

Alibaba Serverless Kubernetes Demo with mongo-express and mongo-db using Terraform and Kubectl

Refs:

- 'Kubernetes Tutorial for Beginners [FULL COURSE in 4 Hours]'
<https://www.youtube.com/watch?v=X48VuDVv0do&t=8507s>
- Example YAML configuration files <https://gitlab.com/nanuchi/youtube-tutorial-series/-/tree/master/demo-kubernetes-components>
- Terraform Registry example code:
https://registry.terraform.io/providers/aliyun/alicloud/latest/docs/resources/cs_serverless_kubernetes#version

Overview:

The purpose of this demo is to leverage Terraform and Kubectl to setup a K8S deployment/service including mongo-express and mongo-db components with Alibaba Serverless Kubernetes or ASK service.

At the end of this demo, user can access to the mongo-express web portal with the public IP.



Server Status

Turn on admin in config.js to view server stats!


High level steps are:

- A to C, install the Visual Studio Code, Terraform and kubectl tools;
- D to E, configure the Terraform with API access key and secret;
- F to G, provision the Serverless Kubernetes Cluster with Terraform and configure the kubectl with cluster connection information;
- H, provision the mongo-db and mongo-express deployments, services and SLB for public access;
- I, clean up the demo setups;

Pre-conditions:

- 1) Have cloud account been created on www.alibabacloud.com;
- 2) Have RAM user with API key/secret and Admin permissions;
- 3) Stable internet connection;

A. Install Visual Studio Code and extensions

- 1) Download and Install VSC via <https://code.visualstudio.com/>
- 2) Open VSC and locate  for extensions on the left-hand side panel;
- 3) Search and install 'HashiCorp Terraform' and 'YAML' extensions;

B. Install Terraform (An Infrastructure As Code tool from HashiCorp)

- 1) Download and Install Terraform from <https://www.terraform.io/downloads.html>
- 2) Check the Aliyun provider page for Terraform registry <https://registry.terraform.io/providers/aliyun/alicloud/latest> and locate the



on the web page. Click it and copy the codes from 'terraform' to the end;

- 3) Create a new directory 'tf' and change directory to it by 'cd tf'
- 4) Create a new file name 'main.tf' and paste the content copied in step 2), save the file;

C. Install kubectl (K8S CLI tool)

- 1) Download and Install kubectl from <https://kubernetes.io/docs/tasks/tools/>
- 2) Open terminal (Mac/Linux) or CMD (Windows), change to your home directory by typing 'cd ~'
- 3) Make a new directory by typing 'mkdir .kube' (Mac/Linux) or 'md .kube' (Windows)
- 4) 'cd .kube' and create a new file save it as 'config'

D. Create RAM user for Alibaba Cloud

- 1) Sign in the <https://homenew-intl.console.aliyun.com/>
- 2) Select RAM service

← → ↻ 🏠 homenew-intl.console.aliyun.com

✕ Alibaba Cloud | 🏠 Workbench 🔍 Search...

📦 Products and Services >

- 🚀 Auto Scaling
- 🏠 Container Service for Kubern...
- 🌐 Global Accelerator
- 🛡️ Web Application Firewall
- ⚙️ Anti-DDoS Pro
- 🛡️ Traffic Security (Anti-DDoS ...)
- 🛡️ Cloud Firewall
- 🔗 Elastic IP Address
- 🏠 Server Load Balancer
- ☁️ Virtual Private Cloud

🔍 RAM

Recent Visits

| | | | |
|----------------------------------|---|----------------------|---|
| Container Service for Kubernetes | ★ | Server Load Balancer | ★ |
| Elastic Container Instance | | Alibaba Cloud DNS | ★ |

A total of 3 products related to **ram** are found.

Analytics

Data Computing & Analytics

Hologres

Management and Governance

Resource Access Management (Select RAM service)

CloudSSO

3) Create RAM user

4) Select 'User' and 'Create User'

RAM / Users

Users

📘 A RAM user is an identity entity. It represents a user or application in your orga

You can manage users in the following steps:

1. Create a RAM user, and set a password for this user to log on to the consc
2. Add the user to a group. To perform this operation, you must have created

Create User 🔍 Enter a logon username, display name, user ID, or Ac

RAM

- Overview
- Identities
 - Users**
 - Groups
 - Roles
 - Settings
- SSO

5) We need only the Open API access mode for Terraform configuration

← Create User

User Account Information

* Logon Name ?

asktestuser

@5346426141302644.onaliyun.com

* Display Name ?

asktestuser

+ Add User

Access Mode ?

☐ Console Access Users access the Alibaba Cloud console using the account and password.

☒ Open API Access Enable AccessKeyId and AccessKeySecret to support access through the API or other development tools.

OK

Return

6) Download the 'AccessKey ID' and 'AccessKey Secret' for Terraform configuration

User Information

Download CSV File

| <input type="checkbox"/> | User Logon Name | Status | Logon Pa |
|--------------------------|---|--|----------|
| | | Create User:● Success | |
| <input type="checkbox"/> | asktestuser@5346426141302644.onaliyun.com | Enable API Call-based Access:● Success | N/A |
| <input type="checkbox"/> | | | |
| <input type="checkbox"/> | Add to Group | Add Permissions | |

7) Go back to 'Users' and select the RAM user we just created and assign the permissions to it, to make the demo easier we authorize the 'AdministrationAccess' to this RAM user which might not be the best practice in the real life scenarios;

| <input type="checkbox"/> | User Logon Name/Display Name | Note | Last Login Date 1↓ | Created 1↓ | Actions |
|--------------------------|--|------|--------------------|-----------------------|---|
| <input type="checkbox"/> | asktestuser@5346426141302644.onaliyun.com asktestuser | | - | Nov 1, 2021, 15:03:24 | Add to Group Delete Add Permissions |

ADD PERMISSIONS

☒ Alibaba Cloud Account
☐ Specific Resource Group

Enter a resource group name. ▼

* Principal

asktestuser@5346426141302644.onaliyun.com ×

* Select Policy

[+ Create Policy](#)

Enter a policy name. ↺

| Authorization Policy Name | Description |
|---------------------------|--|
| AdministratorAccess | Provides full access to Alibaba Cloud service... |

Selected (1) Clear

AdministratorAccess ×

E. Configure the Terraform tool

- 1) Go back to the Terraform directory 'tf' and edit the 'main.tf' file with adding access_key and secret_key and region information as following, save it;

```
//setup of providers
terraform {
  required_providers {
    alicloud = {
      source = "aliyun/alicloud"
      version = "1.140.0"
    }
  }
}

//1) setup of alicloud RAM account with resource permissions and obtain the access_key and access_key_secret
provider "alicloud" {
  access_key = var.my_access_key
  secret_key = var.my_secret_key
  region     = var.my_region
}
```

- 2) Under the same directory create a new file 'terraform.tfvars' and add the RAM user's 'AccessKey ID' and 'AccessKey Secret', save it. **DO NOT share this file if you want to share the code!!**

```
terraform.tfvars > ...
1  my_access_key = ""
2  my_secret_key = ""
```

- 3) Under the same directory create a new file 'variables.tf' and edit it as following with 'sensitive = true' the 'AccessKey ID' and 'AccessKey Secret' will not be shown as the outputs on the terminal or logs;

```

variables.tf > variable "my_ask_name" > default
12
    1 reference
13 variable "my_access_key" {
14     description = "RAM user access_key"
15     sensitive   = true
16 }
17
    1 reference
18 variable "my_secret_key" {
19     description = "RAM user secret_key"
20     sensitive   = true
21 }

```

F.Provision the Serverless Kubernetes Cluster with Terraform

Please use following example files with your own 'terraform.tfvars' to create your own ASK cluster. https://github.com/rickyang1978/Mongo_ASK_demo

 main.tf

 variables.tf

Let's investigate the key contents of the main.tf and some of the key variables defined in the variables.tf

- 1) Include the zone data with following code. According to <https://www.alibabacloud.com/help/doc-detail/40654.htm>
The region 'cn-hongkong' is selected for this demo, there're 3 zones in this region and they're Zone B,C and D;

```

variable "my_region" {
    default = "cn-hongkong"
}

data "alicloud_zones" "abc_zones" {}

```

- 2) Create one VPC with CIDR block

```
//3) setup of VPC
3 references
resource "alicloud_vpc" "test_vpc" {
  vpc_name    = var.my_vpc
  cidr_block  = var.my_vpc_cidr_block
}

1 reference
variable "my_vpc_cidr_block" {
  default = "10.1.0.0/21"
}
```

- 3) Create two vSwitches within the VPC and the corresponding CIDR blocks.
zone_id '0' and '2' are referring to zone B and D in cn-hongkong region;

```
//4) setup of vswitch within the VPC
1 reference
resource "alicloud_vswitch" "test_vswitch" {
  vswitch_name = var.my_vswitch
  vpc_id       = alicloud_vpc.test_vpc.id
  cidr_block   = var.my_vswitch_cidr_block
  zone_id      = data.alicloud_zones.abc_zones.zones.0.id
}

1 reference
resource "alicloud_vswitch" "test_vswitch_bak" {
  vswitch_name = var.my_vswitch_bak
  vpc_id       = alicloud_vpc.test_vpc.id
  cidr_block   = var.my_vswitch_cidr_block_bak
  zone_id      = data.alicloud_zones.abc_zones.zones.2.id
}

1 reference
variable "my_vswitch_cidr_block" {
  default = "10.1.0.0/24"
}

1 reference
variable "my_vswitch_cidr_block_bak" {
  default = "10.1.1.0/24"
}
```

- 4) Create the ASK cluster by specifying the version of cluster, VPC, vswitch info, other parameters are default;

```
resource "alicloud_cs_serverless_kubernetes" "serverless" {
  name           = var.my_ask_name
  version        = "v1.20.4-aliyun.1"
  vpc_id         = alicloud_vpc.test_vpc.id
  vswitch_ids    = [alicloud_vswitch.test_vswitch.id,alicloud_vswitch.test_vswitch_bak.id]
  new_nat_gateway = true
  endpoint_public_access_enabled = true
  deletion_protection = false
  load_balancer_spec = "slb.s2.small"
  time_zone       = "Asia/Shanghai"
  service_cidr    = "172.21.0.0/20"
  service_discovery_types = ["PrivateZone"]
}
```

- 5) Open the terminal or CMD and change to the directory containing the terraform files;
- 6) Initialized the Terraform with Alibaba Provider information with command 'terraform init'

```
Initializing the backend...

Initializing provider plugins...
- Finding aliyun/alicloud versions matching "1.140.0"...
- Installing aliyun/alicloud v1.140.0...
- Installed aliyun/alicloud v1.140.0 (signed by a HashiCorp partner, key ID 47422B4AA9FA381B)

Partner and community providers are signed by their developers.
If you'd like to know more about provider signing, you can read about it here:
https://www.terraform.io/docs/cli/plugins/signing.html

Terraform has created a lock file .terraform.lock.hcl to record the provider
selections it made above. Include this file in your version control repository
so that Terraform can guarantee to make the same selections by default when
you run "terraform init" in the future.

Terraform has been successfully initialized!

You may now begin working with Terraform. Try running "terraform plan" to see
any changes that are required for your infrastructure. All Terraform commands
should now work.

If you ever set or change modules or backend configuration for Terraform,
rerun this command to reinitialize your working directory. If you forget, other
commands will detect it and remind you to do so if necessary.
```

- 7) Validate and make a plan for provisioning with command 'terraform plan -out=test1'
- You should get the similar outputs without Error messages if the code is correct;


```

Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following symbols:
+ create

Terraform will perform the following actions:

# alicloud_cs_serverless_kubernetes.serverless will be created
+ resource "alicloud_cs_serverless_kubernetes" "serverless" {
  + deletion_protection = false
  + endpoint_public_access_enabled = true
  + force_update         = false
  + id                  = (known after apply)
  + load_balancer_spec   = "slb.s2.small"
  + logging_type         = "SLS"
  + name                 = "ask001"
  + new_nat_gateway      = true
  + resource_group_id    = (known after apply)
  + security_group_id    = (known after apply)
  + service_cidr         = "172.21.0.0/20"
  + service_discovery_types = [
    + "PrivateZone",
  ]
  + sls_project_name     = (known after apply)
  + time_zone            = "Asia/Shanghai"
  + version              = "v1.20.4-aliyun.1"
  + vpc_id               = (known after apply)
  + vswitch_ids          = (known after apply)

  + addons {
    + config = (known after apply)
    + disabled = (known after apply)
    + name   = (known after apply)
  }
}

# alicloud_vswitch.test_vswitch_bak will be created
+ resource "alicloud_vswitch" "test_vswitch_bak" {
  + availability_zone = (known after apply)
  + cidr_block       = "10.1.1.0/24"
  + id               = (known after apply)
  + name             = (known after apply)
  + status           = (known after apply)
  + vpc_id           = (known after apply)
  + vswitch_name     = "test_vswitch_bak"
  + zone_id          = "cn-hongkong-d"
}

Plan: 4 to add, 0 to change, 0 to destroy.

Saved the plan to: test1

To perform exactly these actions, run the following command to apply:
  terraform apply "test1"

```

- 8) Apply the validated code to provision the cloud resources to the Alibaba Cloud Platform
 'terraform apply' and confirm it with 'yes'

```
Plan: 4 to add, 0 to change, 0 to destroy.
```

Do you want to perform these actions?

Terraform will perform the actions described above.
 Only 'yes' will be accepted to approve.

Enter a value:

Then it will take about 3 minutes to complete the ASK cluster provisioning.

```

Enter a value: yes
alicloud_vpc.test_vpc: Creating...
alicloud_vpc.test_vpc: Creation complete after 7s [id=vpc-j6cfvx37oa0hnf085ngbi]
alicloud_vswitch.test_vswitch: Creating...
alicloud_vswitch.test_vswitch_bak: Creating...
alicloud_vswitch.test_vswitch: Creation complete after 6s [id=vsw-j6cnazzdoyz9a43cz5ljh]
alicloud_vswitch.test_vswitch_bak: Creation complete after 9s [id=vsw-j6culscso9tksvkq7urit]
alicloud_cs_serverless_kubernetes.serverless: Creating...
alicloud_cs_serverless_kubernetes.serverless: Still creating... [10s elapsed]
alicloud_cs_serverless_kubernetes.serverless: Still creating... [20s elapsed]
alicloud_cs_serverless_kubernetes.serverless: Still creating... [30s elapsed]
alicloud_cs_serverless_kubernetes.serverless: Still creating... [40s elapsed]
alicloud_cs_serverless_kubernetes.serverless: Still creating... [50s elapsed]
alicloud_cs_serverless_kubernetes.serverless: Still creating... [1m0s elapsed]
alicloud_cs_serverless_kubernetes.serverless: Still creating... [1m10s elapsed]
alicloud_cs_serverless_kubernetes.serverless: Still creating... [1m20s elapsed]
alicloud_cs_serverless_kubernetes.serverless: Still creating... [1m29s elapsed]
alicloud_cs_serverless_kubernetes.serverless: Still creating... [1m39s elapsed]
alicloud_cs_serverless_kubernetes.serverless: Still creating... [1m49s elapsed]
alicloud_cs_serverless_kubernetes.serverless: Still creating... [1m59s elapsed]
alicloud_cs_serverless_kubernetes.serverless: Still creating... [2m9s elapsed]
alicloud_cs_serverless_kubernetes.serverless: Still creating... [2m19s elapsed]
alicloud_cs_serverless_kubernetes.serverless: Still creating... [2m29s elapsed]
alicloud_cs_serverless_kubernetes.serverless: Still creating... [2m39s elapsed]
alicloud_cs_serverless_kubernetes.serverless: Still creating... [2m49s elapsed]
alicloud_cs_serverless_kubernetes.serverless: Still creating... [2m59s elapsed]
alicloud_cs_serverless_kubernetes.serverless: Still creating... [3m9s elapsed]
alicloud_cs_serverless_kubernetes.serverless: Creation complete after 3m19s [id=cb40214127ff04b63b91033250274bd8e]

Apply complete! Resources: 4 added, 0 changed, 0 destroyed.

```

- 9) Go back to Alibaba Cloud console to verify the provisioning is successfully done. Search the 'Container Service for Kubernetes' with keyword 'ack'

The screenshot shows the Alibaba Cloud console interface. On the left is a sidebar with 'Products and Services' and a list of services including Auto Scaling, Container Service for Kubernetes, Global Accelerator, Web Application Firewall, Anti-DDoS Pro, Traffic Security, and Cloud Firewall. The main area has a search bar with 'ack' entered. Below the search bar, 'Recent Visits' lists 'Resource Access Management', 'Elastic Container Instance', 'Container Service for Kubernetes' (marked with a star), and 'Server Load Balancer'. A message states 'A total of 3 products related to ack are found.' Below this, there are three categories: 'Elastic Computing', 'Containers', and 'Databases'. Under 'Containers', 'Container Service for Kubernetes' is highlighted with a red box and a star. Other categories like 'Utilities & Tools' and 'Database Backup Service' are also visible.

- 10) Click the Serverless Kubernetes Cluster Name and select 'Connection Information' then copy the contents for kubectl configurations

The screenshot shows the 'Container Service for Kubernetes' (ACK) console. At the top, there's a breadcrumb 'ACK / Clusters' and a 'Shapes' button. The main heading is 'Clusters'. Below it, there's a search bar and a 'Labels' button. A table lists the clusters. The first cluster, 'ask001' with ID 'cb40214127ff04b63b91033250274bd8e', is highlighted with a red box. It is a 'Serverless Kubernetes' cluster in the 'China (Hong Kong)' region, with a status of 'Running'. The table also shows 'Nodes', 'Usage', 'Created At' (Nov 1, 2021, 16:10:55 UTC+8), and 'Version' (v1.20.4-aliyun.1). To the right of the cluster name, there are links for 'Details', 'Applications', 'View Logs', 'Upgrade Cluster', and 'More'.

ask001

Refresh

Open Cloud Shell

Overview

Basic Information

Connection Information

Cluster Resources

Cluster Logs

Connect to a Kubernetes cluster using kubectl

1. Install and configure kubectl. For more information, see [Install and configure kubectl](#).
2. Configure cluster credentials:

Public Access

Internal Access

Generate Temporary kubeconfig

Revoke KubeConfig

Copy the following content to \$HOME/.kube/config on your local computer.

Credentials Expire At: Oct 31, 2024, 16:14:08 UTC+8

```
apiVersion: v1
clusters:
- cluster:
```

Copy

G. Configure the kubectl to connect to the Kubernetes cluster

- 1) Change to the your .kube directory created in step C.3) and paste the content copied from step F.10) to the 'config' file and save it;
- 2) Open terminal or CMD and type 'kubectl cluster-info', you should get following outputs.

```
Kubernetes control plane is running at https://[redacted]
To further debug and diagnose cluster problems, use 'kubectl cluster-info dump'.
```

H. Provision Kubernetes deployments and services with kubectl and YAML configuration files.

The configuration files of mongo-db and mongo-express are referenced from 'Kubernetes Tutorial for Beginners [FULL COURSE in 4 Hours]'

<https://www.youtube.com/watch?v=X48VuDVv0do&t=8507s>

and

<https://gitlab.com/nanuchi/youtube-tutorial-series/-/tree/master/demo-kubernetes-components>

I make some minor updates and upload them to

https://github.com/rickyang1978/Mongo_ASK_demo

1_secret.yaml

2_mongo.yaml

3_mongo-configmap.yaml

4_mongo-express-internal.yaml

- 1) Create the mongo-db's username and password in BASE64 format and save it with 1_secret.yaml, (To generate the BASE64 coded text just type 'echo -n 'THE INFO NEED TO BE CODED' | base64')
- 2) Configure the secret information in the cluster with 'kubectl apply -f 1_secret.yaml'
- 3) Configure the mongo-db deployment and service with 'kubectl apply -f 2_mongo.yaml'
- 4) Configure the config-map information allowing mongo-db being connected via database_url with 'kubectl apply -f 3_mongo-configmap.yaml'
- 5) Configure the mong-express deployment and service with 'kubectl apply -f 4_mongo-express-internal.yaml'
- 6) Verify the configurations are successfully done with 'kubectl get pods'

| NAME | READY | STATUS | RESTARTS | AGE |
|------------------------------------|-------|---------|----------|-------|
| mongo-express-78fcf796b8-2spf9 | 1/1 | Running | 0 | 2m20s |
| mongodb-deployment-8f6675bc5-hdb94 | 1/1 | Running | 0 | 5m26s |

'kubectl get deployment'

| NAME | READY | UP-TO-DATE | AVAILABLE | AGE |
|--------------------|-------|------------|-----------|-------|
| mongo-express | 1/1 | 1 | 1 | 3m22s |
| mongodb-deployment | 1/1 | 1 | 1 | 6m28s |

'kubectl get service'

| NAME | TYPE | CLUSTER-IP | EXTERNAL-IP | PORT(S) | AGE |
|-----------------------|-----------|--------------|-------------|-----------|-------|
| kubernetes | ClusterIP | 172.21.0.1 | <none> | 443/TCP | 49m |
| mongo-express-service | ClusterIP | 172.21.0.171 | <none> | 8081/TCP | 4m11s |
| mongodb-service | ClusterIP | 172.21.4.175 | <none> | 27017/TCP | 7m17s |

- 7) To access the mongo-express portal from Internet, we need to create a SLB with public IP and connect to the backend IP 172.21.0.171 and port 8081. And this need to be done via Console.

Select 'Services' and click 'Create'

The screenshot shows the Kubernetes console interface. On the left sidebar, 'Services' is selected under the 'Network' section. The main panel displays the 'Services' page with a search bar and a table of existing services. The 'Create' button is highlighted with a red box. The table lists three services:

| Name | Labels | Type | Created At | Cluster IP | Internal Endpoint | External Endpoint | Actions |
|-----------------------|--|-----------|-----------------------------|--------------|--------------------------------|-------------------|--|
| kubernetes | component:apiserver provider:kubernetes | ClusterIP | Nov 1, 2021, 16:12:58 UTC+8 | 172.21.0.1 | kubernetes:443 TCP | - | Details Update View in YAML Delete |
| mongo-express-service | - | ClusterIP | Nov 1, 2021, 16:58:34 UTC+8 | 172.21.0.171 | mongo-express-service:8081 TCP | - | Details Update View in YAML Delete |
| mongodb-service | - | ClusterIP | Nov 1, 2021, 16:55:28 UTC+8 | 172.21.4.175 | mongodb-service:27017 TCP | - | Details Update View in YAML Delete |

At the bottom of the table, there is a 'Batch Delete' button and a pagination bar showing 'Total: 3 item(s), Per Page: 25 item(s)'.

Create a SLB and assign the backend server with 'mongo-express', the pod port and service port should be defined according to the YAML configuration files.

Create Service



Name:

Type:

Server Load Balancer

Public Access

Create SLB Instance

slb.s1.small [Modify](#)

! Select the instance type based on business needs. For more information about SLB billing method, see [Billing method](#). If an SLB instance is automatically created, it will be deleted when the Service is deleted.

Backend:

mongo-express

[Add Pod Label](#)

External Traffic Policy:

Local

Port Mapping:

[+](#) Add

| Name [?] | Service Port | Container Port | Protocol |
|-------------------|--------------|----------------|-----------------------------------|
| | 8081 | 8081 | TCP ⌵ ⊖ |

Annotations:

[+](#) Add

Label:

[+](#) Add

Create

Cancel

- 8) Click the 'external endpoint' shown in the SLB service, you should be redirected to the mongo-express web portal which allow you to manipulate the mongo-db.

Services

Create

Create Resources in YAML

| Search by name <input type="text"/> | | | | | | | <input type="button" value="Q"/> | <input type="button" value="Refresh"/> |
|-------------------------------------|---------|--|--|-----------------------------|-------------|---------------------------------------|--|--|
| <input type="checkbox"/> | Name | Labels | Type | Created At | Cluster IP | Internal Endpoint | External Endpoint | Actions |
| <input type="checkbox"/> | app-slb | service.beta.kubernetes.io/hash:7270ca8b94defd6248f7583240fc338769330bdf6c04d26905c837b0 | LoadBalancer Monitoring information | Nov 1, 2021, 17:08:32 UTC+8 | 172.21.8.53 | app-slb:8081 TCP app-slb:30923 TCP | [REDACTED] | Details Update View in YAML Delete |



Mongo Express Database ▾

Databases

Database Name

[+ Create Database](#)

| | | |
|----------------------|--------|---------------------|
| View | admin | Del |
| View | config | Del |
| View | local | Del |

Server Status

Turn on admin in config.js to view server stats!

I.Clean-ups

- 1) Remove SLB we created for mongo-express external access via Alibaba Cloud Console

| <input type="checkbox"/> | Name | Labels | Type | Created At | Cluster IP | Internal Endpoint | External Endpoint | Actions |
|--------------------------|---------|--|--|--------------------------------|-------------|---------------------------------------|-------------------|--|
| <input type="checkbox"/> | app-slb | service.beta.kubernetes.io/hash:7270ca8b94defd6248f7583240fc338769330bdf6c04d26905c837b0 | LoadBalancer Monitoring information | Nov 1, 2021, 17:08:32 UTC+8 | 172.21.8.53 | app-slb:8081 TCP app-slb:30923 TCP | | Details Update View in YAML Delete |

- 2) Go back to your YAML files directory via terminal and execute following commands one by one

'kubectl delete -f 4_mongo-express-internal.yaml'

'kubectl delete -f 3_mongo-configmap.yaml'

'kubectl delete -f 2_mongo.yaml'

'kubectl delete -f 1_secret.yaml'

- 3) Go back to your terraform configuration files directory via terminal and destroy the cloud resources you provisioned for this demo with command

'terraform destroy' and confirm it with 'yes'

```
Enter a value: yes
alicloud_cs_serverless_kubernetes.serverless: Destroying... [id=cb40214127ff04b63b91033250274bd8e]
alicloud_cs_serverless_kubernetes.serverless: Still destroying... [id=cb40214127ff04b63b91033250274bd8e, 10s elapsed]
alicloud_cs_serverless_kubernetes.serverless: Still destroying... [id=cb40214127ff04b63b91033250274bd8e, 20s elapsed]
alicloud_cs_serverless_kubernetes.serverless: Still destroying... [id=cb40214127ff04b63b91033250274bd8e, 30s elapsed]
alicloud_cs_serverless_kubernetes.serverless: Still destroying... [id=cb40214127ff04b63b91033250274bd8e, 40s elapsed]
alicloud_cs_serverless_kubernetes.serverless: Still destroying... [id=cb40214127ff04b63b91033250274bd8e, 50s elapsed]
alicloud_cs_serverless_kubernetes.serverless: Still destroying... [id=cb40214127ff04b63b91033250274bd8e, 1m0s elapsed]
alicloud_cs_serverless_kubernetes.serverless: Still destroying... [id=cb40214127ff04b63b91033250274bd8e, 1m10s elapsed]
alicloud_cs_serverless_kubernetes.serverless: Still destroying... [id=cb40214127ff04b63b91033250274bd8e, 1m20s elapsed]
alicloud_cs_serverless_kubernetes.serverless: Still destroying... [id=cb40214127ff04b63b91033250274bd8e, 1m30s elapsed]
alicloud_cs_serverless_kubernetes.serverless: Still destroying... [id=cb40214127ff04b63b91033250274bd8e, 1m40s elapsed]
alicloud_cs_serverless_kubernetes.serverless: Still destroying... [id=cb40214127ff04b63b91033250274bd8e, 1m50s elapsed]
alicloud_cs_serverless_kubernetes.serverless: Still destroying... [id=cb40214127ff04b63b91033250274bd8e, 2m0s elapsed]
alicloud_cs_serverless_kubernetes.serverless: Still destroying... [id=cb40214127ff04b63b91033250274bd8e, 2m10s elapsed]
alicloud_cs_serverless_kubernetes.serverless: Still destroying... [id=cb40214127ff04b63b91033250274bd8e, 2m20s elapsed]
alicloud_cs_serverless_kubernetes.serverless: Still destroying... [id=cb40214127ff04b63b91033250274bd8e, 2m30s elapsed]
alicloud_cs_serverless_kubernetes.serverless: Still destroying... [id=cb40214127ff04b63b91033250274bd8e, 2m40s elapsed]
alicloud_cs_serverless_kubernetes.serverless: Still destroying... [id=cb40214127ff04b63b91033250274bd8e, 2m50s elapsed]
alicloud_cs_serverless_kubernetes.serverless: Destruction complete after 2m56s
alicloud_vswitch.test_vswitch: Destroying... [id=vsw-j6cnazzdoyz9a43cz5ljh]
alicloud_vswitch.test_vswitch_bak: Destroying... [id=vsw-j6culscso9tksvkq7urit]
alicloud_vswitch.test_vswitch: Destruction complete after 6s
alicloud_vswitch.test_vswitch_bak: Destruction complete after 10s
alicloud_vpc.test_vpc: Destroying... [id=vpc-j6cfvx370a0hnf085ngbi]
alicloud_vpc.test_vpc: Destruction complete after 6s

Destroy complete! Resources: 4 destroyed.
```

-End Of Document-