



## SERVICES

## Services

### Overview

As touched on previously, the Kubernetes architecture is built on the concept of transient, decoupled objects connected together. Services are the agents which connect Pods together, or provide access outside of the cluster, with the idea that any particular Pod could be terminated and rebuilt. Typically using Labels, the refreshed Pod is connected and the microservice continues to provide the expected resource via an **Endpoint** object. Google has been working on Extensible Service Proxy (ESP), based off the nginx HTTP reverse proxy server, to provide a more flexible and powerful object than Endpoints, but ESP has not been adopted much outside of the Google App Engine or GKE environments.

There are several different service types, with the flexibility to add more, as necessary. Each service can be exposed internally or externally to the cluster. A service can also connect internal resources to an external resource, such as a third-party database.

The kube-proxy agent watches the Kubernetes API for new services and endpoints being created on each node. It opens random ports and listens for traffic to the **ClusterIP:Port**, and redirects the traffic to the randomly generated service endpoints.

Services provide automatic load-balancing, matching a label query. While there is no configuration of this option, there is the possibility of session affinity via IP. Also, a headless service, one without a fixed IP nor load-balancing, can be configured.

Unique IP addresses are assigned and configured via the etcd database, so that Services implement iptables to route traffic, but could leverage other technologies to provide access to resources in the future.