

Using Published DSpace Images in Codenvy

This approach is not recommended

 As of Dec 2018, the Codenvy service appears to no longer be supported.

- [Overview](#)
- [Codenvy RAM Usage](#)
- [Try out Codenvy Using this Pre-Built Factory](#)
- [Note about the dspace/tomcat container](#)
- [Copy Files between Containers with SSH](#)
- [Utility Script for Codenvy](#)
- [Configuring a Codenvy Workspace Yourself](#)
 - Create a multi-machine stack that contains the desired images plus add an additional machine (eclipse/ubuntu_jdk8) to run the IDE.
 - Explicitly create a "server" for the container port to expose with a public URL.
 - After the workspace starts, the public address for this "server" will be exposed on the workspace configuration screen.

Overview

Updated Resource

 See <https://dspace-labs.github.io/DSpace-Docker-Images/add-ons/codenvy-scripts/>

The [Codenvy Service](#) provides a cloud-hosted instance of the Eclipse Che editor and a cloud hosted service for running Docker containers.

The service has many compelling features and some limitations. Customer support is non-existent even for paid accounts.

Codenvy RAM Usage

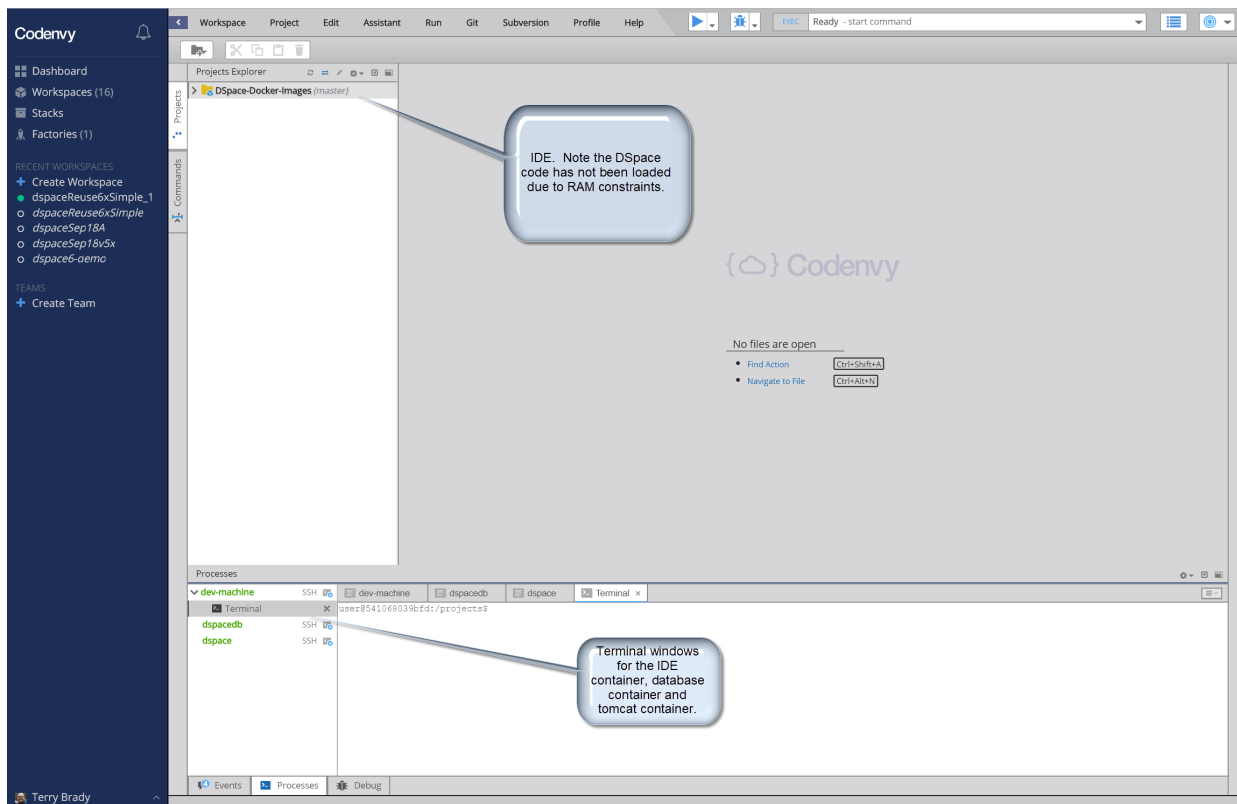
Codenvy provides a free tier account that uses up to 3G of RAM.

This is sufficient to run a small DSpace instance.

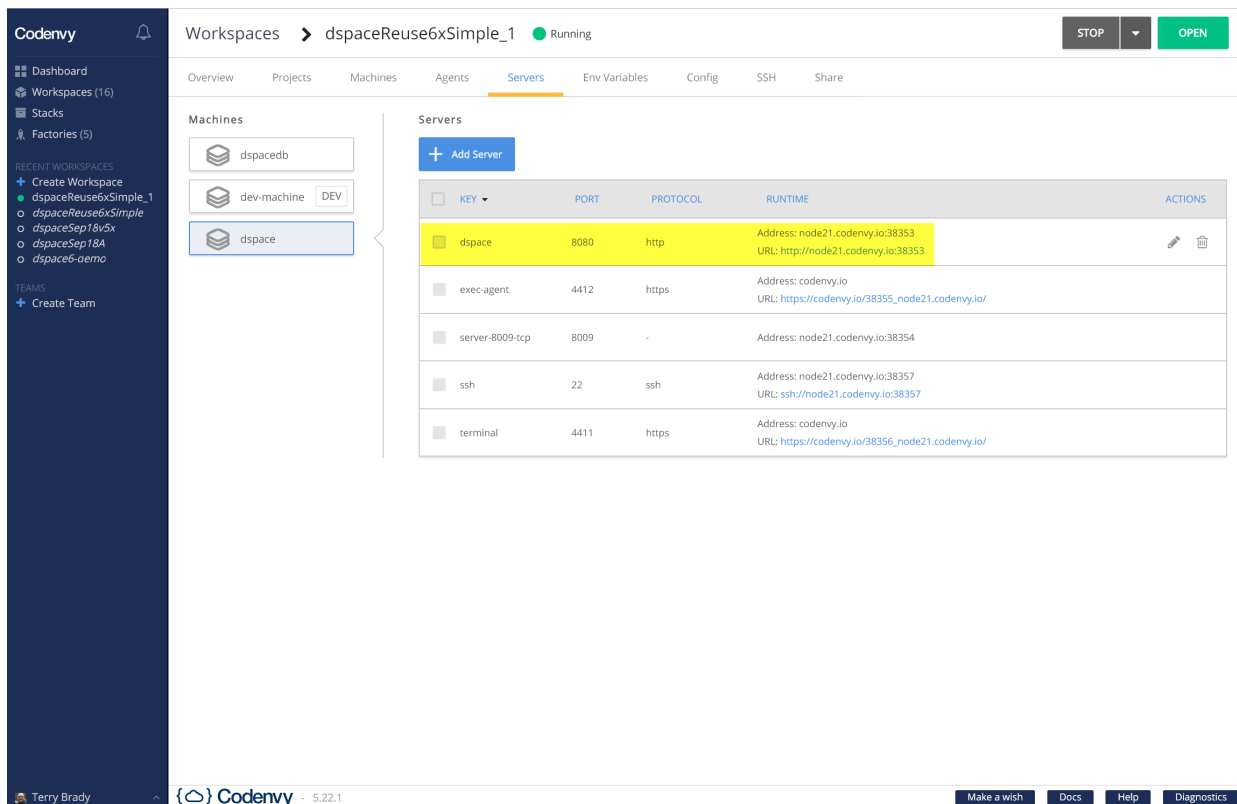
The Codenvy IDE (Che) requires at least 3G of RAM to manage the full DSpace Code base. Therefore, it is impossible to manage code and run DSpace in a free tier workspace.

Try out Codenvy Using this Pre-Built Factory

The following [link](#) to a Codenvy factory will build a 3GB workspace running the latest DSpace 6x published Docker Image.



The workspace properties panel will expose a public URL for the tomcat instance.



Note about the dspace/tomcat container

The tomcat instance (container named dspace) cannot be stopped or restarted. If you wish to reload a webapp, you should modify the webapp web.xml file in /dspace/webapps/*/WEB-INF/web.xml

Copy Files between Containers with SSH

The following link explains how to use SSH keys to move data between containers or from your desktop to one of the containers.

<https://github.com/codenvy/codenvy/issues/2699>

Utility Script for Codenvy

The following utility script will perform some of the special overrides that we have configured within our dspace compose files.

<https://github.com/DSpace-Labs/DSpace-Docker-Images/blob/codenvy-refactor/add-ons/codenvy-scripts/setupCodenvy.sh>

Configuring a Codenvy Workspace Yourself

Create a multi-machine stack that contains the desired images plus add an additional machine (eclipse/ubuntu_jdk8) to run the IDE.

Here is a docker compose configuration that I wish to replicate

Docker Compose Configuration to Port to Codenvy

```
services:
  dspacedb:
    image: dspace/dspace-postgres-pgcrypto
    volumes:
      - pgdata:/pgdata

  dspace:
    image: "dspace/dspace:dspace-6_x"
    ports:
      - 8080:8080
    volumes:
      - "assetstore:/dspace/assetstore"
      - "solr:/dspace/solr"
    depends_on:
      - dspacedb
```

The syntax is a subset of what is available to docker-compose. The flattening of the YAML makes it difficult to edit.

Codenvy Stack Recipe for DSpace

```
"recipe": {
  "type": "compose",
  "content": "services:\n dev-machine:\n  image: eclipse/ubuntu_jdk8\n dspacedb:\n  image: dspace/dspace-postgres-pgcrypto\n  mem_limit: 536870912\n dspace:\n  image: 'dspace/dspace:dspace-6_x'\n  mem_limit: 2147483648\n",
  "contentType": "application/x-yaml"
},
```

Codenvy provides editing tools to add "machines" to a "stack configuration".

Machines

DEV-MACHINE DEV

DSPACE

DSPACEDB

Recipe



Show

Add

Explicitly create a "server" for the container port to expose with a public URL.

Codenvy "Server" Configuration within a Machine

```
"dspace": {
  "attributes": {
    "memoryLimitBytes": "2684354560"
  },
  "servers": {
    "dspace": {
      "properties": {},
      "protocol": "http",
      "port": "8080"
    }
  }
},
```

After the workspace starts, the public address for this "server" will be exposed on the workspace configuration screen.

See the instructions above for finding this URL.