Introduction to CaaS

A brief introduction to container

A container is a standard unit of software that packages up code and all its dependencies so the application can be ported from one computing environment to another where it can run quickly and reliably. Docker is the most popular container runtime. Docker packages(containerize) your application and all its dependencies together in the form of a docker container image that is portable and works seamlessly in any environment that supports docker containers.

Containers are supported in Linux, Windows, and other modern operating systems. A container can run anything from small software or a micro service to large applications.

What is Kubernetes

Kubernetes, popularly known as K8s, is an open-source system for automating deployment, scaling, and management of containerized applications. Kubernetes distributes containerized applications across a cluster of machines. Kubernetes adds distributed computing features on top of containers such as load balancing, autoscaling, self-healing, rollout and roll back features etc.

Docker and kubernetes are related

How Dockers helps

- Docker provides the ability to package and run an application in a loosely isolated environment called a container.

- The isolation and security allow you to run many containers simultaneously on a given host.

- Docker provides tooling and a platform to manage the lifecycle of your containers.

- Containerised application facilitates fast, consistent delivery of your applications.

- Docker’s container-based platform allows for highly portable workloads.

- Docker helps in dynamically managing workloads, scaling up or tearing down applications and services as business needs dictate, in near real time.

- In the long run, docker provides a viable, cost-effective alternative to hypervisor-based virtual machines, so you can use more of your compute capacity to achieve your business goals.

How Kubernetes helps

K8s helps in managing containerized workloads and services, which facilitates both declarative configuration and automation. It provides features that facilitate agile, continuous development, integration and delivery. The K8skey features are:

- **Service discovery and load balancing**: Kubernetes can expose a container using the DNS name or using their own IP address. If traffic to a container is high, Kubernetes is able to load balance and distribute the network traffic so that the deployment is stable.
- **Storage orchestration**: Kubernetes allows you to automatically mount a storage system of your choice, such as local storages, public cloud providers, and more.

- **Automated rollouts and rollbacks**: You can describe the desired state for your deployed containers using Kubernetes, and it can change the actual state to the desired state at a controlled rate. For example, you can automate Kubernetes to create new containers for your deployment, remove existing containers and adapt all their resources to the new container.
- **Automatic bin packing**: You provide Kubernetes with a cluster of nodes that it can use to run containerized tasks. You tell Kubernetes how much CPU and memory (RAM) each container needs. Kubernetes can fit containers onto your nodes to make the best use of your resources.
- **Self-healing**: Kubernetes restarts containers that fail, replaces containers, kills containers that don't respond to your user-defined health check, and doesn't advertise them to clients until they are ready to serve.
- **Secret and configuration management**: Kubernetes lets you store and manage sensitive information, such as passwords, OAuth tokens, and SSH keys. You can deploy and update secrets and application configuration without rebuilding your container images, and without exposing secrets in your stack configuration.