

Virtualization Quick Start Guide

This document provides a guide to set-up and execute a quick start for creating and testing a virtualization using the Openshift console (via Minishift) and Beetle Studio.

Set-up

1. You'll need an running **Openshift** container and we recommend installing **Minishift**.
 - a. <https://www.okd.io/minishift/>
 - b. <https://github.com/minishift/minishift/releases/tag/v1.15.1> (this older version is recommended for this quick start)
 - c. <https://docs.okd.io/latest/minishift/getting-started/installing.html>

2. Open terminal and clone beetle-studio source to your file system
 - a. `git clone https://github.com/teiid/beetle-studio`
 - b. `cd` into `openshift` folder
 - c. Start minishift
 - i. `./minishift-start.sh`
 1. Configures allocated memory and cpus for openshift and starts minishift with access to service-catalog)
 - ii. The following will be logged when complete
`[username@localhost minishift-beetle]$ OpenShift server started.`
The server is accessible via web console at:
<https://192.168.<?>.<?>:8443>

3. Log into Openshift
 - a. `oc login -u developer 192.168.<?>.<?>:8443`
 - i. Your unique Openshift IP address will be shown in in your terminal window
 - b. `oc adm policy add-cluster-role-to-user cluster-admin developer`

4. Open Openshift console
 - Open browser to <https://192.168.<?>.<?>:8443/console>
 - username > developer
 - password > developer

5. The console should show the available service catalog applications and an example "MyProject" project

Install Beetle Studio

These steps will build and deploy the beetle studio UI and all dependent/required applications

1. In terminal, navigate to the */openshift* folder in your **beetle-studio** project
2. Execute `./beetle-os-setup.sh -h 192.168.xx.xxx` (or your IP address)
 - a. This will create a new “**beetle-studio**” project in your console kick off image builds for beetle studio, including a wildfly-swarm, vdb builder and postgresql data store for the vdb-builder’s (komodo) persistence framework
 - i. There should be 4 applications in various states of build
 1. Beetle-studio
 2. Vdb-builder
 3. Vdb-builder-persistence
 4. das
 - b. Note that initially, the **vdb-builder** application will probably take around 30 minutes to build, based on your local hardware/system. If this build fails, just launch it again by selection the “**Start Build**” action in the right hand drop down menu on the **vdb-builder** application. The second build should be faster... around 6-7 minutes.

Load Sample Data

Virtualization is about providing a virtual access to your data. These steps will guide you through setting up a schema in a PostgreSQL database in your Openshift console

1. In the **Add to Project** menu on the right of the header, select **Browse Catalog**
2. Select **PostgreSQL** icon to launch the wizard
3. Enter:
 - **partsdb** (for database service name)
 - **komodo** (username)
 - **komodo** (password)
 - **parts** (database name)
4. select **Next >** and choose the “**Create a secret in beetle-studio to be used later**” binding option then **Create**
 - a. This will take a few minutes to create, deploy and activate the pod
5. Select **Applications > Pods**
 - a. select the pod “**partsdb-*<number>-<id>***” (example: partsdb-1-4r8rj) to view details
 - b. select the **Terminal** tab to show command line terminal
 - c. enter “**psql parts komodo**” and click enter/return
 - d. paste in the text from **parts-postgresql.sql** file in your *.../openshift* folder (This will take a couple minutes)

- e. *When finished, verify the data was loaded by entering "select * from parts;" and pressing enter*
6. *On your **Overview** page, you should now have a **parts** pod and a 1 **PostgreSQL** binding*

Create Connection

These steps will lead you through creating your first data source connection in Beetle Studio.

1. Launch Beetle Studio by opening **Applications > Routes** page
2. Click on the **das** route **host name**
 - a. Example: <https://das-beetle-studio.192.168.42.247.nip.io>
3. Enter **developer / developer** credentials and click **Log In**
4. Click **Allow selected permissions**
5. Click **Add Connection**
 - a. Select the **postgresql** connection type
 - Enter a name (i.e. **partsDB**)
 - verify that your **postgresql** service catalog source is auto-selected
 - select **Create**
 - b. When finished (**Create was successful**) click on **View all Connections**
6. The new connection is processed and will be available for use when the active green check-box is displayed.

Create Virtualization

These steps will lead you through creating your first data source virtualization in Beetle Studio.

1. Select **Virtualizations** tab
2. Click **Add Data Virtualization**
 - a. In the dialog enter
 - virtualization Name (i.e. **partsVirt**)
 - view name (i.e. **supplierInfo**)
 - descriptions (optional)
 - b. Click **OK**
3. When the virtualization is created, the Virtualization editor is opened and your new empty view is selected
4. Click on the **Add Source** database icon in the toolbar
 - a. Expand the tree and select the **supplier** table and click **OK**
 - b. Click **Save** button on the toolbar
 - c. When finished saving, the editor will execute a query for your view and show sample results in the **Preview** panel
5. Select the source icon in the diagram and click the **Add Composition** button in the toolbar
 - a. This will launch a wizard to select a table to join with the first table
6. In the wizard
 - a. Expand the tree and select the **status** table and click **Next >**
 - b. Select the **supplier_status** column for both the **supplier** table and **status_id** for the **status** table for the join criteria.

- c. Click **Finish**
7. The cities source table and composition (named **supplier-status**) will be shown in the diagram.
8. Click **Save** button and the Preview will update to show the new results for the full join from your view

Activate and Preview Virtualization

1. Select **Virtualizations** tab
2. For **partsVirt** virtualization, select the **Activate** dropdown action
 - a. This step will create a testable instance of your virtualization that is not yet an image
3. When activated (Green status) you can click the **Preview** button (spyglass) to see sample data for your **supplierInfo** view

Publish and test your deployed virtualization image

1. For **partsVirt** virtualization, select the **Publish** dropdown action
 - a. Wait for the build job is complete
 - i. It's pod in the console will be named something like:
partsvirtvdb-build-config-1-build
 - b. When build and deployed a green set of gears icon will indicate it's published and a **Odata Preview** button will be displayed
2. Select the **Odata Preview** button to open the Odata query editor
 - a. Choose desired columns and relevant criteria and click **Submit** to see results