Hosting WikiBase on EduCloud

This is a guide to hosting a WikiBase on EduCloud

Before we begin:
Running WikiBase on your machine.
Debugging WikiBase:

Before we begin:

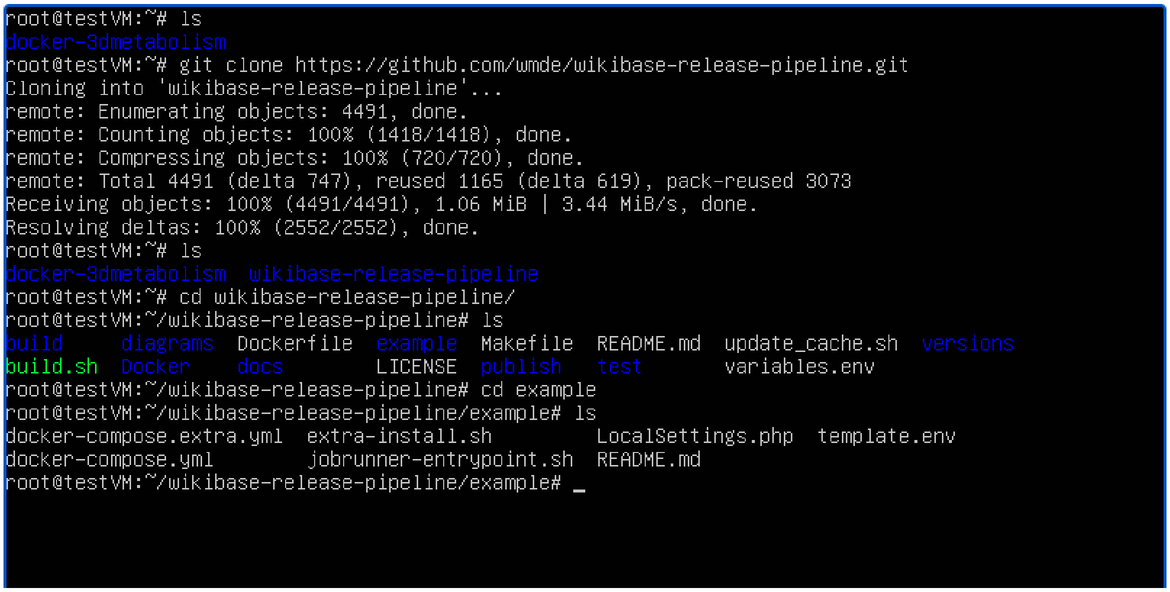
- You must have reached out to your adminstrators to have EduCloud access.
- The WikiBase we will setup is based off the Docker-WikiBase images.
- This document is an extenstion and a specific usecase of the previous document titled, "Hosting Docker on EduCloud". We will assume that you have read and followed the same workflow to setup your Virtual Machine.
- The repository we will clone is actively being maintained. There might be slight variations between this document and the state of the repository you will be working with. I can only promise that the workflow of this document works as of the time I am writing this.

Running WikiBase on your machine.

1. Begin by cloning the WikiBase-release-pipeline repository onto your virtual machine:

   ```
   git clone https://github.com/wmde/wikibase-release-pipeline.git
   ```

2. one cloned, enter the example directory within the repository. Useful commands to achieve this are `cd` and `ls`



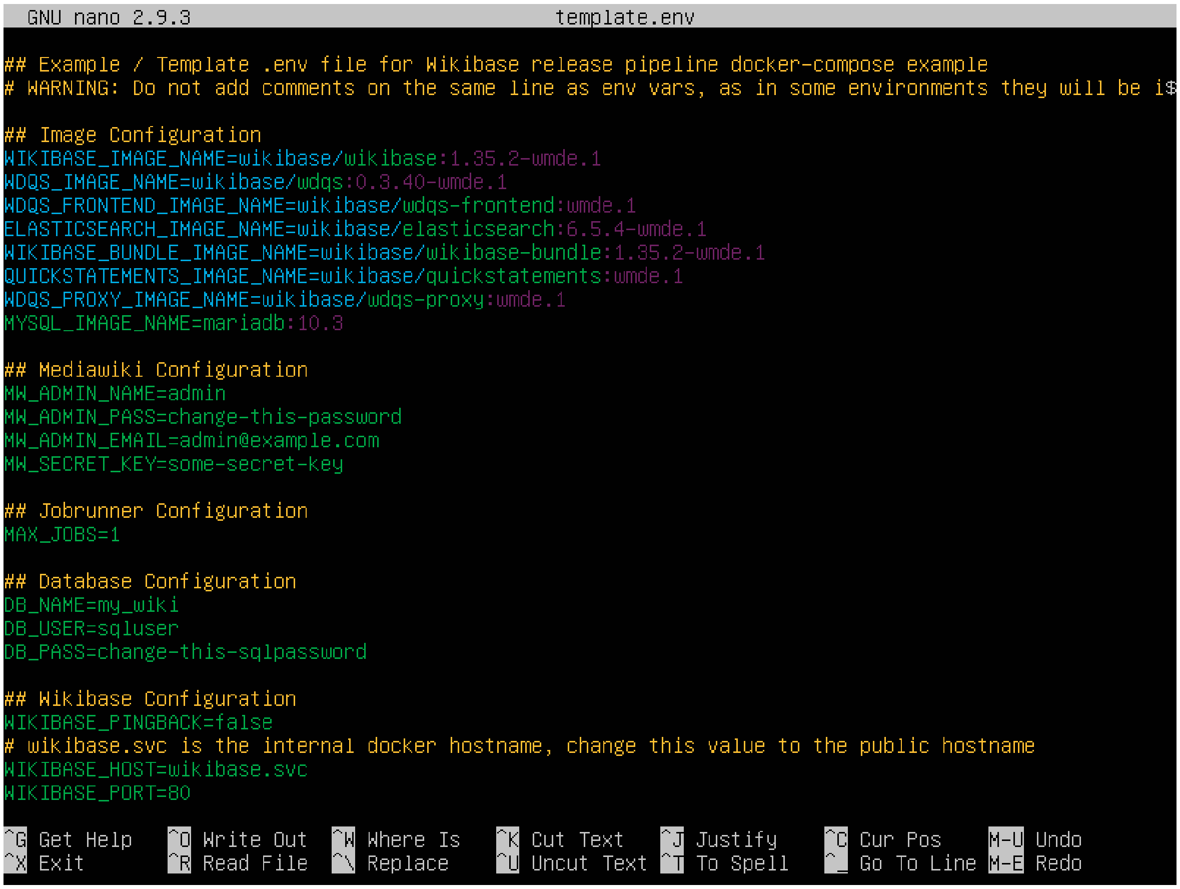
```
root@testVM:~# ls
docker-3dmetabolism
root@testVM:~# git clone https://github.com/wmde/wikibase-release-pipeline.git
Cloning into 'wikibase-release-pipeline'...
remote: Enumerating objects: 4491, done.
remote: Counting objects: 100% (1418/1418), done.
remote: Compressing objects: 100% (720/720), done.
remote: Total 4491 (delta 747), reused 1165 (delta 619), pack-reused 3073
Receiving objects: 100% (4491/4491), 1.06 MiB | 3.44 MiB/s, done.
Resolving deltas: 100% (2552/2552), done.
root@testVM:~# ls
docker-3dmetabolism  wikibase-release-pipeline
root@testVM:~# cd wikibase-release-pipeline/
root@testVM:~/wikibase-release-pipeline# ls
build     diagrams  Dockerfile  example  Makefile  README.md  update_cache.sh  versions
build.sh  Docker    docs        LICENSE  publish   test       variables.env
root@testVM:~/wikibase-release-pipeline# cd example
root@testVM:~/wikibase-release-pipeline/example# ls
docker-compose.extra.yml  extra-install.sh         LocalSettings.php  template.env
docker-compose.yml        jobrunner-entrypoint.sh  README.md
root@testVM:~/wikibase-release-pipeline/example# _
```

3. Edit the `template.env` file. This file contains the configuration settings of your docker-compose images. I recommend using the `nano` command to make the adjustments however you are free to use any editor. Do not save until after step 5 of this section.

    a. It is recommended to change your MediaWiki configuration to secure your Admin settings of the page.

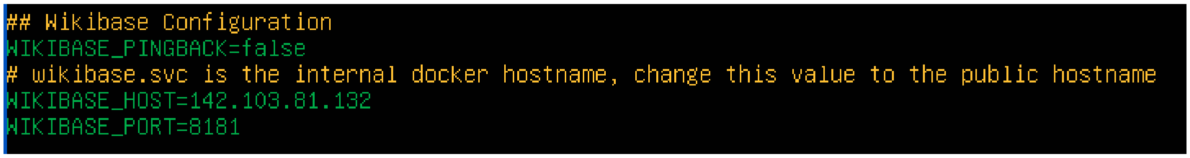
    b. You can adjust the ports in which your end-users will listen into with there computers. These ports should be the same ports in which you ask your EduCloud administrators or UBC-LT to expose for you.

    c. If you plan to further customize the wikibase, it is recommended to create your own WikiBase image and host it on DockerHub. Once done, adjust the "Image Configuration" settings such that it references your custom image on DockerHub.



```
## Example / Template .env file for Wikibase release pipeline docker-compose example
# WARNING: Do not add comments on the same line as env vars, as in some environments they will be i$

## Image Configuration
WIKIBASE_IMAGE_NAME=wikibase/wikibase:1.35.2-wmde.1
WDQS_IMAGE_NAME=wikibase/wdqs:0.3.40-wmde.1
WDQS_FRONTEND_IMAGE_NAME=wikibase/wdqs-frontend:wmde.1
ELASTICSEARCH_IMAGE_NAME=wikibase/elasticsearch:6.5.4-wmde.1
WIKIBASE_BUNDLE_IMAGE_NAME=wikibase/wikibase-bundle:1.35.2-wmde.1
QUICKSTATEMENTS_IMAGE_NAME=wikibase/quickstatements:wmde.1
WDQS_PROXY_IMAGE_NAME=wikibase/wdqs-proxy:wmde.1
MYSQL_IMAGE_NAME=mariadb:10.3

## Mediawiki Configuration
MW_ADMIN_NAME=admin
MW_ADMIN_PASS=change-this-password
MW_ADMIN_EMAIL=admin@example.com
MW_SECRET_KEY=some-secret-key

## Jobrunner Configuration
MAX_JOBS=1

## Database Configuration
DB_NAME=my_wiki
DB_USER=sqluser
DB_PASS=change-this-sqlpassword

## Wikibase Configuration
WIKIBASE_PINGBACK=false
# wikibase.svc is the internal docker hostname, change this value to the public hostname
WIKIBASE_HOST=wikibase.svc
WIKIBASE_PORT=80
```

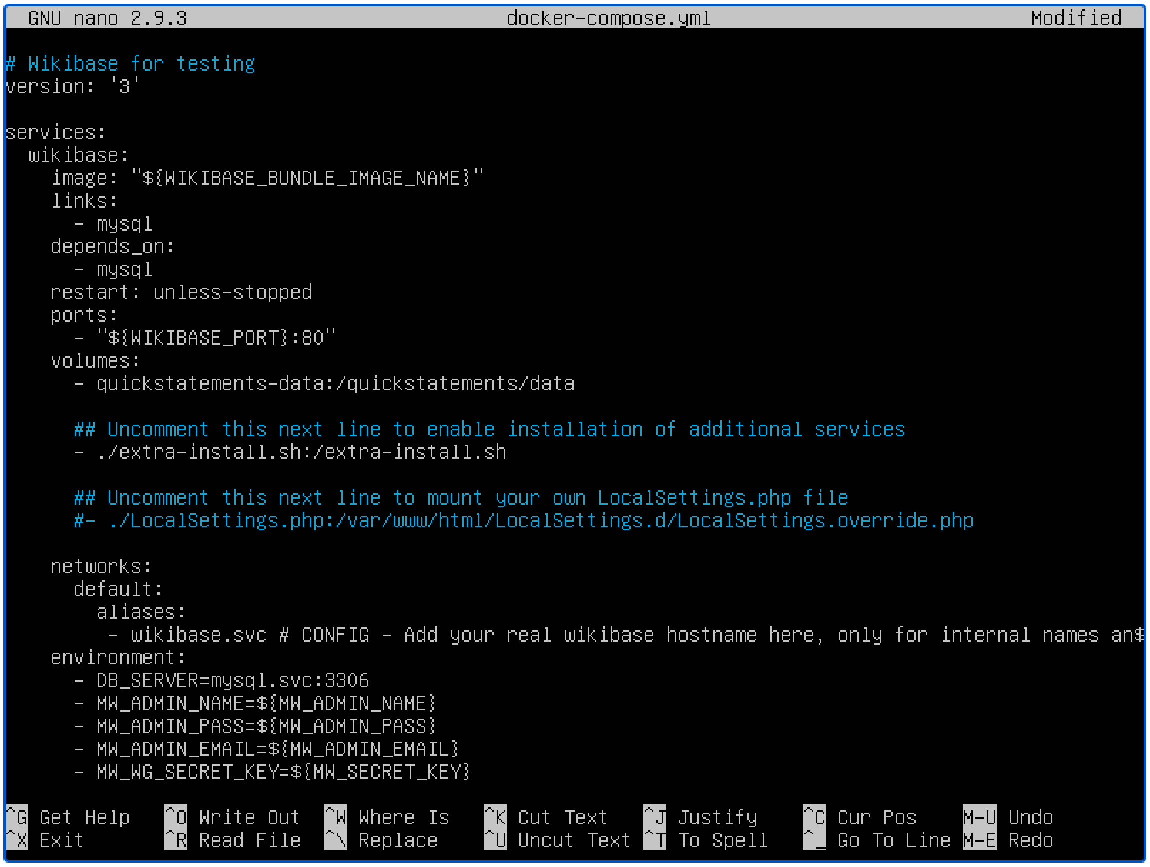
4. Be sure to adjust the `WIKIBASE_HOST` variable such that it is equal to the public-IP address of your virtual machine.



```
## Wikibase Configuration
WIKIBASE_PINGBACK=false
# wikibase.svc is the internal docker hostname, change this value to the public hostname
WIKIBASE_HOST=142.103.81.132
WIKIBASE_PORT=8181
```

5. Once adjusted, save the document as a new file called `.env`. Exit the editor window

6. proceed to edit the `docker-compose.yml` and uncomment the `./extra-install.sh:/extra-install.sh` line. Additionally if you would like to use the LocalSettings.php provided in the example directory, uncomment the additional line shown in the screenshot below. Exit the editor window once completed.



```
# Wikibase for testing
version: '3'

services:
  wikibase:
    image: "${WIKIBASE_BUNDLE_IMAGE_NAME}"
    links:
      - mysql
    depends_on:
      - mysql
    restart: unless-stopped
    ports:
      - "${WIKIBASE_PORT}:80"
    volumes:
      - quickstatements-data:/quickstatements/data

      ## Uncomment this next line to enable installation of additional services
      - ./extra-install.sh:/extra-install.sh

      ## Uncomment this next line to mount your own LocalSettings.php file
      #- ./LocalSettings.php:/var/www/html/LocalSettings.d/LocalSettings.override.php

    networks:
      default:
        aliases:
         - wikibase.svc # CONFIG - Add your real wikibase hostname here, only for internal names an$
    environment:
      - DB_SERVER=mysql.svc:3306
      - MW_ADMIN_NAME=${MW_ADMIN_NAME}
      - MW_ADMIN_PASS=${MW_ADMIN_PASS}
      - MW_ADMIN_EMAIL=${MW_ADMIN_EMAIL}
      - MW_WG_SECRET_KEY=${MW_SECRET_KEY}
```

onnected.

7. We are now ready to run the Docker images. If you have a pre-exisiting WikiBase backup files, this is the time to restore from your backup. Refer to the document titled "WikiBase - Docker: Backup and Restore Docker Volumes". The steps in the document will work to restore the file however there is a small difference in the overall logic; Knowing this may help any debugging issues down the road. instead of concatinating the restore files to the WikiBase volumes, restore directly to an empty volume which you will name. The wikibase docker-images will read from these new volumes you have created.

   Knowing this additional information should not effect any steps that you should follow within that restore-wikibase document.

8. To deploy your WikiBase, run the following command:

   a. **IMPORTANT NOTE**: make sure to always **run** and **stop** the `docker-compose.yml` and `docker-compose.extra.yml` files together. If not, you may run into issues with volumes which require you to do a fresh install of the WikiBase volumes (i.e you loose your data).

```
docker-compose -f docker-compose.yml -f docker-compose.extra.yml up -d
```

9. If maintanance is required, make sure to turn off your WikiBase with the following command

```
docker-compose -f docker-compose.yml -f docker-compose.extra.yml down
```

10. You should be able to access your WikiBase with the public-IP you were given, and the port of the wikibase (refer to your `.env` file you edited prior). The format should be `IPAddress:Port` and should look something like `8.8.8.8:80`

Debugging WikiBase:

This section will have a series of commands I found useful when debugging your Docker setup.

- `lsof -i -a -P | grep LISTEN` → this command allows you to see ports being used by your system, and Docker
- `ping -c5` `google.com` → this allows you to ping any website. This is especially useful to check if you are connected to the internet
- `nano <filename you would like to edit>` → this is a text editor that will allow you to make any changes.
- `l` → this will let you see all files including hidden files within a directory
- `pwd` → this will print out the absolute path you are working in
- `docker container ls` → this will allow you to see all currently running docker containers.
- `docker volume ls` → this will allow you to see all volumes that have been made
- `cp <filename> <destination>` → allows you to copy the filename to the destination you have set. This can also be used to rename files by replacing the destination path to the new filename

- `mv <filename> <destination>` → acts the same like cp however it moves files rather than copies them over
- `docker container prune` → removes all stopped containers.
- `docker image prune` → removes unused dangling images. If `-a` is specified, it will also remove all images not referenced by any container
- `docker volume prune ` → removes all unused local volumes. (not referenced by any containers)
- `docker system prune` → Remove all unused containers, networks, images (both dangling and unreferenced), and optionally, volumes.
- `git checkout HEAD -- <filename>` → allows you to reset any changes you have made that is not on the head of the branch. This only works if you are working under a git repository.
- `docker-compose down --remove-orphans` → this specifically removes containers for services not defined in the compose file. This command is specifically useful for step 8 of the guide if you accidentally run the `docker-compose.yml` and `docker-compose.extra.yml` files seperately and bugs out.