

PLAGIARISM SCAN REPORT

Words 739 Date April 21,2021

Characters 4595 Excluded URL

5%

Plagiarism

95%

Unique

2

Plagiarized
Sentences

38

Unique Sentences

Content Checked For Plagiarism

Container and Container Orchestration

You can think of containers as lightweight, scalable, and isolated VMs (not an actual VM) in which you run your applications. A container is a standalone package that encapsulates application code and all of its dependencies so that the application can run in any environment. Containers give you portability, isolation, and packaging.

To understand the concept of containers, suppose someone works on a project and it works fine in one's machine, but when the same project is run on any other machine it doesn't work. It can be because of any dependency or because of a different version of a library. Containers can prove to be helpful here. Just wrap everything inside an image (containers are running instances of images) and other users can directly run this image.

To illustrate the concept of containers further, suppose a developer is working on a web application that requires PHP 7.2. So, he installs PHP 7.2 on his local machine. Later, he updates the app to use PHP 7.2 to 8.0. As a result, all the developers working on the same web app will also need to update PHP. If some more such libraries or dependencies are updated, all the developers need to update everything again. But while using containers, only one developer needs to update everything and create a new image; all the other developers can use that image since it contains all the updated libraries and dependencies inside it.

In the right figure above, each application is running its copy of the OS (Guest OS). Whereas containers share host OS and hence, are much more efficient, lightweight, and take less time to start. There are varieties of container runtime available like docker, rocket, containers, etc. Docker is the most popular container runtime. Maybe now you can reread paragraph one to understand it better.

Container Orchestration:

There are multiple container orchestrations available like Kubernetes, docker swarm, etc. Kubernetes is the most popular container orchestrator. Developed by a team at Google, it was later donated to Cloud Native Computing Foundation (CNCF) and became open source. It automates container operations. It eliminates many of the manual processes involved in deploying and scaling containerized applications. You can cluster together groups of hosts running containers, and Kubernetes helps you manage those clusters efficiently.

To understand the concept of Kubernetes and what we can do with Kubernetes, suppose one owns a shopping website.

Case 1: Suppose he hosts a server only in the Delhi region to handle all requests from India. But due to some reason (like hardware failure), the server goes down, implying that his site goes down as well. Therefore he should host his website in multiple regions in India like Delhi, Mumbai, and Kolkata. This is called high availability. Problems like hardware failures do not bring his application down because he has multiple instances of his application.

Case 2: Suppose there is too much load in the Delhi region. How can he handle that? By sending traffic to other servers. So he needs a load balancing way to balance the traffic in all the regions.

Case 3: Suppose there is a sale and server traffic increases too much in all the regions. So he needs to scale up the server to handle all the traffic. Now, when the sale is over, he is wasting his scaled resources. So he needs to scale down his server again. Therefore he needs something so that he can easily scale up and down according to his needs.

Case 4: Suppose he has to update his website during the sale, and that needs to be done without any downtime. So he wants zero downtime deployment.

Case 5: He doesn't want any of his services (in a microservice architecture, see image below) as the payment service goes down. So he needs a Health check and Self-healing for all of his services. If anything goes down, it automatically

heals, which means automatic restart of the service.

Using Kubernetes, one can solve all the above problems. There are lots of other features available that anyone can explore. It is open-source, used by different organizations and lots of new features are added in every version. Many companies are getting a head start on Kubernetes and joining the revolution. By using containers and container orchestration, organizations can build and deploy horizontally scalable lightweight applications across multiple types of server hosts, cloud environments, and other infrastructure more effectively and efficiently.

Sources	Similarity
<p>What is orchestration? - Red Hat</p> <p>It eliminates many of the manual processes involved in deploying and scaling containerized applications. There are also managed services available for orchestrating containers. Server configuration and management and application deployments can be orchestrated with a tool like Red Hat Ansible Automation Platform.</p> <p>https://www.redhat.com/en/topics/automation/what-is-orchestration</p>	5%
<p>What is Kubernetes? - Learnk8s</p> <p>Many companies are getting a head start on Kubernetes and joining the revolution. Not just startups and SMEs, but big corporations such as banks, financial institutions and insurance companies are betting on containers and Kubernetes to be the future. Even companies invested in the Internet of Things and embedded systems.</p> <p>https://learnk8s.io/blog/what-is-kubernetes</p>	3%