Devops assignment

Solution approach

Considerations

- `kops`
    - kops state has been stored in S3 (`s3://ounass-swapnil-khedekar-com-kops-state`)
    - Kops are used to install kubernetes on AWS on EC2
    - It has been deployed in `ap-south-1` region
    - It is Highly available cluster spread across 3 AZz
    - It has multiple ig, mixed instance group and lifecycle (spot and ondemand).
    - It has cluster autoscaler for all instance-groups
    - It has lifecycle for all instances-groups(except spot ig)
    - For app ondeamnd and spot instances combined
    - For jobs workload only spot instances has been created
- `Ansible`
    - used ansible to manage configuration like nginx, s3-lifecycle-policies etc
    - Bucket creation is done through ansible along with glacier policies
    - Once the kops cluster has been configured then kubernetes platform charts like nginx-ingress controller, helm, metricserver, HPA controller, tls certificate secret have been created via ansible

    - The application has been deployed using ansible playbook which used helm chart internally
- `AWS`
    - Using S3 for kops state
    - Using S3 for application data(csv)for store files
    - Implemented s3 glacier transition for application bucket(`s3://ounass-csv-upload-files`)
    - Using existing VPC becasue it's already configured properly and it has DNS public zone
    - Also elasticIP quota is already extended for this vpc
    - Network load balancer has been used for application routing
    - IAM user accesskeys and secret keys has been used and it is provided as environment variable to avoid misuse
    - Route53 DNS and public zone (`swapnilkhedekar.com`) has been used for public website has been configured with A record and alias to NLB
    - Website is live at: https://ounass.swapnilkhedekar.com/
    - Zerossl certificates has been issued for 90days
- `Docker`
    - We are using dockerhub to store built images
    - https://hub.docker.com/repository/docker/swapnilkhedekar/ounass/general
    - Image does not contain secrets or sensitive information
    - Images are `slim` and build for `linux/amd64`
    - It can be built for `arm64` to support graviton instances
- `Python flask with bootstrap`
    - Used Python Flask application that uses Bootstrap for styling to build modern and responsive user interface
    - Latest Python3.12 version has been used
    - Virtualenv configuration has been used to avoid conflicts
    - Python app is stateless so that it can be used with kubernetes truly microservice environment
    - it used S3 to store files and retrive existing information
- `Kubernetes`
    - Deployments,secret,configmaps,service,ingress have been used to deploy the application along with routing
    - HPA has been used to autoscaler
    - Readiness/liveness have been configured in deployment
    - nginx and application containers are running in same pod with shared volume for statis files
    - topologySpreadConstraints and affinity have been configured to balance AZ and used on-demand nodes for apps
    - request/limit has been configured to avoid bad scheduling
    - imagePullSecrets have been configured to fetch images from dockerhub

    - ingress nginx controller has been configured to use ingress and it usage NLB
    - terminationGracePeriodSeconds has been configured
    - rollingUpdate has been configured for rolling update

Directory structure

- tree structure

```
#ansible home directory
ansible
├── 00-s3-bucket-playbook.yml
├── 01-pre-app-playbook.yml
├── 02-app-playbook.yml
├── inventory.ini
└── templates
    ├── nginx.conf.j2
    └── s3-lifecycle-policy-glacier.json.j2

#helm home directory
chart
├── Chart.yaml
├── index.yaml
├── templates
│   ├── Deployment.yaml
│   ├── Service.yaml
│   ├── _helpers.tpl
│   ├── configmap.yaml
│   ├── docker-registry-secret.yaml
│   ├── hpa.yaml
│   ├── nginx-ingress.yaml
│   ├── ounass-swapnilkhedekar-com-tls.yaml
│   └── secret.yaml
└── values.yaml


kops
└── kops-cluster-with-ha-mixedig-ondemand-spot-autoscaler.yaml

ounass_swapnilkhedekar_com-ssl-cert
├── ca_bundle.crt
├── certificate.crt
├── complete_certificate.crt
└── private.key


#python app home directory
python-application
├── Dockerfile
├── app.py
├── requirements.txt
├── static
```

```
│   └── images
│       └── burj_khalifa.jpg
└── templates
    ├── index.html
    ├── list_files.html
    └── result.html
```

file path	Description
kops	kops cluster manifest files
ansible	home directory for ansible configuration
ansible/templates	for ansible jinja templates
chart	helm chart for k8s deployment
python-application	Python flask application code
ounass_swapnilkhedekar_com-ssl-cert	SSL certificate and private keys
documentation.md	actual documentation markdown

Tools and services practical guide

Kops cluster

Install kops on aws on EC2 instances

```
brew install kops

export KOPS_STATE_STORE=s3://ounass-swapnil-khedekar-com-kops-state
export CLUSTER_NAME=ounass.swapnilkhedekar.com
export VPC_ID=vpc-067cafcfac66d9ba5
export NETWORK_CIDR=10.10.0.0/16

cd kops

kops create -f kops-cluster-with-ha-mixedig-ondemand-spot-autoscaler.yaml -v=7
kops update cluster ${CLUSTER_NAME} --yes --admin -v=7

kops get cluster --name ounass.swapnilkhedekar.com

kops update cluster --name ounass.swapnilkhedekar.com  --wait 5m
#kops delete cluster --name ounass.swapnilkhedekar.com1 --yes
```

Install minikube

```
brew install minikube
brew unlink minikube
brew link minikube
minikube start --kubernetes-version=v1.29.0
```

AWS

AWS commands

```
brew install aws-cli

[swapnil-kops]
aws_access_key_id=XXXXF3WM
aws_secret_access_key=XXXXbsEY

aws ssm get-parameters --names /aws/service/eks/optimized-ami/1.29/amazon-linux-2/recom
{
    "Parameters": [
        {
            "Name": "/aws/service/eks/optimized-ami/1.29/amazon-linux-2/recommended/ima
            "Type": "String",
            "Value": "ami-0e6e7c1b50fbabc04",
            "Version": 19,
            "LastModifiedDate": "2024-06-28T03:06:44.871000+05:30",
            "ARN": "arn:aws:ssm:ap-south-1::parameter/aws/service/eks/optimized-ami/1.2
            "DataType": "text"
        }
    ],
    "InvalidParameters": []
}

##Move files after 30 days to glacier
aws s3api put-bucket-lifecycle-configuration --bucket ounass-csv-upload-files --lifecyc
```

Docker

```
cd python-application
DOCKER_BUILDKIT=1 docker build --platform linux/amd64 -t swapnilkhedekar/ounass:1.0-lin
docker push swapnilkhedekar/ounass:1.0-linux-amd64
```

Ansible configuration

```
cd ansible

ansible-playbook -i inventory.ini 00-s3-bucket-playbook.yml -vvv
ansible-playbook -i inventory.ini 01-pre-app-playbook.yml
ansible-playbook -i inventory.ini 02-app-playbook.yml
```

Helm

```
# nginx ingress controller
helm repo add ingress-nginx https://kubernetes.github.io/ingress-nginx
helm repo update
helm upgrade -i ingress-nginx ingress-nginx/ingress-nginx \
  --namespace ingress-nginx --create-namespace \
  --set controller.service.type=LoadBalancer \
  --set controller.service.annotations."service\.beta\.kubernetes\.io/aws-load-balancer-


# install hpa/vpa and metric server
helm repo add metrics-server https://kubernetes-sigs.github.io/metrics-server/
helm repo update
helm install -n kube-system metrics-server metrics-server/metrics-server

# Tls certificate
cat complete_certificate.crt | base64 | tr -d '\n'
cat private.key | base64 | tr -d '\n'

# deploy and test app
helm template . -n ounass-app --create-namespace

helm upgrade --install ounass-app . -n ounass-app --create-namespace --debug
```