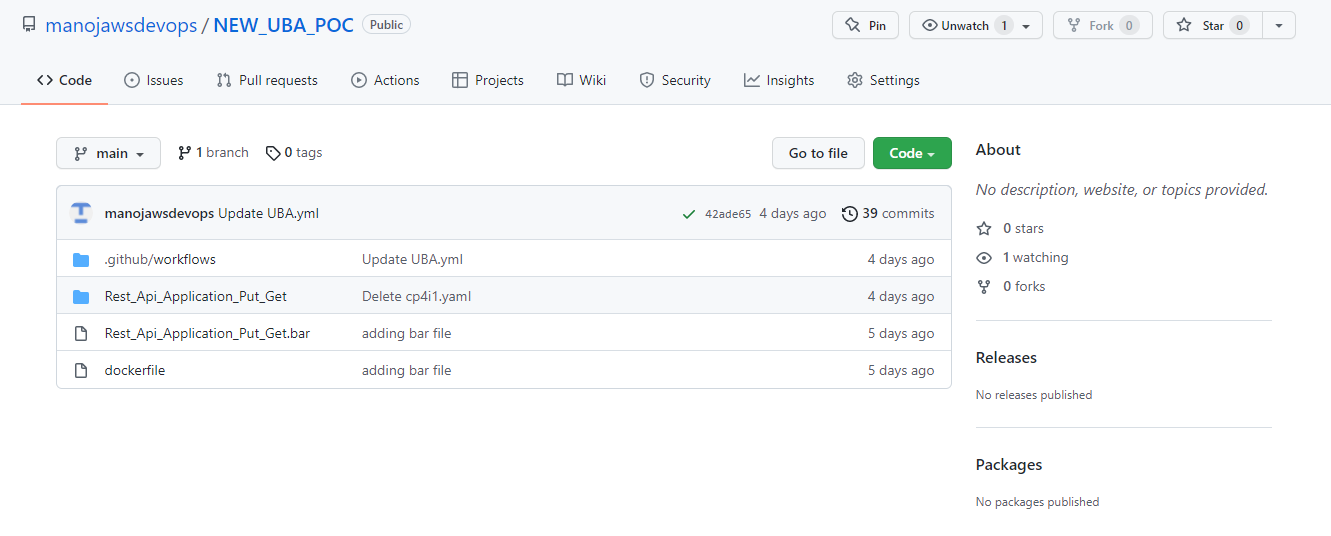
CICD Implementation on ACE to CP4i

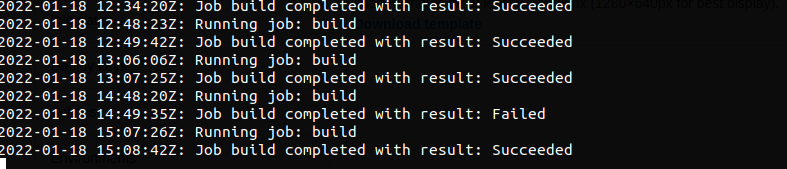
**Prerequisites:**

* Bar file
* GitHub Workflow
* Yaml file For integration server

Create a repository in GitHub and push our code to Repository



We are running on self-hosted runner so we need to create self-hosted runner & configure on repository



Create Github workflow actions yml file

Yml file:

|  |  |
| --- | --- |
| name: UBA\_CICD |  |
|  |  |
|  | on: |
|  | push: |
|  | branches: [ main ] |
|  | pull\_request: |
|  | branches: [ main ] |
|  |  |
|  | env: |
|  | BAR\_CREATION: /opt/ace-11.0.0.7/tools/mqsicreatebar |
|  | BAR\_STORE: /home/bandaru/Templates |
|  | OCR\_USERNAME: ${{secrets.OCR\_USERNAME}} |
|  | DEFAULT\_ROUTE: ${{secrets.DEFAULT\_ROUTE}} |
|  | OC\_TOKEN: ${{secrets.OC\_TOKEN}} |
|  | OC\_SERVER: ${{secrets.OC\_SERVER}} |
|  |  |
|  | jobs: |
|  | build: |
|  | runs-on: self-hosted |
|  |  |
|  | steps: |
|  | - uses: actions/checkout@v2 |
|  | - name: Set up JDK 11 |
|  | uses: actions/setup-java@v2 |
|  | with: |
|  | java-version: '11' |
|  | distribution: 'adopt' |
|  | - name: Building bar file |
|  | run: $BAR\_CREATION -data /root -b $BAR\_STORE/Rest\_Api\_Application\_Put\_Get.bar -a Rest\_Api\_Application\_Put\_Get |
|  |  |
|  | - name: Building Docker Image,Tag & Pushing to OCR Internal Registry |
|  | run: | |
|  | docker build -t cp4i:v11 -f dockerfile . |
|  | docker tag cp4i:v11 $DEFAULT\_ROUTE/cp4i/cp4i:v1 |
|  | docker login -u $OCR\_USERNAME -p $(oc whoami -t) http://default-route-openshift-image-registry.cp4intpg-wdc04-7upm3s-8946bbc006b7c6eb0829d088919818bb-0000.us-east.containers.appdomain.cloud |
|  | docker push $DEFAULT\_ROUTE/cp4i/cp4i:v1 |
|  |  |
|  | - name: oc login |
|  | run: | |
|  | cd /usr/local/bin/ |
|  | ./oc login --token=$OC\_TOKEN --server=$OC\_SERVER |
|  | ./oc project cp4i |
|  | ./oc apply -f /root/Rest\_Api\_Application\_Put\_Get/cp4i.yaml |
|  | ./oc apply -f /root/Rest\_Api\_Application\_Put\_Get/cp4i1.yaml |

In the above Github workflow actions script we are working on

* Creating the barfile and storing the barfile in path
* Building the image using dockerfile.
* Tagging and pushing the builded image to openshift internal registry
* Creating two integration servers
* One is by using barurl creating integration server
* Second is by using image we are creating integration server

**yaml file using barurl:**

**Yaml file:**

apiVersion: appconnect.ibm.com/v1beta1

kind: IntegrationServer

metadata:

name: uba-integration-server-one

namespace: cp4i

spec:

adminServerSecure: true

createDashboardUsers: true

designerFlowsOperationMode: disabled

enableMetrics: true

license:

accept: true

license: L-KSBM-C5JEHP

use: CloudPakForIntegrationNonProduction

pod:

containers:

runtime:

resources:

limits:

cpu: 300m

memory: 350Mi

requests:

cpu: 300m

memory: 300Mi

replicas: 1

router:

timeout: 120s

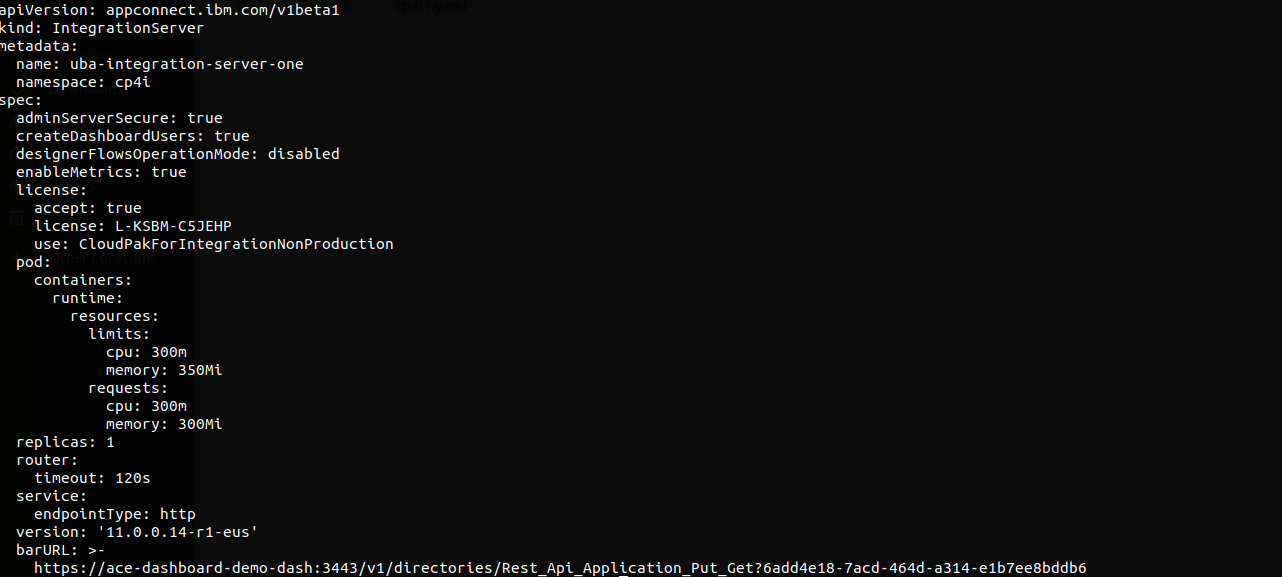
service:

endpointType: http

version: '11.0.0.14-r1-eus'

barURL: >-

https://ace-dashboard-demo-dash:3443/v1/directories/Rest\_Api\_Application\_Put\_Get?6add4e18-7acd-464d-a314-e1b7ee8bddb6



**Note:** For bar url we need to import the bar file in app connect and take that bar url and put it under the barurl field in yaml file

**Yaml file using image:**

**Yaml File:**

apiVersion: appconnect.ibm.com/v1beta1

kind: IntegrationServer

metadata:

name: uba-integration-server-second

namespace: cp4i

spec:

adminServiceSecure: true

license:

accept: true

license: L-KSBM-C5JEHP

use: CloudPakForIntegrationNonProduction

pod:

containers:

runtime:

image: image-registry.openshift-image-registry.svc:5000/cp4i/cp4i:v1

resources:

limits:

cpu: 900m

memory: 900Mi

requests:

cpu: 300m

memory: 300Mi

router:

timeout: 10s

designerFlowOperation: disabled

service:

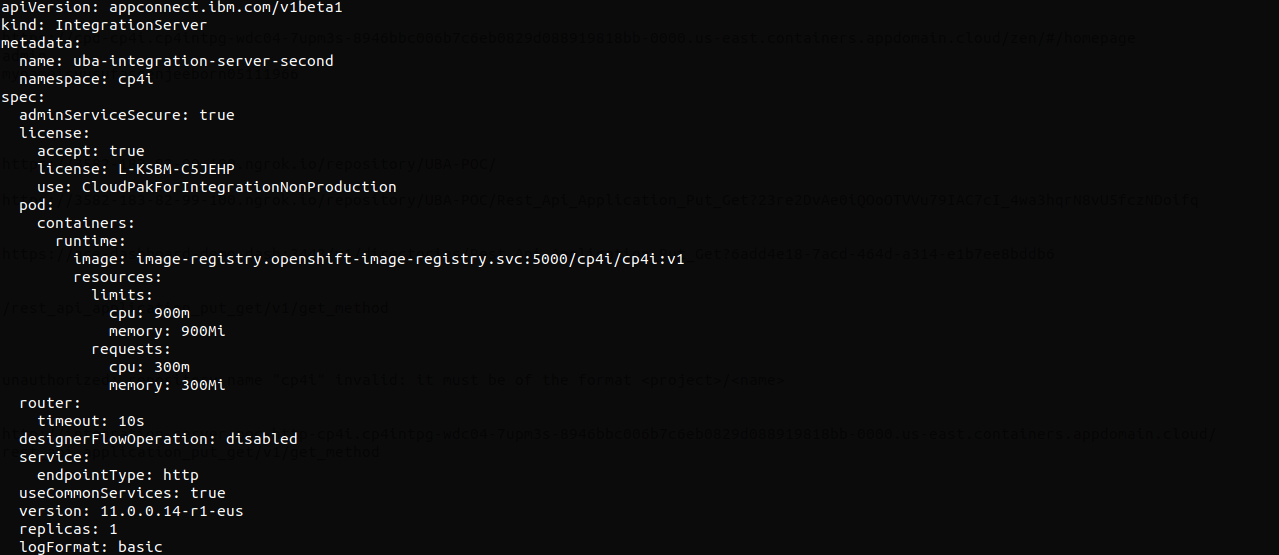
endpointType: http

useCommonServices: true

version: 11.0.0.14-r1-eus

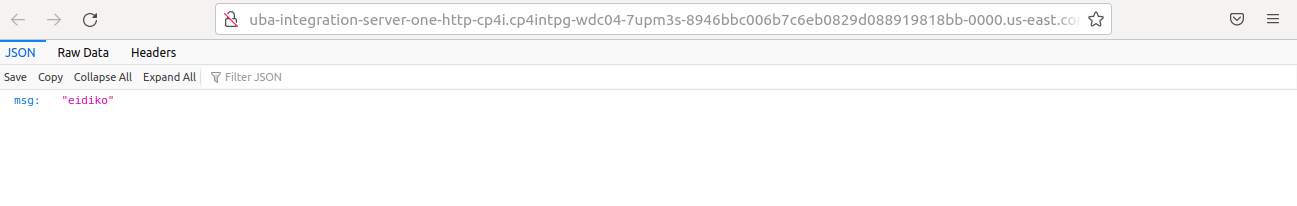
replicas: 1

logFormat: basic



Next type the below commands for creating integration servers

|  |
| --- |
| oc apply -f /root/Rest\_Api\_Application\_Put\_Get/cp4i.yaml |
| oc apply -f /root/Rest\_Api\_Application\_Put\_Get/cp4i1.yaml |
| Go and check whether the integration servers created or not in App connect dash board |
| Integration servers was successfully with names uba-integration-server-one, uba-integration-server-second  Next goto routes and access the application  First integrationserver route |



Second integrationserver route

