The Best place to start your container journey is right here, right now on your Desktop PC

June 6th 2019

Give it A shot…

Shanmuga SVR

|  |  |  |
| --- | --- | --- |
|  |  |  |
| [LinkedIn] : ssvr1  [Twitter] : ssvrcool | [GitHub] : https://github.com/ssvr30 | [Email] : ssvr30@gmail.com |
|  |  |  |

Contents

[Introduction 1](#_Toc21552072)

[Containers? 2](#_Toc21552073)

[MyPC Desktop + Ubuntu + Docker + Internet 3](#_Toc21552074)

[Plethora of capabilities on your PC 4](#_Toc21552075)

[Quick Start: 6](#_Toc21552076)

[Training and advanced learning 7](#_Toc21552077)

[Conclusion 8](#_Toc21552078)

# Introduction

"I think it's fair to say that personal computers have become the most empowering tool we've ever created. They're tools of communication, they're tools of creativity, and they can be shaped by their user."

(In a [speech at University of Illinois Urbana-Champaign](http://www.microsoft.com/billgates/speeches/2004/02-24UnivIllinois.asp), Feb. 24, 2004) ~ Bill Gates.

Never did we imagine that our Personal computer (PC) Desktops or Laptops used primarily for tasks which are pretty minimalistic in nature can one day transform to run production grade hosting platform for application. That is what software containers can offer right out of your desktop.

Indeed, personal computers are empowered with the emergence of software containers, and their technology redefines the potential of our personal devices, a transformation in itself.

All you need is a PC running an operating system (Ubuntu or Windows 10) and a working internet connection.

To test container technology the easiest and best way is to use Docker CE (Community Edition). Docker CE gives us an insight on all the features of modern software containers.

# Containers?

A standardized unit of software

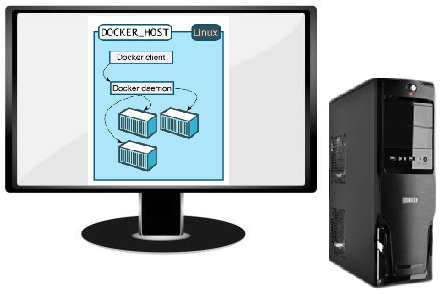
[Docker](https://www.docker.com/resources/what-container)

**?** is the expression I would get when I started introducing this technology a couple of years ago. But the shift from **? >> !** was rather quick.

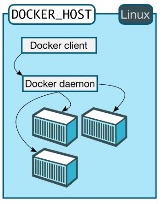
The initial transformation was confined to a limited set of enthusiasts and it took some time to bring in the change of mindset about this technology.

Now the times have changed, what were small groups have become large innovators using this technology to transform modern computing.

# MyPC Desktop + Ubuntu + Docker + Internet





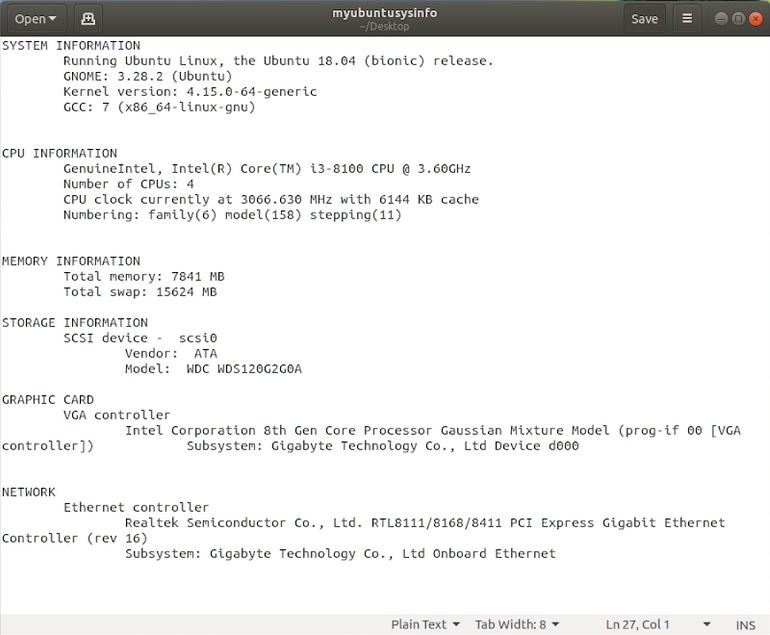


You can pretty much get all you need to know about containers with this combination.

Assessment on the capacity of the resources and the load of the application has to be carried out to make sure the optimal performance criteria are achieved. If you still have the previous generation Pentiums and AMD’s they can still be put to good use.

Let us look at my system configuration:

All my learning and experimentation is carried out on this configuration. Docker is not the only workload that my system is subjected to and with other applications running in parallel, I still get the best out of my system.



# Plethora of capabilities on your PC

Lightweight

Packaged

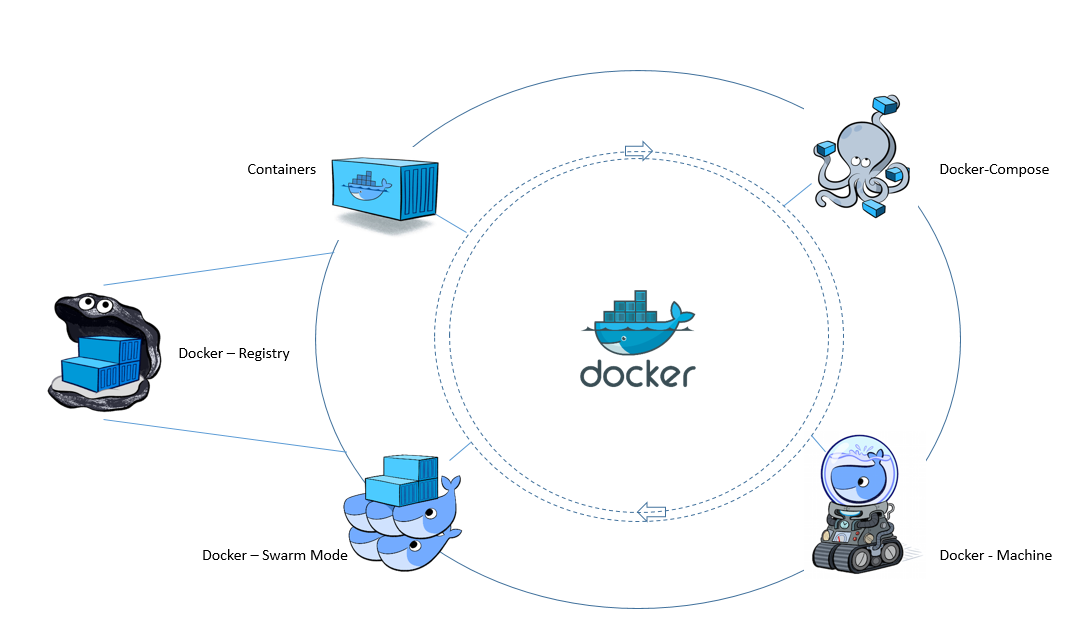
Includes everything

Quick

Heterogeneous

Reliable

Secure

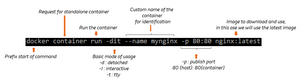


Docker install provides the complete toolchain. On Windows 10, these come bundled together. The Linux install varies to a degree, where packages like Docker-Compose and Docker-Machine has to be installed separately.

The variations between Linux and Windows differs at the install alone, but the capabilities is same irrespective of underlying OS.

In its simplest form a standalone ***container*** (A running image) can be started using the Docker CLI integrated with Linux terminal or windows cmd/powershell talking to the Docker engine/daemon.

A command to spin-up a container which runs an nginx web server is as shown below.



**Docker-Compose** makes it easy by eliminating long commands, housing them in a network, choosing a specific image and adding volume all within the defined declarative YAML file.

Portable

Isolation

Heterogeneous

Reliable

Secure

Docker compose should be the first option to choose when you have a need to run many loosely coupled images, which may need persistent storage, confined network and specific known parameters.

Docker-compose can scale all the way from DEV to UAT and set context on how the final production would look like. It is the sword of choice when it comes to getting your idea visualized and demonstrated.

When you have a need to test multi node capabilities, then **Docker-Machine** is what you need.

**Docker-Machine** lets you place images on different virtual hosts and manage them. We can choose and deploy which node within a virtual host need to run what image. Docker-Machine simulates distributed computing well within the perimeter of your desktop.

The capabilities and control on what we can do with Docker widens as we start to use the orchestration feature called **Swarm mode**. Swarm mode evolved significantly to deliver all the features and richness expected out of container orchestration.

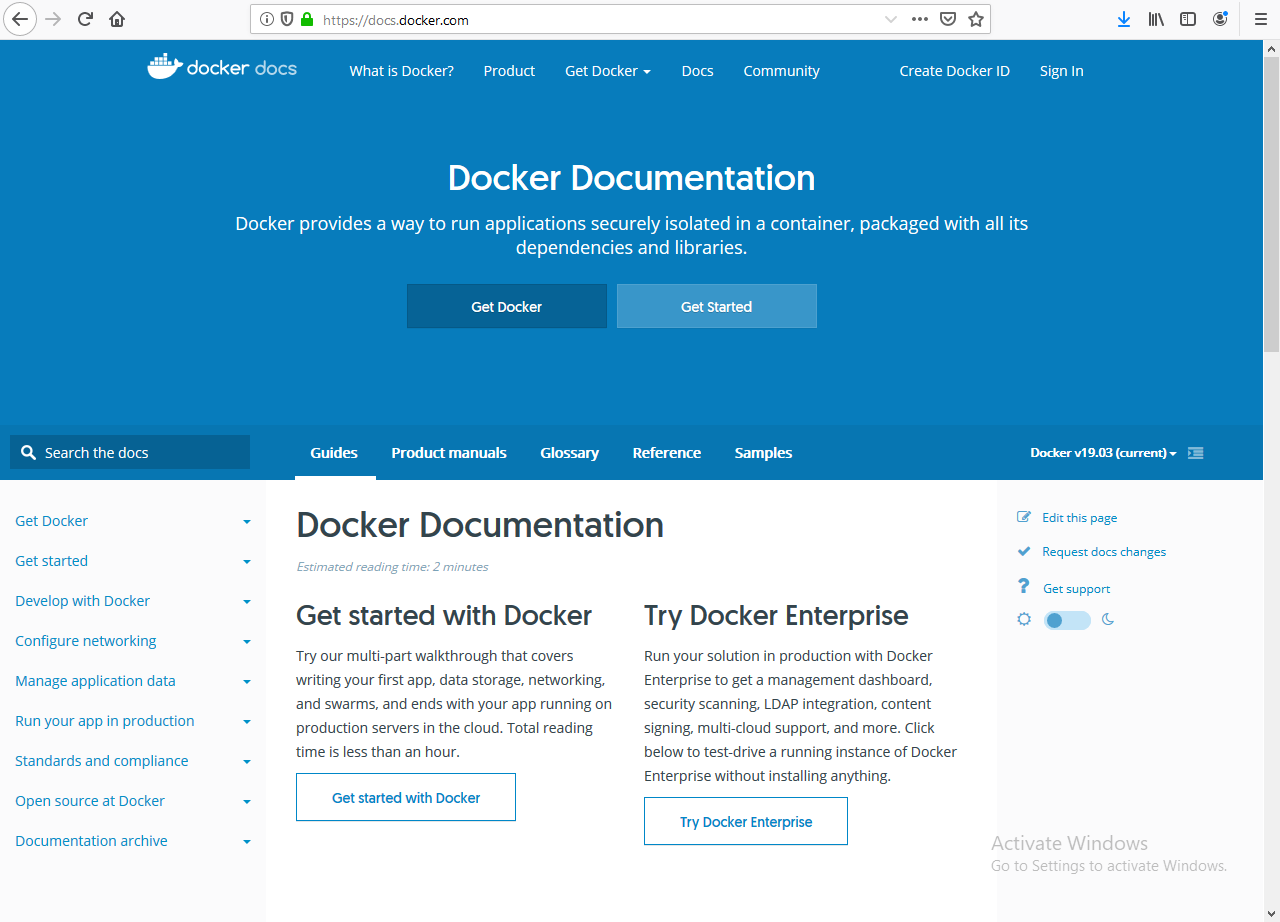
Swarm mode’s potential can be perceived better, if containers are running across multiple networks and distributed across multiple providers, be it hybrid or public cloud. But to learn as to what Swarm mode can offer using Docker-Machine + Stack (A feature in Swarm mode to deploy containers) can give an understanding on how enterprise grade deployments of containers are used.

**Registry** is a central repository, a single source and a go to for all your Docker images. Docker’s registry is called [Hub.Docker.com](https://hub.docker.com/). Registry is a treasury of images ranging from official to community distributions contributed by millions of enthusiasts.

When the need for Enterprise specific registry arises, setting up one is again a simple process and with the right hardware, we can direct all users to our very own custom build registry.

All these features are available to test on your desktop and that is how I have learnt to use them.

# Quick Start:



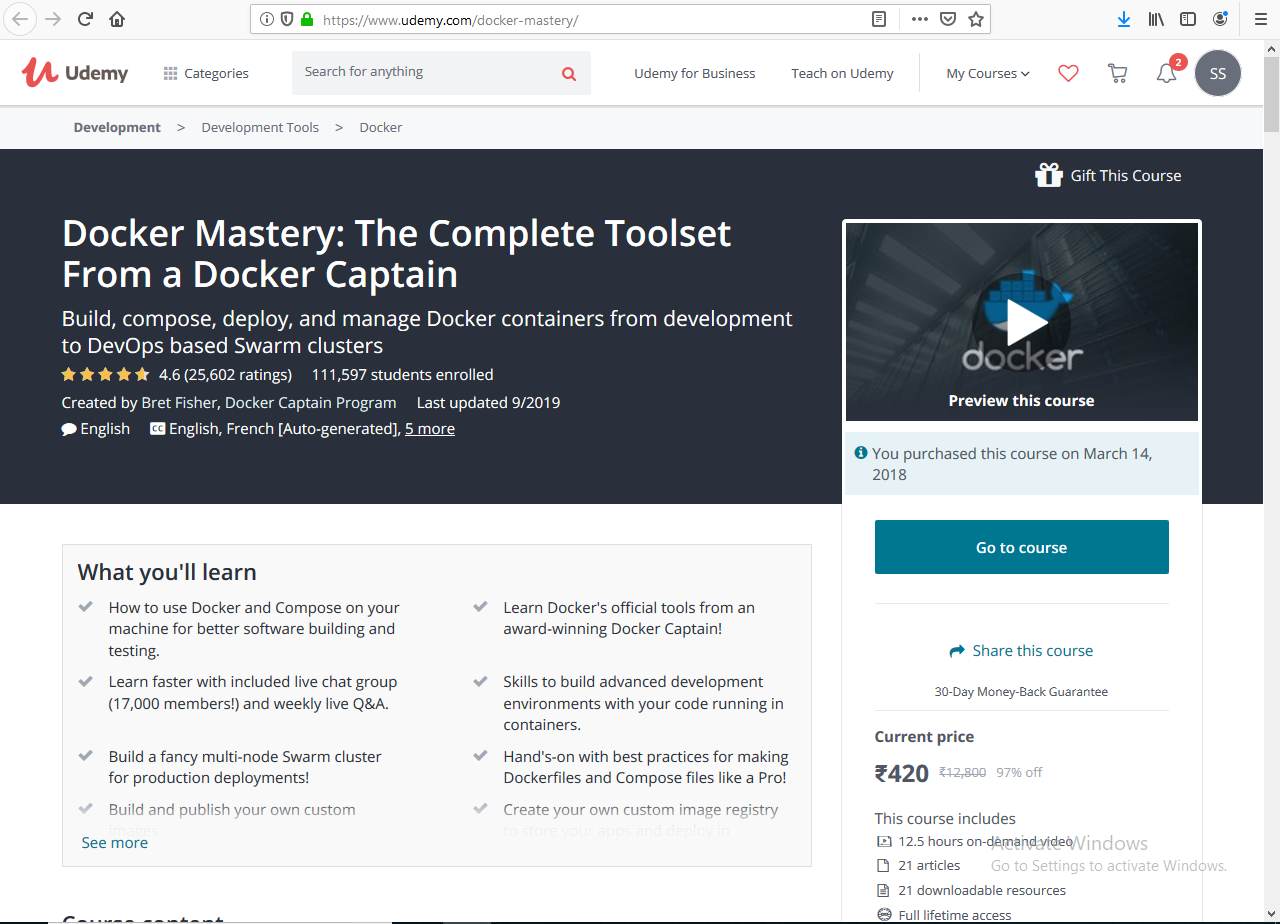
Docker [documentation](https://docs.docker.com/) is exhaustive and should be a reference guide in your Docker container journey.

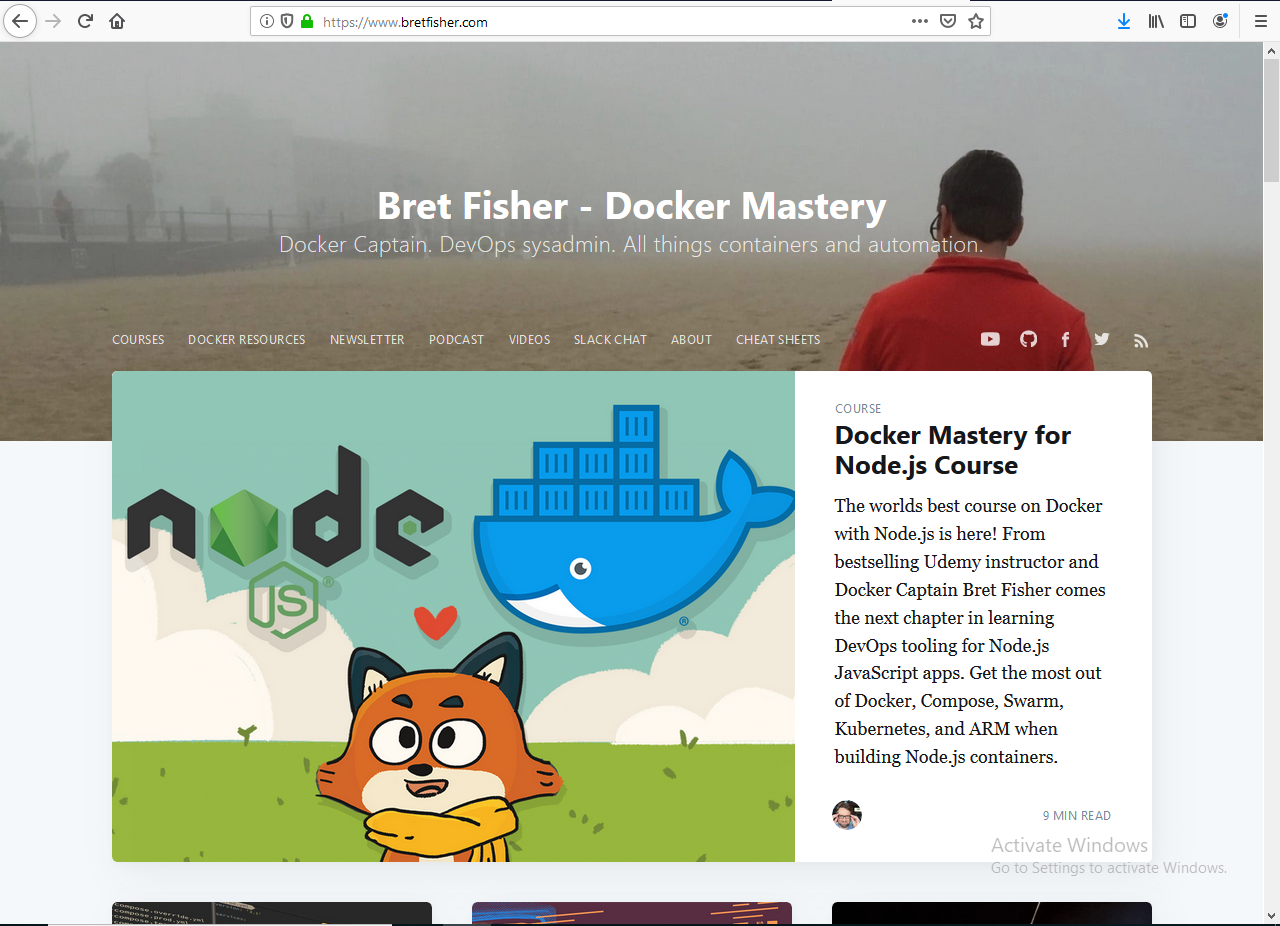
If your choice is to use Linux (especially Ubuntu), then the installation is the most easiest and simple of all.

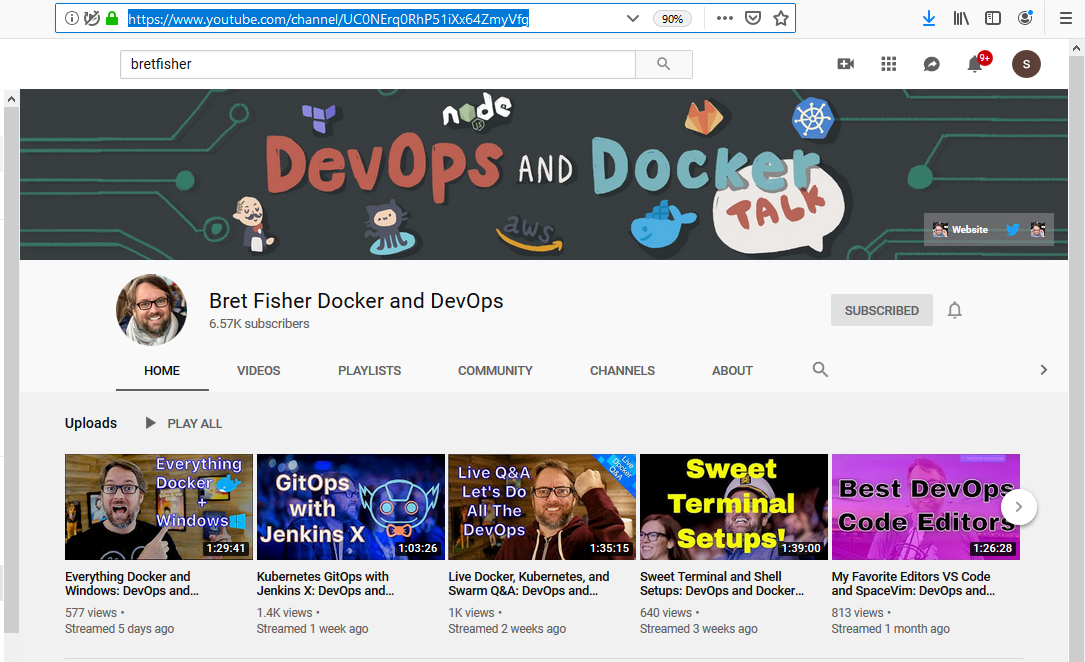
Visit à <https://get.docker.com/> , follow the instructions which are just two steps, add the user to the Docker group and boom you are set to use.

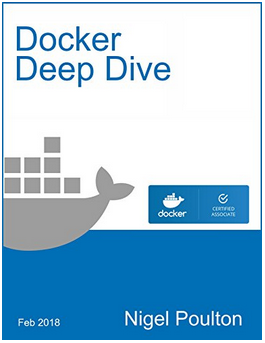
This though would not get you the complete toolchain, but will get you to start spinning up containers of choice immediately.

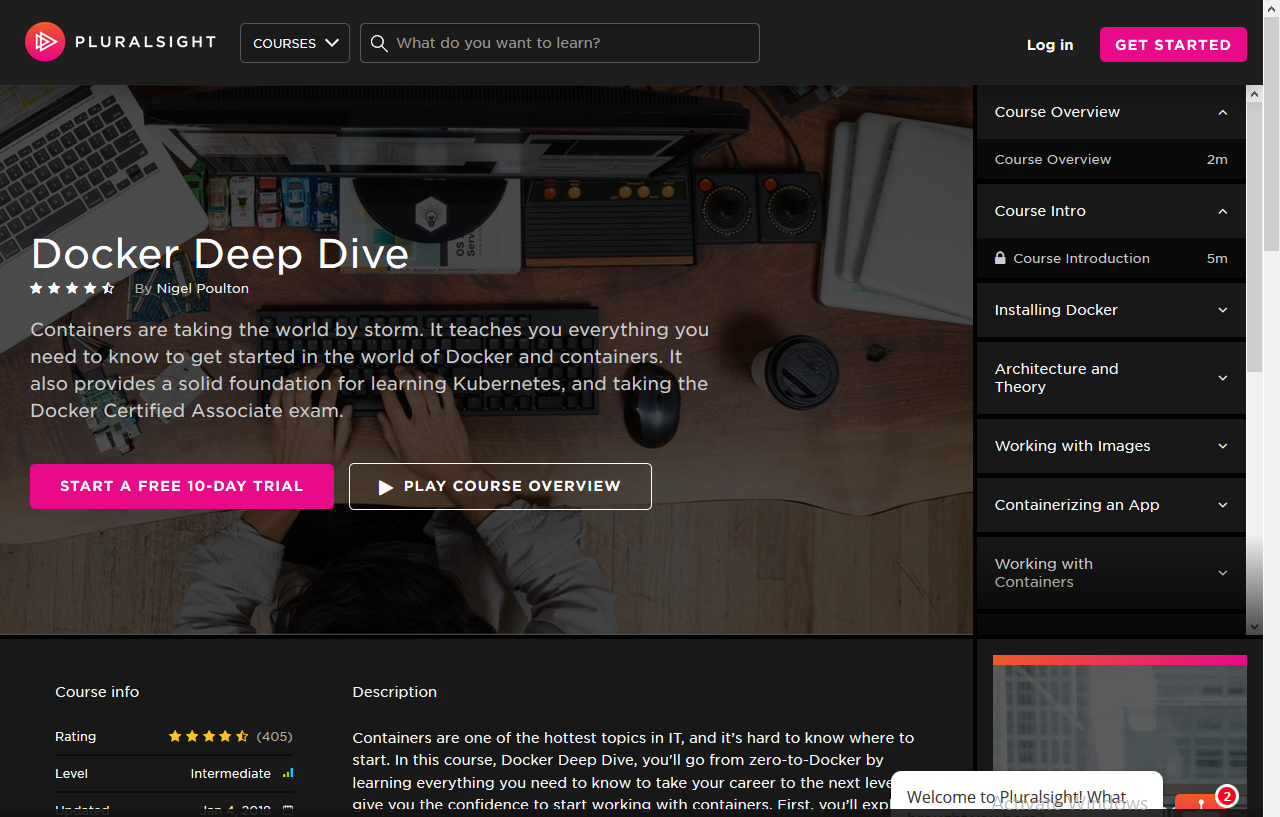
# Training and advanced learning











Even though there are many courses on the web, the best that I started with is a course on Udemy called ([docker-mastery](https://www.udemy.com/course/docker-mastery/)) offered by **Bret Fisher**.

His curriculum is simple, example driven and one of the courses I enjoyed completing in a stretch.

Learning from a [Docker Captain](https://www.docker.com/community/captains) who is best at what he does is an opportunity one should never miss.

Almost 70% into the course, I started experimenting and creating production scenarios and real life feasibility of Docker. His course is that powerful.

To explore other offerings from Bret Fisher is to start with his [website](https://www.bretfisher.com/), alternatively his YouTube channel and podcast are a regular dose of Docker and associated technologies.

If you are planning to go further and master Docker and container technologies, starting with this course will set you on the path towards that direction.

Another course which can get you to prepare for Docker certification is from Nigel Poulton (also a Docker captain) called [Docker Deep Dive](https://www.pluralsight.com/courses/docker-deep-dive-update) on Pluralsight. This program was designed to give all the required deep dive and understanding on inner working of Docker and its premium enterprise edition and security concepts.

All the Docker captains and enthusiasts are active in technology forums, social media and video streaming channels like YouTube. Tag them and get a constant feed of knowledge delivered wherever you are.

# Conclusion

Containers as a concept is much older than we could realize. Work and advancements in this field was constant and Docker’s adaption from this learning is immense and ground breaking.

**Learnuation**

**Contributeuation**

**Shareuation**

Docker is all that is needed to understand what containers are. The best and simplest way to learn and explore is **right here, right now on your Desktop PC**.

**Make a better place**

Technology at its best…

Let’s Docker..!!!!