UNIVERSIDAD DE SAN CARLOS DE GUATEMALA

FACULTAD DE INGENIERÍA

ESCUELA DE CIENCIAS Y SISTEMAS

LAB. SOFTWARE AVANZADO

ING. EVEREST DARWIN MEDINILLA RODRIGUEZ

AUX. DIEGO MOLINA

DOCUMENTACIÓN PRÁCTICA 4

201830313 - DENILSON FLORENTÍN DE LEÓN AGUILAR

201931581 - JONATHAN MARCOS VALIENTE GONZÁLEZ

Índice

Documentación de Implementación del Clúster Kubernetes On-Premises con Alta Disponibilidad 3

1. Instalación y Configuración de Kubernetes Utilizando Kubespray 3

1.1. Creación de Infraestructura Utilizando Terraform: 3

1.2. Configuración del Nodo de Control con Ansible: 3

1.3. Ejecución de Terraform y Preparación de la Infraestructura: 3

2. Despliegue de un Clúster de Alta Disponibilidad con Nodos Maestro y Trabajadores 4

2.1. Preparación del Inventario de Kubespray: 4

2.2. Configuración de Opciones de Kubespray: 4

2.3. Ejecución de Kubespray: 4

3. Integración de la Pila ELK para Monitorización y Logging Distribuido 4

3.1. Configuración de la Pila ELK: 5

3.2. Integración con Kubernetes: 5

4. Gestión y Mantenimiento Continuo del Clúster Kubernetes y la Solución de Monitorización 5

4.1. Actualizaciones y Parches: 5

4.2. Escalado y Balanceo de Carga: 5

4.3. Resolución de Problemas: 6

5. Recomendaciones y Mejores Prácticas para la Configuración Óptima y la Seguridad del Clúster 6

5.1. Seguridad de Red: 6

5.2. Gestión de Identidades y Accesos: 6

5.3. Supervisión y Auditoría: 6

Guía paso a paso para desplegar un clúster de alta disponibilidad con nodos maestro y trabajadores. 7

Documentación sobre la integración de la pila ELK para monitorización y logging distribuido. 9

elasticsearch.yaml 9

kibana.yaml 10

logstash.yaml 11

Instrucciones para la gestión y mantenimiento continuo del clúster Kubernetes y la solución de monitorización. 12

Recomendaciones y Mejores Prácticas para la Configuración Óptima y la Seguridad del Clúster 13

Resultados 14

Documentación de Implementación del Clúster Kubernetes On-Premises con Alta Disponibilidad

1. Instalación y Configuración de Kubernetes Utilizando Kubespray

En primer lugar, se utilizó Kubespray para la instalación y configuración del clúster Kubernetes en un entorno on-premises. Este proceso se dividió en varios pasos detallados a continuación:

1.1. Creación de Infraestructura Utilizando Terraform:

Se empleó Terraform para crear la infraestructura necesaria en Google Cloud Platform (GCP). Se definieron 4 instancias de Google Compute Engine (GCE), una de las cuales se utilizó como nodo de control (controlplane) con Ansible instalado.

1.2. Configuración del Nodo de Control con Ansible:

En el nodo de control, se instaló Ansible y se realizaron las configuraciones necesarias para la ejecución de Kubespray. Se clonó el repositorio de Kubespray, se configuraron las dependencias requeridas y se definieron las claves SSH para el acceso a las demás instancias.

1.3. Ejecución de Terraform y Preparación de la Infraestructura:

Se ejecutaron los comandos de Terraform (terraform init, terraform plan, terraform apply) para crear la infraestructura en GCP. Una vez finalizada la creación, se copió la clave privada SSH al nodo de control para establecer la conexión con el resto de nodos.

2. Despliegue de un Clúster de Alta Disponibilidad con Nodos Maestro y Trabajadores

Una vez preparada la infraestructura y configurado el nodo de control, se procedió al despliegue del clúster Kubernetes con alta disponibilidad. Este proceso implicó los siguientes pasos:

2.1. Preparación del Inventario de Kubespray:

Se copió el directorio de inventario de Kubespray y se modificó el archivo hosts.yaml para incluir las direcciones IP de los nodos maestro y trabajadores. Se asignaron nombres a los nodos y se configuraron las secciones necesarias del archivo.

2.2. Configuración de Opciones de Kubespray:

Se ajustaron las opciones de configuración de Kubespray según las necesidades del clúster. Se habilitó la instalación de Helm y se configuraron los plugins de red (en este caso, se utilizó Flannel).

2.3. Ejecución de Kubespray:

Con el inventario preparado y las opciones configuradas, se ejecutó Kubespray utilizando el archivo de inventario modificado. Se verificó la conectividad con los nodos y se inició la instalación del clúster.

3. Integración de la Pila ELK para Monitorización y Logging Distribuido

Una vez completada la instalación del clúster Kubernetes, se procedió a integrar la pila ELK (Elasticsearch, Logstash y Kibana) para la monitorización y registro de eventos distribuidos dentro del clúster. Este proceso se realizó siguiendo los siguientes pasos:

3.1. Configuración de la Pila ELK:

Se instaló y configuró la pila ELK en el clúster Kubernetes. Se realizaron las configuraciones necesarias para la recopilación, procesamiento y visualización de registros de eventos.

3.2. Integración con Kubernetes:

Se establecieron conexiones entre los componentes de la pila ELK y Kubernetes para permitir la monitorización en tiempo real y el análisis de registros de eventos del clúster.

4. Gestión y Mantenimiento Continuo del Clúster Kubernetes y la Solución de Monitorización

Para garantizar el correcto funcionamiento del clúster Kubernetes y la solución de monitorización a lo largo del tiempo, se proporcionaron instrucciones detalladas sobre la gestión y mantenimiento continuo. Esto incluyó:

4.1. Actualizaciones y Parches:

Se describieron los procedimientos para aplicar actualizaciones y parches de seguridad en el clúster y la pila ELK.

4.2. Escalado y Balanceo de Carga:

Se detallaron las técnicas para escalar y equilibrar la carga de trabajo en el clúster Kubernetes según las demandas del entorno.

4.3. Resolución de Problemas:

Se proporcionaron pautas para identificar y resolver problemas comunes que puedan surgir en la operación diaria del clúster y la solución de monitorización.

5. Recomendaciones y Mejores Prácticas para la Configuración Óptima y la Seguridad del Clúster

Se ofrecieron recomendaciones y mejores prácticas para optimizar la configuración y mejorar la seguridad del clúster Kubernetes y la pila ELK. Esto incluyó:

5.1. Seguridad de Red:

Se sugirieron medidas para proteger la comunicación entre los nodos del clúster y la pila ELK, así como para restringir el acceso no autorizado.

5.2. Gestión de Identidades y Accesos:

Se propusieron estrategias para administrar identidades y accesos en el clúster, como la implementación de autenticación y autorización basadas en roles.

5.3. Supervisión y Auditoría:

Se recomendó la implementación de herramientas de supervisión y auditoría para monitorear la actividad del clúster y detectar posibles violaciones de seguridad.

Estas acciones permitieron garantizar un entorno Kubernetes seguro, confiable y altamente disponible, con capacidades de monitorización y registro para facilitar la detección y resolución de problemas.

Guía paso a paso para desplegar un clúster de alta disponibilidad con nodos maestro y trabajadores.

1) Haber desplegado la infraestructura en google cloud en la carpeta terraform con los siguientes comandos:

```
terraform init
terraform plan -no-color
terraform apply -no-color
```

   a) Toma en cuenta que la llave ssh configurada debes crearla previamente y sin passphrase

2) Luego accede a la máquina virtual que funciona como control node de master1, worker1 y worker2.

```
ssh pharaox@<IP_externa_control_panel>
```

3) Descarga el repositorio de kubespray

   a) https://github.com/kubernetes-sigs/kubespray.git

4) Clona un ejemplo y trabaja sobre el:

```
cp -rfp inventory/sample inventory/dev
declare -a IPS=(10.250.0.3 10.250.0.2 10.250.0.5)
CONFIG_FILE=inventory/dev/hosts.yaml python3
contrib/inventory_builder/inventory.py ${IPS[@]}
```

5) Configurar el archivo inventory/dev/hosts.yaml

```
all:
 hosts:
   master1:
     ansible_host: 10.250.0.3
     ip: 10.250.0.3
     access_ip: 10.250.0.3
   worker1:
     ansible_host: 10.250.0.2
     ip: 10.250.0.2
     access_ip: 10.250.0.2
```

```
    worker2:
      ansible_host: 10.250.0.5
      ip: 10.250.0.5
      access_ip: 10.250.0.5
  children:
    kube_control_plane:
      hosts:
        master1:

    kube_node:
      hosts:
        worker1:
        worker2:

    etcd:
      hosts:
        master1:

    k8s_cluster:
      children:
        kube_control_plane:
        kube_node:
    calico_rr:
      hosts: {}
```

Nota: las ips las configuras de acuerdo a como las genero terraform.

6) Habilitar help y usar calico

    a) helm_enabled: true. En k8s_cluster/addons.yml

    b) calico es la red predeterminada.

7) Ejecutar el ansible para configurar los workers

```
ansible-playbook -i inventory/dev/hosts.yaml --become --become-user=root
cluster.yml --key-file "~/abbabe"
```

8) En caso de error ejecutar estos comandos

```
ansible-playbook -i inventory/dev/hosts.yaml --become
--user=pharaox --become-user=root reset.yml -e
ansible_python_interpreter=/usr/bin/python3 --key-file
"~/google_compute_engine"
```

```
ansible-playbook -i inventory/mycluster/hosts.yaml --become
--user=pharaox --become-user=root cluster.yml -e
ansible_python_interpreter=/usr/bin/python3 --key-file
"~/google_compute_engine"
```

9) Luego se accede al nodo master 1 y se ejecuta los comandos para ver el kluster de kubespray desplegado:

```
mkdir -p $HOME/.kube
sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
sudo chown $(id -u):$(id -g) $HOME/.kube/config
```

10) Verificación del Estado del Clúster:

    a) Una vez finalizada la instalación, verifica el estado del clúster Kubernetes utilizando comandos de `kubectl`.

Documentación sobre la integración de la pila ELK para monitorización y logging distribuido.

Se procede a configurar los servicios de ELK (ElasticSearch, Kibana y Logstash). Se copian estos archivos y se ejecutan con el comando:

a) kubectl apply -f <manifesto.yaml>

elasticsearch.yaml

```
apiVersion: apps/v1
kind: StatefulSet
metadata:
 name: elasticsearch
spec:
 replicas: 3
 serviceName: elasticsearch
 selector:
   matchLabels:
     app: elasticsearch
 template:
   metadata:
     labels:
       app: elasticsearch
   spec:
     containers:
     - name: elasticsearch
       image: docker.elastic.co/elasticsearch/elasticsearch:7.15.0
       ports:
       - containerPort: 9200
         name: rest
         protocol: TCP
       - containerPort: 9300
         name: inter-node
         protocol: TCP
       volumeMounts:
       - name: elasticsearch-data
         mountPath: /usr/share/elasticsearch/data
 volumeClaimTemplates:
 - metadata:
     name: elasticsearch-data
   spec:
     accessModes: [ "ReadWriteOnce" ]
     resources:
```

```
      requests:
        storage: 10Gi
```

kibana.yaml

```
apiVersion: apps/v1
kind: Deployment
metadata:
 name: kibana
spec:
 replicas: 1
 selector:
   matchLabels:
     app: kibana
 template:
   metadata:
     labels:
       app: kibana
   spec:
     containers:
     - name: kibana
       image: docker.elastic.co/kibana/kibana:7.15.0
       ports:
       - containerPort: 5601
         name: http
         protocol: TCP
```

logstash.yaml

```
apiVersion: apps/v1
kind: Deployment
metadata:
 name: logstash
spec:
 replicas: 1
 selector:
   matchLabels:
     app: logstash
 template:
   metadata:
     labels:
       app: logstash
   spec:
```

```
    containers:
    - name: logstash
      image: docker.elastic.co/logstash/logstash:7.15.0
      volumeMounts:
      - name: config-volume
        mountPath: /usr/share/logstash/config
    volumes:
    - name: config-volume
      configMap:
        name: logstash-config
```

Instrucciones para la gestión y mantenimiento continuo del clúster Kubernetes y la solución de monitorización.

Para garantizar un funcionamiento óptimo y continuo del clúster Kubernetes y la solución de monitorización basada en ELK (Elasticsearch, Logstash, Kibana), se deben seguir las siguientes pautas de gestión y mantenimiento:

1. **Actualizaciones y Parches:**
    - Realiza regularmente la aplicación de actualizaciones y parches de seguridad en el clúster Kubernetes y los componentes de la pila ELK. Esto ayudará a mantener el entorno protegido contra posibles vulnerabilidades y fallos de seguridad.
2. **Escalado y Balanceo de Carga:**
    - Monitoriza el rendimiento del clúster Kubernetes y ajusta la capacidad según sea necesario para satisfacer las demandas de la carga de trabajo. Utiliza las capacidades de escalado automático y balanceo de carga para optimizar el uso de recursos y garantizar la disponibilidad de las aplicaciones.
3. **Resolución de Problemas:**
    - Establece procedimientos y herramientas para identificar, diagnosticar y resolver problemas que puedan surgir en el clúster Kubernetes y la solución de monitorización. Mantén registros de eventos y métricas para facilitar la detección y resolución de problemas de manera eficiente.
4. **Copia de Seguridad y Restauración:**
    - Implementa estrategias de copia de seguridad y restauración para proteger los datos y la configuración del clúster Kubernetes y la pila ELK. Realiza copias de seguridad periódicas y prueba regularmente la capacidad de restauración para garantizar la integridad y disponibilidad de los datos.

Recomendaciones y Mejores Prácticas para la Configuración Óptima y la Seguridad del Clúster

Para optimizar la configuración y mejorar la seguridad del clúster Kubernetes y la pila ELK, se recomienda seguir las siguientes mejores prácticas:

1. **Seguridad de Red:**
    - Implementa medidas de seguridad de red, como el uso de firewalls y grupos de seguridad, para proteger la comunicación entre los nodos del clúster y los componentes de la pila ELK. Limita el acceso a los servicios solo a las direcciones IP autorizadas y utilizar conexiones seguras mediante protocolos como SSL/TLS.
2. **Gestión de Identidades y Accesos:**
    - Utiliza mecanismos de autenticación y autorización basados en roles para administrar identidades y accesos en el clúster Kubernetes. Asigna permisos de manera granular según las responsabilidades y privilegios de los usuarios y aplicaciones.
3. **Supervisión y Auditoría:**
    - Implementa herramientas de supervisión y auditoría para monitorear la actividad del clúster Kubernetes y la pila ELK. Utiliza registros de eventos y métricas para realizar un seguimiento del rendimiento, detectar posibles problemas y cumplir con los requisitos de conformidad y seguridad.

Resultados

Instancias de VM

Filtro Ingresar el nombre o el valor de la propiedad

Estado	Nombre	Zona	Recomendaciones	En uso por	IP interna	IP externa	Conectar
✓	controlplane	us-east1-b			10.250.0.4 (nic0)	34.75.221.52 (nic0)	SSH
✓	master1	us-east1-b			10.250.0.2 (nic0)	34.148.193.211 (nic0)	SSH
✓	worker1	us-east1-b			10.250.0.5 (nic0)		SSH
✓	worker2	us-east1-b			10.250.0.3 (nic0)		SSH

Acciones relacionadas

```
Last login: Mon Apr 15 19:35:54 2024 from 10.250.0.4
pharaox@master1:~$ kubectl get no
NAME      STATUS   ROLES           AGE   VERSION
master1   Ready    control-plane   17h   v1.26.11
worker1   Ready    <none>          17h   v1.26.11
worker2   Ready    <none>          17h   v1.26.11
pharaox@master1:~$
```