**Skupper Namespace Connection**

**Summary:** Formal system set-up. Connecting multiple kubernetes namespaces via Skupper, and abstracting details to its respective namespace.

**Prerequisite:** {SEE: Basic\_Skupper\_Setup}

**Log**

| Change Made | Signature |
| --- | --- |
| Created Initial Document | AG, SP |
|  |  |
|  |  |
|  |  |

Initializing Kube Clutster

Create a kubenertes cluster using KIND and export two different config files that will be multiple namespaces later. <This may be optional later but helpful for distinguishing the two namespaces>

| > kind create cluster --name test  export KUBECONFIG=~/.kube/config-one  kind export kubeconfig --name test  export KUBECONFIG=~/.kube/config-two  kind export kubeconfig --name test  Set kubectl context to "kind-test" |
| --- |

Initilizing Load Balencer

Install and properly set up the needed internal Load Balancer to give our Docker isntances a local IP

| > export KUBECONFIG=~/.kube/config-one  kubectl apply -f https://raw.githubusercontent.com/metallb/metallb/v0.12.1/manifests/namespace.yaml  kubectl apply -f https://raw.githubusercontent.com/metallb/metallb/v0.12.1/manifests/metallb.yaml  kubectl get pods -n metallb-system --watch  **Output >**  NAME READY STATUS RESTARTS AGE  controller-6658b8446c-57m5v 0/1 Running 0 13s  speaker-nbwgg 0/1 CreateContainerConfigError 0 13s |
| --- |

Get Docker Subnet

get a range of IP addresses for use in the load balancer for the docker containers

| > docker network inspect -f '{{.IPAM.Config}}' kind  [{172.19.0.0/16 172.19.0.1 map[]} {fc00:f853:ccd:e793::/64 fc00:f853:ccd:e793::1 map[]}] |
| --- |

Create yaml file

Using that CIDR range we got earlier, we want to set up the metal configuration file for generating the IPs. Below will be the 'metal.yaml' file you should create and modify the IP range at the bottom to what works in your system.

| > nano metal.yaml  apiVersion: v1  kind: ConfigMap  metadata:  namespace: metallb-system  name: config  data:  config: |  address-pools:  - name: default  protocol: layer2  addresses:  - 172.19.100.0-172.19.100.100 |
| --- |

Apply yaml file

| > kubectl apply -f metal.yaml **Output >** configmap/config created |
| --- |

Creating the namespaces

| > kubectl create namespace ns1  **>** kubectl create namespace ns2 |
| --- |

Initilize Skupper

| > skupper init -n ns1  **>** skupper init -n ns2  **Output >** Waiting 117 seconds for LoadBalancer IP or hostname...  Waiting 116 seconds for LoadBalancer IP or hostname...  Waiting 115 seconds for LoadBalancer IP or hostname...  Skupper is now installed in namespace 'ns1'. Use 'skupper status' to get more information. |
| --- |

Creating the Skupper Token

| > skupper token create ./SECRET.yaml --uses 100 -n ns1 |
| --- |

Creating the Skupper link

| > skupper link create ./SECRET.yaml -n ns2  **Output >** Site configured to link to https://172.19.100.1:8081/85cfa740-a4d4-11ed-a0e4-acde48001122 (name=link1)  Check the status of the link using 'skupper link status'. |
| --- |

Create the frontend

switch into the NS1 namespace and create the frontend of the site that will take our input

| > kubectl config set-context --current --namespace ns1  kubectl create deployment frontend --image gemajlia/basicwebcontainer:latest |
| --- |

Create the backend

Deploy the backend image we created onto the ns2 namespace.

| > kubectl config set-context --current --namespace ns2  kubectl create deployment backend --image gemajlia/teststrb:latest |
| --- |

Expose the backend

| > kubectl config set-context --current --namespace ns2  skupper expose deployment/backend --port 4123 |
| --- |

Expose the frontend

| > kubectl config set-context --current --namespace ns1  kubectl expose deployment/frontend --port 8080 --target-port 80 --type LoadBalancer |
| --- |

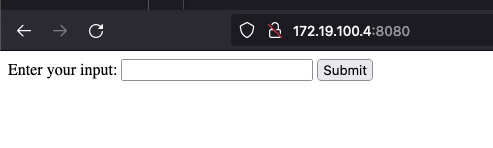
Test everything out!

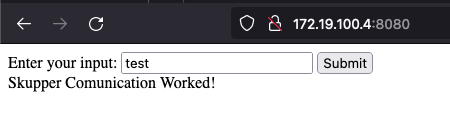
Get the IP of the frontend and load it into your browser

| > kubectl get service/frontend  **Output >**  NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE  frontend LoadBalancer 10.96.119.10 172.19.100.4 8080:31750/TCP 12m |
| --- |

Grab the IP under External IP and place that in your browser

| > http://172.19.100.4:8080 |
| --- |





If you get this message, you are successful at connecting two namespaces with Skupper