



BASICS OF KUBERNETES

Basics of Kubernetes

Components of Kubernetes

Deploying containers and using Kubernetes may require a change in the development and the system administration approach to deploying applications. In a traditional environment, an application (such as a web server) would be a monolithic application placed on a dedicated server. As the web traffic increases, the application would be tuned, and perhaps moved to bigger and bigger hardware. After a couple of years, a lot of customization may have been done in order to meet the current web traffic needs.

Instead of using a large server, Kubernetes approaches the same issue by deploying a large number of small servers, or microservices. The server and client sides of the application are written to expect that there are many possible agents available to respond to a request. It is also important that clients expect the server processes to die and be replaced, leading to a transient server deployment. Instead of a large Apache web server with many httpd daemons responding to page requests, there would be many nginx servers, each responding.

The transient nature of smaller services also allows for decoupling. Each aspect of the traditional application is replaced with a dedicated, but transient, microservice or agent. To join these agents, or their replacements together, we use services. A service ties traffic from one agent to another (for example, a frontend web server to a backend database) and handles new IP or other information, should either one die and be replaced.

Communication is entirely API call-driven, which allows for flexibility. Cluster configuration information is stored in a JSON format inside of etcd, but is most often written in YAML by the community. Kubernetes agents convert the YAML to JSON prior to persistence to the database.



Kubernetes is written in Go Language, a portable language which is like a hybridization between C++, Python, and Java. Some claim it incorporates the best (while some claim the worst) parts of each.

