TF DDF 2019: K8s enhancements for Telco applicability

Date

18 Nov 2019

Minutes

- Pragash and Yuvaraja presenting slides
 - tf-ddf-2019-k8s_enhancements_for_telco_applicability.pdf
- k8s to TF object mapping
 - Namespace single project or shared project
 - ° Pod VM
 - Service, Ingress, Network Policy each have their own analogs in TF
- A single TF controller can manage multiple different types of clusters
- Non-nested, nested (cluster inside of a cluster), standalone
- Namespaces with TF & k8s
 - ^o By default in k8s, all pods can talk to all other pods in the cluster
 - Can also isolate them via namespace
 - Each namespace maps to a Project in k8s
- Pod gets the IP from a controller...
 - $^{\circ}\,$ For every namespace in the cluster, create a virtual network
 - Start with two by default (pod network and service network)
 - ° To isolate things with namespaces, can create isolated networks accordingly
 - Example of how this looks in the TF GUI
 - Syed: When associated networks to a pod with a CRD, how do you ensure that the container in the network gets the appropriate route?
 Multi interfaces in the pod
 - Create the networks in the CRD
 - Queue manager then uses annotations to give things to the controller, that will set up the correct interfaces for the container
 - Syed: But how do you set the routing within the container?
 - To steer the package inside the container, add the static routes (manual process)
 - Plan to enhance this in the future
- Pod creation workflow
 - When a pod is created, TF kubemanager creates a VM, interfaces, and associates with the right vrouter (on the kubemanager side)
 Same for updated pod
 - Kubelet actually gets the same information
 - Creates a boss container w/pod, labels, etc
 - The kubelet triggers a call to the CNI
 - CNI calls the vrouter, which get the information from TF controller
- Multi-interface for pod
 - · A'la what Aniket mentioned in K8s enhancements for Telco applicability Kubernetes Service Chaining
 - When creating a pod, can specify in pod annotations how many networks, their names, etc
- Kubernetes Network policy
 - Which (groups of) pods are allowed to talk to each other and how they can communicate iwth other endpoints
 - k8s network policy was in TF4 then had to rethink it for TF5.
 - TF4: Security groups
 - TF5: Firewall security policy
 - Tags for labels, pods, etc
 - ° Framework to enforce access specification across workloads
 - workloads rep'd and grouped by tags
 - A lot easier to do intent-based
 - It's k8s native firewalling policy
 - k8s network policy constructs map directly to TF firewall policy constructs
 - Can customise network policies to shape the flow of traffic how you want/need
 - ° Currently default allow, but can deny by default by creating a network policy
- Service chaining
 - Basic functions covered in K8s enhancements for Telco applicability Kubernetes Service Chaining
- Live demo! (should the demo gods allow)

Action items