What is Docker?

* Docker is nutshell, It is a software containerization platform it provide abstraction of operating-system level virtualization
* Docker in high level – these container contain the real
  + Operating System
  + Software that you build
  + Dependencies to run the software’s (like –pre –requisite softwares)
  + Environment variables
  + And you name it

How Docker achieved containers?

* It was made possible with the help of LXCs(Linux container)
* LXCs are user space interface for the Linux kernel containment which make it possible to run multiple isolated Linux containers ,on one control host (the LXC host)
* Linux Containers serve as a lightweight alternative to VMs as they don’t require the hypervisor like
  + Virtual box
  + KVM
  + en etc.

As that said, Docker now our solution to Runs in your machine but not in mine

That’s why Docker call themselves

Build 🡪SHIP🡪 RUN

What is the Roadmap of this course?

1. Understand the basic of Docker

2. Use Docker for selenium automation testing multiple containers (machine)

3. Create our very own container for Automation /Development

**Part 2: Understanding and working with Containers**

**What is Container::**

* They are an isolated, resource controlled and portable operating environment
* Basically, a container is an isolated place where an application can run without affecting the rest of the system and without the system affecting the application
* Containers are the next evolution in virtualization
* If you were inside a container ,it would look very much like you were inside a freshly installed physical computer or a virtual machine

**Where can I see the Containers?**

* All the different container are sitting in <https://hub.docker.com/explore/>
* <https://hub.docker.com/r/robcherry/docker-selenium/>
* Is there any other way to browser containers?
* Yes ,you have something called **Kitematic**

**So what these containers have?**

**As said earlier, Containers contains everything you require for your software to run like**

**What if I change something in my software and dependencies?**

**Versioning:**

* Docker includes git –like capabilities for tracking successive versions of a container, inspecting the diff between versions, committing new versions, rolling back etc.
* The history also includes how a container was assembled and by whom ,so you get full traceability from the production server all the way bacvk to the upstream developer.
* Docker also implements incremental uploads and downloads similar to git pull,so new versions of a container can be transferred by only sending diffs

**Component re-uses**

* Any Container can be used as a “base image” to create more specialized components
* This can be done manually or as part of an automated build .for example you can prepare the ideal Python environment, and use it as a base for 10 different applications
* Your ideal Postgresql setup can be re-used for all your future projects .And so on.

Part 3 –Getting Stared with Docker

Understanding docker ecosystem

* Docker has wide ecosystem and it(docker engine) supports multiple operating system ,cloud platforms even local clouds
* As that said, Docker supports mac ,window and linux

Different tools from docker

There are different tools docker provides like

* Native support docker engine for Mac/Window/Linux
* Docker hub
* Docker cloud
* Docker Compose
* Docker machine (will be deprecated soon,but is mandatory component for older OS)

Docker for Windows

Docker is free , hence you can download docker from

<https://download.docker.com/win/stable/InstallDocker.msi>

Pre-requisite for docker in windows

* Docker requires windows 10 pro 64-bit edition(with latest update)
* Note Docker for windows require Microsoft Hyder-V to run
* After Hyder-V is enabled,VirtualBox will no longer work, but any Virtual Box VM images

Part 4- Working with Docker and Containers