

Configuring Minikube Kubernetes cluster and running Zena Kubernetes task

1. INSTALL MINIKUBE CLUSTER

Follow the instructions from <https://minikube.sigs.k8s.io/docs/start/> to download the minikube binary for the OS of your choice and start the minikube cluster.

```
minikube start --apiserver-ips=<IP Address> --apiserver-name=<Server host name> --driver=none
```

2. INSTALL KUBECTL

For example, on an UBUNTU box "sudo apt update && sudo apt -y install kubectl"

3. Create Clusterrole to provide permissions for Kubernetes Cluster API Server access

```
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRole
metadata:
  name: api-permissions
  labels:
    rbac.example.com/aggregate-to-monitoring: "true"
rules:
- apiGroups: [""]
  resources: ["services", "endpoints", "pods", "pods/log"]
  verbs: ["get", "list", "watch", "create", "update", "patch", "delete"]
- apiGroups: ["extensions", "apps"]
  resources: ["deployments"]
  verbs: ["get", "list", "watch", "create", "update", "patch", "delete"]
- apiGroups: [""]
  resources: ["pods"]
  verbs: ["get", "list", "watch"]
- apiGroups: ["batch", "extensions"]
  resources: ["jobs"]
  verbs: ["get", "list", "watch", "create", "update", "patch", "delete"]
```

This yaml file provides API Server access to the following Kubernetes resource APIs: services, pods, deployments, jobs etc.

4. Bind the created clusterrole to the default user

```
kubectl create clusterrolebinding api-permissions --clusterrole api-permissions --serviceaccount=default:default
```

Note: Though we use the default user for convenience in this doc, any user can be used.

5. Obtain Cluster API server Token and create Web Login in Zena:

Create a 'Secret' token and apply it on the default service account of the Kubernetes cluster, then copy the token and use it in a Zena Web Login.

Follow official documentation: <https://kubernetes.io/docs/tasks/access-application-cluster/access-cluster/>

For example, on an Ubuntu server:

```
kubectl apply -f - <<EOF
apiVersion: v1
kind: Secret
metadata:
  name: default-token
  annotations:
    kubernetes.io/service-account.name: default
type: kubernetes.io/service-account-token
EOF
```

```
TOKEN=$(kubectl describe secret default-token | grep -E 'token' | cut -f2 -d':' | tr -d " ")
echo $TOKEN
```

Copy the printed TOKEN and create a Web Login in Zena using the "Uses API Key/Token" option.

5. Add the Kubernetes cluster API server's SSL certificate to the trusted CA certificate store of the Zena Agent

1) Get the certificate using a tool of your choice. For example, using openssl binary:
openssl s_client -connect waldevzenarp01.dev.rocketsoftware.com:8443 </dev/null | sed -ne '/-BEGIN CERTIFICATE-/,/-END CERTIFICATE-/p'
> waldevminikube.cert

2) Add the certificate to the Zena Agent's JVM truststore

```
keytool -import -trustcacerts -alias minikubeCA -file c:\dev\41\waldevminikube.cert -keystore "c:\Program Files\Java\jdk1.8.0_65\jre\lib\security\cacerts"
```

3) Restart Zena Agent

Now, the Zena Agent is ready to create resources on Kubernetes cluster. Use the Web Login created above on the login field of the Kubernetes task and use the yaml configuration you want to apply on the "YAML Specification" field.