

# Pre-study on CDS

How to setup CDS step by step:

## Objective

The purpose of this page is to show how to run the Blueprints Processor microservice locally, using the docker-compose.yaml file provided in the project.

## Check out the CDS' code

Check out the latest code from Gerrit: <https://gerrit.onap.org/r/#/admin/projects/ccsdk/cds>

## Build CDS locally

In the checked out directory, type

```
mvn clean install -DskipTests=true -Dmaven.test.skip=true -Dmaven.javadoc.skip=true -Dadditionalparam=-Xdoclint:none
```

## Create the needed Docker images

The Blueprints Processor microservice project has a module, called distribution, that provides a docker-compose.yaml file that can be used to spin up Docker containers to run this microservice.

The first step is to create any custom image needed, by building the distribution module. From the CDS home directory (where the code was checked out), navigate to the module:

```
cd ms/blueprintsprocessor/
```

Build it using the Maven profile called Docker:

```
mvn clean install -Pdocker  
blocked URL
```

## Start Docker containers using docker-composer

Navigate to the docker-compose file in the distribution module:

```
cd application/src/main/dc/
```

From there, start the containers(unfortunately, this default docker-compose.yaml does not work properly in my env, I have to revise it a little bit manually. [Fast track, just copy https://gitlab.com/yanhuanwang/cds/blob/master/docker-compose.yaml](https://gitlab.com/yanhuanwang/cds/blob/master/docker-compose.yaml)):

```
docker-compose up -d
```

This will spin the Docker containers declared inside the docker-compose.yaml file in the background:

[blocked URL](#)

To verify the logs generated by docker-composer, type:

```
docker-compose logs -f
```

## Testing the environment

create CBA file:

```
cd cds/components/model-catalog/blueprint-model/uat-blueprints/pnf_config
```

```
zip -r ~/pnf_config.zip .
```

pnf\_config.zip is a valid CBA file.

then post this CBA file to CDS with command:

```
curl -X POST -u ccsdkapps:ccsdkapps -F file=@$HOME/pnf_conig.zip http://localhost:8000/api/v1/blueprint-model/publish | json_pp
```

console output:

```
chengkaiyan@Chengkais-MBP:~/cds/cds/components/model-catalog/blueprint-model/uat-blueprints/pnf_config$ curl -X POST -u ccscdkapps:ccscdkapps -F file=@$HOME/pnf_config.zip http://localhost:8000/api/v1/blueprint-model/publish | json_pp
```

```
% Total % Received % Xferd Average Speed Time Time Time Current
Dload Upload Total Spent Left Speed
100 11395 100 421 100 10974 1019 26571 --:--:-- --:--:-- --:--:-- 27590
```

```
{
  "blueprintModel" : {
    "updatedBy" : "Rodrigo Ottero",
    "tags" : "pnf, restconf, config, configuration",
    "artifactUUid" : null,
    "published" : "Y",
    "id" : "d212524b-9fec-45c5-bc61-1dc920e5b15e",
    "artifactDescription" : "Controller Blueprint for configuration_over_restconf:1.0.0",
    "createdDate" : "2019-11-11T12:39:08.000Z",
    "internalVersion" : null,
    "artifactType" : "SDNC_MODEL",
    "artifactVersion" : "1.0.0",
    "artifactName" : "configuration_over_restconf"
  }
}
```

afterwards, you can see the model list with command:

```
curl -u ccscdkapps:ccscdkapps http://localhost:8000/api/v1/blueprint-model/ | json_pp
```

console output:

```
chengkaiyan@Chengkais-MBP:~/cds/cds/components/model-catalog/blueprint-model/uat-blueprints/pnf_config$ curl -u ccscdkapps:ccscdkapps http://localhost:8000/api/v1/blueprint-model/ | json_pp
```

```
% Total % Received % Xferd Average Speed Time Time Time Current
Dload Upload Total Spent Left Speed
100 1203 100 1203 0 0 13670 0 --:--:-- --:--:-- --:--:-- 13670
```

```
{
  {
    "blueprintModel" : {
      "id" : "408cb6fd-1aa1-4d66-8078-51ff1ac3dbc3",
      "artifactVersion" : "1.0.0",
      "artifactUUid" : null,
      "artifactDescription" : "Controller Blueprint for capability-cli:1.0.0",
      "updatedBy" : "Brinda Santh Muthuramalingam",
      "internalVersion" : null,
      "createdDate" : "2019-11-11T08:40:44.000Z",
      "artifactName" : "capability-cli",
      "tags" : "brinda, tosca",
      "artifactType" : "SDNC_MODEL",
      "published" : "Y"
    }
  },
  {
    "blueprintModel" : {
      "published" : "Y",
      "tags" : "brinda, tosca",
      "artifactType" : "SDNC_MODEL",
      "artifactVersion" : "1.0.0",
      "id" : "b1902cff-5d2f-406d-bfc8-1533cda4cf40",
      "artifactDescription" : "Controller Blueprint for baseconfiguration:1.0.0",
      "artifactUUid" : null,
      "updatedBy" : "Brinda Santh Muthuramalingam",
      "artifactName" : "baseconfiguration",
      "internalVersion" : null,
      "createdDate" : "2019-11-11T08:26:47.000Z"
    }
  },
  {
    "blueprintModel" : {
      "id" : "d212524b-9fec-45c5-bc61-1dc920e5b15e",
      "artifactVersion" : "1.0.0",
      "artifactUUid" : null,
      "artifactDescription" : "Controller Blueprint for configuration_over_restconf:1.0.0",
      "updatedBy" : "Rodrigo Ottero",
      "createdDate" : "2019-11-11T12:39:08.000Z",
      "artifactName" : "configuration_over_restconf",
      "internalVersion" : null,
      "tags" : "pnf, restconf, config, configuration",
      "artifactType" : "SDNC_MODEL",
      "published" : "Y"
    }
  }
}
```

