Utkarsh Pandit

Three node MongoDB replica setup

Tools Used

- Terraform
- Ansible
- Bash script

Note

- few steps in `ScaleGrid/setup.sh` are specific to Ubuntu/Debian distro automation as they use 'apt' as their package manager. These steps include insatllation of ansible and terraform binary
- If you use any other distro, you'll have to manually install Ansible and Terraform first before you run the setup script

Content of zip

- [ssh_pub_keys/] : folder contains a ssh private and public key, to establish communication between ansible and remote machines
- [terraform-launch-instance/] : folder contains all the terraform recepies, file-teplates and config files to create and manahe ec2 instances on aws.
- [ansible-roles] : folder contains ansible roles that confires and manages the internal configuration of instances.
- [setup.sh] : file that further automates the run process of the above two tools. One click to setup everything for you

Terraform configuraion explained:

Inside terraform-launch-instance/ you will see the following files

- variables.tf: here all variables are defined, which are then used in other recepies. This includes, image ami ID, vpc_name, vpc_cidr, subnet_name, instance name,, instance size, ods disk size, etc.
    - AWS access and security keys are also defined here.
- vpc.tf: this defines and creates the virtual private network and creates subnet, internet gateway, toute table and network security groups.
- compute.tf: main file that defines and creates the ec2 instances.
- outputs.tf: This defines the parameters that needs to be printed on the terminal after the terraform run is successful.
- dev_hosts.tpl: a file template that will be used by ansble to get inforamtion of hosts it need to communicate with.
- replicahosts.tpl: a file template that will be by ansible to define hosts in remote machine

> *Default configuration of ec2 instances as defined in variables.tf: Subnet_CIDR = 192.168.10.0/24 Inbound open ports: 80, 443, 22, 27017(mongodb), 9090(prometheus), 9216(mongodb_exporter) boot_volume_size_in_gbs: 10 insatnce_type: t2.small UBUNTU ami_id: ami-055147723b7bca09a ebs_volume_size_in_gbs: 8*

To create insatnces using simply run this command:

```
cd ScaleGrid/terraform-launch-instance
terraform init
terraform apply -auto-approve
```

This command will do the following:

- Create 3 ec2 insatnces with a shared VPC.
- Attach and mount an EBS on all the instances (/datamongo)
- copies ssh public key from 'ssh_pub_keys' to remote machines
- Install few binaries in the instances using provisioner resource
- export 'hosts' file using 'dev_hosts.tpl' template to ansible-roles folder/, which will be used by ansible to communicate to machines
- export 'replicahosts' file using 'replicahosts.tpl' template to ansible-roles/ folder, which will be used to setup hosts address in all the mongodb machines

Ansible configuraion explained:

Inside ansible-roles/ folder you will see the following files:

- ansible.cfg: ansible config file that defines the default configuration of ansible to be used
- hosts: defines remote hosts address and path of private ssh key to be used to communicate with them. This file is generated automatically after terraform execution.
- mongo.yml: main file to execute or ansible play file that triggers other roles.
- mongodb/: ansible role that defines recipie to configure mongodb and replica set on remote machines
- prometheus/: ansible role that configures promethues server and mongodb_exporter.
- passwords/: contains keyfile as the shared password for authenticating other members in the mongodb replica set
- replicahosts: information in this file is used to configures /etc/hosts file in remote machines for mongodb replica. This file is generated automatically after terraform execution.

> *mongodb config file: ScaleGrid/ansible-roles/mongodb/files/mongod.conf ssl certs for mongodb: ScaleGrid/ansible-roles/mongodb/files/*.replica.com/*

To configure instances using ansible, run the following command:

```
cd ScaleGrid/ansible-roles
ansible-playbook mongo.yml -v
```

This command will do the following:

- install mongodb binaries in all the instances
- create and own several directories owned by mongodb user
- generate to and copy from keyfile from passwords/ to machines
- replace default mongod.conf with our custom mongod.conf
- copy ssl certs to the respective instances
- changes instances setting for mongodb high availabilty
- update /etc/hosts files in all the instances which has mongo replica server address.
- initiate replica set

One step to setup everything

you can just run the bash script `ScaleGrid/setup.sh` For ubuntu/debian distro, this will automatically install Ansible and terraform for you.