How do I install my ODM Silver topology with CP4BA 21.0.3?

By NICOLAS PEULVAST

https://community.ibm.com/community/user/automation/blogs/nicolas-peulvast/2022/07/21/odm-silver-topology-2103

Target audience: ODM user with ODM Administrator role
Estimated duration: 240 minutes

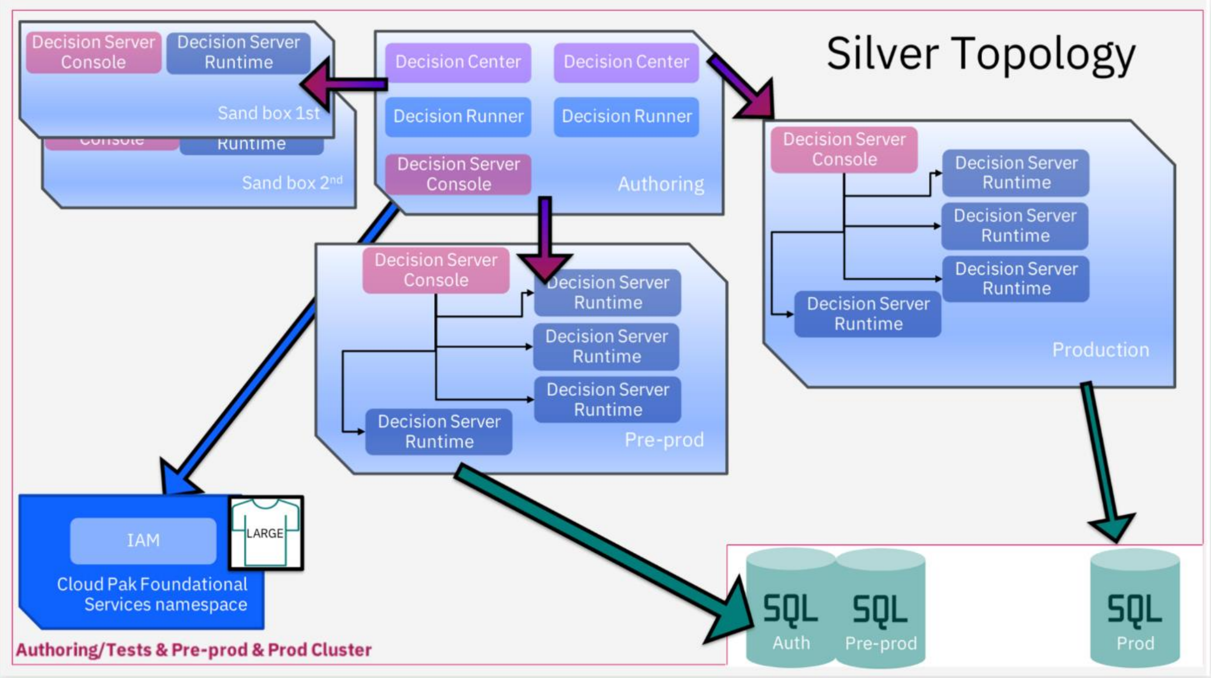
This article is part of an article series around Operational Decision Manager (ODM) topologies in context of Cloud Pak for Business Automation (CP4BA).
For more information about ODM environments and the topologies, see CP4BA ODM topologies on OpenShift.

Table of Contents

This document aims at describing how to make an ODM Silver topology deployment on OpenShift for Container Platform (OCP) as a component of CP4BA 21.0.3.

The ODM Silver topology is a deployment of several ODM environments, each in an individual namespace, within a single cluster.

Minimum Silver topology consists of an Authoring, a Sand box, and a Production environment. Full Silver topology consists of an Authoring, Sandboxes, a Pre-prod and a Production environment.



There is one Decision Center to govern all Decision Servers.
All environments use the same Identity Access Management (IAM) for authentication and the databases are externalized.

Silver topology is best suited for applications with **medium production constraints.**

For more information about ODM environments and the topologies, see CP4BA ODM topologies on OpenShift.

Summary

Silver topology is based on default bronze topology with additional customizations to allow several ODM environments within the same cluster.
To install ODM Silver topology, it is recommended to start with the ODM Bronze topology article and obtain a baseline CP4BA deployment Custom Resource (CR) YAML file.
Use this CR file and customize it per ODM environment.
Other settings such as IAM configuration and certificates management are discussed in the later part of this article.

Procedure

1. Generate a CR file for ODM Bronze topology. See [ODM Bronze topology]() for more information;
2. Copy the CR file and rename it as `<your_odm_env>.yaml`;
3. Assign a value to the metadata.name parameter in the CR. For example, metadata.name: sandbox
4. Set `spec.shared_configuration.sc_deployment_profile_size` to `medium` for Cloud Pak deployment profile. The deployment profile of CP4BA is `small` by default. It is recommended to set to **medium** for Silver topology environments and set the IBM Cloud Platform UI (Zen) service to the same size as Cloud Pak. For more information, see https://ibmdocs-test.mybluemix.net/docs/en/cloud-paks/cp-biz-automation/21.0.3?topic=ppd-system-requirements
5. Remove the following unwanted parameters:
    - *sc_deployment_fncm_license: "<Required>"*
    - *sc_deployment_baw_license: "<Required>"*
    - *sc_ingress_enable: false*
    - *sc_ingress_tls_secret_name: <Required>*
    - *sc_cpe_limited_storage: false*
6. Fill in `image_pull_secrets` per your specific shared image pull secrets (if not so).
7. Fill in `ldap_configuration` per your LDAP configuration.
8. In `datasource_configuration` section, fill in `dc_odm_datasource` per your database configuration.
    - An example for DB2 with SSL enabled:

```
datasource_configuration:
  dc_odm_datasource:
    database_servername: <db2_hostname>
    dc_common_database_instance_secret: <db2_credentials>
    dc_common_database_name: <odm_db_name>
    dc_common_database_port: '60001'
    dc_common_ssl_enabled: true
    dc_database_type: db2
    dc_ssl_secret_name: <db2_ssl_cert>
  dc_ssl_enabled: true
```

    - If SSL is used to secure the database connection, specify the name of the SSL secret by running the following command:

```
oc create secret generic odm-db-ssl-secret --from-file=db2-server-certificate=server.crt
```

Where server.crt is the DB2 SSL certificate public key in ASCII format:

```
-----BEGIN CERTIFICATE-----

MIIHDzCCBfegAwIBAgIQCKZtYygfn9pg13D0uAX YzANBgkqhkiG9
w0BAQsFADBg ... 3R7IrdK8aS1WUGlKulqEDiV4TJ 1XpcoUq8wt
mBSw1fyV7g=

-----END CERTIFICATE-----
```

For more details on how to generate the DB2 SSL certificate, see Self-signing digital certificates.

9. Move the parameter `spec.odm_configuration.deployment_profile_size` to `custom`.

```
odm_configuration:
  deployment_profile_size: custom
```

10. Modify spec.odm_configuration and spec.olm_production_option.decisions sections according to your specific ODM environment.

Authoring environment

Authoring environment consists of:

- Decision Server Console;
- 2 Decision Center;
- 2 Decision Runner;

Edit your Authoring CR file to install these components.

For example:

```
odm_configuration:
  deployment_profile_size: custom
  decisionCenter:
    enabled: true
    replicaCount: 2
    resources:
      limits:
        cpu: '2'
        memory: 8Gi
      requests:
        cpu: '1'
        memory: 4Gi
  decisionServerRuntime:
    enabled: false
  decisionRunner:
    enabled: true
    replicaCount: 2
    resources:
      limits:
        cpu: '2'
        memory: 2Gi
      requests:
        cpu: 500m
        memory: 2Gi
  decisionServerConsole:
    resources:
      limits:
        cpu: '2'
        memory: 1Gi
      requests:
        cpu: 500m
        memory: 512Mi
...
olm_production_option:
  decisions:
    bai: false
    decisionCenter: true
    decisionRunner: true
    decisionServerRuntime: false
```

Sandbox environment

Sandbox environment consists of :

- Decision Server Console;
- Decision Server Runtime;

Edit your sandbox CR file to only install Decision Server Console and Decision Server Runtime.

For example:

```
odm_configuration:
  deployment_profile_size: custom
  decisionCenter:
    enabled: false
  decisionServerRuntime:
    enabled: true
    replicaCount: 1
    resources:
      limits:
        cpu: '2'
        memory: 2Gi
      requests:
        cpu: '2'
        memory: 2Gi
  decisionRunner:
    enabled: false
  decisionServerConsole:
    resources:
      limits:
        cpu: '2'
        memory: 1Gi
      requests:
        cpu: 500m
        memory: 512Mi
...
olm_production_option:
...
  decisions:
    bai: false
    decisionCenter: false
    decisionRunner: false
    decisionServerRuntime: true
```

Production & Pre-Production environment

Production environment consists of:

- Decision Server Console;
- 3 Decision Server Runtime;

Edit your Production CR file to install these components.

For example:

```
odm_configuration:
  deployment_profile_size: custom
  decisionCenter:
    enabled: false
  decisionServerRuntime:
    enabled: true
    replicaCount: 3
    resources:
      limits:
        cpu: '2'
        memory: 2Gi
      requests:
        cpu: '2'
        memory: 2Gi
  decisionRunner:
    enabled: false
  decisionServerConsole:
    resources:
      limits:
        cpu: '2'
        memory: 1Gi
      requests:
        cpu: 500m
        memory: 512Mi
...
olm_production_option:
  decisions:
    bai: false
    decisionCenter: false
    decisionRunner: false
    decisionServerRuntime: true
```

Perform some additional tuning to this Production or Pre-Production environment:
See Performance Check List of OCP for CP4BA.

Cluster Configuration

IAM config

You have several authentication mechanisms available by default in the Cloud Pak.

You can configure your Lightweight Directory Access Protocol (LDAP) connection to use the service that provides authentication, role-based access control, and user management.

Along with the LDAP enterprise-wide authentication system, you can also choose to register additional users that will use a simpler but faster authentication system.

As the LDAP mechanism has a small (but not null) overhead, you can use the Basic Authentication system as a replacement, especially for the Runtime part, which is supposed to be used at a higher rate usage.

LDAP

To avoid SSL error when connecting to LDAP, you have to manually configure the LDAP certificate following instructions from this page: https://www.ibm.com/docs/en/cpfs?topic=ldap-configuring-over-ssl

Once connected, you can refine your user management role.

Use the "spec.odm_configuration.customization.authSecretRef" parameter of the CR file to specify your own custom secret and change the webSecurity.xml file section.

See: https://www.ibm.com/docs/en/odm/8.11.0?topic=access-configuring-user-without-openid

As an example:

```
[...]

  <group name="Business User" roles="rtsUser,resDeployerDev,resDeployerTest"/>

  <group name="Release Manager" roles="resDeployerTest,resDeployerProd,resDeployerDev,rtsConfigManager"/>

  <group name="Rule Developer" roles="rtsUser,resDeployerDev"/>

  <group name="Integrator" roles="resDeployerDev,resDeployerTest,resMonitorProd,rtsUser"/>

  <group name="Permission Manager" roles="rtsAdministrator,resDeployerDev,resDeployerTest,resDeployerProd"/>
  <variable name="odm.resExecutors.group1" value="group:odm-ldap/cn=resDeployerProd,{{ ldap_groups_dn }},{{ ldap_basedn_upper }}"/>

  <variable name="odm.resExecutors.group2" value="group:odm-ldap/cn=resMonitorProd<,{{ ldap_groups_dn }},{{ ldap_basedn_upper }}"/>

[...]
```

Basic Authentication

You must override the default user registry configuration and include it in a secret that you pass to ODM Configuration on CP4BA through the CR file.

Use the `spec.odm_configuration.customization.authSecretRef` parameter of the CR file.

You must adapt the group mapping on the resExecutors group that is inside `webSecurity.xml` file section.

To execute with resExecutor for example, you need something like :

```
<variable name="odm.resExecutors.group1" value="group:basic/basicResExecutors"/>
```

If you want to execute with odmAdmin, you need :

```
<variable name="odm.resExecutors.group2" value="group:basic/basicRtsAdministrators"/>
```

and so on ...

As an example, the newly created `my-company-odm-oidc-auth-operator-secret` of prod namespace, the `webSecurity.xml` file contains:

```
<server>
     <basicRegistry id="basic" realm="basic">
         <user name="dbauser" password="dbauser"/>
         <user name="odmAdmin" password="odmAdmin"/>
         <group name="resExecutors">
             <member name="dbauser"/>
             <member name="odmAdmin"/>
         </group>
     </basicRegistry>

     <variable name="odm.resExecutors.group2" value="group:basic/resExecutors"/>
     <variable name="odm.resAdministrators.group2" value="group:basic/resExecutors"/>
     <variable name="odm.rtsAdministrators.group2" value="group:basic/resExecutors"/>
</server>
```

For more information about this customization, please read the following documentation: https://www.ibm.com/docs/en/odm/8.11.0?topic=access-configuring-user-without-openid

Other Customizations

In addition to the webSecurity.xml file in which you define the users allowed to access the application server, you have the option to specify up to four additional files to configure the access to an ODM instance. These customizations are implemented with "spec.odm_configuration.customization.authSecretRef" parameter of the CR file.

See: https://www.ibm.com/docs/en/cloud-paks/cp-biz-automation/21.0.3?topic=access-optional-user-configurations

Certificate management

Along with the LDAP connection issue that you can meet in secured connection context, the secured connection to other namespaces is not configured by default and needs to manually push the certificate.

LDAP

As mentioned in the IAM Configuration above, you must manually configure the LDAP certificate to avoid SSL error connecting to LDAP, using this page: https://www.ibm.com/docs/en/cpfs?topic=ldap-configuring-over-ssl

Namespace

By default, a secured connection between Decision Center to another namespace (let's say a Decision Server Console as an example), leads to an error like the following one:

javax.net.ssl.SSLHandshakeException: java.security.cert.CertificateException: Path does not chain with any of the trust anchors

To overcome this issue, you need to import cert Production environment to Authoring environment:

- Extract the RES certificate, downloaded from the RES console prod environment using a browser.
- Create a new secret in authoring namespace (`key=tls.crt` with cert content);

```
oc create secret generic my-prod-namespace-secret --from-file=
tls.crt=your_cert_path/server-cert
```

- Specify this secret as your custom one in the list of secrets registered in the `spec.shared_configuration.trusted_certificate_list` parameter of the CR file.

```
shared_configuration:
   trusted_certificate_list:
     - my-prod-namespace-secret
```

- Wait for some minutes while the ODM Pods restart

Rule Designer

To be able to securely connect your Rule Designer to the Decision Server and Decision Center components that are running in an OCP cluster, you need to establish a Transport Layer Security (TLS) connection through a security certificate.

See: https://www.ibm.com/docs/en/cloud-paks/cp-biz-automation/21.0.3?topic=designer-importing-security-certificate-in-rule

Reaching out external services

As a conclusion, to integrate with an external service in general, you must first import its TLS certificate into the operator trust list.

These certificates are added to the truststore of each component in the Cloud Pak.

The procedure is described in the following documentation part: https://www.ibm.com/docs/en/cloud-paks/cp-biz-automation/21.0.3?topic=services-importing-certificate-external-service

Once everything is well deployed, you can perform the verification described in the last section of your generated installation playbook (see the Bronze Topology Article). Further additional validations can be made at ODM level which are described here: validate your ODM topology.