**Terraform Module Creation Order**

1. **Network & Security Foundation** (in root)
   * VPC
   * Subnets (public/private)
   * Route tables, IGW, NAT gateways
   * Security Groups (e.g., for RDS, Bastion, Load Balancer)

A screen shot of a computer program

AI-generated content may be incorrect.

1. **IAM Roles & Policies** (modules/iam\_roles)
   * Permissions needed by other resources ( EC2, CloudTrail)

A screen shot of a computer

AI-generated content may be incorrect.

A computer screen with white text

AI-generated content may be incorrect.

A screen shot of a computer program

AI-generated content may be incorrect.

A screen shot of a computer

AI-generated content may be incorrect.

A computer screen shot of white text

AI-generated content may be incorrect.

A computer screen shot of white text

AI-generated content may be incorrect.

A computer screen with white text

AI-generated content may be incorrect.

1. **KMS** (modules/kms)
   * For encryption keys used by Secrets Manager, RDS, S3, etc.

A black screen with white text

AI-generated content may be incorrect.

A computer code on a black background

AI-generated content may be incorrect.

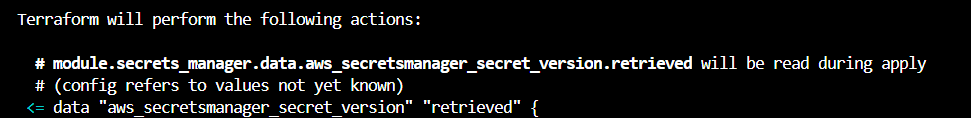
A black screen with white text

AI-generated content may be incorrect.

1. **Secrets Manager** (modules/secrets\_manager)
   * Create secrets storing DB passwords , encrypted by KMS keys

A screenshot of a computer

AI-generated content may be incorrect.



A computer code with white text

AI-generated content may be incorrect.

Run this to restore the secret name after using terraform destroy:

aws secretsmanager restore-secret --secret-id my\_secret

A computer screen with white text

AI-generated content may be incorrect.

By the time I publish this, all these keys will be deleted. It is to prove my skills to recruiters

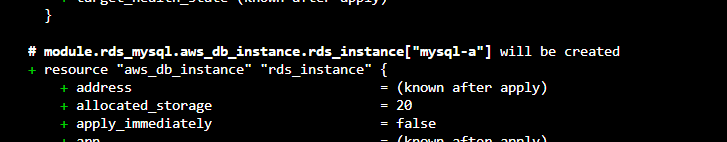
A screenshot of a computer program

AI-generated content may be incorrect.

1. **RDS MySQL** (modules/rds-mysql)

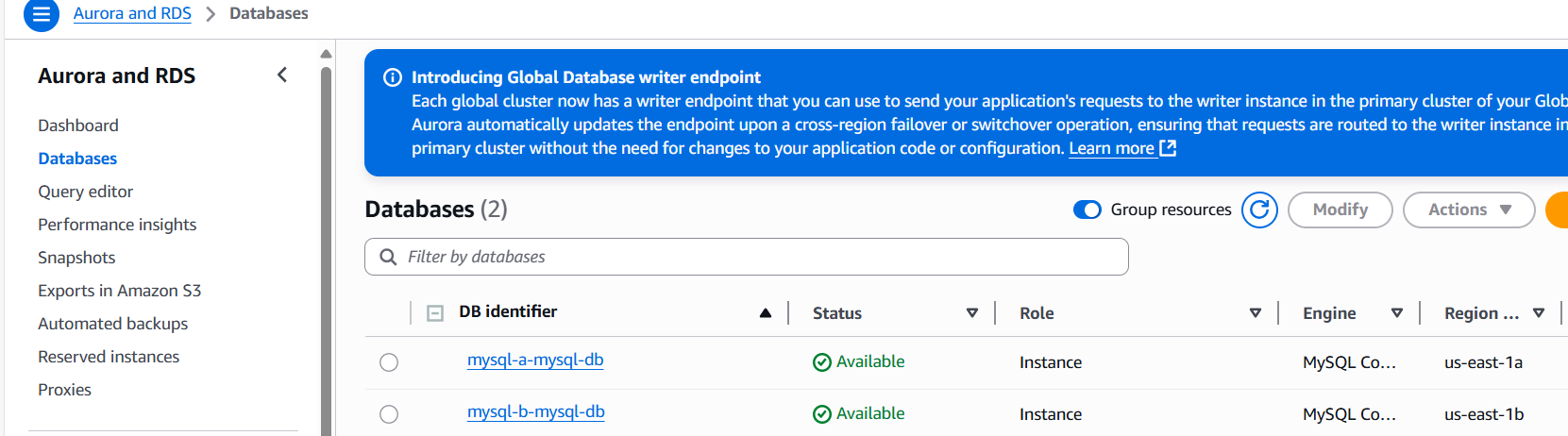
A screenshot of a computer program

AI-generated content may be incorrect.



A computer screen with white text

AI-generated content may be incorrect.



A computer screen shot of a black screen

AI-generated content may be incorrect.

A screen shot of a computer program

AI-generated content may be incorrect.

1. **Bastion Host** (modules/bastion)
   * Depends on network & security groups to allow controlled access

A black screen with white text

AI-generated content may be incorrect.

A computer screen shot of white text

AI-generated content may be incorrect.

A screen shot of a computer

AI-generated content may be incorrect.

A screen shot of a computer program

AI-generated content may be incorrect.



1. **Load Balancer** (modules/load-balancer)
   * Depends on network & security groups

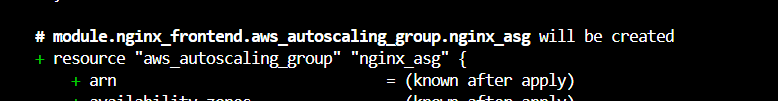
A computer screen with white text

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

1. Autoscaling group



A screenshot of a computer

AI-generated content may be incorrect.

1. **Nginx Frontend** (modules/nginx\_frontend)
   * Depends on Load Balancer and network

A black background with white text

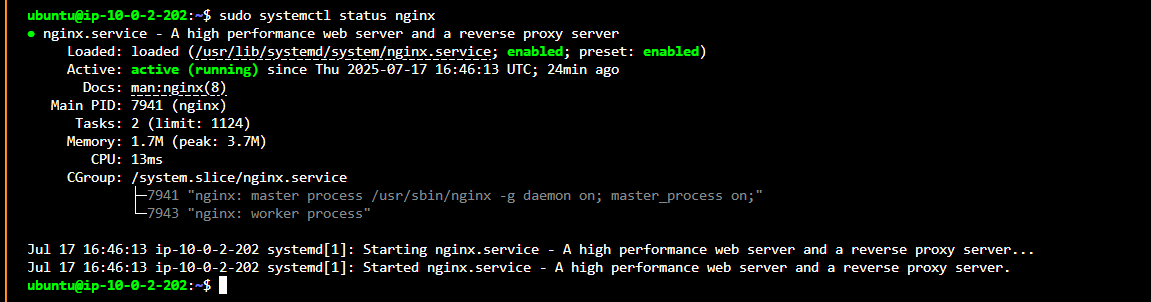
AI-generated content may be incorrect.

A black screen with white text

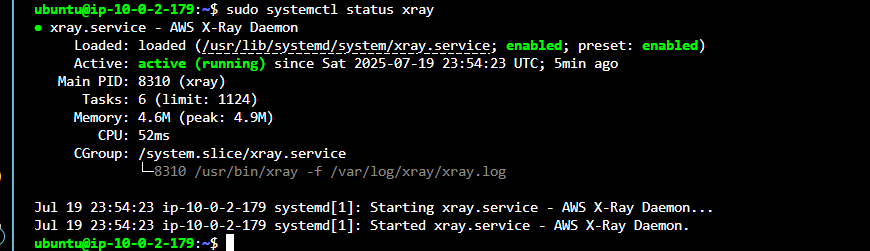
AI-generated content may be incorrect.

A computer screen shot of white text

AI-generated content may be incorrect.



1. **X-ray install on nginx**



A screen shot of a computer screen

AI-generated content may be incorrect.

I created a test\_xray.py

from flask import Flask

from aws\_xray\_sdk.core import xray\_recorder

from aws\_xray\_sdk.ext.flask.middleware import XRayMiddleware

app = Flask(\_\_name\_\_)

xray\_recorder.configure(service='test-xray-app')

XRayMiddleware(app, xray\_recorder)

@app.route('/')

def index():

return "X-Ray test!"

if \_\_name\_\_ == '\_\_main\_\_':

app.run(host='0.0.0.0', port=5000)

**Commands to create and activate a safe Python environment:**

**# 1. Install venv if not already installed**

**sudo apt install python3-venv -y**

**# 2. Create a virtual environment**

**python3 -m venv xray-env**

**# 3. Activate the virtual environment**

**source xray-env/bin/activate**

**# 4. Install Flask and AWS X-Ray SDK inside the venv**

**pip install flask aws-xray-sdk**

**ran this test:**

**AWS\_XRAY\_DAEMON\_ADDRESS=127.0.0.1:2000 python test\_xray.py**

**A screen shot of a computer

AI-generated content may be incorrect.**

**Then, this -tail -f /var/log/xray/xray.logay/xray.log**

A screen shot of a computer

AI-generated content may be incorrect.A screenshot of a computer

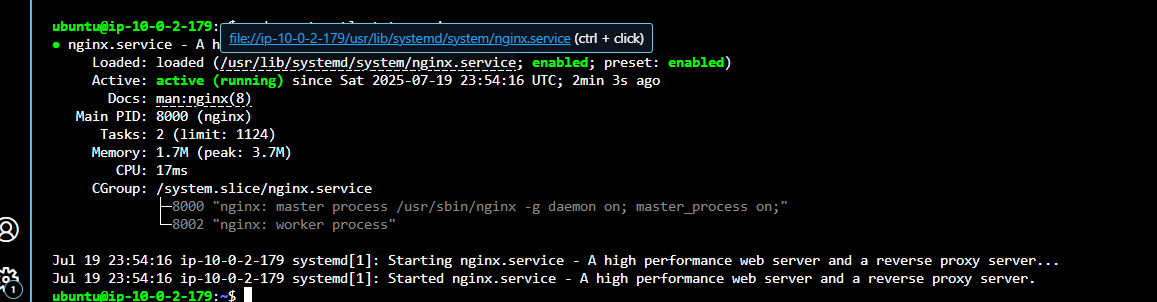
AI-generated content may be incorrect.

A screenshot of a computer

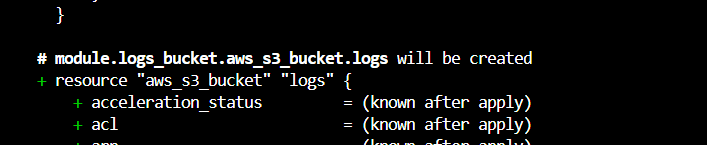
AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.



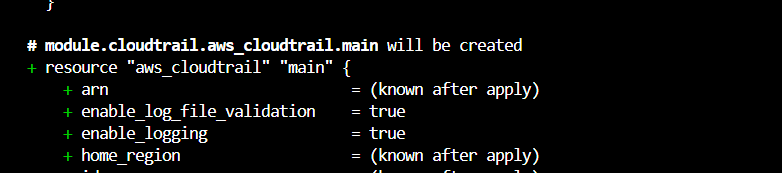
1. **S3\_bucket**



A computer screen shot of a computer screen

AI-generated content may be incorrect.

1. **CloudTrail** (modules/cloudtrail)
   * For audit logging (usually independent but better after IAM & KMS)



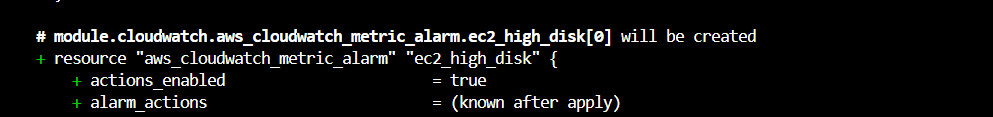
1. **CloudWatch** (modules/cloudwatch)
   * Monitoring for all services

A screen shot of a computer screen

AI-generated content may be incorrect.

A computer screen with white text

AI-generated content may be incorrect.



A screen shot of a computer

AI-generated content may be incorrect.

A screen shot of a computer

AI-generated content may be incorrect.

A computer screen shot of white text

AI-generated content may be incorrect.

A screen shot of a computer

AI-generated content may be incorrect.

✅ Then verify installation:

ls /opt/aws/amazon-cloudwatch-agent/bin/

✅ Manual local configuration

sudo /opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent-config-wizard

🟩 start agent

sudo /opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent-ctl \

-a fetch-config \

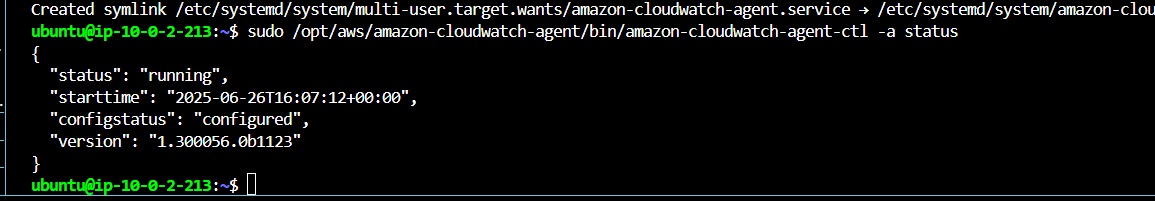
-m ec2 \

-c file:/opt/aws/amazon-cloudwatch-agent/bin/config.json \

-s

sudo /opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent-ctl -a status

🟩 Check Agent Status



1. **WAF** (modules/waf)
   * Applied typically on Load Balancer (needs Load Balancer ready)

A screen shot of a computer program

AI-generated content may be incorrect.

A screen shot of a computer program

AI-generated content may be incorrect.

A computer screen with white text

AI-generated content may be incorrect.

1. **Cost Optimization** (modules/cost\_optimization)
   * Usually optional and for monitoring costs

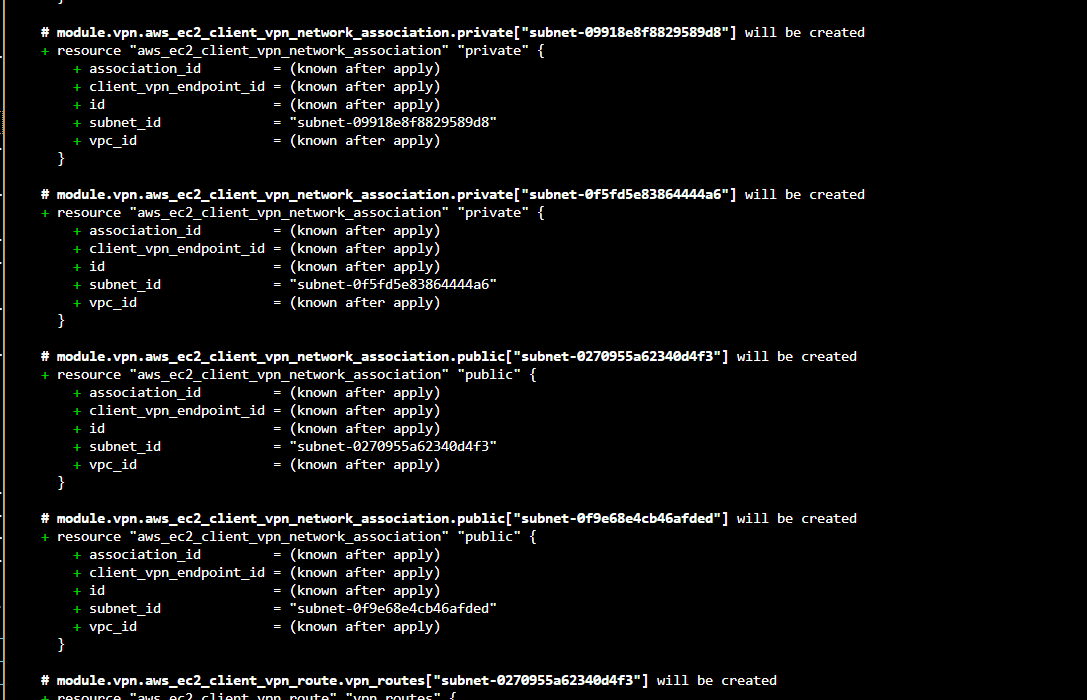
A computer screen with white text

AI-generated content may be incorrect.

1. **VPN**

**A computer screen with white text

AI-generated content may be incorrect.**

****

**A screenshot of a computer program

AI-generated content may be incorrect.**

1. **VPC\_ Peering** (modules/pvc\_peering)

A screen shot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

1. **Lambda\_cleanup**

**A computer screen shot of white text

AI-generated content may be incorrect.**

**A black screen with white text

AI-generated content may be incorrect.**

1. **Jenkins server**

**A computer screen with text on it

AI-generated content may be incorrect.**

**A computer screen shot of white text

AI-generated content may be incorrect.**

**A computer screen shot of a program code

AI-generated content may be incorrect.**

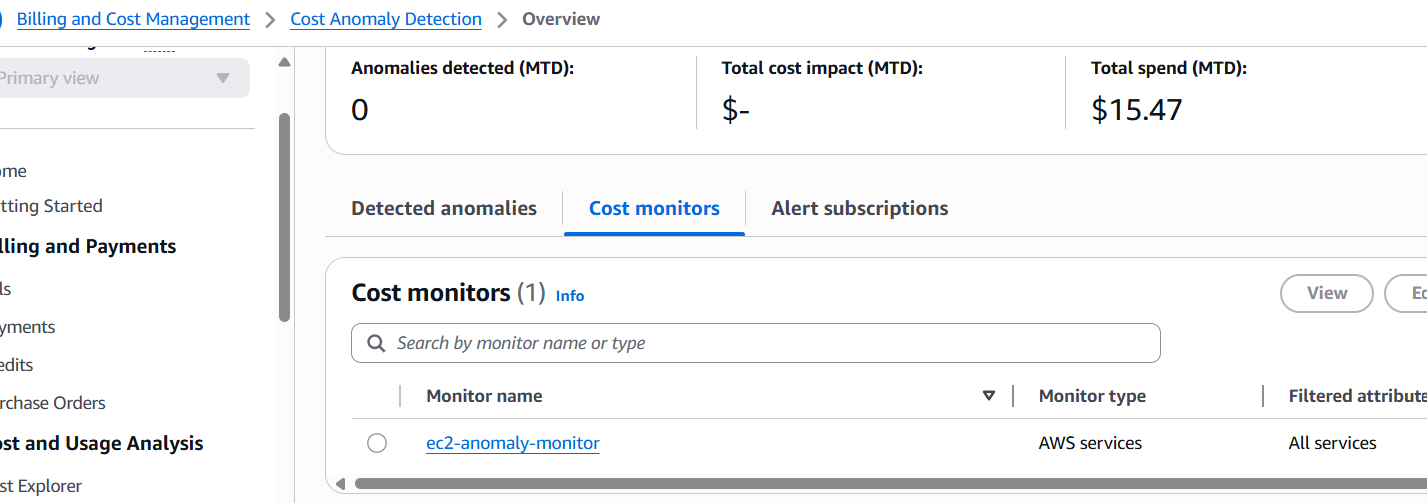
**A black screen with white text

AI-generated content may be incorrect.**

**A computer screen with text and numbers

AI-generated content may be incorrect.**

**Cost optimization**

****

**A screenshot of a computer

AI-generated content may be incorrect.**

**Apply complete! Resources: 112 added, 0 changed, 0 destroyed.**

Outputs:

aws\_sns\_topic\_alerts\_arn = "arn:aws:sns:us-east-1:435329769674:production-cloudwatch-alerts"

bastion\_sg\_id = "sg-05a8916a9ecb76dc2"

elb\_security\_group\_ids = "sg-0bce02dcb022b4419"

jenkins\_security\_group = "sg-0411aeab88deabdba"

kms\_key\_id = "783646e1-6418-4f0c-a616-46036b87a3ee"

nginx\_security\_group\_ids = "sg-02437daa5bf823f6b"

peer\_private\_route\_table\_ids = [

"rtb-080d4c9aa5bb0f8a5",

"rtb-05124d651b44605ce",

]

peer\_public\_route\_table\_ids = [

"rtb-01148a8e51e8117b6",

"rtb-017975d81f49d7f86",

]

private\_route\_table\_ids = [

"rtb-0dbc94692c533c6a6",

"rtb-04a14fc69acb732cb",

]

private\_subnet\_ids = [

"subnet-095575572d18e956f",

"subnet-0f47768ed5ea5b4c5",

]

public\_route\_table\_ids = [

"rtb-0546f428615782458",

]

public\_subnet\_ids = [

"subnet-01f3a610c0dd0ebd9",

"subnet-02bdf00ef8a1125dd",

]

rds\_endpoints = {

"mysql-a" = "mysql-a-mysql-db.cjjbcu9s6nug.us-east-1.rds.amazonaws.com:3306"

"mysql-b" = "mysql-b-mysql-db.cjjbcu9s6nug.us-east-1.rds.amazonaws.com:3306"

}

rds\_instance\_ids = {

"mysql-a" = "db-BITZUJ2UGFBCYLEUL4IMWAAMEA"

"mysql-b" = "db-TCH74Z4ORMGEHON3RVSMZJWZAM"

}

rds\_kms\_key\_id = "f8e78535-dcc1-44c9-b1b0-40b9dc4681bf"

rds\_sg\_id = "sg-064826e673775e25a"

s3\_kms\_key\_id = "af34e426-effd-4e11-a150-4f63b68e7921"

secret\_name = <sensitive>

secrets\_manager\_kms\_key\_id = "783646e1-6418-4f0c-a616-46036b87a3ee"

vpc\_id = "vpc-044050eb08f71eb6c"

vpn\_security\_group\_ids = "sg-05b33842645568995"

kmkou@DESKTOP-OUARLAS MINGW64 /c/refonte-training/CloudEngineer (master)