Automating Dev Ops with Docker Application Technology Shell Scripts

Franklin Smith
University of Oregon
fsmith4@uoregon.edu

ABSTRACT

With an emerging rise of Dev Ops technology like Docker and other application containers comes an underlying challenge that has been plaguing the computer industry for years, how to efficiently learn and use the technology in a timely manner. Most users are tired of long and meaningless online tutorials and videos which shove irrelevant information down the throat of the consumer. I have solved this problem by programming a shell script that automates the dev ops process with docker while allowing the user to interact and choose where, what, and how they would like to learn about the technology. With a computer execution run time of 2-3 minutes, one can now learn to: set up their docker environment; build an image and run as one container; scale their application to run multiple containers; distribute their application across a cluster; stack their services by adding a back end database; and deploy their application to production.

KEYWORDS

Docker, Dev Ops, Software Engineering, Container, Cluster, Image, Server, Shell Script, Application, Technology

1. INTRODUCTION

Docker is a platform for engineers, developers, and sys admins to create, develop, run and deploy applications with containers. The use of linux containers has risen in popularity. Using them to launch applications is called containerization. They are growing in use and popularity due to the fact that this new technology makes it possible to get more applications running on the same old servers. It also is easy to ship and package Programs.

Dev Ops is a fairly new term that is defined by the "practice of operations and development engineers working together in the entire software service Lifecycle from design through development process to production support" [1]. The new age of this software development life cycle and dev ops has sparked an IT talent war, converting many of the old sys admins into seasoned new leaders of dev ops teams.

Docker as a company was founded in 2010 and was originally called dot cloud. The creator is YCombinator alumni Solomon Hykes. The company received a 1.3-Billion-dollar valuation last year and has received 272.9 million dollars in 9 different investment rounds over the course of 7 years.

Some big-name companies that are using docker include PayPal, eBay, the Washington post, GE, Groupon, uber and Shopify. There appears to be an increase in rising market share from incoming competitors. While docker currently holds 48 percent of the application container market share, rising competitors include: Kubernetes (founded by google) who currently hold 29 percent of the application container market share; Vagrant who currently hold 5.9 percent of the application container market share; Apache Mesos who currently hold 5.59 of the application container market share; Rancher OS who currently hold 3.25 percent of the application container market share; and Red hat open shift who currently hold 2.5 percent of the application container market share.

1.2.THE PROBLEM

The problem that currently exists is that for new and previous technology like docker and other dev ops distribution services, the tutorials to learn these tools and technologies are extremely difficult and painful especially when you are trying to learn fast. Many software engineers find them: time consuming; annoying; and also find it hard to remember specific commands. In addition to the fact that many tutorials do not include all of the resources, libraries, dependencies, and files needed or show how it is required to set up any of the above.

This has been a stemming issue from the computer science community and software engineering community for many years. Samir Gosh, CEO of wavemaker says "Implementing docker is complicated. There are a lot of supporting technologies needed for things like container management, orchestration, app stack packaging, intercontainer networking, data snapshots and so on." [1] This excerpt proved relevant due to the fact that an established tech CEO goes into detail about the needed improvements for the docker technology and helpful tutorials. These improvements reflect key topics and fundamentals which are covered in the shell script that I created. A partial but not complete response to this problem has been the emergence of community forums and websites to providence guidance and assistance with troubleshooting technology and software. Some of these forums and websites include but are not limited to YouTube, stack overflow, reddit, Quora, hacker news, lecture hunt, and google.

1.3 WHY THE PROBLEM IS IMPORTANT

This problem establishes a high priority of importance because for any creator and user of the technology, one cannot learn and use it efficiently in an appropriate and timely manner. This includes but is not limited to; students in the computer science or a technology field; Professors of computer science using the technology to teach, mentor, or conduct research; Software engineers at a software company that needs to build, ship, release, or update a new or existing product; Entrepreneurs who wish to leverage this technology to build a business system or software platform; or anyone else that wants to use the technology. Docker as a company has tutorials hosted on their website. All of which cannot be learned or used in an appropriate and timely manner.

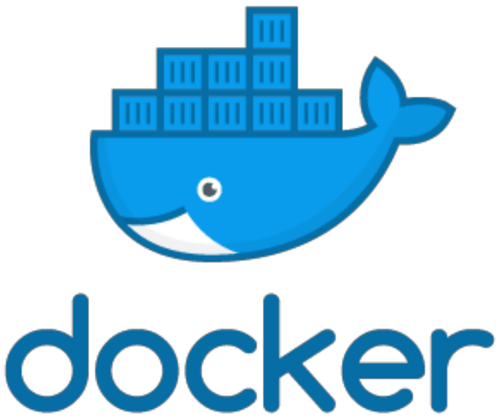


Figure 1: The notorious docker company whale logo

At the end of one of the specific tutorials on the docker website, there is text that reads "And that's a wrap! After a long, exhaustive but fun tutorial you are now ready to take the world by storm! If you followed along until the end, then you should definitely be proud of yourself." [2] This is a prime example of how useless, time consuming, and uninteresting the online docker technology tutorials are.

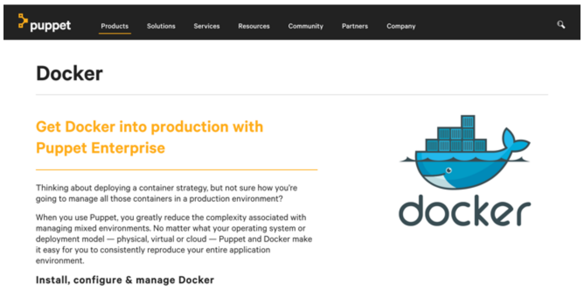
1.4 WHAT IS THE SOLUTION

My solution to this problem was programming a shell script that automates the basic on boarding tutorial commands with minimal user input and setup. The solution solves the problem because the user can execute the shell script from their local machine in an efficient and timely manner, following this step the shell script does all of the difficult and time-consuming work and terminal executions for the user.

1.5 WHY THE SOLUTION IS NOVEL

The solution is novel because executing the shell script is not time consuming or annoying and this technology allows the user to allocate their time where it’s specifically more needed, such as modifying the shell script to tailor to the specific project needs or learning on the go. Another reason the script is helpful is because the user can view the code in the shell script file and identify the most important commands that are executed in addition to quickly viewing a single-line comment explanation of the command. Thus, the user does not have to waste time reading paragraphs and bible like pages of irrelevant information after executing one single command.

In addition to the reasons stated in the above paragraph, the solution is also novel because there has recently been a rise of automation tools that automate the docker technology. For example, the technology company named Puppet which is located in my hometown of Portland Oregon.



Docker

Get Docker into production with Puppet Enterprise

Thinking about deploying a container strategy, but not sure how you're going to manage all those containers in a production environment?

When you use Puppet, you greatly reduce the complexity associated with managing mixed environments. No matter what your operating system or deployment model — physical, virtual or cloud — Puppet and Docker make it easy for you to consistently reproduce your entire application environment.

Install, configure & manage Docker

Figure 2: The website for the Docker Image Application Module provided by the Portland technology company Puppet.

They sell software that includes a docker module that installs, configures, manages the docker application in addition to the hosts and services running on it. [3]

The shell script that I have created benefits the user in a way that allows them to learn what is going on and provide interactivity in the terminal. The script also eliminates the need and unwanted problem of dealing with the terrible experience of nested layers upon of technology and software to execute one simple and fundamental task, which is learning how to use docker.

1.6 KEY RESULTS AND IMPLICATION OF THIS WORK

The key results from this work, was the creation of a shell script that is executed in the user’s terminal on their local machine. Upon creating a docker account, and simply installing the docker application from the website to the users local machine, the user will then execute the shell script which sets up the docker environment; builds an image and runs that image as one container; scales the users application to run on multiple containers; distributes the users application along the cluster; stacks the services by adding a back end database; and deploys the application to production.

1.7 FUTURE WORK

Future work for this project includes the ideology that with this new technology or dev ops services the creators and authors of the technology can create a shell script in addition to the included tutorial. This would be effective due that to the fact that the simple tutorial and shell script would allow the user to quickly execute and understand through trial and error, rather than having to read through a bible like online tutorial with disproportionate text to executable command ratios also including a wealth of irrelevant information. With this technology, users can modify the script and input to tailor needs towards their specific project or technology they intend to deploy. This is useful because users no longer have to watch time consuming YouTube tutorials.

There may soon be an emerging rise of automated on-board technology application processes in the fields of businesses and software. I feel this way because in the knowledge work age humans become easily distracted and have short attention spans. Thus, the company or technology that can familiarize their technology or product with the user the fastest, simply wins. World renown legendary computer science professor and book author Cal Newport states in his recent publication titled “Deep Work” that “To remain valuable in our economy, therefore you must master the art of quickly learning complicated things. This task requires deep work. If you don’t cultivate this ability, you’re likely to fall behind as technology advances.” [6] This citation proves relevant and is strongly supportive to the thesis of this research because it acknowledges the transformative importance of learning new technologies quickly and leveraging the technology to perform at your highest level or deliver the best possible software product. Currently Docker and other new emerging technology tutorials prevent one from achieving this due to the fact that they are time consuming, difficult, and unnecessarily complex.

2. BACKGROUND

To execute the shell script for this project, you only need a internet connection and the docker application. The equipment I specifically used was a 2014 MacBook pro with a 2.5 Ghz Intel Core i7 processor with 16 GB ram.

2.1 WHAT OTHERS HAVE DONE FOR THIS

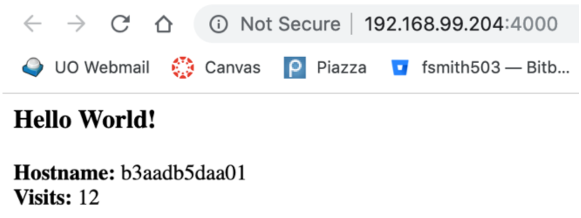
Some solutions that other technologists executed for this specific problem were using technologies like Chef and Puppet which automate the process. Chef is a framework that provides resources for installing docker as well as building, managing, and running docker containers. Puppet has a module available for download that installs, configures, and managers Docker. In addition to managing docker images and containers.

2.2 PROBLEMS AND SHORTCOMINGS OF THEIR WORK

The problem with these frameworks is that they also require tutorials to learn the technology at a slow and painful pace. This again sends the user into an experience of unwanted and revolting nested application technology tutorials.

2.3 HOW MY WORK IS DIFFERENT AND BETTER

My specific approach to this problem is different because there exists a single shell script that is only executed once. The script serves the purpose of executing the commands for the user while the user can read and learn the output of the script or read and modify the shell script themselves.



Hello World!

Hostname: b3aadb5daa01
Visits: 12

Figure 3: The python web application that is hosted on the virtual machine server at a web address. This is a hello world program which increments the number of visits every time a get request is sent to the docker server.

This is better because the user can build and learn rather than reading, watching, failing, all while navigating different bible-like tutorial webpages of the technology. For example, listed in an online review of a docker tutorial book for sale on the amazon website James Bridgewater states that “Very few practical examples of how to use Docker this book is basically just brief Docker documentation in e-book format. On the off chance you want Docker docs on your e-reader you might want this. If you are new to Docker look elsewhere.

If you like books where the author has some personality look elsewhere. The best thing about this book is the logo on the cover". [4] The citation is significant because it supports the fact and provides an example of how useless, time consuming, and uninteresting the online docker technology tutorials are.

3. PROJECT

The approach to this problem was programming a shell script that automates the basic on boarding tutorial commands with minimal user input and setup. I felt that this solved the problem because the user is able to get the application technology running quickly while having the option to choose what commands and parts of the framework, they would be interested in learning rather than iterating over every single irrelevant instruction.

3.1 DEISGN

The design I chose for this project was trivial and straight forward. The user downloads the required files and scripts in a folder from an online GitHub repository. Following this step, they then execute the shell script in the terminal with a single command.

3.2 WHAT I HAD

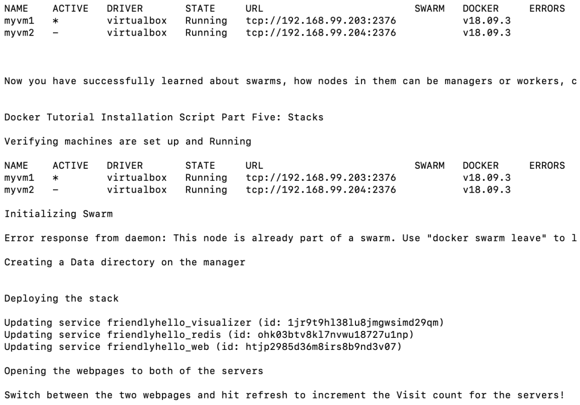
What I already had in place for the technology was little to no code. I took it upon myself to view a wealth of different docker tutorials and videos on the internet, then decipher which commands were actually relevant, necessary, and required to get the fundamental parts of the application technology up and running in an efficient manner. Most of the commands grew from trial and error in my own terminal of attempting to setup my docker application technology automation script. I had to test what worked and what didn't work many times.

3.3 WHAT I ADDED AND CHANGED

I added and changed commands to fit the specific information that is parsed from the user through the shell script. For example, this includes user name for the docker account, repository name for storing the docker image or application that the user is working on and intends to deploy on a server.

3.4 WHAT DID AND DID NOT WORK

Some things that did work were parsing input form the user and pushing and pulling the created images and applications in to the docker repository which is stored online. Some things that did not work and issues I ran into were configuring multiple virtual machines due to the IP addresses being different. I was able to solve this problem by prompting the user to input the IP address of the first virtual machine. After this the first virtual machine becomes the manager of the "swarm" (multiple machines) and the other virtual machine becomes the "worker".



```
NAME     ACTIVE   DRIVER       STATE     URL                          SWARM   DOCKER     ERRORS
myvm1    *        virtualbox   Running   tcp://192.168.99.203:2376            v18.09.3
myvm2    -        virtualbox   Running   tcp://192.168.99.204:2376            v18.09.3


Now you have successfully learned about swarms, how nodes in them can be managers or workers, c


Docker Tutorial Installation Script Part Five: Stacks

Verifying machines are set up and Running

NAME     ACTIVE   DRIVER       STATE     URL                          SWARM   DOCKER     ERRORS
myvm1    *        virtualbox   Running   tcp://192.168.99.203:2376            v18.09.3
myvm2    -        virtualbox   Running   tcp://192.168.99.204:2376            v18.09.3

Initializing Swarm

Error response from daemon: This node is already part of a swarm. Use "docker swarm leave" to l

Creating a Data directory on the manager


Deploying the stack

Updating service friendlyhello_visualizer (id: 1jr9t9hl38lu8jmgwsimd29qm)
Updating service friendlyhello_redis (id: ohk03btv8kl7nvwu18727u1np)
Updating service friendlyhello_web (id: htjp2985d36m8irs8b9nd3v07)

Opening the webpages to both of the servers

Switch between the two webpages and hit refresh to increment the Visit count for the servers!
```

Figure 4: Output from the executable shell script solution. At this point the script program has initialized two virtual machines on two different ip addresses, one is the "manager" and the other is a "worker" in the swarm.

Some interesting observations I made were that after the swarm configuration both of the virtual machines were hosting the same python application on their server. I gained knowledge of this information through discovery of the webpages because I could navigate between the multiple web pages of the application with different web addresses, but the website visit count which is displayed on the webpage was still the same, it increased by one every time a get request was made to access that webpage.

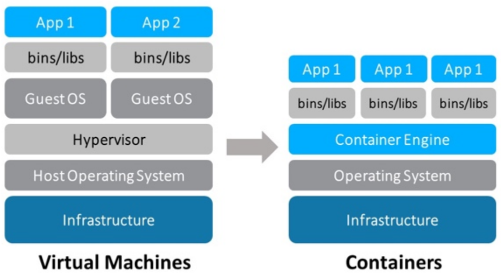


Figure 5: Graphical Explanation of the technical architecture of the newly emerging docker containers compared to the older Virtual Machines.

I also find it fascinating that the runtime for this script is around 2-3 minutes, which I consider to be efficient and a good use of the user's time.

4. RESULTS

The results of this work are that the user that intends to learn docker now has access to the shell script, necessary files, and a readme file that are all available to download from the online GitHub repository. In addition to this there now exists a thorough, relevant, docker tutorial that lists and teachers the relevant information about how to use the docker application technology.

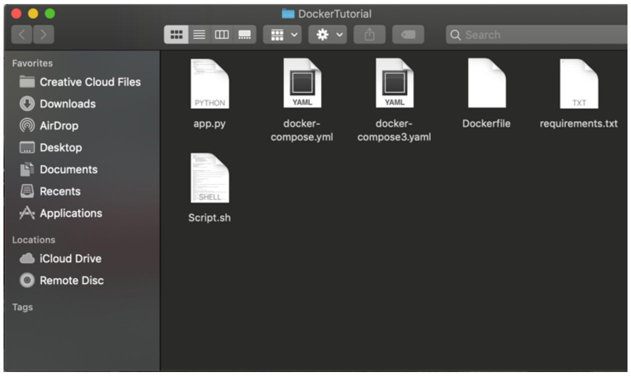


Figure 6: The contents of the docker tutorial folder that is available to download from the GitHub repository. This includes the python application, docker-compose.yml file, Docker file, requirements.txt for library dependencies, and the executable shell script.

5. SUMMARY

In this research project and over the five weeks of time allocated for this project, we grasp a wealth of knowledge on a new emerging software tool and dev ops style of application deployment. While also identifying a key issue within the use of this technology, evaluate the importance of that problem and quickly brainstorm an approach to a solution. Then work to that solution through trial and error while evaluating the key results and implication of the solution. Publish the technological implementation of the solution and elaborate future the work of this research.

Through this technological endeavor I have learned about a new application technology and discovered a relevant and popular problem in the process of learning and adapting how to use and leverage that technology. And conducted market research about how users wish to improve this issue. Followed by the implementation of a fast solution to this problem by using technology. Then reflected on the future work and implications of the research and outcome.

6. CONCLUSION

I gained a wealth of knowledge throughout the duration of my noble efforts involved in this project, this includes knowledge about docker as a company and a product, dev ops and software development lifecycle within organizations and how to use docker. In addition to this, my shell scripting skills have greatly improved, and I have become increasingly more proficient at trouble shooting docker technology setup issues and debugging shell scripts.

In regard to things that I would have done differently, this includes organizing and gathering users to conduct research about the likes and dislikes of the user experience during the script execution. This would have allowed me to collect critical feedback. In addition to creating and implementing improvements while programming the shell script and designing the user experience and product presentation for the users of the script. In addition to this I would have also liked to integrate an implementation of current graphical user interface technology that allows the user to point and click into the application and enter text into a text field that is more interactive and user friendly. Something I also would have liked to have done differently was to not have procrastinated on this project.

In regard to the future work of this project I strongly think that there is a large underlying customer market for a business opportunity or software company to create the product of interactive user-friendly software and technology tutorials for consumers.

7. ACKNOWLEDGEMENTS

Thank you to University of Oregon professor Ramakrishnan Durairajan for providing me with extensive knowledge and guidance and mentorship throughout the duration of this research project.

8. APPENDIX

This project, shell script, document, and tutorial are available for download at the following link:

https://github.com/fsmith503/ComputerScience/tree/master/FranklinSmithDockerTutorialScript

REFERENCES

[1] Serdar Yegulalp (2015 March 16) *Revealed: The best and worst of docker* https://www.infoworld.com/article/2896895/best-and-worst-about-docker.html

[2] Prakhar Srivastav (2019 March 15) *Learn to build and deploy your distributed applications easily to the cloud with docker* https://docker-curriculum.com/

[3] Puppet (2019 March 15) *Get into production with Puppet Enterprise* https://puppet.com/products/managed-technology/docker

[4] Amazon (2019 March 15) *Docker tutorial for beginners Build Ship and Run https://www.amazon.com/Docker-Tutorial-Beginners-Build-Ship/product-reviews/1719033498*

[5] Mike, Mackroy (2018 May 2nd) *Playing catch up with docker containers* https://rancher.com/playing-catch-docker-containers/

[6] Newport, Cal. (2016). *Deep Work*. London: Piatkus