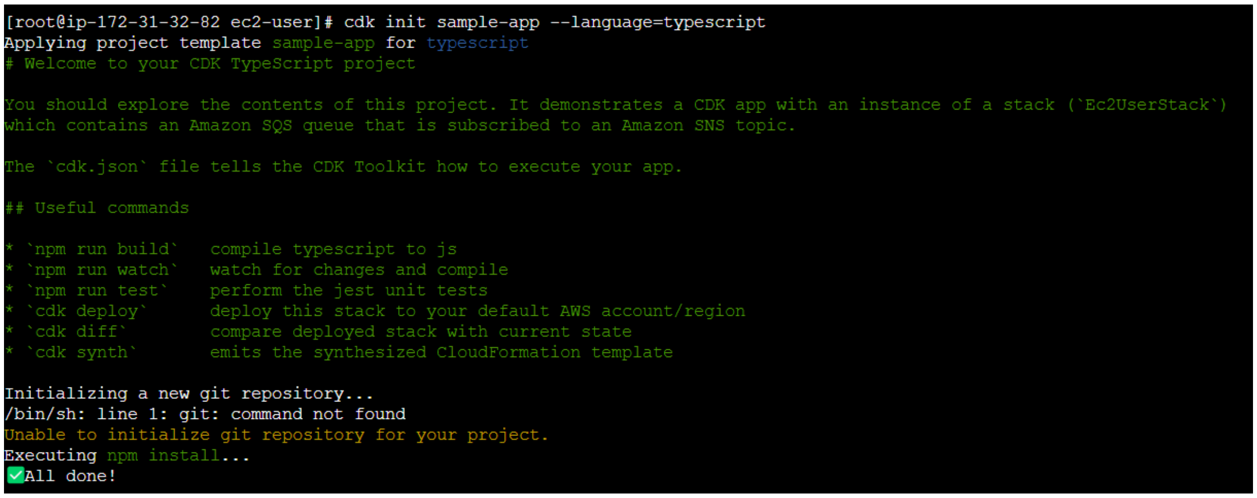
Assignment 4 - CDK

Problem Statement
Create a CDK project that includes the creation of a bucket with versioning enabled, addition of a KMS key and lifecycle rules. Launch an EC2 instance with read/write permissions to the S3 buckets and install an Nginx server on it
Solution:
Initialization of the CDK project.



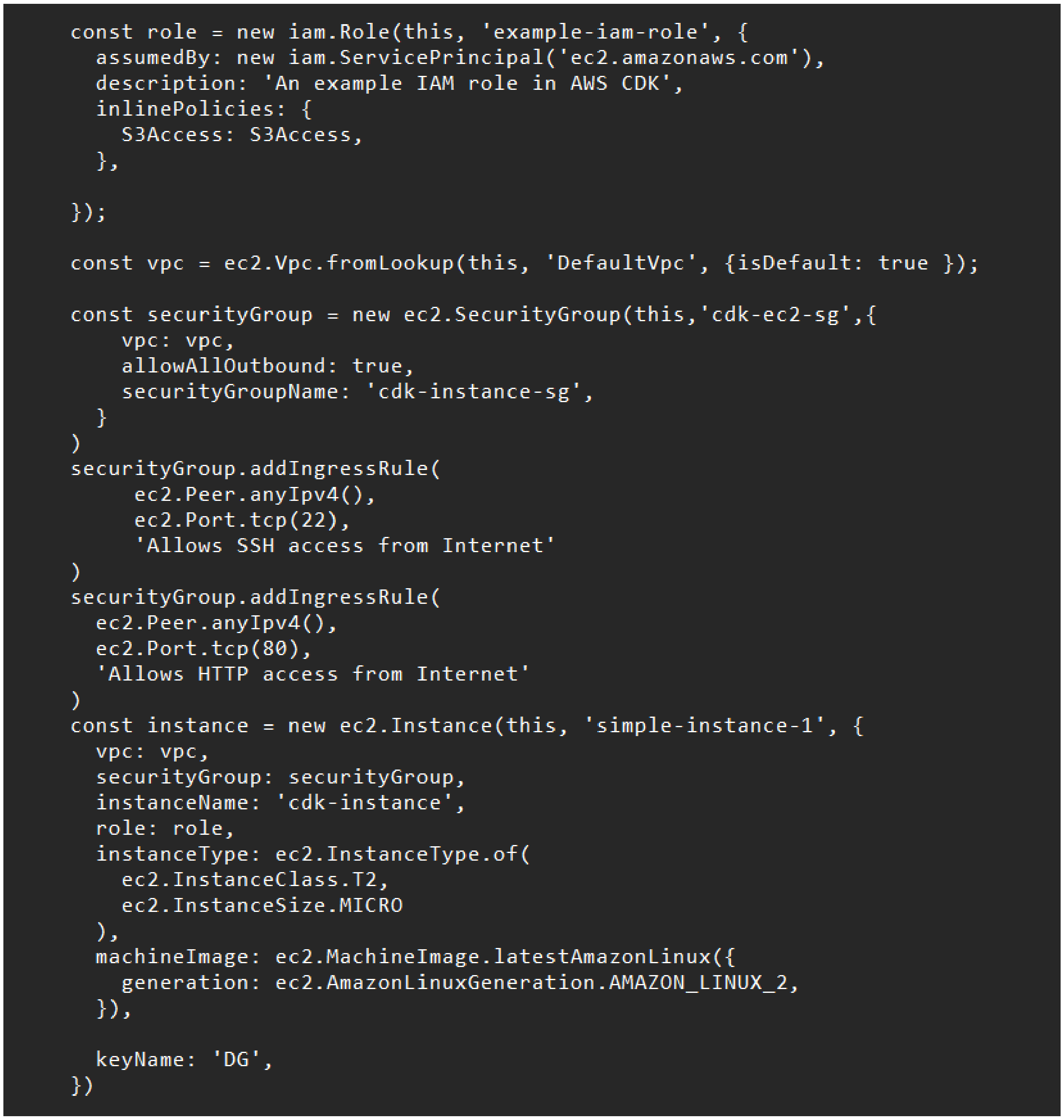
```
[root@ip-172-31-32-82 ec2-user]# cdk init sample-app --language=typescript
Applying project template sample-app for typescript
# Welcome to your CDK TypeScript project

You should explore the contents of this project. It demonstrates a CDK app with an instance of a stack (`Ec2UserStack`)
which contains an Amazon SQS queue that is subscribed to an Amazon SNS topic.

The `cdk.json` file tells the CDK Toolkit how to execute your app.

## Useful commands

* `npm run build`   compile typescript to js
* `npm run watch`   watch for changes and compile
* `npm run test`    perform the jest unit tests
* `cdk deploy`      deploy this stack to your default AWS account/region
* `cdk diff`        compare deployed stack with current state
* `cdk synth`       emits the synthesized CloudFormation template

Initializing a new git repository...
/bin/sh: line 1: git: command not found
Unable to initialize git repository for your project.
Executing npm install...
✅All done!
```

Replace the following content in the respective files.

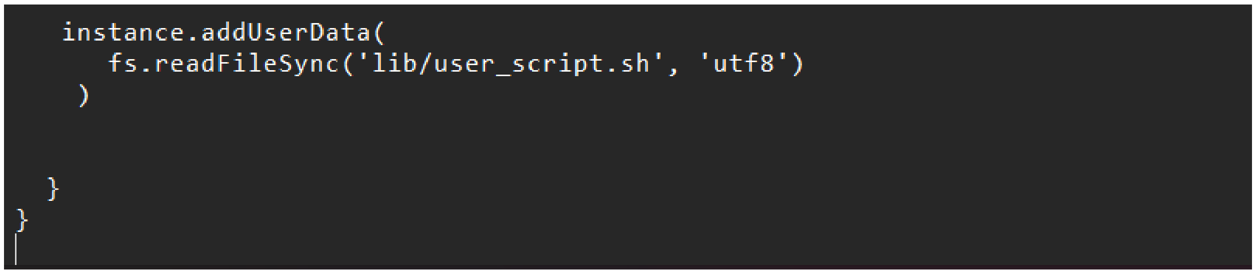
CDK code to create a bucket with versioning enabled, addition of a KMS key and lifecycle rules. Launch an EC2 instance with read/write permissions to the S3 buckets and install an Nginx server on it.

lib/cdk-test-stack.ts

```
import { Duration, Stack, StackProps } from 'aws-cdk-lib';
import * as s3 from 'aws-cdk-lib/aws-s3';
import * as iam from 'aws-cdk-lib/aws-iam';
import * as ec2 from 'aws-cdk-lib/aws-ec2';
import * as kms from 'aws-cdk-lib/aws-kms';
import { Construct } from 'constructs';
import * as fs from 'fs'


export class CdkTestStack extends Stack {
  constructor(scope: Construct, id: string, props?: StackProps) {
    super(scope, id, props);


    const s3Bucket = new s3.Bucket(this, 'Bucket-dg-stack-1', {
    bucketName: "cdk-bucket-test-dg",
    versioned: true,
    encryptionKey: new kms.Key(this, 's3BucketKMSKey'),
    lifecycleRules: [
        {
          transitions: [
            {
              storageClass: s3.StorageClass.INFREQUENT_ACCESS,
              transitionAfter: Duration.days(30),
            },
            {
              storageClass: s3.StorageClass.GLACIER,
              transitionAfter: Duration.days(90),
            },
          ],
        },
      ],
    });

 const S3Access = new iam.PolicyDocument({
      statements: [
        new iam.PolicyStatement({
          resources: ['arn:aws:s3:::*'],
          actions: ['s3:*'],
        }),
      ],
    });
```

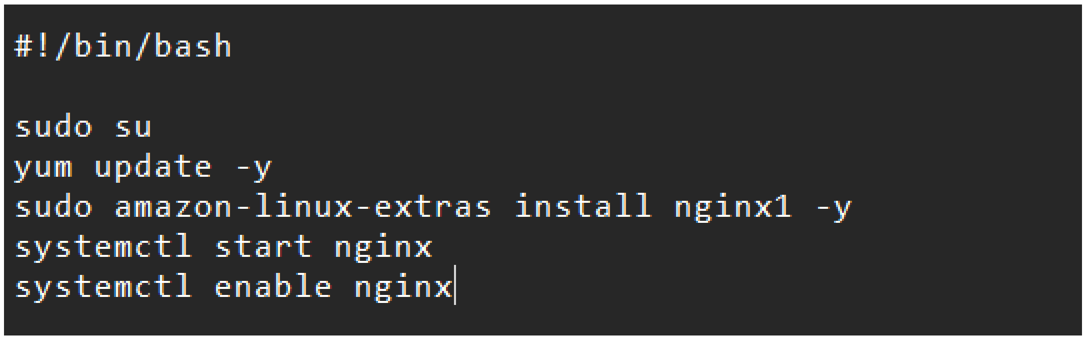


```
    const role = new iam.Role(this, 'example-iam-role', {
      assumedBy: new iam.ServicePrincipal('ec2.amazonaws.com'),
      description: 'An example IAM role in AWS CDK',
      inlinePolicies: {
        S3Access: S3Access,
      },

    });

    const vpc = ec2.Vpc.fromLookup(this, 'DefaultVpc', {isDefault: true });

    const securityGroup = new ec2.SecurityGroup(this,'cdk-ec2-sg',{
        vpc: vpc,
        allowAllOutbound: true,
        securityGroupName: 'cdk-instance-sg',
      }
    )
    securityGroup.addIngressRule(
         ec2.Peer.anyIpv4(),
         ec2.Port.tcp(22),
         'Allows SSH access from Internet'
    )
    securityGroup.addIngressRule(
      ec2.Peer.anyIpv4(),
      ec2.Port.tcp(80),
      'Allows HTTP access from Internet'
    )
    const instance = new ec2.Instance(this, 'simple-instance-1', {
      vpc: vpc,
      securityGroup: securityGroup,
      instanceName: 'cdk-instance',
      role: role,
      instanceType: ec2.InstanceType.of(
        ec2.InstanceClass.T2,
        ec2.InstanceSize.MICRO
      ),
      machineImage: ec2.MachineImage.latestAmazonLinux({
        generation: ec2.AmazonLinuxGeneration.AMAZON_LINUX_2,
      }),

      keyName: 'DG',
    })
```



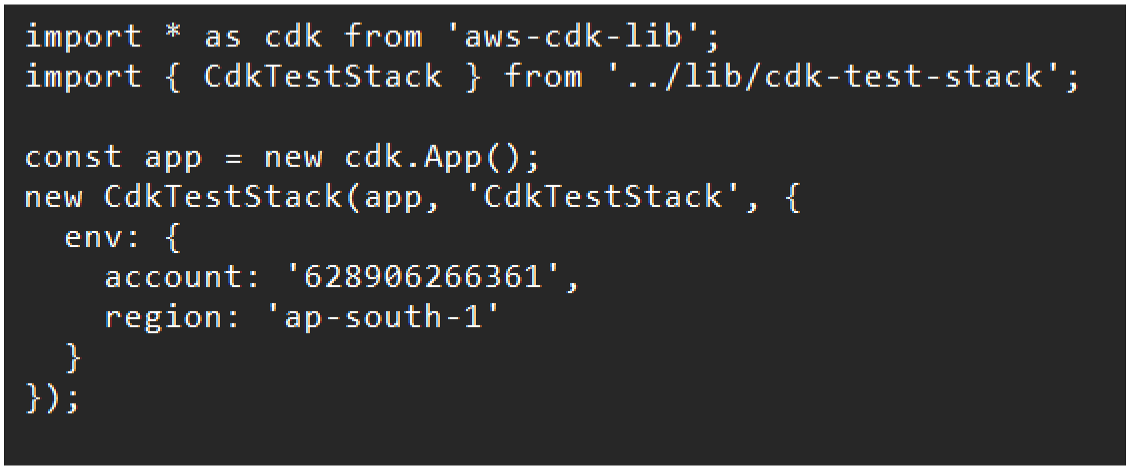
```
   instance.addUserData(
      fs.readFileSync('lib/user_script.sh', 'utf8')
    )


  }
}
```

lib/user_script.sh



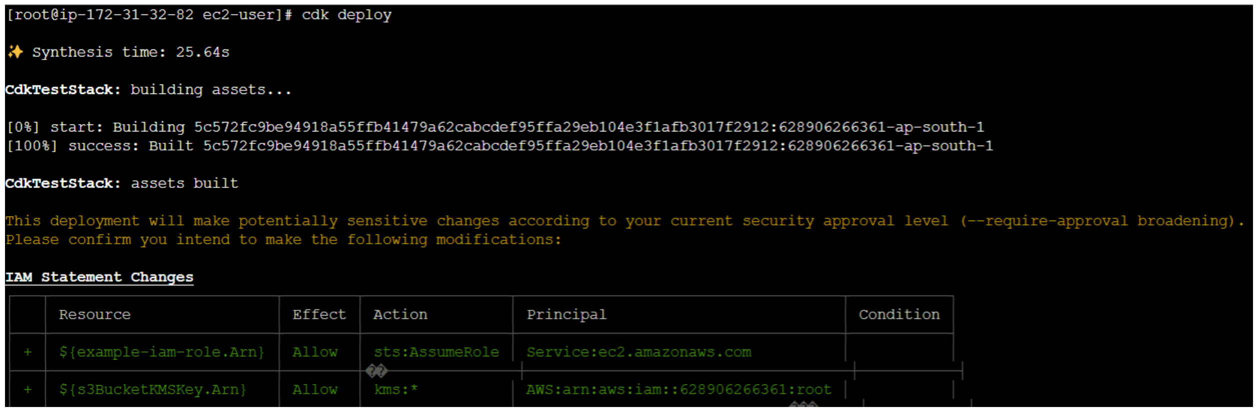
```bash
#!/bin/bash

sudo su
yum update -y
sudo amazon-linux-extras install nginx1 -y
systemctl start nginx
systemctl enable nginx
```

bin/cdk-test.ts



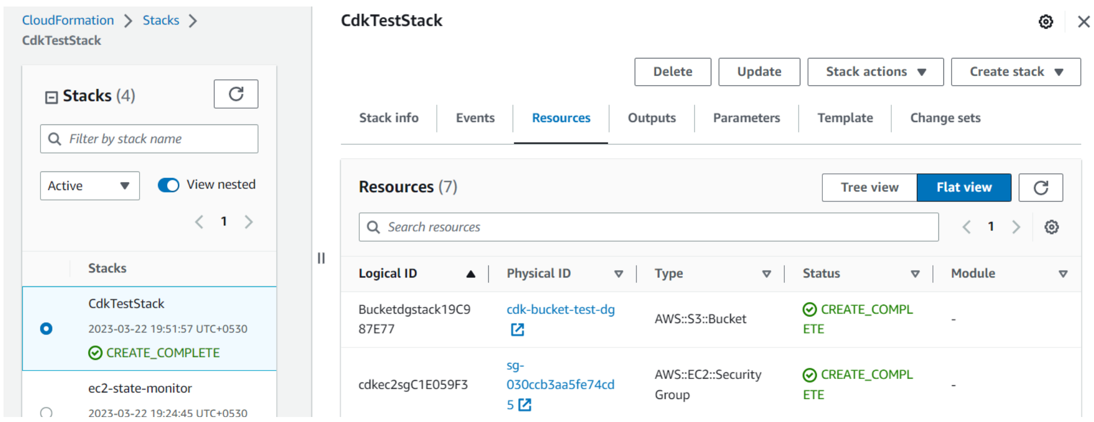
```typescript
import * as cdk from 'aws-cdk-lib';
import { CdkTestStack } from '../lib/cdk-test-stack';

const app = new cdk.App();
new CdkTestStack(app, 'CdkTestStack', {
  env: {
    account: '628906266361',
    region: 'ap-south-1'
  }
});
```

Deploying CDK



```
[root@ip-172-31-32-82 ec2-user]# cdk deploy

✨ Synthesis time: 25.64s

CdkTestStack: building assets...

[0%] start: Building 5c572fc9be94918a55ffb41479a62cabcdef95ffa29eb104e3f1afb3017f2912:628906266361-ap-south-1
[100%] success: Built 5c572fc9be94918a55ffb41479a62cabcdef95ffa29eb104e3f1afb3017f2912:628906266361-ap-south-1

CdkTestStack: assets built

This deployment will make potentially sensitive changes according to your current security approval level (--require-approval broadening).
Please confirm you intend to make the following modifications:

IAM Statement Changes
┌───┬─────────────────────────┬────────┬──────────────────┬──────────────────────────────────────────┬───────────┐
│   │ Resource                │ Effect │ Action           │ Principal                                │ Condition │
├───┼─────────────────────────┼────────┼──────────────────┼──────────────────────────────────────────┼───────────┤
│ + │ ${example-iam-role.Arn} │ Allow  │ sts:AssumeRole   │ Service:ec2.amazonaws.com                │           │
├───┼─────────────────────────┼────────┼──────────────────┼──────────────────────────────────────────┼───────────┤
│ + │ ${s3BucketKMSKey.Arn}   │ Allow  │ kms:*            │ AWS:arn:aws:iam::628906266361:root       │           │
```

Resources successfully created with the cloudformation.



CloudFormation > Stacks > CdkTestStack

Stacks (4)

Filter by stack name

Active ▼ View nested

	Stacks
●	CdkTestStack 2023-03-22 19:51:57 UTC+0530 CREATE_COMPLETE
○	ec2-state-monitor 2023-03-22 19:24:45 UTC+0530

CdkTestStack

Delete | Update | Stack actions ▼ | Create stack ▼

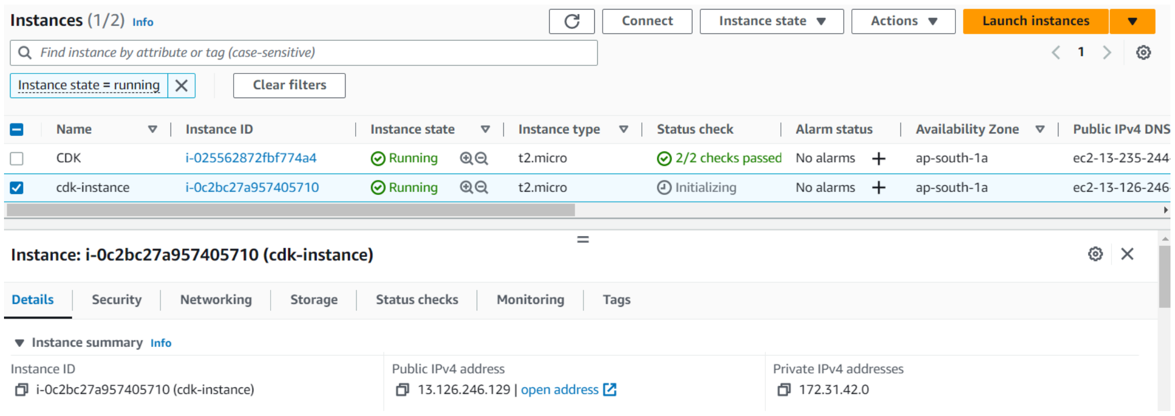
Stack info | Events | Resources | Outputs | Parameters | Template | Change sets

Resources (7) Tree view | Flat view

Search resources

Logical ID	Physical ID	Type	Status	Module
Bucketdgstack19C987E77	cdk-bucket-test-dg	AWS::S3::Bucket	CREATE_COMPLETE	-
cdkec2sgC1E059F3	sg-030ccb3aa5fe74cd5	AWS::EC2::Security Group	CREATE_COMPLETE	-

New instance launched with the CDK code.



Instances (1/2) Info

Connect | Instance state ▼ | Actions ▼ | Launch instances ▼

Find instance by attribute or tag (case-sensitive)

Instance state = running ✕ | Clear filters

	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS
☐	CDK	i-025562872fbf774a4	Running	t2.micro	2/2 checks passed	No alarms +	ap-south-1a	ec2-13-235-244
☑	cdk-instance	i-0c2bc27a957405710	Running	t2.micro	Initializing	No alarms +	ap-south-1a	ec2-13-126-246

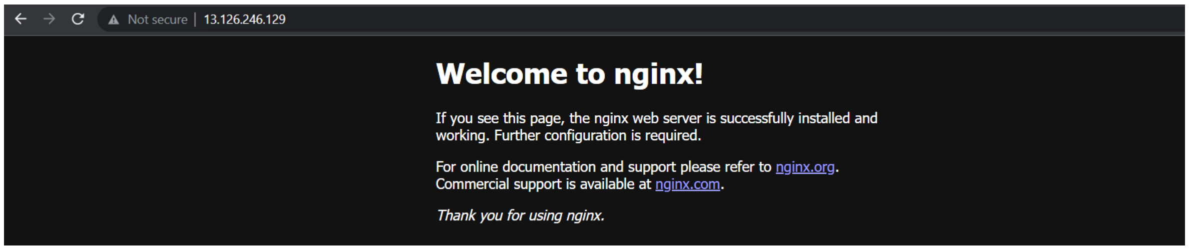
Instance: i-0c2bc27a957405710 (cdk-instance)

Details | Security | Networking | Storage | Status checks | Monitoring | Tags

▼ Instance summary Info

Instance ID	Public IPv4 address	Private IPv4 addresses
i-0c2bc27a957405710 (cdk-instance)	13.126.246.129 | open address	172.31.42.0

Tried to access the nginx server with the public IP address.



Not secure | 13.126.246.129

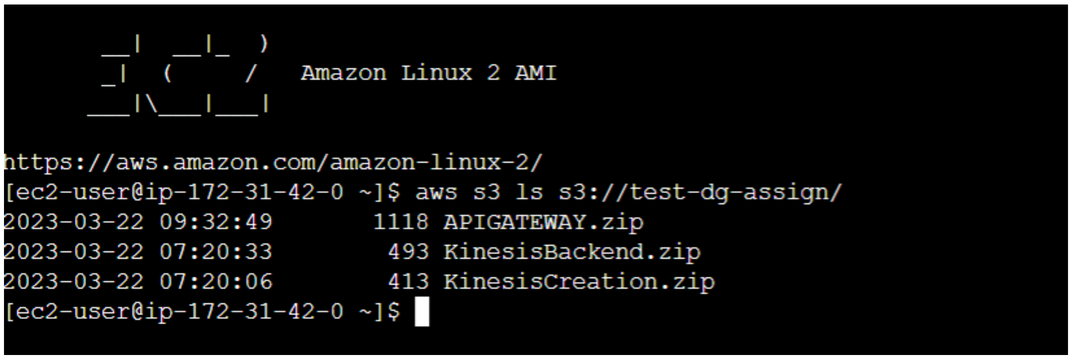
Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.

Objects in the bucket accessed successfully.



```
       __|  __|_  )
       _|  (     /   Amazon Linux 2 AMI
      ___|\___|___|

https://aws.amazon.com/amazon-linux-2/
[ec2-user@ip-172-31-42-0 ~]$ aws s3 ls s3://test-dg-assign/
2023-03-22 09:32:49       1118 APIGATEWAY.zip
2023-03-22 07:20:33        493 KinesisBackend.zip
2023-03-22 07:20:06        413 KinesisCreation.zip
[ec2-user@ip-172-31-42-0 ~]$
```