**Service Topology:**

1. Service Topology enables a service to route traffic based upon the Node topology of the cluster.
2. For example, a service can specify that traffic be preferentially routed to endpoints that are on the same Node as the client, or in the same availability zone.
3. By default, traffic sent to a ClusterIP or NodePort Service may be routed to any backend address for the Service or in other words by default traffic can goes on any node.
4. But Kubernetes 1.7 made it possible to route "external" traffic to the Pods running on the same Node that received the traffic or same region or zone.
5. If your cluster has the Service Topology feature gate enabled, you can control Service traffic routing by specifying the topologyKeys field on the Service spec.
6. If topologyKeys is not specified or empty, no topology constraints will be applied.
7. Consider a cluster with Nodes that are labeled with their hostname, zone name, and region name. Then you can set the topologyKeys values of a service to direct traffic as follows.

Examples The following are common examples of using the Service Topology feature.

1. **Only Node Local Endpoints :** A Service that only routes to node local endpoints. If no endpoints exist on the node, traffic is dropped

1. **Prefer Node Local Endpoints :** A Service that prefers node local Endpoints but falls back to cluster wide endpoints if node local endpoints do not exist:

apiVersion: v1

kind: Service

metadata:

name: my-service

spec:

selector:

app: my-app

ports:

- protocol: TCP

port: 80

targetPort: 9376

topologyKeys:

- "kubernetes.io/hostname

1. **Only Zonal or Regional Endpoints :** A Service that prefers zonal then regional endpoints. If no endpoints exist in either, traffic is dropped

topologyKeys:

- "topology.kubernetes.io/zone"

- "topology.kubernetes.io/region"

1. **Prefer Node Local, Zonal, then Regional Endpoints:** A Service that prefers node local, zonal, then regional endpoints but falls back to cluster wide endpoints

topologyKeys:

- "kubernetes.io/hostname"

- "topology.kubernetes.io/zone"

- "topology.kubernetes.io/region"

- "\*"

**Constraints**

1. Service topology is not compatible with externalTrafficPolicy=Local, and therefore a Service cannot use both of these features.
2. It is possible to use both features in the same cluster on different Services, only not on the same Service.
3. Valid topology keys are currently limited to kubernetes.io/hostname, topology.kubernetes.io/zone, and topology.kubernetes.io/region, but will be generalized to other node labels in the future.
4. Topology keys must be valid label keys and at most 16 keys may be specified.
5. The catch-all value, "\*", must be the last value in the topology keys, if it is used/

**DNS for Services and Pods:**

1. Kubernetes creates DNS records for Services and Pods. You can contact Services with consistent DNS names instead of IP addresses.
2. Every Service defined in the cluster is assigned a DNS name.
3. By default, a client Pod's DNS search list includes the Pod's own namespace and the cluster's default domain.
4. If are pod on test namespace and service in prod namespace then without specifying the namespace it not possible that we can find service from the test namespace.
5. For more details about dns refer this link