Kchat Observability Stack

Basic Setup:

1- Create the VPC and other networking components.

2- Create an Amazon Linux AMI based EC2 instance with following specs:

Instance type

m5.4xlarge

Volume size (GiB)

/dev/xvda 50

/dev/sdb/ 500

3- Create and EIP and attach to the instance.

Keypair at:

Secret name	Description
prod/kchat-prod-observability-stack/secrets	Observability Instance Secret keypair for ssh.

Github URL for code:

https://github.com/uneycom/kchat-observability-stack.git Connect your Github account

Create .env file in the instance:

*SENTRY_SECRET_KEY=1&-5!sjh-m9+2ya2r+^my9*9dp)@algy^t&*x4jyc^b%^f%p9a
SENTRY_POSTGRES_HOST=sentry-postgres
SENTRY_POSTGRES_PORT=5432
SENTRY_DB_NAME=sentry
SENTRY_DB_USER=sentry
SENTRY_DB_PASSWORD=89PsZXyRStOT2
SENTRY_REDIS_HOST=sentry-redis
SENTRY_REDIS_PORT=6379
SENTRY_EMAIL_HOST=smtp.gmail.com
SENTRY_EMAIL_PORT=587
SENTRY_EMAIL_USER=sabit@uney.com
SENTRY_EMAIL_PASSWORD="kspl gmsv puit vkhh"
SENTRY_SERVER_EMAIL=sabit@uney.com
SENTRY_EMAIL_USE_TLS=true
SENTRY_EMAIL_SUBJECT_PREFIX="[Sentry]"
SENTRY_EVENT_RETENTION_DAYS=30

Creating the required directories:

```
sudo mkdir -p sentry-base kibana grafana-data alertmanager-data sentry-postgres prometheus-data sentry-redis elasticsearch1 elasticsearch2 elasticsearch3 logstash-pipeline esdata
```

```
# Set permissions for Sentry components
sudo chown -R 1000:1000 /mnt/ebs/sentry-base
sudo chmod -R 755 /mnt/ebs/sentry-base
# Set permissions for Kibana
sudo chown -R 1000:1000 /mnt/ebs/kibana
sudo chmod -R 755 /mnt/ebs/kibana
# Set permissions for Grafana
sudo chown -R 472:472 /mnt/ebs/grafana-data
sudo chmod -R 755 /mnt/ebs/grafana-data
# Set permissions for Alertmanager
sudo chown -R 65534:65534 /mnt/ebs/alertmanager-data
sudo chmod -R 755 /mnt/ebs/alertmanager-data
# Set permissions for Postgres
sudo chown -R 999:999 /mnt/ebs/sentry-postgres
sudo chmod -R 700 /mnt/ebs/sentry-postgres
# Set permissions for Prometheus
sudo chown -R 65534:65534 /mnt/ebs/prometheus-data
sudo chmod -R 755 /mnt/ebs/prometheus-data
# Set permissions for Redis
sudo chown -R 999:999 /mnt/ebs/sentry-redis
sudo chmod -R 755 /mnt/ebs/sentry-redis
# Set permissions for Elasticsearch
sudo chown -R 1000:1000 /mnt/ebs/elasticsearch1
sudo chmod -R 755 /mnt/ebs/elasticsearch1
sudo chown -R 1000:1000 /mnt/ebs/elasticsearch2
sudo chmod -R 755 /mnt/ebs/elasticsearch2
sudo chown -R 1000:1000 /mnt/ebs/elasticsearch3
sudo chmod -R 755 /mnt/ebs/elasticsearch3
# Set permissions for Logstash
sudo chown -R 1000:1000 /mnt/ebs/logstash-pipeline
sudo chmod -R 755 /mnt/ebs/logstash-pipeline
docker-compose -f observability-stack/observability-stack.yaml up -d
Investigate Index State
Check the state of the indices to see if they need to be closed or deleted:
curl -X GET "localhost:9200/_cat/indices?v"
If an index is corrupted or not needed, you may consider deleting it:
curl -X DELETE "localhost:9200/index-name"
```

Running the observability stack:

```
sudo yum update -y
sudo yum install -y zip unzip
sudo yum install -y nano
sudo yum install -y curl
sudo yum install -y wget
sudo yum install -y git
sudo su
mkfs -t ext4 /dev/{external-drive-letter}
lsblk
mkdir /mnt/ebs
mount /dev/{external-drive-letter} /mnt/ebs
echo '/dev/{external-drive-letter} /mnt/ebs ext4 defaults,nofail 0 2' | sudo tee -a /etc/fstab
exit
sudo yum install -y docker
sudo service docker start
```

```
sudo groupadd docker
sudo usermod -aG docker $USER
sudo chown root:docker /var/run/docker.sock
sudo chown -R root:docker /var/run/docker
newgrp docker
sudo systemctl enable docker.service
sudo systemctl enable containerd.service
sudo systemctl enable docker.service
sudo systemctl enable containerd.service
sudo systemctl status containerd.service
sudo curl -L https://github.com/docker/compose/releases/latest/download/docker-compose-$(uname  -s)-$(uname -m) -o /usr/local/bin/docker-compose
sudo chmod +x /usr/local/bin/docker-compose
docker-compose version
sudo yum install -y python3-pip
sudo pip3 install awscli
echo "Installation completed successfully."
sudo nano /mnt/ebs/.htpasswd
values from AWS Secrets Manager prod/kchat-prod-observability-stack/htpasswd-secrets
cat /mnt/ebs/.htpasswd/
git clone https://github.com/uneycom/kchat-observability-stack.git
cd observability-stack
docker-compose -f observability-stack.yaml up -d
docker ps
docker-compose -f observability-stack.yaml down
sudo docker system prune -a
curl -XGET "http://localhost:9200/_cluster/health?pretty"
curl -XGET "http://localhost:9200/_cluster/health?pretty"clear
```

Create the stack first with volumes and then copy all the data from volumes mount path to the external ebs volume directory to have observability stack configure with the external ebs volumes for enhanced persistence.