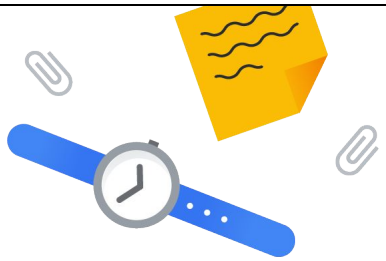

Today's agenda

01 Introduction

02 Clusters and services

03 Lab: Creating Services and Ingress Resources

04 Pod DNS

05 Quiz

Making an application highly available

- Cymbal Bank has a Kubernetes cluster that is spread across multiple zones.
- Sasha is asked to
  - Deploy a new application that needs to be available 24/7.
  - Improve application performance by reducing the number of DNS queries that need to be made.
- What can Sasha do?

What Sasha can do

- Sasha can use Service Directory for GKE to register your services in multiple zones.
- Service Directory for GKE
  - Ensures that your services are available even if one zone becomes unavailable.
  - Can resolve requests using DNS, HTTP, and gRPC.
  - Can populate Cloud DNS records that match services in Service Directory.
  - Improves the performance of service discovery by reducing the number of DNS queries that need to be made.

Google Cloud

When you use Service Directory for GKE, you register your services with Service Directory. Service Directory then provides a DNS name for each service that you can use to resolve the service endpoints. Your applications then can use the DNS name to resolve the service endpoints. The applications don't have to make a separate DNS call for each endpoint.

Suppose you have a service called my-service that runs on two pods. You can register the service with Service Directory and give it the DNS name my-service.default.svc.cluster.local. Your applications can then use the DNS name my-service.default.svc.cluster.local to resolve the service endpoints without having to make two separate DNS calls.

For more information, refer to the Service Directory cheat sheet on the Google Cloud blog site and Service Directory for GKE overview in the Google Cloud documentation.

Service Directory for GKE

- Uses IAM to assign and control service visibility and permissions.
- Provides Google Cloud CLI and Google Cloud console support for interacting with Service Directory.
- Uses Cloud Monitoring and Cloud Logging for monitoring, auditing, and debugging Service Directory operations.

Google Cloud

Integration with Cloud DNS. Service Directory zones allow services to be made available on Virtual Private Cloud (VPC).

Service Directory limitations

- You can only associate a Service Directory zone with a namespace when you create the zone.
- A Service Directory zone must be in the same project as the Service Directory namespace that it’s associated with.
- A Service Directory zone can't also be a forwarding zone, a regular private zone, or a public zone.

Other ways to implement service discovery

Cloud DNS and kube-dns

Standard mode clusters: resolve service names and external names with:

- **kube-dns**: the default Kubernetes DNS name server, deployed by default in all GKE clusters.
- **Cloud DNS**: tells GKE to use Cloud DNS.

Note

Cloud DNS is the only DNS provider for newer Autopilot clusters.

Google Cloud

For some situations - instead of kube-dns, Cloud DNS, or Service Directory for GKE - you may want to use CoreDNS. For more information, refer to How to run CoreDNS on Kubernetes Engine in the Google Cloud documentation.

Cloud DNS is the only DNS provider for newer Autopilot clusters running version 1.25.9-gke.400 and later, and version 1.26.4-gke.500 and later.

For information about Cloud DNS, refer to Cloud DNS overview in the Google Cloud documentation.

Other ways to implement service discovery

NodeLocal DNSCache

NodeLocal DNSCache runs as a DaemonSet that schedules a DNS cache Pod on every cluster node which

- Improves DNS lookup latency.
- Makes DNS lookup times more consistent
- Can reduce the number of DNS queries to kube-dns or Cloud DNS.

Google Cloud

The NodeLocal DNSCache runs on standard mode and autopilot mode clusters.

kube-dns

- kube-dns
  - Is provided by Kubernetes and runs on each node in the cluster.
  - Resolves the names of pods to their IP addresses.
  - Is designed for highly availability and resiliency.

Google Cloud

You can specify a DNS server to use in the kube-dns-config.yaml file.

To customize kube-dns in GKE, refer to Setting up a custom kube-dns Deployment in the Google Cloud documentation.

Cloud DNS

- Cloud DNS is a high-performance, feature-rich, resilient, global Domain Name System (DNS) offered by Google Cloud.
- Cloud DNS can be used in GKE, in place of kube-dns.
- Compared to kube-dns, Cloud DNS is:
  - More scalable.
  - Easier to manage.

After you enable Cloud DNS for a cluster, the settings only apply if you upgrade existing node pools or you add new node pools to the cluster.

Google Cloud

Cloud DNS provides Pod and Service DNS resolution without a cluster-hosted DNS provider like kube-dns. The Cloud DNS controller automatically provisions DNS records for pods and services in Cloud DNS for ClusterIP, headless and external name services.

Cloud DNS is designed to be more extensible than kube-dns. If you have latency issues with kube-dns, you should consider using Cloud DNS. Refer to Performance limitations with kube-dns in the Google documentation for more information. (Alternately, kube-dns latency issues can be addressed as described in Scaling up kube-dns.)

Cloud DNS can be managed using Google Cloud Console or using the Google Cloud CLI. This makes Cloud DNS easier to manage than kube-dns.

After you enable Cloud DNS, after you upgrade the existing node pools, the nodes are recreated.

Cloud DNS

Cluster scope

- Use cluster scope for the highest level of pod isolation.
- Pods can't be accessed by pods in other clusters.
- Pods in your cluster can only be accessed by:
  - Other pods in your cluster.
  - Pods in other clusters that are also using cluster scope DNS.
- By default, GKE Cloud DNS uses cluster scope.

Google Cloud

DNS records are only resolvable within the cluster, which is the same behavior as kube-dns. Only nodes running in the GKE cluster can resolve Service names. By default, clusters have DNS names that end in *.cluster.local. These DNS names are only visible within the cluster and do not overlap or conflict with *.cluster.local DNS names for other GKE clusters in the same project.

Refer to Using Cloud DNS for GKE and Restrictions and limitations in the Google Cloud documentation for more information.

Cloud DNS

VPC scope

- Use VPC scope for a lower level of pod isolation.
- This scope is useful when pods need to be accessed by pods that are outside of the cluster.
- Use firewall rules or network policies to control which pods can access each other.

Google Cloud

DNS records are resolvable within the entire VPC. Compute Engine VMs and on-premises clients can connect using Cloud Interconnect or Cloud VPN and directly resolve GKE Service names. Set a unique custom domain for each cluster, which ensures that all Service and Pod DNS records are unique within the VPC. This mode reduces communication friction between GKE and non-GKE resources. Refer to Using Cloud DNS for GKE and Restrictions and limitations in the Google Cloud documentation for more information.

You will learn about network policies later in this course.