COMP2014 Distributed Systems Lab Options, 2022/2023

2023-01-25, v1.1

1 Introduction

The labs involve running and experimenting with a number of networked applications. These are generally written in Java, as everyone should be familiar with that from previous modules. This first "lab" is just to make sure that everyone has a consistent development environment for the later exercises.

The supported development environment for the labs is:

- Visual Studio Code (for browsing source),
- git (for obtaining source and saving any changes) and
- Docker containers (for building and running in).

The recommended approach is to install and run VSCode and Docker Desktop on your own laptop. This should work with Windows, MacOS and Linux. There are a few other options or variants:

- If you cannot install/run VSCode on your own machine then you can use VSCode on the Computer Science Windows Virtual Desktop or the desktop PCs in A32 and get a temporary Azure Labs Virtual Machine to run Docker on.
- If you are already using a hypervisor on your laptop (esp. VirtualBox) then you may find some compatibility issues between this and Docker Desktop, and it may be safer to run docker in a local VM.
- If you cannot install/run Docker on your own machine then you can also get a temporary Azure Labs Virtual Machine to run Docker on.

Note: if you are using your own machine you may find it quicker to install the required software and download the docker images (and VM images if required) outside the lab, i.e. avoiding eduroam.

2 Gitlab access

Resources for labs are provided via projects.cs.nott.ac.uk. E.g.,
https://projects.cs.nott.ac.uk/comp2014/comp2014-2022/comp2014-2022-lab0

Please contact RST if you are unable to authenticate with projects.cs.nott.ac.uk at all.

If you can log in but don't have access to these repositories then email the module convenor to request access.

Note, you will need an access token (e.g. personal access token with at least read repository permissions) set up in gitlab in order to check out the repo over HTTPS.

3 Option 1: Use your own Machine

Note: you may find it quicker to install the required software and download the docker images (and VM images if required) outside the lab.

3.1 Visual Studio Code

Please install Visual Studio Code, https://code.visualstudio.com/download

- Please also install the Remote Development extension, https://code.visualstudio.com/docs/remote/remote-overview

3.2 Docker Desktop

This is the recommended option, unless you are already using VirtualBox (especially if it is version 6 or older) on your computer, in which case you should consider setting up a VM running Docker and/or checking carefully for potential compatibility issues between the VirtualBox and Docker.

Please install Docker Desktop, https://docs.docker.com/desktop/

3.3 Docker in Virtual Machine

Consider installing and using Vagrant first to create new desktop virtual machines, https://developer.hashicorp.com/vagrant/downloads

If you want to use Docker in a Virtual Machine then create a new VM, e.g. Ubuntu 20.04 LTS, and install Docker Engine, https://docs.docker.com/engine/install/ubuntu/

> Note, you will need to add the user account you are using to the docker group so that it can run docker without sudo (see installation docs, post-installation steps). You also need to ensure that the SSH server has TCP Port Forwarding enabled (usually in /etc/ssh/sshd_config).

See also https://projects.cs.nott.ac.uk/comp2014/comp2014-2022/comp2014-2022-lab0#vmdocker-setup for more information.

3.4 Docker images

These images will be automatically downloaded when they are needed, but if you are using your own machine or VM then you may save time by downloading them before the lab. Run these at a terminal, if using Docker Desktop then on that desktop, or if using a VM then on that VM:

If using docker directly:

```
docker pull gradle:7-jdk17-jammy
```

If using dev containers:

```
docker pull mcr.microsoft.com/devcontainers/java:0-17
```

Optionally, for one of the later labs:

```
docker pull redis:7
```

4 Option 2: Use an Azure Labs VM

4.1 Docker in Azure Lab VM

Please complete this form in order to request an Azure Labs VM for the COMP2014 lab sessions,

https://forms.office.com/e/6Q8zmz3z6V

The VM is Ubuntu 20.04 and has Docker engine pre-installed. It can only accessed via SSH (i.e. there is no graphical desktop).

You will be sent an invite email which with a link to take you to Azure labs. When you first start the VM and try to connect to it you will have to set a new password for the VM, which can take a few minutes.

You will need to start the VM each time you want to use it, and it will turn off automatically if it is unused for 15 minutes (5 minutes after disconnecting) if you forget to turn it off. The VM has a quota on the number of hours it can be run; you will need to contact the module convener if you need this to be increased.

It is access via a non-standard port which **may** be blocked from the campus network (lab PCs or eduroam) or company networks. I've requested a firewall exception but… If you have problems with connection timeout WHEN THE VM IS DEFINITELY RUNNING then try accessing it from the Computer Science Windows Virtual Desktop.

4.2 Visual Studio Code

You can use Visual Studio Code:

- On your own computer (see 3.1; may have firewall issues on campus),
- On the Windows Virtual Desktop (currently **NOT Dev Containers** - Remote-SSH/docker only), or
- On the Computer Science desktop PCs, e.g. in A32 (currently **NOT Dev Containers** - Remote-SSH/docker only, may also have firewall issues)

Note, as mentioned above the Azure Lab VM non-standard SSH port **may** be blocked from the campus network (lab PCs or eduroam) or company networks. I've requested a firewall exception but… In this case try accessing from the Computer Science Windows Virtual Desktop.

Note, the version of VSCode on the lab PCs and the Windows Virtual Desktop is currently a few months old, and doesn't support a recent enough version of the VSCode Dev Containers extension to work, so you will need to use Remote-SSH access and Docker only (not Visual Studio Code Dev Containers).

5 Other options

You can just use ssh/vi/docker directly if you like :-)

You can use other IDEs or editors, but they are unsupported, and will probably have different or more limited support for remote or containerized development.

As an alternative to the Azure Lab VM you could use your own cloud-hosted VM, provided it is accessible over SSH and has a reasonably recent version of docker installed (see 3.3). E.g.

- https://azure.microsoft.com/en-gb/free/ or https://azure.microsoft.com/en-gb/free/students/

- https://cloud.google.com/free
- https://aws.amazon.com/free/