**Kubernetes Assignment 4**

1. What are the benefits of having operators?

* Lets fist see operators, in Kubernetes an operator is a way to automate the management of an application, including its deployment and package management, using the kubectl tooling and the Kubernetes API .
* Operators are software extensions to Kubernetes that make use of custom resources to manage applications and their components.
* Below are some of the benefits:
* Reduce complexity

In older times applications deployment was very complex which includes installing, updating, securing, backing up, restoring applications and time consuming. with the help of kubernetes operators we can convert high level into low level design with the help of automations

So you can say we/app developers need to be experts in there container application. Kubernetes operators allow easy installation and quick update and can easily scale .

* CRUD   
   There is no need to apply basic operations on your app as we can use Kubernetes extensibility through HTTP and kubectl.
* Improve consistency

Like in olden days we need manual intervention for maintenance now that maintenance and other tasks traditionally performed by an operations team are automated, removing the risk of human error and freeing up time for more valuable work. When it comes to improving consistency, Kubernetes Operators offer multiple benefits.

Kubernetes Operators monitor the cluster once it is deployed by continuously scanning for anything abnormal behaviour.

Suppose if there is any error found then it is identified and corrected automatically. This approach reduces support burdens by identifying and correcting application problems on Kubernetes clusters.

* Automate at the level you need

we can customize and automate as per the user requirements like into multiple sub task/level

Level 1  
Simple Installation which provides automated application provisioning and configuration management.

Level 2  
Let say a new version of that software is realised then we can easily automate with upgrades and patching.

Level 3  
Full life cycle, which encompasses both the app and storage life cycle—including backup and failure recovery.

Level 4  
Fully we can enable the monitoring, logs check-up, alerting part and performance checkups.

Level 5  
Depending upon business requirement we can scale horizontal and vertical and auto tuning of application.

* In short if we want to conclude then process of managing applications in Kubernetes isn't as straightforward as managing stateless applications, where reaching the desired status and upgrades are both handled the same way for every replica. In stateful applications, upgrading each replica might require different handling due to the stateful nature of the app, each replica might be in a different status. As a result, we often need a human operator to manage stateful applications. Kubernetes Operator is supposed to assist with this.

1. What exactly is GKE?

* GKE full form - Google Kubernetes Engine that is used for managing and orchestrating systems for Docker containers.
* With the help of Google Public Cloud, we can also orchestrate the container cluster
* Google Kubernetes Engine provides a managed environment for deploying, managing, and scaling your containerized applications using Google infrastructure cloud
* The GKE environment consists of multiple machines like Compute Engine instances which are grouped together to form a cluster.

1. What is the best way to operate Kubernetes locally?

* we can run kubernetes locally in multiple ways but to say which is the best or which we choose really depends on what fits your situation the best. then also if given a chance will use **minikube** though there are other ways to use them will just go through below list.
* **Minikube**
* Kubernetes can be set up locally using the Minikube tool. It runs a single-node cluster, bunch in a VM on the computer. It offers the perfect way for users who have just starting learning Kubernetes
* IT is the best tool for local Kubernetes application development, and to support all Kubernetes features that fit
* **Microk8s**
* it is an open-source system for automating deployment, scaling, and management of containerised applications. It provides the functionality of core Kubernetes components, it is scalable from a single node to a high-availability production cluster.
* **k3s**
* It is a lightweight tool designed to run production-level Kubernetes workloads for low resourced and remotely located IoT and Edge devices
* **Kind**
* It is another Kubernetes SIGs project but is quite different compared to minikube. As the name suggests it moves the cluster into Docker containers. This leads to a significantly faster startup speed compared to spawning VM.
* there are few more ways which we can operate kubernetes locally.

1. What is the difference between Kubernetes and Docker Swarm?

Let's see the major difference listed

|  |  |
| --- | --- |
| Kubernetes | Docker swarm |
| Installation is complicated | Installation is easy |
| Once installation is done cluster is robust | Once installation is done cluster is not robust |
| It can process auto-scaling | It cannot process auto scaling |
| Manually configuration is required for load balancing | It does auto load balancing |
| Kubernetes provides such integrated tools for logging and monitoring purposes. | Docker requires third-party tools such as the ELK stack for logging and monitoring. |
| In Kubernetes, we can only share storage volumes with containers in the same pod. | In Docker Swarm, we can easily share storage volumes with any container. |
| we can deploy rolling updates as well as automatic rollbacks. | We can deploy rolling updates in Docker Swarm but can't deploy automatic rollbacks. |
| Kubernetes provides a GUI in the form of a dashboard. | Docker Swarm doesn't provide a GUI. |

1. How does Kubernetes make containerized deployment easier?

* Now a days any large/small software companies alike are now deploying thousands of container instances daily
* so as the containers increase the complexity also increase to manage of them.
* Kubernetes makes it easy to deploy and operate applications in a microservice architecture. it manages easily by creating an abstraction layer on top of a group of hosts, so that development teams can deploy their applications and let Kubernetes manage the following activities:
* Controlling resource consumption by application or dev team
* Load balancing or we can say evenly spreading application load across a host infrastructure
* Automatically load balancing requests across the different instances of an application
* Monitoring resource consumption and resource limits to automatically stop applications from consuming too many resources and restarting the applications again
* It moved an application instance from one host to another if there are resource shortage or even if say a host crashes due to any reason it will automatically add additional resources made available when a new host is added to the cluster
* Kubernetes helps you move faster
* It works on multiple clouds
* Cloud providers will manage Kubernetes for you
* Automated Scheduling/Automates various manual processes - for instance, Kubernetes will control for you which server will host the container, how it will be launched etc.
* Interacts with several groups of containers - Kubernetes is able to manage more cluster at the same time
* Provides additional services - as well as the management of containers, Kubernetes offers security, networking and storage services
* Self-monitoring - Kubernetes checks constantly the health of nodes and containers
* Horizontal scaling - Kubernetes allows you scaling resources not only vertically but also horizontally, easily and quickly
* Storage orchestration: Kubernetes mounts and add storage system of your choice to run apps
* Automates rollouts and rollbacks: if after a change to your application something goes wrong, Kubernetes will rollback for you sometimes also known for Self-Healing Capabilities
* Container balancing: Kubernetes always knows where to place containers, by calculating the “best location” for them
* Run everywhere: Kubernetes is an open source tool and gives you the freedom to take advantage of on-premises, hybrid, or public cloud infrastructure, letting you move workloads to anywhere you want.
* lastly it is very cost effective.