docker run –d –p 8080:5000 –v $(pwd)/data:/data --name container1 lab4

docker exec –it A637jd9dja /bin/bash

docker create –v /data --name datastore1 hello-world /bin/true”

Discuss any other alternative to Docker containers.

There are several commands that can be run in the Docker environment which will allow us to do certain things to images or containers. The three commands above commands Docker run and Docker exec deal with the containers and Docker create deals with both the image file and container.

“Docker run” allows us to run a pre-built container (built using “Docker build”). Many different arguments can also be passed into the command on execution to alter how the container executes. In this example we have a few different examples of arguments that have been passed into the command. “–D” means run as a daemon or background image so we can avoid having to open up a new terminal screen to execute more commands. –P specifies a port number or range, in this case we have 8080:5000, but it can be any port number(s) that is / are available. –V mounts a data directory to the container so that things such as files within the container. Along with the –V tag, we must also pass a directory “$(pwd)/data:/data” which basically means a data folder in our current working directory. The final arguments are --name and lab4. The name tab is self-explanatory, it just assigns a name to the container (container 1 in this case) and lab4 is just the name of the image to you wish to run the container off.

“Docker exec” allows us to run a command in an already running container. –I and –T arguments can work hand in hand and are very useful when used together. –I informs the container to keep the STDIN open, even if it is not running so there is no unnecessary crashing / malfunctioning of the container. –T tells the container to open up a pseudo-TTY or a virtual terminal within the container to execute any commands that will be passed in by the user. The user can do things such as navigate directories, create and delete files etc. whilst inside the containers virtual terminal. The last two arguments are used to specify a container name / ID that you wish to execute your commands in and the location in which you wished to be placed inside the container.

“Docker create” allows you to create a brand new container from an existing image such as the default “Hello world” image that Docker provides. –V again allows us to mount a volume or folder to the container if we wish to persist and access files that will be generated or stored within it. --Name allows us to specify a name for the container (in this case we have chosen “datastore1”). Hello world specifies the image the container will be built / created from. In this case there is a “Hello world” image created by Docker that will return some simple facts about Docker when run. The “/bin/bash” argument is used as an entry point into the container so that it can be executed. If we were creating a data only container and left this argument out, would not be able to use “Docker run” to execute / run this container. Specifying “bin/bash” gives our container and entry point or somewhere to execute from in case we need to run the container in the future.

There are 3 main alternatives to Docker. They are Rocket, Drawbridge and LXD. Each platform functions slightly differently to each other as well as Docket and they each have their own pros and cons.

Rocket was introduced to deal with libcontainers as opposed to entire applications. Libcontainers allow for greater flexibility and portability due to the fact that there is more customisation and less of the application to deal with as opposed to Docker. Rocket is often seen as a better alternative for bigger companies who rely on devops for these reasons.

One of the big problems with Docker is that it is Unix based and this possess a big problem for companies using the Windows Azure platform. The counteract the problem, Microsoft have released their own Docker equivalent in Drawbridge which provides similar functionality for windows systems. It operates by using virtual machines to run the native applications.

If you hate your containers sharing processes, then LXD is for you. LXD was released to solve this problem as well as provide other features such as native support for user namespaces. This allows non privileged users to do things like run containers without having to worry about security issues and user-privileges.