Allan MacLean

allan@yellowzone.co.uk

ZIGLU SRE Tech Challenge

Deployment Instructions
The code installs from a single command into your Kubernetes, the output is in the logs.

kubectl apply -f https://raw.githubusercontent.com/yellowzoneallan/ziglu/master/ziglu.yaml

Backlog
Given the time constraints there are three small issues:

The output is for all namespaces not a specific namespace.
The slack channel code is present, but untested and commented out.
Would like to clarify with customer whether the app should continuously poll or run once.

Command line
To initially solve this problem via command line would have been my gut instinct, as this is a more of a coding challenge I'll complete the intended path, however this command might do the trick:

kubectl get deployments -n <namespace> | slack-cli -d devteam

Estimated steps
My key deliverable is to ensure it works from a single command to show my automation skills…

Check reference python libraries for similar
Write simple python code
Package as docker image
Place image on docker hub
Write yaml for install
Place yaml on GitHub
Lots of testing
Documentation of install
If time permits, evaluate slack bits

Write simple python code

The code works, although for all namespaces not current namespace, a simple grep later in the process can fix this. The slack stuff is prepped but not tested given time constraints. Quite a lot of time was spent working out the python API to call deployments as it wasn't in the coreAPI.

```
from kubernetes import client, config # kubernetes lib
    import os # os lib
    import slack # slack lib
    config.load_incluster_config() # load kubernetes config
    # client =
    slack.WebClient(token=os.environ['SLACK_API_TOKEN']) #
    setup slack
    v1a=client.AppsV1Api() # prep k8s api for listing
    deployments
    print("Listing deployments:")
    ret = v1a.list_deployment_for_all_namespaces(watch=False) #
    fetch all deployments
    for i in ret.items:
    print("%s\t%s\t%s" % (i.metadata.namespace,
    i.metadata.name, i.metadata.labels)) # print some info
    about each deployment
    # response =
    client.chat_postMessage(channel='#random',text=i.metadata.n
    ame) # slack to be tested
    # assert response["ok"]  # slack to be tested
```

Switch back to config.load_kube_config() from config.load_incluster_config() to run interactively.

```
[vagrant@master docker]$ python3 ziglu.py
Listing deployments:
default ziglu-deployment       {'app': 'ziglu'}
kube-system   calico-kube-controllers       {'k8s-app': 'calico-kube-controllers'}
kube-system   coredns       {'k8s-app': 'kube-dns'}
```

Package as docker image

As expected the container could not access the Kubernetes configuration, so the config.load_kube_config() failed. The solution was to switch the code to config.load_incluster_config() and setup RBAC. Had time allowed I would have spent time reducing the size of the container, the update to find the python3 package was extravagant. Port 5555 was permitted during testing but this could be removed.

```
FROM ubuntu:latest
    RUN apt-get update -y
    RUN apt-get install -y python3-pip
    RUN pip3 install kubernetes
    RUN pip3 install slackclient
    COPY . /app
    WORKDIR /app
    EXPOSE 5555
    CMD /usr/bin/python3 ./ziglu.py
```

sudo docker build --tag yellowzone/ziglu .

Place image on docker hub

To get the installation down to one command the docker image would have to be placed somewhere accessible to the target Kubernetes environment, a public repository in docker hub is ideal. This location would be referenced in the deployment yaml.

sudo docker login --username=yellowzone
sudo docker push yellowzone/ziglu

Write YAML for install

The initial deployment yaml was straight forward, however the RBAC setup to allow the container to access Kubernetes made this much more complex, and was split up into four sections.

Setup the service account to allow the deployment to call Kubernetes:

```
apiVersion: v1
    kind: ServiceAccount
    metadata:
    name: deployment-robot
```

`Setup a restricted readonly ClusterRole. Pod read access was permitted during testing but this could be removed.

```
apiVersion: rbac.authorization.k8s.io/v1
    kind: ClusterRole
    metadata:
    namespace: default
    name: deployment-role
    rules:
    - apiGroups: ["apps",""]
    resources: ["deployments","pods"]
    verbs: ["get", "list", "watch"]
```

The ClusterRole and the SeriveAccount were linked with a ClusterRoleBinding.

```
apiVersion: rbac.authorization.k8s.io/v1
    kind: ClusterRoleBinding
    metadata:
    name: deployment-rolebind
    namespace: default
    subjects:
    - kind: ServiceAccount
    name: deployment-robot # Name of the ServiceAccount
    namespace: default
    roleRef:
    kind: ClusterRole # This must be Role or ClusterRole
    name: deployment-role # This must match the name of the
    Role or ClusterRole you wish to bind to
    apiGroup: rbac.authorization.k8s.io
```

The Deployment references the image stored in docker hub. Port 5555 was permitted during testing but this could be removed.

```
apiVersion: apps/v1
    kind: Deployment
    metadata:
    name: ziglu-deployment
    labels:
    app: ziglu
    spec:
    replicas: 1
    selector:
    matchLabels:
    app: ziglu
    template:
    metadata:
    labels:
    app: ziglu
    spec:
    containers:
    - name: ziglu
    image: docker.io/yellowzone/ziglu:latest
    ports:
    - containerPort: 5555
    serviceAccountName: deployment-robot  # Name of the
    ServiceAccount
```

Place YAML on GitHub

All the files were placed on GitHub for Ziglu to review.

```
git init
git add *
git commit -m "Allan MacLean Ziglu test files"
git config --global user.name "yellowzoneallan"
git config --global user.email "allan@yellowzone.co.uk"
git remote add origin https://github.com/yellowzoneallan/ziglu.git
git push -u origin master
```

Minimal test

The minimal test was just execute the command on a simple Kubernetes setup.

```
[vagrant@master ~]$ kubectl apply -f https://raw.githubusercontent.com/yellowzoneallan/ziglu/master/ziglu.yaml
serviceaccount/deployment-robot created
clusterrole.rbac.authorization.k8s.io/deployment-role created
clusterrolebinding.rbac.authorization.k8s.io/deployment-rolebind created
deployment.apps/ziglu-deployment created
[vagrant@master ~]$ kubectl get pods
NAME                                READY   STATUS      RESTARTS   AGE
ziglu-deployment-5459495b7d-whrq9   0/1     Completed   2          82s
[vagrant@master ~]$ kubectl logs ziglu-deployment-5459495b7d-whrq9
Listing deployments:
default ziglu-deployment       {'app': 'ziglu'}
kube-system   calico-kube-controllers         {'k8s-app': 'calico-kube-controllers'}
kube-system   coredns          {'k8s-app': 'kube-dns'}
```

Full test

The full test was just execute the command on a bigger Kubernetes setup.

```
[vagrant@master ~]$ kubectl apply -f https://raw.githubusercontent.com/yellowzoneallan/ziglu/master/ziglu.yaml
serviceaccount/deployment-robot created
clusterrole.rbac.authorization.k8s.io/deployment-role created
clusterrolebinding.rbac.authorization.k8s.io/deployment-rolebind created
deployment.apps/ziglu-deployment created

[vagrant@master ~]$ kubectl get deployments
NAME                 READY   UP-TO-DATE   AVAILABLE   AGE
adservice            1/1     1            1           5m40s
cartservice          1/1     1            1           5m41s
checkoutservice      1/1     1            1           5m41s
currencyservice      1/1     1            1           5m41s
emailservice         0/1     1            0           5m41s
frontend             1/1     1            1           5m41s
loadgenerator        1/1     1            1           5m41s
paymentservice       1/1     1            1           5m41s
productcatalogservice   1/1     1            1           5m41s
recommendationservice   0/1     1            0           5m41s
redis-cart           1/1     1            1           5m40s
shippingservice      1/1     1            1           5m40s
ziglu-deployment     0/1     1            0           17s

[vagrant@master ~]$ kubectl get pods
NAME                                 READY   STATUS             RESTARTS   AGE
adservice-58c85c77d8-8nrcm           1/1     Running            0          5m47s
cartservice-579bdd6865-w9wzp         1/1     Running            0          5m48s
checkoutservice-66d68cbdd-msfjm      1/1     Running            0          5m48s
currencyservice-65dd85f486-czpcn     1/1     Running            0          5m47s
emailservice-84c98657cb-5tbpf        0/1     CrashLoopBackOff   5          5m48s
frontend-788f7bdc86-kdb85            1/1     Running            0          5m48s
loadgenerator-7699dc7d4b-k7d94       1/1     Running            2          5m48s
paymentservice-5c54c9887b-9fp85      1/1     Running            0          5m48s
productcatalogservice-7df777f796-7h62t   1/1     Running            0          5m48s
recommendationservice-89547cff8-2jscn    0/1     CrashLoopBackOff   5          5m48s
redis-cart-5f59546cdd-tnwns          1/1     Running            0          5m47s
shippingservice-778db496dd-qfqzr     1/1     Running            0          5m47s
ziglu-deployment-5459495b7d-w7n2b    1/1     Running            2          24s

[vagrant@master ~]$ kubectl logs ziglu-deployment-5459495b7d-w7n2b
Listing deployments:
default adservice       None
default cartservice     None
default checkoutservice         None
default currencyservice         None
default emailservice    None
default frontend        None
default loadgenerator None
default paymentservice          None
default productcatalogservice           None
default recommendationservice           None
default redis-cart      None
default shippingservice         None
default ziglu-deployment        {'app': 'ziglu'}
kube-system   calico-kube-controllers           {'k8s-app': 'calico-kube-controllers'}
kube-system   coredns           {'k8s-app': 'kube-dns'}
```