**Sample Application Migration on CP4I**

Jun Shen

[junshen@au1.ibm.com](mailto:junshen@au1.ibm.com)

Version: 0.4

Revision History

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Number** | **Date** | **Section** | **Changed By** | **Comments** |
| 0.1 | 16/02/2023 |  | Jun Shen | Initial creation |
| 0.2 | 01/03/2023 |  | Jun Shen | Add MQ session |
| 0.3 | 07/03/2023 |  | Jun Shen | Add SpringBoot session |
| 0.4 | 08/03/2023 |  | Jun Shen | Updated based on peer review |

Reference Documents

|  |  |
| --- | --- |
| **Name** | **Link** |
| CP4I topology design | CP4I Topology design for WSV-v0.7.docx |
| CP4I Deployment | CP4I Deployment for WSV Non-Prod v0.5.docx |

Table of Contents

[1. APIc Migration 4](#_Toc129187832)

[2. ACE Migration 19](#_Toc129187833)

[3. DataPower Migration 28](#_Toc129187834)

[4. MQ Migration 35](#_Toc129187835)

[5. SpringBoot Migration 39](#_Toc129187836)

[6. ArgoCD deployment 41](#_Toc129187837)

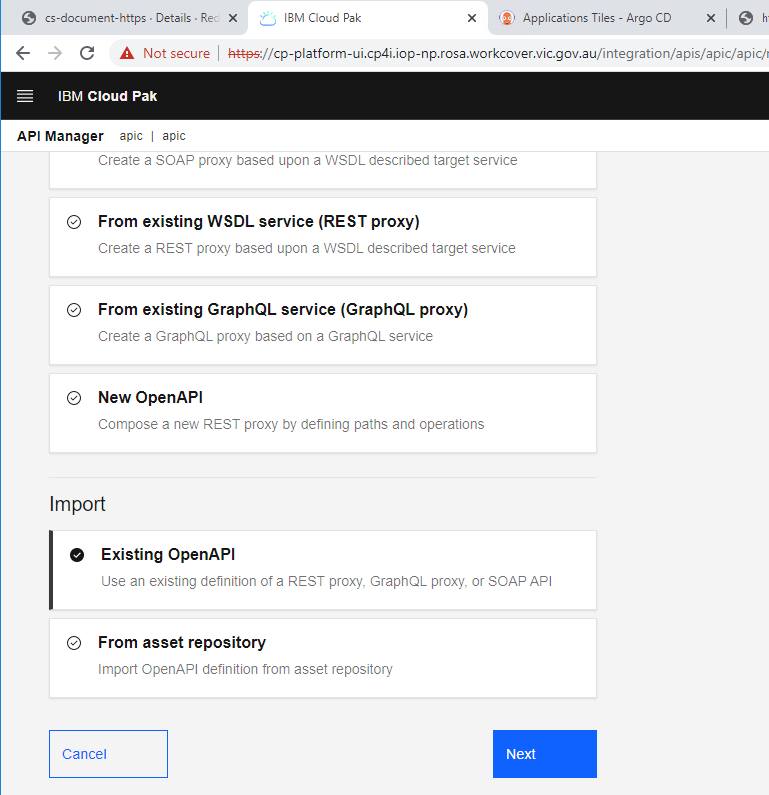
[7. Troubleshooting 42](#_Toc129187838)

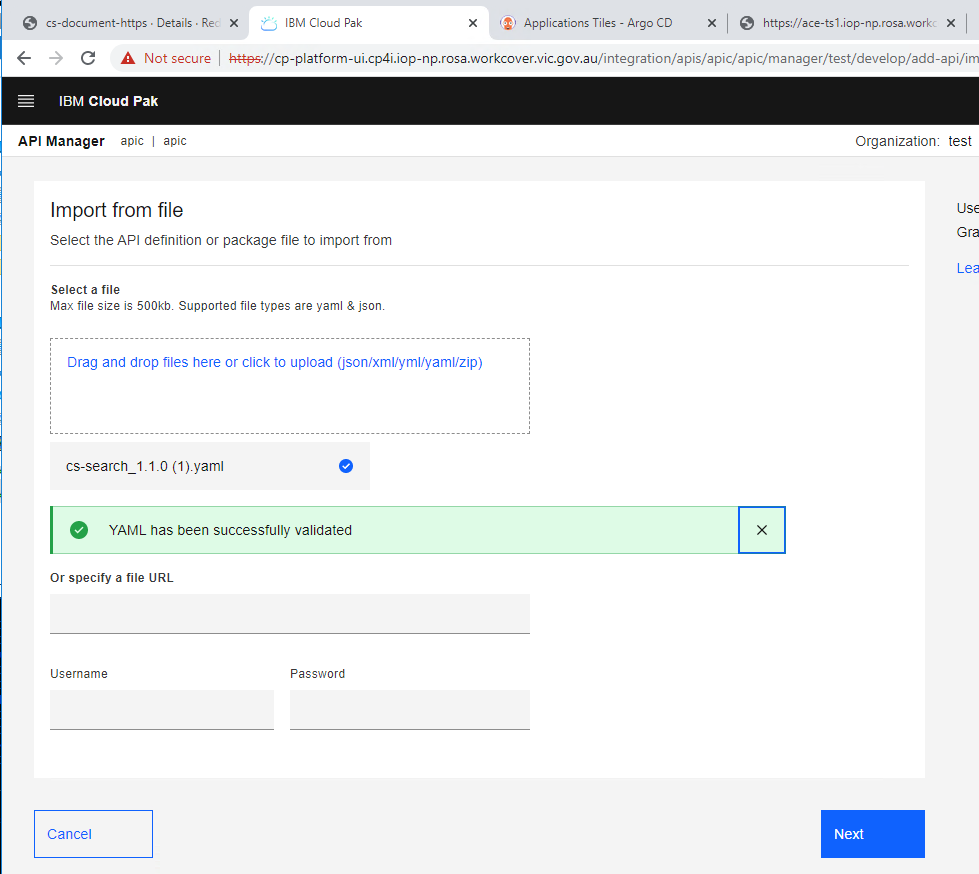
# **APIc Migration**

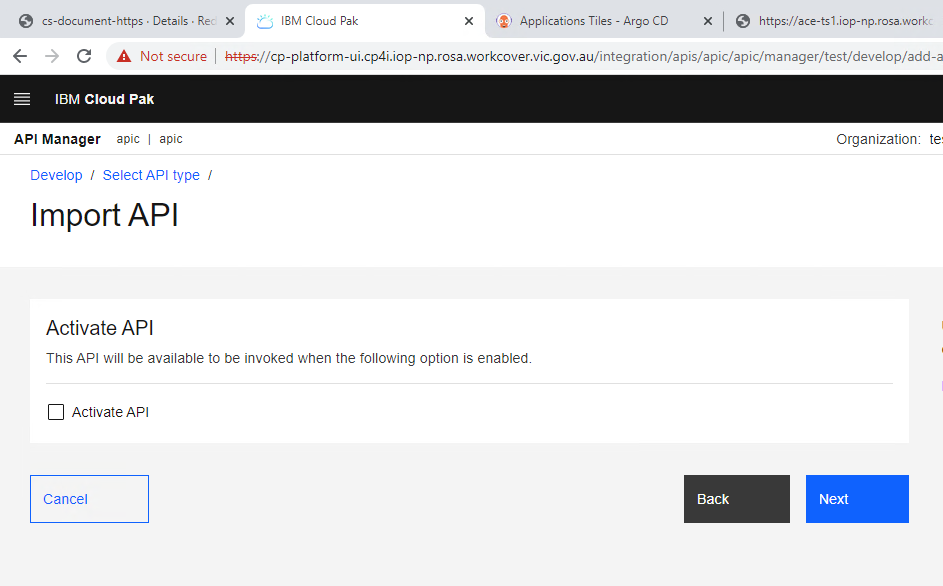
* 1. **Clone APIC artefacts from Bitbucket**

git clone <https://bitbucket-aws.itss.vic.gov.au/scm/rosa/rosa-app-migration.git>

* 1. **Import API yaml file**







* 1. **Migrate API v5 to v10**
* Change the gateway type

From

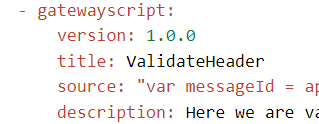


To

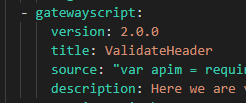


* Change the policy version from 1.0.0 to 2.0.0

Eg, from

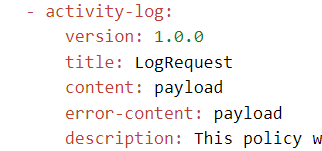


To

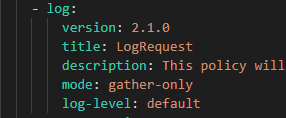


* Change activity-log

From

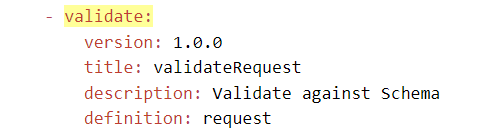


To

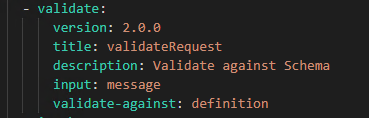


* Update validate policy

From



To



* Add “apim” var before calling apim.getvariable

From

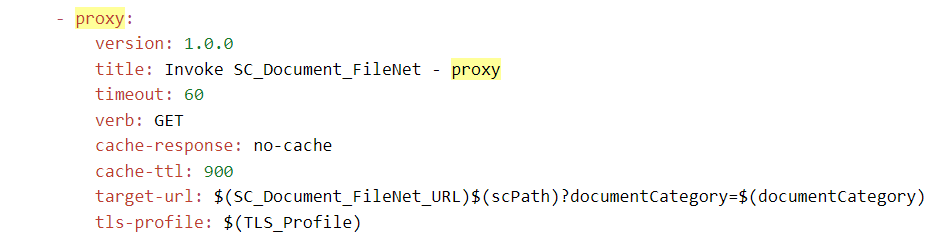


To

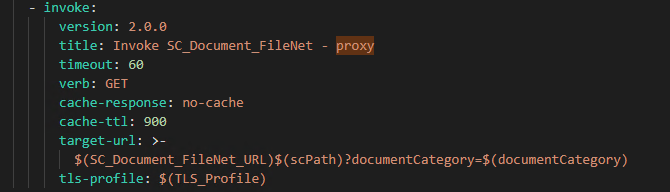


* Change proxy policy

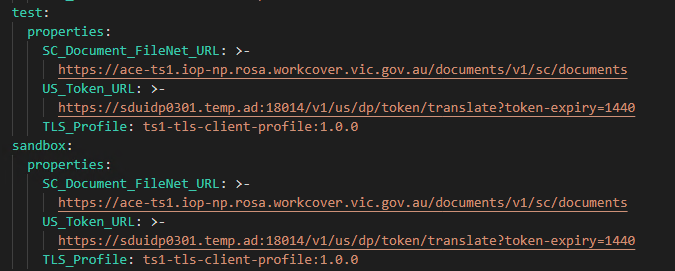
From



To

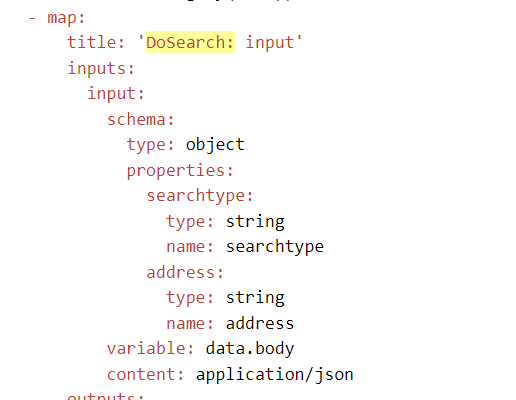


* Add Sandbox and Test catalog properties

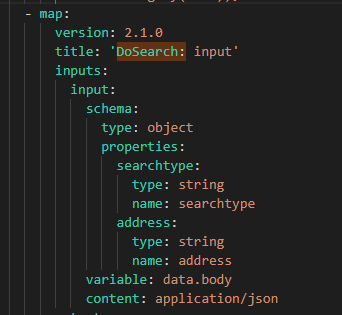


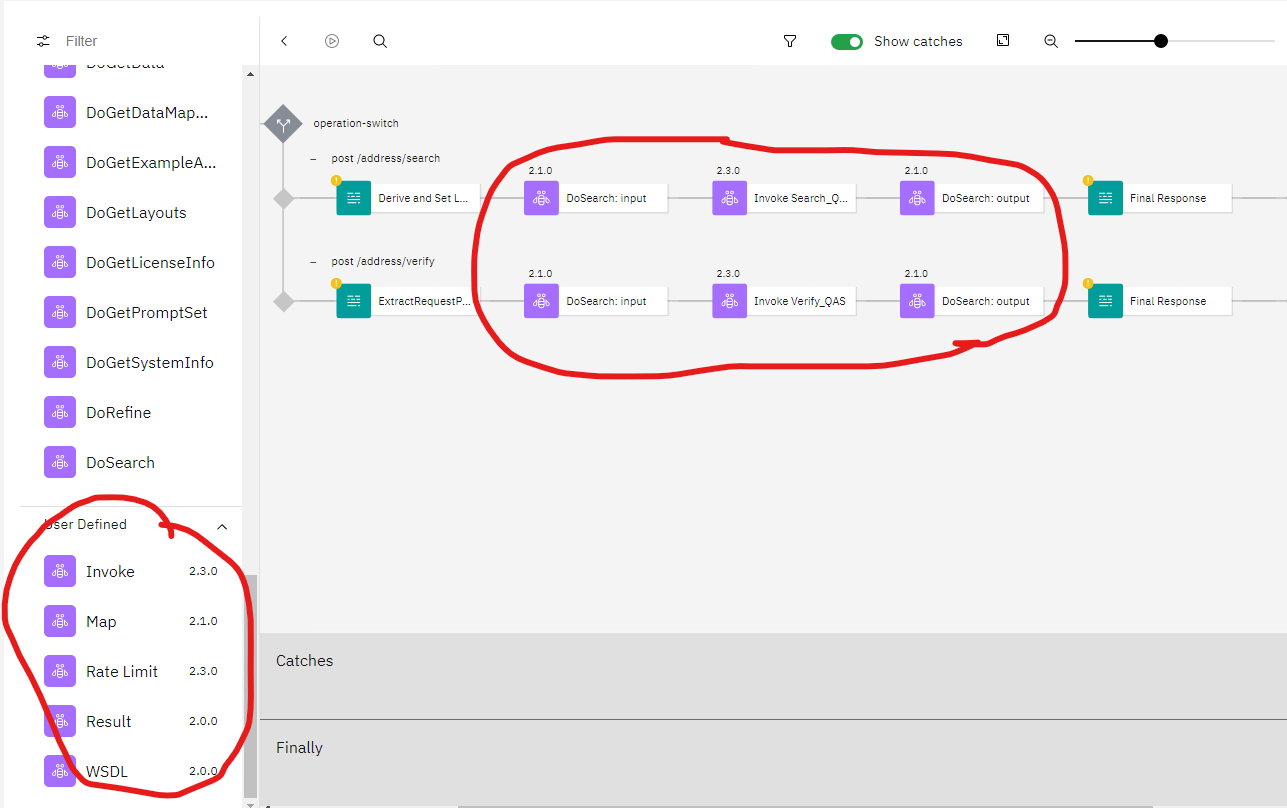
* Set version number for User Defined Policy

From

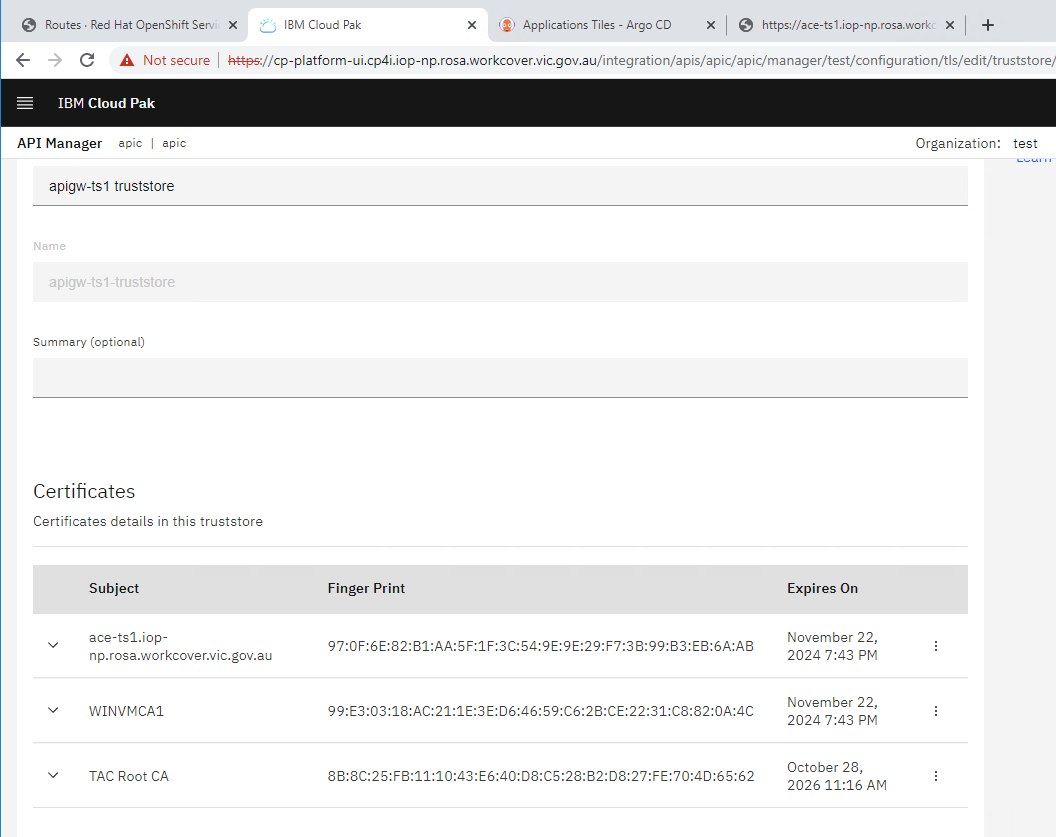


To

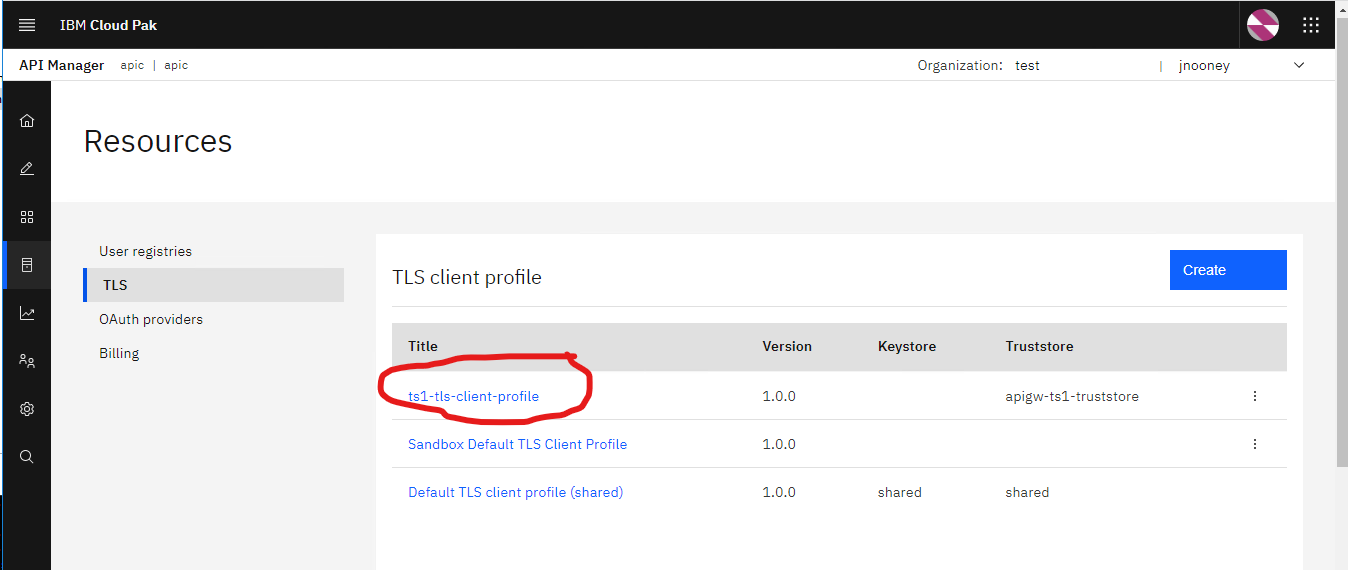




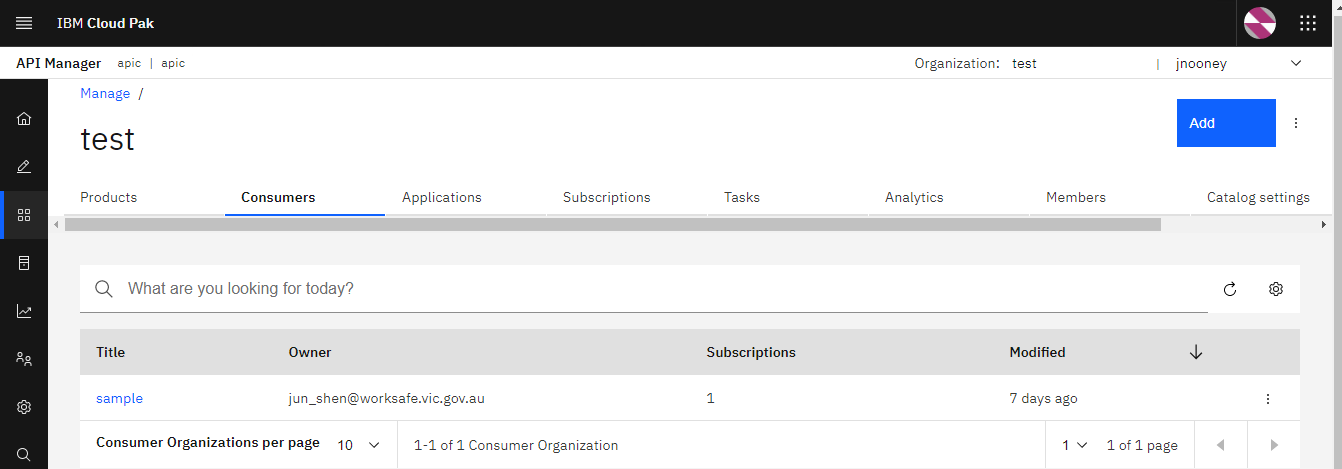
* 1. **Create a truststore**



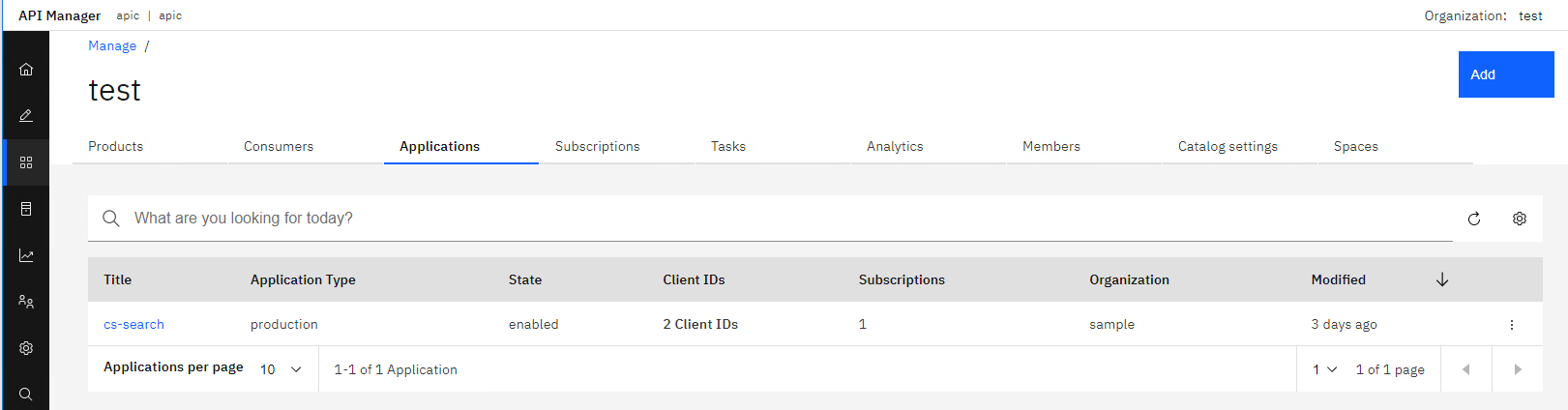
* 1. **Create a TLS client profile**



* 1. **Create a consumer org**



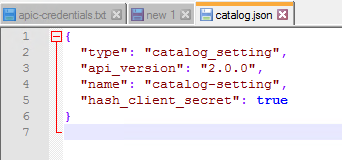
* 1. **Create an application**



* 1. **Import subscriptions**
* Login API Manager

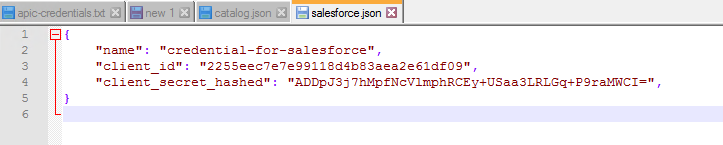
apic login --username jnooney --password Aditya@369 --server https://cp-platform-ui.cp4i.iop-np.rosa.workcover.vic.gov.au/integration/apis/apic/apic --realm provider/default-idp-2

* Enable “hash\_client\_secret” for Catalog



apic catalog-settings:update --server https://cp-platform-ui.cp4i.iop-np.rosa.workcover.vic.gov.au/integration/apis/apic/apic --org test --catalog test catalog.json

* Import subscriptions



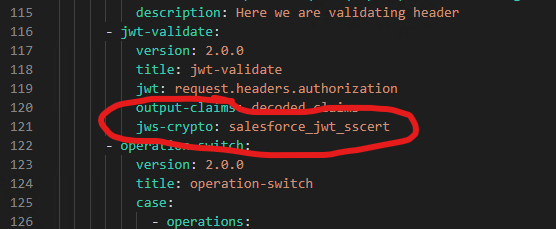
apic credentials:create --server https://cp-platform-ui.cp4i.iop-np.rosa.workcover.vic.gov.au/integration/apis/apic/apic --org test --catalog test --consumer-org sample --app cs\_search salesforce.json --format json --output -

* List subscriptions

apic credentials:list --server https://cp-platform-ui.cp4i.iop-np.rosa.workcover.vic.gov.au/integration/apis/apic/apic --org test --catalog test --consumer-org sample --app cs\_search

* 1. **Update APIGW**

If an API has jwt-validate policy, need deploy the jws-cryto to APIGW.



* First create a secret for the certificate.

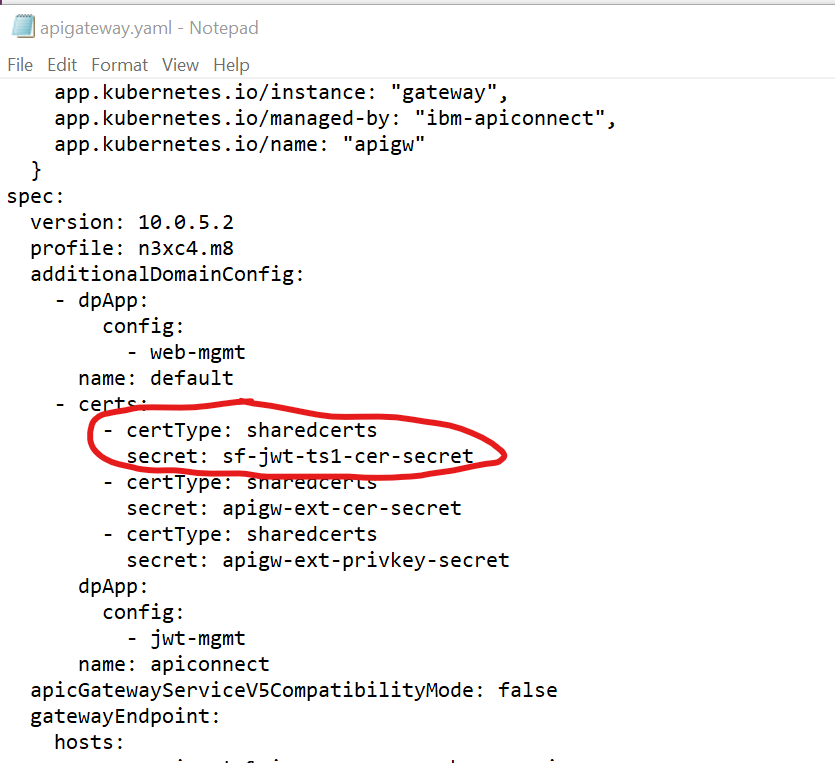
oc create secret generic sf-jwt-ts1-cer-secret --from-file=sf-jwt-ts1.cer -n apigw-ts1

* Second create a configmap for it.



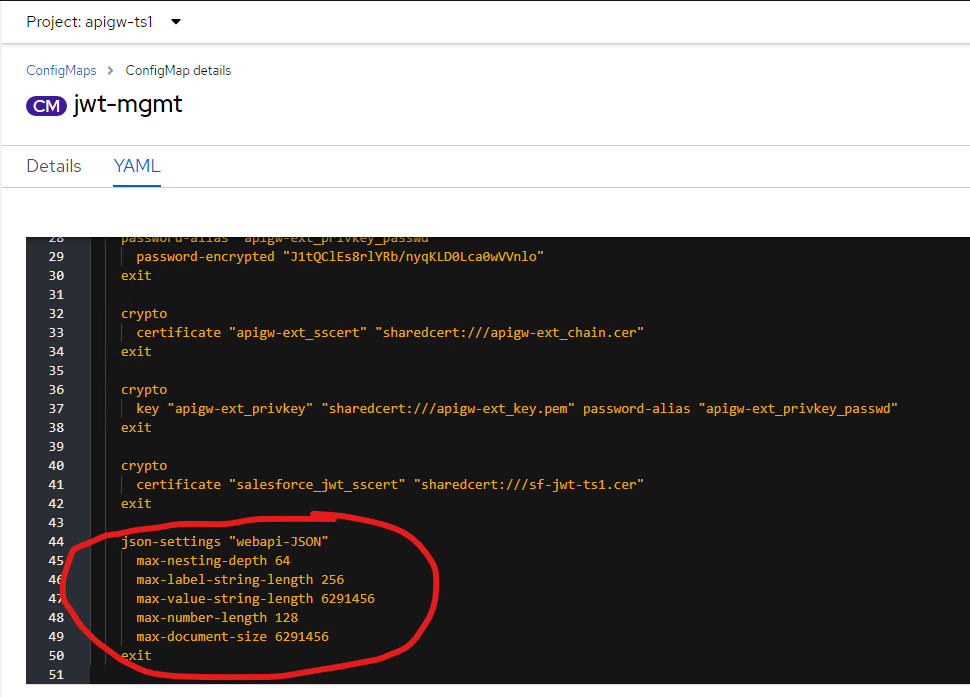
* Last update APIGW CR.

APIGW CR link: <https://console-openshift-console.apps.rosa-iop-np.b3hz.p1.openshiftapps.com/k8s/ns/apigw-ts1/clusterserviceversions/ibm-apiconnect.v3.2.1/gateway.apiconnect.ibm.com~v1beta1~GatewayCluster/apigw/yaml>

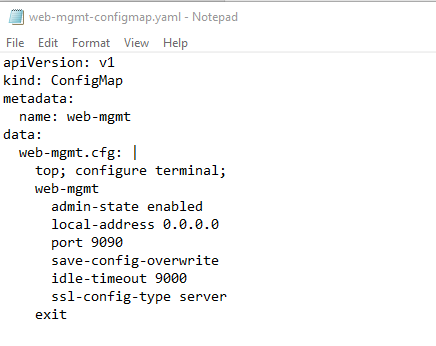


* 1. **Configure JSONSetting on APIGW**
* Add JSONSetting in APIGW jwt-mgmt configmap

The jwt-mgmt configmap link: <https://console-openshift-console.apps.rosa-iop-np.b3hz.p1.openshiftapps.com/k8s/ns/apigw-ts1/configmaps/jwt-mgmt/yaml>

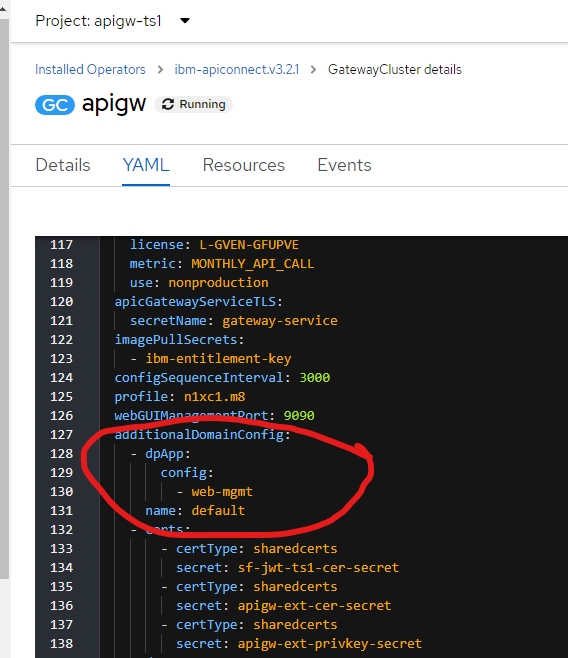


* 1. **Enable WebUI on APIGW**
* Create a web-mgmt ConfigMap

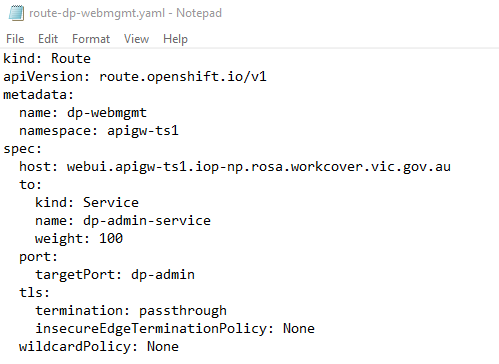


* Update APIGW CR.

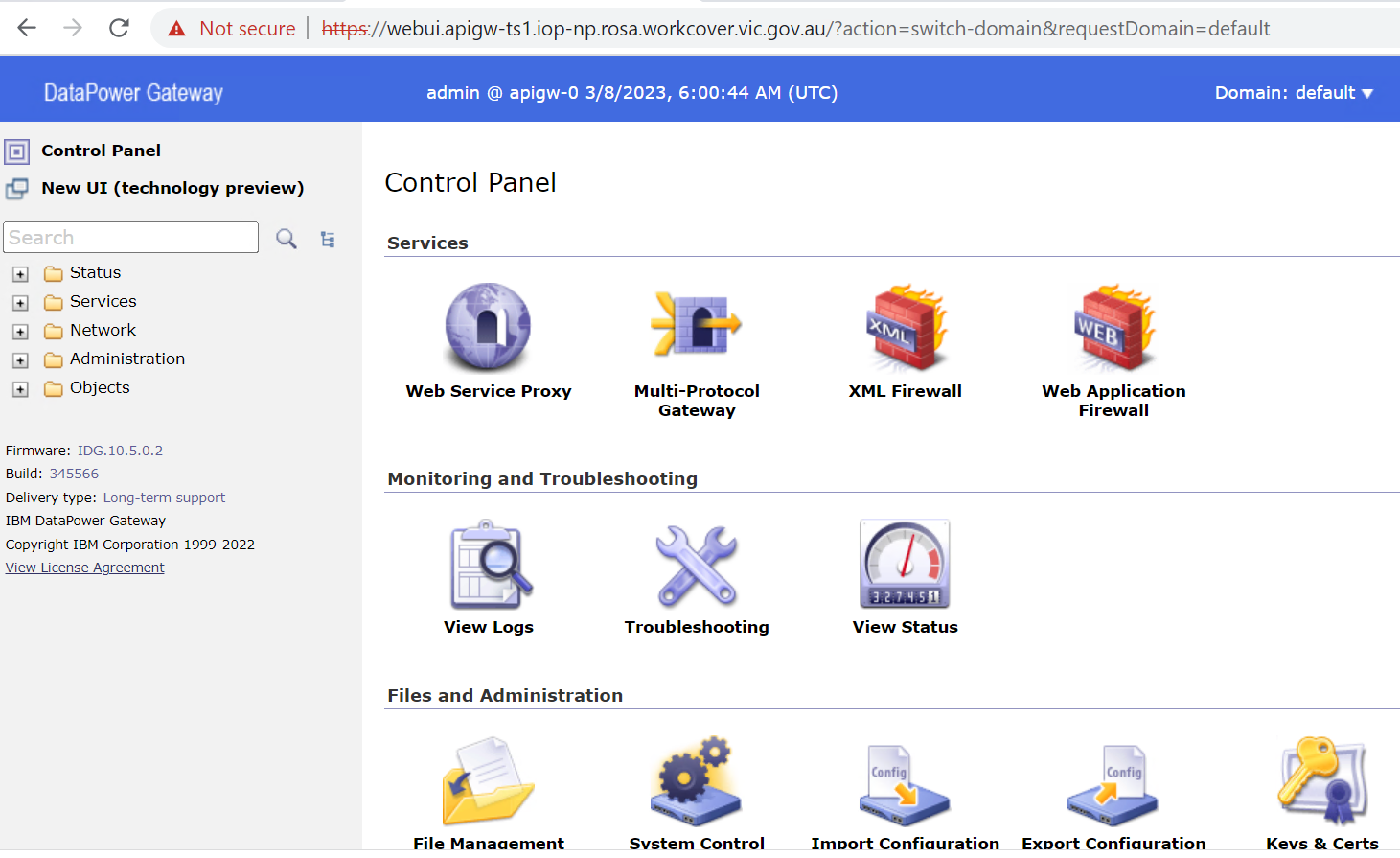
APIGW CR link: <https://console-openshift-console.apps.rosa-iop-np.b3hz.p1.openshiftapps.com/k8s/ns/apigw-ts1/clusterserviceversions/ibm-apiconnect.v3.2.1/gateway.apiconnect.ibm.com~v1beta1~GatewayCluster/apigw/yaml>



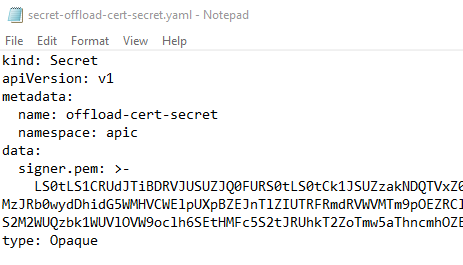
* Create a route for WebUI.



* Open WebUI via <https://webui.apigw-ts1.iop-np.rosa.workcover.vic.gov.au>

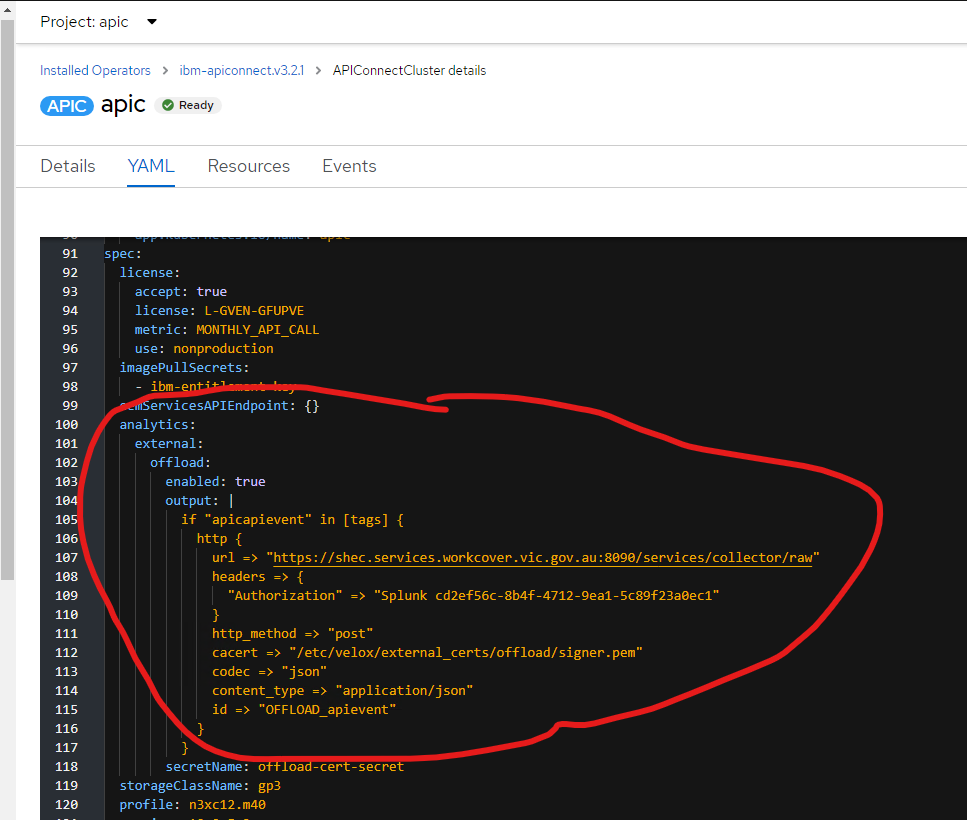


* 1. **Enable offload analytics to Splunk**
* Create a secret for offload certificate



* Update APIConnectCluster CR

APIConnectCluster CR link: <https://console-openshift-console.apps.rosa-iop-np.b3hz.p1.openshiftapps.com/k8s/ns/apic/clusterserviceversions/ibm-apiconnect.v3.2.1/apiconnect.ibm.com~v1beta1~APIConnectCluster/apic/yaml>



# **ACE Migration**

Deploy ACE integration server via flying mode without building ACE image.

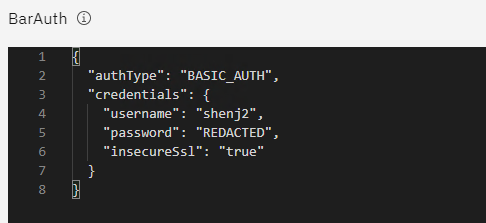
* 1. **Clone APIC artefacts from Bitbucket**

git clone <https://bitbucket-aws.itss.vic.gov.au/scm/rosa/rosa-app-migration.git>

* 1. **Create BarAuth configuration**

Reference link: <https://www.ibm.com/docs/en/app-connect/container?topic=types-barauth-type>

* Define the JSON content



* Create a secret

oc create secret generic barauth-55dd8 --from-file=configuration=/mytempdir/barauth.json --namespace=iop-ts1

* Create BarAuth yaml file

apiVersion: appconnect.ibm.com/v1beta1

kind: Configuration

metadata:

  name: barauth

  namespace: iop-ts1

spec:

  secretName: barauth-55dd8

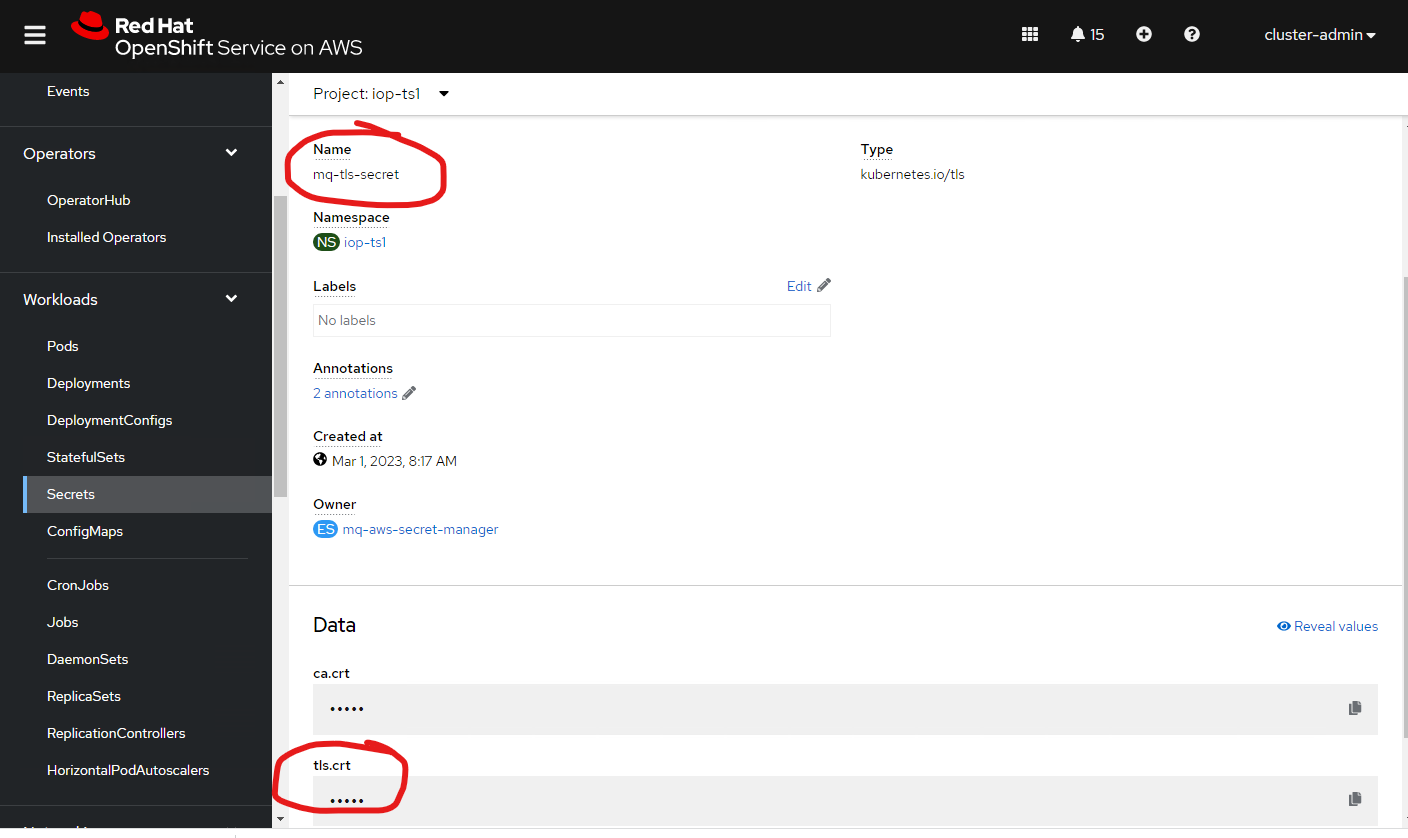
  type: barauth

  version: 12.0.7.0-r2

* Apply the yaml file

oc apply -f barauth.yaml

* 1. **Create mqkey configuration**
* Save MQ certificate as a file: TS1TM01.cer



* Import TS1TM01.cer to mqkey.kdb

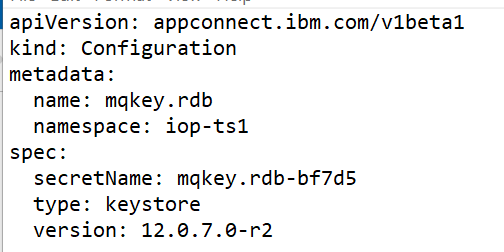
Run the command on a MQ server.

runmqakm -cert -add -db mqkey.kdb -label mqservercert -file TS1TM01.cer -format ascii –stashed

* Create a secret

oc create secret generic mqkey.kdb-7sr6b --from-file=configuration=mqkey.kdb --namespace=iop-ts1

* Create mqkey.kdb yaml file and apply the file via oc



* 1. **Create truststore.jks configuration**
* Create truststore.jks

keytool -import -file ace-ts1-tls.crt -alias aceroot2 -keystore truststore.jks -storepass changeit

keytool -import -file truststore-tacroot.crt -alias tacroot -keystore truststore.jks -storepass changeit

keytool -import -file truststore-wsvroot.crt -alias wsvroot -keystore truststore.jks -storepass changeit

keytool -import -file truststore-filenet.crt -alias filenet -keystore truststore.jks -storepass changeit

keytool -import -file truststore-mdm.crt -alias mdm -keystore truststore.jks -storepass changeit

keytool -import -file truststore-mdmroot.crt -alias mdmroot -keystore truststore.jks -storepass changeit

keytool -import -file apigw-key.crt -alias apigwts1 -keystore truststore.jks -storepass changeit

* Create configuration via the same approach as mqkey
  1. **Create keystore.jks configuration**
* Create keystore.jks

openssl pkcs12 -export -in ace-ts1-tls.crt -inkey ace-ts1-tls.key -certfile ace-ts1-ca.crt -out ace.p12 -passout pass:changeit -name acekey

keytool -importkeystore -srckeystore ace.p12 -srcstoretype PKCS12 -srcstorepass changeit -destkeystore keystore.jks  -deststoretype JKS -deststorepass changeit

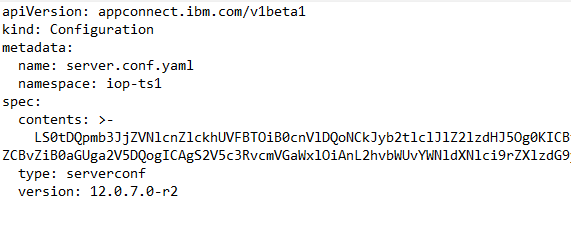
* Create a configuration via the same approach as mqkey
  1. **Create server.conf.yaml configurations**
* Update the server.conf.yaml

1. mqKeyRepository path
2. truststoreFile path
3. jvmSystemProperty path
4. KeystoreFile path

* Convert the file content to base64

cat server.conf.yaml | base64

* Create a yaml file and apply it



* 1. **Create odbc.ini configurations**
* Update odbc.ini

1. KeyStore=/home/aceuser/generic/dbkey.p12
2. TrustStore=/home/aceuser/generic/dbkey.p12

* Convert the file content to base64

cat odbc.ini | base64

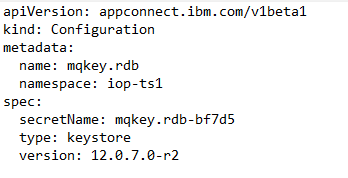
* Create a yaml file and apply it



* 1. **Create mqkey.rdb configurations**
* Create a secret

oc create secret generic mqkey.rdb-bf7d5 --from-file=configuration=mqkey.rdb --namespace=iop-ts1

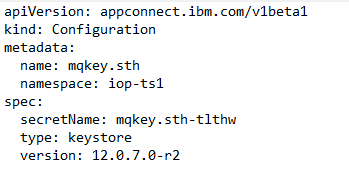
* Create mqkey.rdb configuration yaml file and apply the file via oc



* 1. **Create mqkey.sth configurations**
* Create a secret

oc create secret generic mqkey.sth-tlthw --from-file=configuration=mqkey.sth --namespace=iop-ts1

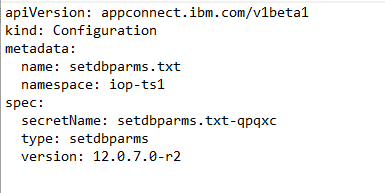
* Create mqkey.sth configuration yaml file and apply the file via oc



* 1. **Create setdbparms.txt configuration**
* Create a secret

oc create secret generic setdbparms.txt-qpqxc --from-file=configuration= setdbparms.txt --namespace=iop-ts1

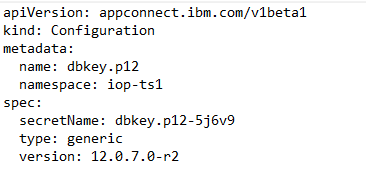
* Create setdbparms.txt configuration yaml file and apply the file via oc



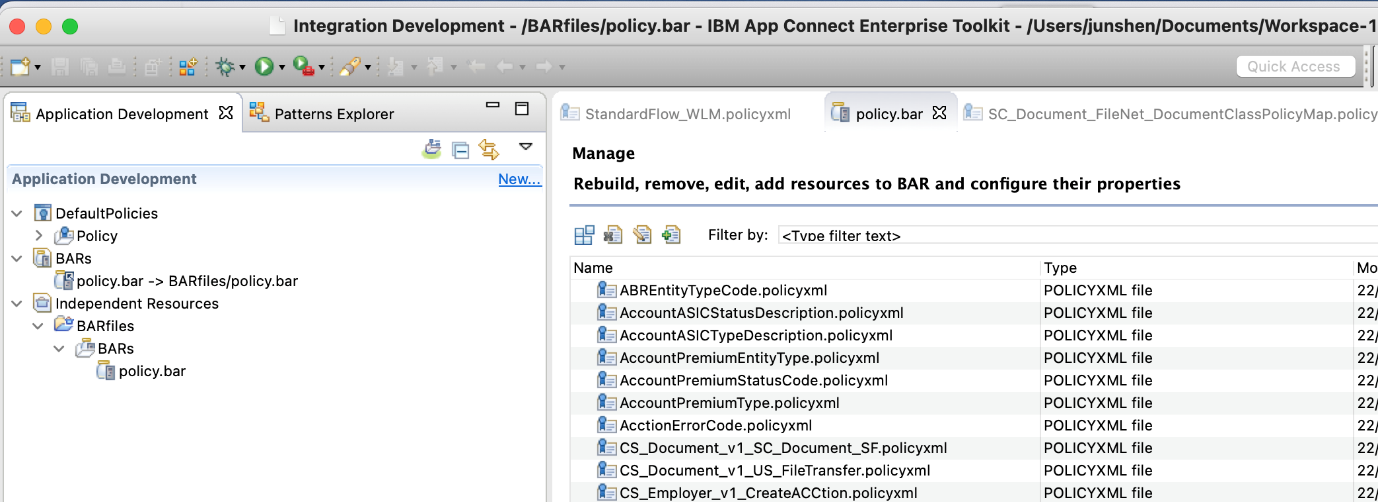
* 1. **Create dbkey.p12 configuration**
* Create a secret

oc create secret generic dbkey.p12-5j6v9 --from-file=configuration=dbkey.p12 --namespace=iop-ts1

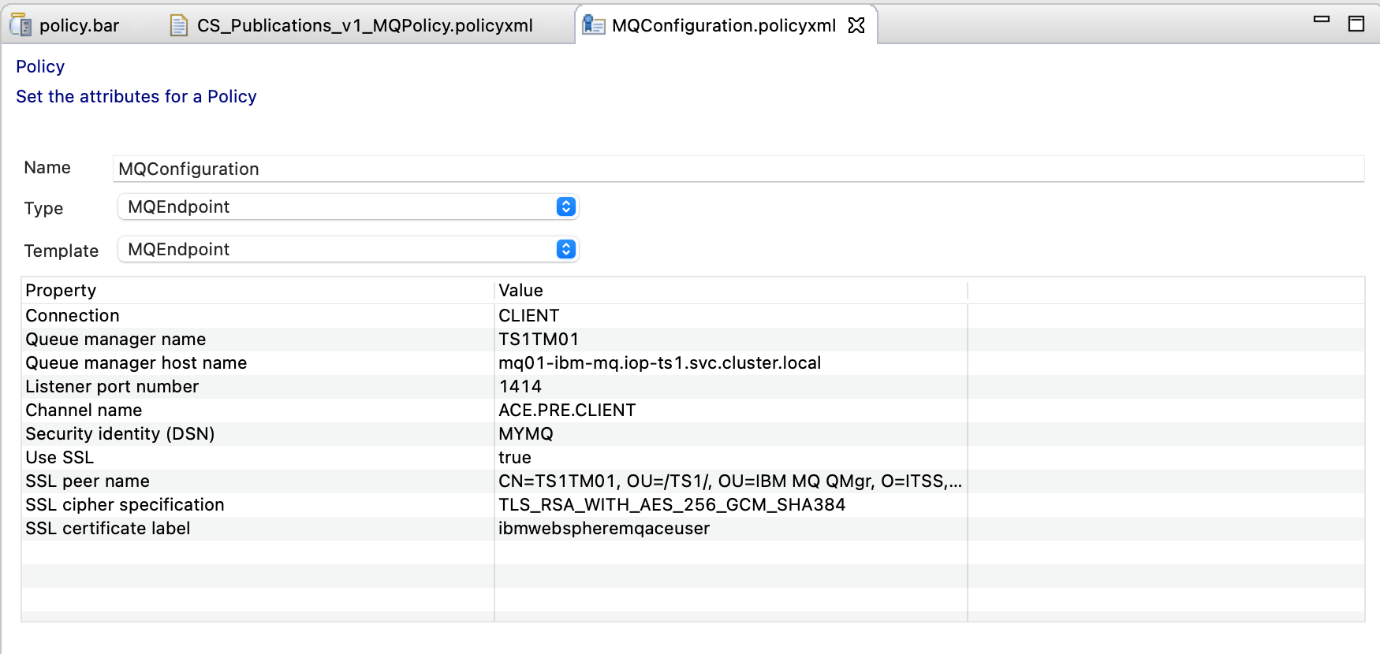
* Create dbkey.p12 configuration yaml file and apply the file via oc



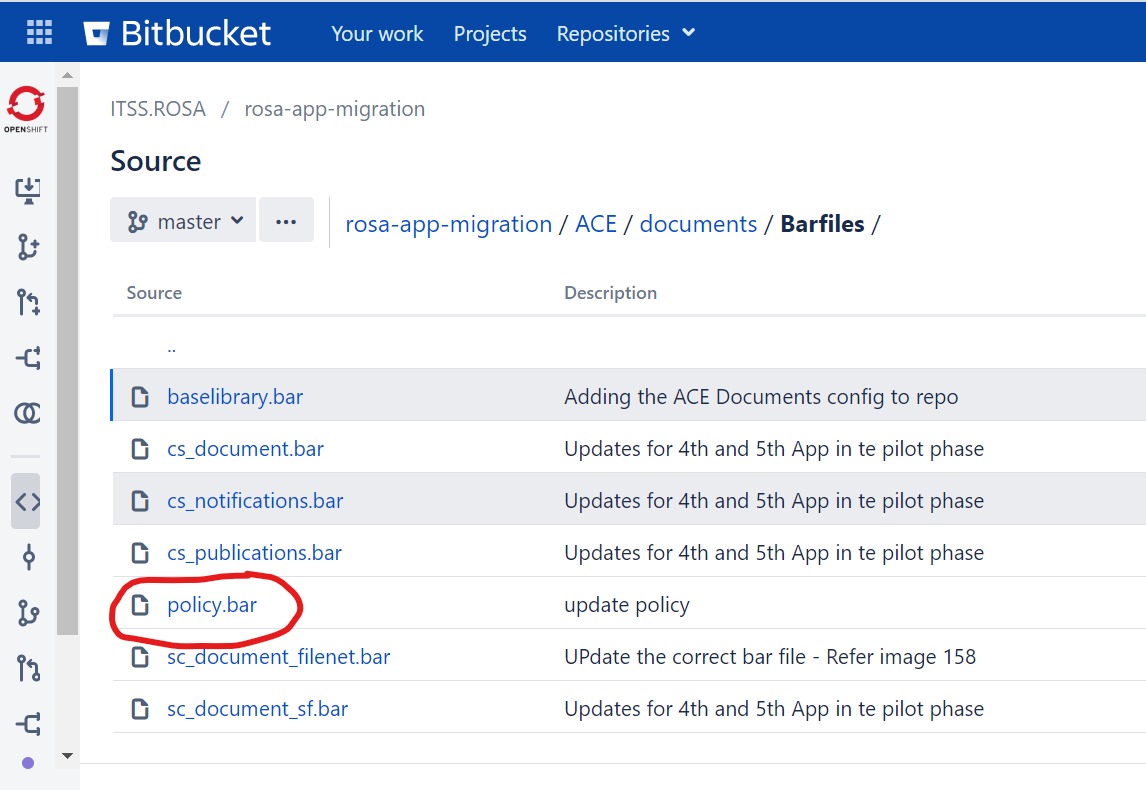
* 1. **Create a policy project in ACE toolkit**



* 1. **Update MQ policy**
* Queue manager name
* Host name
* SSL peer name
* SSL certificate label



* 1. **Export the policy bar and upload to rosa-app-migration Bitbucket**



* 1. **Create Integration Server yaml file**

apiVersion: appconnect.ibm.com/v1beta1

kind: IntegrationServer

metadata:

  name: is-cs-search

  namespace: iop-ts1

spec:

  enableMetrics: true

  license:

    accept: true

    license: L-APEH-CJUCNR

    use: CloudPakForIntegrationProduction

  pod:

    containers:

      runtime:

        resources:

          limits:

            cpu: 300m

            memory: 368Mi

          requests:

            cpu: 300m

            memory: 368Mi

  adminServerSecure: true

  router:

    timeout: 120s

  designerFlowsOperationMode: disabled

  createDashboardUsers: true

  service:

    endpointType: https

  version: 12.0.7.0-r2

  replicas: 1

  barURL: >-

    https://bitbucket-aws.itss.vic.gov.au/projects/ROSA/repos/rosa-app-migration/raw/ACE/search/Bar%20Files/baselibrary.bar,https://bitbucket-aws.itss.vic.gov.au/projects/ROSA/repos/rosa-app-migration/raw/ACE/search/Bar%20Files/sc\_search\_rds.bar

  configurations:

    - barauth

    - dbkey.p12

    - iop-log4j2.xml

    - keystore.jks

    - odbc.ini

    - setdbparms.txt

    - truststore.jks

    - server.conf.yaml

    - mqkey.kdb

    - mqkey.rdb

    - mqkey.sth

* Apply the yaml file

oc apply -f is-cs-search.yaml

* 1. **Create a route**

kind: Route

apiVersion: route.openshift.io/v1

metadata:

  name: cs-search-https

  namespace: iop-ts1

  annotations:

    haproxy.router.openshift.io/rewrite-target: /

spec:

  host: ace-ts1.iop-np.rosa.workcover.vic.gov.au

  path: /search

  to:

    kind: Service

    name: is-cs-search-is

    weight: 100

  port:

    targetPort: https

  tls:

    termination: reencrypt

    certificate: |-

      -----BEGIN CERTIFICATE-----

      -----END CERTIFICATE-----

    key: |-

      -----BEGIN PRIVATE KEY-----

      -----END PRIVATE KEY-----

    caCertificate: |-

      -----BEGIN CERTIFICATE-----

      -----END CERTIFICATE-----

    destinationCACertificate: |-

      -----BEGIN CERTIFICATE-----

      -----END CERTIFICATE-----

    insecureEdgeTerminationPolicy: Redirect

  wildcardPolicy: None

1. Certificate is the certificate of the ACE server
2. Key is the key of the ACE server
3. caCertificate is the CA certificate to sign the certificate of the ACE server, which is WINVMCA1 and TACROOT
4. destinationCACertificate is also the certificate of the ACE server

* Apply the yaml file

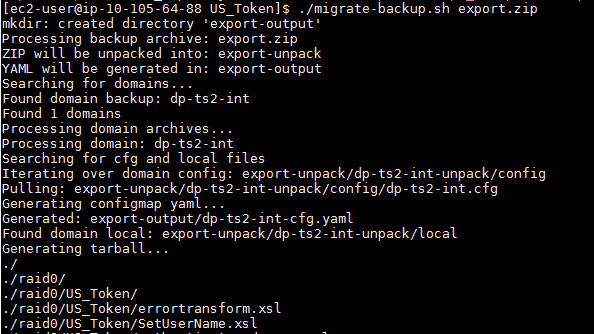
oc apply -f cs-search-route.yaml

# **DataPower Migration**

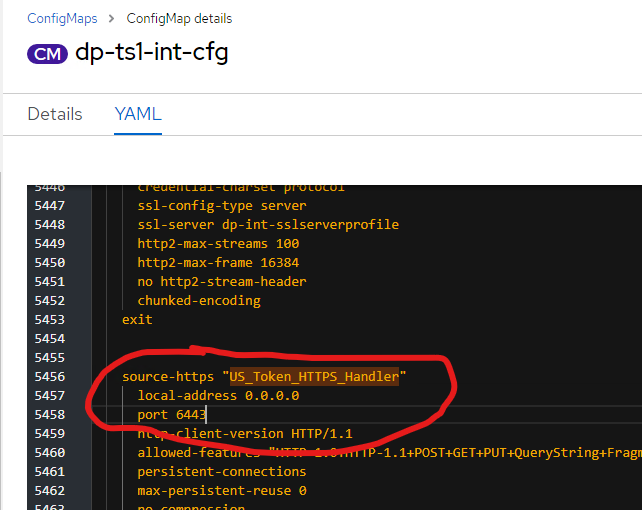
* Export DP domain
* Download the transfer tool

<https://github.com/IBM/datapower-operator-scripts#migrate-backupsh>

* Transfer the DP domain export file



* Update US\_Token\_HTTPS\_Handler local address and port



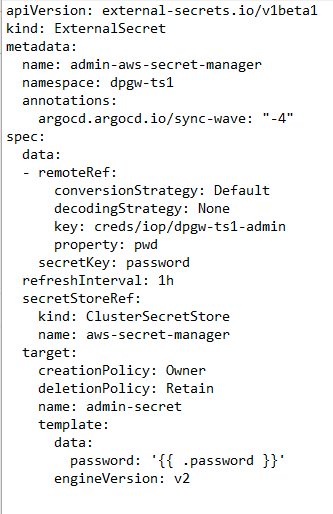
* Creating ConfigMaps generated by the above tool

cd backup-output

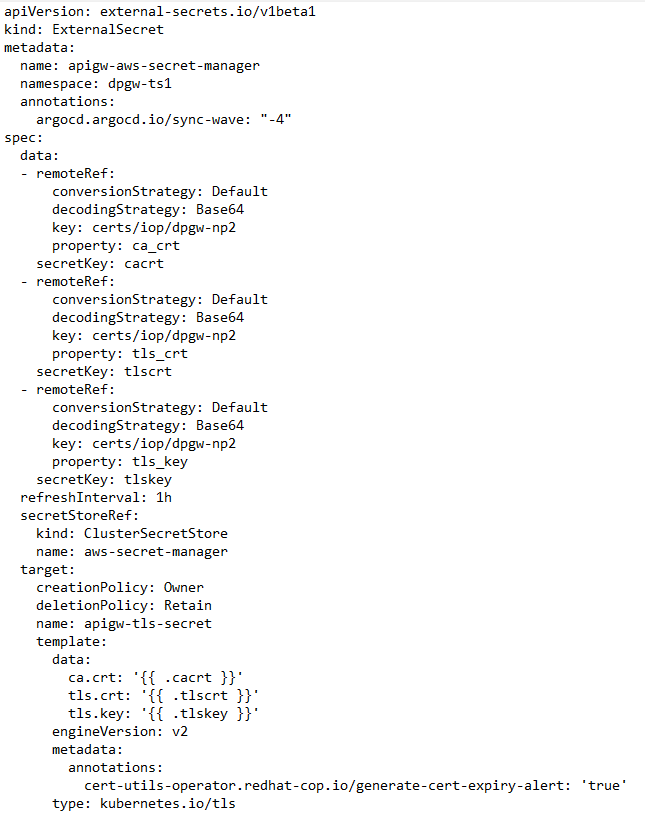
oc apply -f dp-ts1-int-cfg.yaml –n dpgw-ts1

oc apply -f dp-ts1-int-local.yaml –n dpgw-ts1

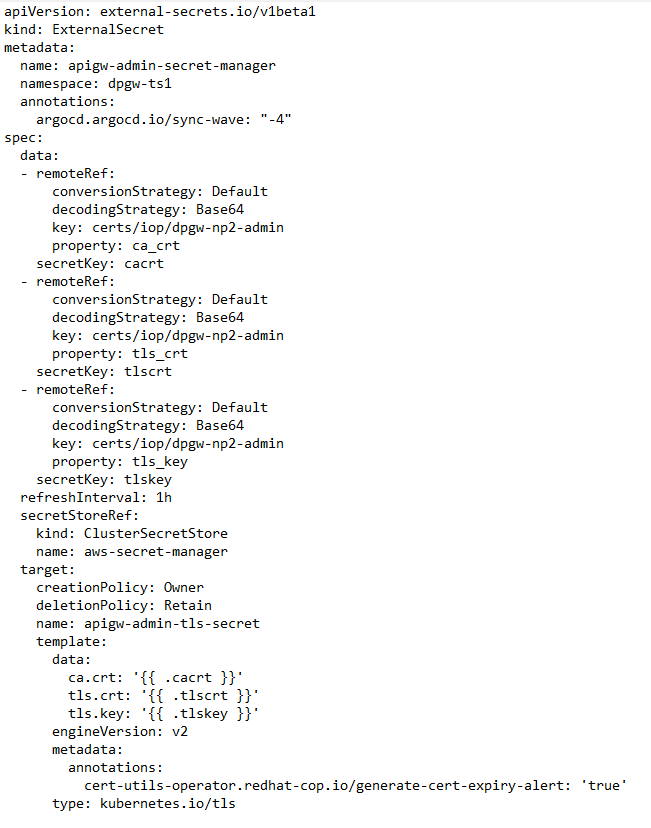
* Create the password secret



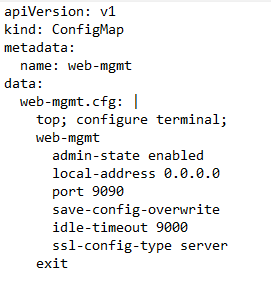
* Create a TLS key



* Create WebUI admin TLS key



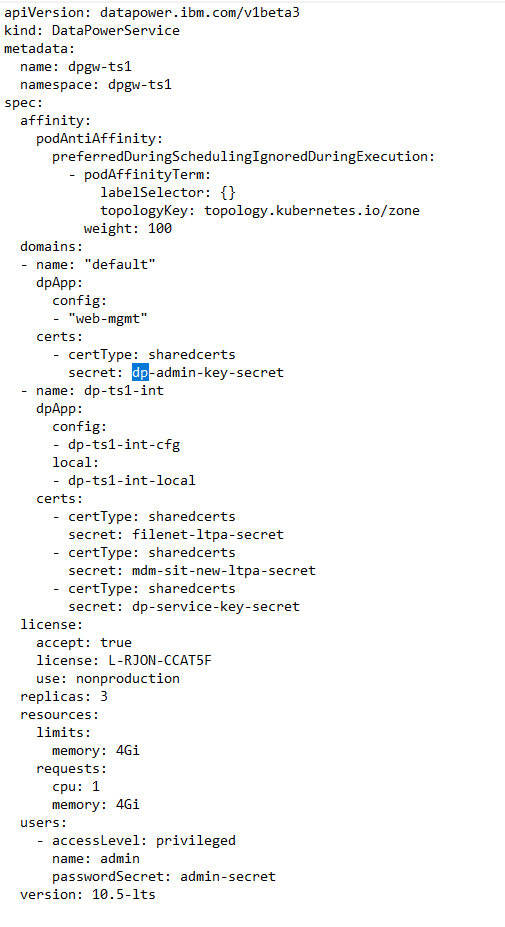
* Create a web management ConfigMap



* Create secrets for the certificates and keys which could not be exported from the old Datapower, eg, DataPower service certificate and key.

oc create secret generic dp-service-key-secret --from-file=apigw-np1\_key.pem --from-file=apigw-np1.cer

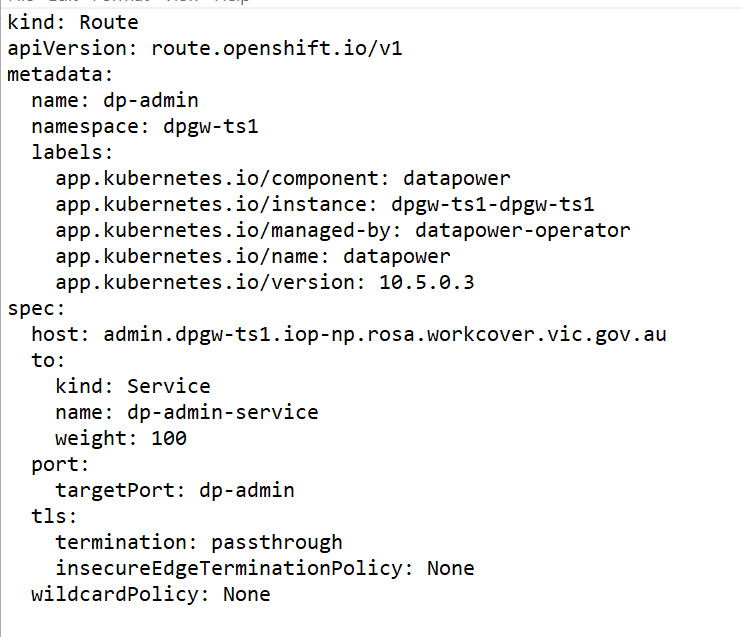
* Create DataPower Service CR



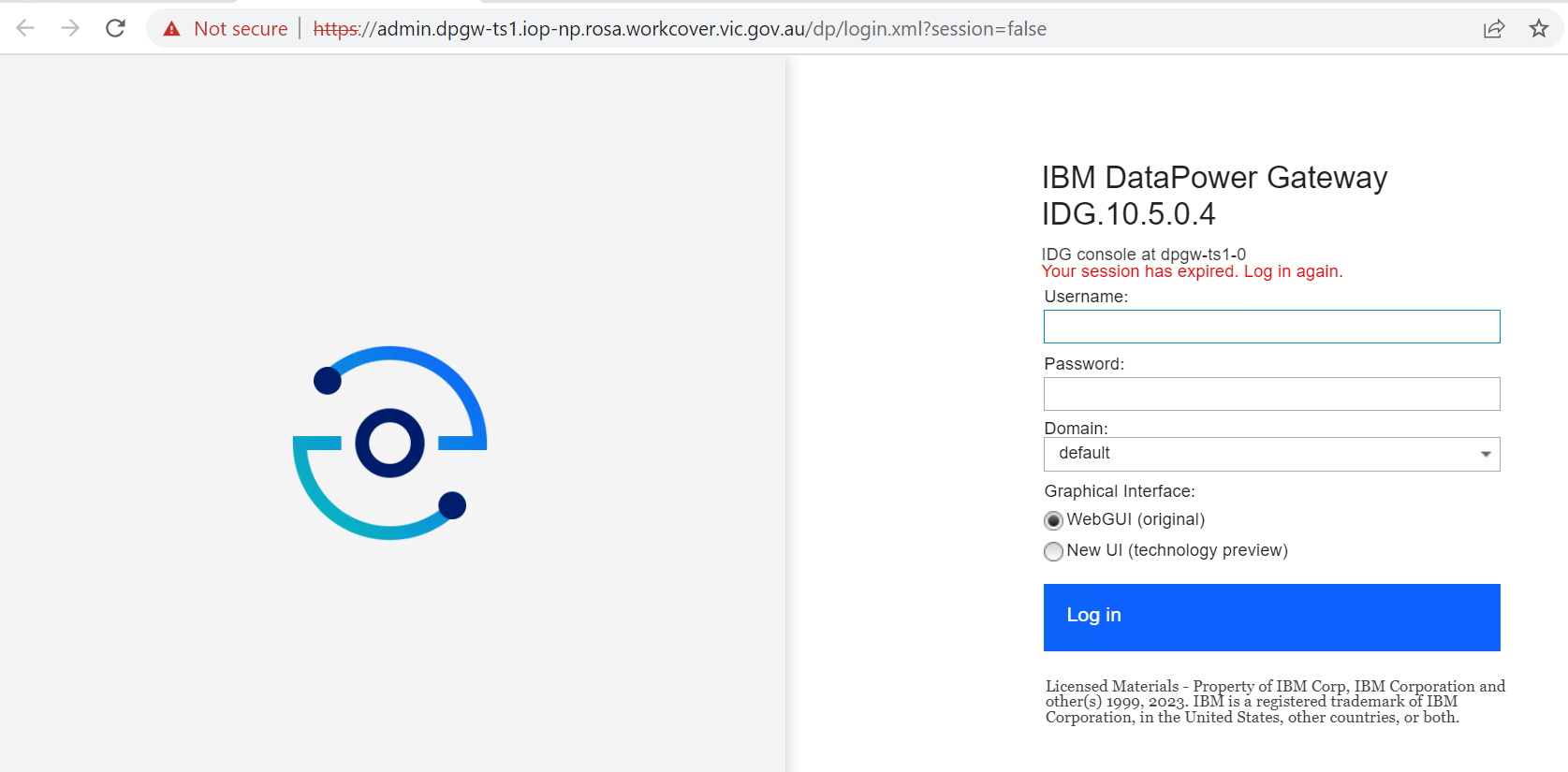
* Create a service yaml file



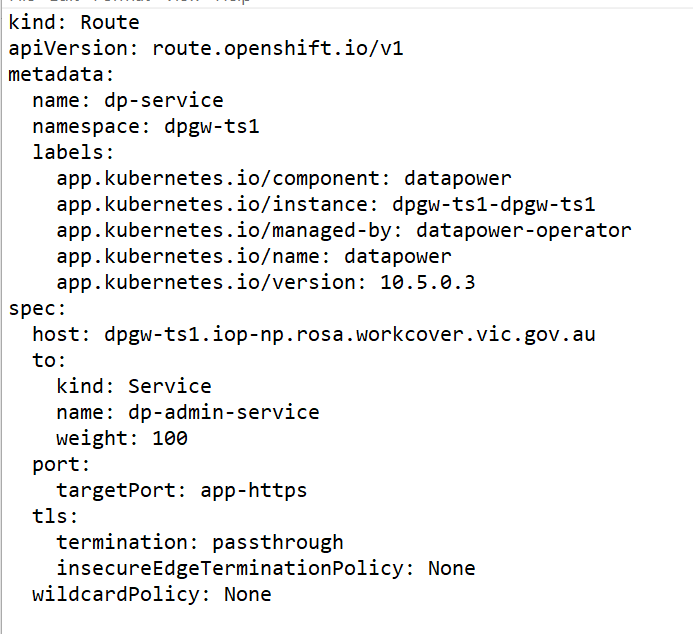
* Create a route for WebUI



* Open the WebUI via <https://admin.dpgw-ts1.iop-np.rosa.workcover.vic.gov.au>



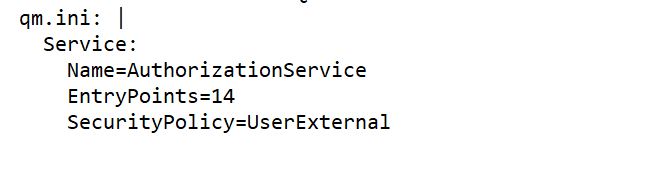
* Create a route for DP service



# **MQ Migration**

* Add a qm.ini file

The configuration is to enable MQ to support users which are not on the local OS or LDAP.



* Remove the listener definition
* Update Channel configurations

define channel('ACE.PRE.CLIENT') chltype(SVRCONN) trptype(TCP)

MCAUSER('app') sslcauth(REQUIRED) sslciph(TLS\_RSA\_WITH\_AES\_256\_GCM\_SHA384)

SSLPEER('CN=ACE-TS1 Client,OU=TS1/IBM MQ

Client,O=ITSS,L=Geelong,ST=VIC,C=AU') REPLACE

alter authinfo(SYSTEM.DEFAULT.AUTHINFO.IDPWOS) authtype(IDPWOS)

chckclnt(NONE)

* Configure User “app” permission

set authrec principal('app') objtype(qmgr) authadd(all)

set authrec profile('SC.\*\*') principal('app') objtype(queue) authadd(all)

set authrec profile('CS.\*\*') principal('app') objtype(queue) authadd(all)

set authrec profile('TEST.\*\*') principal('app') objtype(queue) authadd(all)

set authrec profile('TS1TM01.DLQ') principal('app') objtype(queue)

authadd(all)

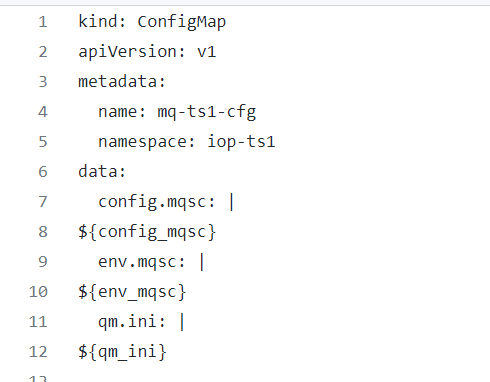
set authrec profile('SYSTEM.SALESFORCE.\*\*') principal('app') objtype(queue)

authadd(all)

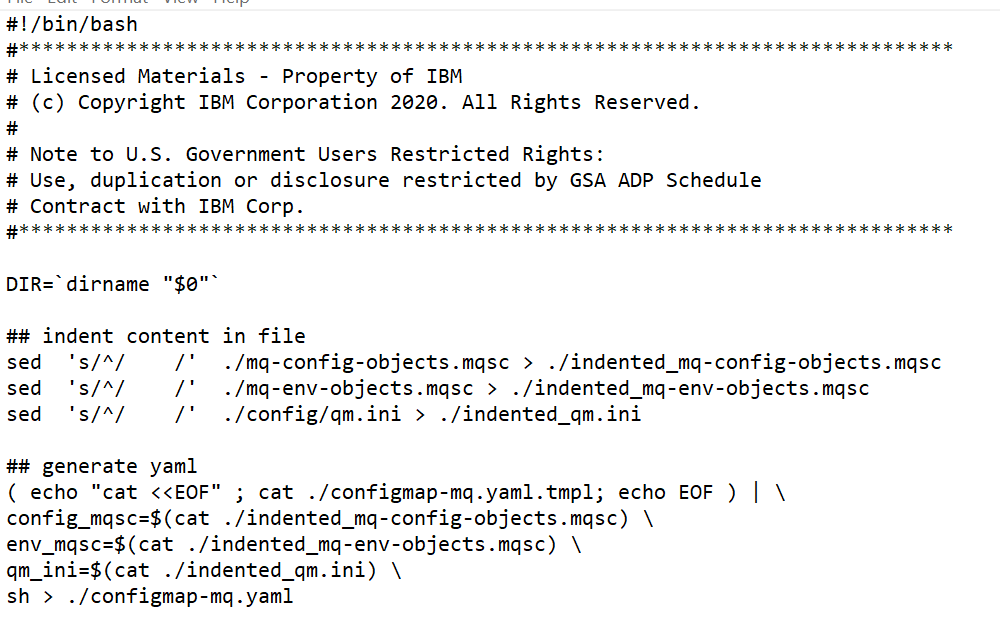
set authrec profile('SF.\*\*') principal('app') objtype(topic) authadd(all)

set authrec profile('IOP.\*\*') principal('app') objtype(topic) authadd(all)

* Create a ConfigMap template file: configmap-mq.yaml.tmpl



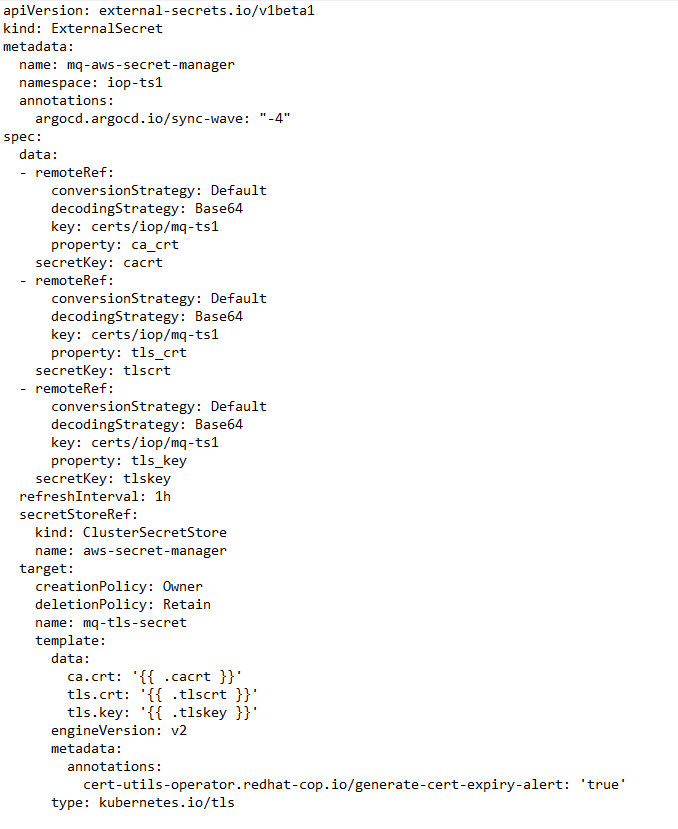
* Create a ConfigMap file via gen-mq.sh



* Apply the configMap file

oc apply –f configmap-mq.yaml

* Create a MQ TLS secret

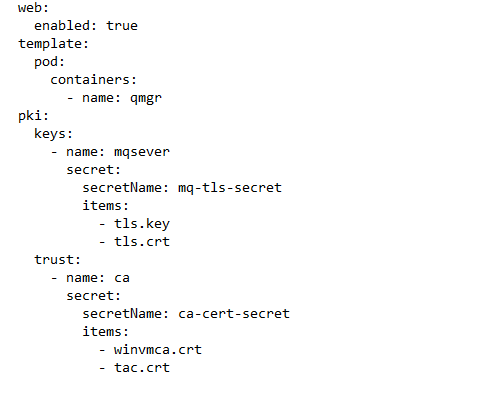


* Create a MQ ca secret

oc create secret generic ca-cert-secret --from-file=winvmca.crt --from-file=tac.crt –n iop-ts1

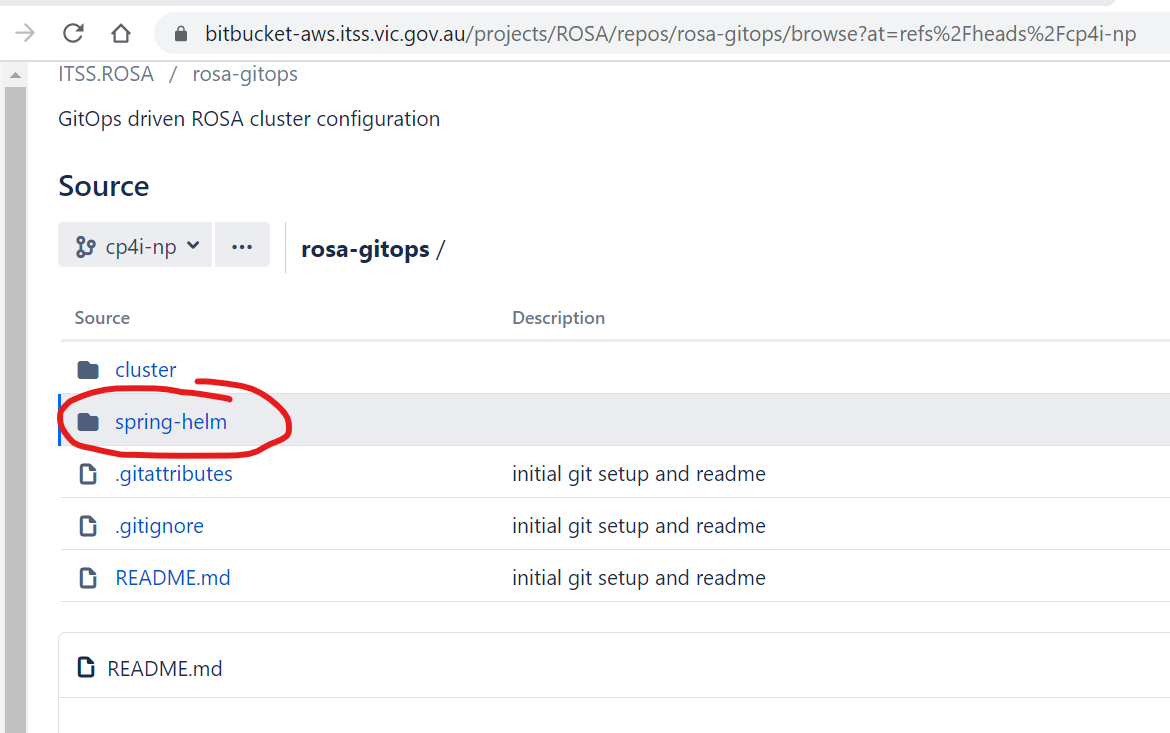
* Create a MQ CR





# **SpringBoot Migration**

* Upload Spring-helm directory to ITSS.ROSA/rosa-gitops bitbucket.



* Clone rosa-springboot on Bastion server

git clone <https://bitbucket-aws.itss.vic.gov.au/scm/rosa/rosa-springboot.git>

cd rosa-springboot/cicd-spring-01/scripts/csdecipha/ts1

* Create Configmaps and Secrets

chmod +x \*.sh

./generateAppConfigmap.sh jsb-dv1-csdecipha-config spring-ts1 /home/ec2-user/app-migration/rosa-app-migration/config-cs-decipha/config/ts1

./generateAppSecret.sh jsb-dv1-csdecipha-wsvsecret spring-ts1

./generatesslftpkeySecret.sh jsb-dv1-csdecipha-sslftpkeysecret spring-ts1 /home/ec2-user/app-migration/rosa-app-migration/config-cs-decipha/config/ts1

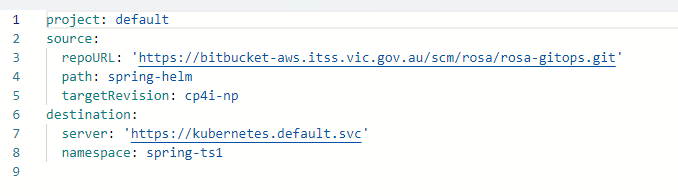
* Update Helm Chart Deployment Template

1. Change runAsUser: 1000
2. Remove fsGroup: 1001
3. Remove securityContext. Capabilities

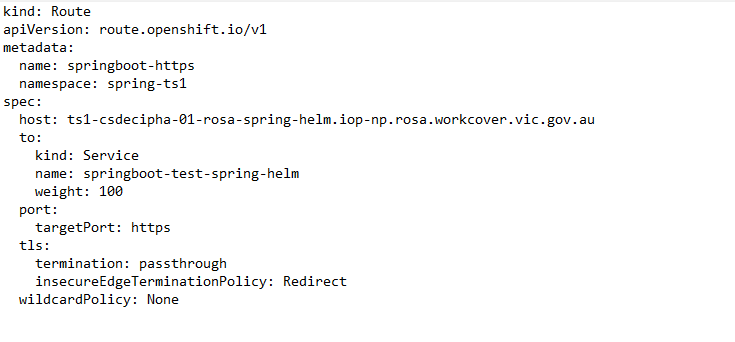
* Update Helm Chart Value.yaml

1. Update image. Repository Update image. Repository
2. Update image. Tag
3. Update springAppConfig, springAppSecret and sslftpkeySecret
4. Remove podAnnotations.prometheus.io/scrape
5. Remove nodeSelector.hostgroup and nodeSelector.ace

* Create an ArgoCD app and Sync the app.



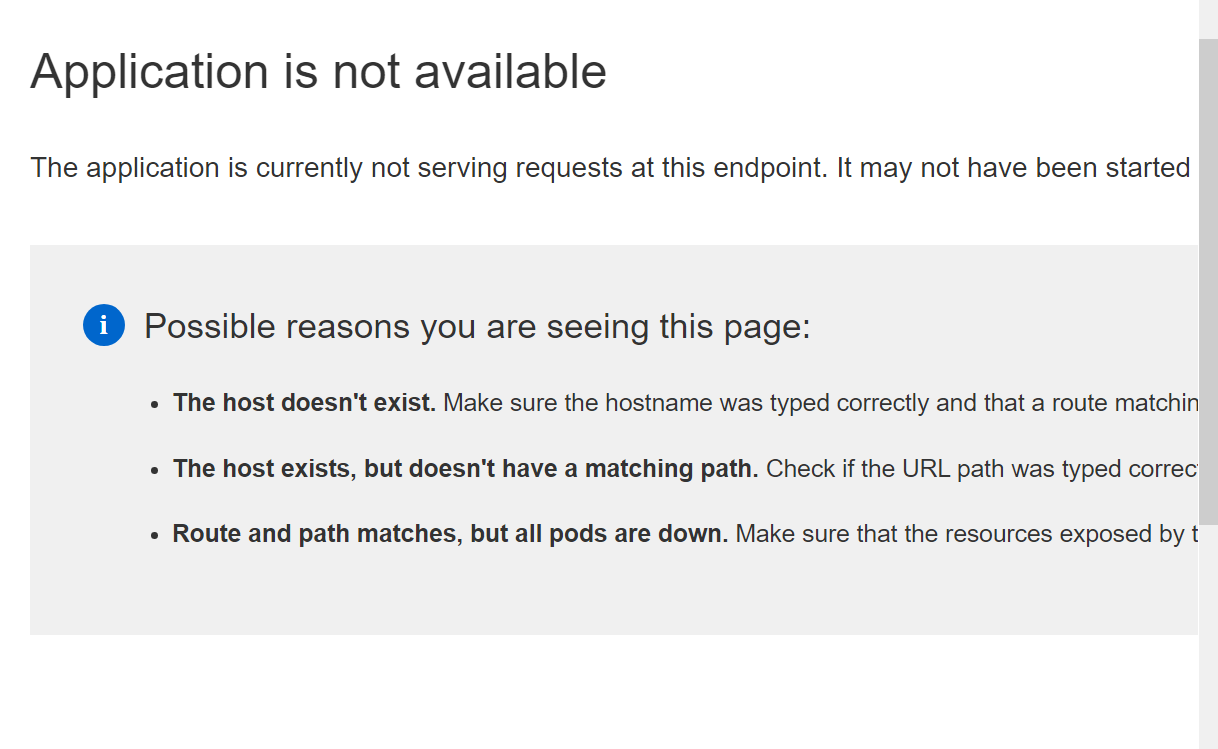
Create a route



# **ArgoCD deployment**

# **Troubleshooting**

* 1. **Platform UI is not available**

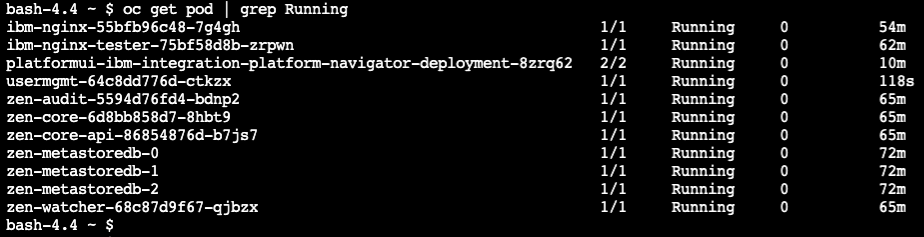


Solution: restart all platformUI pods under Namespace cp4i

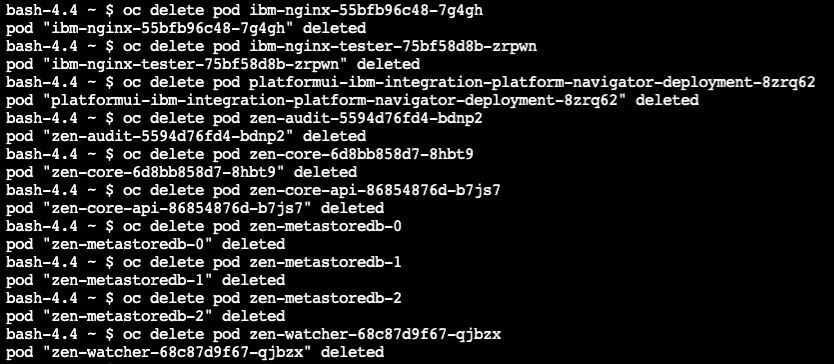
* Check the pods under Namespace cp4i

oc project cp4i

oc get pod | grep Running



* Delete pods

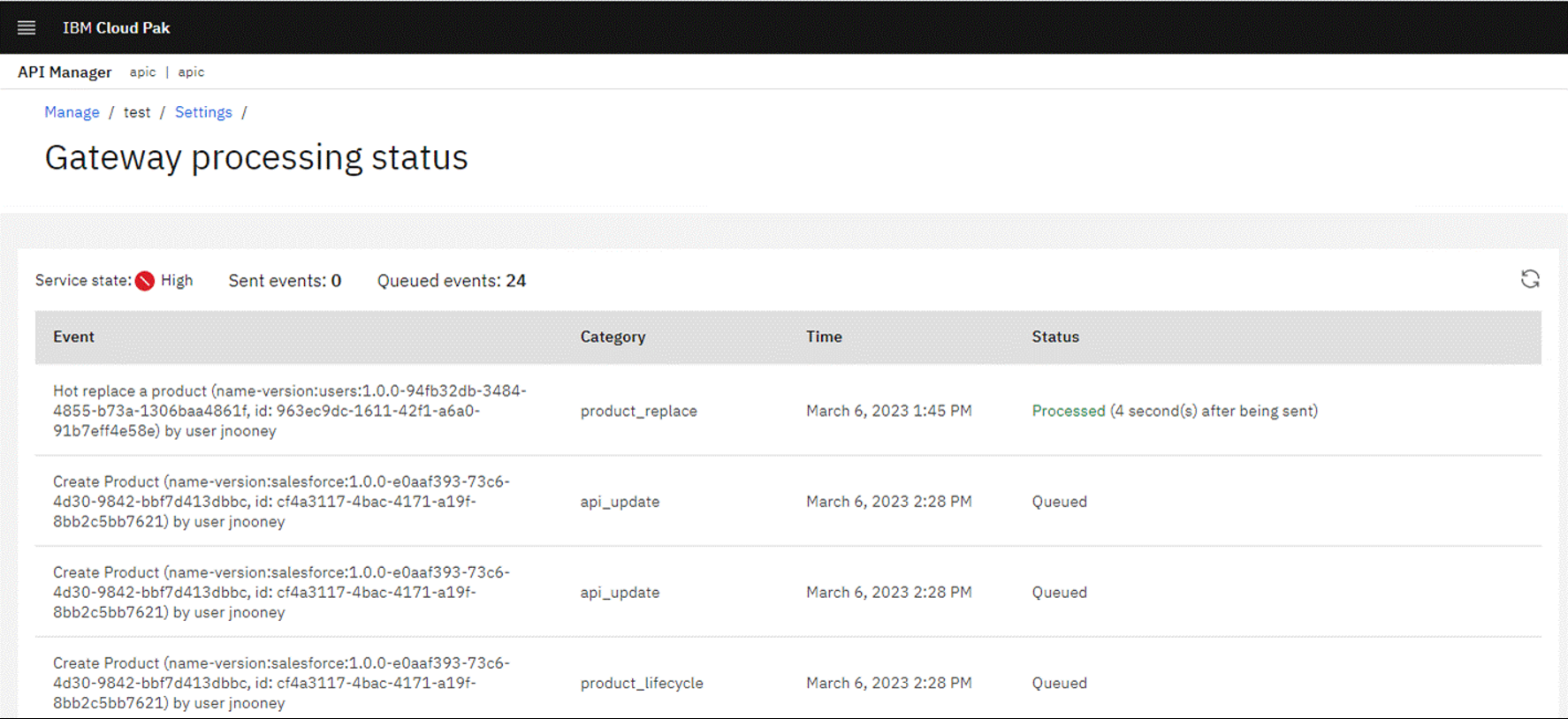


* Check all pods are ready

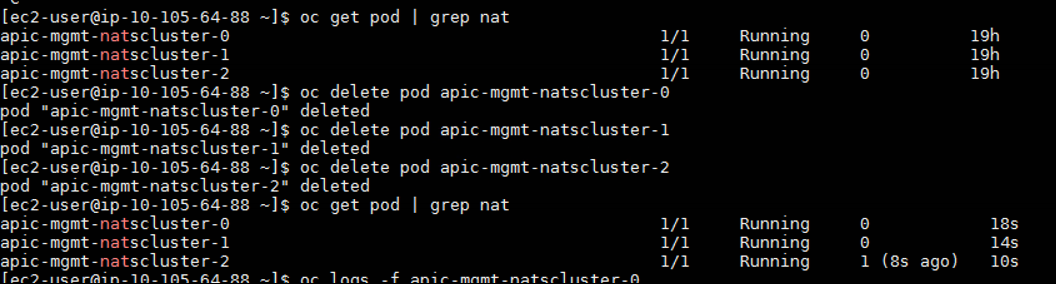
Text

Description automatically generated

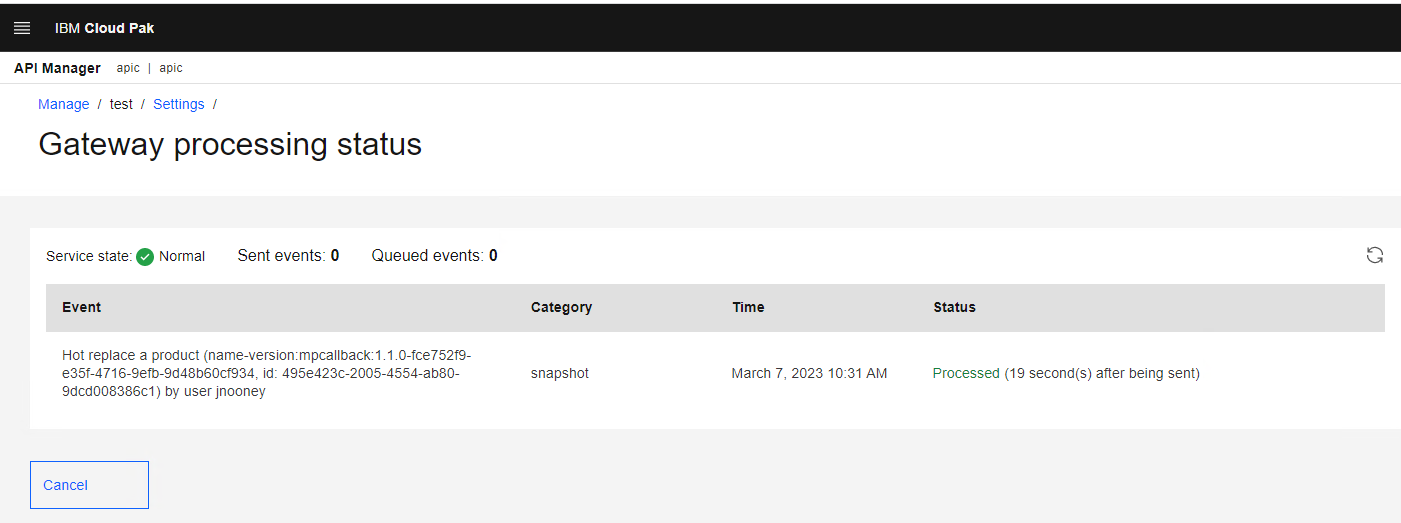
* 1. **API returns 404 Error**
* Check APIGW processing status and found a lot of events are queued.



* Solution: restart apic-mgmt-natscluster pods in apic namespace.



* Recheck APIGW processing status and make sure all events are processed.



* Publish APIs and retest the APIs.