Elasticsearch and Kibana setup using with Terraform

Prerequisite

- AWS Console access with Required permissions.
- IAM user's Access Key ID and Secret Access Key ID with required permission
- AWS CLI V2 installed on Local System
- Terraform latest version Installed on Local system

1. Install and configure AWS CLI on local system

Download and Install AWS CLI v2 from the below link,
https://aws.amazon.com/cli/

Configure AWS cli on local system,

```
PS D:\> aws configure
AWS Access Key ID [****************LGHI]:
AWS Secret Access Key [****************9yjB]:
Default region name [us-east-1]:
Default output format [json]:
PS D:\>
```

2. Install Terraform on local system

https://developer.hashicorp.com/terraform/downloads?product_intent=terraform

3. Made the required changes on the terraform script

Create EC2 instance key pairs, security groups and elastic IPs on AWS and made the required changes on the input.tfvars file.

```
#################### EC2 Details ###################
ami_id                      = "ami-01bc990364452ab3e"
instance_type               = "t3.large"
key_name                    = "testserver"
security_group_id           = "sg-0890ef3a3242f01cd"
subnet_id                   = ["subnet-054327b091e0adc75", "subnet-
0879843e4f7d4a120"]
associate_public_ip         = "true"            # "true" or "false"
iam_role_name               = "ec2-InstanceRole" # iam role for ec2
eip_allocation_ids          = ["eipalloc-08da4be62fed552b9", "eipalloc-
0a7a673263a2f52fc"]
elastic_ip                  = ["3.220.16.77", "35.172.95.248"]
region                      = "us-east-1"
elastic_server_tag          = "Elastic-Server"
kibana_server_tag           = "Kibana-Server"
```

4. Initialize require Terraform modules

terraform init command is used to initialize a working directory containing Terraform configuration files and require modules

```
PS D:\terraform> .\terraform.exe init

Initializing the backend...
Initializing modules...
- instance in modules\ec2

Initializing provider plugins...
- Finding latest version of hashicorp/aws...
- Finding latest version of hashicorp/null...
- Installing hashicorp/aws v5.24.0...
- Installed hashicorp/aws v5.24.0 (signed by HashiCorp)
- Installing hashicorp/null v3.2.1...
- Installed hashicorp/null v3.2.1 (signed by HashiCorp)

Terraform has created a lock file .terraform.lock.hcl to record the provider
selections it made above. Include this file in your version control repository
so that Terraform can guarantee to make the same selections by default when
you run "terraform init" in the future.

Terraform has been successfully initialized!
```

```
You may now begin working with Terraform. Try running "terraform plan" to see
any changes that are required for your infrastructure. All Terraform commands
should now work.

If you ever set or change modules or backend configuration for Terraform,
rerun this command to reinitialize your working directory. If you forget, other
commands will detect it and remind you to do so if necessary.
```

5. Validate terraform script

terraform validate command is used to validate the syntax of the terraform files.

```
PS D:\terraform> .\terraform.exe validate
Success! The configuration is valid.
```

6. Create infrastructure using terraform

terraform plan command will analyze your Terraform configuration files, query the current state of your infrastructure and then output a summary of the changes to be made.

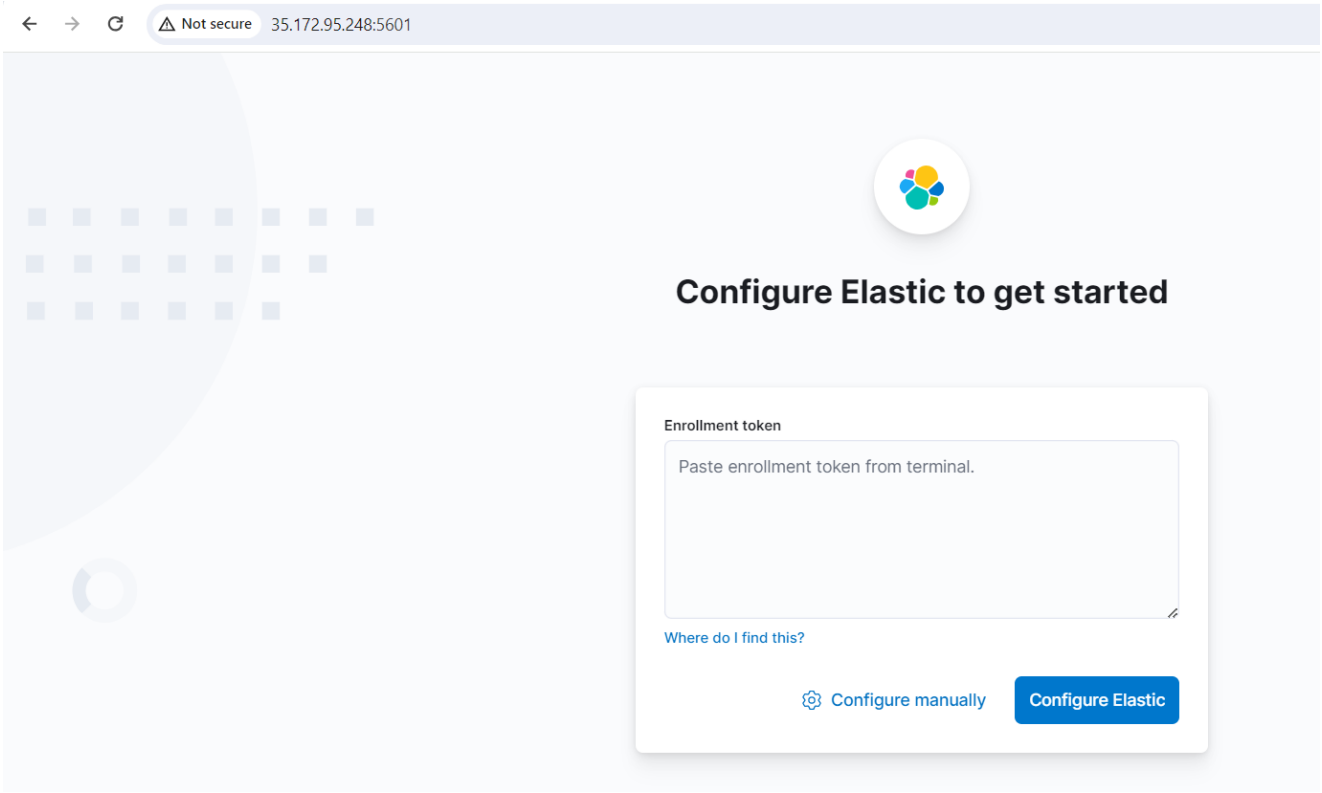
```
D:\terraform>terraform plan -var-file=input.tfvars

data.aws_iam_instance_profile.ecs_profile: Reading...
data.aws_iam_instance_profile.ecs_profile: Read complete after 1s

Terraform used the selected providers to generate the following execution plan.
Resource actions are indicated with the following symbols:
  + create

Terraform will perform the following actions:

  # module.instance["Elastic-Server"].aws_eip_association.eip_assoc will be created
  + resource "aws_eip_association" "eip_assoc" {
      + allocation_id        = "eipalloc-08da4be62fed552b9"
      + id                   = (known after apply)
      + instance_id          = (known after apply)
      + network_interface_id = (known after apply)
      + private_ip_address   = (known after apply)
      + public_ip            = (known after apply)
    }
```

terraform apply command will create the elasticsearch and kibana server

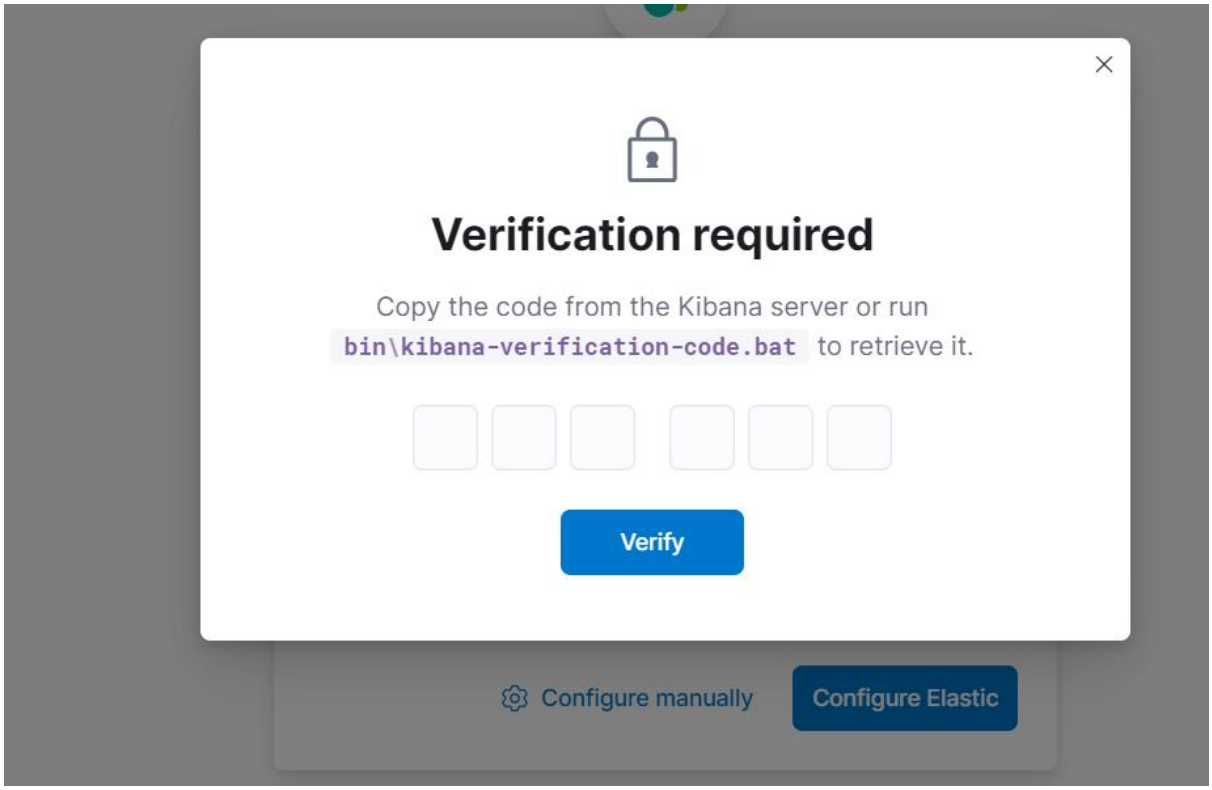
```
 D:\terraform>terraform apply -var-file=input.tfvars -auto-approve

module.instance["Elastic-Server"].null_resource.example: Still creating... [4m10s
elapsed]
module.instance["Kibana-Server"].null_resource.example: Still creating... [4m10s elapsed]
module.instance["Elastic-Server"].null_resource.example: Still creating... [4m20s
elapsed]
module.instance["Kibana-Server"].null_resource.example: Still creating... [4m20s elapsed]
module.instance["Elastic-Server"].null_resource.example: Still creating... [4m30s
elapsed]
module.instance["Kibana-Server"].null_resource.example: Still creating... [4m30s elapsed]
module.instance["Elastic-Server"].null_resource.example: Still creating... [4m40s
elapsed]
module.instance["Kibana-Server"].null_resource.example: Still creating... [4m40s elapsed]
module.instance["Elastic-Server"].null_resource.example: Still creating... [4m50s
elapsed]
module.instance["Kibana-Server"].null_resource.example: Still creating... [4m50s elapsed]
module.instance["Elastic-Server"].null_resource.example: Still creating... [5m0s elapsed]
module.instance["Kibana-Server"].null_resource.example: Still creating... [5m0s elapsed]
module.instance["Elastic-Server"].null_resource.example (remote-exec): Password for the
[elastic] user successfully reset.
module.instance["Elastic-Server"].null_resource.example (remote-exec): New value:
W3dmcREyiqelQ=g43a8h
module.instance["Elastic-Server"].null_resource.example (remote-exec):
eyJ2ZXIiOiI4LjAuMSIsImFkciI6WyIxMC4xMC4xLjExMjo5MjAwIl0sImZnciI6IjY3NTUyZGU4MzY4YzBlYzM3M
zk3NWE2OTVlYzJkNTQ1NDBjYmJlMDhjMjM5MTU1YjJlZGQyOTM1MDM3ODZiMzYiLCJrZXkiOiJfZFZSbFlzQnV1Y2
84Umc4THBPYjpNSHQ2UXA4dlNrV3NwWmxTOWJNQzVBIn0=
module.instance["Kibana-Server"].null_resource.example (remote-exec): Your verification
code is:  572 541
module.instance["Kibana-Server"].null_resource.example: Creation complete after 5m9s
[id=430851787]
module.instance["Elastic-Server"].null_resource.example: Creation complete after 5m9s
[id=656673897]

Apply complete! Resources: 6 added, 0 changed, 0 destroyed.

D:\terraform>
```

7. Configure Kibana server to connect with Elasticsearch

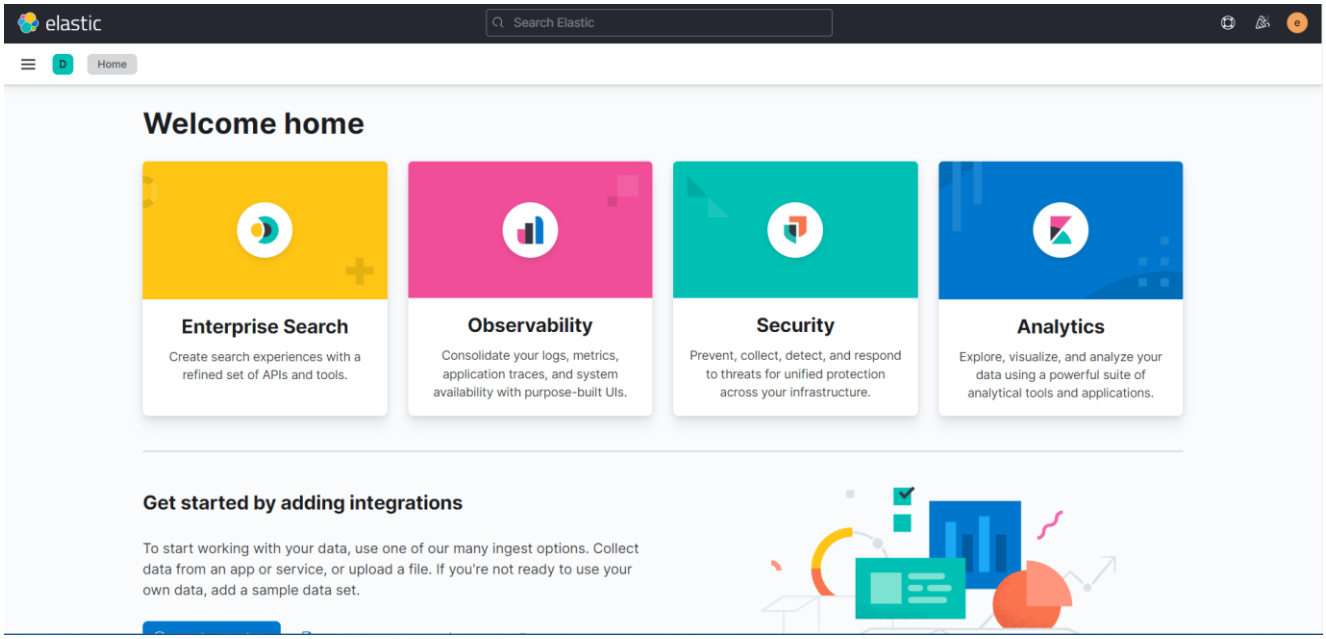
Open Kibana server URL in web browser - http://Public_IP:5601/



Provide the enrollment token and verification code for the Kibana server, which are generated in Terraform output



Log in to the Kibana URL using the username and password of the elastic user.



Destroy infrastructure using terraform

terraform destroy -var-file=input.tfvars -auto-approve

```
D:\terraform>terraform destroy -var-file=input.tfvars -auto-approve

data.aws_iam_instance_profile.ecs_profile: Reading...
data.aws_iam_instance_profile.ecs_profile: Read complete after 1s
[id=AIPA5GDEM7WAV625Y36MS]
module.instance["Kibana-Server"].aws_instance.vm: Refreshing state... [id=i-
05b89ecdbb4e1f1e4]
module.instance["Elastic-Server"].aws_instance.vm: Refreshing state... [id=i-
05ad6d68bb6eee33e]
module.instance["Elastic-Server"].aws_eip_association.eip_assoc: Refreshing state...
[id=eipassoc-081a6960f4ceab779]
module.instance["Kibana-Server"].null_resource.example: Refreshing state...
[id=430851787]
module.instance["Kibana-Server"].aws_eip_association.eip_assoc: Refreshing state...
[id=eipassoc-0f57664b93d914dae]
module.instance["Elastic-Server"].null_resource.example: Refreshing state...
[id=656673897]
module.instance["Kibana-Server"].aws_instance.vm: Still destroying... [id=i-
05b89ecdbb4e1f1e4, 30s elapsed]
module.instance["Elastic-Server"].aws_instance.vm: Still destroying... [id=i-
05ad6d68bb6eee33e, 30s elapsed]
module.instance["Kibana-Server"].aws_instance.vm: Destruction complete after 31s
module.instance["Elastic-Server"].aws_instance.vm: Still destroying... [id=i-
05ad6d68bb6eee33e, 40s elapsed]
module.instance["Elastic-Server"].aws_instance.vm: Destruction complete after 41s

Destroy complete! Resources: 6 destroyed.
```