DevOps Case Study

Below are some questions we would like you to work on. We expect to receive your feedback within 5 business days. Please feel free to contact us if you found the questions are unclear.

Configuration management

Suggested environment: Ubuntu 20 LTS, ansible 2.9.16, puppet 4.5 or above.

1) Which ansible command can display all ansible_ configuration for a host.

Ans:

```
ops@ansible:~$ ansible --version
ansible [core 2.12.10]
  config file = /etc/ansible/ansible.cfg
  configured module search path = ['/home/ops/.ansible/plugins/modules', '/usr/share/ansible/plugins/modules']
  ansible python module location = /usr/lib/python3/dist-packages/ansible
  ansible collection location = /home/ops/.ansible/collections:/usr/share/ansible/collections
  executable location = /usr/bin/ansible
  python version = 3.8.10 (default, Sep 11 2024, 16:02:53) [GCC 9.4.0]
  jinja version = 2.10.1
  libyaml = True
ops@ansible:~$
```

We can also generate ansible configuration file with the below command.

$ ansible-config init --disabled > ansible.cfg path: `/etc/ansible/ansible.cfg`

2) Please configure a cron job that runs logrotate on all machines every 10 minutes between 2h - 4h.

Ans: Attached Ansible playbook

3) Please deploy ntpd package to the following 3 servers:

app-vm1.fra1.internal (192.168.0.2)
db-vm1.fra1.db (192.168.0.3)
web-vm1.fra1.web (192.168.0.4)

with custom config of /etc/ntpd.conf:

```
tinker panic 0
restrict default kod nomodify notrap nopeer noquery
restrict -6 default kod nomodify notrap nopeer noquery
restrict 127.0.0.1
restrict -6 ::1
server   192.168.0.252 minpoll 4 maxpoll 8
server   192.168.0.251 minpoll 4 maxpoll 8
server   192.168.0.0 # local clock
fudge    192.168.0.0 stratum 10
driftfile /var/lib/ntp/drift
keys     /etc/ntp/keys
```

Ans: Attached Ansible playbook role based ntp. But the IP configuration is different. It is tested in azure two node setup. server-1: Ansible server+agent, and Server-2: agent alone

We also need to deploy monitoring template onto our nagios server "monitoring.fra1.internal", each of the above machines should use the following nagios templates:

```
define host {
  host_name                machine_name
  address                  machine_ip
  check_command            check-ping
  active_checks_enabled    1
  passive_checks_enabled   1
}

define service {
  service_description      ntp_process
  host_name                machine_name
  check_command            check_ntp
  check_interval           10
}
```

Ans: Attached Ansible playbook role based nagios. This could not be tested as I am not familiar with Nagios. I have experience in Zabbix as a monitoring tool.

Docker/Kubernetes

Suggested environment: Ubuntu 20 LTS, docker 19 or above

1) Prepare a docker-compose for a nginx server.

   Requirements:

   - nginx logs need to survive between nginx container restarts
   - docker should use network bridge subnet 172.20.8.0/24

Ans:

```
services:
  nginx:
    image: nginx:latest
    container_name: nginx_server
    ports:
      - "80:80"  # Map host port 80 to container port 80
    volumes:
      - nginx_logs:/var/log/nginx  # Volume for persistent logs
    networks:
      nginx_bridge:
        ipv4_address: 172.20.8.10  # Specify static IP for container

volumes:
  nginx_logs:
    driver: local  # Use local volume driver to ensure persistence

networks:
  nginx_bridge:
    driver: bridge #Default network though
    ipam:
      config:
        - subnet: 172.20.8.0/24
```

2) Which Kubernetes command you will use to identify the reason for a pod restart in the project "internal" under namespace "production".

Ans:

List the pod first to find cause of restart

```
kubectl get pods -n production
```

Now check pod

```
kubectl describe pod <pod-name> -n production
```

Additionally we check pod logs

```
kubectl logs <pod-name> -n production
```

3) Consider the followings:

POD	NAME	CPU(cores)	MEMORY(bytes)
java-app-7d9d44ccbf-lmvbc	java-app	3m	951Mi
java-app-7d9d44ccbf-lmvbc	java-app-logrotate	1m	45Mi
java-app-7d9d44ccbf-lmvbc	java-app-fluentd	1m	84Mi
java-app-7d9d44ccbf-lmvbc	mongos	4m	62Mi

Application pod has the following resource quota:

- Memory request & limit: 1000 & 1500
- CPU request & limit: 1000 & 2000
- Xmx of 1000M

Java-app keep restarting at random. From Kubernetes configuration perspective, what are the possible reasons for the pod restarts?

Ans:

— Memory usage is very close to the allocated quota. Heap size is also an important tweaking parameter. Running “kubectl describe” is a useful command to check the **OOMKilled** event.
— CPU utilization is 0.3%, far below the limit, so this is unlikely to be the cause.
— Separating containers from the same POD could be an analysis option to isolate memory allocation issues.
— Xmx can be increased. It is a critical area to monitor increasing. If increasing Xmx solves the problem then raising a RED flag to the development team is crucial. They need to work on garbage collection to free up memory accordingly.
— Imgaepull Error is also notable to work on. I have seen from the following task of elasticsearch helm template deployment.

Helm

Please use the accompanied elasticsearch helm template to create a Kubernetes deployment of elasticsearch. Provide a screenshot & deployment yaml of the resultant deployment in Kubernetes.

Ans:

```
ops@kubernete:~$ kubectl get pods --namespace default
NAME                           READY   STATUS             RESTARTS         AGE
customer-abc-elasticsearch-0   0/4     ImagePullBackOff   12 (5m24s ago)   55m
customer-abc-elasticsearch-1   0/4     CrashLoopBackOff   11 (2m57s ago)   55m
customer-abc-elasticsearch-2   0/4     CrashLoopBackOff   12 (15s ago)     55m
ops@kubernete:~$ cd elasticsearch/
ops@kubernete:~/elasticsearch$ ls
Chart.yaml  deployment.yaml  envs  templates
ops@kubernete:~/elasticsearch$ vi deployment.yaml
ops@kubernete:~/elasticsearch$ kubectl get pods --namespace default
NAME                           READY   STATUS             RESTARTS         AGE
customer-abc-elasticsearch-0   0/4     ErrImagePull       13 (5m50s ago)   63m
customer-abc-elasticsearch-1   0/4     CrashLoopBackOff   12 (3m16s ago)   63m
customer-abc-elasticsearch-2   0/4     CrashLoopBackOff   13 (43s ago)     63m
ops@kubernete:~/elasticsearch$ kubectl get statefulsets --namespace default
NAME                      READY   AGE
customer-abc-elasticsearch   0/3     63m
ops@kubernete:~/elasticsearch$ kubectl get svc --namespace default
NAME                                TYPE        CLUSTER-IP      EXTERNAL-IP   PORT(S)                      AGE
customer-abc-elasticsearch          ClusterIP   10.107.33.173   <none>        9200/TCP,9300/TCP,9114/TCP   63m
customer-abc-elasticsearch-headless   ClusterIP   None            <none>        9200/TCP,9300/TCP,9114/TCP   63m
kubernetes                          ClusterIP   10.96.0.1       <none>        443/TCP                      108m
ops@kubernete:~/elasticsearch$
```

These are also investigated. The errors found have been got rid of mostly but one imagepullerror. I will continue to check these. Primarily fluentd, elasticsearch-exporter, and logrotate images are found problematic.

kubectl logs customer-abc-elasticsearch-0 --namespace default
kubectl logs customer-abc-elasticsearch-1 --namespace default
kubectl logs customer-abc-elasticsearch-2 --namespace default

kubectl describe pod customer-abc-elasticsearch-0 --namespace default
kubectl describe pod customer-abc-elasticsearch-1 --namespace default
kubectl describe pod customer-abc-elasticsearch-2 --namespace default

kubectl get events --namespace default

Deployment.yml

```
---
# Source: elasticsearch/templates/secret.yaml
apiVersion: v1
kind: Secret
metadata:
  name: es-secret
  namespace: default
type: Opaque
stringData:
  esURI: http://customer-abc-headless:9200
---
# Source: elasticsearch/templates/configmap.yaml
apiVersion: v1
kind: ConfigMap
metadata:
  name: elasticsearch-fluentd-config
  namespace: default
data:
  fluentd.conf: |
    <source>
      @type tail
      path /usr/share/elasticsearch/logs/*.log
      pos_file /tmp/log.pos
      read_from_head true
      <parse>
        @type regexp
        expression /^(?<msg>.*)$/
        #13 Nov 2020 10:53:45.069,
        #time_format %d %m %Y %H:%M:%S.mili
      </parse>
      tag graylog2
    </source>

    <filter graylog2.**>
      @type record_transformer
      <record>
        facility "elasticsearch"
        component "elasticsearch"
```

Metrics

1) Explain how Prometheus work.
2) How do you create custom Prometheus alerts and alerting rules for Kubernetes monitoring? Provide an example alert rule and its configuration.
3) What is the Prometheus query you can use in Granfana to properly show usage trend of an application metric that is a counter?

Ans: I possess five years experience on zabbix only. No other monitoring tools I have worked with. Have a positive mindset to learn new tools.

Databases

Suggested environment: Cassandra 4.0 or above, mongo 4.4.0 or above

1) ### Cassandra

   Query to db cluster returns different result each time. Users reported query result has data records that they deleted days ago.

   Explain what the likely reason for the behavior and how to avoid it.

   Ans: I do not have experience on Cassandra at all. But I am willing to learn. However I Know elasticsearch and MongoDB to a very good extent.

2) ### Mongo

   We have mongodb replicaset_1 with the following db and collections.

```
# mongo
MongoDB shell version v3.6.18
connecting to: mongodb://127.0.0.1:27017/?gssapiServiceName=mongodb
Implicit session: session { "id" : UUID("c9f6a47b-3155-4855-992d-65ed218f7bf5") }
MongoDB server version: 3.6.18
sServer has startup warnings:
mongos> show dbs
admin                0.000GB
config               0.010GB
sanfrancisco         65.825GB
test                 0.000GB

mongos> use sanfrancisco
switched to db sanfrancisco

mongos> show collections
company_name
street_name
product_type
market_segment
```

A sample record from company_name:

```
{
      "_id" : 5,
      "market_segment_id" : 1,
      "legal_name" : "ABC Bakery",
      "friendly_name" : {
            "default" : "ABC Bakery"
      }
}
```

Performance is bad as the hardware of replicaset_1 is not capable to handle the database sanfrancisco. We added a new replicaset_2.

Please provide all steps required to shard the collection sanfrancisco.company_name based on _id.

Ans:

Prerequisite: A MongoDB cluster needs to be up and running with multiple replicaset including sharding configuration. In this case replicaset_1 and replicaset_2.

Sharding the sanfrancisco.company_name collection based on the _id field across multiple replica sets (replicaset_1 and replicaset_2) is a good approach.

Configure mongos Router

Ensure the mongos instance is connected to the config servers and is aware of the replica sets.

mongos --configdb configReplSet/hostname:port --port 27017

Add Replica Sets to the Sharded Cluster

sh.addShard("replicaset_1/replicaset_1_host:port")

sh.addShard("replicaset_2/replicaset_2_host:port")

Enable Sharding on the Database

sh.enableSharding("sanfrancisco")

Shard the Collection

Shard the company_name collection using the _id field as the shard key

sh.shardCollection("sanfrancisco.company_name", { "_id": 1 })

Verify Shard Distribution

sanfransisco.company_name.getShardDistribution()