

```
1 # Source: https://gist.github.com/dc4ba562328c1d088047884026371f1f
2
3 #####
4 # Using Knative To Deploy And Manage Serverless Workloads #
5 #####
6
7 #####
8 # Installing Knative #
9 #####
10
11 # Docker Desktop (docker-5gb-4cpu.sh):
12   https://gist.github.com/bf30b06cbec9f784c4d3bb9ed1c63236)
13 # Minikube (minikube-5gb-4cpu.sh):
14   https://gist.github.com/1a2ffc52a53f865679e86b646502c93b)
15 # GKE (gke-simple.sh): https://gist.github.com/ebe4ad31d756b009b2e6544218c712e4)
16 # EKS (eks-simple.sh): https://gist.github.com/8ef7f6cb24001e240432cd6a82a515fd)
17 # AKS (aks-simple.sh): https://gist.github.com/f3e6575dcefcee039bb6cef6509f3fdc)
18
19 kubectl apply \
20   --filename https://github.com/knative/serving/releases/download/v0.19.0/serving-
21   crds.yaml
22
23 kubectl apply \
24   --filename https://github.com/knative/serving/releases/download/v0.19.0/serving-
25   core.yaml
26
27 kubectl --namespace knative-serving \
28   get pods
29
30 git clone \
31   https://github.com/vfarcic/devops-catalog-code.git
32
33 cd devops-catalog-code
34
35 git pull
36
37 cd knative/istio
38
39 istioctl install --skip-confirmation
40
41 kubectl --namespace istio-system \
42   get pods
43
44 kubectl label namespace knative-serving \
45   istio-injection=enabled
46
47 cat peer-auth.yaml
48
49 kubectl apply --filename peer-auth.yaml
50
51 kubectl apply \
52   --filename https://github.com/knative/net-
53   istio/releases/download/v0.19.0/release.yaml
54
55 # Only if Minikube
56 export INGRESS_IP=$(minikube ip)
57
58 # Only if Minikube
59 export INGRESS_PORT=$(kubectl \
60   --namespace istio-system \
```

```
56     get service istio-ingressgateway \
57     --output jsonpath='{.spec.ports[?(@.name=="http2")].nodePort}' )
58
59 # Only if Minikube
60 export INGRESS_HOST=$INGRESS_IP:$INGRESS_PORT
61
62 # Only if Docker Desktop
63 export INGRESS_HOST=127.0.0.1
64
65 # Only if GKE or AKS
66 export INGRESS_IP=$(kubectl \
67     --namespace istio-system \
68     get service istio-ingressgateway \
69     --output jsonpath='{.status.loadBalancer.ingress[0].ip}')
70
71 # Only if GKE or AKS
72 export INGRESS_HOST=$INGRESS_IP.xip.io
73
74 # Only if EKS
75 export INGRESS_HOST=$(kubectl \
76     --namespace istio-system \
77     get service istio-ingressgateway \
78     --output jsonpath='{.status.loadBalancer.ingress[0].hostname}')
79
80 kubectl --namespace knative-serving \
81     get configmap config-domain \
82     --output yaml
83
84 echo "apiVersion: v1
85 kind: ConfigMap
86 metadata:
87   name: config-domain
88   namespace: knative-serving
89 data:
90   $INGRESS_HOST: |
91 " | kubectl apply --filename -
92
93 kubectl --namespace knative-serving \
94     get pods
95
96 #####
97 # Painting The Big Picture #
98 #####
99
100 kubectl create namespace production
101
102 kubectl label namespace production \
103     istio-injection=enabled
104
105 kn service create devops-toolkit \
106     --namespace production \
107     --image vfaric/devops-toolkit-series \
108     --port 80
109
110 kubectl --namespace production \
111     get routes
112
113 # Only if Minikube, Docker Desktop, or EKS
114 curl -H "Host: devops-toolkit.production.example.com" \
115     http://$INGRESS_HOST
```

```
116
117 # Only if GKE or AKS
118 open http://devops-toolkit.production.$INGRESS_HOST
119
120 kubectl --namespace production \
121     get pods
122
123 # Only if Minikube, Docker Desktop, or EKS
124 curl -H "Host: devops-toolkit.production.example.com" \
125     http://$INGRESS_HOST
126
127 # Only if GKE or AKS
128 open http://devops-toolkit.production.$INGRESS_HOST
129
130 kn service delete devops-toolkit \
131     --namespace production
132
133 #####
134 # Defining Knative Applications As Code #
135 #####
136
137 cat devops-toolkit.yaml
138
139 kubectl --namespace production apply \
140     --filename devops-toolkit.yaml
141
142 # Only if Minikube, Docker Desktop, or EKS
143 curl -H "Host: devops-toolkit.production.example.com" \
144     http://$INGRESS_HOST
145
146 # Only if GKE or AKS
147 open http://devops-toolkit.production.$INGRESS_HOST
148
149 kubectl --namespace production \
150     get kservice
151
152 kubectl --namespace production \
153     get configuration
154
155 kubectl --namespace production \
156     get revisions
157
158 kubectl --namespace production \
159     get deployments
160
161 kubectl --namespace production \
162     get services,virtualservices
163
164 kubectl --namespace production \
165     get podautoscalers
166
167 kubectl --namespace production \
168     get routes
169
170 # Only if Minikube or EKS
171 kubectl run siege \
172     --image yokogawa/siege \
173     --generator run-pod/v1 \
174     -it --rm \
175     -- --concurrent 500 --time 60S \
```

```
176 --header "Host: devops-toolkit.production.example.com" \  
177 "http://$INGRESS_HOST" \  
178 && kubectl --namespace production \  
179 get pods  
180  
181 # Only if GKE or AKS  
182 kubectl run siege \  
183 --image yokogawa/siege \  
184 --generator run-pod/v1 \  
185 -it --rm \  
186 -- --concurrent 500 --time 60S \  
187 "http://devops-toolkit.production.$INGRESS_HOST" \  
188 && kubectl --namespace production \  
189 get pods  
190  
191 kubectl --namespace production \  
192 get pods  
193  
194 #####  
195 # Destroying The Resources #  
196 #####  
197  
198 kubectl --namespace production delete \  
199 --filename devops-toolkit.yaml  
200  
201 kubectl delete namespace production  
202  
203 cd ../../..  
204  
205 # Only if EKS  
206 kubectl --namespace istio-system \  
207 delete service istio-ingressgateway  
208
```