**Kubernetes Assignment 3**

1. How do you monitor the Kubernetes cluster?

* there are various tools and methods for monitoring. Let me list few of them
* Heapster, cAdvisor, Prometheus, InfluxDB, Grafana
* let see few of them
* Prometheus is an application that is used for monitoring and alerting. It can be called out to your systems, grab real-time metrics, compress it, and stores properly in a database.
* to monitor your Kubernetes cluster is by using a combination of Heapster to collect metrics, InfluxDB to store it in a time series database, and Grafana to present and aggregate the collected information.
* The Prometheus ecosystem consists of multiple components.
* Mainly Prometheus server which scrapes and stores time-series data.
* Special-purpose exporters for services like StatsD, HAProxy, Graphite, etc.
* An alert manager to handle alerts on various support tools.
* Client libraries for instrumenting application code.
* Push gateway for supporting short-lived jobs.

1. How do we get POD's central logs?

* Pod is smallest unit inside worker node insidee which containers are running. it will be encapsulated inside the pod. you can n no. of pods. under which your applications runs
* A pod is the smallest deployable unit that can be managed by Kubernetes.
* A pod is a logical group of one or more containers that share the same IP address and port space.
* The main purpose of a pod is to support co-located processes, such as an application server and its local cache
* This architecture depends upon the application and many other factors.
* let us see few of the common logging patterns
* Streaming sidecar container.
* Node level logging agent.
* Export logs directly from the application.
* Sidecar container with the logging agent.
* In the setup, journalbeat and filebeat are running as daemonset. Logs collected by these are dumped to the kafka topic which is eventually dumped to the ELK stack.
* The same can be achieved using EFK stack and fluentd-bit.

1. What are the components of a Kubernetes node?

* A node is a machine either physical or virtual machine on which kubernetes is installed
* A node is a worker machine and this is where containers inside the pods will be launched by kubernetes
* Node components run on every node, maintaining running pods and providing the Kubernetes runtime environment
* you can have N no. of nodes, depending on how much big your app is.
* Inside worker node we below:

1. **pod**

* It is smallest unit inside worker node inside which containers are running. it will be encapsulated inside the pod. you can n no. of pods. under which you applications runs

1. **Docker**

* it provides containerization solution

1. **Kubelet**

* It is a node agent which tracks/stores, create, update,delete contrainer. Component is being managed by kubelet.
* Ensures that containers are running in a Pod by interacting with the Docker engine , the default program for creating and managing containers. Takes a set of provided PodSpecs and ensures that their corresponding containers are fully operational.

1. **Kube proxy**

* it is used for network proxy
* **Manages network connectivity and maintains network rules across nodes. Implements the Kubernetes Service concept across every node in a given cluster.**

1. How does Kubernetes make containerized deployment more manageable?

Below are the few points or highlights which makes kubernetes more manageable. Let us go in little depth

* Kubernetes is also known as K8s is an open-source system for automating deployment, scaling, and management of containerized applications.
* it manages containers
* It groups containers that make up an application into logical units for easy management and discovery.
* Kubernetes is used to manage your containers
* It is an open-source container Orchestration Framework.
* it manages containers - to manage applications that are made of containers physical machines, virtual machines, hybrid environments
* A **Kubernetes cluster** is a set of nodes that run containerized applications.
* If you’re running Kubernetes, you’re running a cluster.
* Containerizing applications packages an app with its dependences and some necessary services. They are more lightweight and flexible than virtual machines.
* Kubernetes clusters allow for applications to be more easily developed, moved and managed.
* Kubernetes clusters allow containers to run across multiple machines and environments: virtual, physical, cloud-based, and on-premises.
* Kubernetes containers are not restricted to a specific operating system, unlike virtual machines. Instead, they are able to share operating systems and run anywhere.
* Kubernetes clusters are comprised of **one master node and a number of worker nodes.** These nodes can either be physical computers or virtual machines, depending on the cluster.
* The master node controls the state of the cluster; for example, which applications are running and their corresponding container images. The master node is the origin for all task assignments. It coordinates processes such as:
  1. Scheduling and scaling applications
  2. Maintaining a cluster’s state
  3. Implementing updates
* The worker nodes are the components that run these applications. Worker nodes perform tasks assigned by the master node. They can either be virtual machines or physical computers, all operating as part of one system.
* There must be a minimum of one master node and one worker node for a Kubernetes cluster to be operational.

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

Below are the major differences:

|  |  |
| --- | --- |
| 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 |

1. What exactly does Kubernetes controller manager imply?

* It is a part of Master Node
* task of Control Manager is to create note, delete note, update.
* it keeps a track of each resources, eg../how many/ what no. of nodes are running. etc.
* Runs controller processes and reconciles the cluster’s actual state with its desired specifications. Manages controllers such as node controllers, endpoints controllers and replication controllers.