

Leverage Docker Capabilities in Test Execution

▶ THOUGHT PAPER



Abstract

Docker is continuously gaining popularity among quality assurance (QA) side of the software field; QA teams are using the containers created using Docker for test executions. If we compare old test execution methods with the new testing culture, we can see that Docker containers used for test executions has created huge impact which is obviously testers favorable with respect test executions and resource utilizations.

We all know that in this evolving software industry, if we get some application to test, QA teams need to cover all the aspects with respect to executions in multiple browser. In this case Docker comes very handy which will allow us to create multiple containers with different configurations and leverage the same for our executions and then destroy the containers once our testing purpose is fulfilled so that memory and space optimizations can also be achieved which in turn will not degrade system and OS performance.

Key Takeaways

The key areas covered in this thought paper are as follows:

1

Docker Usage in Test Executions

2

Challenges In Cross Browser Execution

3

Benefits Of Using Docker in Test Execution

4

Containers Can be Big Asset to QA Team

Introduction

Docker is easy to use and a popular containerization platform that development and QA teams use to build, package applications, execute tests, and many more using Dockerized containers. When we compare Docker containers with virtual machines, the Docker containers are single unit containing all dependencies and packages needed to create suitable environment for dev or QA team and it offers optimum resource utilization, efficient build, faster application execution and many more which is big plus for anyone working in software industry.

Docker provide multiple benefits for development and testing team as well. In test automation, you can create docker image of automation code and create dockerise grid for test executions.

As a QA leader in your organization, you might have heard from your development counterpart regarding how much difference Docker have created in their daily task and how easy there life might have become due to introduction of Docker in software industry.

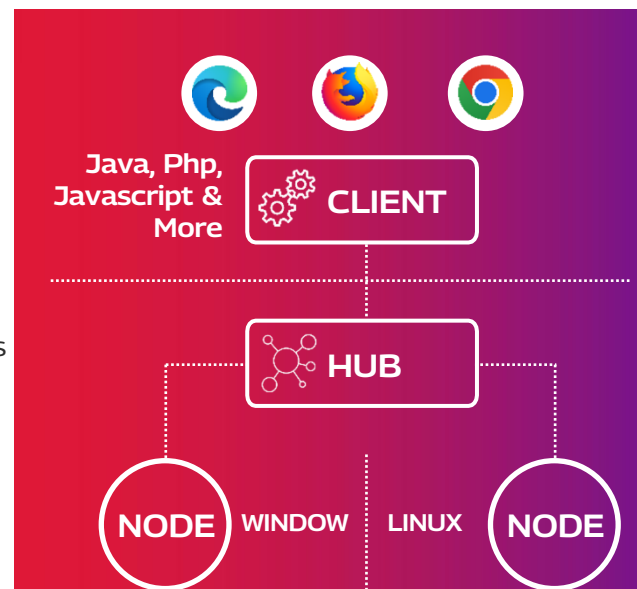
Docker has created similar effect in software QA industry as well due to multiple benefits it provides to run our regression tests and other tests in docker created containers.

Docker Usage in Test Executions

Docker can serve as a driver or facilitator to increase test execution speed; you can create number of containers of your choice/configuration using docker files or using other container creation mechanism and can use the same for scaling up your test execution in single machine containing multiple containers. It can be used in Windows or Linux based operating system to drive executions in multiple containers created using relevant Images.





As we all knows virtual machine uses guest and host OS, lot of memory/performance of CPU will be degraded and if we need to start a virtual machine every time as full OS will be loaded which makes it very slow, But when it comes to docker containers, they make use of host OS only and if we will create multiple containers it will share host OS/resources/relevant libraries and it will be very quick to start/create or scale up the containers as per the requirements without compromising the performance of the system. Containers are lightweight and faster as compared to virtual machines.

Cross browser testing on different browser-OS platforms can be performed on a fly with the use of Docker containers which will facilitates us to check whether websites are working as expected.










The steps below can be performed for test executions in Docker container:

-  Pull the Docker image which contains all the dependencies required to run test cases developed using any automation framework
-  Creating the Docker container with required image
-  Creating a sample test file in automation tools
-  Executing the test cases

Challenges In Cross Browser Execution

Now a days, cross browser testing has become a necessity in QA teams if we want to test some application or build rigorously across multiple browsers, through cross browser testing we can ensure that application is running fine across multiple browsers, browser version. Cross browser testing ensures that e2e test created for application yields same results for different browsers.

There are challenges which can come while setting up cross browser testing or remote executions environment -

-  In-house infrastructure setup and maintenance is a burden for QA's
-  Cloud based approach for achieving cross browser testing can be time taking process and requires multiple authentication to their application on cloud and on tools side also, multiple capabilities need to set before performing executions
-  Testing in multiple browsers including older version cannot be available or would be difficult to achieve in on-premises setup
-  Frequent browsers updates can also be one issue team might encounter
-  Automating test cases using some tools and feeding the capability to tool to run in different re-remote machines can be troublesome sometime

Benefits Of Using Docker in Test Execution

The software industry is evolving at a fast pace. So methodologies/ technologies used in our industry also need to be evolved. Docker attracts developers/testers and their teams because of its speed and simplicity. With Docker, developers can ship more and ensure faster delivery. Every team faces resource availability crunch like VM's/remote machines not being available, network connectivity issue in remote places, remote machines needed for test executions may not be powered on etc. All these issues are very well addressed using Dockers in which only single machine can act as a hub and node as well.

Running tests in Docker can actually provide multiple benefits and advantages, some are listed below:



Maintains and shares uniform environment: Developers and testers can share common containers in QA and development environments which in turn reduces dependency of operations team and builds can be delivered in much rapid speed.



Maintains execution privacy: Containers are ephemeral in nature, which makes tests to run independently and in exile environment without interfering with one another.



Easy maintenance: Containers are very easy to create and destroy in no time and containers can be repeatedly created as per defined definition file.



Continuous integration adaptable: Containerized test cases are very much adaptable to CI environment which in turn facilitates builds maintenance and trigger execution through CI/CD tools like Jenkins.



Scaling: No matter if you are using machines on site or in cloud, you will need to increase the number of them eventually. Using docker configuration is a lot smaller than with traditional approach.





Containers can be big asset to QA team, anyone can easily figure out with below points:

As QA team need to test applications in multiple environments to check the stability of application under test, it would consume lot of system resource and degrade system performance, so it is better to create containers with multiple configuration then test the application. Once tested, the container can be destroyed so that resources taken from host OS will be assigned back to host and performance issue can be corrected.

- ▶ It becomes very easy for QA engineer to get bug fixed with the help of container because they can easily share containers with dev team to reproduce the bug and get it fixed.
- ▶ All type of bugs can be tracked and traced if development team share the containers in which they did testing of their code with QA team, thus make a build bug free and stable.
- ▶ Multiple containers having multiple images can be created using single click which can be simultaneously utilized to achieve parallel executions of tests thus saving time of executions.
- ▶ Docker containers are very easy to create and destroy.





Conclusion

It is clear that Docker can be useful for the QA industry to make their job easy. When it comes to achieving resource optimization without compromising test coverage and performances, containerization using Docker can be better choice as compared to virtualization.

QA team can achieve the targets like execution of tests using Dockerized container in less time in a better way which in turn can deliver the product in short span of time. In the current era of software testing, which is evolving continuously, every team should use this approach of Dockerized environment to achieve their execution targets and should motivate other teams also to do the same.

Author



Manish Tiwari

*Tech Lead, Digital Assurance Services,
Tech Mahindra*

Manish Tiwari is working as an automation test architect and has around 9 years of experience in software industry. He has expertise in multiple open-source and licensed tools, like Selenium, Worksoft Certify, Tricentis Tosca, Cypress, Rest assured etc. Manish has an excellent experience in designing framework from scratch with respect to multiple tools for different QA teams. He is very passionate about learning new technologies and has worked in multiple testing methodologies, e.g., Agile, Waterfall, and has deep interest in DevOps culture.



TECH
mahindra