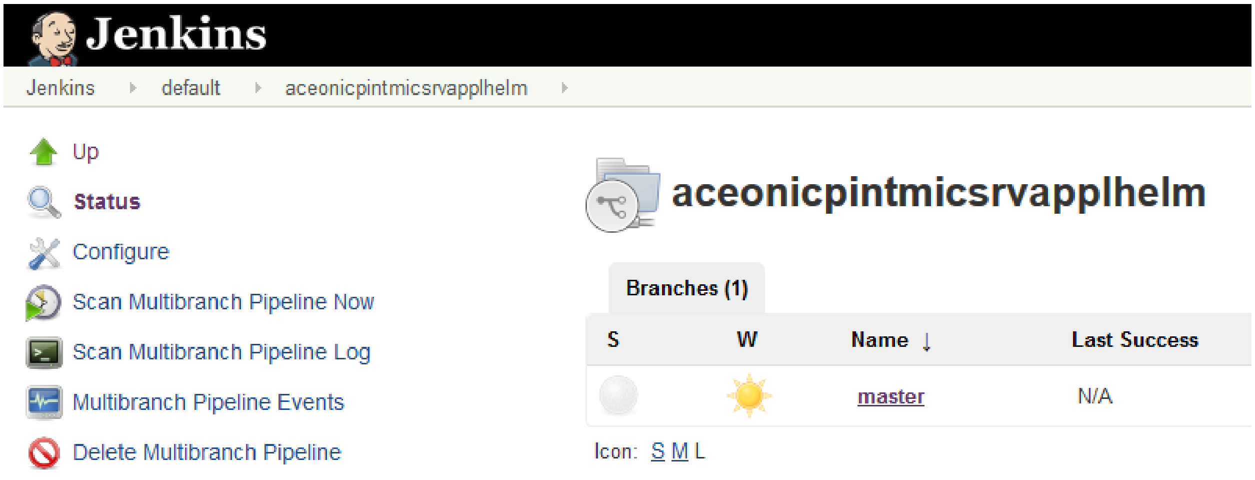


aceonicpintmicsrvapplhelm

Pipeline name
aceonicpintmicsrvapplhelm

Repository location
https://github.com/DAVEXACOM/ACEonICPIntMicSrvApplHelm.git

Credentials
davexagitunpw


Jenkins
Jenkins ▸ default ▸ aceonicpintmicsrvapplhelm ▸
Up
Status
Configure
Scan Multibranch Pipeline Now
Scan Multibranch Pipeline Log
Multibranch Pipeline Events
Delete Multibranch Pipeline

aceonicpintmicsrvapplhelm

Branches (1)

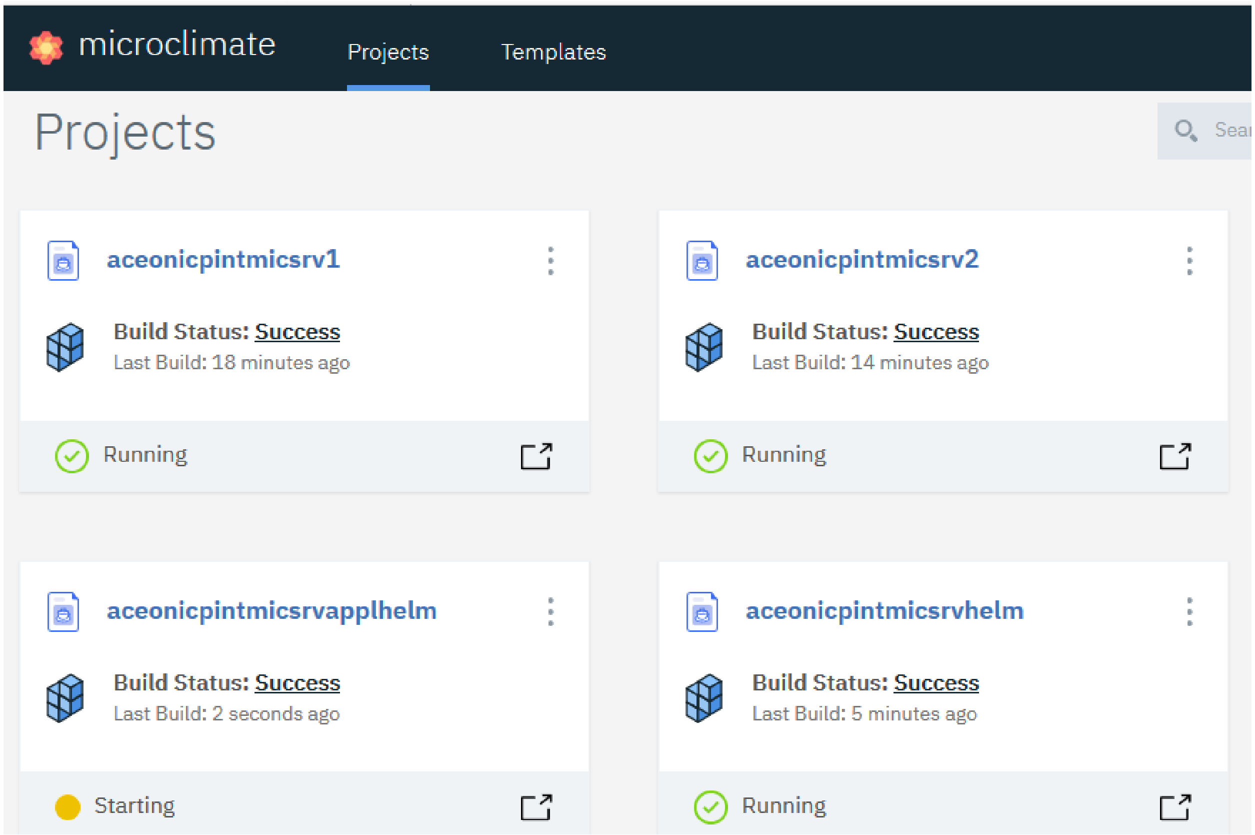
S	W	Name ↓	Last Success
○	☀	master	N/A

Icon: S M L

Review of what’s been created on ICP

Microclimate Projects

You should now have 5 Microclimate projects

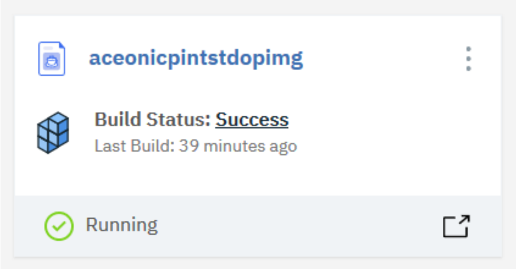


aceonicpintmicsrv1
Build Status: Success
Last Build: 18 minutes ago
Running

aceonicpintmicsrv2
Build Status: Success
Last Build: 14 minutes ago
Running

aceonicpintmicsrvapplhelm
Build Status: Success
Last Build: 2 seconds ago
Starting

aceonicpintmicsrvhelm
Build Status: Success
Last Build: 5 minutes ago
Running



aceonicpintstdopimg

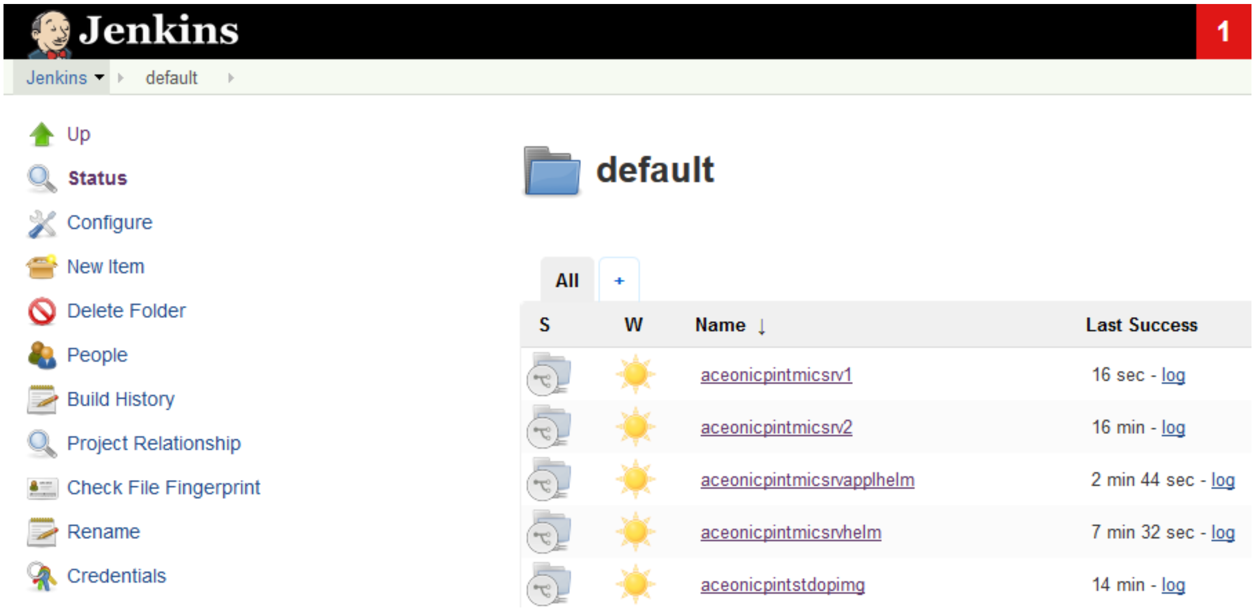
Build Status: Success

Last Build: 39 minutes ago

Running

Jenkins pipelines

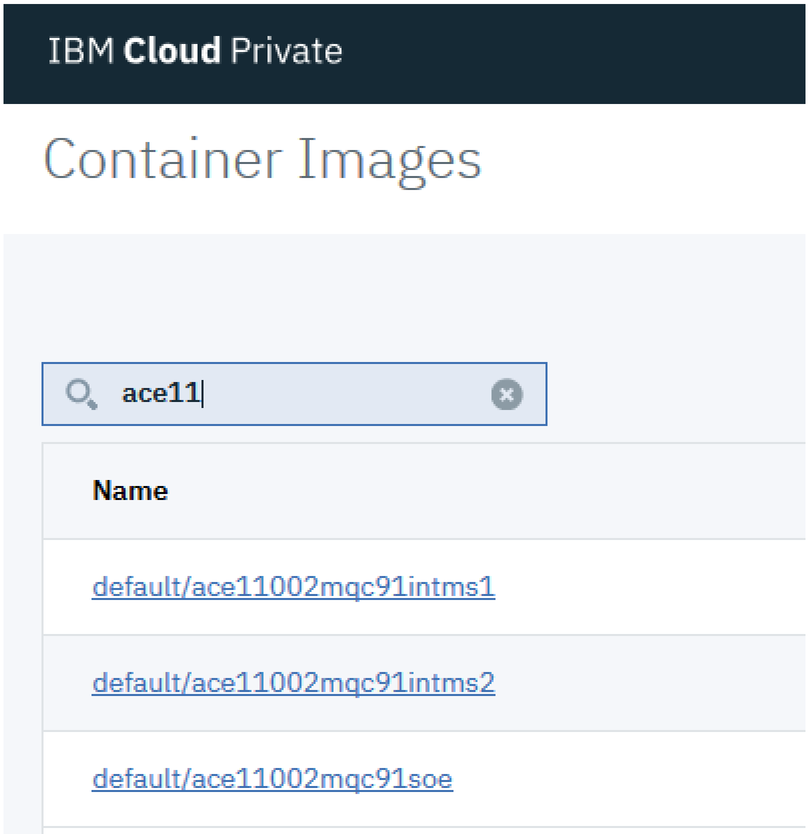
And five Jenkins pipelines



default

S	W	Name ↓	Last Success
	☀	aceonicpintmicsrv1	16 sec - log
	☀	aceonicpintmicsrv2	16 min - log
	☀	aceonicpintmicsrvapplhelm	2 min 44 sec - log
	☀	aceonicpintmicsrvhelm	7 min 32 sec - log
	☀	aceonicpintstdopimg	14 min - log

ICP Image repository images



IBM **Cloud** Private

Container Images

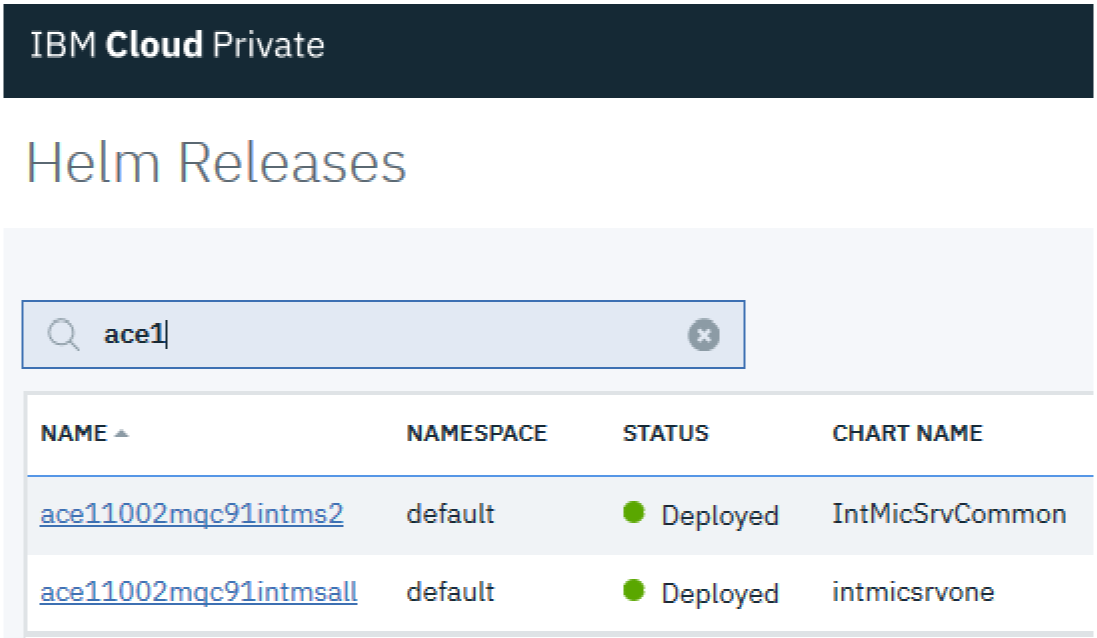
ace11

Name
default/ace11002mqc91intms1
default/ace11002mqc91intms2
default/ace11002mqc91soe

ICP Helm releases

The main purpose of the environment creation step was to get the images onto ICP. However,

We can see that in creating to two Helm release pipelines, they have been executed and the release deployed.



IBM **Cloud** Private

Helm Releases

ace1

NAME	NAMESPACE	STATUS	CHART NAME
ace11002mqc91intms2	default	● Deployed	IntMicSrvCommon
ace11002mqc91intmsall	default	● Deployed	intmicsrvone

Exploring the initial helm releases on ICP

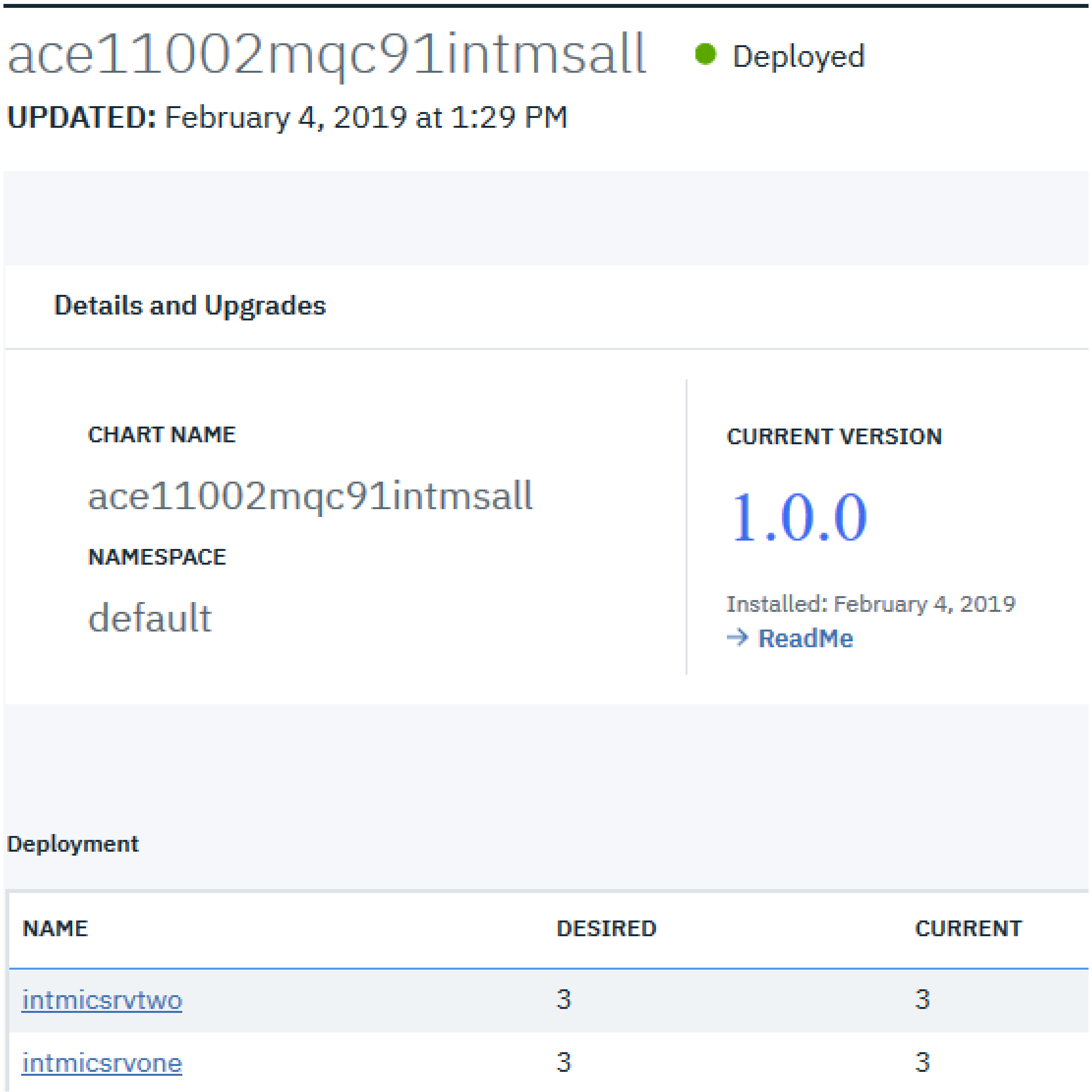
Integration Micros Services Application helm release

From the ICP console select Helm Releases

Filter on ace and select the Helm release that deploys multiple integration micro services

ace11002mqc91intmsall

Note the 2 Integration micro services are deployed and each is highly available with 3 instances



ace11002mqc91intmsall ● Deployed

UPDATED: February 4, 2019 at 1:29 PM

Details and Upgrades

CHART NAME

ace11002mqc91intmsall

NAMESPACE

default

CURRENT VERSION

1.0.0

Installed: February 4, 2019

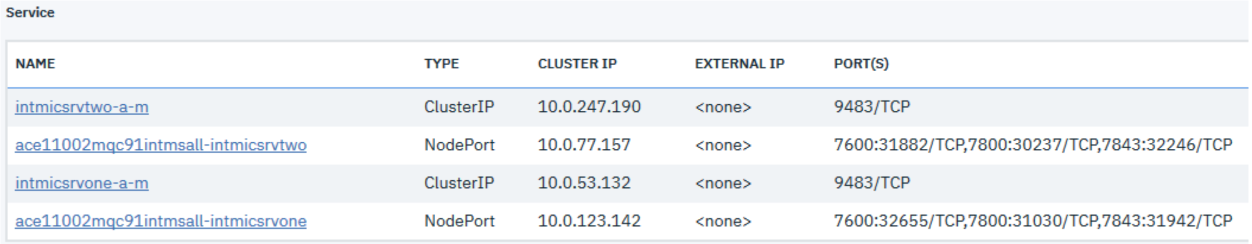
→ ReadMe

Deployment

NAME	DESIRED	CURRENT
intmicsrvtwo	3	3
intmicsrvone	3	3

Explore Integration micro service 2

Scroll down to the services and select integration micro service two

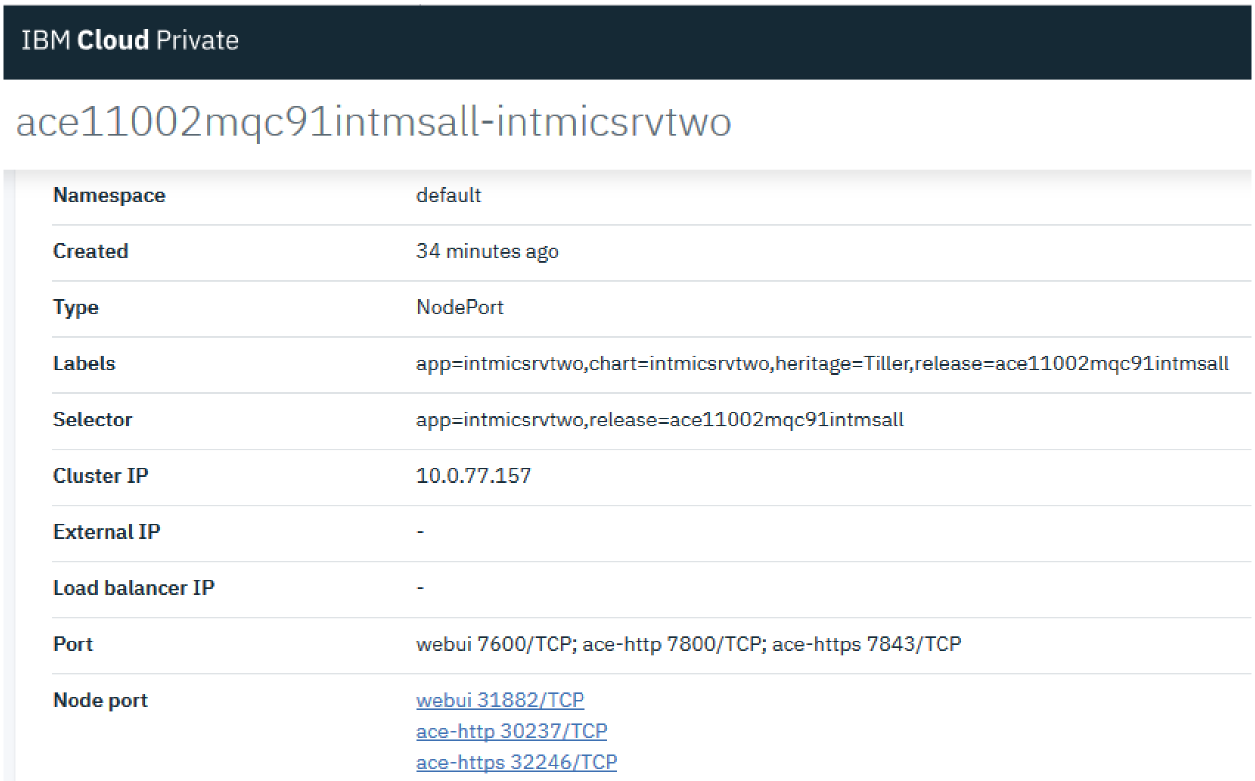


Service

NAME	TYPE	CLUSTER IP	EXTERNAL IP	PORT(S)
intmicsrvtwo-a-m	ClusterIP	10.0.247.190	<none>	9483/TCP
ace11002mqc91intmsall-intmicsrvtwo	NodePort	10.0.77.157	<none>	7600:31882/TCP,7800:30237/TCP,7843:32246/TCP
intmicsrvone-a-m	ClusterIP	10.0.53.132	<none>	9483/TCP
ace11002mqc91intmsall-intmicsrvone	NodePort	10.0.123.142	<none>	7600:32655/TCP,7800:31030/TCP,7843:31942/TCP

Click on the link

ace11002mqc91intmsall-intmicsrvtwo

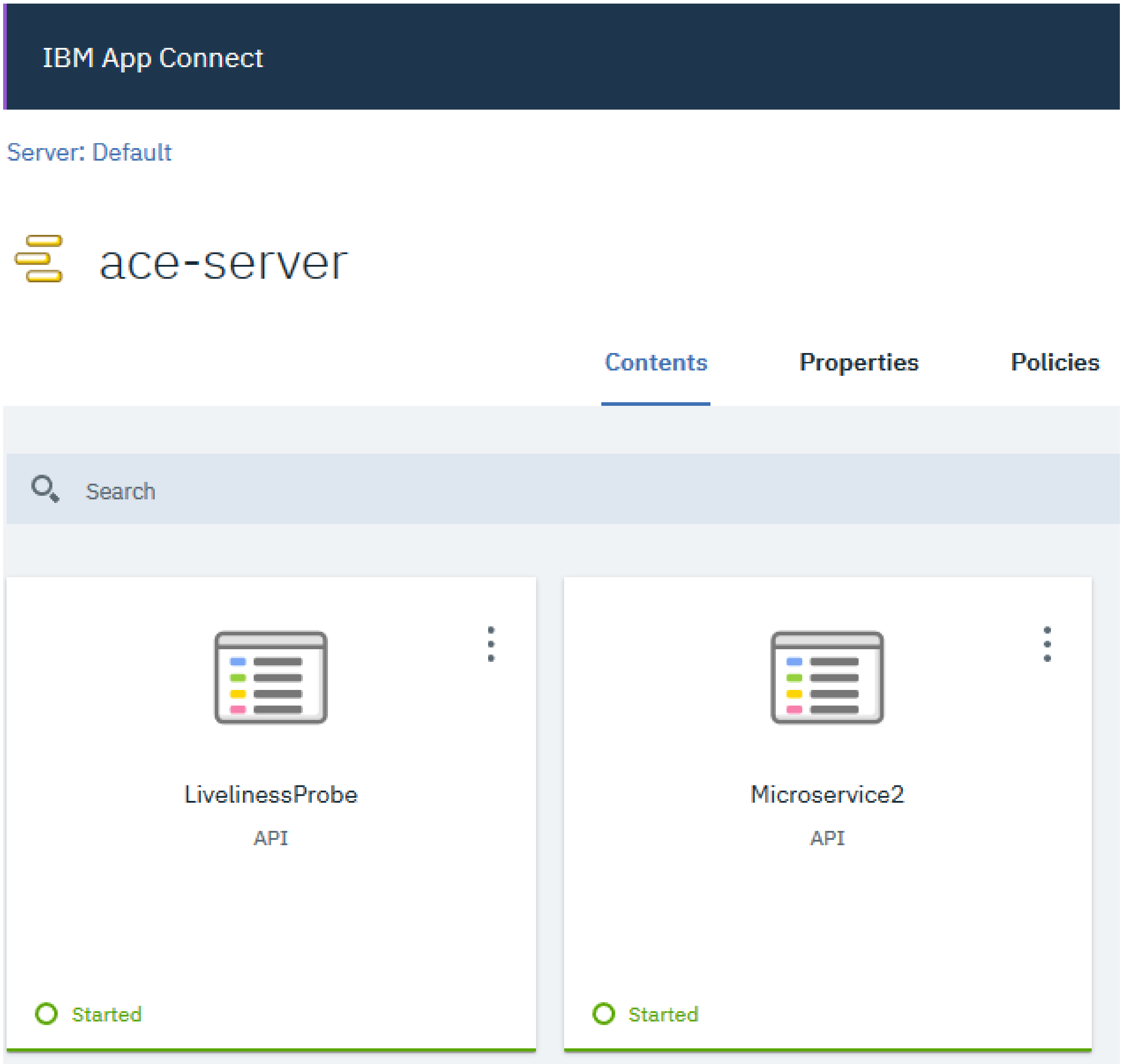


IBM **Cloud** Private

ace11002mqc91intmsall-intmicsrvtwo

Namespace	default
Created	34 minutes ago
Type	NodePort
Labels	app=intmicsrvtwo,chart=intmicsrvtwo,heritage=Tiller,release=ace11002mqc91intmsall
Selector	app=intmicsrvtwo,release=ace11002mqc91intmsall
Cluster IP	10.0.77.157
External IP	-
Load balancer IP	-
Port	webui 7600/TCP; ace-http 7800/TCP; ace-https 7843/TCP
Node port	webui 31882/TCP ace-http 30237/TCP ace-https 32246/TCP

Select the WebUI link to bring up the ACE WebUI



IBM App Connect

Server: Default

ace-server

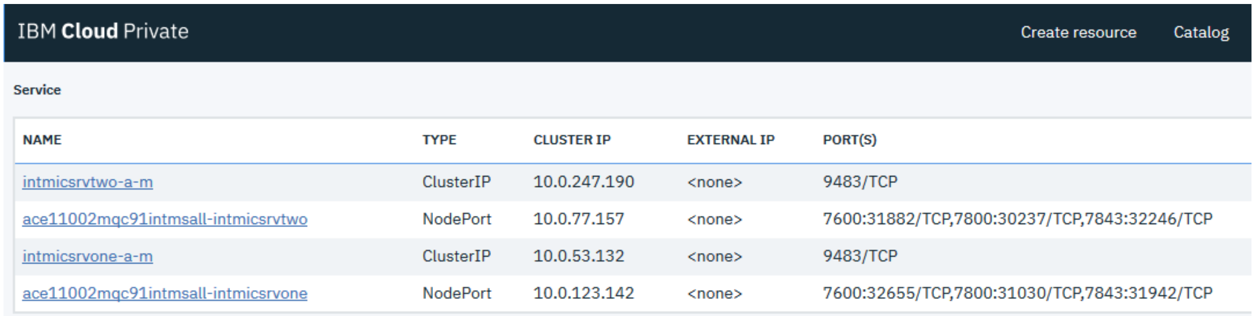
LivelinessProbe
API
Started

Microservice2
API
Started

Note Micro service two has its Microservice2 Application API service deployed and the Standard operating environments LivelinessProbe application.

Now go back to the services list for this Helm Releases

Explore Integration micro service 1



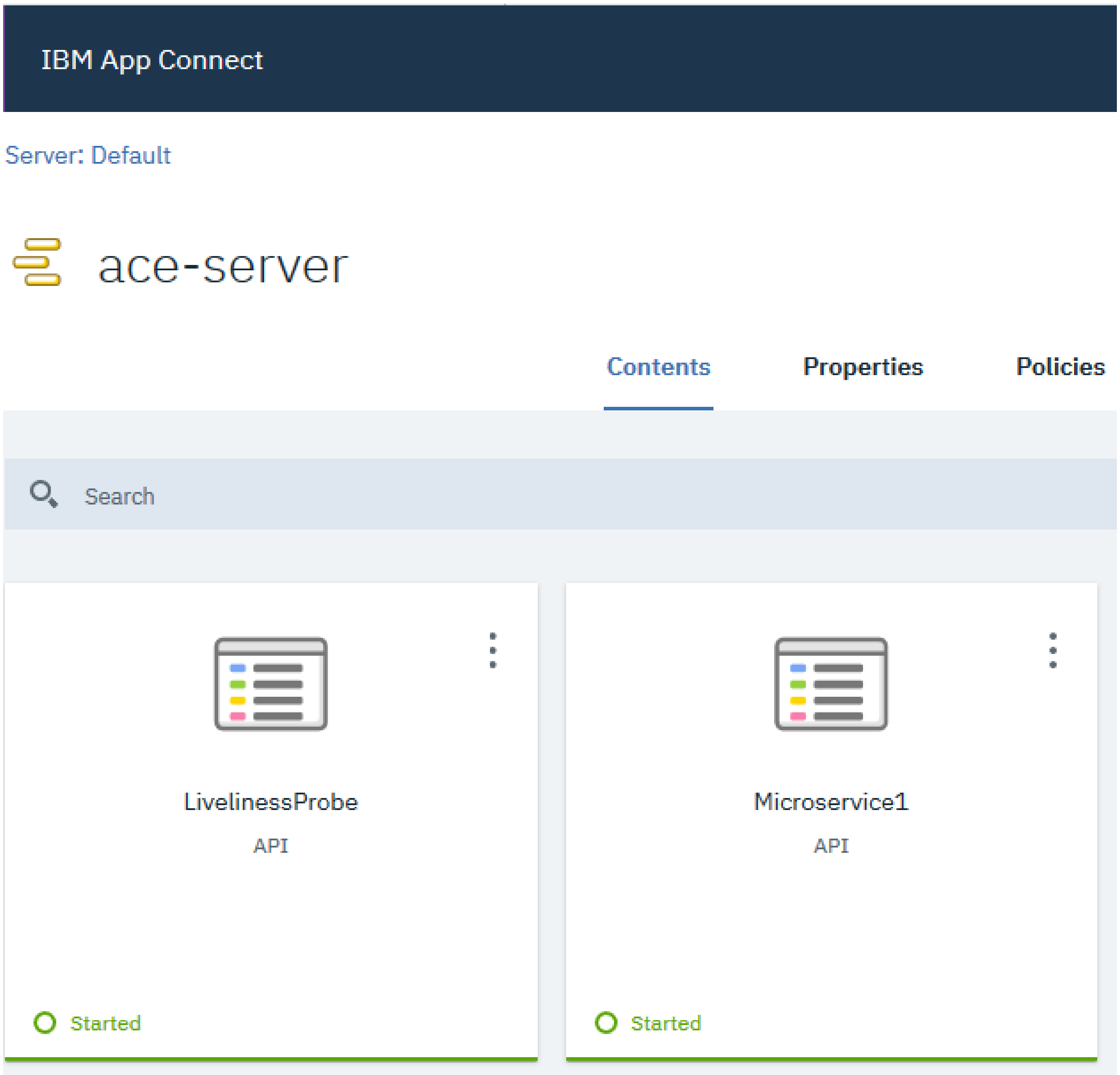
Service

NAME	TYPE	CLUSTER IP	EXTERNAL IP	PORT(S)
intmicsrvtwo-a-m	ClusterIP	10.0.247.190	<none>	9483/TCP
ace11002mqc91intmsall-intmicsrvtwo	NodePort	10.0.77.157	<none>	7600:31882/TCP,7800:30237/TCP,7843:32246/TCP
intmicsrvone-a-m	ClusterIP	10.0.53.132	<none>	9483/TCP
ace11002mqc91intmsall-intmicsrvone	NodePort	10.0.123.142	<none>	7600:32655/TCP,7800:31030/TCP,7843:31942/TCP

and follow the above steps to check out micro service 1 following the service link

ace11002mqc91intmsall-intmicsrvone

Bring up the ACE WebUI



Server: Default

ace-server

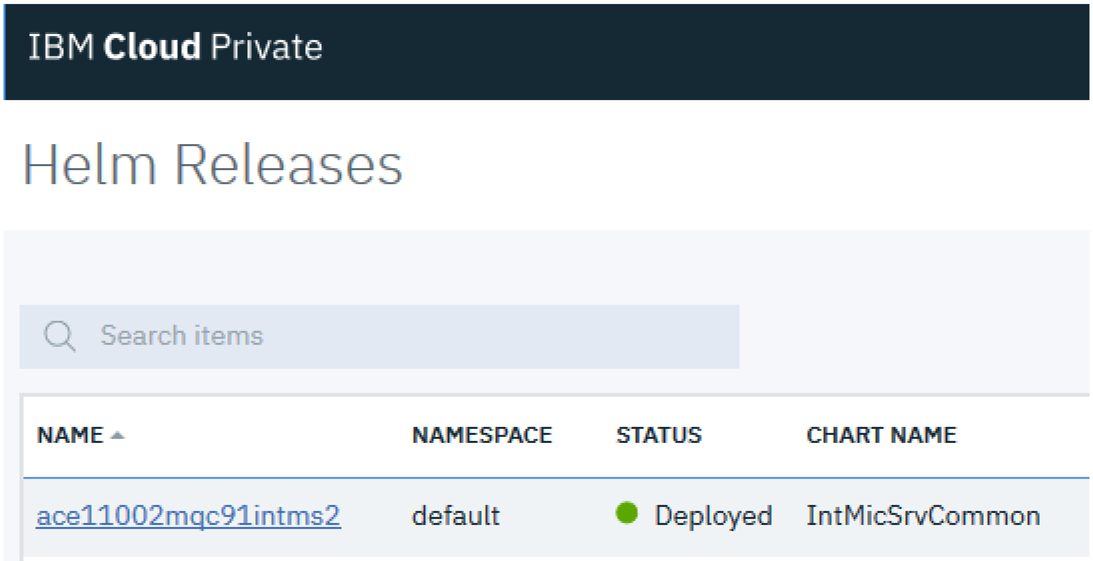
LivelinessProbe
API
Started

Microservice1
API
Started

Note Micro service two has its Microservice2 Application API service deployed and the Standard operating environments LivelinessProbe application.

Individual Integration micros service helm release

From ICP Console select Workloads->Helm Releases

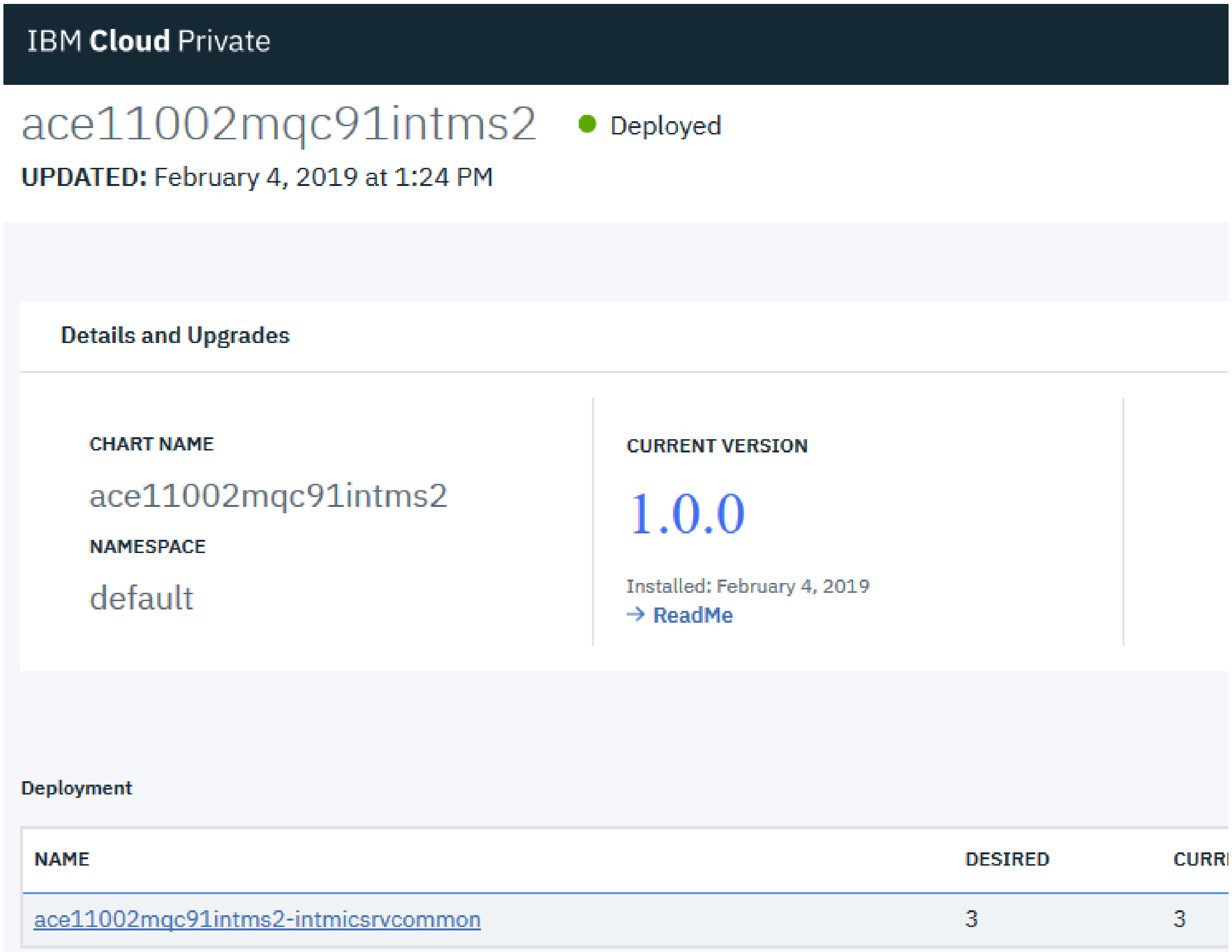


Helm Releases

NAME	NAMESPACE	STATUS	CHART NAME
ace11002mqc91intms2	default	● Deployed	IntMicSrvCommon

Filter on ace and select

ace11002mqc91intms2



IBM **Cloud** Private

ace11002mqc91intms2 ● Deployed

UPDATED: February 4, 2019 at 1:24 PM

Details and Upgrades

CHART NAME

ace11002mqc91intms2

NAMESPACE

default

CURRENT VERSION

1.0.0

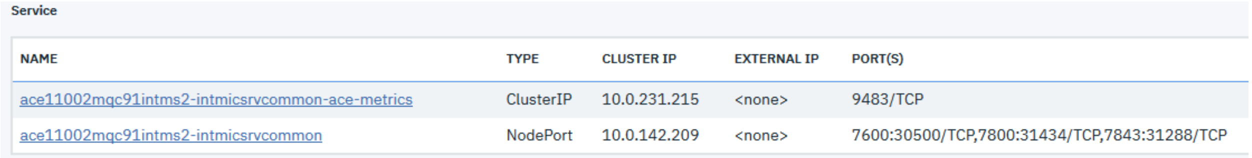
Installed: February 4, 2019

→ ReadMe

Deployment

NAME	DESIRED	CURR
ace11002mqc91intms2-intmicsrvcommon	3	3

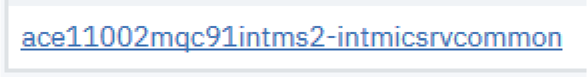
Scroll down to the services



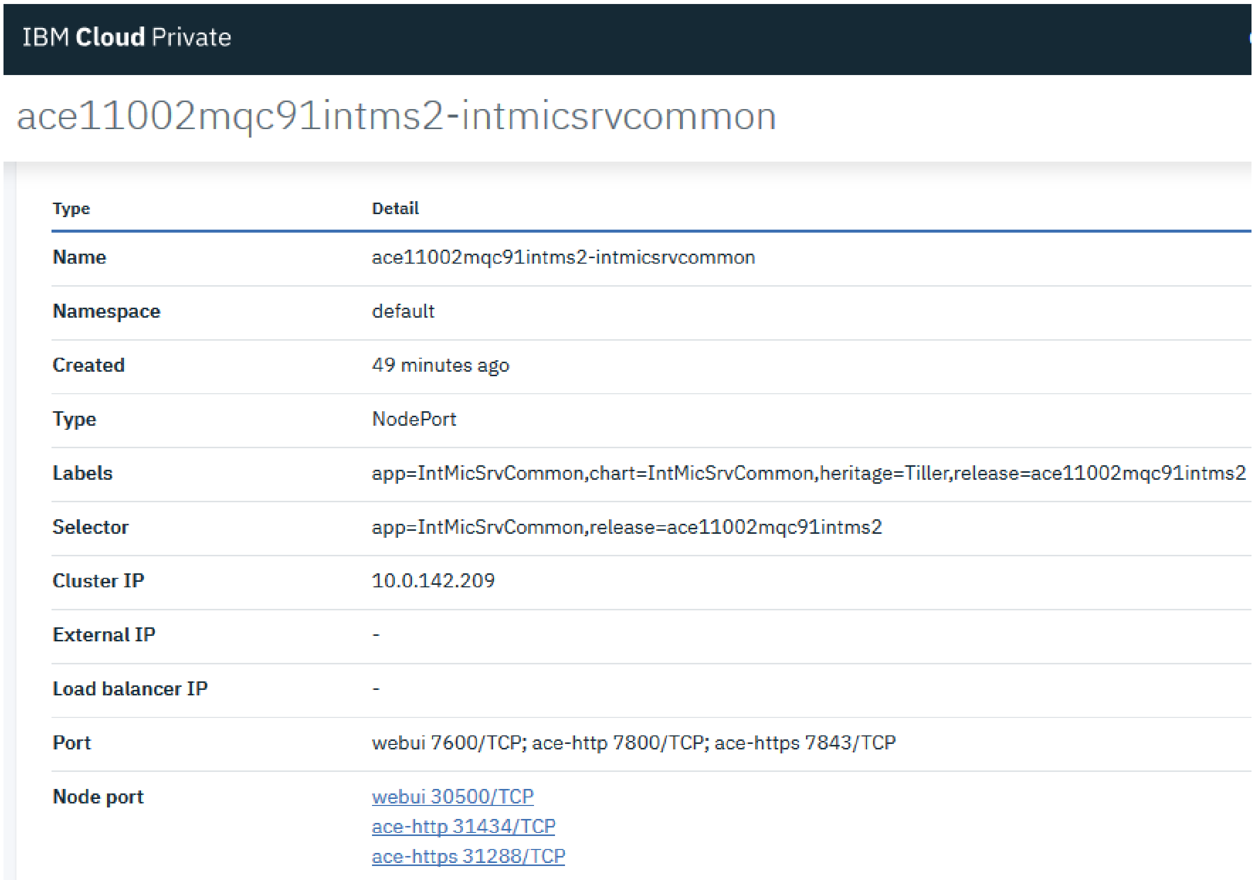
Service

NAME	TYPE	CLUSTER IP	EXTERNAL IP	PORT(S)
ace11002mqc91intms2-intmicsrvcommon-ace-metrics	ClusterIP	10.0.231.215	<none>	9483/TCP
ace11002mqc91intms2-intmicsrvcommon	NodePort	10.0.142.209	<none>	7600:30500/TCP,7800:31434/TCP,7843:31288/TCP

Select integration micro service 2 link



ace11002mqc91intms2-intmicsrvcommon

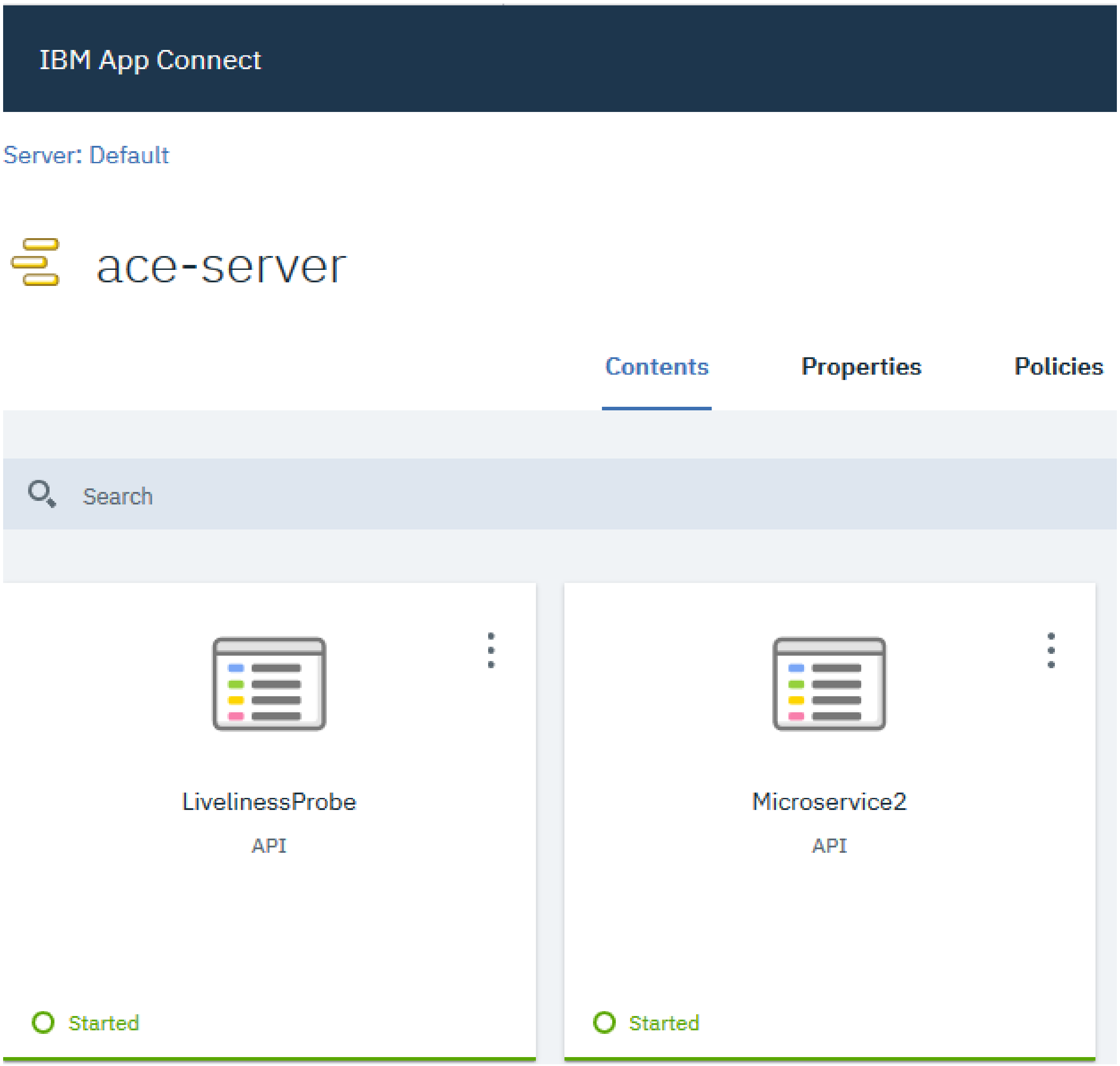


IBM **Cloud** Private

ace11002mqc91intms2-intmicsrvcommon

Type	Detail
Name	ace11002mqc91intms2-intmicsrvcommon
Namespace	default
Created	49 minutes ago
Type	NodePort
Labels	app=IntMicSrvCommon,chart=IntMicSrvCommon,heritage=Tiller,release=ace11002mqc91intms2
Selector	app=IntMicSrvCommon,release=ace11002mqc91intms2
Cluster IP	10.0.142.209
External IP	-
Load balancer IP	-
Port	webui 7600/TCP; ace-http 7800/TCP; ace-https 7843/TCP
Node port	webui 30500/TCP ace-http 31434/TCP ace-https 31288/TCP

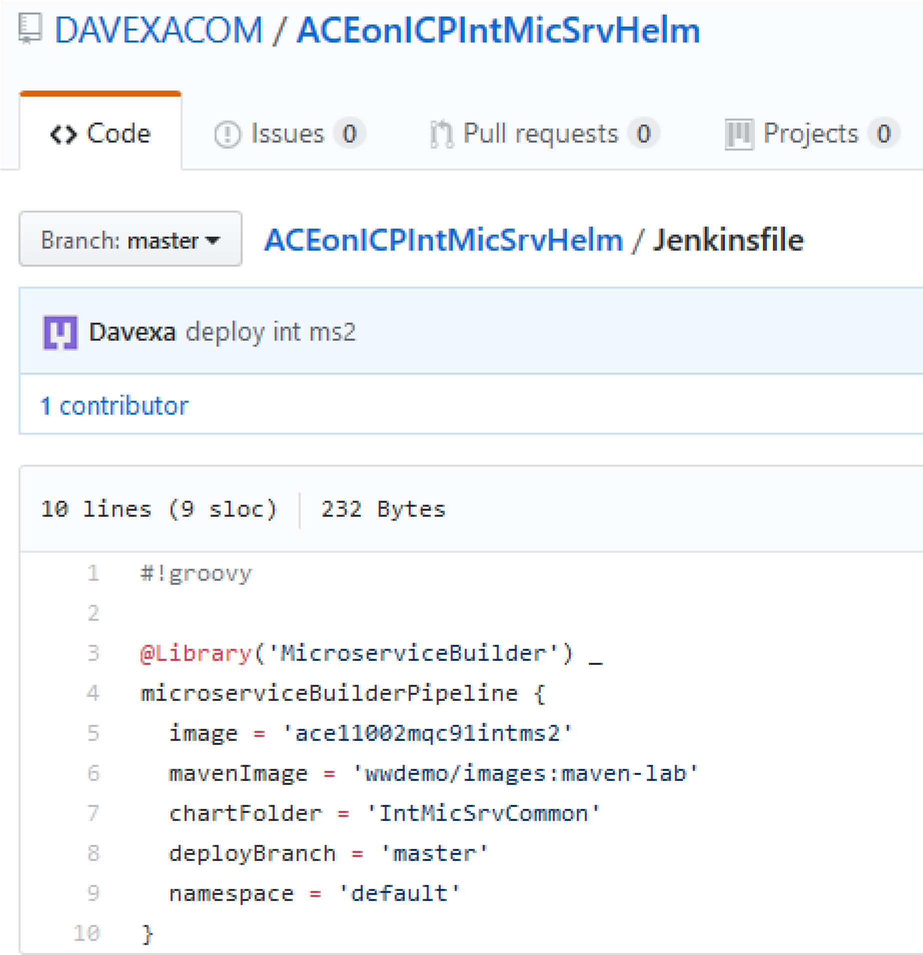
Select the webUI link to bring up the ACE WebUI



Note Micro service two has its Microservice2 Application API service deployed and the Standard operating environments LivelinessProbe application.

The individual microservice2 was deployed because at the time the initial creation of the artifacts on ICP via the Jenkins pipeline referring to https://github.com/DAVEXACOM/ACEonICPIntMicSrvHelm

The Jenkins file was set to



DAVEXACOM / ACEonICPIntMicSrvHelm

ACEonICPIntMicSrvHelm / Jenkinsfile

Davexa deploy int ms2

1 contributor

10 lines (9 sloc) | 232 Bytes

```
#!groovy

@Library('MicroserviceBuilder') _
microserviceBuilderPipeline {
  image = 'ace11002mqc91intms2'
  mavenImage = 'wwdemo/images:maven-lab'
  chartFolder = 'IntMicSrvCommon'
  deployBranch = 'master'
  namespace = 'default'
}
```

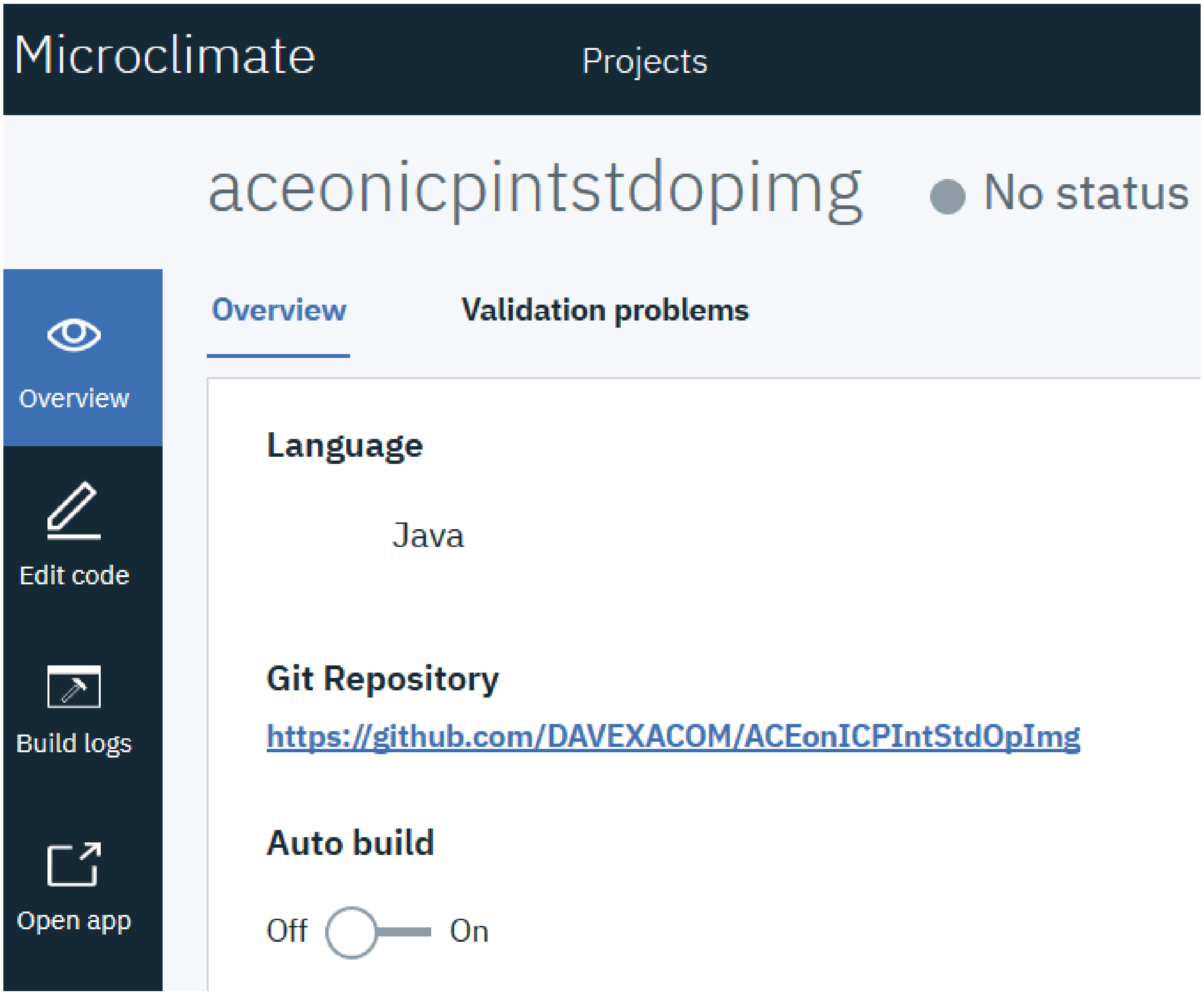
Running the Pipelines manually to load the images onto ICP

Building the Image for SoE – GitHub ACE SoE Build

Source Github repository

https://github.com/DAVEXACOM/ACEonICPIntStdOpImg

Microclimate SOE Project



aceonicpintstdopimg ● No status

Language

Java

Git Repository

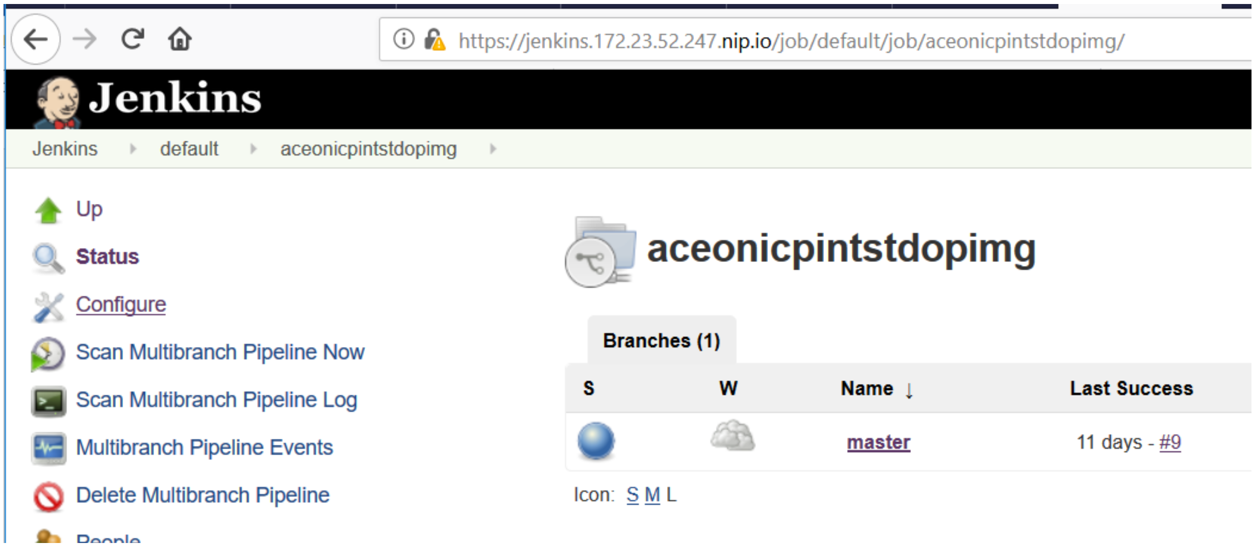
https://github.com/DAVEXACOM/ACEonICPIntStdOpImg

Auto build

Off ○ On

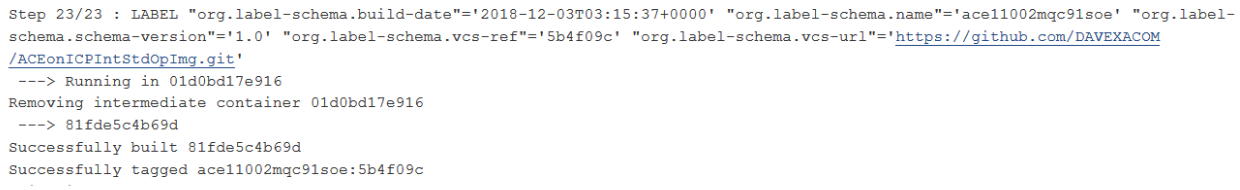
```
aceonicpintstdopimg
https://github.com/DAVEXACOM/ACEonICPIntStdOpImg.git
```

Jenkins Pipeline for SoE

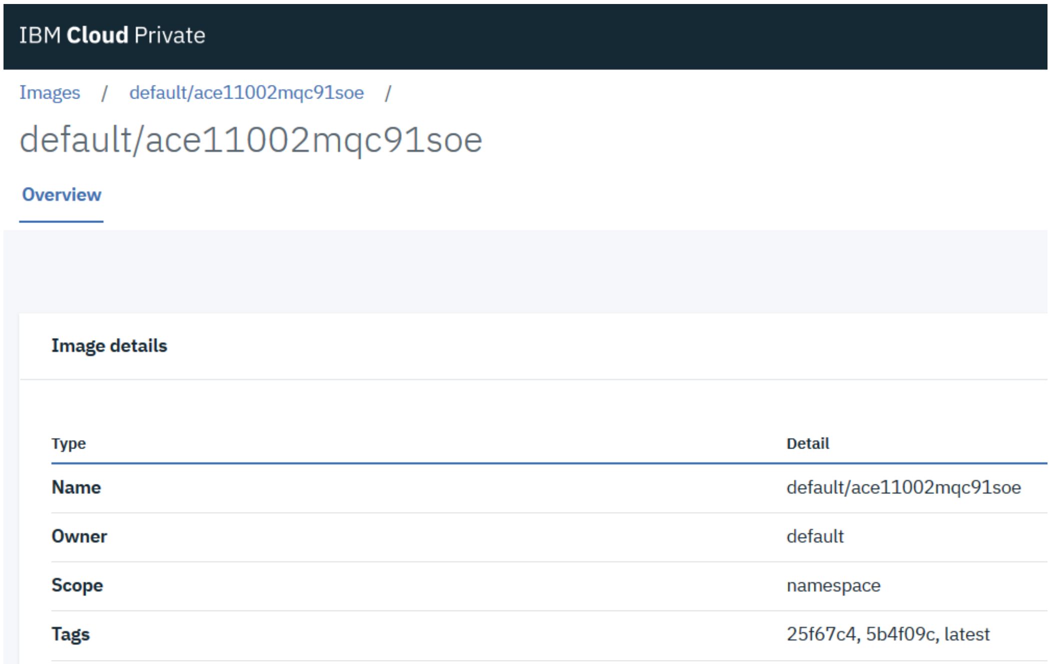

https://jenkins.172.23.52.247.nip.io/job/default/job/aceonicpintstdopimg/

Jenkins
Jenkins ▸ default ▸ aceonicpintstdopimg ▸
Up
Status
Configure
Scan Multibranch Pipeline Now
Scan Multibranch Pipeline Log
Multibranch Pipeline Events
Delete Multibranch Pipeline
People
aceonicpintstdopimg
Branches (1)
Icon: S M L

S	W	Name ↓	Last Success
●	☁	master	11 days - #9


```
Step 23/23 : LABEL "org.label-schema.build-date"='2018-12-03T03:15:37+0000' "org.label-schema.name"='ace11002mqc91soe' "org.label-
schema.schema-version"='1.0' "org.label-schema.vcs-ref"='5b4f09c' "org.label-schema.vcs-url"='https://github.com/DAVEXACOM
/ACEonICPIntStdOpImg.git'
 ---> Running in 01d0bd17e916
Removing intermediate container 01d0bd17e916
 ---> 81fde5c4b69d
Successfully built 81fde5c4b69d
Successfully tagged ace11002mqc91soe:5b4f09c
```

SoE Image on ICP



default/ace11002mqc91soe

Image details

Type	Detail
Name	default/ace11002mqc91soe
Owner	default
Scope	namespace
Tags	25f67c4, 5b4f09c, latest

Deploying the SoE

The SoE Image is never deployed it is the image that Micro Service 1 and 2 are build FROM.

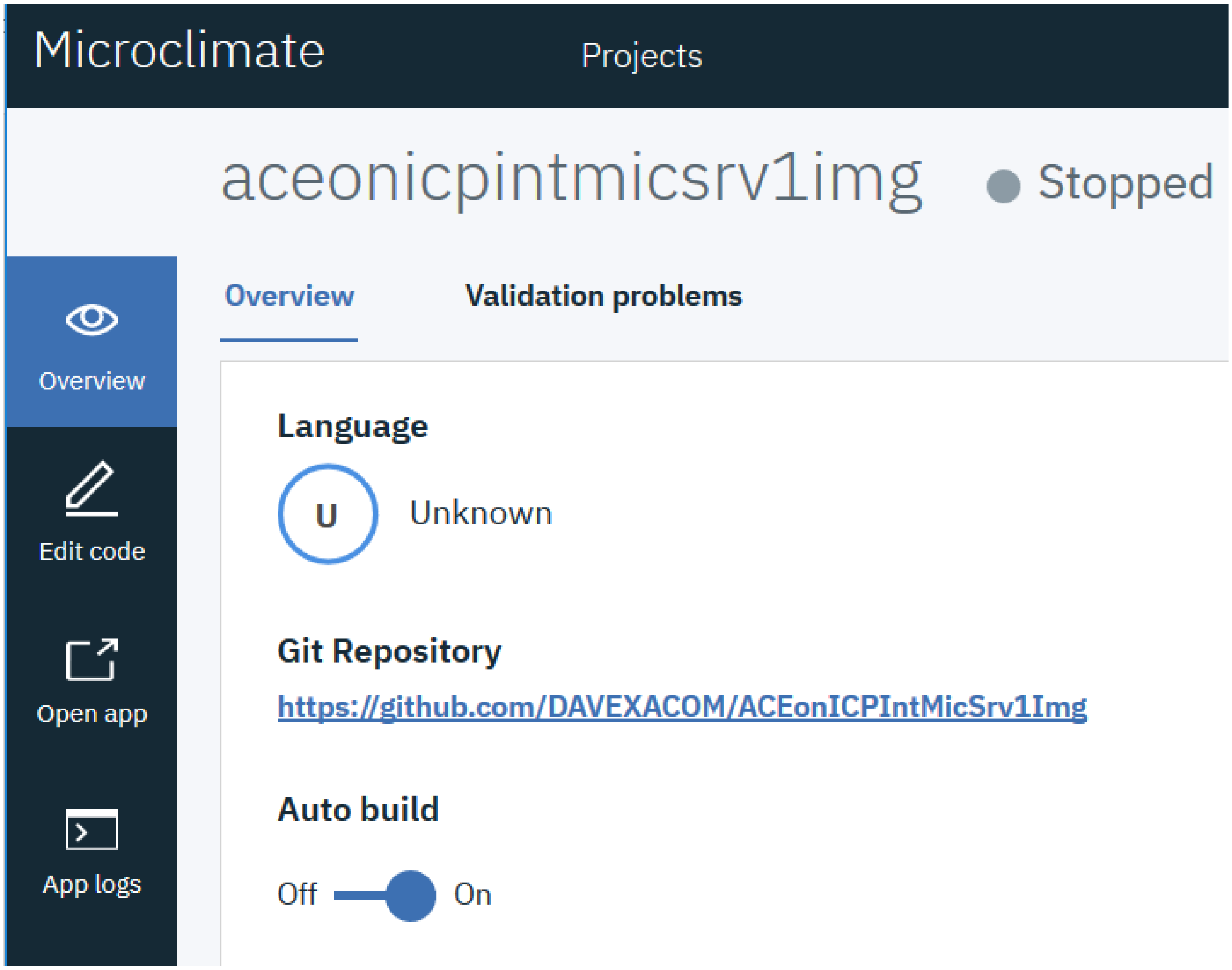
Building the Image for Micro Service 1 – GitHub ACE Micro Service 1 Build

Source Github repository

https://github.com/DAVEXACOM/ACEonICPIntMicSrv1Img

You need to copy the MicroService1.bar from the ACE Micro Service 1 source repository in the https://github.com/DAVEXACOM/ACEonICPIntMicSrv1Img/tree/master/acesoe/binary directory

Microclimate Micro Service 1 Project



aceonicpintmicsrv1img ● Stopped

Language

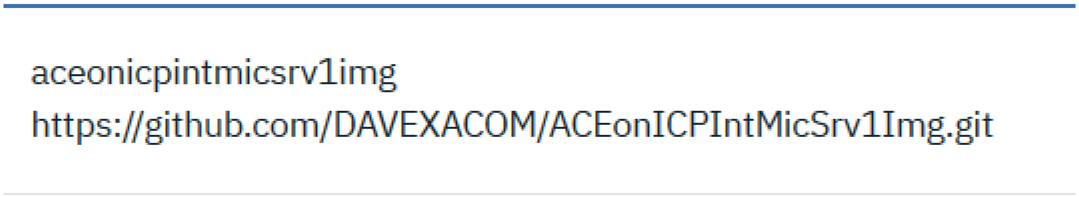
U Unknown

Git Repository

https://github.com/DAVEXACOM/ACEonICPIntMicSrv1Img

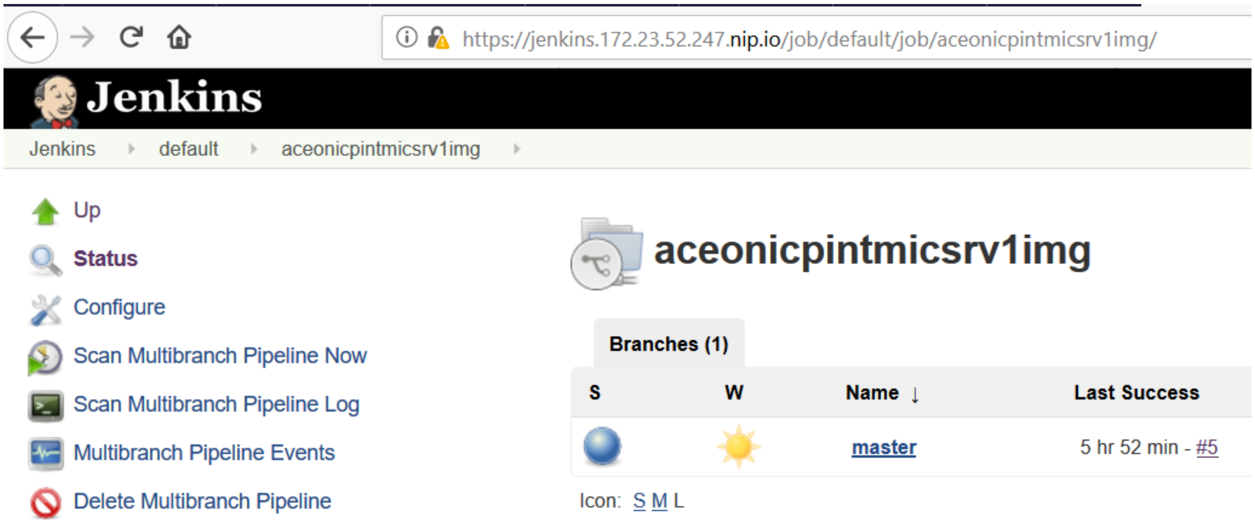
Auto build

Off On

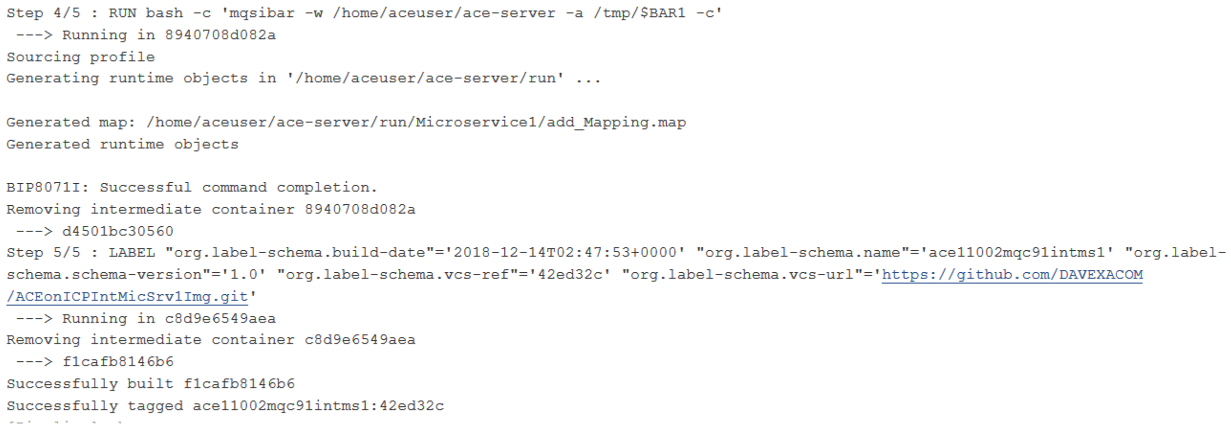


aceonicpintmicsrv1img
https://github.com/DAVEXACOM/ACEonICPIntMicSrv1Img.git

Jenkins Pipeline for Micro Service 1


https://jenkins.172.23.52.247.nip.io/job/default/job/aceonicpintmicsrv1img/
Jenkins
Jenkins
default
aceonicpintmicsrv1img
Up
Status
Configure
Scan Multibranch Pipeline Now
Scan Multibranch Pipeline Log
Multibranch Pipeline Events
Delete Multibranch Pipeline
aceonicpintmicsrv1img
Branches (1)
Icon: S M L

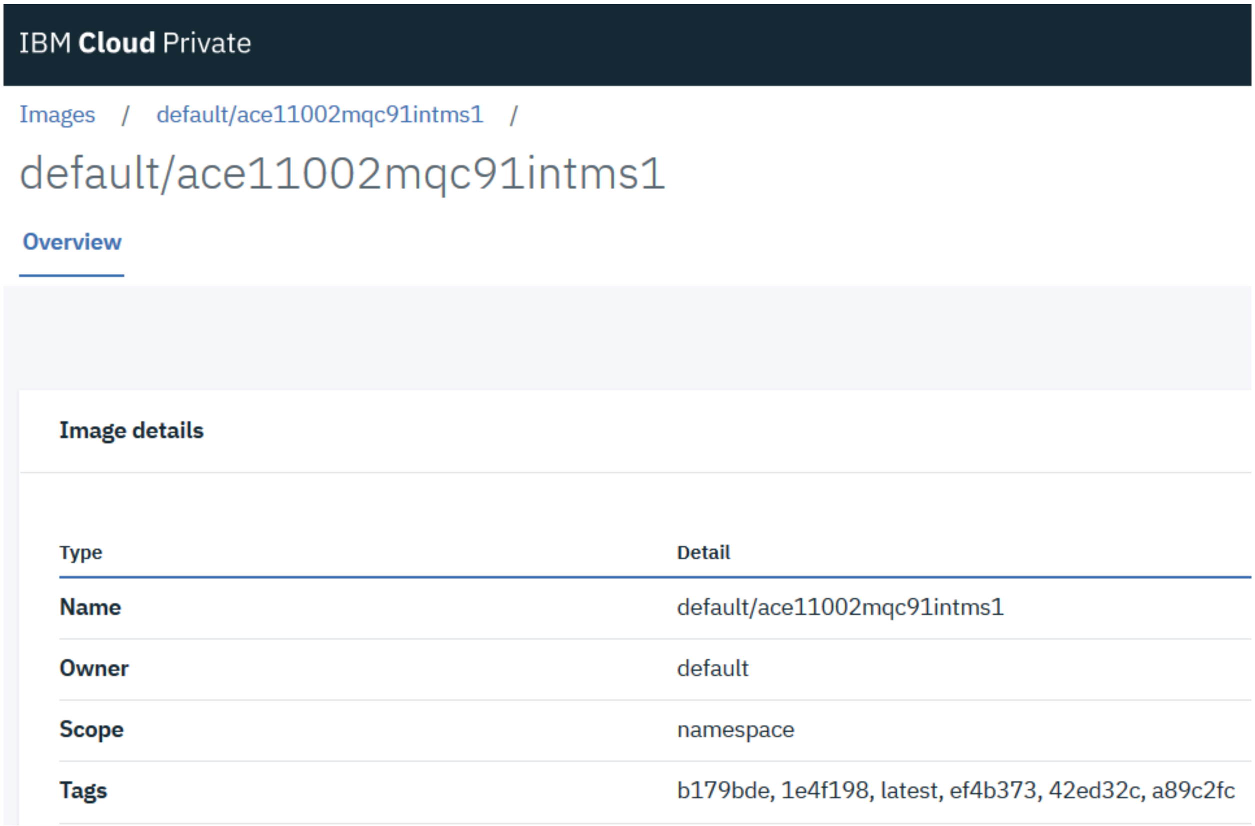
S	W	Name ↓	Last Success
●	☀	master	5 hr 52 min - #5



```
Step 4/5 : RUN bash -c 'mqsibar -w /home/aceuser/ace-server -a /tmp/$BAR1 -c'
 ---> Running in 8940708d082a
Sourcing profile
Generating runtime objects in '/home/aceuser/ace-server/run' ...

Generated map: /home/aceuser/ace-server/run/Microservice1/add_Mapping.map
Generated runtime objects

BIP8071I: Successful command completion.
Removing intermediate container 8940708d082a
 ---> d4501bc30560
Step 5/5 : LABEL "org.label-schema.build-date"='2018-12-14T02:47:53+0000' "org.label-schema.name"='ace11002mqc91intms1' "org.label-
schema.schema-version"='1.0' "org.label-schema.vcs-ref"='42ed32c' "org.label-schema.vcs-url"='https://github.com/DAVEXACOM
/ACEonICPIntMicSrv1Img.git'
 ---> Running in c8d9e6549aea
Removing intermediate container c8d9e6549aea
 ---> f1cafb8146b6
Successfully built f1cafb8146b6
Successfully tagged ace11002mqc91intms1:42ed32c
```

Note: The pipeline will try and deploy but fail. We don’t want this pipeline doing the deploy. This is an image build only

Micro Service 1 Image on ICP



default/ace11002mqc91intms1

Image details

Type	Detail
Name	default/ace11002mqc91intms1
Owner	default
Scope	namespace
Tags	b179bde, 1e4f198, latest, ef4b373, 42ed32c, a89c2fc

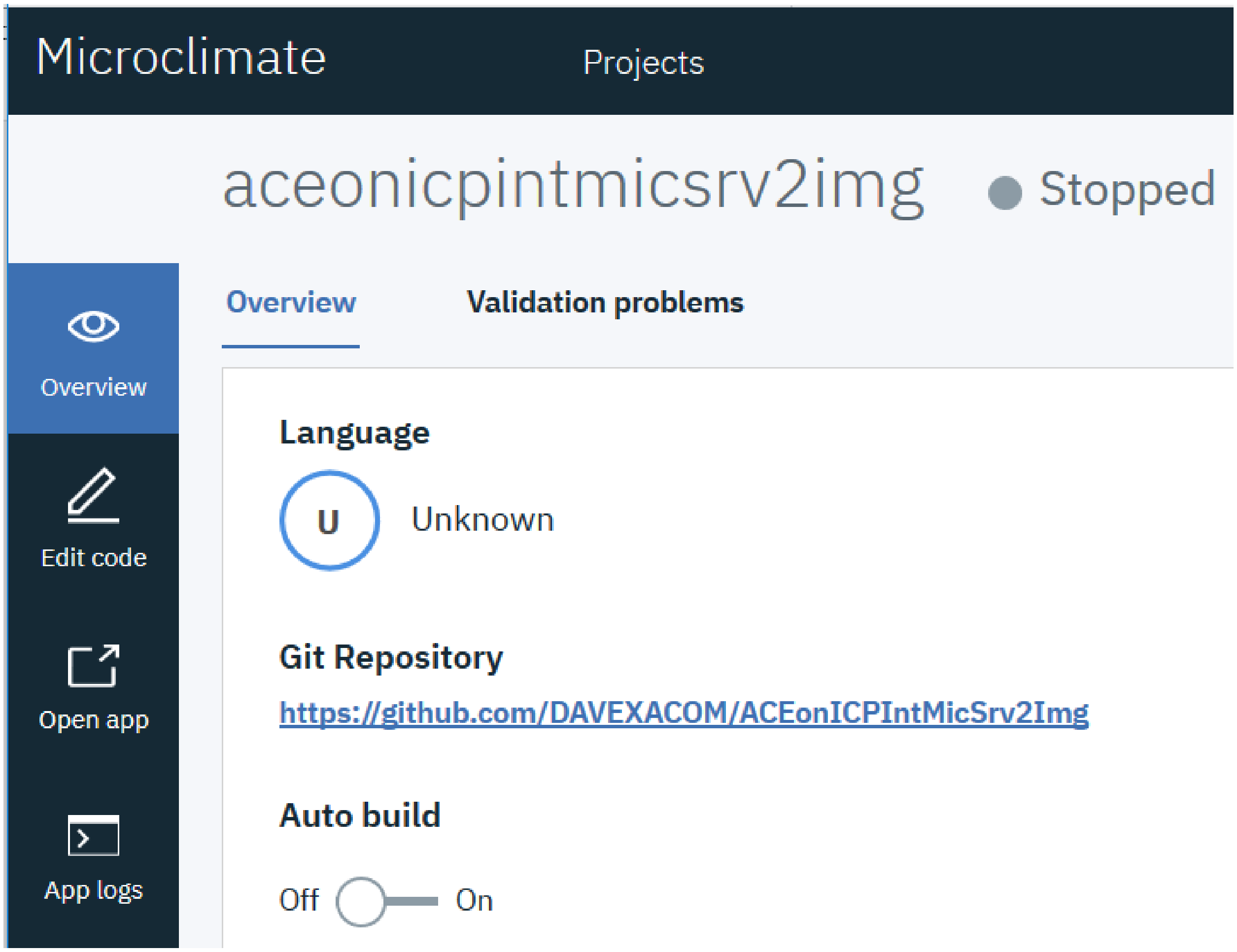
Building the Image for Micro Service 2 – GitHub ACE Micro Service 2 Build

Source Github repository

https://github.com/DAVEXACOM/ACEonICPIntMicSrv2Img

You need to copy the MicroService2.bar from the ACE Micro Service 2 source repository into the https://github.com/DAVEXACOM/ACEonICPIntMicSrv2Img/tree/master/acesoe/binary directory

Microclimate Micro Service 2 Project



aceonicpintmicsrv2img ● Stopped

Language

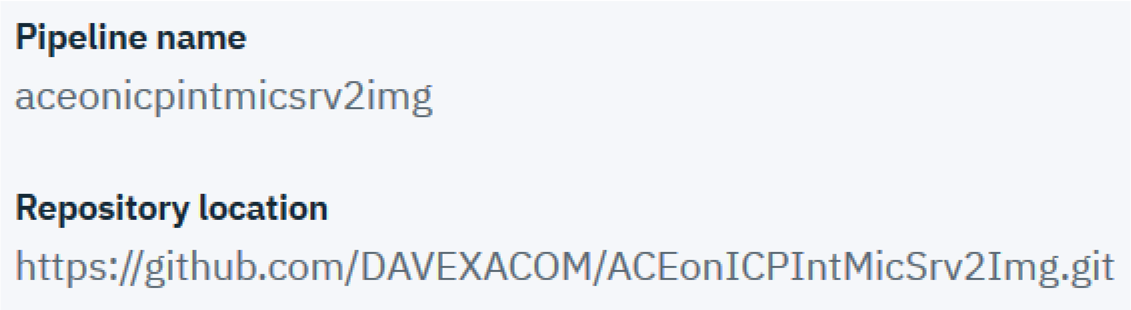
U Unknown

Git Repository

https://github.com/DAVEXACOM/ACEonICPIntMicSrv2Img

Auto build

Off ○— On



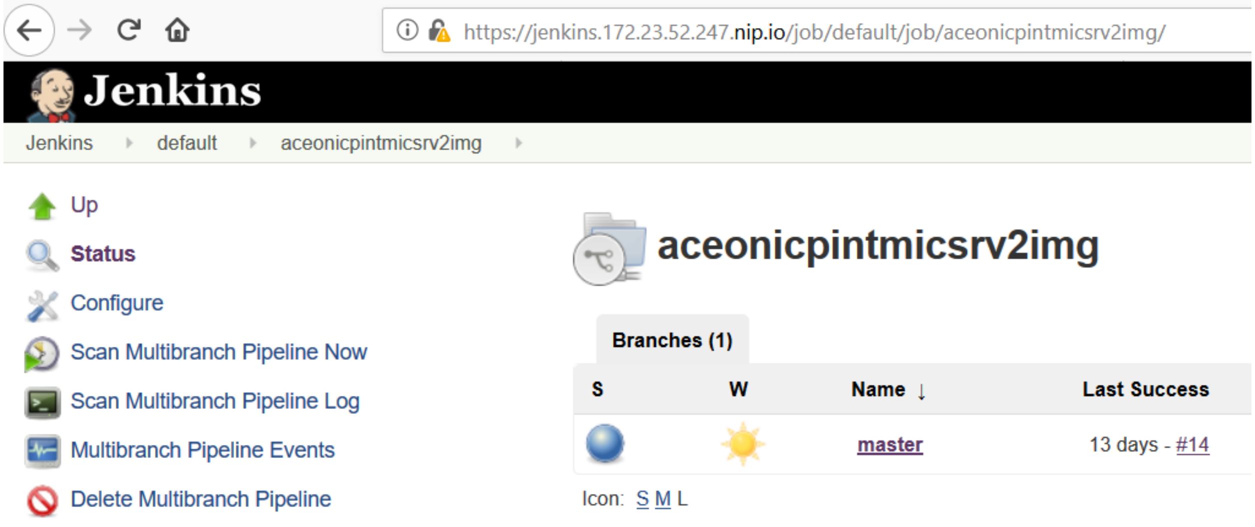
Pipeline name

aceonicpintmicsrv2img

Repository location

https://github.com/DAVEXACOM/ACEonICPIntMicSrv2Img.git

Jenkins Pipeline for Micro Service 2

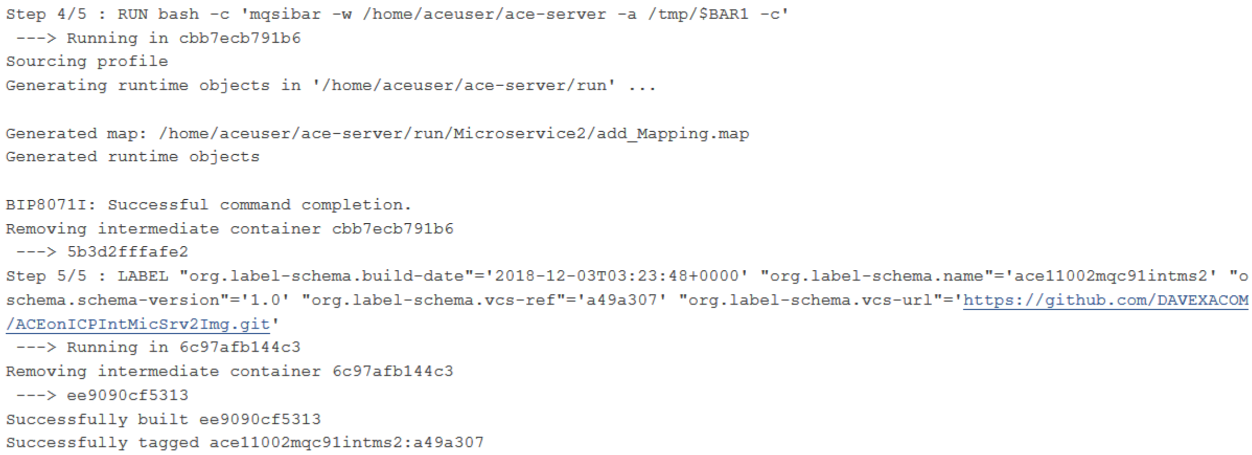

https://jenkins.172.23.52.247.nip.io/job/default/job/aceonicpintmicsrv2img/
Jenkins
Jenkins ▸ default ▸ aceonicpintmicsrv2img ▸
Up
Status
Configure
Scan Multibranch Pipeline Now
Scan Multibranch Pipeline Log
Multibranch Pipeline Events
Delete Multibranch Pipeline

aceonicpintmicsrv2img

Branches (1)

S	W	Name ↓	Last Success
●	☀	master	13 days - #14

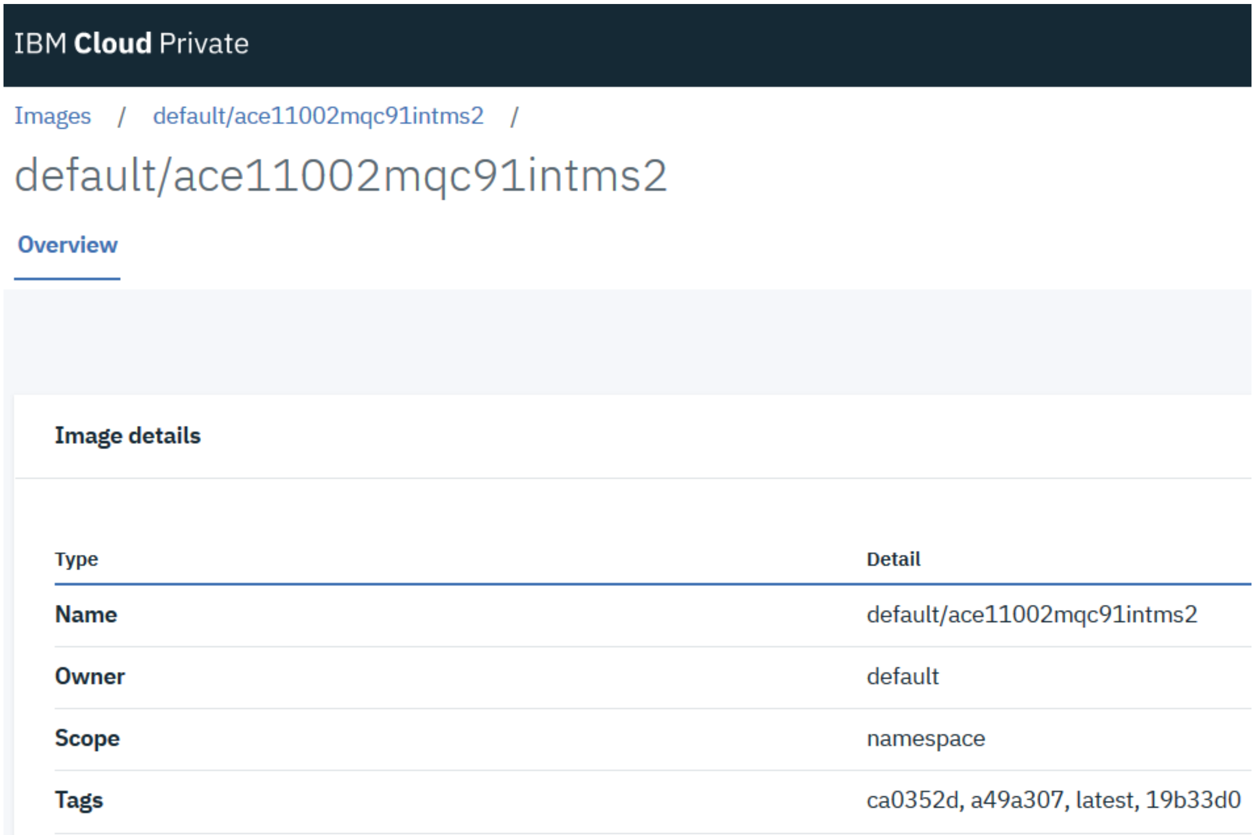
Icon: S M L



```
Step 4/5 : RUN bash -c 'mqsibar -w /home/aceuser/ace-server -a /tmp/$BAR1 -c'
 ---> Running in cbb7ecb791b6
Sourcing profile
Generating runtime objects in '/home/aceuser/ace-server/run' ...

Generated map: /home/aceuser/ace-server/run/Microservice2/add_Mapping.map
Generated runtime objects

BIP8071I: Successful command completion.
Removing intermediate container cbb7ecb791b6
 ---> 5b3d2fffafe2
Step 5/5 : LABEL "org.label-schema.build-date"='2018-12-03T03:23:48+0000' "org.label-schema.name"='ace11002mqc91intms2' "o
schema.schema-version"='1.0' "org.label-schema.vcs-ref"='a49a307' "org.label-schema.vcs-url"='https://github.com/DAVEXACOM
/ACEonICPIntMicSrv2Img.git'
 ---> Running in 6c97afb144c3
Removing intermediate container 6c97afb144c3
 ---> ee9090cf5313
Successfully built ee9090cf5313
Successfully tagged ace11002mqc91intms2:a49a307
```

Note: The pipeline will try and deploy but fail. We don’t want this pipeline doing the deploy. This is an image build only

Micro Service 2 Image on ICP



default/ace11002mqc91intms2

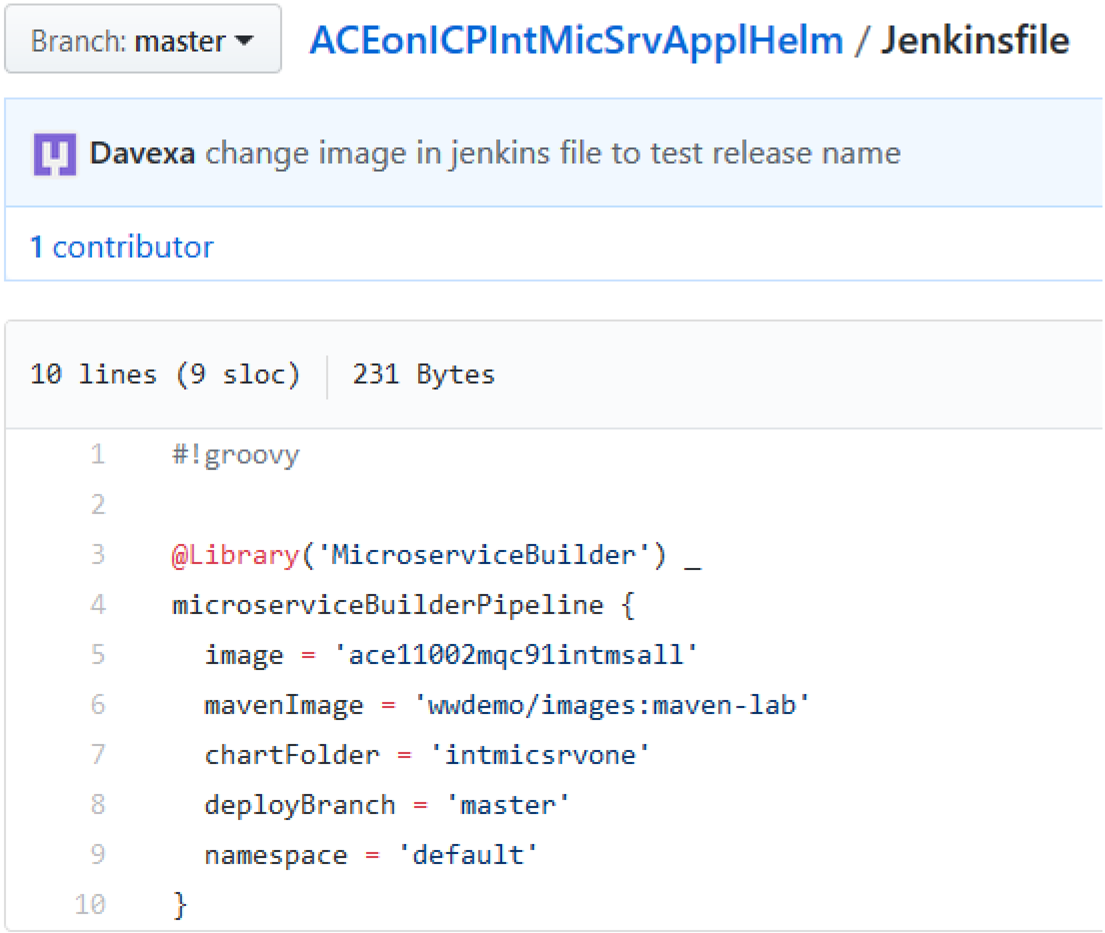
Image details

Type	Detail
Name	default/ace11002mqc91intms2
Owner	default
Scope	namespace
Tags	ca0352d, a49a307, latest, 19b33d0

Deploying Testing and Demoing

Deploying Micro Service 1 and Micro Service 2 together as a Micro Services Application

Github repository - ACEonICPIntMicSrvHelm
https://github.com/DAVEXACOM/ACEonICPIntMicSrvApplHelm



Branch: master ACEonICPIntMicSrvApplHelm / Jenkinsfile

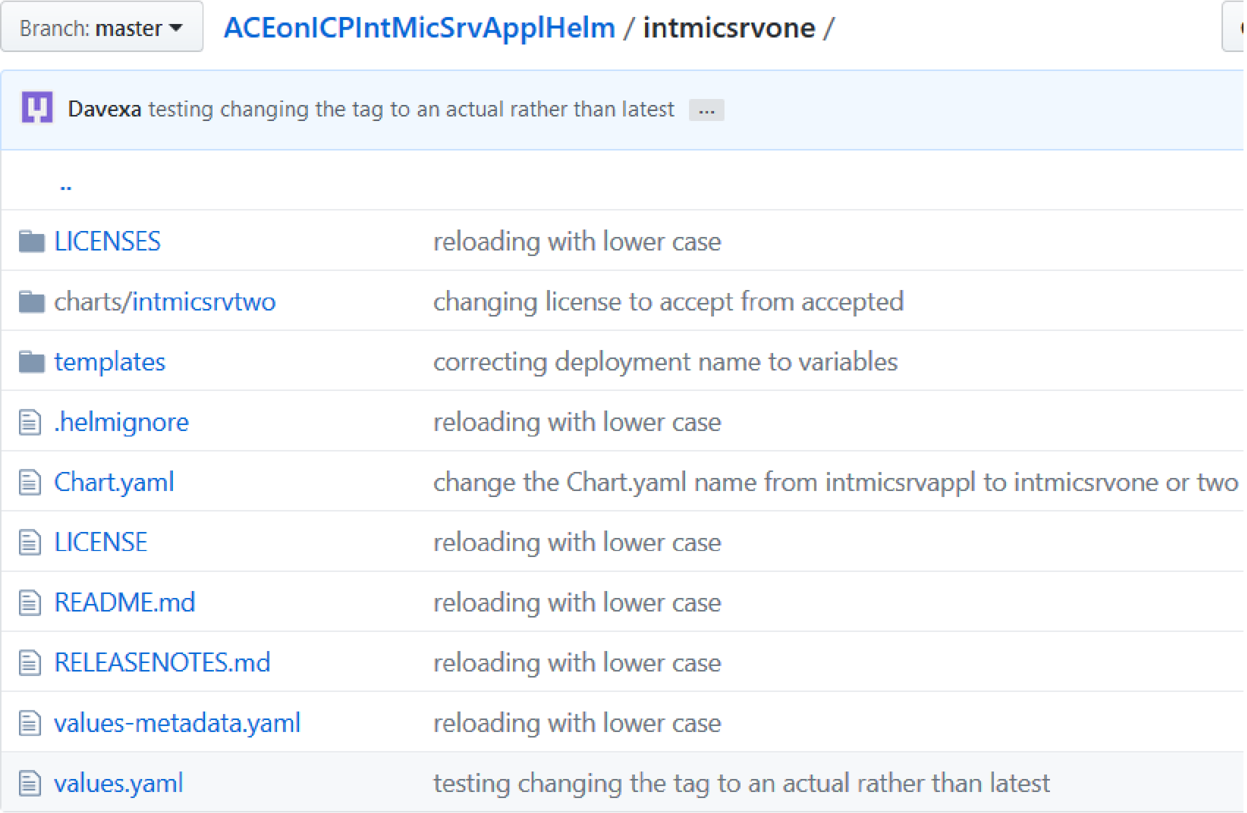
Davexa change image in jenkins file to test release name

1 contributor

10 lines (9 sloc) | 231 Bytes

```
#!groovy

@Library('MicroserviceBuilder') _
microserviceBuilderPipeline {
  image = 'ace11002mqc91intmsall'
  mavenImage = 'wwdemo/images:maven-lab'
  chartFolder = 'intmicsrvone'
  deployBranch = 'master'
  namespace = 'default'
}
```

Chart Files for Integration Micro Service 1
https://github.com/DAVEXACOM/ACEonICPIntMicSrvApplHelm/tree/master/intmicsrvone

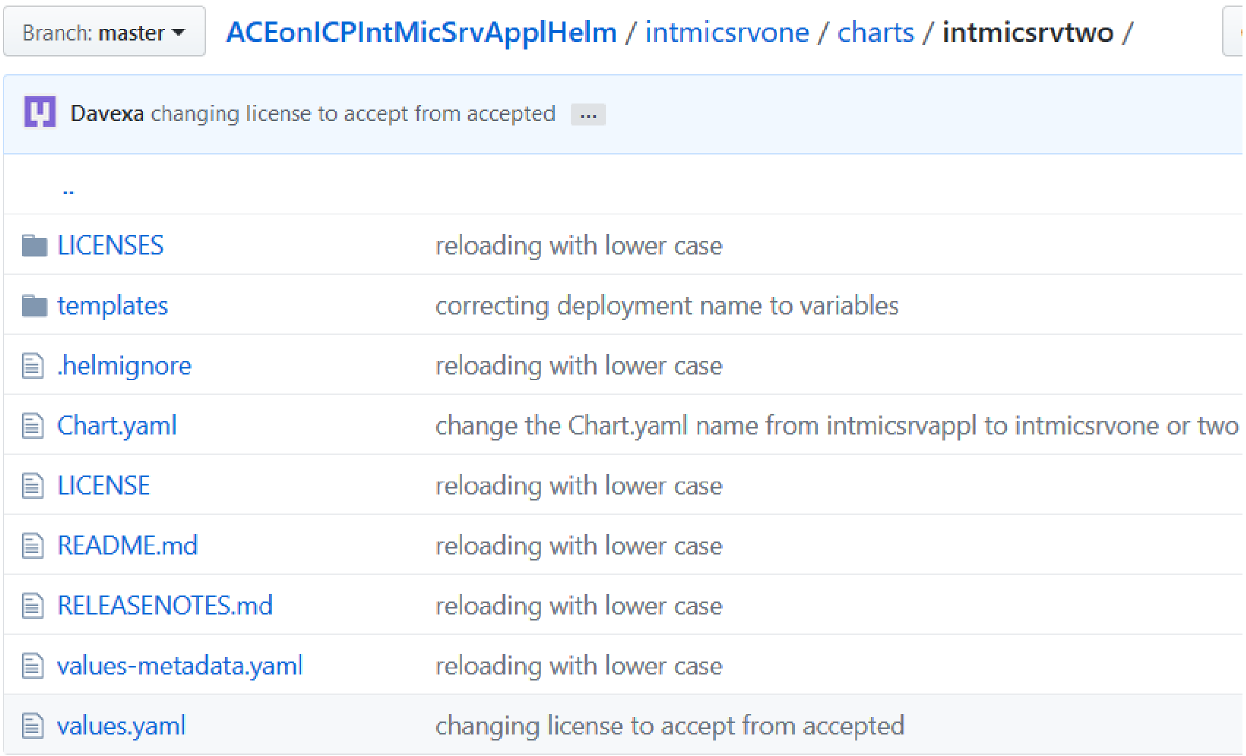


Branch: master ▾ ACEonICPIntMicSrvApplHelm / intmicsrvone /

Davexa testing changing the tag to an actual rather than latest …

..	
LICENSES	reloading with lower case
charts/intmicsrvtwo	changing license to accept from accepted
templates	correcting deployment name to variables
.helmignore	reloading with lower case
Chart.yaml	change the Chart.yaml name from intmicsrvappl to intmicsrvone or two
LICENSE	reloading with lower case
README.md	reloading with lower case
RELEASENOTES.md	reloading with lower case
values-metadata.yaml	reloading with lower case
values.yaml	testing changing the tag to an actual rather than latest

Sub Chart files for Integration Micro Service 2
https://github.com/DAVEXACOM/ACEonICPIntMicSrvApplHelm/tree/master/intmicsrvone/charts/intmicsrvtwo



Branch: master ▾ ACEonICPIntMicSrvApplHelm / intmicsrvone / charts / intmicsrvtwo /

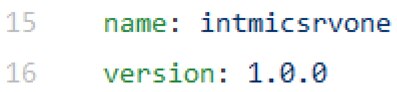
Davexa changing license to accept from accepted ...

..	
LICENSES	reloading with lower case
templates	correcting deployment name to variables
.helmignore	reloading with lower case
Chart.yaml	change the Chart.yaml name from intmicsrvappl to intmicsrvone or two
LICENSE	reloading with lower case
README.md	reloading with lower case
RELEASENOTES.md	reloading with lower case
values-metadata.yaml	reloading with lower case
values.yaml	changing license to accept from accepted

YAML Chart file modifications - How it hangs together

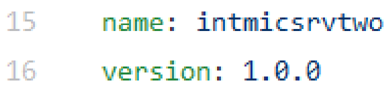
The following YAMl files have been modified from the OT4i content. The main modifications are in the templates YAMLs. Because the sub charts folder and content for integration service two is a copy of main charts files we need to ensure that the secrets and services etc are not created with the same names.

Chart – Chart.yaml



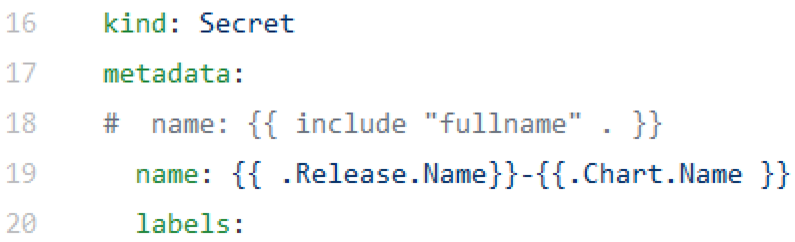
```
name: intmicsrvone
version: 1.0.0
```

Sub chart – Chart.yaml



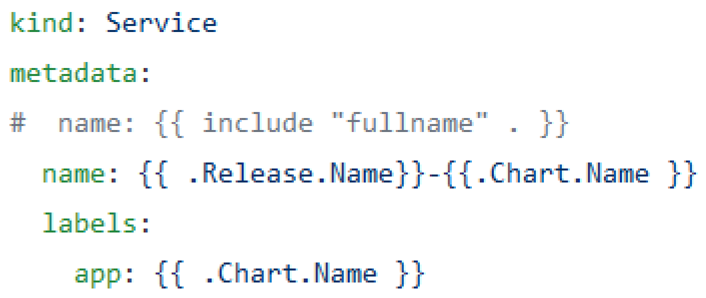
```
name: intmicsrvtwo
version: 1.0.0
```

Secrets.yaml for example



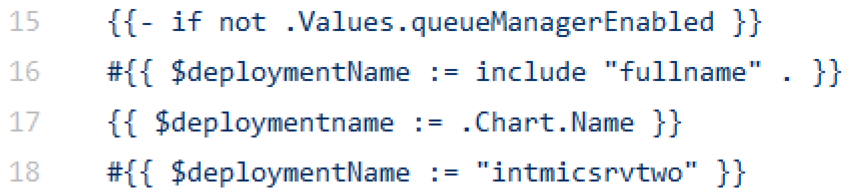
```
kind: Secret
metadata:
#  name: {{ include "fullname" . }}
  name: {{ .Release.Name}}-{{.Chart.Name }}
  labels:
```

Service.yaml



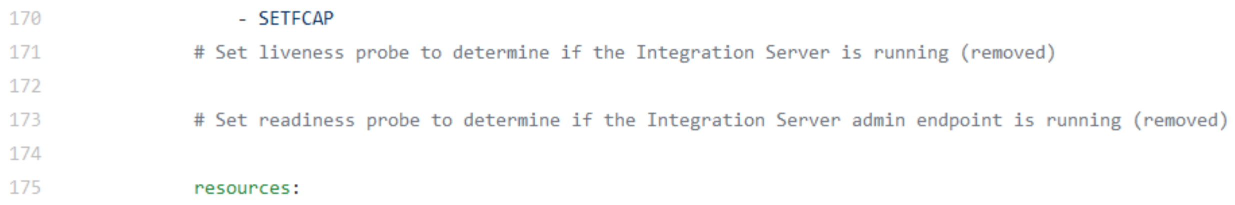
```
kind: Service
metadata:
#  name: {{ include "fullname" . }}
  name: {{ .Release.Name}}-{{.Chart.Name }}
  labels:
    app: {{ .Chart.Name }}
```

Deployment.yaml



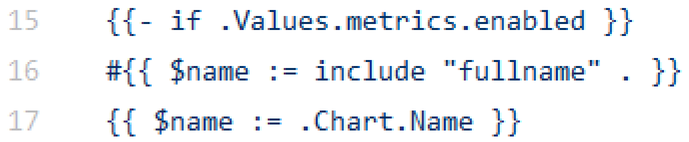
```
{{- if not .Values.queueManagerEnabled }}
#{{ $deploymentName := include "fullname" . }}
{{ $deploymentname := .Chart.Name }}
#{{ $deploymentName := "intmicsrvtwo" }}
```

Also removed the cloud pack liveness and readiness probes we have our own in the SOE build image.



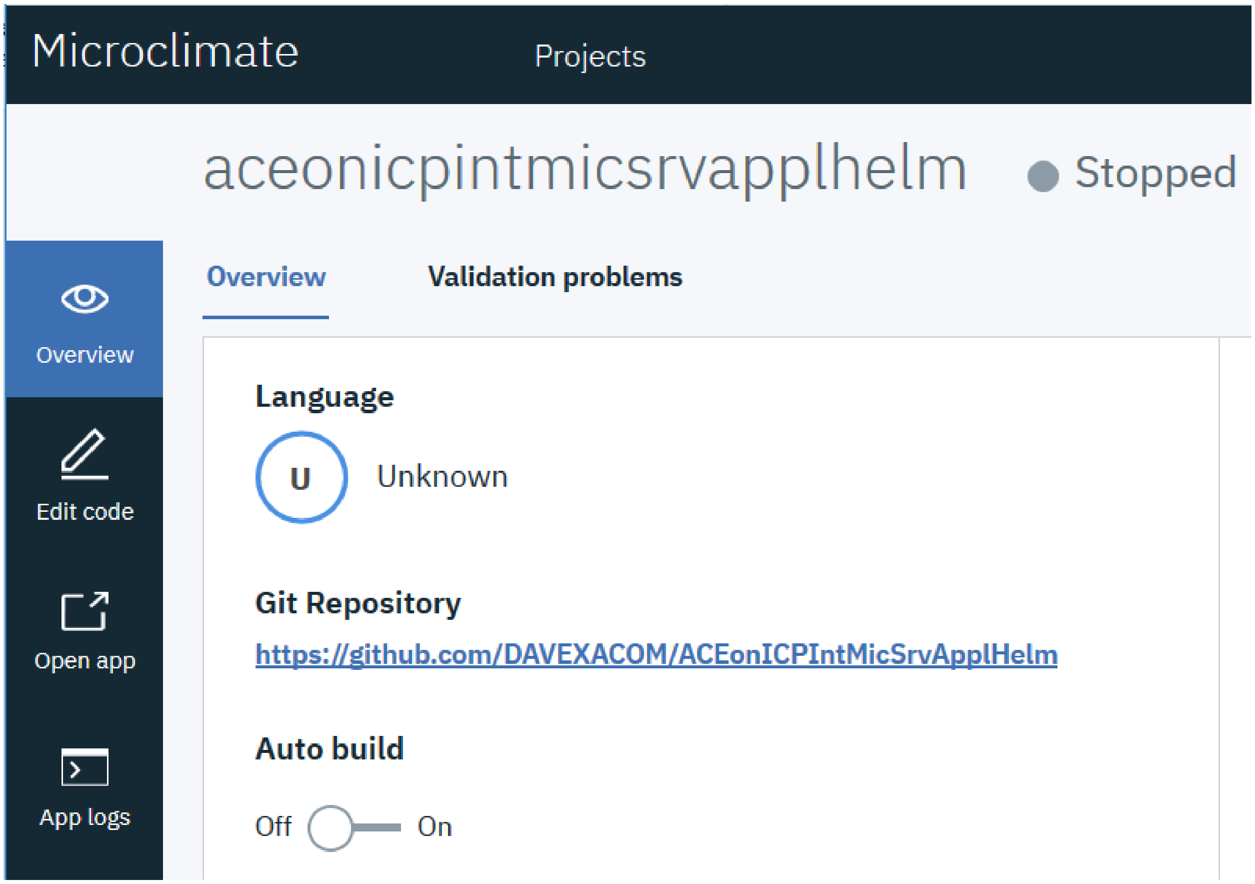
```
    - SETFCAP
# Set liveness probe to determine if the Integration Server is running (removed)

# Set readiness probe to determine if the Integration Server admin endpoint is running (removed)

resources:
```

Metrics-service-ace.yaml



```
{{- if .Values.metrics.enabled }}
#{{ $name := include "fullname" . }}
{{ $name := .Chart.Name }}
```

Microclimate project aceonicpintmicsrvapplhelm



aceonicpintmicsrvapplhelm ● Stopped

Language

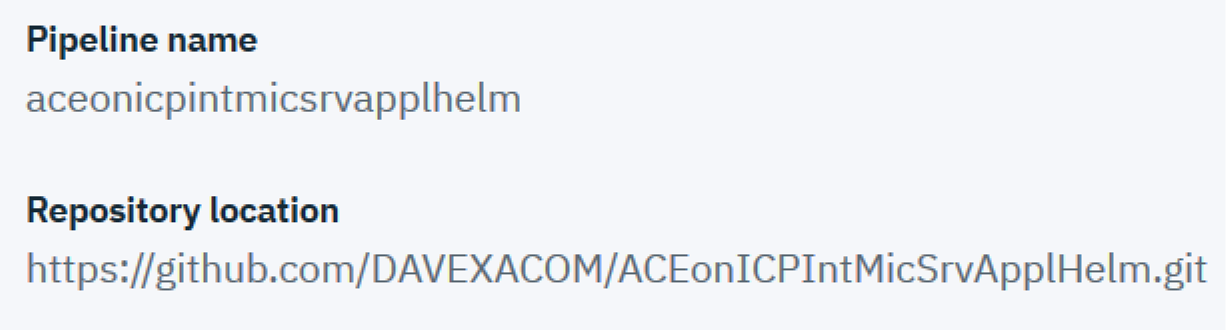
Unknown

Git Repository

https://github.com/DAVEXACOM/ACEonICPIntMicSrvApplHelm

Auto build

Off ○ On



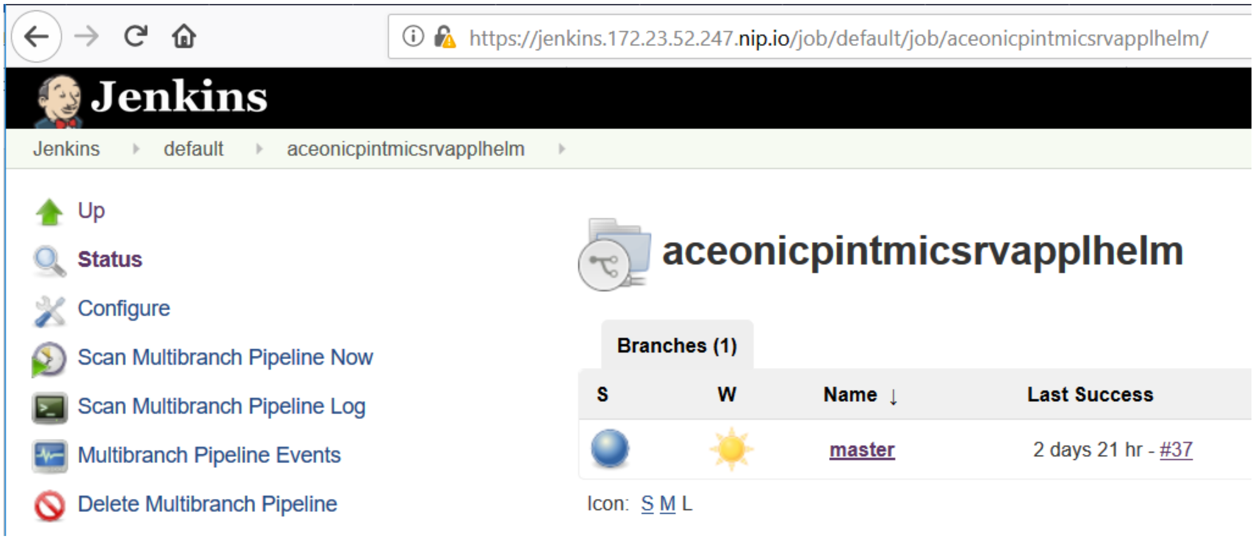
Pipeline name

aceonicpintmicsrvapplhelm

Repository location

https://github.com/DAVEXACOM/ACEonICPIntMicSrvApplHelm.git

Jenkins Pipeline - aceonicpintmicsrvapplhelm



https://jenkins.172.23.52.247.nip.io/job/default/job/aceonicpintmicsrvapplhelm/

Jenkins

Jenkins › default › aceonicpintmicsrvapplhelm ›

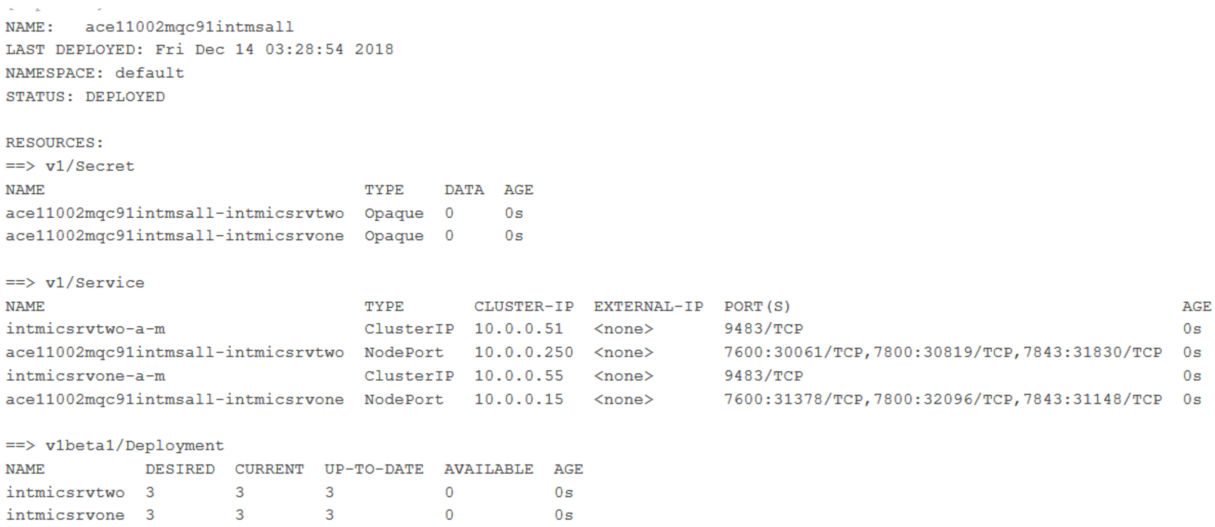
- Up
- Status
- Configure
- Scan Multibranch Pipeline Now
- Scan Multibranch Pipeline Log
- Multibranch Pipeline Events
- Delete Multibranch Pipeline

aceonicpintmicsrvapplhelm

Branches (1)

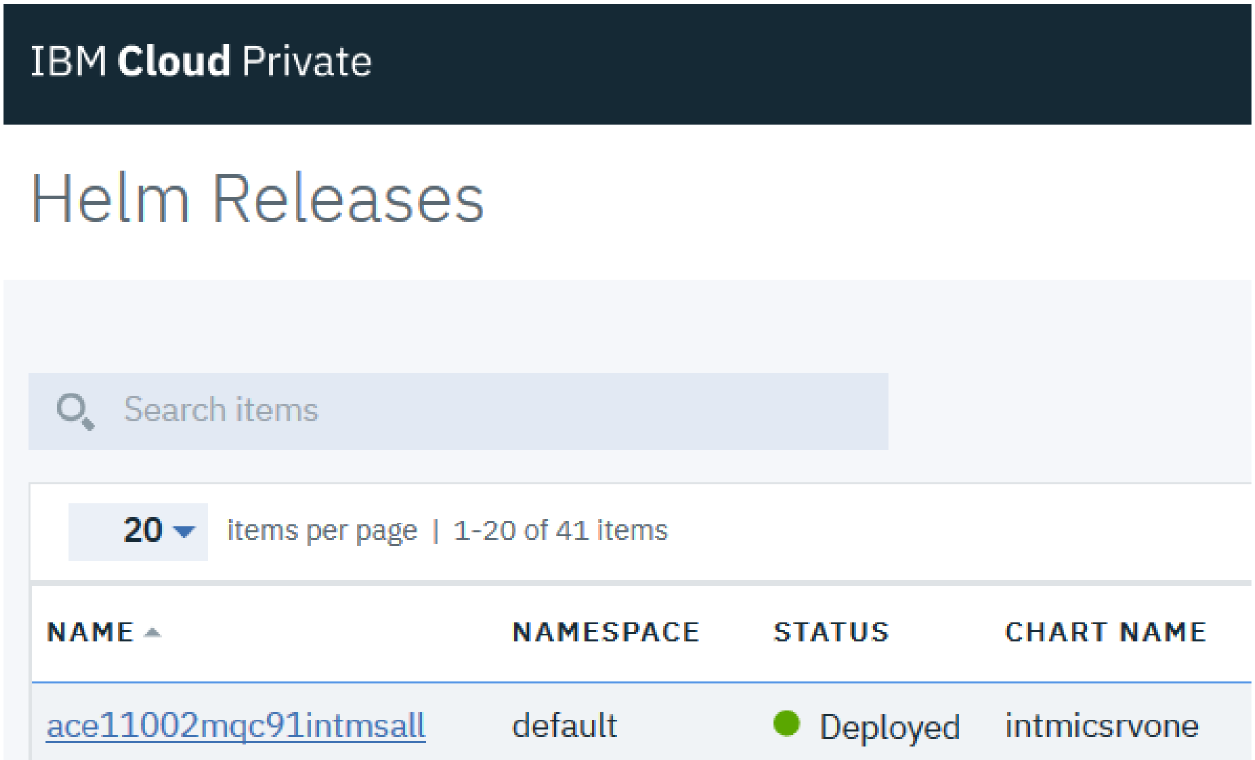
S	W	Name ↓	Last Success
●	☀	master	2 days 21 hr - #37

Icon: S M L



```
NAME:   ace11002mqc91intmsall
LAST DEPLOYED: Fri Dec 14 03:28:54 2018
NAMESPACE: default
STATUS: DEPLOYED

RESOURCES:
==> v1/Secret
NAME                                 TYPE    DATA  AGE
ace11002mqc91intmsall-intmicsrvtwo  Opaque  0     0s
ace11002mqc91intmsall-intmicsrvone  Opaque  0     0s

==> v1/Service
NAME                                 TYPE       CLUSTER-IP  EXTERNAL-IP  PORT(S)                                         AGE
intmicsrvtwo-a-m                     ClusterIP  10.0.0.51   <none>       9483/TCP                                        0s
ace11002mqc91intmsall-intmicsrvtwo  NodePort   10.0.0.250  <none>       7600:30061/TCP,7800:30819/TCP,7843:31830/TCP  0s
intmicsrvone-a-m                     ClusterIP  10.0.0.55   <none>       9483/TCP                                        0s
ace11002mqc91intmsall-intmicsrvone  NodePort   10.0.0.15   <none>       7600:31378/TCP,7800:32096/TCP,7843:31148/TCP  0s

==> v1beta1/Deployment
NAME          DESIRED  CURRENT  UP-TO-DATE  AVAILABLE  AGE
intmicsrvtwo  3        3        3           0          0s
intmicsrvone  3        3        3           0          0s
```

Helm Release on ICP for Integration Micro Services Application



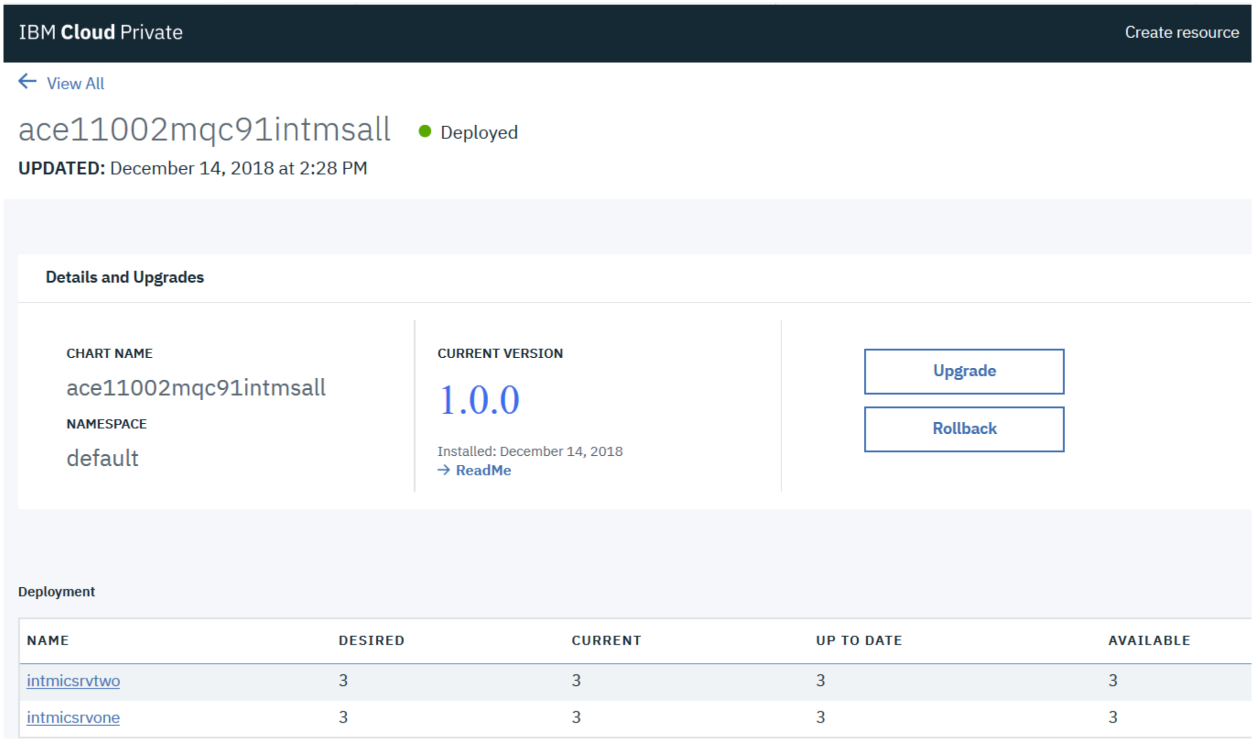
IBM **Cloud** Private

Helm Releases

Search items

20 items per page | 1-20 of 41 items

NAME	NAMESPACE	STATUS	CHART NAME
ace11002mqc91intmsall	default	● Deployed	intmicsrvone



IBM **Cloud** Private

ace11002mqc91intmsall ● Deployed

UPDATED: December 14, 2018 at 2:28 PM

Details and Upgrades

CHART NAME
ace11002mqc91intmsall

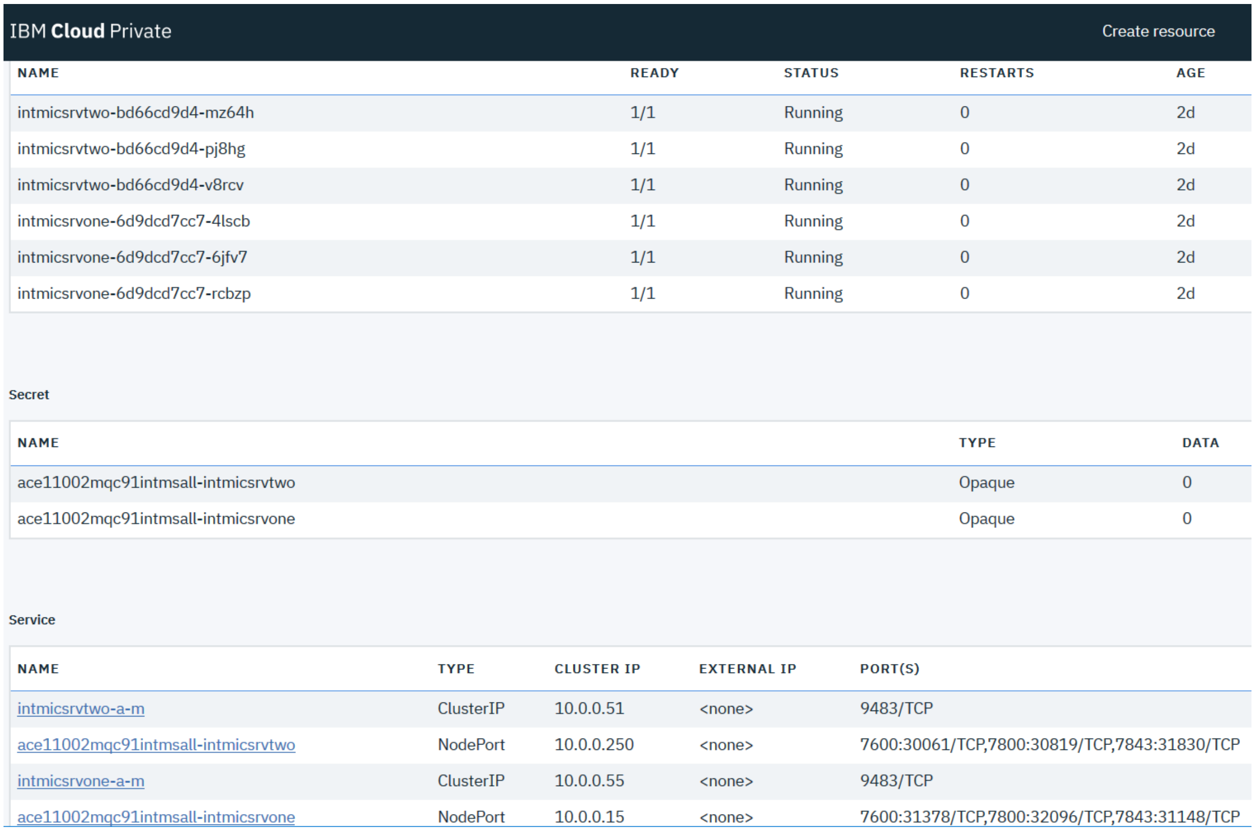
NAMESPACE
default

CURRENT VERSION
1.0.0

Installed: December 14, 2018

Deployment

NAME	DESIRED	CURRENT	UP TO DATE	AVAILABLE
intmicsrvtwo	3	3	3	3
intmicsrvone	3	3	3	3



NAME	READY	STATUS	RESTARTS	AGE
intmicsrvtwo-bd66cd9d4-mz64h	1/1	Running	0	2d
intmicsrvtwo-bd66cd9d4-pj8hg	1/1	Running	0	2d
intmicsrvtwo-bd66cd9d4-v8rcv	1/1	Running	0	2d
intmicsrvone-6d9dcd7cc7-4lscb	1/1	Running	0	2d
intmicsrvone-6d9dcd7cc7-6jfv7	1/1	Running	0	2d
intmicsrvone-6d9dcd7cc7-rcbzp	1/1	Running	0	2d

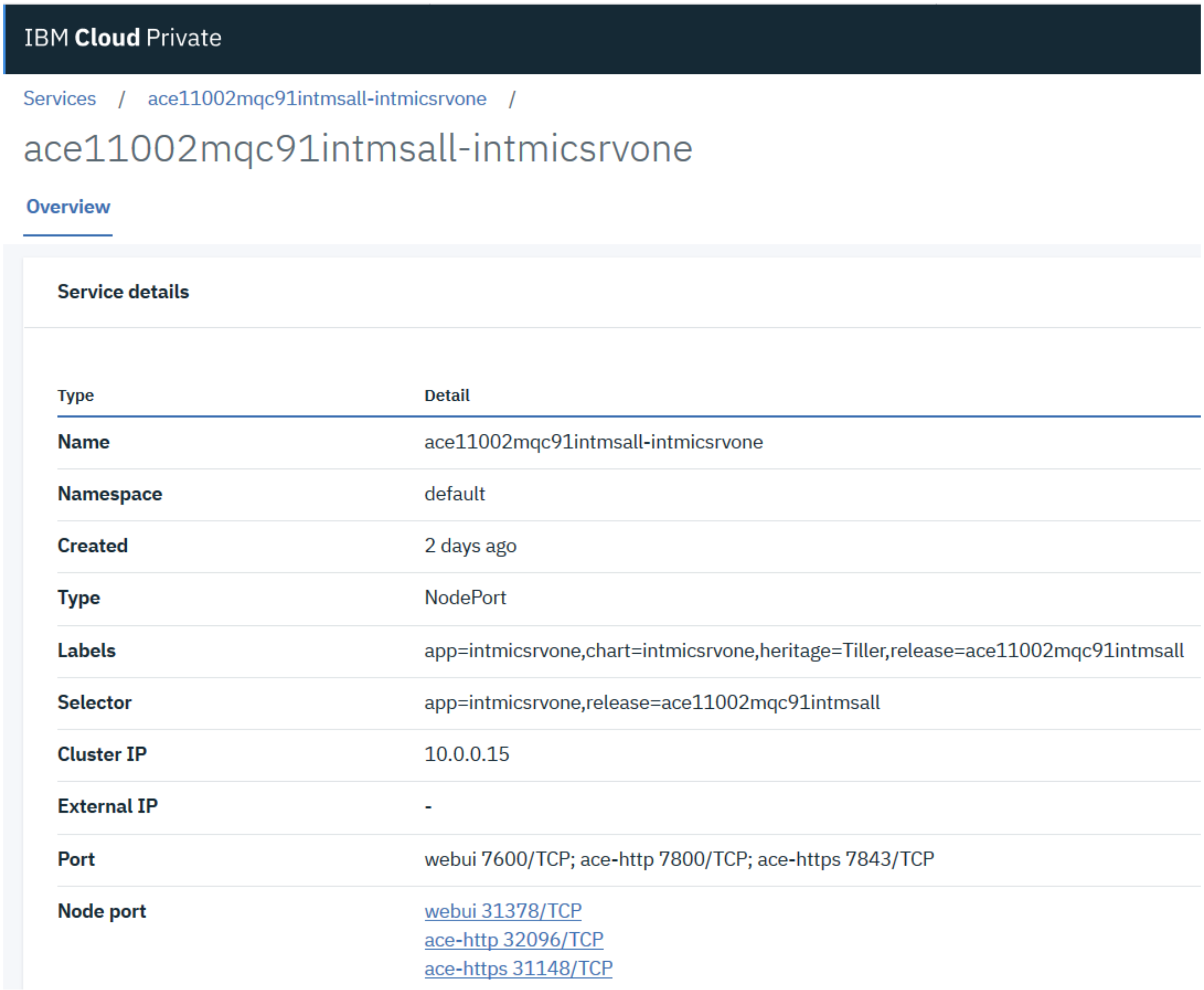
Secret

NAME	TYPE	DATA
ace11002mqc91intmsall-intmicsrvtwo	Opaque	0
ace11002mqc91intmsall-intmicsrvone	Opaque	0

Service

NAME	TYPE	CLUSTER IP	EXTERNAL IP	PORT(S)
intmicsrvtwo-a-m	ClusterIP	10.0.0.51	<none>	9483/TCP
ace11002mqc91intmsall-intmicsrvtwo	NodePort	10.0.0.250	<none>	7600:30061/TCP,7800:30819/TCP,7843:31830/TCP
intmicsrvone-a-m	ClusterIP	10.0.0.55	<none>	9483/TCP
ace11002mqc91intmsall-intmicsrvone	NodePort	10.0.0.15	<none>	7600:31378/TCP,7800:32096/TCP,7843:31148/TCP

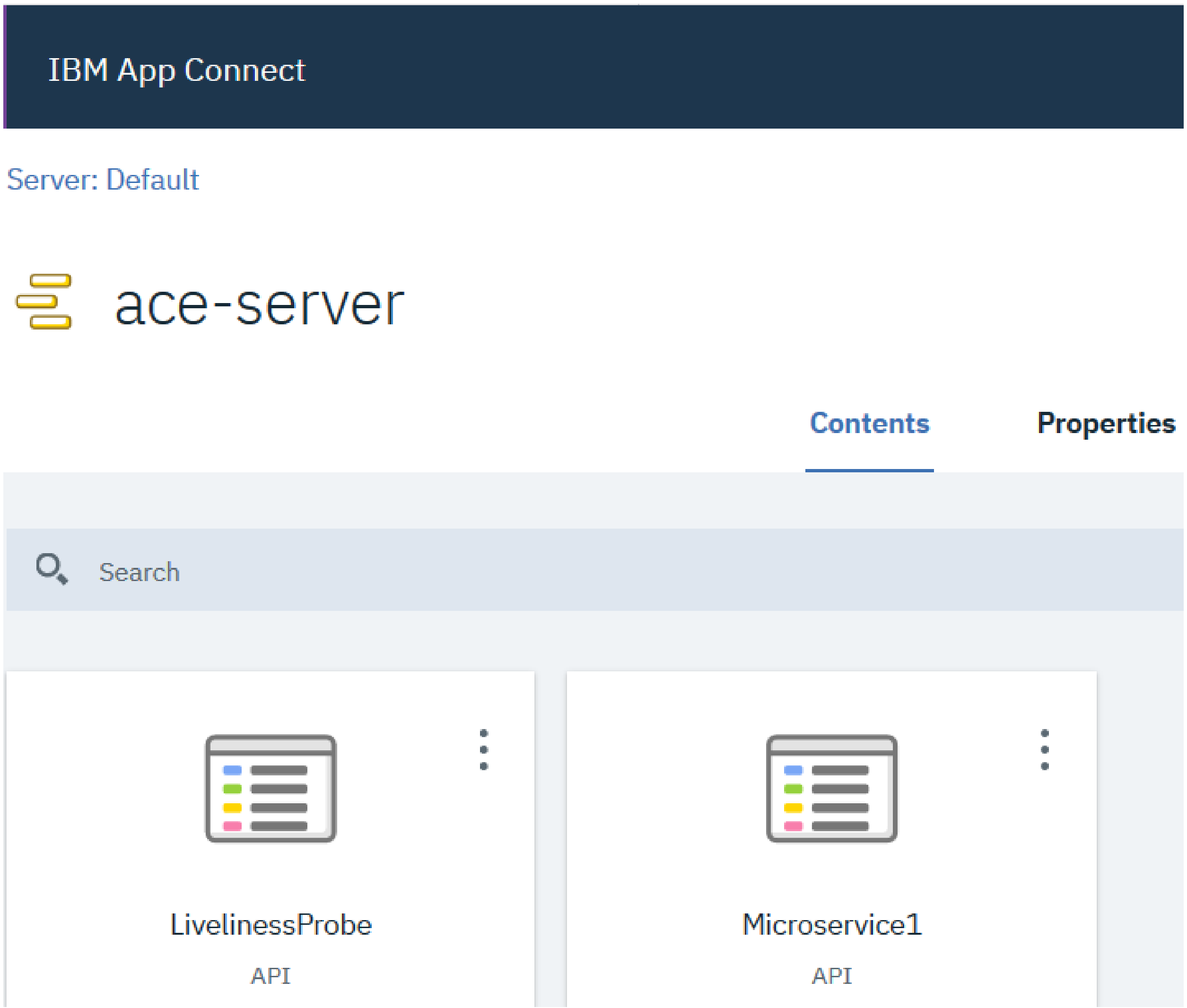
Integration Micro Service One



ace11002mqc91intmsall-intmicsrvone

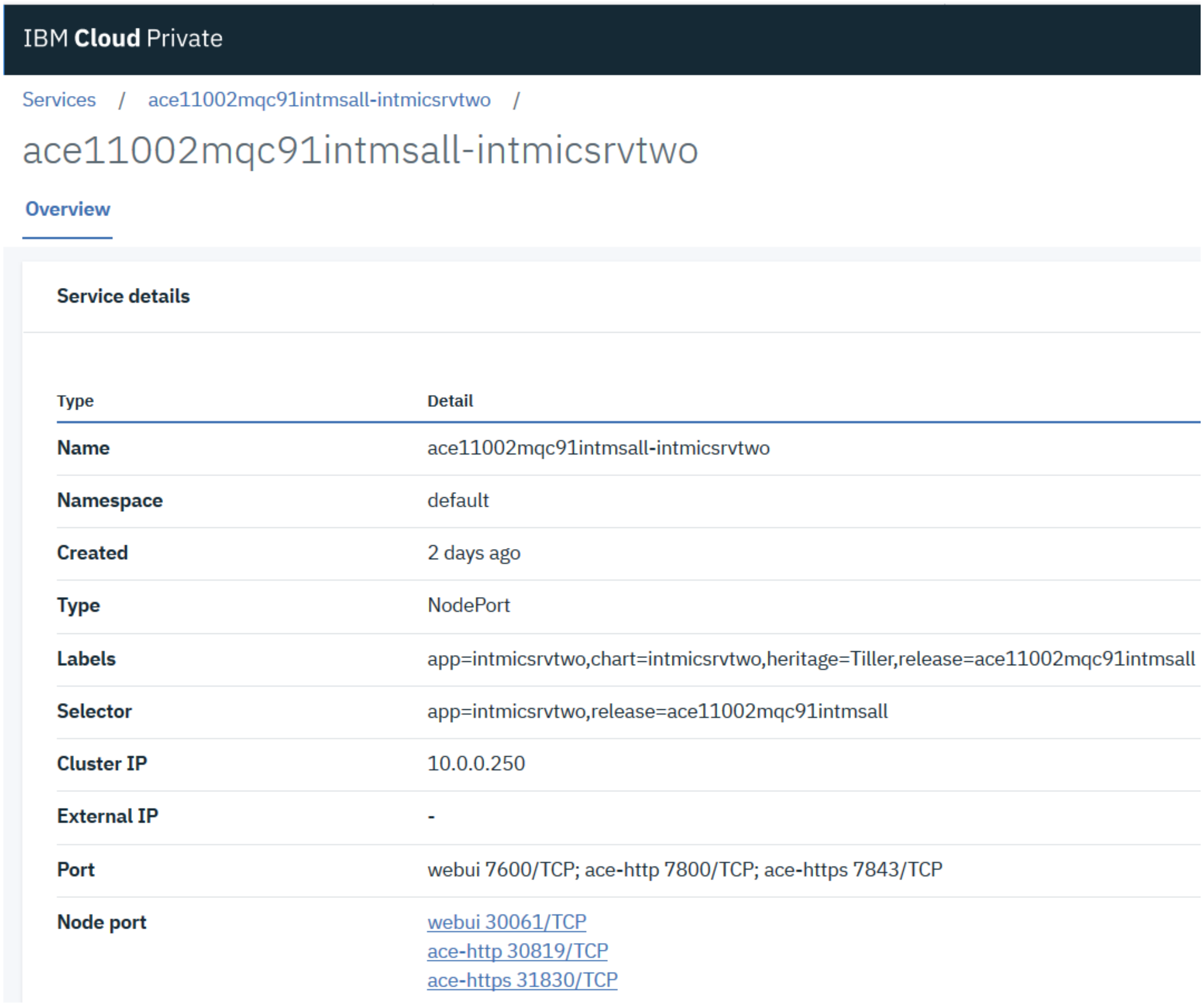
Service details

Type	Detail
Name	ace11002mqc91intmsall-intmicsrvone
Namespace	default
Created	2 days ago
Type	NodePort
Labels	app=intmicsrvone,chart=intmicsrvone,heritage=Tiller,release=ace11002mqc91intmsall
Selector	app=intmicsrvone,release=ace11002mqc91intmsall
Cluster IP	10.0.0.15
External IP	-
Port	webui 7600/TCP; ace-http 7800/TCP; ace-https 7843/TCP
Node port	webui 31378/TCP ace-http 32096/TCP ace-https 31148/TCP



Integration Micro Service Two

Lets take a look at integration micro service 2



IBM **Cloud** Private

Services / ace11002mqc91intmsall-intmicsrvtwo /

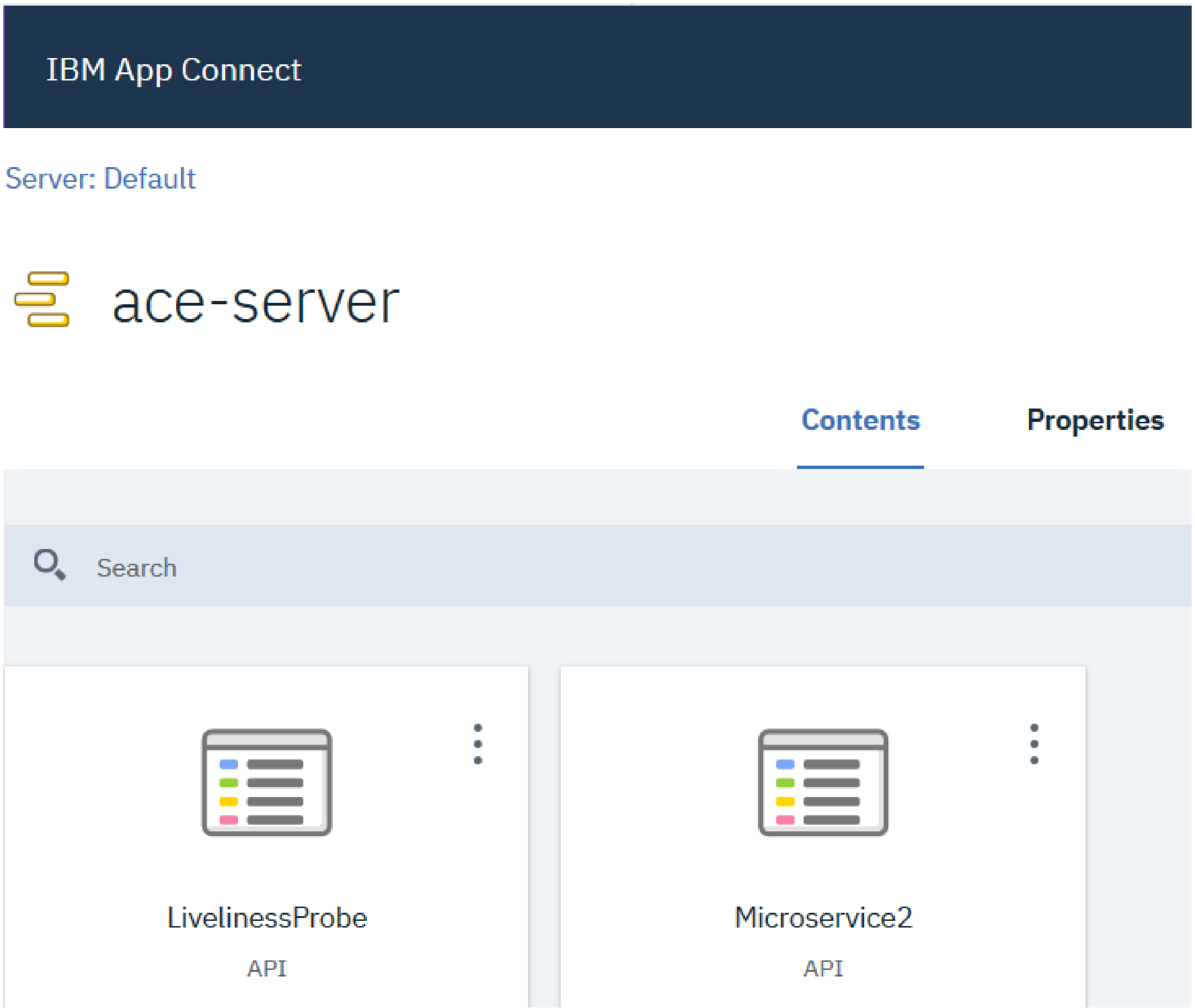
ace11002mqc91intmsall-intmicsrvtwo

Overview

Service details

Type	Detail
Name	ace11002mqc91intmsall-intmicsrvtwo
Namespace	default
Created	2 days ago
Type	NodePort
Labels	app=intmicsrvtwo,chart=intmicsrvtwo,heritage=Tiller,release=ace11002mqc91intmsall
Selector	app=intmicsrvtwo,release=ace11002mqc91intmsall
Cluster IP	10.0.0.250
External IP	-
Port	webui 7600/TCP; ace-http 7800/TCP; ace-https 7843/TCP
Node port	webui 30061/TCP ace-http 30819/TCP ace-https 31830/TCP

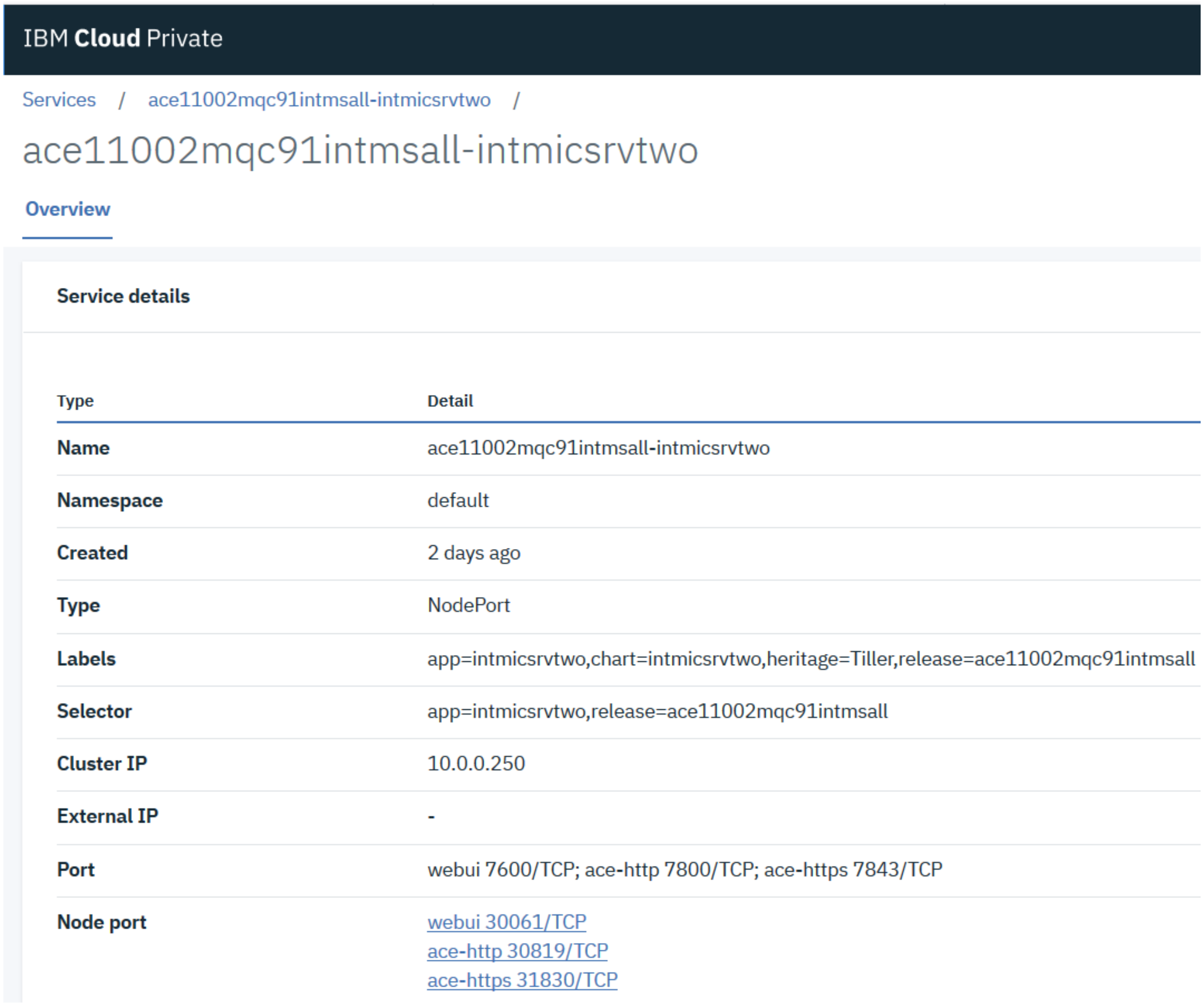
Select the webUI



Note we have the LivelinessProbe and MicroService2

Testing the Integration Micro Services Application

Return to the list of service details for the micro service 2 service.



IBM **Cloud** Private

Services / ace11002mqc91intmsall-intmicsrvtwo /

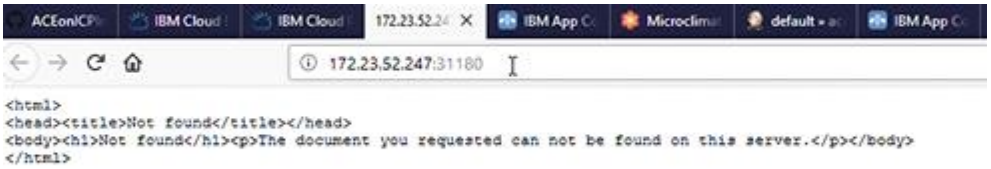
ace11002mqc91intmsall-intmicsrvtwo

Overview

Service details

Type	Detail
Name	ace11002mqc91intmsall-intmicsrvtwo
Namespace	default
Created	2 days ago
Type	NodePort
Labels	app=intmicsrvtwo,chart=intmicsrvtwo,heritage=Tiller,release=ace11002mqc91intmsall
Selector	app=intmicsrvtwo,release=ace11002mqc91intmsall
Cluster IP	10.0.0.250
External IP	-
Port	webui 7600/TCP; ace-http 7800/TCP; ace-https 7843/TCP
Node port	webui 30061/TCP ace-http 30819/TCP ace-https 31830/TCP

A quick way to get the URL for the ACE HTTP listener is to click on the link to bring up a browser window from which you can copy the actual URL



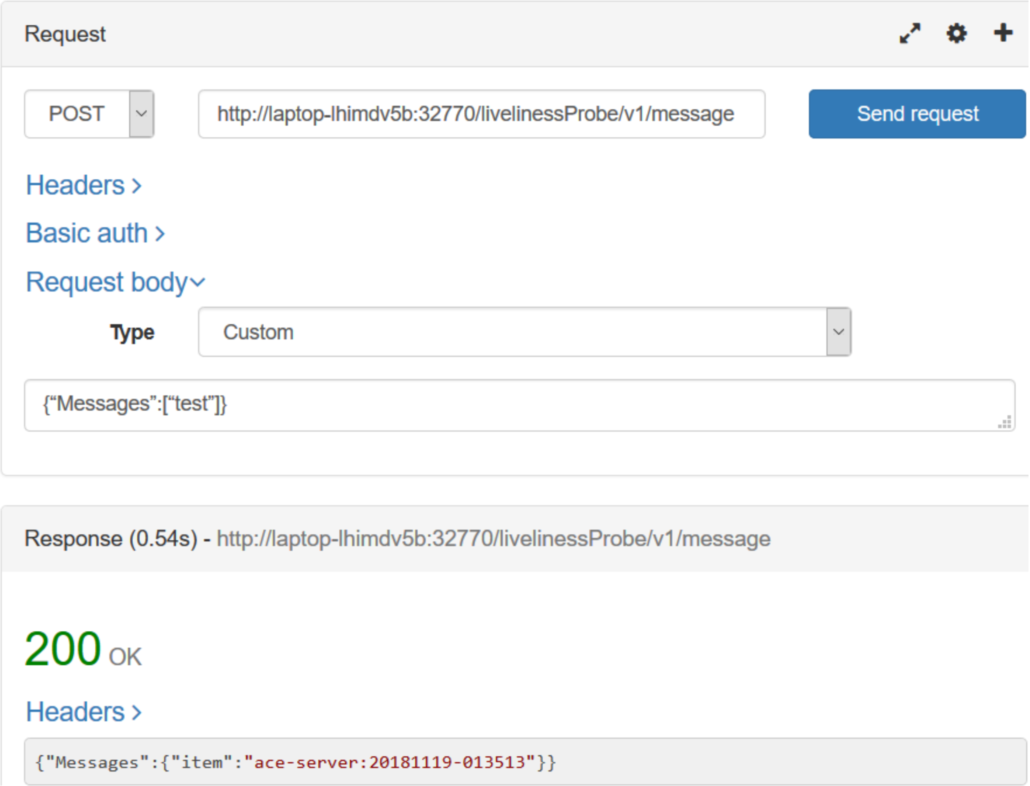
```
<html>
<head><title>Not found</title></head>
<body><h1>Not found</h1><p>The document you requested can not be found on this server.</p></body>
</html>
```

You can use the IPAddress and port number in the URL on the HTTP POST in a REST client.

Then you can test the Liveliness Probe

input: {“Messages”:[“test”]}

Example URL: http://172.23.52.247:31180/livelinessProbe/v1/message



POST http://laptop-lhimdv5b:32770/livelinessProbe/v1/message

Type Custom

{"Messages":["test"]}

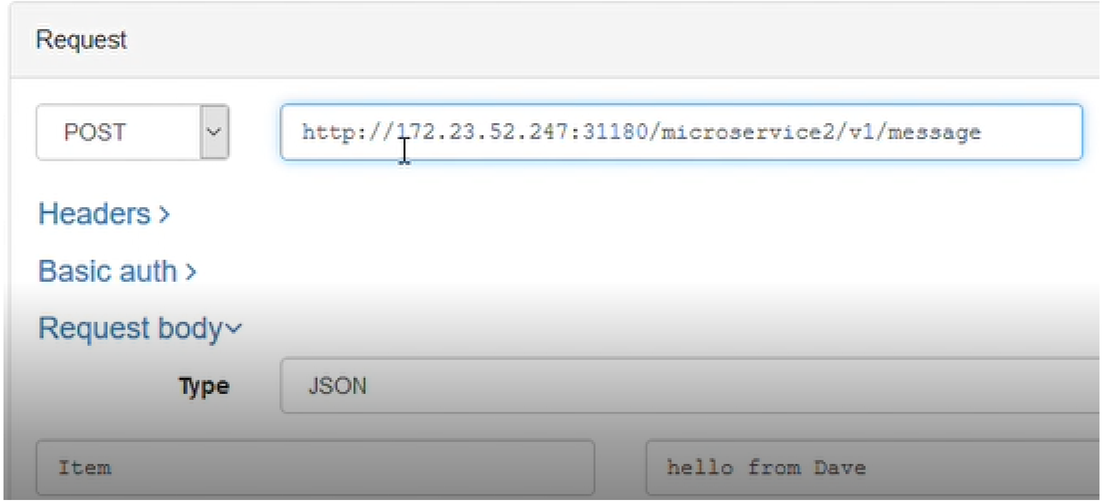
Response (0.54s) - http://laptop-lhimdv5b:32770/livelinessProbe/v1/message

200 OK

```
{"Messages":{"item":"ace-server:20181119-013513"}}
```

Note it returns the integration server name plus a current timestamp

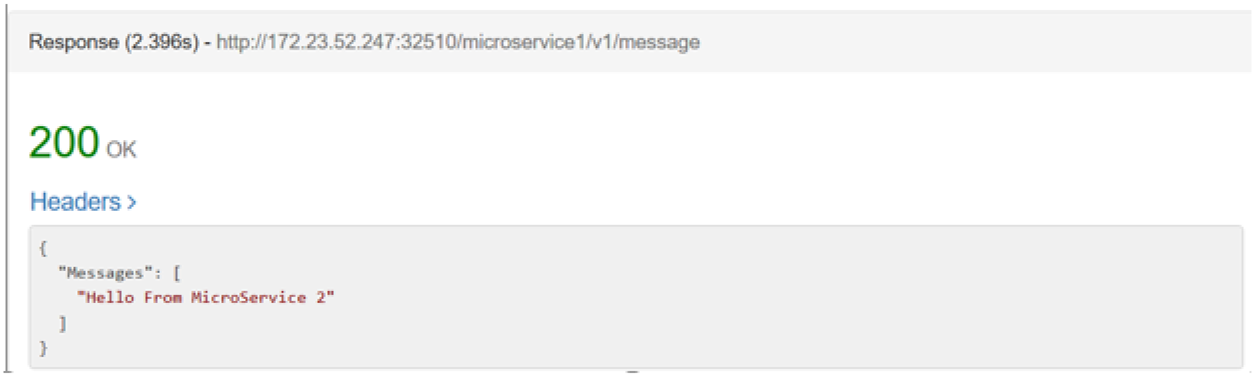
And you can test MicroService 2



POST http://172.23.52.247:31180/microservice2/v1/message

Type JSON

Item hello from Dave



Response (2.396s) - http://172.23.52.247:32510/microservice1/v1/message

200 OK

```
{
  "Messages": [
    "Hello From MicroService 2"
  ]
}
```

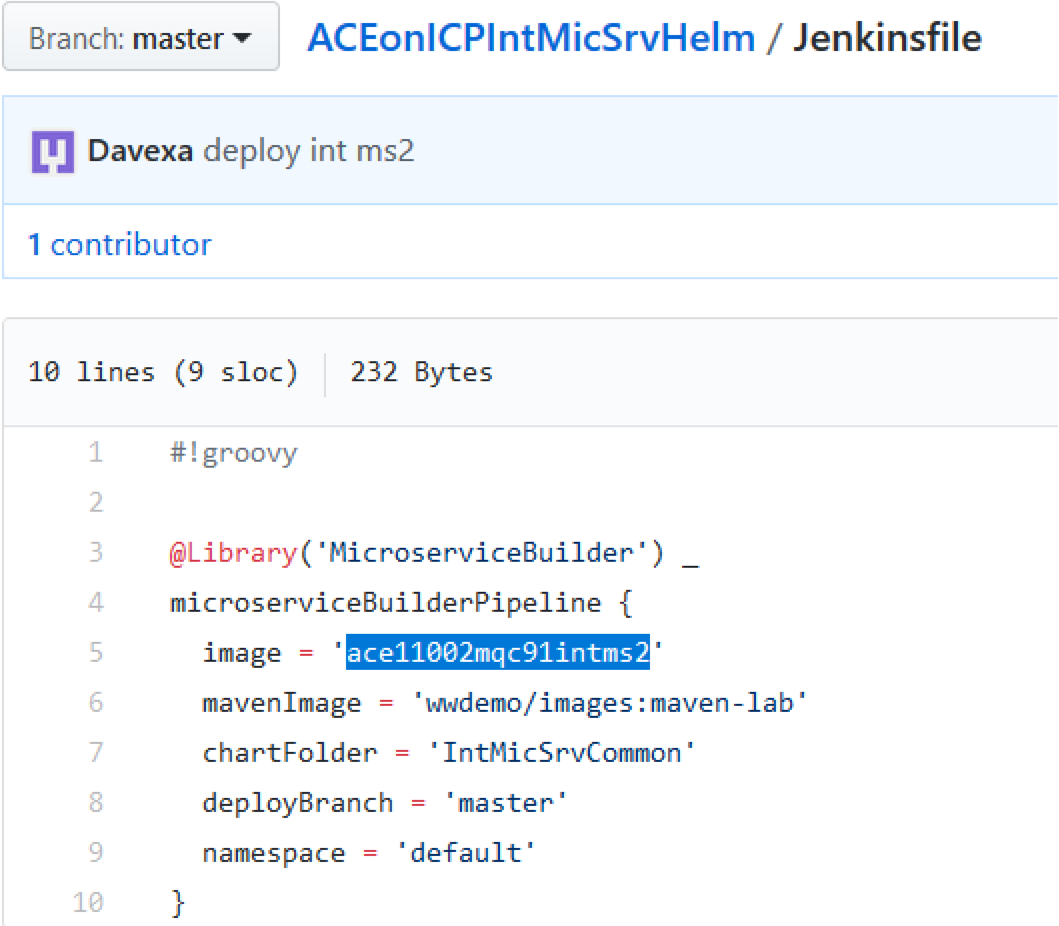
Deploying Micro Service 2 on it’s own for testing

Source Github repository - ACEonICPIntMicSrvHelm

https://github.com/DAVEXACOM/ACEonICPIntMicSrvHelm

Use this Helm Chart Repos to toggle between Micro Service 1 and Micro Service 2 helm release deployments for testing.

Modify the Jenkins File



Branch: master ▾ ACEonICPIntMicSrvHelm / Jenkinsfile

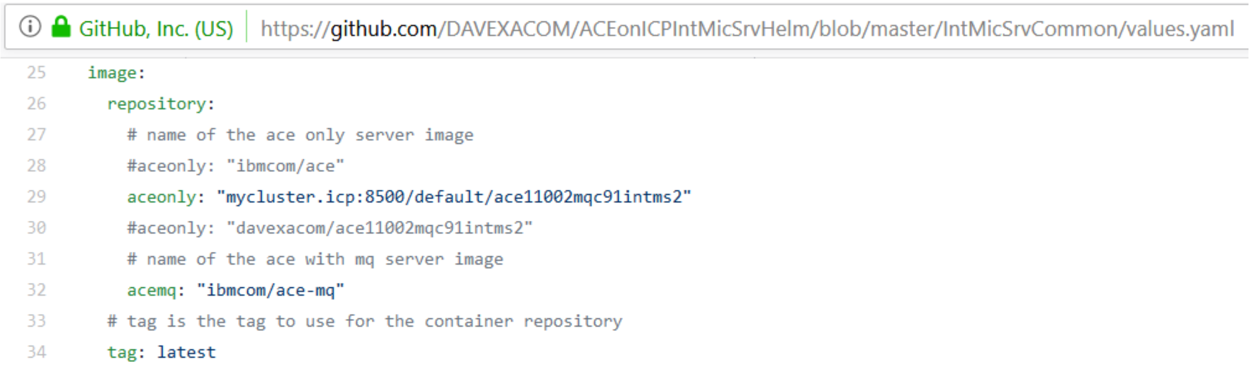
Davexa deploy int ms2

1 contributor

10 lines (9 sloc) | 232 Bytes

```
#!groovy

@Library('MicroserviceBuilder') _
microserviceBuilderPipeline {
  image = 'ace11002mqc91intms2'
  mavenImage = 'wwdemo/images:maven-lab'
  chartFolder = 'IntMicSrvCommon'
  deployBranch = 'master'
  namespace = 'default'
}
```

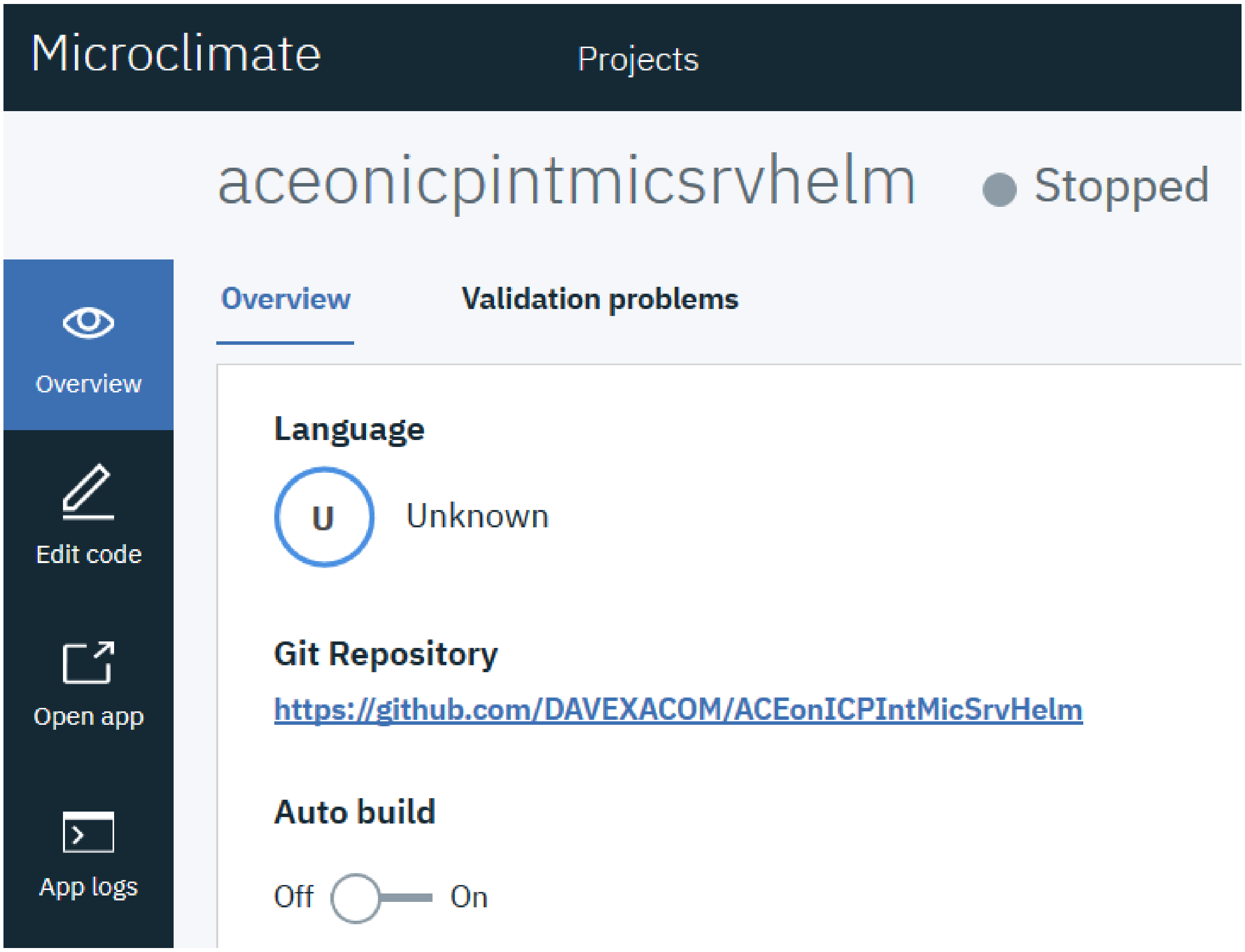
Modify the Values.yaml file



ⓘ 🔒 GitHub, Inc. (US) | https://github.com/DAVEXACOM/ACEonICPIntMicSrvHelm/blob/master/IntMicSrvCommon/values.yaml

```
image:
  repository:
    # name of the ace only server image
    #aceonly: "ibmcom/ace"
    aceonly: "mycluster.icp:8500/default/ace11002mqc91intms2"
    #aceonly: "davexacom/ace11002mqc91intms2"
    # name of the ace with mq server image
    acemq: "ibmcom/ace-mq"
  # tag is the tag to use for the container repository
  tag: latest
```

Microclimate Project



aceonicpintmicsrvhelm ● Stopped

Language

U Unknown

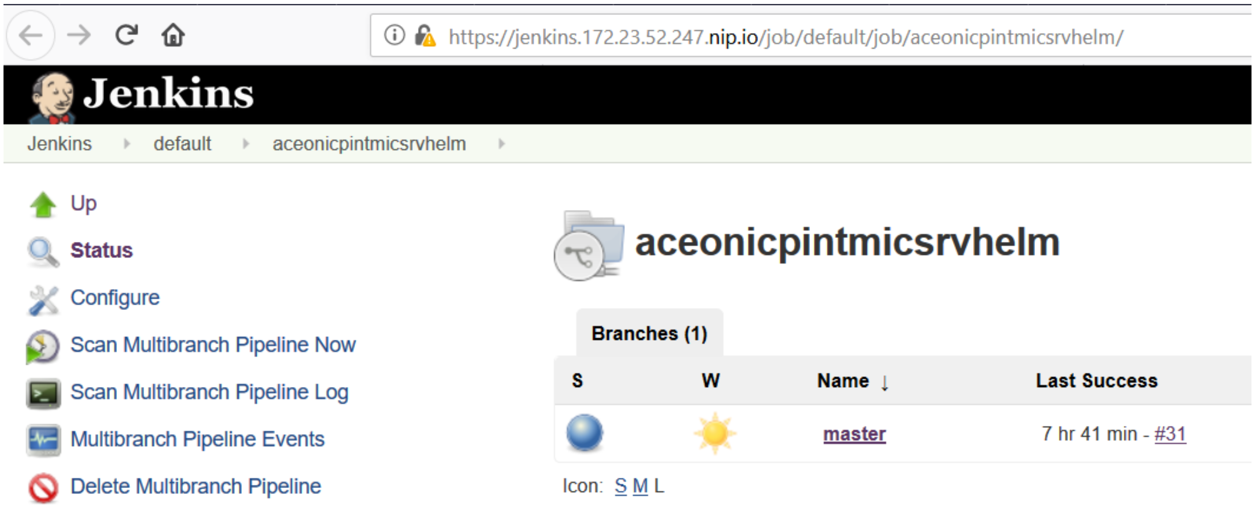
Git Repository

https://github.com/DAVEXACOM/ACEonICPIntMicSrvHelm

Auto build

Off ○— On

Jenkins Pipeline


Jenkins
Jenkins
default
aceonicpintmicsrvhelm
Up
Status
Configure
Scan Multibranch Pipeline Now
Scan Multibranch Pipeline Log
Multibranch Pipeline Events
Delete Multibranch Pipeline
Branches (1)
Icon: S M L

https://jenkins.172.23.52.247.nip.io/job/default/job/aceonicpintmicsrvhelm/

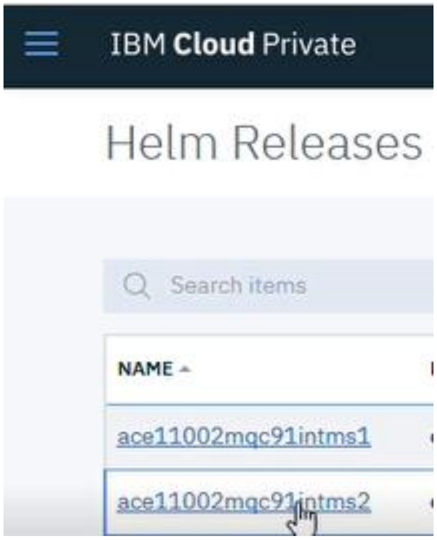
aceonicpintmicsrvhelm

S	W	Name ↓	Last Success
●	☀	master	7 hr 41 min - #31

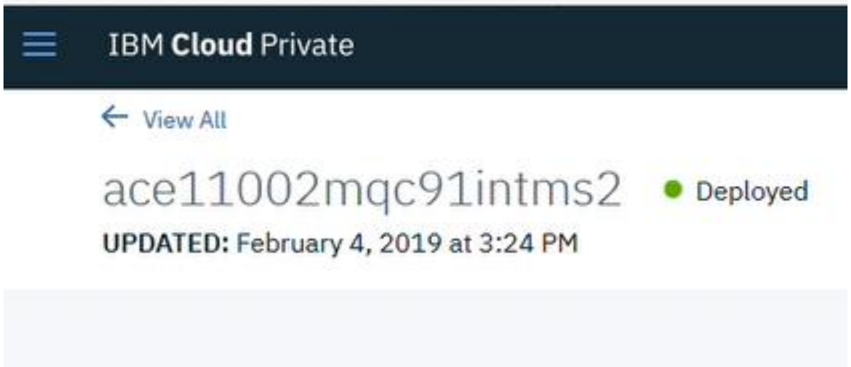
```
NAME:   ace11002mqc91intms2
LAST DEPLOYED: Fri Dec 14 01:07:51 2018
NAMESPACE: default
STATUS: DEPLOYED

RESOURCES:
==> v1/Secret
NAME                                 TYPE    DATA  AGE
ace11002mqc91intms2-intmicsrvcommon  Opaque  2     1s

==> v1/Service
NAME                                             TYPE       CLUSTER-IP  EXTERNAL-IP  PORT(S)                                          AGE
ace11002mqc91intms2-intmicsrvcommon-ace-metrics  ClusterIP  10.0.0.184  <none>       9483/TCP                                         1s
ace11002mqc91intms2-intmicsrvcommon              NodePort   10.0.0.60   <none>       7600:31961/TCP,7800:30571/TCP,7843:30979/TCP  1s

==> v1beta1/Deployment
NAME                                 DESIRED  CURRENT  UP-TO-DATE  AVAILABLE  AGE
ace11002mqc91intms2-intmicsrvcommon  3        2        2           0          1s

==> v1/Pod(related)
NAME                                                  READY  STATUS             RESTARTS  AGE
ace11002mqc91intms2-intmicsrvcommon-795888d88b-b84vg  0/1    Pending            0         1s
ace11002mqc91intms2-intmicsrvcommon-795888d88b-dx579  0/1    Pending            0         1s
ace11002mqc91intms2-intmicsrvcommon-795888d88b-gkxwq  0/1    ContainerCreating  0         1s
```

Testing Integration Micro Service 2 standalone

From ICP Console -> Workloads->Helm Releases

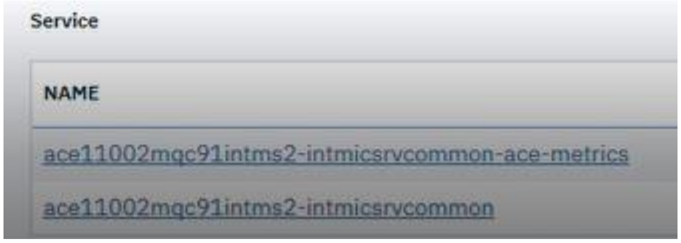


Select the Integration Microservice 2 release



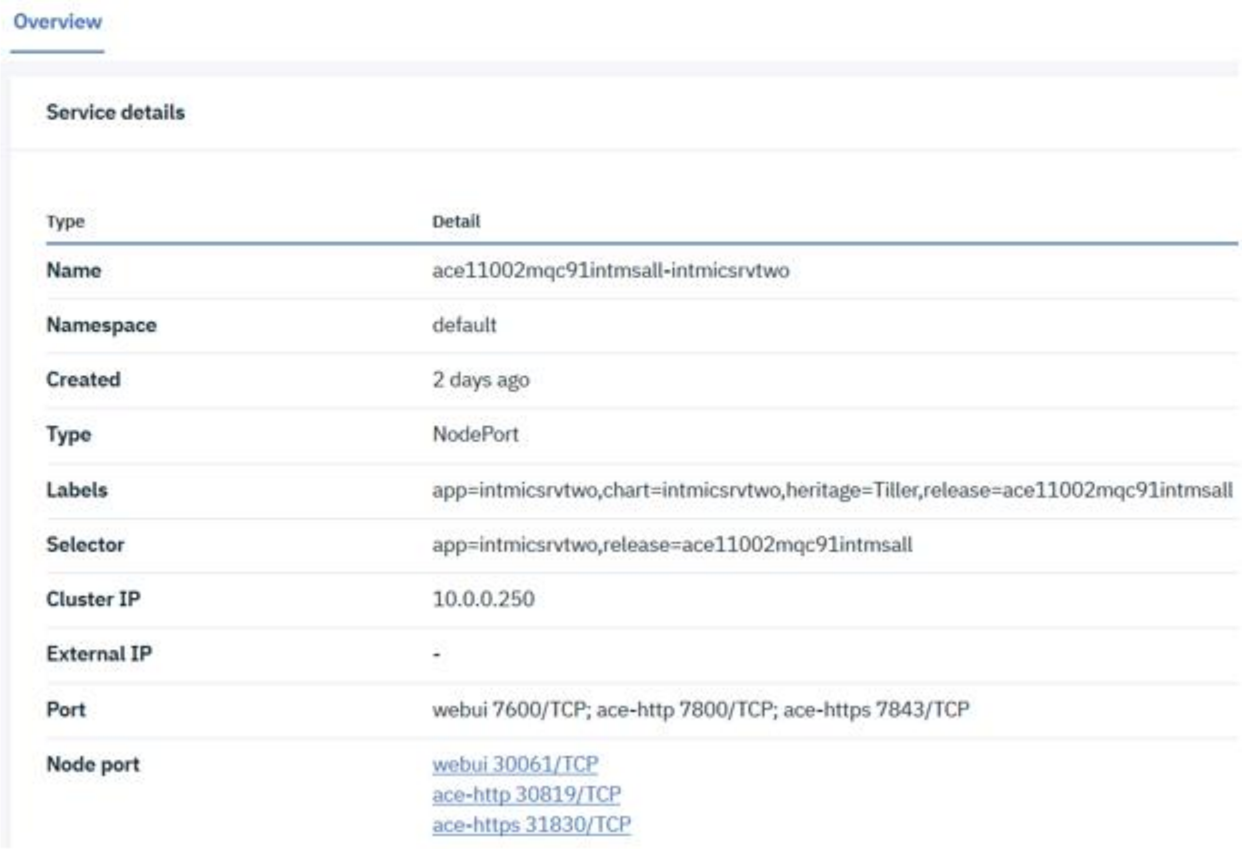
UPDATED: February 4, 2019 at 3:24 PM

Scroll down to services and select the link (not the ace-metrics link)



Service

NAME
ace11002mqc91intms2-intmicsrvcommon-ace-metrics
ace11002mqc91intms2-intmicsrvcommon

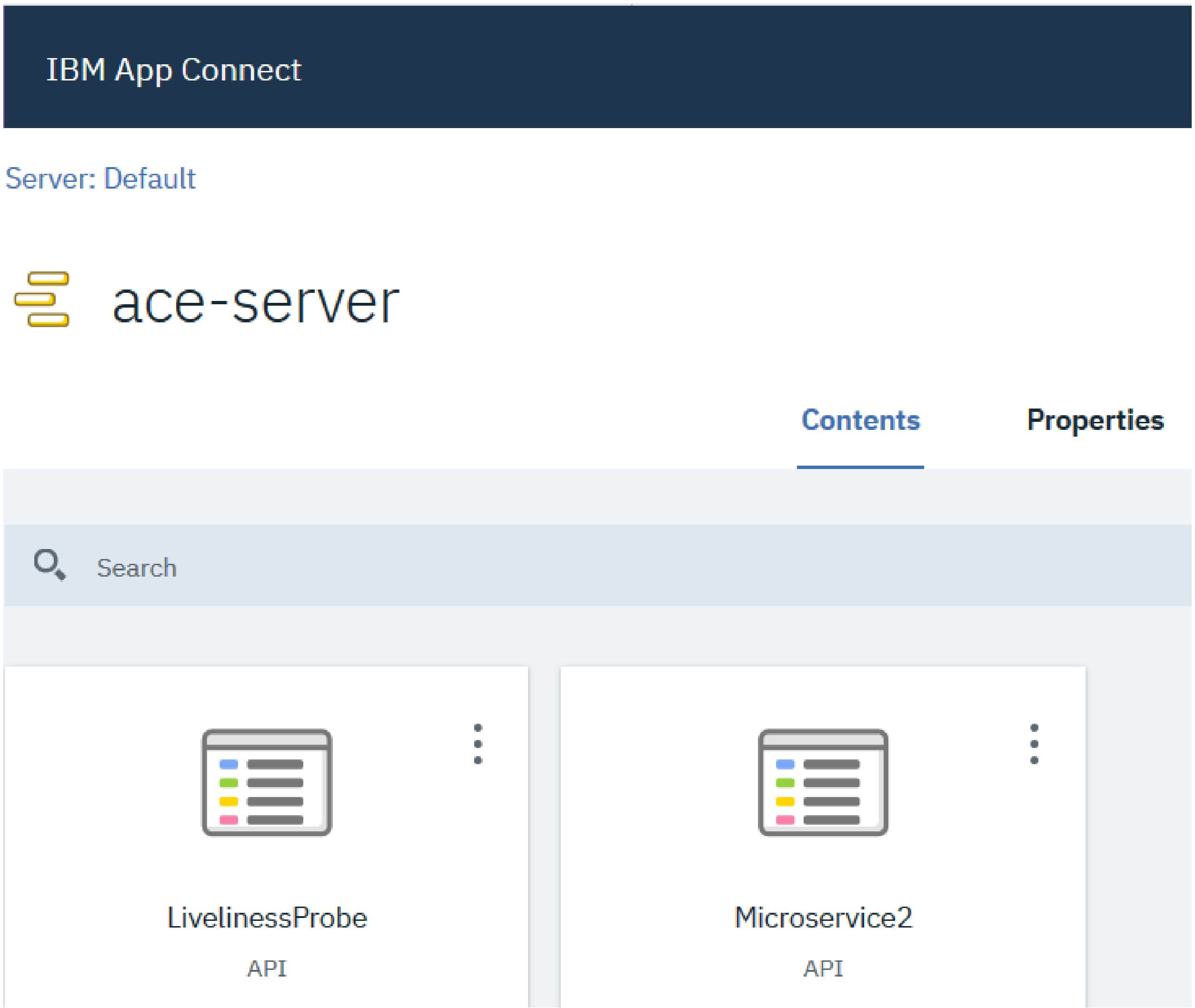


Overview

Service details

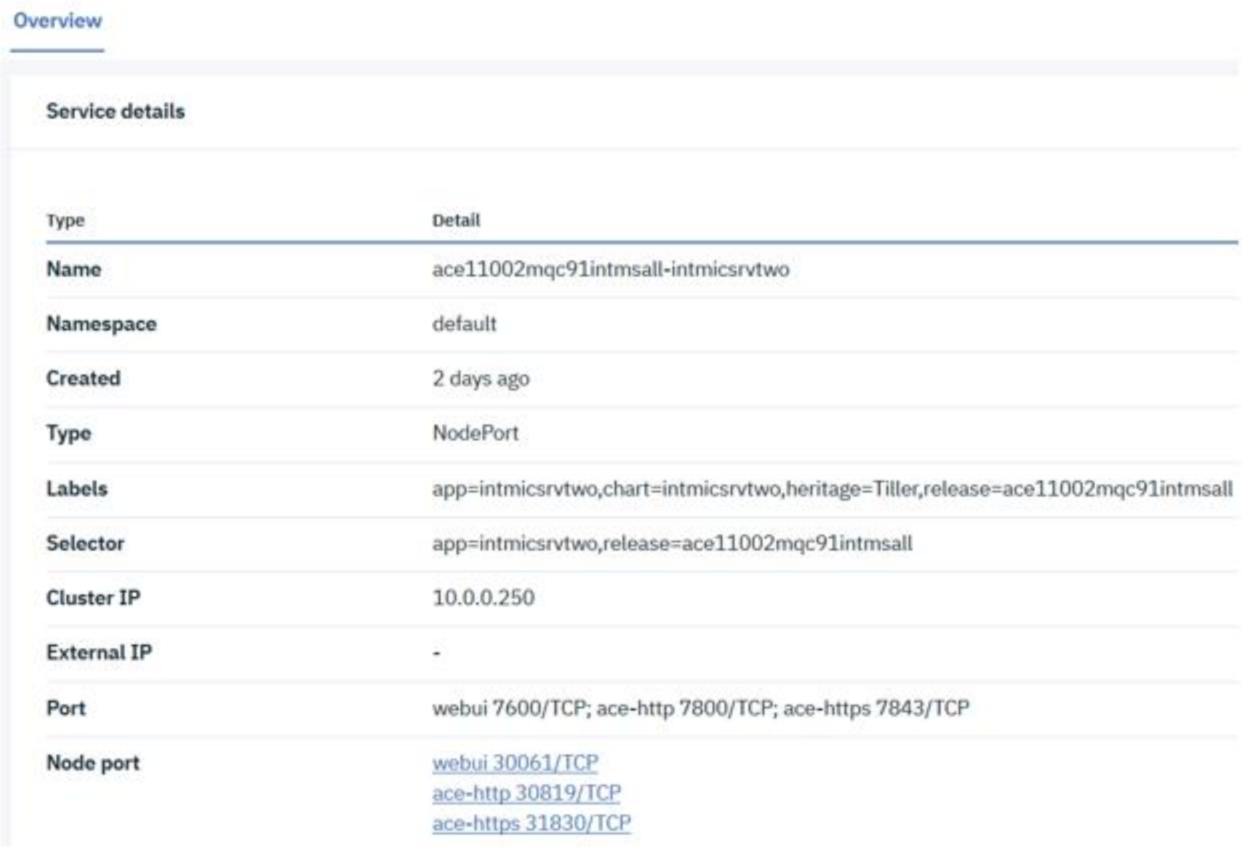
Type	Detail
Name	ace11002mqc91intmsall-intmicsrvtwo
Namespace	default
Created	2 days ago
Type	NodePort
Labels	app=intmicsrvtwo,chart=intmicsrvtwo,heritage=Tiller,release=ace11002mqc91intmsall
Selector	app=intmicsrvtwo,release=ace11002mqc91intmsall
Cluster IP	10.0.0.250
External IP	-
Port	webui 7600/TCP; ace-http 7800/TCP; ace-https 7843/TCP
Node port	webui 30061/TCP ace-http 30819/TCP ace-https 31830/TCP

Select the webUI to bring up the ACE WebUI



The WebUI verifies that the Liveliness Probe and Microservice2 are deployed

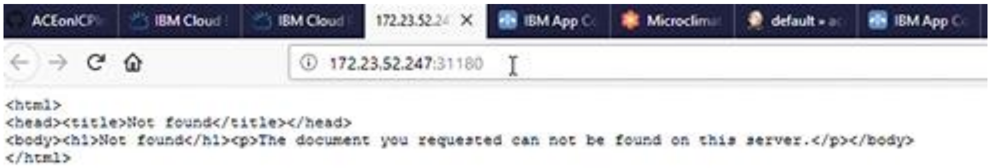
Return to the service details



Service details

Type	Detail
Name	ace11002mqc91intmsall-intmicsrvtwo
Namespace	default
Created	2 days ago
Type	NodePort
Labels	app=intmicsrvtwo,chart=intmicsrvtwo,heritage=Tiller,release=ace11002mqc91intmsall
Selector	app=intmicsrvtwo,release=ace11002mqc91intmsall
Cluster IP	10.0.0.250
External IP	-
Port	webui 7600/TCP; ace-http 7800/TCP; ace-https 7843/TCP
Node port	webui 30061/TCP ace-http 30819/TCP ace-https 31830/TCP

A quick way to get the URL for the ACE HTTP listener is to click on the link to bring up a browser window from which you can copy the actual URL



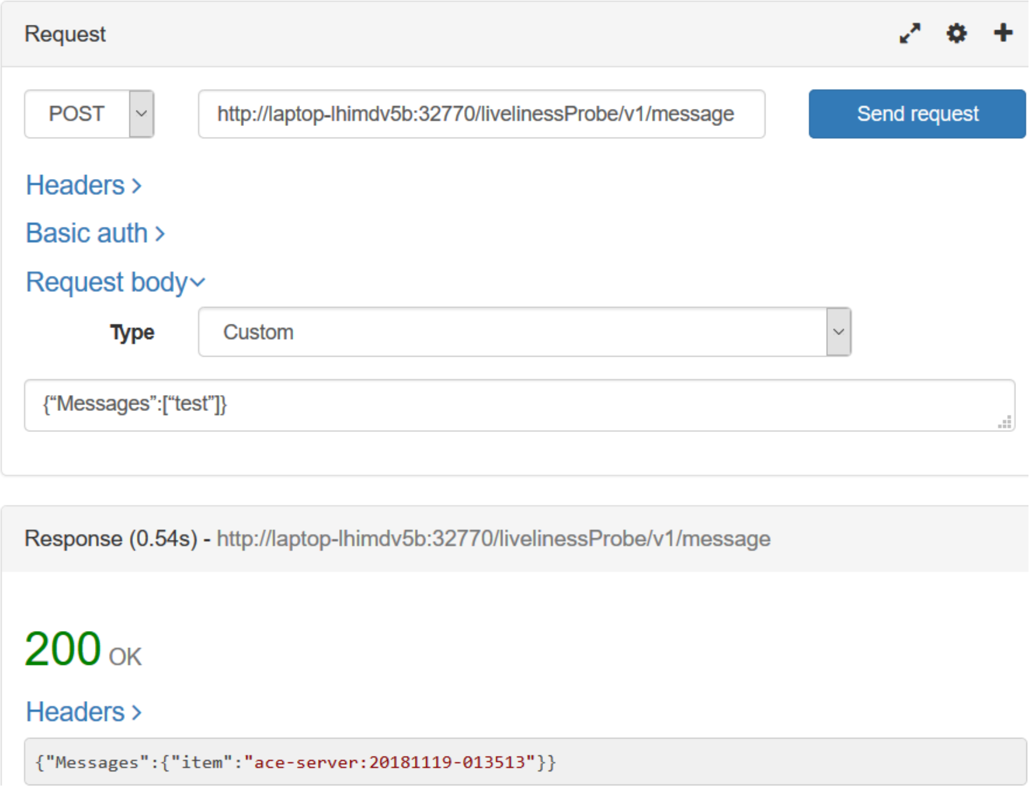
```
<html>
<head><title>Not found</title></head>
<body><h1>Not found</h1><p>The document you requested can not be found on this server.</p></body>
</html>
```

You can use the IPAddress and port number in the URL on the HTTP POST in a REST client.

Then you can test the Liveliness Probe

input: {“Messages”:[“test”]}

Example URL: http://172.23.52.247:31180/livelinessProbe/v1/message



POST http://laptop-lhimdv5b:32770/livelinessProbe/v1/message

Type: Custom

{“Messages”:[“test”]}

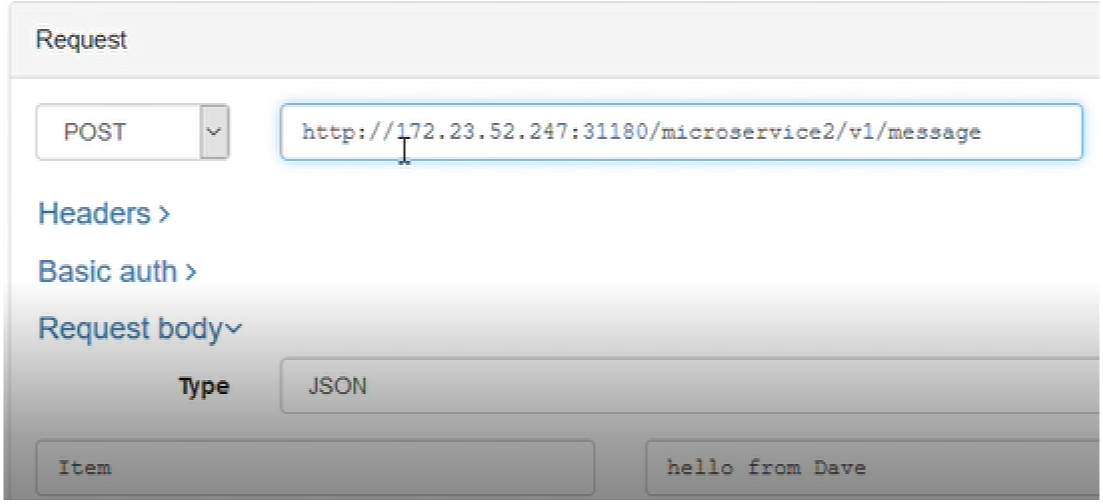
Response (0.54s) - http://laptop-lhimdv5b:32770/livelinessProbe/v1/message

200 OK

```
{"Messages":{"item":"ace-server:20181119-013513"}}
```

Note it returns the integration server name plus a current timestamp

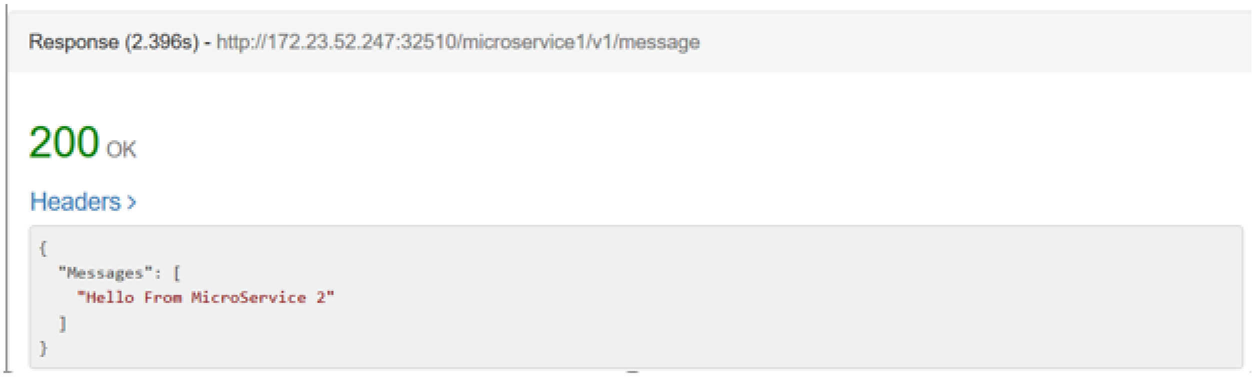
And you can test MicroService 2 in a similar way.



POST http://172.23.52.247:31180/microservice2/v1/message

Type: JSON

Item: hello from Dave



Response (2.396s) - http://172.23.52.247:32510/microservice1/v1/message

200 OK

```
{
  "Messages": [
    "Hello From MicroService 2"
  ]
}
```

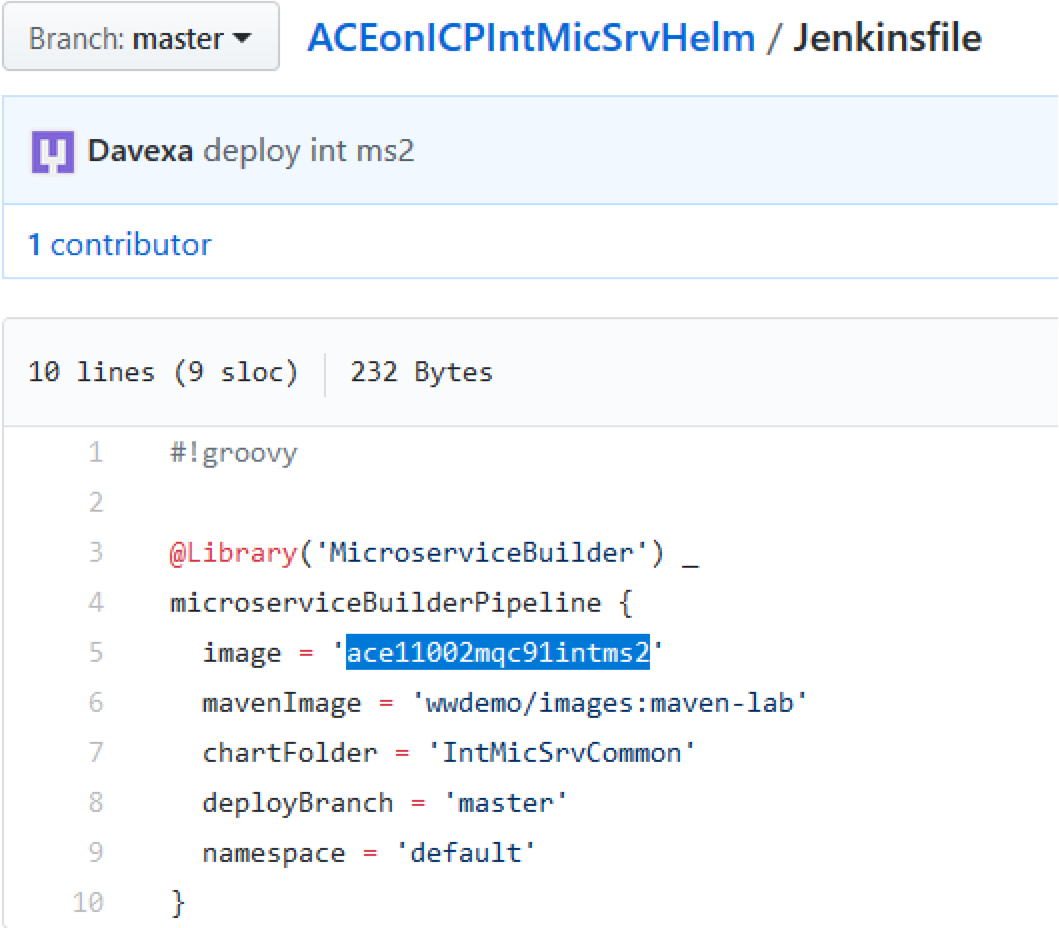
Deploying Micro Service 1 on it’s own for testing

Source Github repository - ACEonICPIntMicSrvHelm

https://github.com/DAVEXACOM/ACEonICPIntMicSrvHelm

Use this Helm Chart Repos to toggle between Micro Service 1 and Micro Service 2 helm release deployments for testing.

Modify the Jenkins File (in this example you’ll need to switch from 2 to 1)



Branch: master ▾ ACEonICPIntMicSrvHelm / Jenkinsfile

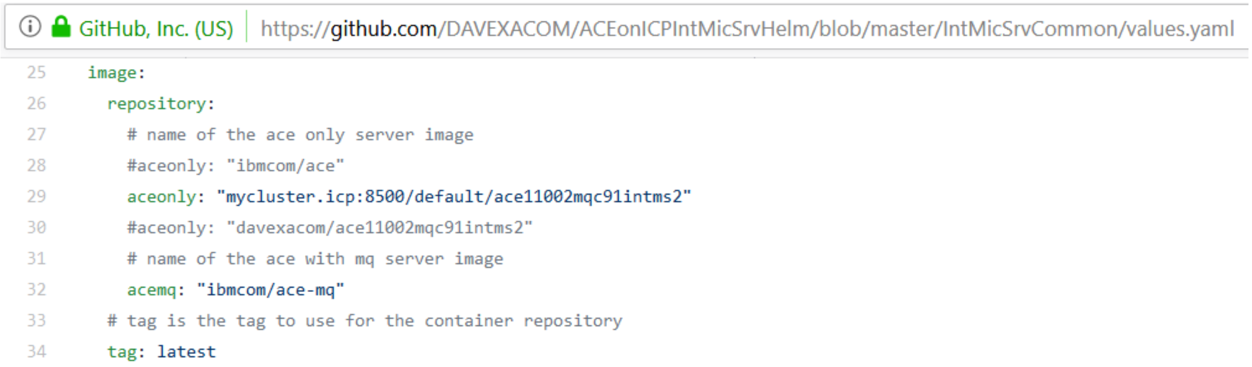
Davexa deploy int ms2

1 contributor

10 lines (9 sloc) | 232 Bytes

```
#!groovy

@Library('MicroserviceBuilder') _
microserviceBuilderPipeline {
  image = 'ace11002mqc91intms2'
  mavenImage = 'wwdemo/images:maven-lab'
  chartFolder = 'IntMicSrvCommon'
  deployBranch = 'master'
  namespace = 'default'
}
```

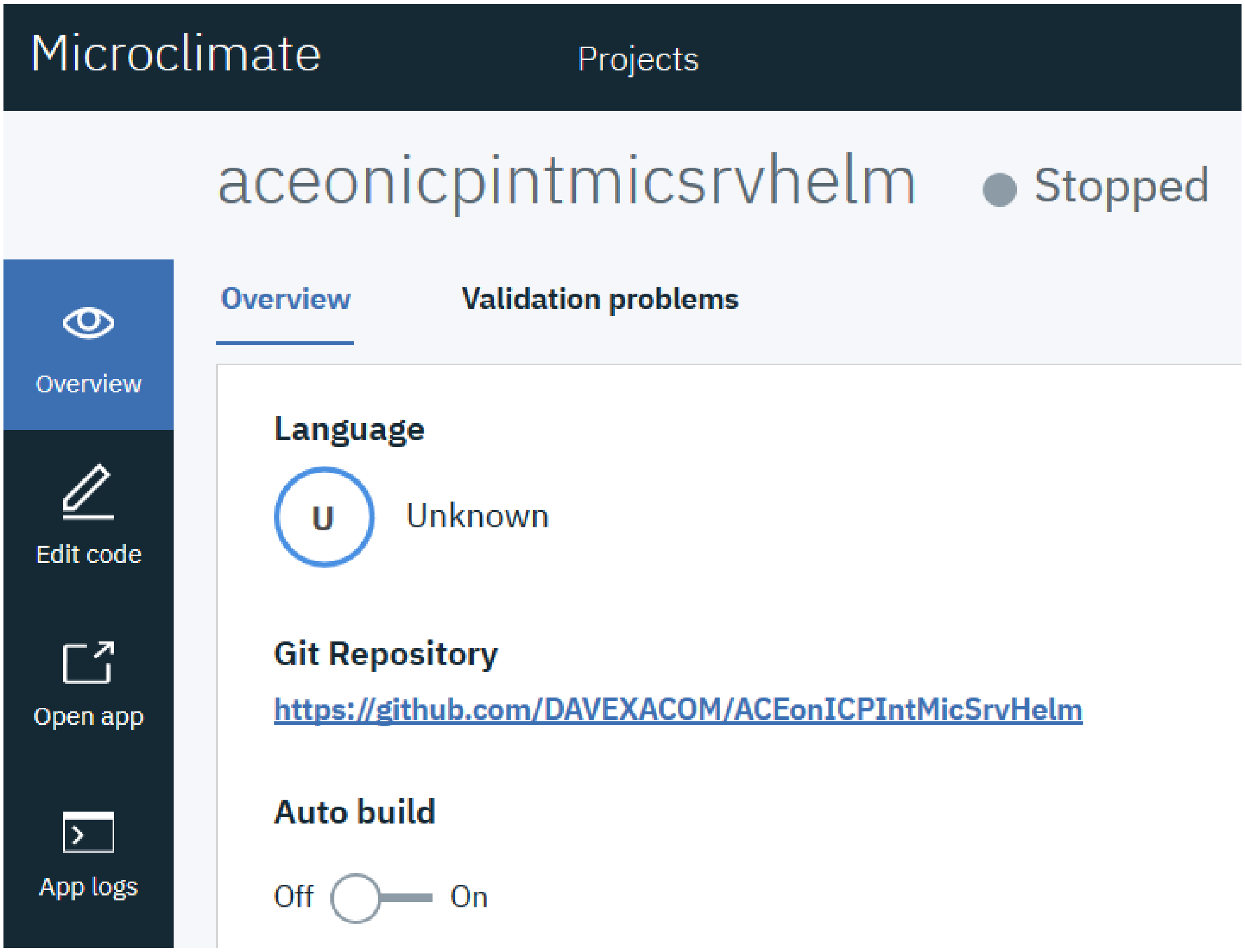
Modify the Values.yaml file



GitHub, Inc. (US) | https://github.com/DAVEXACOM/ACEonICPIntMicSrvHelm/blob/master/IntMicSrvCommon/values.yaml

```
image:
  repository:
    # name of the ace only server image
    #aceonly: "ibmcom/ace"
    aceonly: "mycluster.icp:8500/default/ace11002mqc91intms2"
    #aceonly: "davexacom/ace11002mqc91intms2"
    # name of the ace with mq server image
    acemq: "ibmcom/ace-mq"
  # tag is the tag to use for the container repository
  tag: latest
```

Microclimate Project



aceonicpintmicsrvhelm ● Stopped

Language

U Unknown

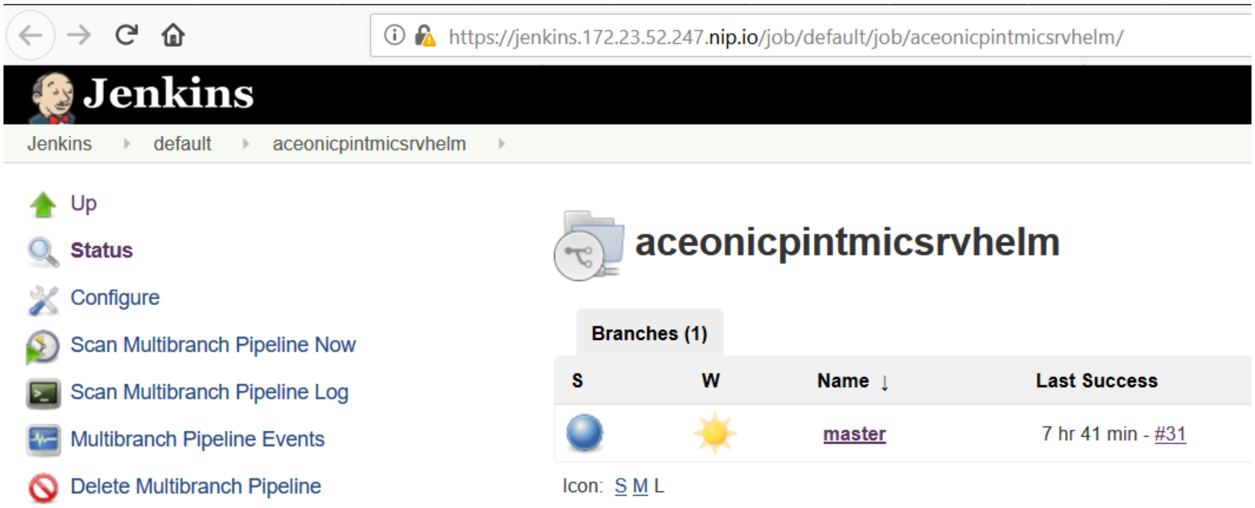
Git Repository

https://github.com/DAVEXACOM/ACEonICPIntMicSrvHelm

Auto build

Off ○— On

Jenkins Pipeline


Jenkins
Jenkins › default › aceonicpintmicsrvhelm ›
Up
Status
Configure
Scan Multibranch Pipeline Now
Scan Multibranch Pipeline Log
Multibranch Pipeline Events
Delete Multibranch Pipeline
Icon: S M L

https://jenkins.172.23.52.247.nip.io/job/default/job/aceonicpintmicsrvhelm/

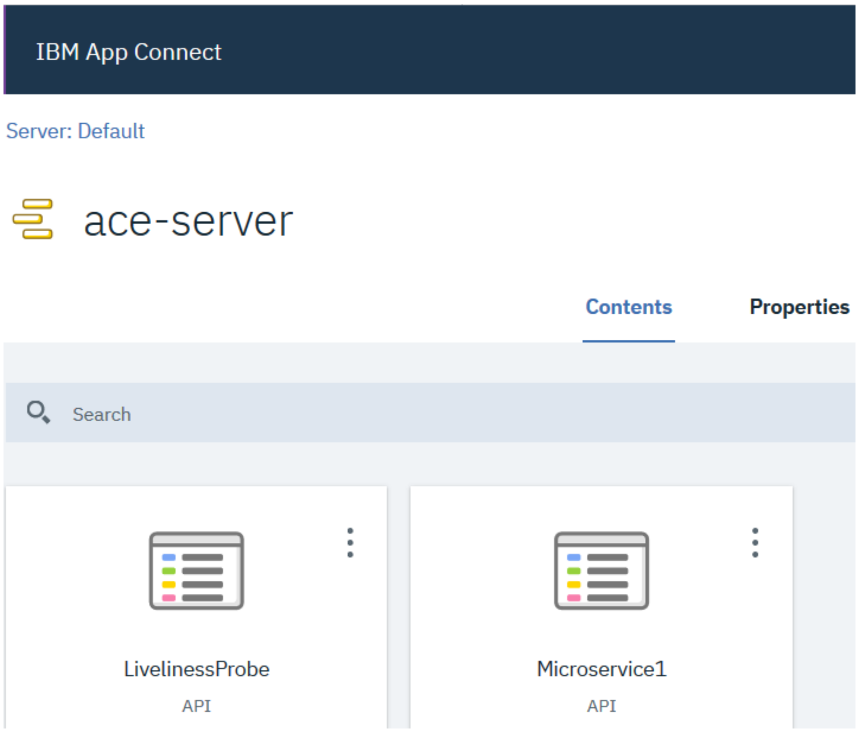
aceonicpintmicsrvhelm

Branches (1)

S	W	Name ↓	Last Success
●	☀	master	7 hr 41 min - #31

```
NAME:   ace11002mqc91intms1
LAST DEPLOYED: Mon Dec  3 04:40:48 2018
NAMESPACE: default
STATUS: DEPLOYED

RESOURCES:
==> v1/Secret
NAME                                 TYPE    DATA  AGE
ace11002mqc91intms1-intmicsrvcommon  Opaque  2     1s

==> v1/Service
NAME                                             TYPE       CLUSTER-IP  EXTERNAL-IP  PORT(S)
ace11002mqc91intms1-intmicsrvcommon-ace-metrics  ClusterIP  10.0.0.230  <none>       9483/TCP
ace11002mqc91intms1-intmicsrvcommon              NodePort   10.0.0.129  <none>       7600:31614/TCP,7800:32143/TCP,7843:30530/TCP

==> v1beta1/Deployment
NAME                                 DESIRED  CURRENT  UP-TO-DATE  AVAILABLE  AGE
ace11002mqc91intms1-intmicsrvcommon  3        3        3           0          1s

==> v1/Pod(related)
NAME                                                 READY  STATUS             RESTARTS  AGE
ace11002mqc91intms1-intmicsrvcommon-86fdd665-c9ksl  0/1    ContainerCreating  0         1s
ace11002mqc91intms1-intmicsrvcommon-86fdd665-dvj6b  0/1    ContainerCreating  0         1s
ace11002mqc91intms1-intmicsrvcommon-86fdd665-k5957  0/1    ContainerCreating  0         1s
```

Explore the helm release on ICP and navigate to the services and open the ACE WebUI

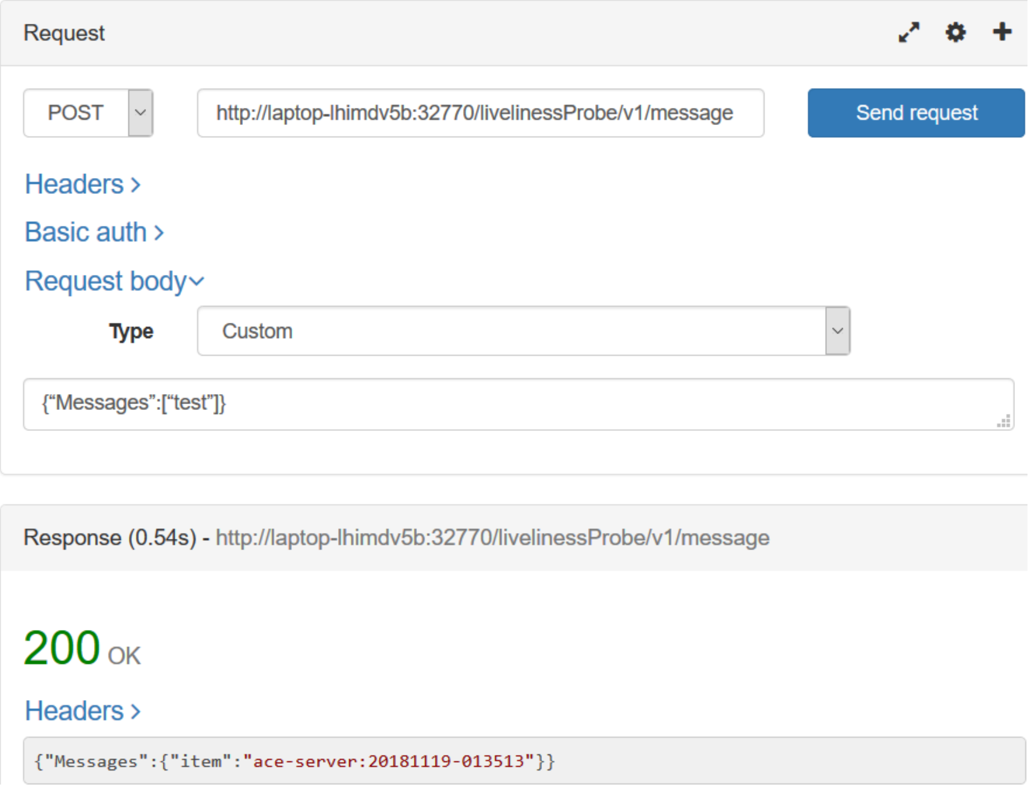


Testing Integration Micro Service 1 standalone

Micro service 1 is designed to call micro service 2 so testing it standalone will fail because micro service 2 is not deployed. You can however test the liveliness probe.

Testing Liveliness Probe

Note it returns the integration server name plus a current timestamp for input: {“Messages”:[“test”]}



POST http://laptop-lhimdv5b:32770/livelinessProbe/v1/message

Type Custom

```
{“Messages”:[“test”]}
```

Response (0.54s) - http://laptop-lhimdv5b:32770/livelinessProbe/v1/message

200 OK

```
{"Messages":{"item":"ace-server:20181119-013513"}}
```