

Kubernetes Extended Cases

Author: Bokotei Oleksandr

GitHub: <https://github.com/nonamecoder2002/GLBaseCamp2021>

[Base]:

Task1: "Deploy go-demo-app & do troubleshoot"

At first let's grab go-demo-app from GitHub:

```
alex@DESKTOP-LBU2UOH:~/hw_5$ git clone https://github.com/den-vasyliev/go-demo-app.git
Cloning into 'go-demo-app'...
remote: Enumerating objects: 2740, done.
remote: Counting objects: 100% (210/210), done.
remote: Compressing objects: 100% (159/159), done.
remote: Total 2740 (delta 121), reused 117 (delta 48), pack-reused 2530
Receiving objects: 100% (2740/2740), 127.29 MiB | 2.15 MiB/s, done.
Resolving deltas: 100% (1483/1483), done.
alex@DESKTOP-LBU2UOH:~/hw_5$ ls -l
```

Before installing the app using HELM, we need to create namespace "demo" for that app manually:

```
alex@DESKTOP-LBU2UOH:~/go-demo-app$ kubectl get ns
NAME                STATUS    AGE
default             Active   55m
demo                Active   52m
kube-node-lease     Active   55m
kube-public         Active   55m
kube-system         Active   55m
```

Now try to install the app to the "demo" namespace:

```
alex@DESKTOP-LBU2U0H:~/go-demo-app$ helm install --namespace demo testapp ./helm/  
manifest_sorter.go:192: info: skipping unknown hook: "crd-install"  
manifest_sorter.go:192: info: skipping unknown hook: "crd-install"  
Error: unable to build kubernetes objects from release manifest: unable to recognize "": no matches for kind "NatsCluster" in version "nats.io/v1alpha2"  
alex@DESKTOP-LBU2U0H:~/go-demo-app$ |
```

After doing some research I managed to fix this error. The solution is to mkdir "crds" in app/helm/charts/nats & move "customresourcedefinition.yaml" from /helm/charts/nats/templates to "crds" dir created earlier

```
├── nats  
│   ├── Chart.yaml  
│   ├── README.md  
│   ├── config  
│   │   └── client-auth.json  
│   ├── crds  
│   │   └── customresourcedefinition.yaml  
│   ├── templates  
│   │   ├── NOTES.txt  
│   │   ├── _helpers.tpl  
│   │   ├── deployment.yaml  
│   │   ├── natscluster.yaml  
│   │   ├── rbac.yaml  
│   │   ├── secret.yaml  
│   │   └── serviceaccount.yaml  
│   └── values.yaml
```

After this, the installation goes successful the app is deployed

Workloads REFRESH DEPLOY DELETE

Cluster

Namespace

demo

RESET SAVE PREVIEW

Workloads are deployable units of computing that can be created and managed in a cluster.

Filter

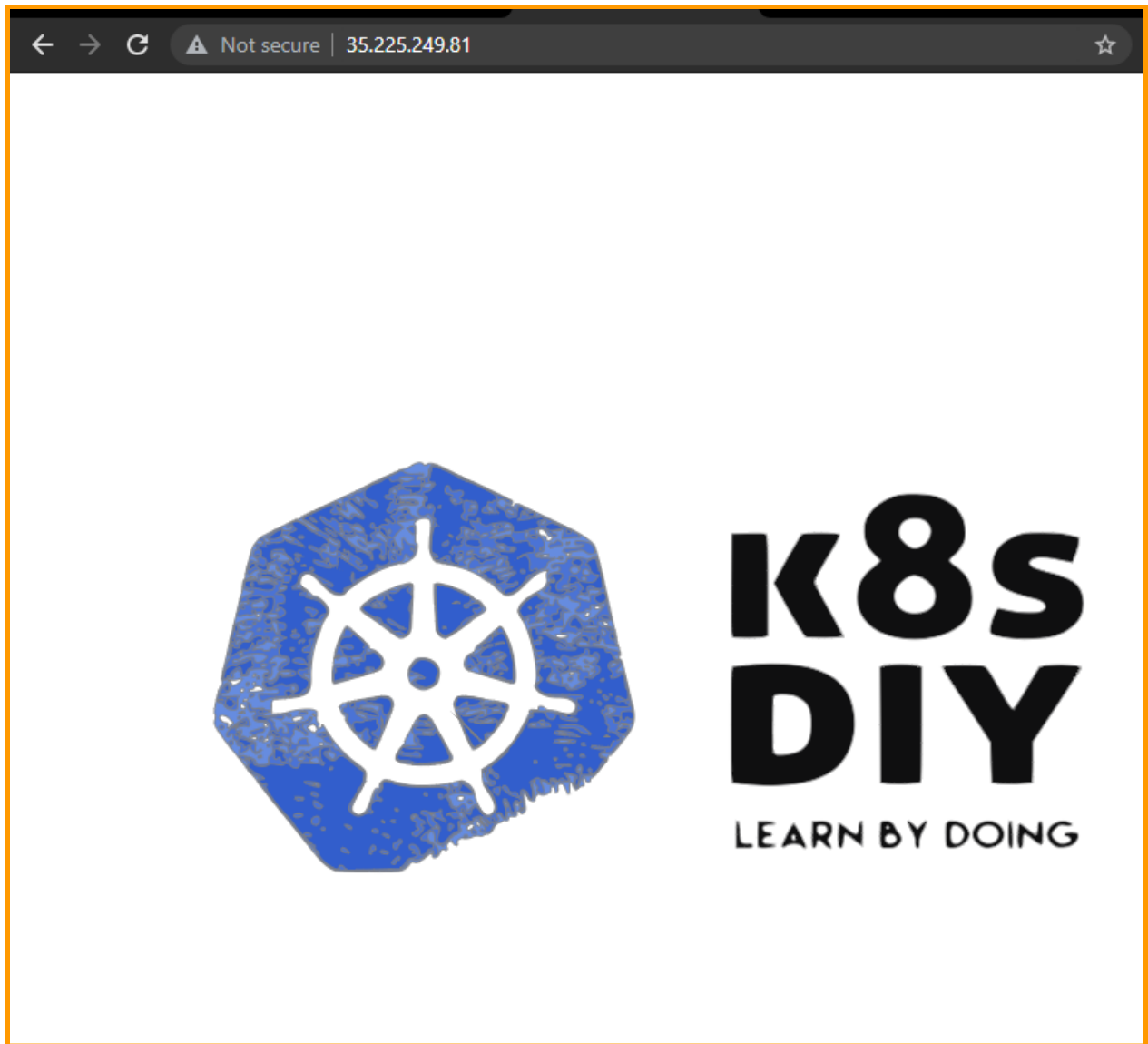
Is system object : False

 Filter workloads

<input type="checkbox"/>	Name	Status	Type	Pods	Namespace	Cluster
<input type="checkbox"/>	ambassador	OK	Deployment	1/1	demo	cluster-1
<input type="checkbox"/>	cache	OK	Deployment	1/1	demo	cluster-1
<input type="checkbox"/>	db	OK	Deployment	1/1	demo	cluster-1
<input type="checkbox"/>	nats-cluster-1	Running	Pod	1/1	demo	cluster-1
<input type="checkbox"/>	testapp-api	OK	Deployment	1/1	demo	cluster-1
<input type="checkbox"/>	testapp-ascii	OK	Deployment	1/1	demo	cluster-1
<input type="checkbox"/>	testapp-data	OK	Deployment	1/1	demo	cluster-1
<input type="checkbox"/>	testapp-front	OK	Deployment	1/1	demo	cluster-1
<input type="checkbox"/>	testapp-img	OK	Deployment	1/1	demo	cluster-1
<input type="checkbox"/>	testapp-nats-operator	OK	Deployment	1/1	demo	cluster-1

```
alex@DESKTOP-LBU2UOH:~/hw_5/go-demo-app$ kubectl get svc -o wide -n demo
NAME                TYPE                CLUSTER-IP      EXTERNAL-IP      PORT(S)
ambassador           LoadBalancer        10.8.15.124     35.225.249.81   80:32330/TCP
ambassador-admin     ClusterIP            10.8.11.17      <none>           8877/TCP
cache                ClusterIP            10.8.15.193     <none>           6379/TCP
db                   ClusterIP            10.8.7.20       <none>           3306/TCP
nats-cluster         ClusterIP            10.8.11.58      <none>           4222/TCP
nats-cluster-mgmt    ClusterIP            None            <none>           6222/TCP,8222/TCP,7777/TCP
testapp-api          ClusterIP            10.8.12.234     <none>           80/TCP
testapp-ascii        ClusterIP            10.8.3.48       <none>           80/TCP
testapp-data         ClusterIP            10.8.2.132      <none>           80/TCP
testapp-front        ClusterIP            10.8.15.139     <none>           80/TCP
testapp-img          ClusterIP            10.8.0.180      <none>           80/TCP
alex@DESKTOP-LBU2UOH:~/hw_5/go-demo-app$ |
```

Now open-up browser and go to ambassador EXTERNAL IP:



To verify that the app works correctly let's run the following commands from the terminal:

```
alex@DESKTOP-LBU2UOH:~/hw_5/go-demo-app$ wget -O /tmp/g.png https://www.google.com/images/branding/googlelogo/1x/googlelogo_color_272x92dp.png
```

```
alex@DESKTOP-LBU2UOH:~/hw_5/go-demo-app$ curl -F 'image=@/tmp/g.png' curl -F 'image=@/tmp/g.png'
' 35.225.249.81/img/
curl: (6) Could not resolve host: curl
.;1ttttli.
if1:....:i.
iL;
ff. .1tt11: 11::11, if1::if1 .tf::;tL1 1; i1::;t1.
;Li .tL.it t1,Li ;L,,L. fi 1: t;::;,.
;fti::,,;tf: ;ti,,;t:;tf::,ff..ft::,iLi 1; 1t::,::
:i1111i: ;iii;, :1111;. ;11i;L1 ;, ;:iii:
.11,,1L:
,i111i.
```

As we can see, the app works correctly

Task 2: "Install Elastic on kuber cluster"

Let's create elastic-operator & its crds:

```
alex@DESKTOP-LBU2UOH:~/hw_5/go-demo-app$ kubectl apply -f https://download.elastic.co/downloads/eck/1.6.0/all-in-one.yaml
namespace/elastic-system created
serviceaccount/elastic-operator created
secret/elastic-webhook-server-cert created
configmap/elastic-operator created
Warning: apiextensions.k8s.io/v1beta1 CustomResourceDefinition is deprecated in v1.16+, unavailable in v1.22+; use apiextensions.k8s.io/v1
customresourcedefinition.apiextensions.k8s.io/agents.agent.k8s.elastic.co created
customresourcedefinition.apiextensions.k8s.io/apmservers.apm.k8s.elastic.co created
customresourcedefinition.apiextensions.k8s.io/beats.beat.k8s.elastic.co created
customresourcedefinition.apiextensions.k8s.io/elasticmapsservers.maps.k8s.elastic.co created
customresourcedefinition.apiextensions.k8s.io/elasticsearches.elasticsearch.k8s.elastic.co created
customresourcedefinition.apiextensions.k8s.io/enterprisesearches.enterprisesearch.k8s.elastic.co created
customresourcedefinition.apiextensions.k8s.io/kibanas.kibana.k8s.elastic.co created
clusterrole.rbac.authorization.k8s.io/elastic-operator created
clusterrole.rbac.authorization.k8s.io/elastic-operator-view created
clusterrole.rbac.authorization.k8s.io/elastic-operator-edit created
clusterrolebinding.rbac.authorization.k8s.io/elastic-operator created
service/elastic-webhook-server created
statefulset.apps/elastic-operator created
```

Then let's deploy ElasticSearch:

```
cat <<EOF | kubectl apply -f -
apiVersion: elasticsearch.k8s.elastic.co/v1
kind: Elasticsearch
metadata:
  name: quickstart
spec:
  version: 7.13.1
  nodeSets:
  - name: default
    count: 1
    config:
      node.store.allow_mmap: false
EOF
```

```
alex@DESKTOP-LBU2UOH:~/hw_5/go-demo-app$ kubectl get elasticsearch -A
NAMESPACE   NAME           HEALTH   NODES   VERSION   PHASE           AGE
default     quickstart     unknown   1       7.13.1    ApplyingChanges  4s
```

Then deploy Kibana:

```
cat <<EOF | kubectl apply -f -
apiVersion: kibana.k8s.elastic.co/v1
kind: Kibana
metadata:
  name: quickstart
spec:
  version: 7.13.1
  count: 1
  elasticsearchRef:
    name: quickstart
EOF
```

```
alex@DESKTOP-LBU2UOH:~/hw_5/go-demo-app$ kubectl get svc
```

NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE
kubernetes	ClusterIP	10.8.0.1	<none>	443/TCP	74m
quickstart-es-default	ClusterIP	None	<none>	9200/TCP	50m
quickstart-es-http	ClusterIP	10.8.9.226	<none>	9200/TCP	50m
quickstart-es-transport	ClusterIP	None	<none>	9300/TCP	50m
quickstart-kb-http	ClusterIP	10.8.6.46	<none>	5601/TCP	5s

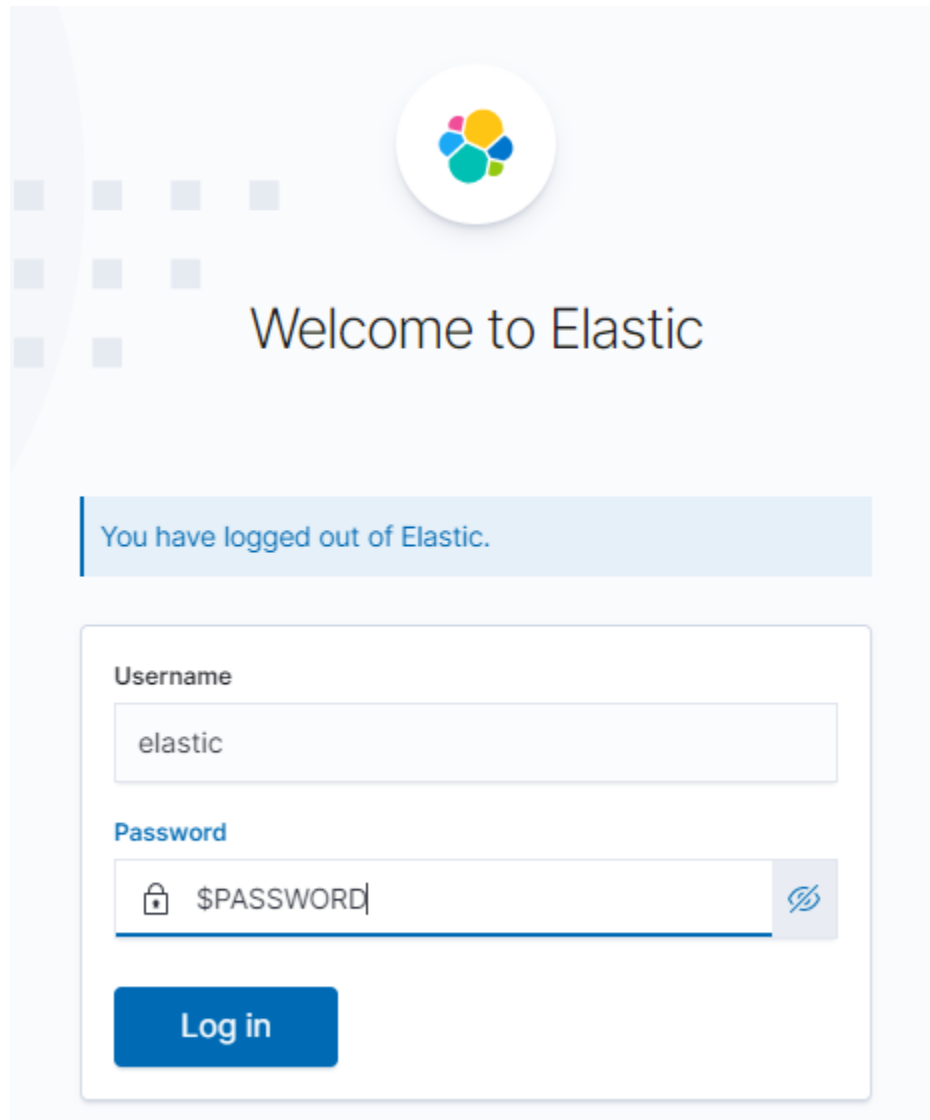
Then retrieve the credentials:

```
PASSWORD=$(kubectl get secret quickstart-es-elastic-user\
-o go-template='{{.data.elastic | base64decode}}')
```

Then port-forward kibana svc:

```
alex@DESKTOP-LBU2UOH:~/hw_5/go-demo-app$ kubectl port-forward svc/quickstart-kb-http 5601
Forwarding from 127.0.0.1:5601 -> 5601
Forwarding from [::1]:5601 -> 5601
Handling connection for 5601
Handling connection for 5601
Handling connection for 5601
Handling connection for 5601
Handling connection for 5601
Handling connection for 5601
Handling connection for 5601
Handling connection for 5601
Handling connection for 5601
Handling connection for 5601
Handling connection for 5601
Handling connection for 5601
```

Open-up browser & go to localhost:5601 port:



The image shows the Elasticsearch login interface. At the top, there is a circular logo with five colored dots (yellow, green, blue, red, and purple) arranged in a circle. Below the logo, the text "Welcome to Elastic" is displayed. A light blue banner below the welcome message states "You have logged out of Elastic." The login form is a white box with a blue border. It contains two input fields: "Username" with the value "elastic" and "Password" with a masked value "\$PASSWORD". A blue "Log in" button is at the bottom of the form. The background is a light gray with a subtle pattern of small squares.

Welcome to Elastic

You have logged out of Elastic.

Username

elastic

Password

\$PASSWORD

Log in

Login using the retrieved credentials:

