Wisilica Infra DB Cluster Automation

Prerequisites:

- Install ansible latest version(Tested on version 2.5.1).
- Should have to add ssh keys in each server.
- SSH user should have sudo privileges to all database servers.
- Required 3 or 3x3 Ubuntu 18.04 for Timescaledb, Mysql Galera, Redis DB clusters (Required separate three servers if need to implement each database clusters separately)
- Each database node should have the inter node connection to the following ports,
    - For Postgresql TimescaleDB cluster.
        - Tcp:2379 for etcd
        - Tcp 8008 for patroni
        - Tcp 5432 for PostgreSQL
    - For MysQL cluster
        - Tcp 3306 for MySQL
    - For Redis Sentinel cluster
        - 6379 for Redis
        - 26379 for Sentinel

Steps to Automate the Cluster Setup:

Extract the zip file to the ansible server where the SSH access is available to the database cluster nodes.

unzip ansible_db_clusters.zip

Change directory to db_clusters

cd db_clusters

Update Inventory file:

Open the inventory file and update the host IP address details,

For three database (TimescaleDB, MySQL and Redis)cluster configuration required a minimum nine instances if needed to create the each database cluster in a separate server or it required a minimum three instances.

List out the 3x3 set of servers IP addresses for each database cluster for running the database clusters seperately for each database cluster.

1. Three servers for configuring the PostgreSQL + TimescaleDB + Patroni + etcd cluster required a minimum of three servers.
2. Three servers for configuring the MySQL galera multi master cluster.
3. Three servers for configuring Redis Sentinel Cluster.

Or three servers for all three database clusters.

In this below example I am using 3 server for configuring all three database clusters,

IP address List

172.31.34.236
172.31.34.214
172.31.41.1

Open the inventory file and update the server IP address details.

vim inventory

```
==================================

################################
#       PostgreSQL  Cluster       #
################################

#ETCD Cluster
[etcd_cluster]  # recommendation: 3 or 5-7 nodes
172.31.34.236
172.31.34.214
172.31.41.1
# PostgreSQL nodes
[master]
172.31.34.236 hostname=db-node01

[replica]
172.31.34.214 hostname=db-node02
```

```
172.31.41.1 hostname=db-node03

[postgres_cluster:children]
master
replica
#################################

################################
#       Mysql Cluster          #
################################

[mysql_cluster]
172.31.34.236 hostname=db-node01
172.31.34.214 hostname=db-node02
172.31.41.1 hostname=db-node03
################################

################################
#       Redis Cluser           #
################################
[redis_master]
172.31.34.236 hostname=db-node01

[redis_slave]
172.31.34.214 hostname=db-node02
172.31.41.1 hostname=db-node03

[redis_cluster:children]
redis_master
redis_slave
################################

# Connection settings
[all:vars]
ansible_connection='ssh'
ansible_ssh_port='22'
ansible_user='ubuntu'

=====================
```

Note: make sure to update the ansible ssh user, port in the inventory connection settings. Also update the hostname required for each database cluster.

Update variables:

Update the Common variables in the vars/main.yml file.

vim vars/main.yml

Update the database clusters(install_postgresql_cluster, install_mysql_cluster, install_redis_cluster) as true for required database clusters.

For example to install PostgreSQL(timescaledb), MySQL Galera, Redis cluster add the following line in the common variable files,

install_postgresql_cluster: true
install_mysql_cluster: true
install_redis_cluster: true

If only required one database cluster PostgreSQL cluster then select only install_postgresql_cluster as “true”

install_postgresql_cluster: true

Update the PostgreSQL cluster name, super user, replicator user and its passwords in the postgresql configuration section in the main.yml file,

#Postgresql Cluster variables
patroni_cluster_name: "postgres-cluster" # specify the cluster name
patroni_superuser_username: "postgres"
patroni_superuser_password: "postgres-pass" # please change password
patroni_replication_username: "replicator"
patroni_replication_password: "replicatorpass" # please change password

Update the MySQL Galera cluster name and password in the MySQL field,

#MySQL Galera CLuster Variables
mysql_cluster_name: mysql_cluster_name
mysql_root_password: "your_mysql_root_password"

Update the Redis password if the Redis cluster needs to set a password.

#Redis Cluster variables
redis_password: test123 # please change password

Change the value of is_redis_password to "false" when redis cluster start without password
is_redis_password: "true"

If password is not required, make redis_password field as empty.

redis_password: ‘’
is_redis_password to "false"

Update specific Database variables:

TimescaleDB Cluster:

For updating the postgresql timecaledb cluster update the variables in the vars/postgresql.yml file.

vim vars/postgresql.yml

==========
In the postgresql.yml file we can update the postgresql, patroni and etcd cluster parameters,

patroni version latest or specific version

patroni_install_version: "latest"

Update to 'true' for enable synchronous database replication, 'false' for disable synchronous database replication

synchronous_mode: 'false'

version for deploy etcd cluster

etcd_ver: "v3.3.19"

Update etcd data directory

etcd_data_dir: "/var/lib/etcd"

Update the postgresql port

postgresql_port: "5432"

Update the postgresql parameters as shown below,

```
postgresql_parameters:
  - { option: "max_connections",                  value: "100"   }
  - { option: "superuser_reserved_connections",   value: "5"     }
  - { option: "max_locks_per_transaction",        value: "64"    }
  - { option: "max_prepared_transactions",        value: "0"     }
  - { option: "huge_pages",                 value: "try"    }
  - { option: "log_filename",               value: "'postgresql-%a.log'"        }
  - { option: "log_directory",              value: "{{ postgresql_log_dir }}" }
  - { option: "shared_preload_libraries",   value: "'timescaledb'"       }
```

MySQL Galera Cluster:

For updating the MySQL Galera cluster update the variables in the vars/mysql.yml file.

vim vars/mysql.yml

==========

*mysql_cluster_name: **mysql_cluster_name***
swap_file: /swapfile
swap_space: 512

Redis Sentinel Cluster:

For updating the Redis sentinel cluster configuration update the parameters in vars/mysql.yml file

vim vars/redis.yml

==========

Update the redis version

redis_version: 4.0.9

Update the redis source code url,
redis_download_url: http://download.redis.io/releases/redis-4.0.9.tar.gz

Update the redis master name
redis_master_name: redis-primary

Update the redis config path
redis_config_file: /etc/redis/redis.conf

Update the redis log path
redis_log_file: /var/log/redis/redis.log

Update the redis data directory
redis_data_dir: /var/lib/redis

Update the redis binary command line path
redis_executable: /usr/local/bin/redis-server

Update the redis redis configuration file parameters,

```
redis_cluster_instances:
  - port: 6379
        protected_mode: 'no'
        maxmemory: 5000mb
        timeout: 86400
        loglevel: notice
        persistence_save: ""
        repl_diskless_sync: 'no'
        repl_timeout: 60
        repl_backlog_size: 1mb
        repl_backlog_ttl: 3600
        maxmemory_policy: volatile-lru
        maxmemory_samples: 27
        cluster_enabled: 'no'
        cluster_node_timeout: 15000
        cluster_slave_validity_factor: 10
        cluster_migration_barrier: 1
        cluster_require_full_coverage: 'no'
        client_output_buffer_limit_slave_hard: 256mb
        client_output_buffer_limit_slave_soft: 64mb
        client_output_buffer_limit_slave_time: 60
```

Run Ansible playbook:

Once updated the variables and inventory details run the ansible playbook as following,

ansible-playbook deploy_dbcluster.yml