Docker Containers Usage in the Internet of Things: A Survey

* Mohd Syukor Abdul[1], Suriani Mohd Sam[2], Norliza Mohamed[3], Kamilia Kamardin[4], Rudzidatul Akmam Dziyauddin[5]

Universiti Teknologi Malaysia

Razak Faculty of Technology and Informatics
msyukor@graduate.utm.my[1], suriani.kl@utm.my[2], norlizam.kl@utm.my[3], kamilia@utm.my[4], rudzidatul.kl@utm.my[5]

Article history

Received:
9 Sept 2019

Received in revised form:
31 Oct 2019

Accepted:
20 Dec 2019

Published online:
30 Dec 2019

*Corresponding author
mysyukor@graduate.utm.my

Abstract

The Internet of Things (IoT) opened the way for enabling many of our everyday objects (things) interact with their environment to collect data, analyze and automating jobs based on specific rules. Within the constraint environment, the requirement of lightweight IoT application are tremendously indeed required to ensure the IoT application can be run efficiently. Docker containers is a promising technology to enable IoT application running smoothly, fast and efficient. In this paper, an introduction to Docker is presented. Then we explore the usage of Docker containers in the IoT application. Finally, we briefly discuss why Docker containers are usage in the IoT applications.

Keywords: *Internet of Things, IoT, Docker, Docker containers.*

1. Introduction

Internet of Things (IoT) will provide technological advances for modernization to improve productivity and better scope coverage of the domain. Internet protocols, smart sensors, communication technologies and analytics are factors of enabling IoT. IoT will boost people to automate processes, predict situations and improve existing daunting tasks even in real time for daily use. Their application spread out and advances rapidly in the smart homes, smart cities, transportation, manufacturing, healthcare, agriculture, and water management and energy sector. People worldwide trying to adopt IoT in their production environment as to improve their yields, productivities and better management (Paul & Saraswathi, 2017).

The structure of IoT is based on three layers as shows in Figure 1. From the bottom to top, the layers are called Perception Layer, Network Layer and Application Layer. The Perception Layer is the layer for sensing with the environment. The Network Layer is the layer where data transfer happen and the Application Layer is for data storage, data manipulation and analytics.

* *Corresponding author. mysyukor@graduate.utm.my*

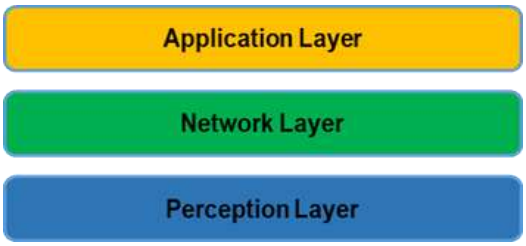

Figure 1. Three layers of IoT structure

* *Corresponding author. msyukor@graduate.utm.my*

Even with great advances and improvement in the IoT domain, IoT is still evolving as can be seen in several reviews (Singh, Tripathi, & Jara, 2014)(Al-Fuqaha, Guizani, Mohammadi, Aledhari, & Ayyash, 2015)(Tyagi & Kumar, 2017)(Paul & Saraswathi, 2017).

Docker technology is a new promising technology to revolutionize the workflow of application development. The main feature of Docker is the lightweight application virtualization that enables more density of applications that can be run within a single host. This lightweight characteristic is the main attraction of worldwide developers to adopt Docker lifecycle into their development, testing and production environment. The beauty of the Docker is that all application environment (development, testing and production) is actually are the same rather than different within the traditional application development, testing and deployment.

This paper begins with an introduction to the Docker technology, which represent the building blocks of Docker, such as Docker infrastructure, Docker containers creation and as well benefits of Docker containers. One of the goals of this work is to look into Docker usage in the IoT domain. This includes investigate which part of IoT domain Docker containers are used and why they are used. This work is based only on conferences and journal publication from IEEE digital publication.

2. Docker Technology

Docker container technology starts in 2013 when it was launched by Docker Inc. Docker environment provides facilities for application development in development environment, application testing in testing environment and application deployment in the production environment either for on-premises or in the cloud. (Docker, 2018a). Nowadays, Docker not only available in the Linux environment such as Fedora, Ubuntu, Debian, Red Hat Enterprise Linux, Alpine and CentOS but also in MacOS, IBM Power Z, Azure Cloud, AWS Cloud and Windows 10 operating system. (Docker, 2018c).

Docker introduce new terms regarding the usage of the Docker technology. Docker Container, Docker Image, Docker Engine, Docker Client, Docker Registry and Docker Volume are the terms introduced by the Docker. Figure 2 illustrates the meaning of the terms.

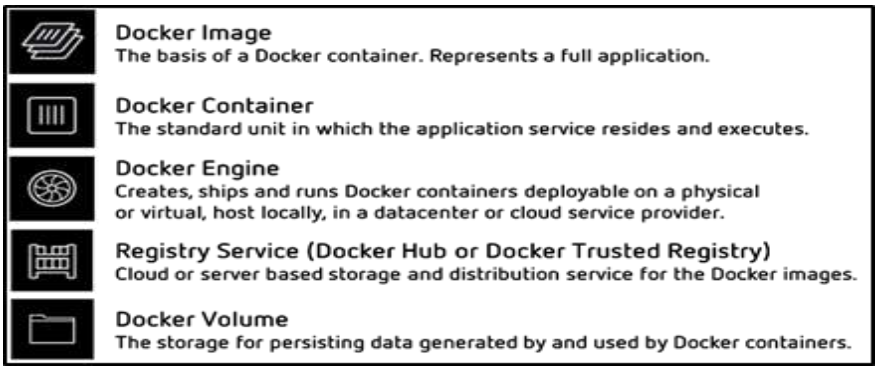

Figure 2. Docker technology terms.

A Docker image is an immutable. It does not have state and never changes. It is a set of layers starting with base layer, make changes that are saved in layers forming another image. A running Docker image is called Docker container. The Docker container run independently from each other. Docker enables the separation of applications from the infrastructure so the developer can deliver software quickly. The image holds all the necessary data for a process to run in the container. During the running mode, a new layer is appended to the existing base layer in the container. Each container have their own resources even they are from the same Docker Image. Figure 3 illustrates the layered Docker Image.

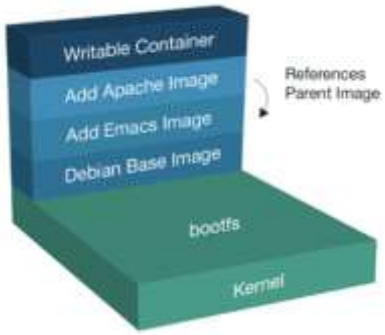

Figure 3. Docker image layered structure.

As in Figure 4 (Docker, 2018a), Docker Engine is a Docker lightweight runtime for building, run and orchestrate container based applications. The orchestration capabilities of Docker Engine are Swarm Mode Manager, Swarm Mode Worker, Certificate Authority, Transport Layer Security (TLS), Service Discovery, Load Balancing and Distributed Store.

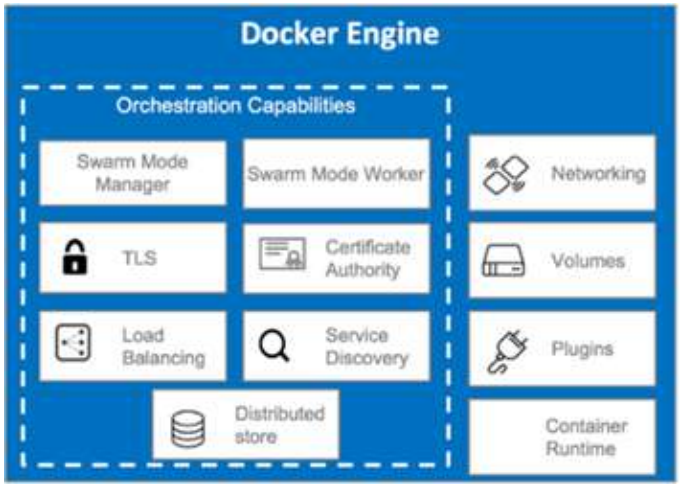

Figure 4. Docker Engine.

Docker Swarm Mode Manager and Swarm Mode Worker are used for Docker to do node clustering (Docker, 2018d). Certificate Authority and Transport Layer Security (TLS) are for container security. Service Discovery is function for enabling container services based on the application requirements in the container itself. While Load Balancing features is used to control the traffic load during container runtime and scaling and spawning more containers based on predefined traffic load rules. The Distributed Store is the repository for Docker Image. This repository is also known as Docker Hub (Docker, 2018b). Docker Engine also facilitate networking, plugins and volumes for the Docker container use during run time (Docker, 2018e).

In virtual machine infrastructure, each virtual machine have their own operating system, software libraries and the application itself. Figure 5 illustrate the virtual machine infrastructure.

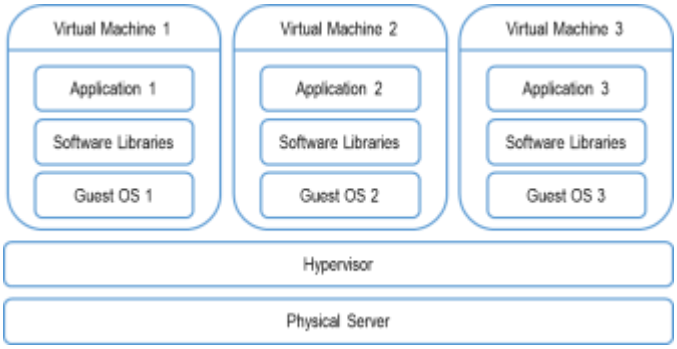

Figure 5. Virtual machine infrastruture.

While Docker sits directly on the operating system utilizing containers feature from within the operating system's kernel itself as depicted in Figure 6. Only application and required libraries are bundled together. This enable Docker to be more lightweight than virtual machine.

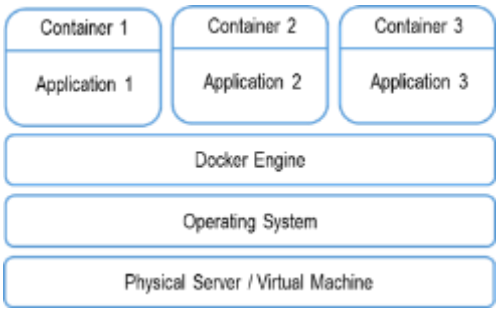

Figure 6. Docker infrastructure.

A Docker container can be built from a Dockerfile. This file is a special recipe file for Docker to build a Docker image. With this Dockerfile, more consistency in application development since all developer team used the same recipe in the application development. After the Docker Image built, we can invoke the Docker Image into running mode and is called a Docker container. A Docker container then can be committed back as a new Docker Image and pose differences from the original Docker Image. Figure 7 shows the interrelation between a Dockerfile, a Docker Image and a Docker container.

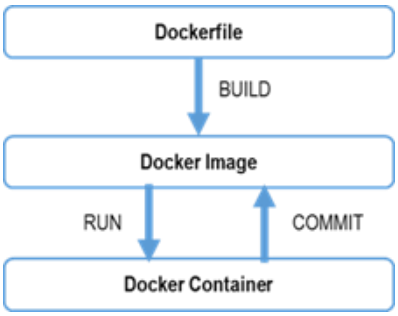

Figure 7. Docker image and Docker container interrelation.

Docker containers creation can be described as illustrated in the Figure 8. Docker client sent requests to the Docker daemon. Docker daemon is a special program, which is a part of Docker Engine that managed the entire Docker environment. Docker client can be either Docker command lines application or Docker Desktop application that are bundled and provided by the Docker. We can run the Docker daemon and Docker client on the same host. Docker client can also be connected other host as long as there is Docker installed in that host.

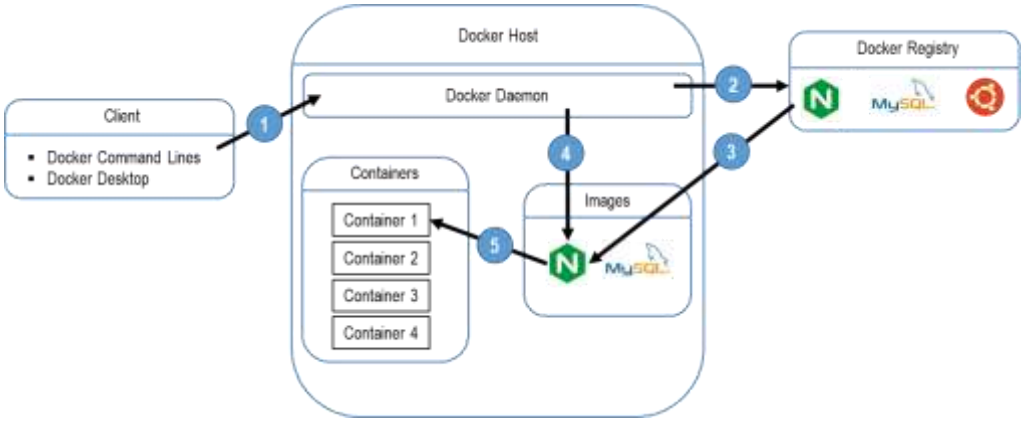

Figure 8. Docker container creation invocation.

Docker eliminate the situation of "that application works on development environment but not in the production environment". This is due to all containers are created from exactly the same Docker image. The developer team and operation team are just need to deploy the same Docker image from the development environment into production environment. With the single source of running application from Docker image, the attack surface for the application is small since the application security hardening procedures can be apply direct to base image before development team and operation team used the image. This will avoid security incident after the launching of the application.

Docker also provide facilities of public repository of Docker Image called Docker Hub. Docker Hub is large container images collection repository from around the world contributed by Docker team and by the others application developers. An application developer easily able to pull down freely the container image the Docker Hub and use it in their development and production environment. As of November 2018, there are more than 1.8 million container images available in the Docker Hub (Docker, 2018b).

3. Docker Used In the IoT

Here, we will explore the usage of Docker containers in the IoT application based on current research. Here, we briefly describe the usage of Docker for the IoT. These findings are based on the journals and conferences published in the IEEE digital library from year 2014 until 2018.

3.1. Message Queuing Telemetry Transport (MQTT) Broker Cluster

MQTT broker is used to exchanges measurement data and other data between IoT devices in the IoT environment. Some researchers (Jutadhamakorn et al., 2017) proposed implementation of MQTT broker that is low cost and scalable by using Docker containers. The MQTT cluster is implemented by using Raspberry Pi cluster with the Linux operating system. The MQTT broker is running in the Docker container within the Raspberry Pi. By using Raspberry Pi cluster rather than normal server for MQTT broker, better message broker throughput achieve and at a fraction of cost of normal server implementation.

3.2. Edge/Fog Computing

In (Deshpande & Liu, 2017), Docker resided in the IoT edge node. The Beaglebone Black is used to for the edge node. Due to lightweight and consume less memory, Docker is chose for the project to provide low latencies real time applications in the IoT edge computing.

In this paper (Mayer, Graser, Gupta, Saurez, & Ramachandran, 2017), authors proposed emulation framework called EmuFog for Fog computing. This emulation framework used Docker to build emulation platform that can simulate real Fog computing environment based on users defined requirement scripts. With the EmuFog, workloads and performance of the current studied Fog computing

environment can be evaluated and can be adjusted to suite the requirement of the users for the deployment of Fog computing infrastructure.

Foggy (Santoro, Zozin, Pizzolli, Pellegrini, & Cretti, 2017) is a workload orchestration and resources management platform based on the open source software to support IoT in Fog infrastructure. Docker containers are used to implement Foggy Negotiator, Foggy Orchestrator and Foggy Inventory within clustering environment for Fog computing infrastructure. With Docker and Kubernetes cluster, workload orchestration and resource negotiation can be implemented in Foggy platform.

Researchers (Tseng & Lin, 2018) experiment containerization of IoT/M2M platform called oneM2M. Fog Manager, Fog Workers and Fog Nodes are implemented in Docker environment. They also used Docker Swarm to managed containers cluster in the platform.

3.3. IoT Analytics

IoT systems may be located in a distant place between each other. Data analysis latency is expected to be at low latency between the user and the IoT application. Researchers (Cheng, Papageorgiou, Cirillo, & Kovacs, 2015) designed GeeLytics, an analytics platform which performs real-time analytics at network edge an in the cloud using Docker environment. With GeeLytics, they can achieve low latency analytics result and able to reduce the bandwidth consumption between the IoT edges and the IoT cloud.

Docker containers are also used in the analytics application as in (Hyun, Huh, & Park, 2018) to analyze greenhouse environmental data. The greenhouse environmental data are supplied to analytics application which in running in Docker containers. Docker containers are used in this analytics application due to easy deployment.

3.4. Smart Healthcare Monitoring System

Docker containers can also be used in the IoT application in healthcare area (Jaiswal et al., 2018). Patients' data collection can be automated with the function of the Raspberry Pi as edge devices and Docker is installed in the Raspberry Pi. This will assist doctor to diagnose patients easily disregarding patients location. Vital data of a patient will automatically collected and processed within the Docker containers in the Raspberry Pi. Current patient's status are viewed by using web browser or with mobile application. This so called IoT-Cloud healthcare model enable up to date patients' data and improve doctor response towards the patients.

3.5. Vehicle-to-Everything (V2X) Applications

Researchers (Rufino, Alam, & Ferreira, 2017) proposed that by using Docker, V2X applications be easily deployed and monitored though DevOps methodology. DevOps methodology enable software developer team and operation team worked together to achieve better software development and deployment lifecycle. The Docker containers are used as self-contained applications for V2X application.

Clustering of applications are done with the help of Docker Swarm. Thus, the real-time monitoring of V2X applications can be done.

3.6. Aquaculture

Docker containers and Docker Swarm are used in the intelligent turtle breeding system (Song, Xie, Huang, Wang, & Yu, 2018). A few of the Raspberry Pi are located at each of turtle breeding station to monitor environmental data. Each of them are installed with the Docker platform. Docker Swarm are used to clustering all Raspberry Pi within a breeding station. Therefore with the Docker implementation in the turtle's breeding base, not only reduces the operating cost of the turtle breeding station but also improve the turtle's survival rate.

3.7. Precision Agriculture

Containerized applications are used in the precision agriculture AgroDat.hu project (Marosi, Farkas, & Lovas, 2018). This project implement Docker for the data collector framework for more than 8000 hectares of precision agriculture site in Hungary. They also used Docker Swarm to manage the cluster of Docker. With Docker and virtual machines, they are able to create a knowledge center for precision agriculture.

3.8. Smart Homes

Researchers (Letondeur, Ottogalli, & Coupaye, 2018) showcased a "Smart Bell in a Collaborative Neighborhood" IoT application. They used Docker to deploy applications for smart homes for Fog computing. Their application lifecycle for Fog computing is based on Docker technologies.

Some researchers are working on a DeCyMo architecture for the use of blockchain for cyber physical systems for home and industry (Gallo, Nguyen, Barone, & van Hien, 2018). In DeCyMo, IoT gateway not only act as MQTT broker but also act as blockchain node. The Raspberry Pi is used to implement IoT gateway in the DeCyMo.

3.9. Energy Management

Docker containers and Docker Swarm are used to implement a prototype of IoT-Cloud (Nastic, Sehic, Le, Truong, & Dustdar, 2014) services for the smart energy (Kim, Kim, & Kim, 2017). In the cloud-based environment, the services provide users with a web interface called "Control Room" to maintain a proper temperature for server rooms and lowering operating costs.

3.10. Industrial IoT (IIoT)

A modular and scalable architecture based on Docker is proposed for Industrial IoT (IIoT) applications (Rufino, Alam, Ferreira, Rehman, & Tsang, 2017). Docker environment is used in the Raspberry Pi for cyber-physical and border gateway. Docker also used in the enterprise systems of the proposed architecture. With the Docker orchestration, reliability can be achieve for the IIoT.

Cyber physical systems are combination of the mechanical components such as Programmable Logic Controllers (PLCs) and software components tightly integrated to form a single entity. In paper (Tasci, Melcher, & Verl, 2018), Timur Tasci, Jan Melcher and Alexander Verl proposed an architecture for modularization of real time control applications by utilizing Docker containers in the preemptive real time Linux kernel. Linux kernel patched with preemptive real time function is necessary to enable real time control in the Linux environment. By doing this, it is feasible to run real time application in the Docker containers.

In (Garcia et al., 2018), the researchers proposed an industrial robot control platform by using Docker because of its lightweight and flexible characteristics. Consistency is achieved throughout different development and production releases with this architecture.

3.11. Smart Cities

Researchers in (Jang, Lee, Shin, & Lee, 2018) describe the use of Docker containers in the application of video analysis to guarantee public safety. They designed IoT camera by using Raspberry Pi and Raspberry Pi Camera module. Docker is installed on the Raspberry Pi and containerized IoT camera application in wireless connection.

4. Discussions

Docker platform is a new technology that enable containerization of application. Only the application and application dependencies are included in the container. The containerized application is smaller in size than current standard application packaging. Because of the current standard application packaging, operating system dependencies are still needed and based on the machine hosting the application.

Docker containers are more lightweight than virtual machine since virtual machine still required guest operating system to be installed first before the application can run in the virtual machine. While in containers based on Docker, the kernel of the operating system is shared between all containers. This reduces the size of the application container. Docker platform not provides containerization of application but also provide clustering facility called Docker Swarm. Docker Swarm is lightweight clustering and provides real time monitoring of the Docker containers cluster.

Based on our survey, there exist Docker environment usage in the IoT applications. Table 1 shows Docker usage in the IoT domain.

Table 1. Docker Usage in the IoT Domain

Internet of Things (IoT) Domain	Any Docker Usage?
Smart Homes	Yes
Wearables	No
Connected Cars	Yes
Industrial IoT	Yes
Smart Retail	No
Smart Healthcare	Yes

Smart Supply Chain	No
Smart Agriculture	Yes
Smart Water Management	No
Smart Cities	Yes
Smart Energy Management	Yes
Smart Poultry and Farming	No
Aquaculture	Yes

From the Table 1, we found that Docker containers are not used in certain IoT domain, such as in wearables, smart retail, smart supply chain, smart water management, and smart poultry. This is because IoT devices in wearables have a limited computing power, a limited operating system and a small size. Since Docker platform is still new in the market, cost of implementation is also considered before entering new technologies, such as for smart retail, smart water management and smart poultry and farming.

Docker platform is used in the IoT environment mainly due to its lightweight, better application development, fast application deployment and clustering features. Consistency is a factor in Docker's choice, since application in the development stage can easily be deployed more quickly in the production environment than traditional application development and deployment.

Today, performance is becoming a key driver of containers, but the complexity is still associated. And that's why there is such a diverse ecosystem: it's needed for users to build architectures that can take containers from the laptop and into the cloud that are fast and efficient. Thus by utilizing this kind of technology, IoT applications can be delivered at the same pace of the application deployment to the cloud. Docker environment provides possibility to do the same within the IoT environment.

Docker containers are more used in the edge/fog computing (Deshpande & Liu, 2017)(Mayer et al., 2017)(Santoro et al., 2017)(Tseng & Lin, 2018). In Edge/Fog computing environment, more processing functions are needed but still within the limitation of constraint capabilities of IoT devices. Thus, Docker features such as lightweight, easy application orchestration are among factors for the researchers and IoT developers choosing Docker platform for their IoT applications. Docker also offers clustering of distributed IoT devices and this enable many IoT devices to be cluster together for better management, monitoring and deployment.

In papers (Jutadhamakorn et al., 2017)(Jaiswal et al., 2018)(Song et al., 2018)(Gallo et al., 2018)(Rufino, Alam, Ferreira, et al., 2017)(Jang et al., 2018), we saw Raspberry Pis are used to deploy Docker platform. While in paper (Deshpande & Liu, 2017), BeagleBone Black boards are used to deploy Docker platform. These are due to Raspberry Pi and BeagleBone Black are low cost and flexible general purpose boards that support Linux-based operating system that can be installed with Docker platform. There is evident that prove that Raspberry Pi can be used for Docker platform. In Docker conference, DockerCon 2015, a Raspberry Pi 2 is used to run more than 2000 containers of web server (Victor Colsne, 2015).

From (Jutadhamakorn et al., 2017) and (Gallo et al., 2018), Docker containers are used to deploy MQTT broker for the IoT environment. By using Docker containers

for MQTT cluster, the researchers able to deploy manageable MQTT broker cluster for better performance MQTT broker at low cost. In paper (Gallo et al., 2018), Docker containers not only used for MQTT broker but also for blockchain node. This multifunction of IoT gateway can be realized due to lightweight feature of Docker containers.

Docker also perform well in the Industrial IoT as found in the (Rufino, Alam, Ferreira, et al., 2017)(Tasci et al., 2018)(Garcia et al., 2018). Docker platform also is used in real-time control application (Tasci et al., 2018) by using real-time kernel of Linux-based operating system.

Containers clustering are also seen being used in the Docker platform. Researchers (Jutadhamakorn et al., 2017), (Rufino, Alam, & Ferreira, 2017), (Song et al., 2018) and (Marosi et al., 2018) are using Docker Swarm to provide clustering of containers for IoT applications. Docker Swarm is chose due to its lightweight clustering and provide functions for easily managing containers cluster in the IoT devices.

The performance overhead of using Docker containers are considered low and near to native running applications but with better management for applications (Morabito, 2017). For a better security in the Docker containers, Center for Internet Security (CIS) published CIS Docker Benchmarks, a step-by-step checklist to secure Docker (Center for Internet Security, 2017).

5. Conclusion

This paper introduces the IoT and Docker containers. It is possible to run Docker containers in the IoT environment especially in the edge/fog computing since the devices is more capabilities in computing power since Docker platform required Linux-based operating system to be operational. Lightweight, consistency and easy orchestration of application containers in Docker is the main attraction for researchers and developers to explore the possibility of integrating the Docker platform into their IoT applications.

The IoT, which is currently in rapid advances, sparks new ideas on how to implement IoT applications. With the narrowing gap between IoT, cloud computing and fog/edge computing, IoT applications are designed for improved lifecycle development, robust application orchestration, fast deployment and tolerance of failures.

Docker environment is still considered to be large and limited in the IoT environment due to the lack of computing power and low memory capacity of the IoT devices. However, with the rapid development and advances in the IoT technologies, this gap will be narrow in the future and Docker containers can be widely used in the IoT environment.

Acknowledgments

This work is supported by the Universiti Technologi Malaysia. Thanks to Universiti Technologi Malaysia for providing access to many resources used for this research.

6. References

[1] Al-Fuqaha, A., Guizani, M., Mohammadi, M., Aledhari, M., & Ayyash, M. (2015). Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications. IEEE Communications Surveys & Tutorials, 17(4), 2347–2376. https://doi.org/10.1109/COMST.2015.2444095

[2] Center for Internet Security. (2017). CIS Docker Benchmarks. Retrieved November 3, 2018, from https://www.cisecurity.org/benchmark/docker/

[3] Cheng, B., Papageorgiou, A., Cirillo, F., & Kovacs, E. (2015). GeeLytics: Geo-distributed edge analytics for large scale IoT systems based on dynamic topology. 2015 IEEE 2nd World Forum on Internet of Things (WF-IoT), 565–570. https://doi.org/10.1109/WF-IoT.2015.7389116

[4] Deshpande, L., & Liu, K. (2017). Edge computing embedded platform with container migration. 2017 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computed, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI), 1–6. https://doi.org/10.1109/UIC-ATC.2017.8397578

[5] Docker. (2018a). About Docker. Retrieved October 17, 2018, from https://www.docker.com/company

[6] Docker. (2018b). Docker Hub. Retrieved October 17, 2018, from https://hub.docker.com/

[7] Docker. (2018c). Future proof your Windows apps and drive continuous innovation. Retrieved October 14, 2018, from https://www.docker.com/

[8] Docker. (2018d). Swarm mode overview - Docker Documentation. Retrieved October 18, 2018, from https://docs.docker.com/engine/swarm/

[9] Docker. (2018e). Use volumes - Docker Documentation. Retrieved October 18, 2018, from https://docs.docker.com/storage/volumes/

[10] Gallo, P., Nguyen, U. Q., Barone, G., & van Hien, P. (2018). DeCyMo: Decentralized Cyber-Physical System for Monitoring and Controlling Industries and Homes. 2018 IEEE 4th International Forum on Research and Technology for Society and Industry (RTSI), 1–4. https://doi.org/10.1109/RTSI.2018.8548507

[11] Garcia, C. A., Garcia, M. V., Irisarri, E., Perez, F., Marcos, M., & Estevez, E. (2018). Flexible Container Platform Architecture for Industrial Robot Control. 2018 IEEE 23rd International Conference on Emerging Technologies and Factory Automation (ETFA), 1, 1056–1059. https://doi.org/10.1109/ETFA.2018.8502496

[12] Hyun, W., Huh, M. Y., & Park, J. (2018). Implementation of docker-based smart greenhouse data analysis platform. 2018 International Conference on Information and Communication Technology Convergence (ICTC), 1103–1106. https://doi.org/10.1109/ICTC.2018.8539551

[13] Jaiswal, K., Sobhanayak, S., Turuk, A. K., Bibhudatta, S. L., Mohanta, B. K., & Jena, D. (2018). An IoT-Cloud Based Smart Healthcare Monitoring System Using Container Based Virtual Environment in Edge Device. 2018 International Conference on Emerging Trends and Innovations In Engineering And Technological Research (ICETIETR), 1–7. https://doi.org/10.1109/ICETIETR.2018.8529141

[14] Jang, S. Y., Lee, Y., Shin, B., & Lee, D. (2018). Application-Aware IoT Camera Virtualization for Video Analytics Edge Computing. 2018 IEEE/ACM Symposium on Edge Computing (SEC), 132–144. https://doi.org/10.1109/SEC.2018.00017

[15] Jutadhamakorn, P., Pillavas, T., Visoottiviseth, V., Takano, R., Haga, J., & Kobayashi, D. (2017). A scalable and low-cost MQTT broker clustering system. 2017 2nd International Conference on Information Technology (INCIT), 1–5. https://doi.org/10.1109/INCIT.2017.8257870

[16] Kim, S., Kim, C., & Kim, J. (2017). Reliable smart energy IoT-cloud service operation with container orchestration. 2017 19th Asia-Pacific Network Operations and Management Symposium (APNOMS), 378–381. https://doi.org/10.1109/APNOMS.2017.8094152

[17] Letondeur, L., Ottogalli, F. G., & Coupaye, T. (2018). A demo of application lifecycle management for IoT collaborative neighborhood in the Fog: Practical experiments and lessons learned around docker. 2017 IEEE Fog World Congress, FWC 2017, 1–6. https://doi.org/10.1109/FWC.2017.8368526

[18] Marosi, A. C., Farkas, A., & Lovas, R. (2018). An Adaptive Cloud-Based IoT Back-end Architecture and Its Applications. 2018 26th Euromicro International Conference on Parallel, Distributed and Network-Based Processing (PDP), 513–520. https://doi.org/10.1109/PDP2018.2018.00087

[19] Mayer, R., Graser, L., Gupta, H., Saurez, E., & Ramachandran, U. (2017). EmuFog: Extensible and scalable emulation of large-scale fog computing infrastructures. 2017 IEEE Fog World Congress (FWC), 1–6. https://doi.org/10.1109/FWC.2017.8368525

[20] Morabito, R. (2017). Virtualization on Internet of Things Edge Devices With Container Technologies: A Performance Evaluation. IEEE Access, 5, 8835–8850. https://doi.org/10.1109/ACCESS.2017.2704444

[21] Nastic, S., Sehic, S., Le, D.-H., Truong, H.-L., & Dustdar, S. (2014). Provisioning Software-Defined IoT Cloud Systems. 2014 International Conference on Future Internet of Things and Cloud, 288–295. https://doi.org/10.1109/FiCloud.2014.52

[22] Paul, P. V., & Saraswathi, R. (2017). The Internet of Things — A comprehensive survey. 2017 International Conference on Computation of Power, Energy Information and Commuincation (ICCPEIC), 421–426. https://doi.org/10.1109/ICCPEIC.2017.8290405

[23] Rufino, J., Alam, M., & Ferreira, J. (2017). Monitoring V2X Applications using DevOps and Docker. 2017 International Smart Cities Conference (ISC2), 1–5. https://doi.org/10.1109/ISC2.2017.8090868

[24] Rufino, J., Alam, M., Ferreira, J., Rehman, A., & Tsang, K. F. (2017). Orchestration of containerized microservices for IIoT using Docker. 2017 IEEE International Conference on Industrial Technology (ICIT), 1532–1536. https://doi.org/10.1109/ICIT.2017.7915594

[25] Santoro, D., Zozin, D., Pizzolli, D., Pellegrini, F. De, & Cretti, S. (2017). Foggy: A Platform for Workload Orchestration in a Fog Computing Environment. 2017 IEEE International Conference on Cloud Computing Technology and Science (CloudCom), 231–234. https://doi.org/10.1109/CloudCom.2017.62

[26] Singh, D., Tripathi, G., & Jara, A. J. (2014). A survey of Internet-of-Things: Future vision, architecture, challenges and services. 2014 IEEE World Forum on Internet of Things (WF-IoT), 287–292. https://doi.org/10.1109/WF-IoT.2014.6803174

[27] Song, Y., Xie, J., Huang, Q., Wang, M., & Yu, J. (2018). Design and Implementation of Turtle Breeding System Based on Embedded Container Cloud. 2018 2nd IEEE Advanced Information Management,Communicates,Electronic and Automation Control Conference (IMCEC), 2531–2534. https://doi.org/10.1109/IMCEC.2018.8469537

[28] Tasci, T., Melcher, J., & Verl, A. (2018). A Container-based Architecture for Real-Time Control Applications. 2018 IEEE International Conference on Engineering, Technology and Innovation (ICE/ITMC), 1–9. https://doi.org/10.1109/ICE.2018.8436369

[29] Tseng, C.-L., & Lin, F. J. (2018). Extending scalability of IoT/M2M platforms with Fog computing. 2018 IEEE 4th World Forum on Internet of Things (WF-IoT), 825–830. https://doi.org/10.1109/WF-IoT.2018.8355143

[30] Tyagi, V., & Kumar, A. (2017). Internet of Things and social networks: A survey. 2017 International Conference on Computing, Communication and Automation (ICCCA), 1268–1270. https://doi.org/10.1109/CCAA.2017.8230013

[31] Victor Colsne. (2015). Update: Raspberry Pi DockerCon Challenge - Docker Blog. Retrieved November 5, 2018, from https://blog.docker.com/2015/09/update-raspberry-pi-dockercon-challenge/