

# DevOps Topics

## Overview

The purpose of this document is to specify different devops topics from building, testing and deploying to developing in a container with docker.

## Visual Studio Code

VS Code will be used for all major development. For the most up to date IDE, it is encouraged that users develop on the cloud development machine. The Remote-Containers and Remote-SSH extensions must be installed. Any additional extensions will be automatically installed when setting up the remote environment

## Setup

- On the local machine with VS Code, you will need to generate a keypair for Gitea

```
ssh-keygen -t ed25519 -C "your_email@example.com"
```

- Be sure to add it to the agent with

```
eval "$(ssh-agent -s)"
```

```
ssh-add <path-to-ssh-key>
```

Once your key has been added to Gitea, you can develop either on a remote machine or your local machine.

## On same local machine

- On the local machine with VS Code, generate another keypair to put on the remote machine

```
ssh-keygen -t ed25519 -C "your_email@example.com"
```

- On the local machine add the keypair to your ssh agent

```
eval "$(ssh-agent -s)"
```

```
ssh-add <path-to-ssh-key>
```

- On the remote machine, you can put the **public key** in your ~/.authorized\_keys file
- Open the remote location in VS Code

- You should be prompted in the bottom left corner to reopen the project in a container

## On a remote machine

- Make sure docker is installed and the docker plugin is installed in VS Code
- Open the folder in VS Code
- You should be prompted in the bottom left corner to reopen the project in a container

# Starting a new project

To start a new project, you will need to make a template repository with the Hello World Microservice repository. This can easily be done with Gitea by adding

[kheestand/HelloWorldMicroservice](#) as a template repository when creating the repository.

Once you have the HelloWorldMicroservice added as a base repository, you will need to modify the .env file located at /workspace/.openapi/.env. Your .env file should look something like this:

```
{
  "schemaLink" : "<link to openapi schema>"
}
```

**Note:** you will need a link to a valid openapi schema. You can use the [pet store api](#) if you just want to test these instructions.

Once your .env file is ready, you can select the **Start new Project** task in VS code by pressing Ctrl + Shift + P on your keyboard and selecting *Tasks: Run Tasks* and then *Start New Project*.

When the project is generated you should also edit the name of the devcontainer environment to something other than "Microservice v1.0.0.0". This can be done by navigating to the /workspace/.devcontainer/devcontainer.json file and editing the "name" field at the top of the file.

## Updating the template repository

In VS code you can use the following commands to update the template

```
git fetch --all
git merge template/master
```

After running git fetch you might need to commit the changes to finish the merge. This can be done through VS Code's Source Control menu. You also might need to allow unrelated histories which you can do by modifying the git merge command to

# Viewing Schema in SwaggerUI

The [OpenAPI Preview](#) extension should be installed when the devcontainer is set up. If it is, go to the directory `/workspace/server/openapi_server/openapi/openapi.yaml` then press `Ctrl + Shift + P`. Then select or type in "OpenAPI Preview" from the menu

## Run, Test, Build and Deploy

### Running the Code for Debugging

Code can be ran in the following ways:

- Press F5
- Go to the Run and Debug menu (`Ctrl + Shift + D`) and select "Launch OpenAPI2SpringBoot"

### Running Tests

In progress...

### Building

Jenkins will be used for building all microservices. Each repo will have a dockerfile located in the root directory that will tell the system how to build the image. An example can be found on the HelloMicroservice gitea repo. A sample keystore.jks file is included in the `.crypto` folder and is used to sign the jar during development. The password for the keystore is `test123`

Note that microservices that are built in production will be signed using the jarsigner utility with the production keystore.

### Deploying

In progress...

## What's included in the Dev Container?

### Docker-compose environment

#### MariaDB

Default username `root` and password `mariadb`

[Docker Hub Link](#)

# VS Code Extensions

VSCode-MySQL-Client

[VS Code Store Link](#)

OpenApi-Preview

[VS Code Store Link](#)

---

Revision #10

Created 2022-02-02 22:54:56 UTC by Kyle Heestand

Updated 2022-05-10 01:32:19 UTC by Kyle Heestand