Cloudgoat - glue_privesc

progress report

Mentor	Niko
Date	August 13, 2024 (mon)
Track	Digital Forensic
Name	Go DongHyeon

목차

1. Install scenario.. 3

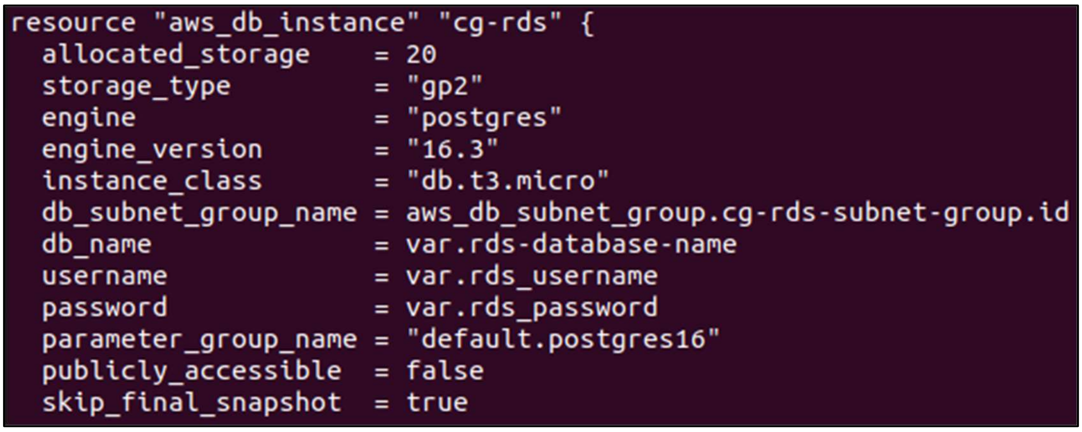
1. Install scenario

Before installation, it is necessary to modify the configuration file first.

Creating the scenario directly will result in errors because the PostgreSQL engine version is not specified.

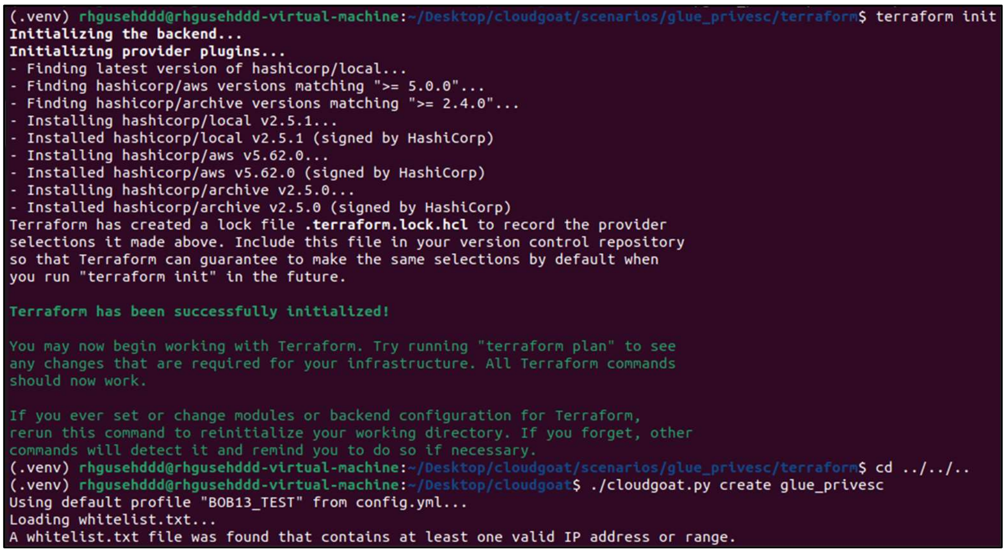
To resolve this issue, set the version to 16.3 in the rdf.tf file and save it.

After, initialize Terraform and install the scenario.



```
resource "aws_db_instance" "cg-rds" {
  allocated_storage    = 20
  storage_type         = "gp2"
  engine               = "postgres"
  engine_version       = "16.3"
  instance_class       = "db.t3.micro"
  db_subnet_group_name = aws_db_subnet_group.cg-rds-subnet-group.id
  db_name              = var.rds-database-name
  username             = var.rds_username
  password             = var.rds_password
  parameter_group_name = "default.postgres16"
  publicly_accessible  = false
  skip_final_snapshot  = true
```

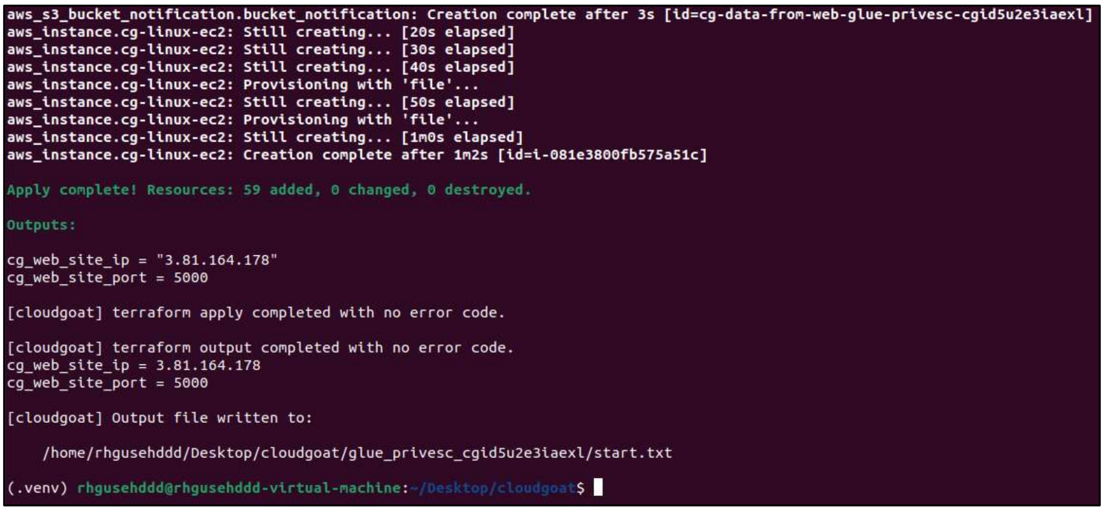
Figure 1 – rdf.tf setting



```
(.venv) rhgusehddd@rhgusehddd-virtual-machine:~/Desktop/cloudgoat/scenarios/glue_privesc/terraform$ terraform init
Initializing the backend...
Initializing provider plugins...
- Finding latest version of hashicorp/local...
- Finding hashicorp/aws versions matching ">= 5.0.0"...
- Finding hashicorp/archive versions matching ">= 2.4.0"...
- Installing hashicorp/local v2.5.1...
- Installed hashicorp/local v2.5.1 (signed by HashiCorp)
- Installing hashicorp/aws v5.62.0...
- Installed hashicorp/aws v5.62.0 (signed by HashiCorp)
- Installing hashicorp/archive v2.5.0...
- Installed hashicorp/archive v2.5.0 (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.
(.venv) rhgusehddd@rhgusehddd-virtual-machine:~/Desktop/cloudgoat/scenarios/glue_privesc/terraform$ cd ../../..
(.venv) rhgusehddd@rhgusehddd-virtual-machine:~/Desktop/cloudgoat$ ./cloudgoat.py create glue_privesc
Using default profile "BOB13_TEST" from config.yml...
Loading whitelist.txt...
A whitelist.txt file was found that contains at least one valid IP address or range.
```

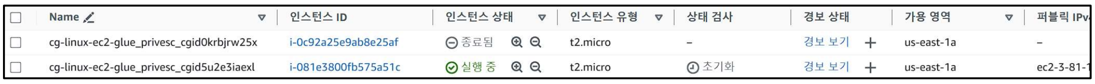
Figure 2 – terraform init & glue_privesc install

Once the installation is complete without errors, it can be verified by checking the AWS account to see that instances for the scenario have been created.



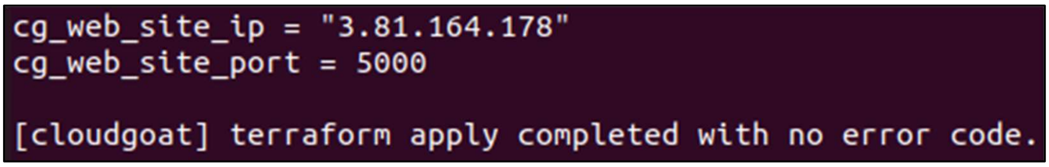
```
aws_s3_bucket_notification.bucket_notification: Creation complete after 3s [id=cg-data-from-web-glue-privesc-cgid5u2e3iaexl]
aws_instance.cg-linux-ec2: Still creating... [20s elapsed]
aws_instance.cg-linux-ec2: Still creating... [30s elapsed]
aws_instance.cg-linux-ec2: Still creating... [40s elapsed]
aws_instance.cg-linux-ec2: Provisioning with 'file'...
aws_instance.cg-linux-ec2: Still creating... [50s elapsed]
aws_instance.cg-linux-ec2: Provisioning with 'file'...
aws_instance.cg-linux-ec2: Still creating... [1m0s elapsed]
aws_instance.cg-linux-ec2: Creation complete after 1m2s [id=i-081e3800fb575a51c]

Apply complete! Resources: 59 added, 0 changed, 0 destroyed.

Outputs:

cg_web_site_ip = "3.81.164.178"
cg_web_site_port = 5000

[cloudgoat] terraform apply completed with no error code.

[cloudgoat] terraform output completed with no error code.
cg_web_site_ip = 3.81.164.178
cg_web_site_port = 5000

[cloudgoat] Output file written to:

    /home/rhgusehddd/Desktop/cloudgoat/glue_privesc_cgid5u2e3iaexl/start.txt

(.venv) rhgusehddd@rhgusehddd-virtual-machine:~/Desktop/cloudgoat$ 
```

Figure 3 – glue_privesc install



□	Name	인스턴스 ID	인스턴스 상태	인스턴스 유형	상태 검사	경보 상태	가용 영역	퍼블릭 IPv
□	cg-linux-ec2-glue_privesc_cgid0krbjrw25x	i-0c92a25e9ab8e25af	⊖ 종료됨	t2.micro	–	경보 보기 +	us-east-1a	–
□	cg-linux-ec2-glue_privesc_cgid5u2e3iaexl	i-081e3800fb575a51c	✓ 실행 중	t2.micro	초기화	경보 보기 +	us-east-1a	ec2-3-81-1

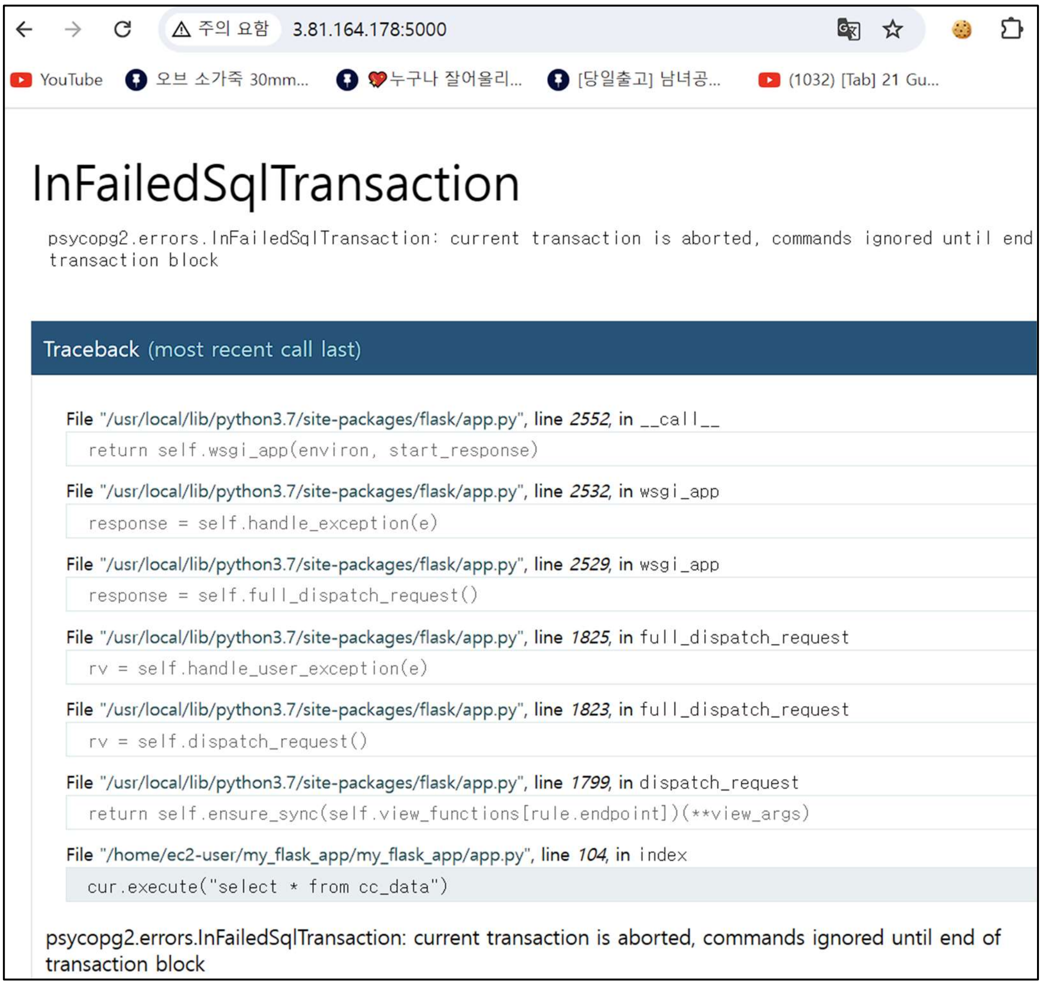
Figure 4 – AWS Instance



```
cg_web_site_ip = "3.81.164.178"
cg_web_site_port = 5000

[cloudgoat] terraform apply completed with no error code.
```

Figure 5 - site IP / port

When the scenario is installed correctly, it will provide a site IP address and port, as shown in the figure. Accessing this site should display a window similar to "Figure 6" below. If the instance is running correctly, it should be possible to access the site. Therefore, any issues related to this should be resolved first.



InFailedSqlTransaction

```
psycopg2.errors.InFailedSqlTransaction: current transaction is aborted, commands ignored until end
transaction block
```

Traceback (most recent call last)

```
File "/usr/local/lib/python3.7/site-packages/flask/app.py", line 2552, in __call__
  return self.wsgi_app(environ, start_response)
File "/usr/local/lib/python3.7/site-packages/flask/app.py", line 2532, in wsgi_app
  response = self.handle_exception(e)
File "/usr/local/lib/python3.7/site-packages/flask/app.py", line 2529, in wsgi_app
  response = self.full_dispatch_request()
File "/usr/local/lib/python3.7/site-packages/flask/app.py", line 1825, in full_dispatch_request
  rv = self.handle_user_exception(e)
File "/usr/local/lib/python3.7/site-packages/flask/app.py", line 1823, in full_dispatch_request
  rv = self.dispatch_request()
File "/usr/local/lib/python3.7/site-packages/flask/app.py", line 1799, in dispatch_request
  return self.ensure_sync(self.view_functions[rule.endpoint])(**view_args)
File "/home/ec2-user/my_flask_app/my_flask_app/app.py", line 104, in index
  cur.execute("select * from cc_data")
```

psycopg2.errors.InFailedSqlTransaction: current transaction is aborted, commands ignored until end of transaction block

Figure 6 - site access

The subsequent tasks involve ensuring that the site is running correctly and analyzing the scenario-specific attacks and logs to identify the attacks.