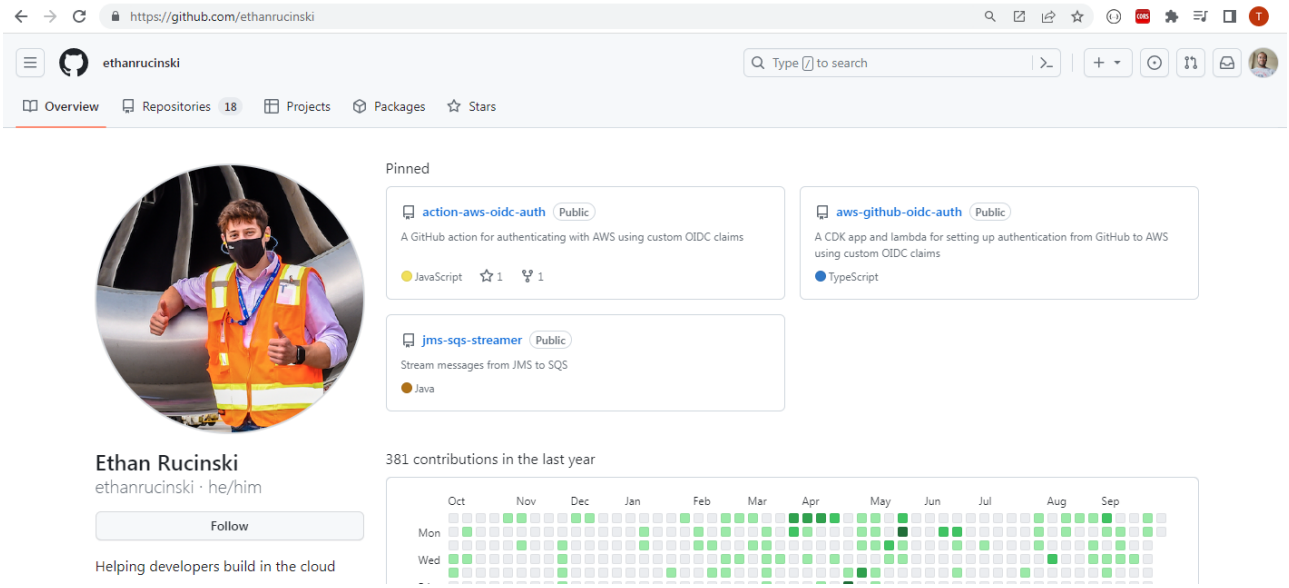
AWS re:Invent 2022 - How to reuse patterns when developing infrastructure as code (DOP302)

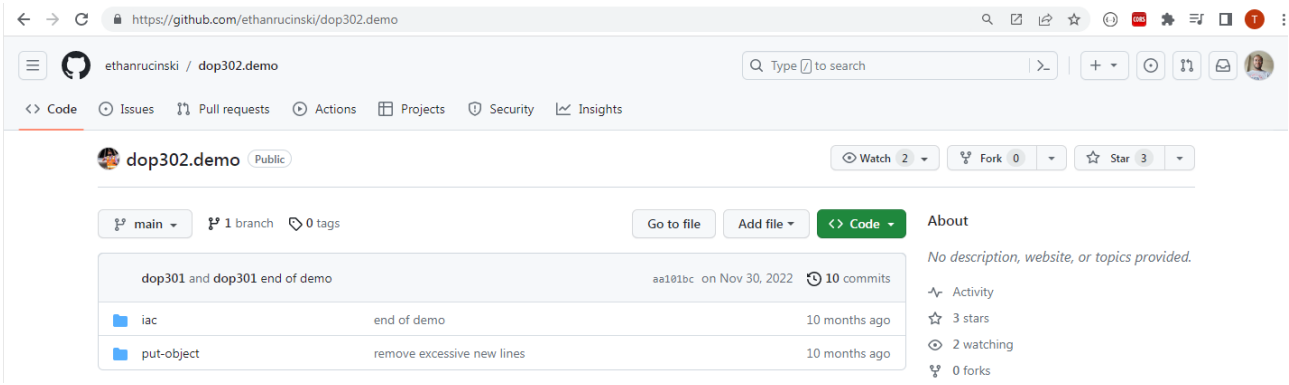
AWS Events
93.8K subscribers
Subscribe

72 Share Download Clip ...

4,387 views Dec 5, 2022 AWS re:Invent 2022 - SysOps - Chris Williams
Explore AWS Cloud Development Kit (AWS CDK) constructs and AWS CloudFormation modules and how they can help make it easier to build applications on AWS. Learn best practices from United Airlines, who shares how they have used AWS CDK to create an internal library of infrastructure components that their development team can reuse to accelerate application delivery.



https://github.com/ethanrucinski/dop302.demo

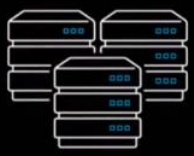


What am I going to talk about?

- How infrastructure is evolving
- Challenges organizations face
- CDK @ United
- IaC Community Engagement Program

"One of my most productive days was throwing away 1000 lines of code."

How infrastructure is evolving



On-premises infrastructure

Build environments with available infrastructure

Lengthy delivery times

Limited innovation

Cloud infrastructure

Ubiquitous access to infrastructure through on-premises virtualization

Deliver infrastructure as code

Eliminate one roadblock to deploying application environments automatically

Application-based EaaS

Ubiquitous access to infrastructure and AWS Cloud services

Unique requirements for different applications

Ability to automate the process of deploying application environments

The skills gap makes this more difficult

Huge demand for application environments

Low supply of people with the skills to support them

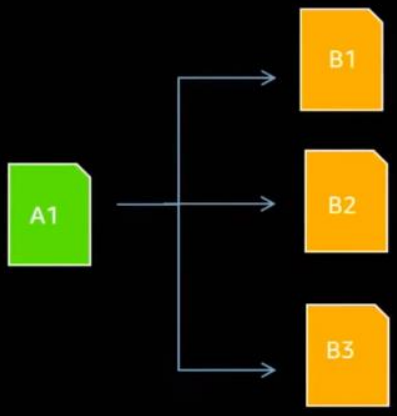
Bottlenecks that slow down delivery and reduce quality (velocity)

Challenges

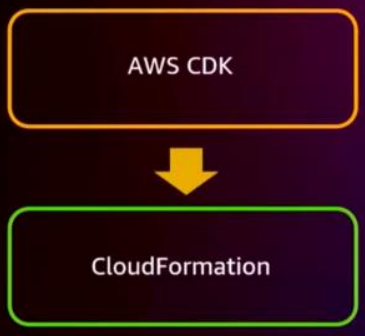
- Builders are spending too much time trying to figure out resource intricacies and best practices
- Applications often share common architectural elements
  - For example, one or more Lambda functions that can read/write to an Amazon DynamoDB table
- Modeling infrastructure becomes repetitive, “re-implementing the wheel”
  - Defining and configuring each service
  - Integrating with other services
  - Adhering to best practices
- Takes valuable development time from building unique features

AWS to the rescue

AWS CloudFormation modules



AWS Cloud Development Kit (AWS CDK)



AWS CDK at United Airlines

- Business considerations
- Agility
- People processes
- Cloud adoption

Our roadmap to the solution

Key steps we took and decisions we made

How we solved the problem

Where we are today

About me

ETHAN RUCINSKI | PRINCIPAL ARCHITECT AT UNITED AIRLINES

- Chicago, IL
- Previously
  - Manager – network operations
  - Operations data analyst
- Fun facts
  - Have flown ~1M miles with United

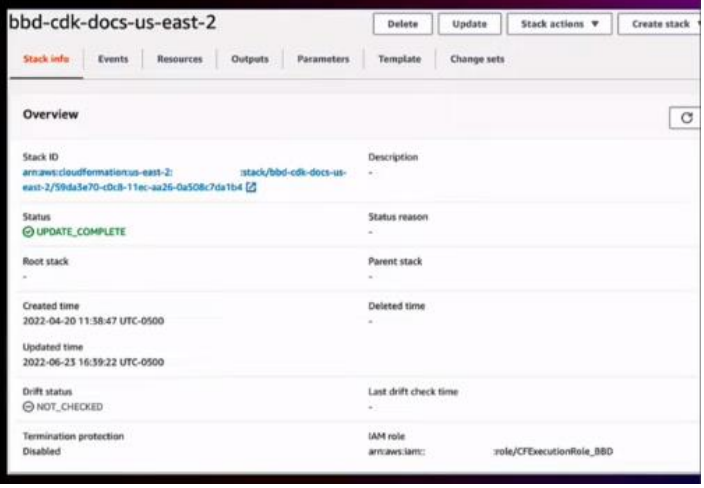
United has a large and growing cloud footprint

- >200 apps in the cloud
- >120 active AWS accounts
- ~100 different services

United's mass cloud migration is in full swing

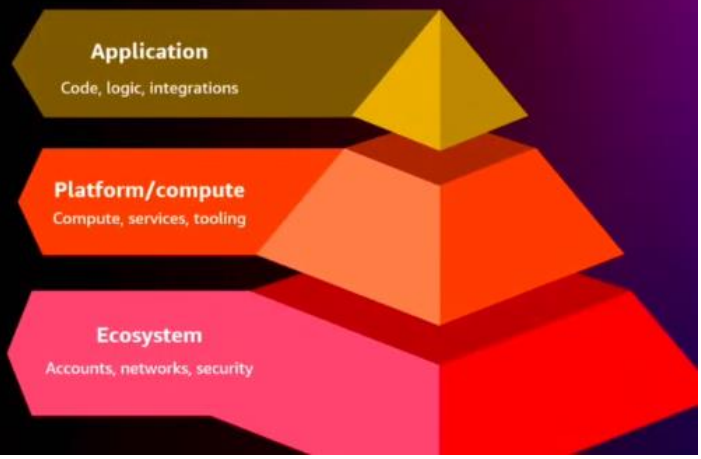
App teams own their cloud destiny at United

- Developers implement and deploy their own cloud infrastructure
- Strict policies protect higher environments
- Infrastructure as code enforces consistency across deployments



Where does PlatformOps fit in?

- CI/CD tooling
- Cloud reference architectures
- Support app teams new to the cloud



First-mover teams become experts in AWS CloudFormation

```yaml
Service:
    Type: AWS::ECS::Service
    Properties:
        Cluster:
            Fn::ImportValue: !Sub ${AppID}-ecs-cluster
        DesiredCount: !Sub ${DesiredCount}
        LoadBalancers:
            - ContainerName: !Sub ${AppID}-${ServiceName}
              ContainerPort: 3000
              TargetGroupArn: !Ref "TargetGroup"
        NetworkConfiguration:
            AwsvpcConfiguration:
                AssignPublicIp: DISABLED
                SecurityGroups:
                    - Fn::ImportValue: !Sub ${AppID}-ecs-service-security-group
                Subnets:
                    - !Ref "SubnetA"
                    - !Ref "SubnetB"
                    - !Ref "SubnetC"
        LaunchType: FARGATE
        ServiceName: !Sub ${AppID}-${ServiceName}
        TaskDefinition: !Ref TaskDefintion
        PropagateTags: SERVICE
```

First-mover teams connect everything

```yaml
TaskDefintion:
    Type: AWS::ECS::TaskDefinition
    Properties:
        Cpu: !Sub ${T
        RequiresCompa
            - FARGATE
        Family: !Sub
        NetworkMode:
        Memory: !Sub
        ExecutionRole
            Fn::Impor
        TaskRoleArn:
            Fn::Impor
        ContainerDefi
```

```yaml
Service:
    Type: AWS::ECS::Service
    Properties:
        Cluster:
            Fn::ImportValue: !S
        DesiredCount: !Sub ${De
        LoadBalancers:
            - ContainerName: !S
              ContainerPort: 30
              TargetGroupArn:
        NetworkConfiguration:
            AwsvpcConfiguration
                AssignPublicIp:
                SecurityGroups:
                    - Fn::Impor
                Subnets:
                    - !Ref "Sub
                    - !Ref "Sub
                    - !Ref "Sub
```

```yaml
TargetGroup:
    Type: AWS::ElasticLoadBalancingV2::TargetGroup
    Properties:
        Name: !Sub
        Port: 3000
        Protocol: HT
        TargetType:
        VpcId: !Ref
        HealthCheckP
        HealthCheckP
        Tags:
            - Key:
              Value:
```

```yaml
ApiHttpsListenerRule:
    Type: AWS::ElasticLoadBalancingV2::ListenerRule
    Properties:
        Actions:
            - Type: forward
              TargetGroupArn: !Ref TargetGroup
        Conditions:
            - Field: path-pattern
              Values:
                  - !Sub "/${RoutingRulePath}*"
        ListenerArn:
            Fn::ImportValue: !Sub ${AppID}-elb-https-listener
        Priority: !Ref LoadBalancerPriority
```

First-mover teams repeat each other

```
Service:
  Type: AWS::ECS::Service
  Properties:
    Cluster:
      Fn::ImportValue: !Sub ${AppID}-ecs-cluster
    DesiredCount: !Sub ${DesiredCount}
    LoadBalancers:
      - ContainerName: !Sub ${AppID}-${ServiceNam
        ContainerPort: 3000
        TargetGroupArn: !Ref "TargetGroup"
    NetworkConfiguration:
      AwsvpcConfiguration:
        AssignPublicIp: DISABLED
        SecurityGroups:
          - Fn::ImportValue: !Sub ${AppID}-ec
        Subnets:
          - !Ref "SubnetA"
          - !Ref "SubnetB"
          - !Ref "SubnetC"
    LaunchType: FARGATE
    ServiceName: !Sub ${AppID}-${ServiceName}
    TaskDefinition: !Ref TaskDefintion
    PropagateTags: SERVICE
    Tags:
      - Key: ApplicationID
        Value: !Ref AppID
```

```
Service:
  Type: AWS::ECS::Service
  Properties:
    Cluster:
      Fn::ImportValue: !Sub ${AppID}-ecs-cluster
    DesiredCount: !Sub ${DesiredCount}
    LoadBalancers:
      - ContainerName: !Sub ${AppID}-${ServiceNam
        ContainerPort: 80
        TargetGroupArn: !Ref "TargetGroup"
    NetworkConfiguration:
      AwsvpcConfiguration:
        AssignPublicIp: DISABLED
        SecurityGroups:
          - Fn::ImportValue: !Sub ${AppID}-ec
        Subnets:
          - !Ref "SubnetA"
          - !Ref "SubnetB"
          - !Ref "SubnetC"
    LaunchType: FARGATE
    ServiceName: !Sub ${AppID}-${ServiceName}
    TaskDefinition: !Ref TaskDefintion
    PropagateTags: SERVICE
    Tags:
      - Key: ApplicationID
        Value: !Ref AppID
```

```
Service:
  Type: AWS::ECS::Service
  Properties:
    Cluster:
      Fn::ImportValue: !Sub ${AppID}-ecs-cluster
    DesiredCount: !Sub ${DesiredCount}
    LoadBalancers:
      - ContainerName: !Sub ${AppID}-${ServiceName}
        ContainerPort: 8080
        TargetGroupArn: !Ref "TargetGroup"
    NetworkConfiguration:
      AwsvpcConfiguration:
        AssignPublicIp: DISABLED
        SecurityGroups:
          - Fn::ImportValue: !Sub ${AppID}-ecs-service-security-grou
        Subnets:
          - !Ref "SubnetA"
          - !Ref "SubnetB"
          - !Ref "SubnetC"
    LaunchType: FARGATE
    ServiceName: !Sub ${AppID}-${ServiceName}
    TaskDefinition: !Ref TaskDefintion
    PropagateTags: SERVICE
    Tags:
      - Key: ApplicationID
        Value: !Ref AppID
```

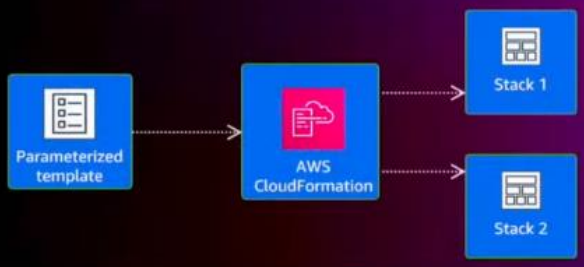
Parameterized templates reduce repetition

- DevOps teams produce standard templates
- Applications leverage latest templates for current best practices
- Expert teams build their own templates for new features

```
AWSTemplateFormatVersion: "2010-09-09"
Description: Template for standing up generic Lambda
Parameters:
  LambdaName:
    Type: String
    Description: Name for the Lambda and associated resources
  AppCI:
    Type: String
    Description: Lowercase AppCI Example abc
  LambdaFunctionHandler:
    Description: The handler for your Lambda function
    Type: String
    Default: main.lambda_handler
  LambdaPackageS3Bucket:
    Description: Bucket where the lambda package is placed Example this-is-my-s3-bucket-name
    Type: String
  LambdaPackage:
    Description: The name of your Lambda deployment package Example lambda.zip
    Type: String
    Default: lambda.zip
  LambdaRuntimeEnvironment:
    Description: The runtime environment for your Lambda function
    Type: String
    Default: python3.6
  ExecutionRoleArn:
    Description: ARN of the role for the lambda to use
    Type: String
  SecurityGroup:
    Description: ID of a security group to attach to this lambda
    Type: AWS::EC2::SecurityGroup::Id
  SubnetA:
    Description: ID of a subnet in which to run the lambda
    Type: AWS::EC2::Subnet::Id
  SubnetB:
    Description: ID of a subnet in which to run the lambda
    Type: AWS::EC2::Subnet::Id

Resources:
  Lambda:
    Type: AWS::Lambda::Function
    Properties:
      FunctionName: !Ref LambdaName
```

Growing pains

CloudFormation
Parameters and intrinsic functions



Things that get stale	Why they got stale
Best practices	Best practices change constantly
Policies	Policies adapt to new features and new risks
The code I got from Bob	Bob's code had bugs, and he wrote it last year

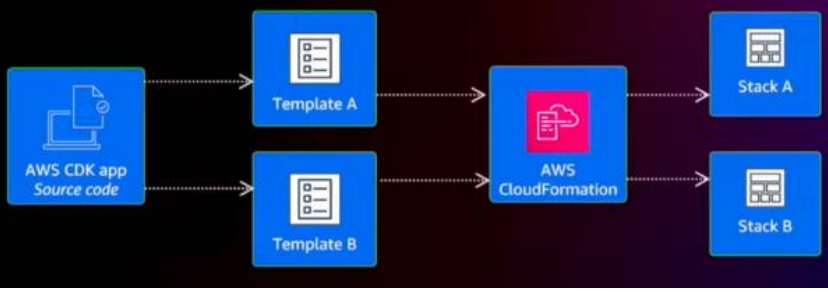
Supporting United's new cloud workforce

- Meeting teams where they're at
- Centrally managed living common patterns
- Policies encapsulated and updated behind the scenes
- Extensive customization for advanced teams

Advanced IaC tooling

CloudFormation
Parameters and intrinsic functions

AWS CDK
Typed OO language: loops, conditions, inheritance, etc.



AWS CDK

A MULTI-LANGUAGE SOFTWARE DEVELOPMENT FRAMEWORK FOR MODELING CLOUD INFRASTRUCTURE AS REUSABLE COMPONENTS

```
class UrlShortener extends Stack {
  constructor(scope: App, id: string, props?: UrlShortenerProps) {
    super(scope, id, props);

    const vpc = new ec2.Vpc(this, 'vpc', { maxAzs: 2 });
    const cluster = new ecs.Cluster(this, 'cluster', { vpc: vpc });
    const service = new patterns.NetworkLoadBalancedFargateService(this, 'sample-app', {
      cluster,
      taskImageOptions: {
        image: ecs.ContainerImage.fromAsset('ping'),
      },
      dom
    });
    // Setup AutoScaling policy
    const scaling = service.service.autoScaleTas
    scaling.scaleOnCpuUtilization('CpuScaling', {
      targetUtilizationPercent: 50,
      scaleInCooldown: Duration.seconds(60),
      scaleOutCooldown: Duration.seconds(60)
    });
  }
}
```

domainName
domainZone

(property) patterns.NetworkLoadBalancedServiceBaseProps.domainName?: string | undefined

The domain name for the service, e.g. "api.example.com."

@default

No domain name.

Familiar
Your language
Just code

Tool support
Autocomplete
Inline documentatio

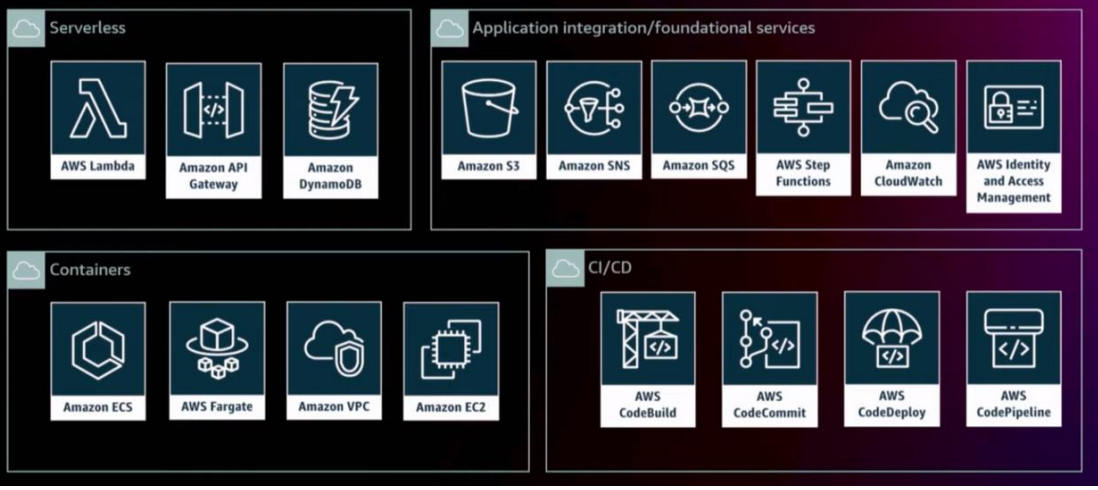
Abstraction
Sane defaults
Reusable classes

Python JS TS Java C# Go

Construct levels

L3+	Purpose-built constructs	Opinionated abstractions
L2	AWS constructs	High-level service constructs
L1	CloudFormation resources	Automatically generated

AWS Construct Library



L1

```
new CfnBucket(this, 'MyBucket', {bucketName: 'my-bucket'})
```

cdk synth

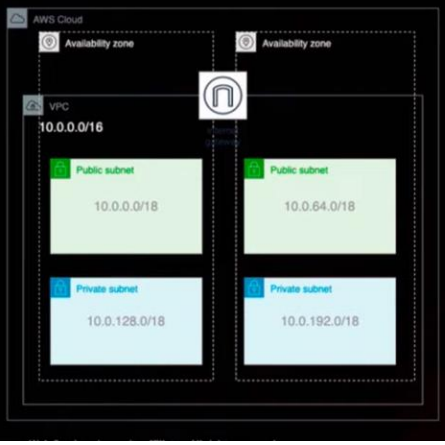
```
Resources:
  myBucket:
    Type: AWS::S3::Bucket
    Properties:
      BucketName: my-bucket
```

- Generated mappings from CloudFormation specificatic
- abc.CfnXyz → AWS::ABC::XYZ CloudFormation resourc
- ec2.CfnInstance → AWS::EC2::Instance
- kms.CfnKey → AWS::KMS::Key

L2

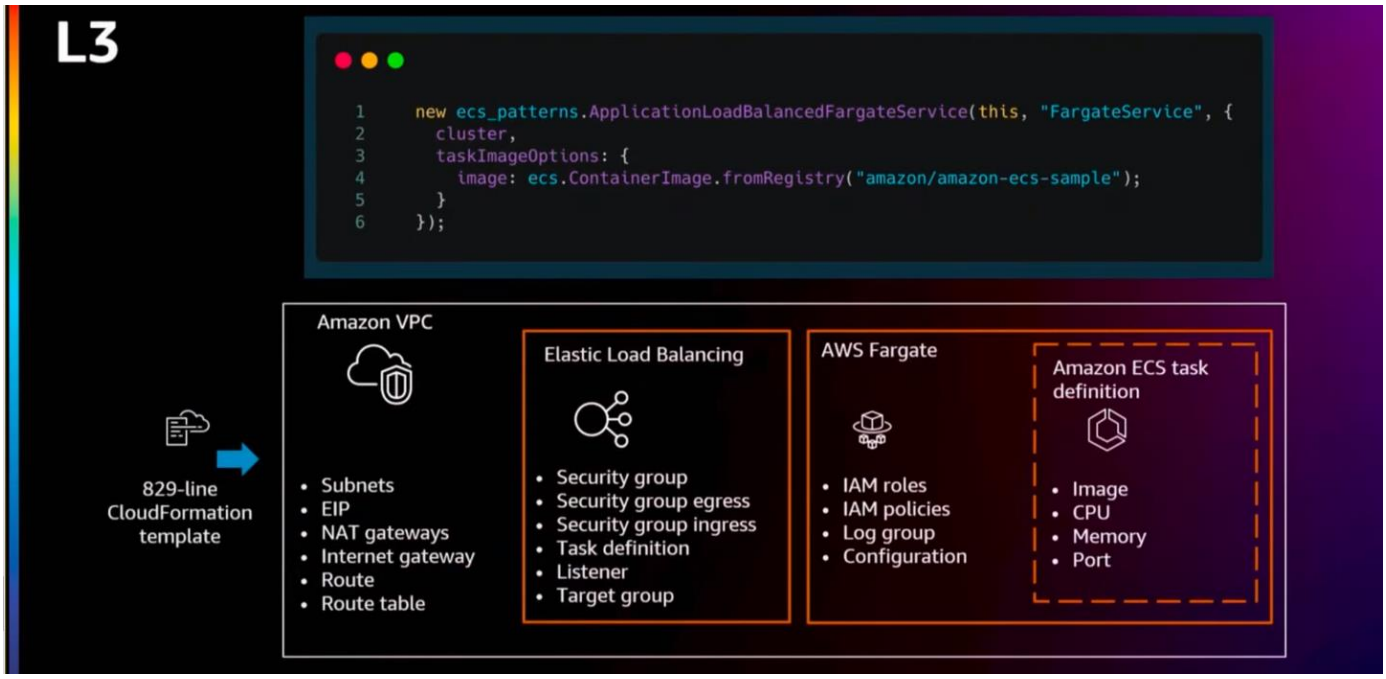
```
new ec2.Vpc(this, 'MyVPC')
```

cdk synth

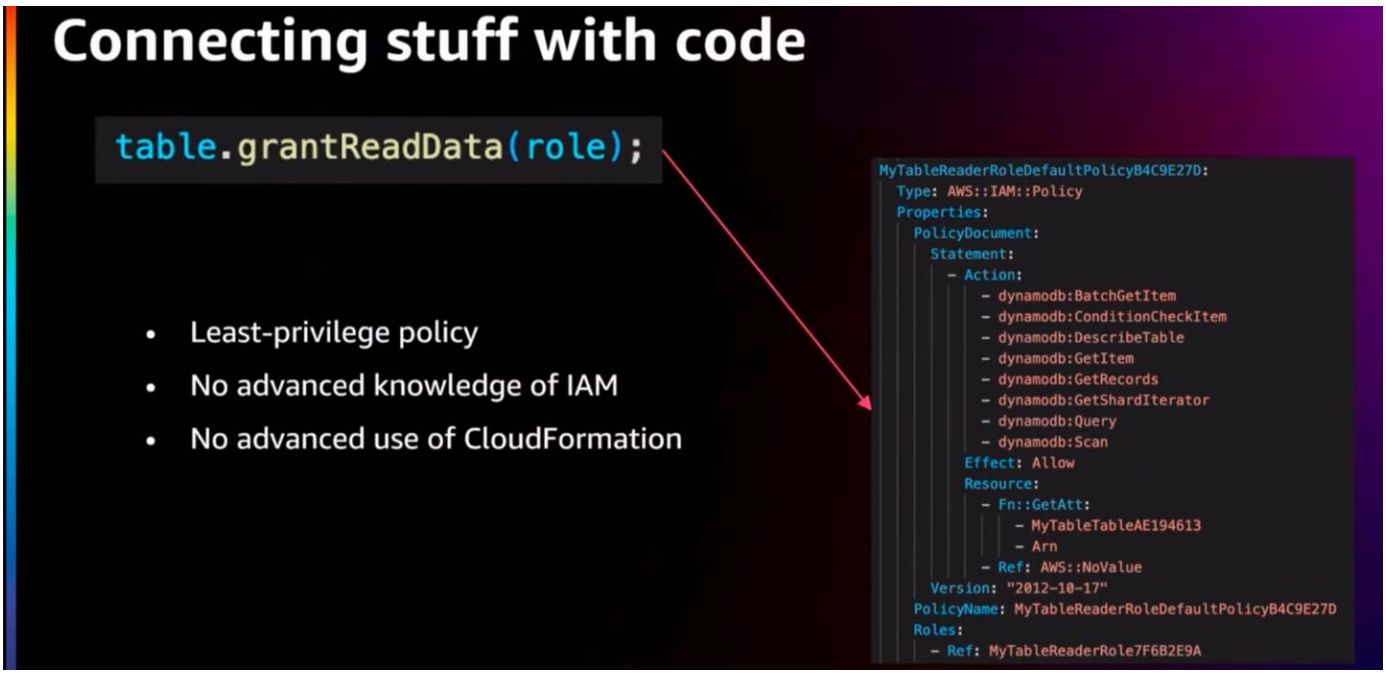


- Ready-to-use VPC setup
- 65,536 IPs split equally between 4 subnets
- If you provide a Region → adjust to 3 AZs
- Everything is optional; change any parameter
- Sane default values

aws



This is an application load balanced Fargate service



We extended AWS Constructs from their library for our own use-cases to start with.

Aspects remove traditional policy speed bumps

- New cloud users don't have PhDs in IAM
- Tagging, naming, etc. don't generally affect applications
- Policies need to appear in infrastructure as code

```
const boundary = iam.ManagedPolicy.fromManagedPolicyArn(
  this,
  'permissions-boundary',
  `arn:aws:iam::${
    this.account
  }:policy/AppPolicy_${props.applicationCI.toUpperCase()}`
);
iam.PermissionsBoundary.of(this).apply(boundary);

cdk.Tags.of(this).add('ApplicationCI', this.applicationCI);
cdk.Tags.of(this).add('Environment', this.ualEnvironment);
cdk.Tags.of(this).add('Region', this.region);
```

L2 at United

```
const table = new UalTable(this, "MyTable", {
    partitionKey: {
        name: "pk",
        type: AttributeType.STRING,
    },
});
```

- Policy-compliant Amazon DynamoDB table
- Encryption enabled
- All tags enforced
- SSM Parameter for CloudFormation-generated name

```
MyTableTableAE194613:
  Type: AWS::DynamoDB::Table
  Properties:
    KeySchema:
      - AttributeName: pk
        KeyType: HASH
    AttributeDefinitions:
      - AttributeName: pk
        AttributeType: S
    BillingMode: PAY_PER_REQUEST
    SSESpecification:
      SSEEnabled: true
    Tags:
      - Key: ApplicationCI
        Value: abc
      - Key: Environment
        Value: dev
      - Key: Region
        Value: us-east-2
      - Key: ual-cdk:version
        Value: 4.4.0
```

```
MyTableTableMyTableSSMParameter86BAEA87:
  Type: AWS::SSM::Parameter
  Properties:
    Type: String
    Value:
      Ref: MyTableTableAE194613
    Name: /abc/dynamodb/dev/MyTable_TABLE_NAME
    Tags:
      ApplicationCI: abc
      Environment: dev
      Region: us-east-2
      ual-cdk:version: 4.4.0
```

Aspects at United

```
const role = new Role(this, "MyTableReaderRole", {
    assumedBy: new AccountPrincipal(this.account),
});
```

- Policy-compliant IAM role
- All tags enforced
- Permissions boundary applied

```
MyTableReaderRole7F6B2E9A:
  Type: AWS::IAM::Role
  Properties:
    AssumeRolePolicyDocument:
      Statement:
        - Action: sts:AssumeRole
          Effect: Allow
          Principal:
            AWS:
              Fn::Join:
                - ""
                - - "arn:"
                  - Ref: AWS::Partition
                  - :iam::12345678910:root
      Version: "2012-10-17"
    PermissionsBoundary: arn:aws:iam::12345678910:policy/AppPolicy_A
    Tags:
      - Key: ApplicationCI
        Value: abc
      - Key: Environment
        Value: dev
      - Key: Region
        Value: us-east-2
      - Key: ual-cdk:version
        Value: 4.4.0
```

United-specific ECS reference architecture

- 34 lines of TypeScript = 451 lines of CloudFormation
- Private hosted zone, Amazon ECS cluster, VPC, and subnet lookups
- Built in instrumentation

```typescript
import { UalStack, UalStackProps } from "ual-cdk";
import { ecs } from "ual-cdk";
import { aws_ecs as awsEcs } from "aws-cdk-lib";
import { ecs_patterns as ecsPatterns } from "ual-cdk";
import { Construct } from "constructs";

export class UalCdkDocsStack extends UalStack {
    constructor(scope: Construct, id: string, props: UalStackProps) {
        super(scope, id, props);

        const cluster = new ecs.UalCluster(this, "DocsCluster", {
            useIdInClusterNameSSMParameterName: true,
        });

        const albfs = new ecsPatterns.UalApplicationLoadBalancedFargateService(
            this,
            "ualcdk",
            {
                certificateArn:
                    this.region == "us-east-1"
                        ? "arn:aws:acm:us-east-1:12345678910:certificate/*****"
                        : "arn:aws:acm:us-east-2:12345678910:certificate/*****",
                cluster: cluster,
                containerPort: 80,
                taskImageOptions: {
                    image: awsEcs.ContainerImage.fromAsset("../"),
                    containerPort: 80,
                    enableLogging: true,
                },
                circuitBreaker: { rollback: true },
            }
        );
    }
}
```

Happy surprises

- No more hardcoded VPC IDs
- Amazon ECS workloads on AWS Fargate Spot in dev
- Change management
- Auto-generated documentation

United CDK apps look up their own networks

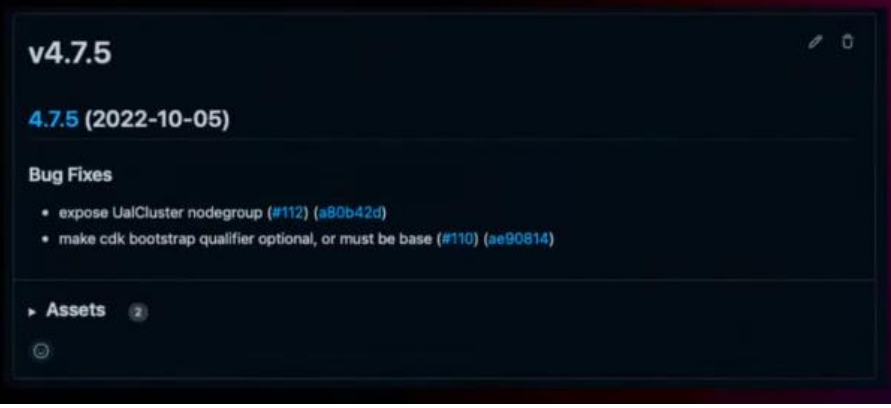
```typescript
// Load vpc by id or default logic
if (props && props.vpcId) {
  this.vpc = ec2.Vpc.fromLookup(this, `${id}Vpc`, {
    vpcId: props.vpcId,
    subnetGroupNameTag: 'Name',
    isDefault: props.isDefault != null ? props.isDefault : false,
  });
} else {
  if (props?.isDefault) {
    console.warn(
      'isDefault defaults to false if vpcId is not provided to UalNetwork construct'
    );
  }
  this.vpc = ec2.Vpc.fromLookup(this, `${id}Vpc`, {
    isDefault: false,
    vpcName: `VPC-${ualEnvironment.toUpperCase()}-${stack.region}`,
    subnetGroupNameTag: 'Name',
  });
}
```

```typescript
let ualNetwork: UalNetwork;
if (!props.vpcId) {
  ualNetwork = new UalNetwork(scope, `${id}UalNetwork`, {
    vpcId: props.vpcId,
  });
} else {
  ualNetwork = stack.getUalNetwork();
}
generated.vpcSubnets = {
  subnets: ualNetwork.subnets.appTier,
};
```

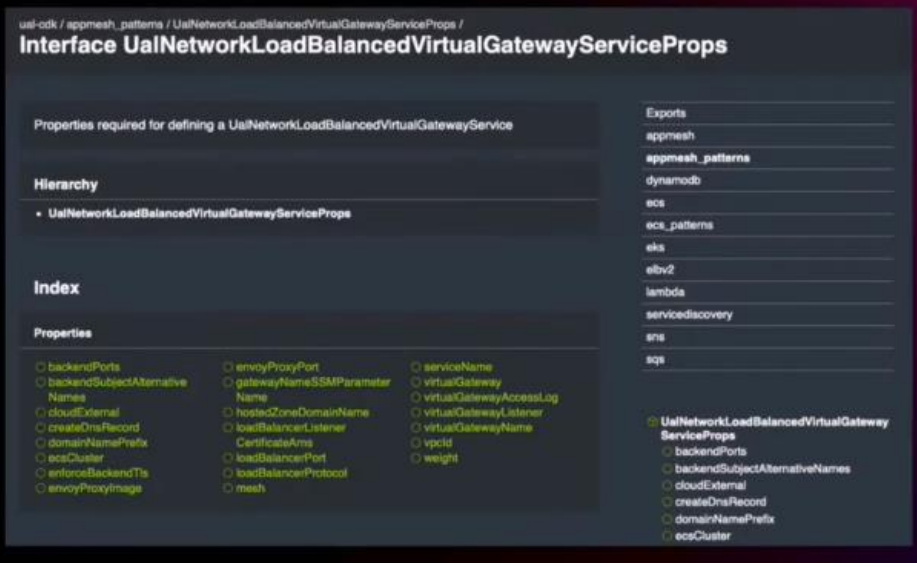
Fargate applications use Fargate Spot

```
generated.capacityProviderStrategies = [
  {
    capacityProvider:
      stack.ualEnvironment == UalEnvironment.DEV
        ? 'FARGATE_SPOT'
        : 'FARGATE',
    weight: 100,
  },
];
```

CDK construct libraries have versions and releases



United's CDK construct library auto-generates documentation



It gets more fun

We can use the JSSI tool to generate CDK code in other languages

This . . .

```typescript
export class MyTableStack extends UalStack {
    constructor(scope: Construct, id: string, props: UalStackProps) {
        super(scope, id, props);

        const role = new Role(this, "MyTableReaderRole", {
            assumedBy: new AccountPrincipal(this.account),
        });

        const table = new UalTable(this, "MyTable", {
            partitionKey: {
                name: "pk",
                type: AttributeType.STRING,
            },
        });

        table.grantReadData(role);
    }
}
```

TS

. . . is the same as this . . .

```python
class MyTableStack(UalStack):

    def __init__(self, scope: Construct, construct_id: str, **kwargs) -> None:
        super().__init__(scope, construct_id, **kwargs)

        role = iam.Role(self, "MyTableReaderRole",
                        iam.AccountPrincipal(self.account))

        table = dynamodb.UalTable(self, "MyTable", partition_key=aws_dynamodb.Attribute(
            name="pk", type=aws_dynamodb.AttributeType.STRING))

        table.grant_read_data(role)
```

. . . is the same as this . . .

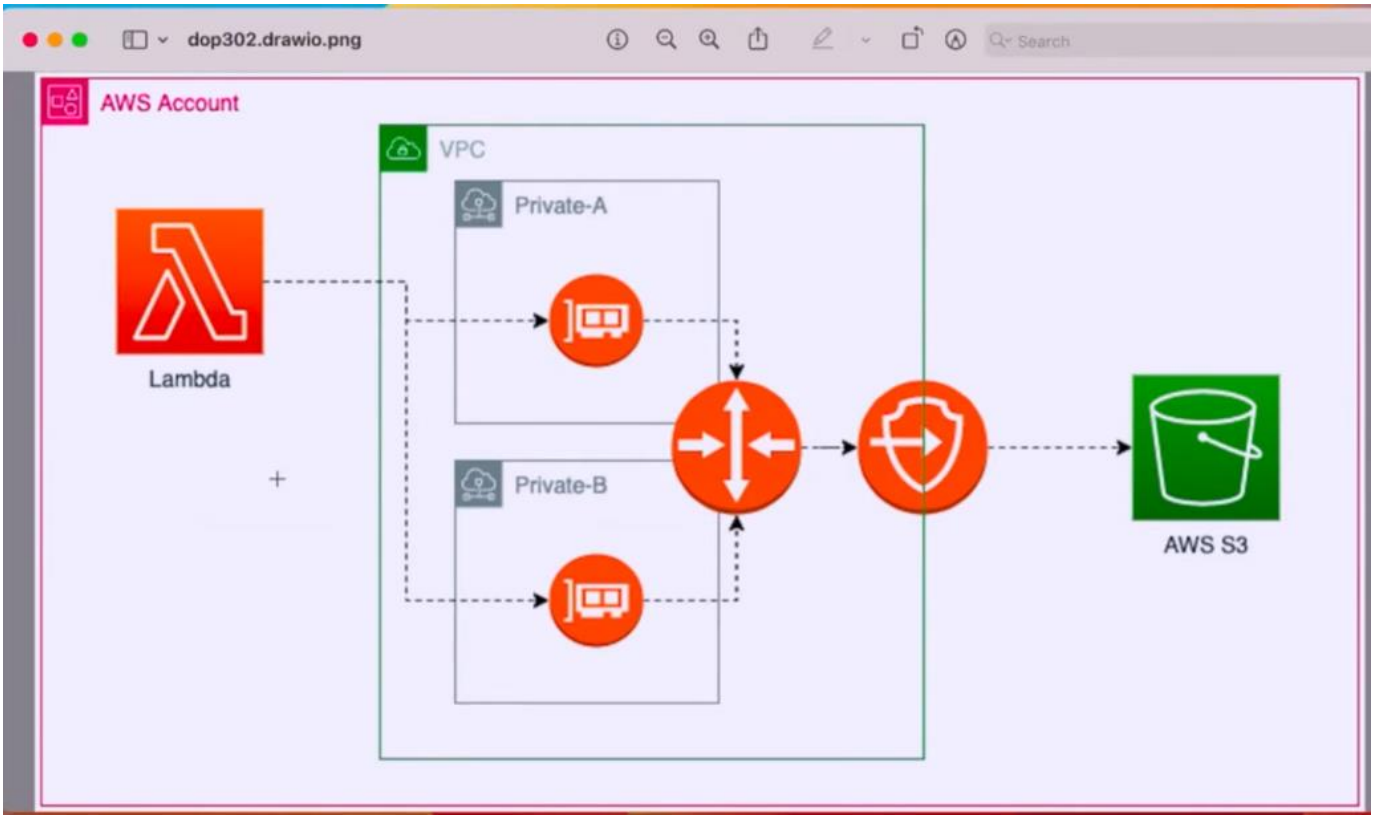
```java
public class MyTableStack extends UalStack {

    public MyTableStack(final Construct scope, final String id, final UalStackProps props) {
        super(scope, id, props);

        final Role role = new Role(this, id: "MyTableReaderRole",
                RoleProps.builder().assumedBy(new AccountPrincipal(this.getAccount())).build());

        final UalTable table = new UalTable(this, "MyTable",
                UalTableProps.builder()
                        .partitionKey(Attribute.builder().name(name: "pk").type(AttributeType.STRING).build()).build());

        table.grantReadData(role);
    }
}
```

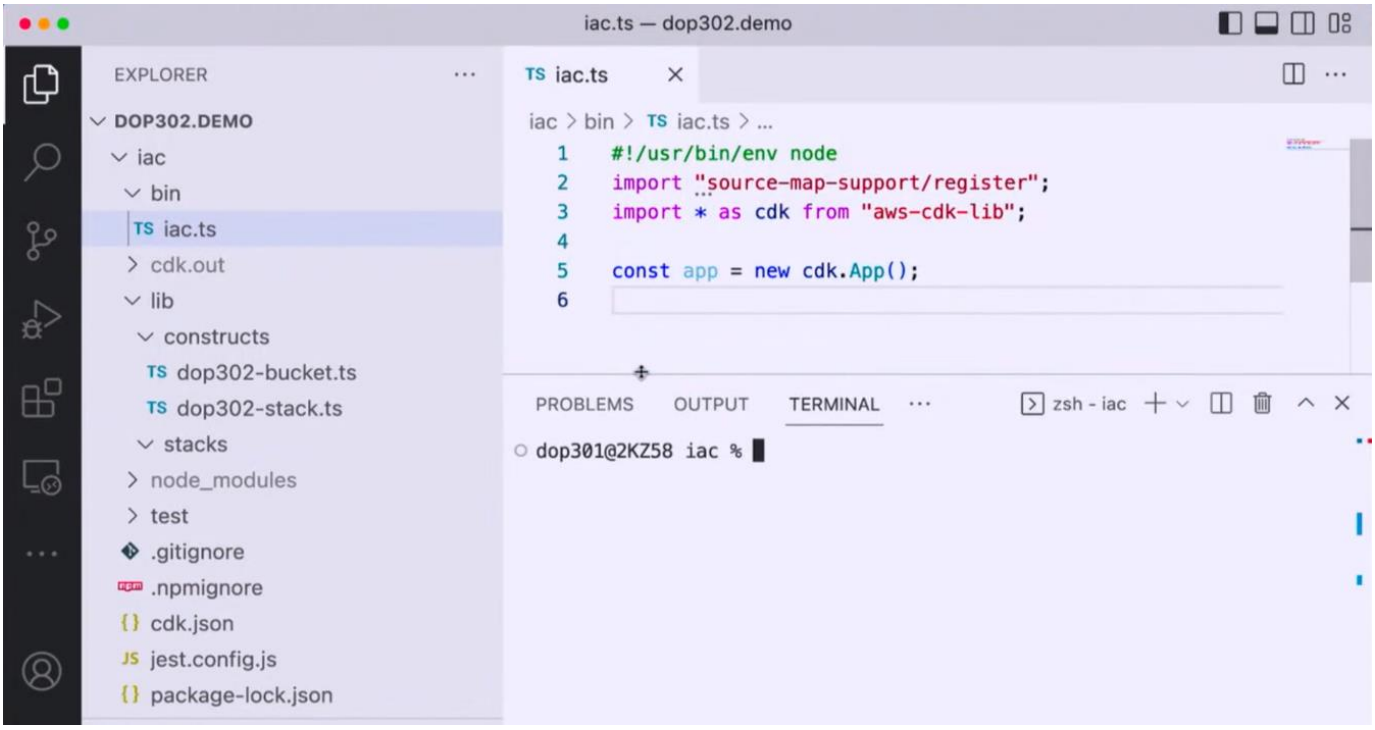
Java

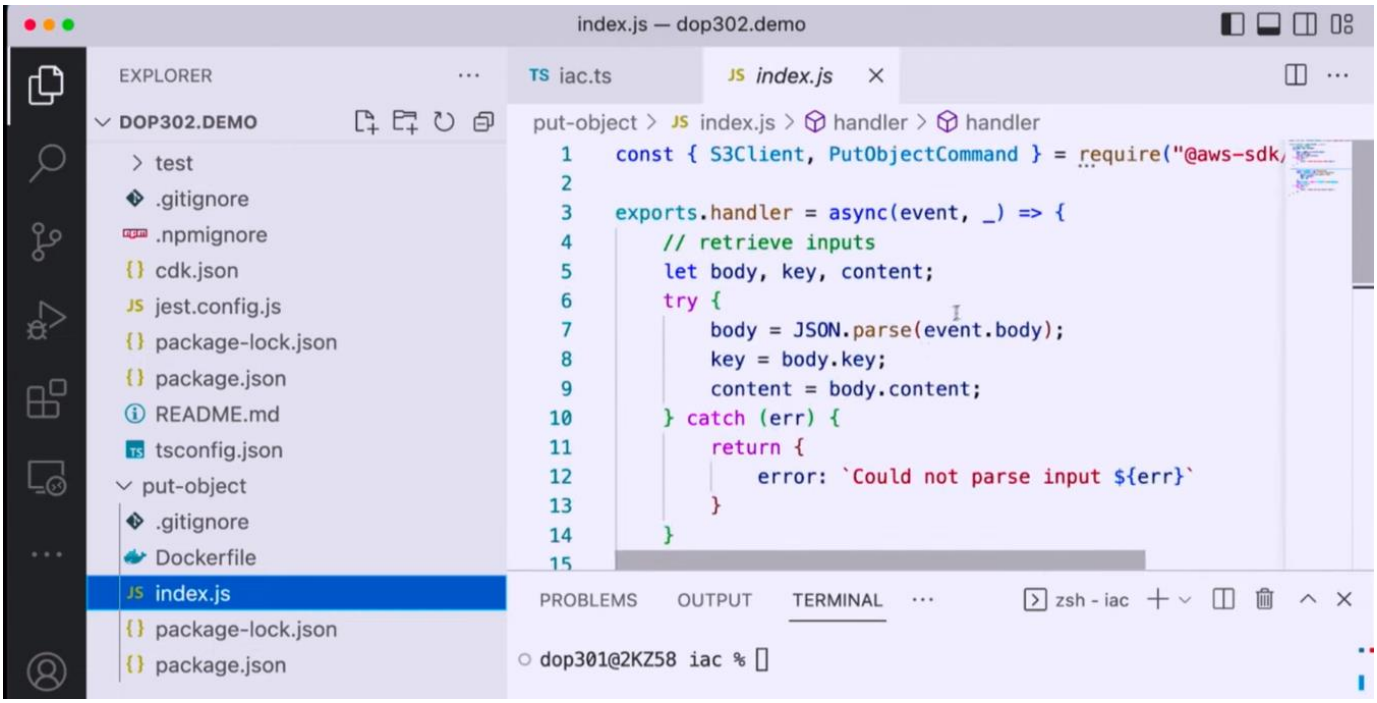
. . . is the same as this

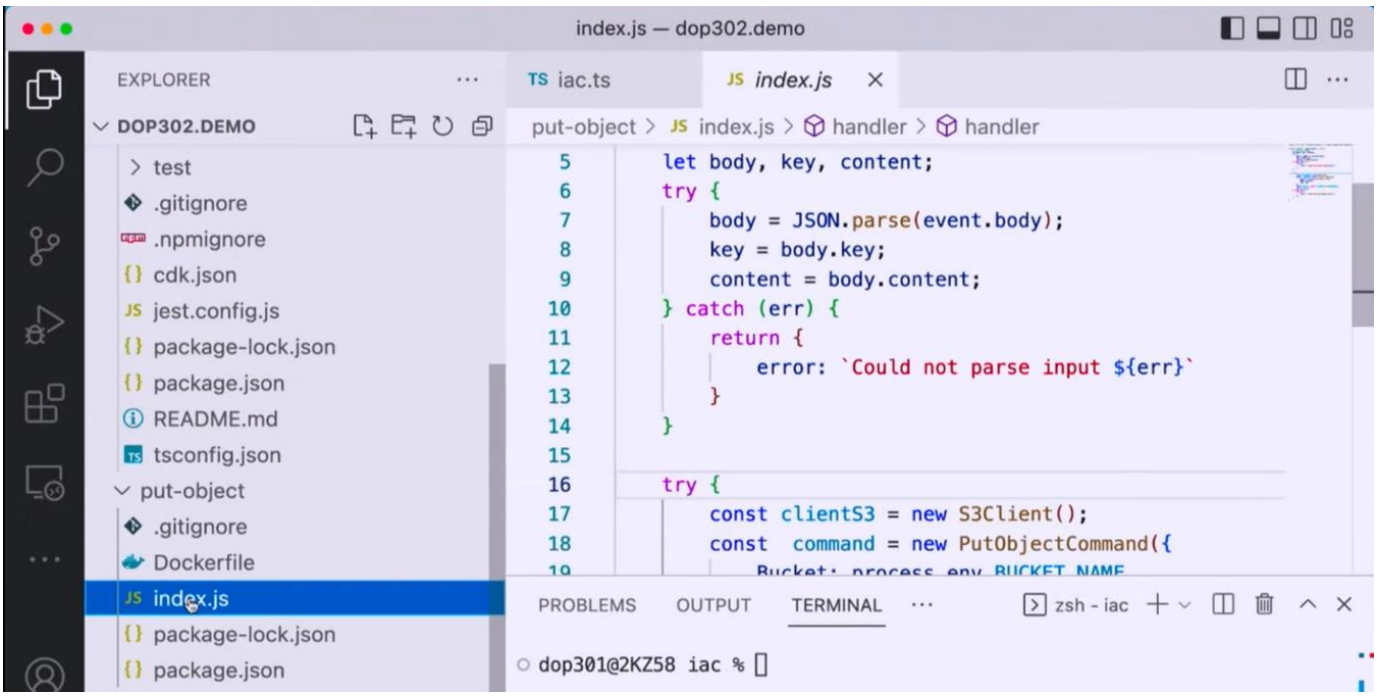
```csharp
public class MyTableStack : UalStack
{
    0 references
    internal MyTableStack(Construct scope, string id, IUalStackProps props) : base(scope, id, props)
    {
        var role = new Role(this, "MyTableReaderRole", new RoleProps
        {
            AssumedBy = new AccountPrincipal(this.Account)
        });

        var table = new UalTable(this, "MyTable", new UalTableProps
        {
            PartitionKey = new Attribute
            {
                Name = "pk",
                Type = AttributeType.STRING
            }
        });

        table.GrantReadData(role);
    }
}
```

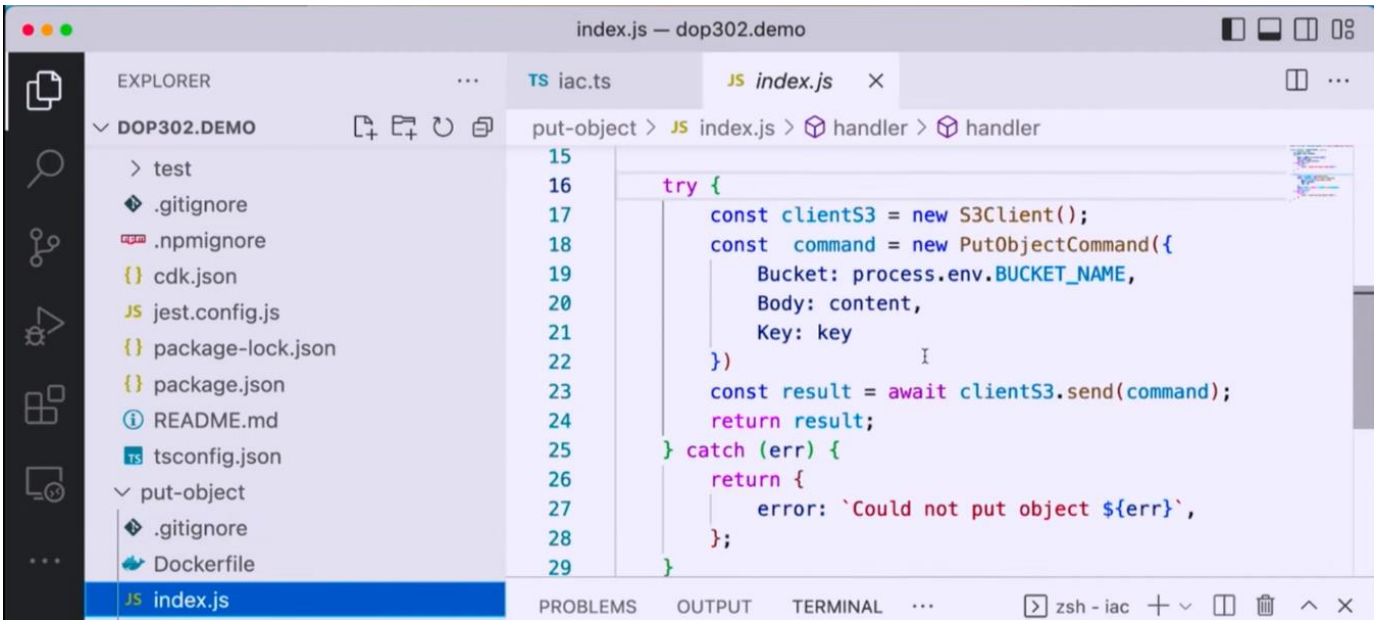
Demo

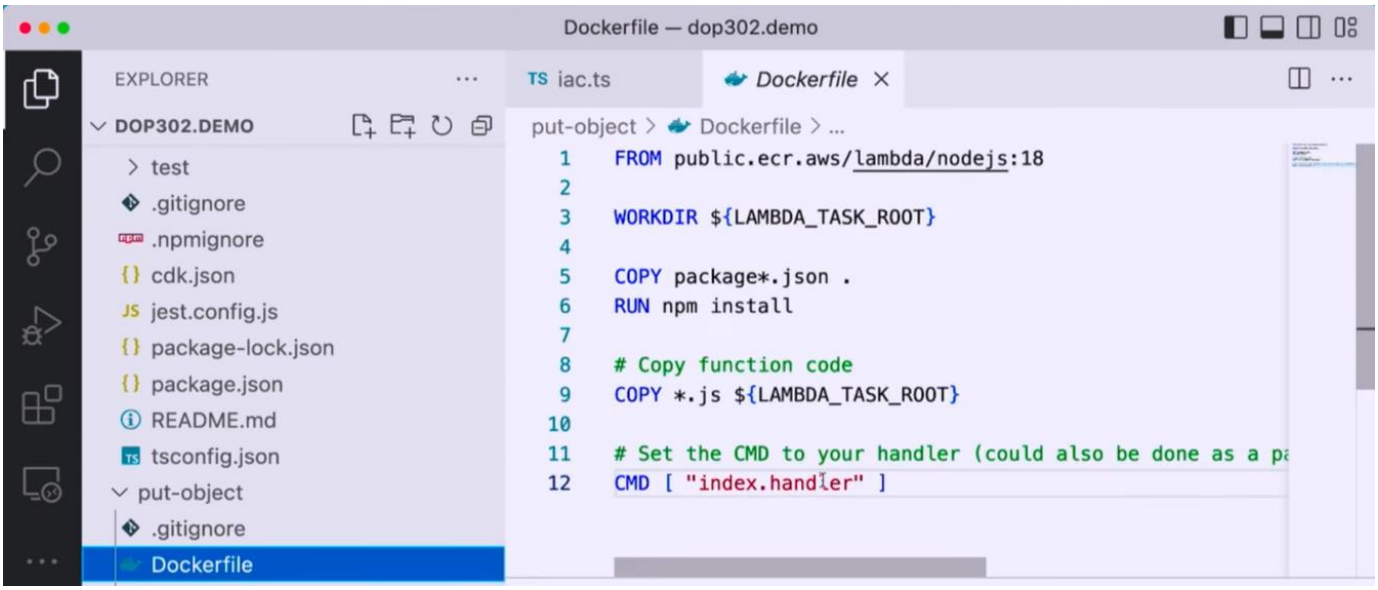

dop302.drawio.png
Search
AWS Account
VPC
Private-A
Private-B
Lambda
AWS S3

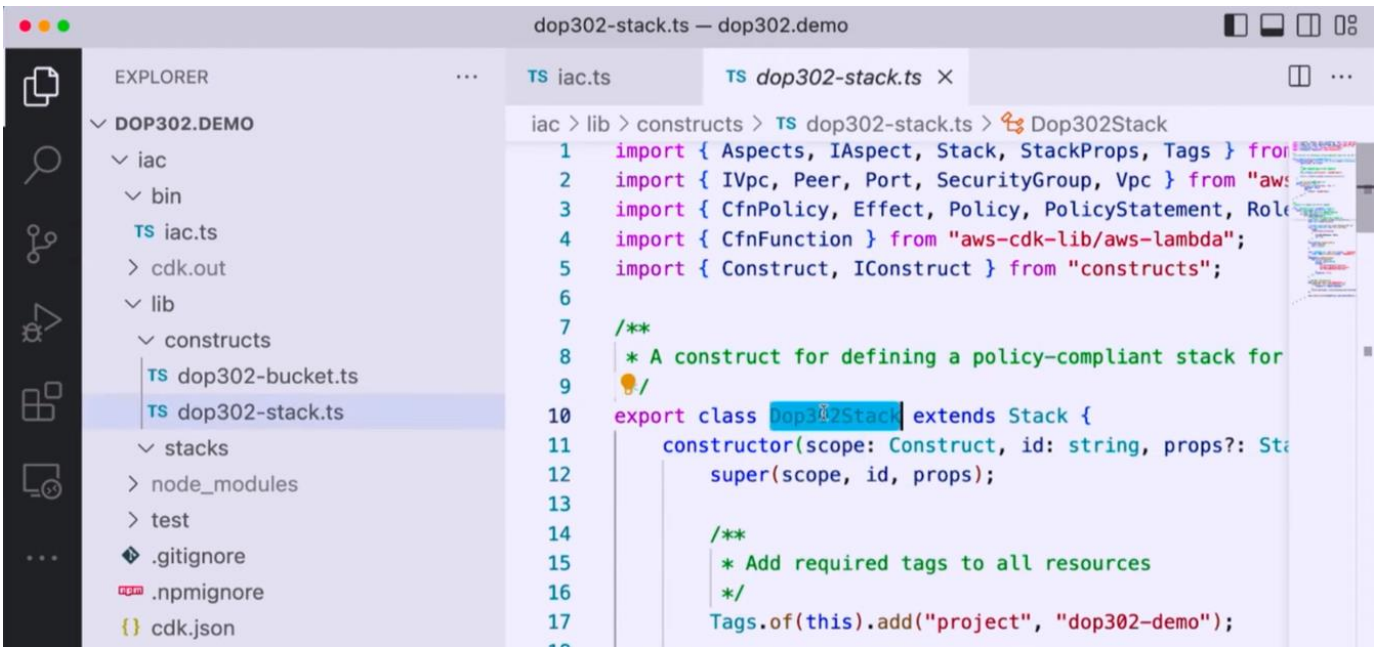

iac.ts — dop302.demo
EXPLORER
DOP302.DEMO
iac
bin
iac.ts
cdk.out
lib
constructs
dop302-bucket.ts
dop302-stack.ts
stacks
node_modules
test
.gitignore
.npmignore
cdk.json
jest.config.js
package-lock.json
iac.ts
iac > bin > iac.ts > ...
1 #!/usr/bin/env node
2 import "source-map-support/register";
3 import * as cdk from "aws-cdk-lib";
4
5 const app = new cdk.App();
6
PROBLEMS
OUTPUT
TERMINAL
zsh - iac
dop301@2KZ58 iac %

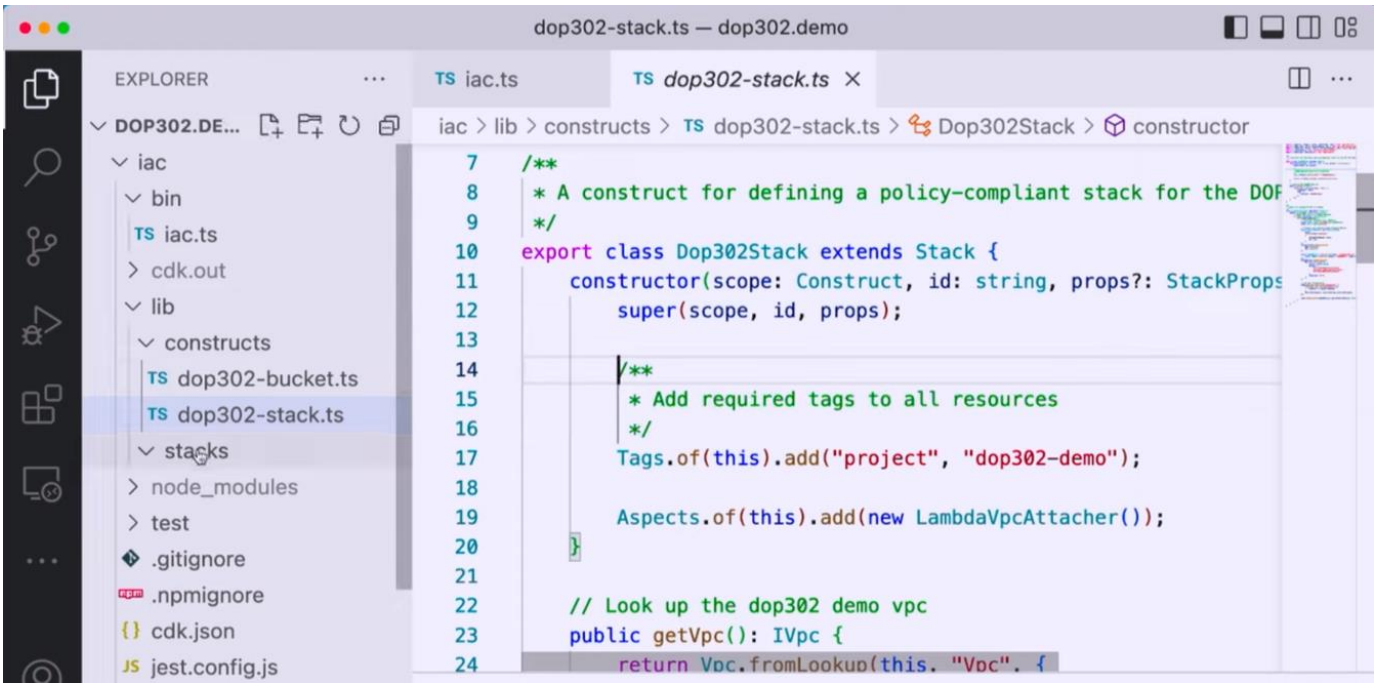

index.js — dop302.demo
EXPLORER
DOP302.DEMO
test
.gitignore
.npmignore
cdk.json
jest.config.js
package-lock.json
package.json
README.md
tsconfig.json
put-object
.gitignore
Dockerfile
index.js
package-lock.json
package.json
iac.ts
index.js
put-object > index.js > handler > handler
const { S3Client, PutObjectCommand } = require("@aws-sdk/
exports.handler = async(event, _) => {
// retrieve inputs
let body, key, content;
try {
body = JSON.parse(event.body);
key = body.key;
content = body.content;
} catch (err) {
return {
error: `Could not parse input ${err}`
}
}
PROBLEMS
OUTPUT
TERMINAL
zsh - iac
dop301@2KZ58 iac %

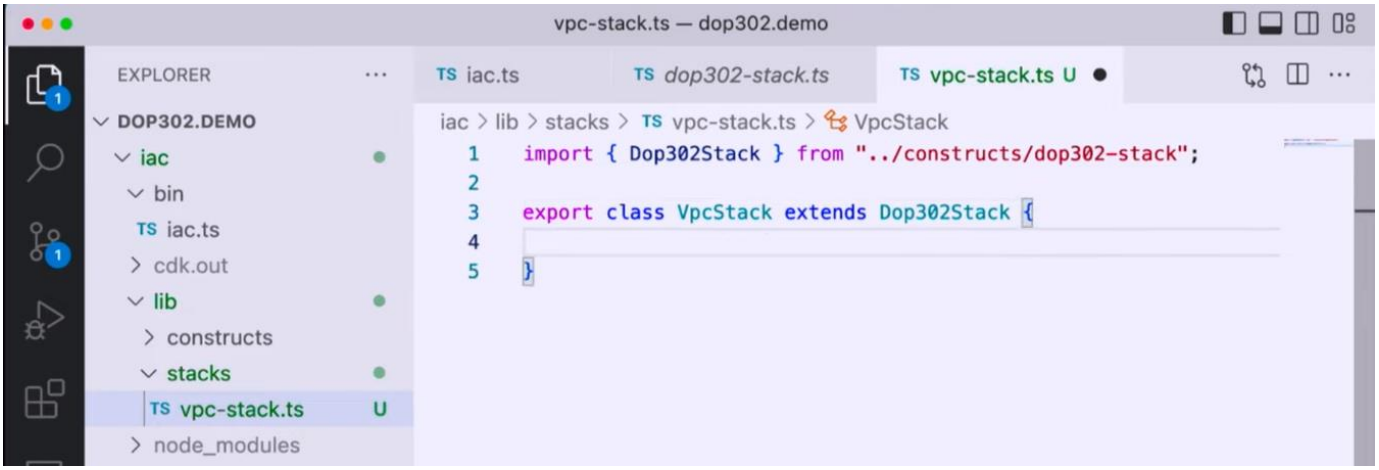

index.js — dop302.demo
EXPLORER
DOP302.DEMO
test
.gitignore
.npmignore
cdk.json
jest.config.js
package-lock.json
package.json
README.md
tsconfig.json
put-object
.gitignore
Dockerfile
index.js
package-lock.json
package.json
iac.ts
index.js
put-object > index.js > handler > handler
let body, key, content;
try {
body = JSON.parse(event.body);
key = body.key;
content = body.content;
} catch (err) {
return {
error: `Could not parse input ${err}`
}
}
try {
const clientS3 = new S3Client();
const command = new PutObjectCommand({
PROBLEMS
OUTPUT
TERMINAL
zsh - iac
dop301@2KZ58 iac %

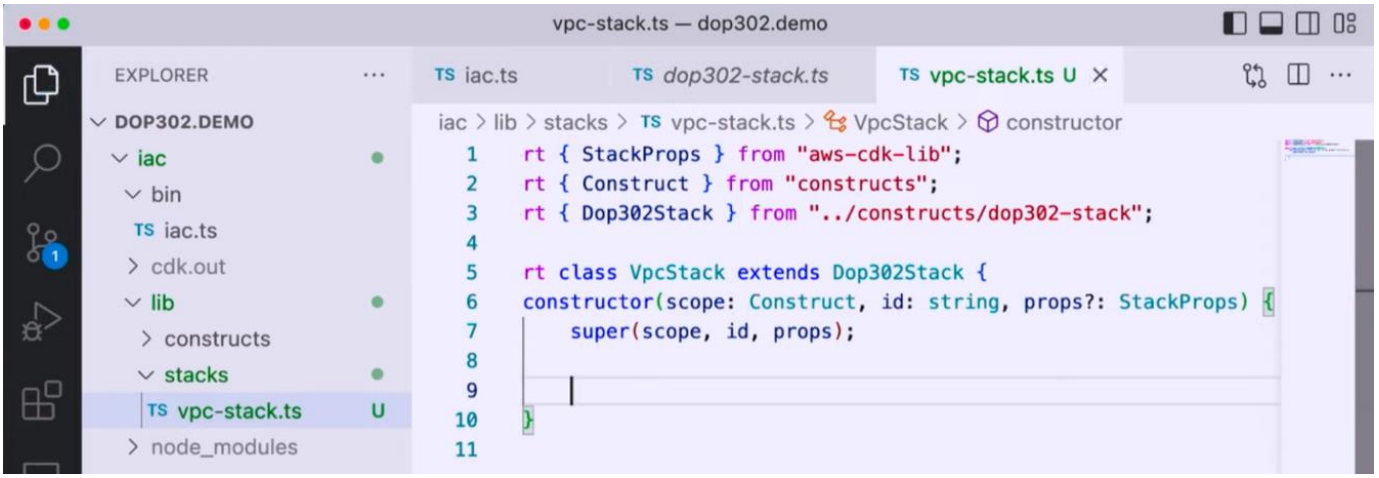

index.js — dop302.demo
EXPLORER
DOP302.DEMO
test
.gitignore
.npmignore
cdk.json
jest.config.js
package-lock.json
package.json
README.md
tsconfig.json
put-object
.gitignore
Dockerfile
index.js
iac.ts
index.js
put-object > index.js > handler > handler
try {
const clientS3 = new S3Client();
const command = new PutObjectCommand({
Bucket: process.env.BUCKET_NAME,
Body: content,
Key: key
})
const result = await clientS3.send(command);
return result;
} catch (err) {
return {
error: `Could not put object ${err}`,
};
}
PROBLEMS
OUTPUT
TERMINAL
zsh - iac

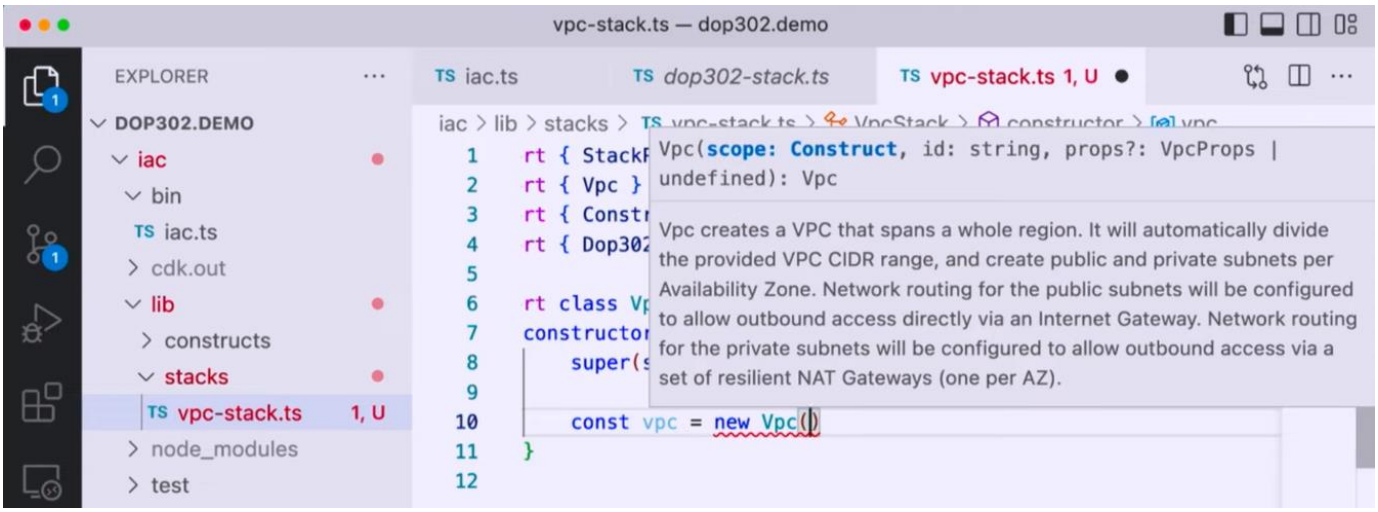

Dockerfile — dop302.demo
EXPLORER
DOP302.DEMO
test
.gitignore
.npmignore
cdk.json
jest.config.js
package-lock.json
package.json
README.md
tsconfig.json
put-object
.gitignore
Dockerfile
iac.ts
Dockerfile
put-object > Dockerfile > ...
FROM public.ecr.aws/lambda/nodejs:18
WORKDIR ${LAMBDA_TASK_ROOT}
COPY package*.json .
RUN npm install
Copy function code
COPY *.js ${LAMBDA_TASK_ROOT}
Set the CMD to your handler (could also be done as a pa
CMD ["index.handler"]


dop302-stack.ts — dop302.demo
EXPLORER
DOP302.DEMO
iac
bin
iac.ts
cdk.out
lib
constructs
dop302-bucket.ts
dop302-stack.ts
stacks
node_modules
test
.gitignore
.npmignore
cdk.json
iac.ts
dop302-stack.ts
iac > lib > constructs > dop302-stack.ts > Dop302Stack
import { Aspects, IAspect, Stack, StackProps, Tags } fro
import { IVpc, Peer, Port, SecurityGroup, Vpc } from "aws
import { CfnPolicy, Effect, Policy, PolicyStatement, Role
import { CfnFunction } from "aws-cdk-lib/aws-lambda";
import { Construct, IConstruct } from "constructs";
/**
* A construct for defining a policy-compliant stack for
*/
export class Dop302Stack extends Stack {
constructor(scope: Construct, id: string, props?: Sta
super(scope, id, props);
/**
* Add required tags to all resources
*/
Tags.of(this).add("project", "dop302-demo");

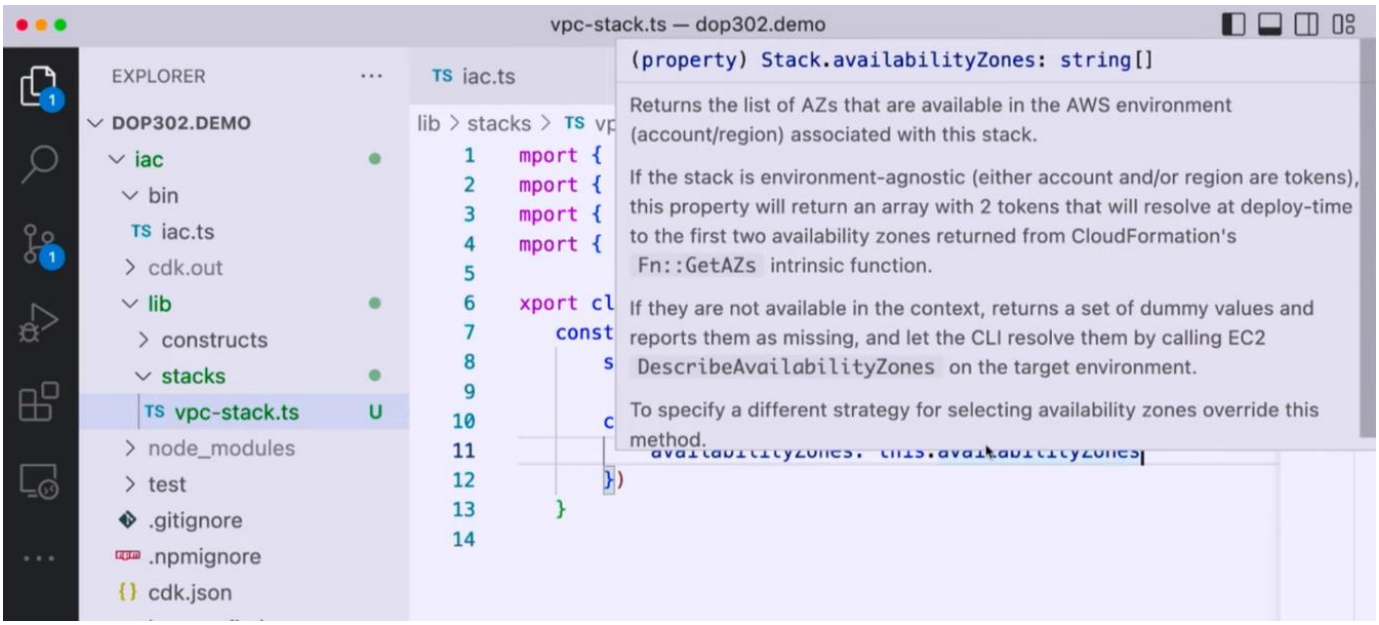

dop302-stack.ts — dop302.demo
EXPLORER
DOP302.DE...
iac
bin
iac.ts
cdk.out
lib
constructs
dop302-bucket.ts
dop302-stack.ts
stacks
node_modules
test
.gitignore
.npmignore
cdk.json
jest.config.js
iac.ts
dop302-stack.ts
iac > lib > constructs > dop302-stack.ts > Dop302Stack > constructor
/**
* A construct for defining a policy-compliant stack for the DOP
*/
export class Dop302Stack extends Stack {
constructor(scope: Construct, id: string, props?: StackProps
super(scope, id, props);
/**
* Add required tags to all resources
*/
Tags.of(this).add("project", "dop302-demo");
Aspects.of(this).add(new LambdaVpcAttacher());
}
// Look up the dop302 demo vpc
public getVpc(): IVpc {
return Vpc.fromLookup(this, "Vpc", {

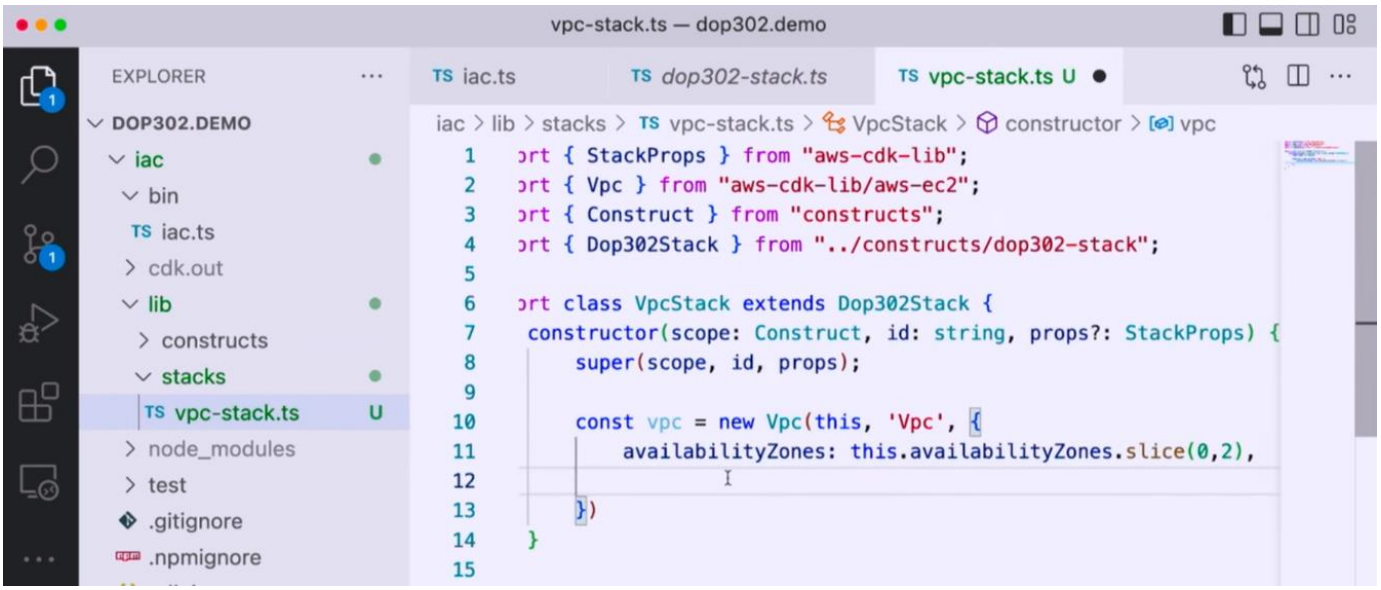

vpc-stack.ts — dop302.demo
EXPLORER
DOP302.DEMO
iac
bin
iac.ts
cdk.out
lib
constructs
stacks
vpc-stack.ts
U
node_modules
iac.ts
dop302-stack.ts
vpc-stack.ts U
iac > lib > stacks > vpc-stack.ts > VpcStack
import { Dop302Stack } from "../constructs/dop302-stack";
export class VpcStack extends Dop302Stack {
}


vpc-stack.ts — dop302.demo
EXPLORER
DOP302.DEMO
iac
bin
iac.ts
cdk.out
lib
constructs
stacks
vpc-stack.ts
U
node_modules
iac.ts
dop302-stack.ts
vpc-stack.ts U
iac > lib > stacks > vpc-stack.ts > VpcStack > constructor
rt { StackProps } from "aws-cdk-lib";
rt { Construct } from "constructs";
rt { Dop302Stack } from "../constructs/dop302-stack";
rt class VpcStack extends Dop302Stack {
constructor(scope: Construct, id: string, props?: StackProps) {
super(scope, id, props);
}

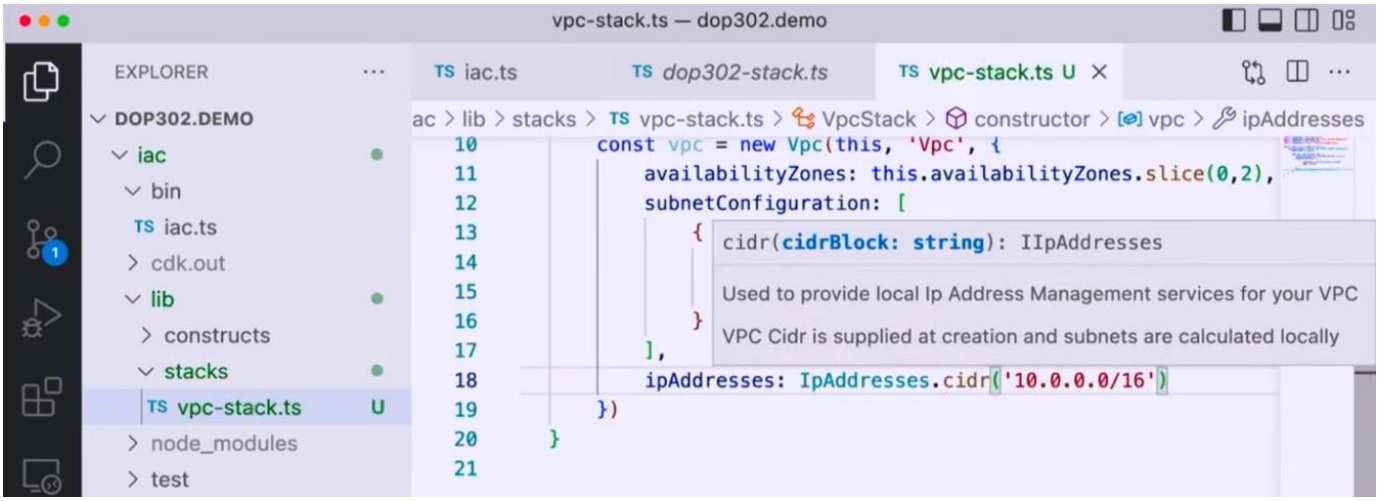

vpc-stack.ts — dop302.demo
EXPLORER
DOP302.DEMO
iac
bin
iac.ts
cdk.out
lib
constructs
stacks
vpc-stack.ts 1, U
node_modules
test
iac.ts
dop302-stack.ts
vpc-stack.ts 1, U
iac > lib > stacks > vpc-stack.ts > VpcStack > constructor > vpc
Vpc(scope: Construct, id: string, props?: VpcProps | undefined): Vpc
Vpc creates a VPC that spans a whole region. It will automatically divide the provided VPC CIDR range, and create public and private subnets per Availability Zone. Network routing for the public subnets will be configured to allow outbound access directly via an Internet Gateway. Network routing for the private subnets will be configured to allow outbound access via a set of resilient NAT Gateways (one per AZ).
const vpc = new Vpc()

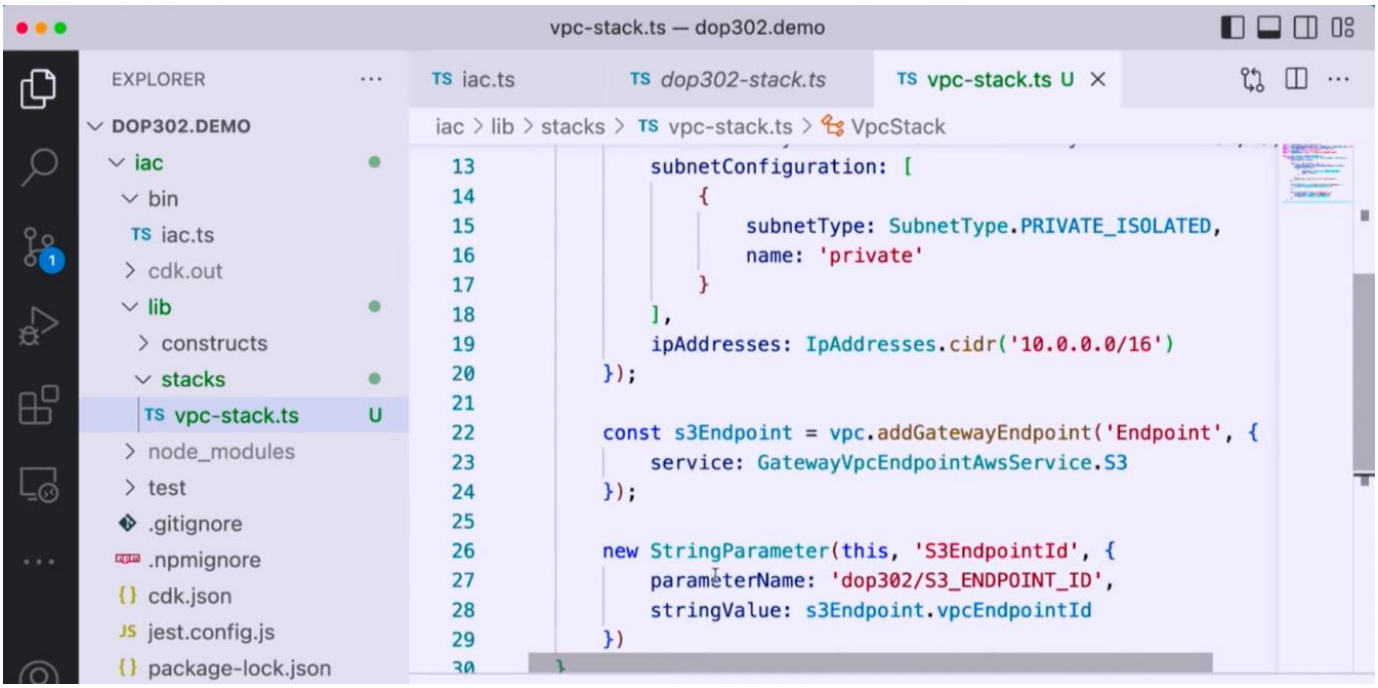

vpc-stack.ts — dop302.demo
EXPLORER
DOP302.DEMO
iac
bin
iac.ts
cdk.out
lib
constructs
stacks
vpc-stack.ts 1, U
node_modules
test
.gitignore
.npmignore
iac.ts
dop302-stack.ts
vpc-stack.ts 1, U
iac > lib > stacks
(property) VpcProps.availabilityZones?: string[] | undefined
Availability zones this VPC spans.
Specify this option only if you do not specify maxAzs .
@default
- a subset of AZs of the stack
Cannot find name 'availabilityZones'. Did you mean the instance member 'this.availabilityZones'? ts(2663)
View Problem (⌥F8) Quick Fix... (⌘.)
availabilityZones

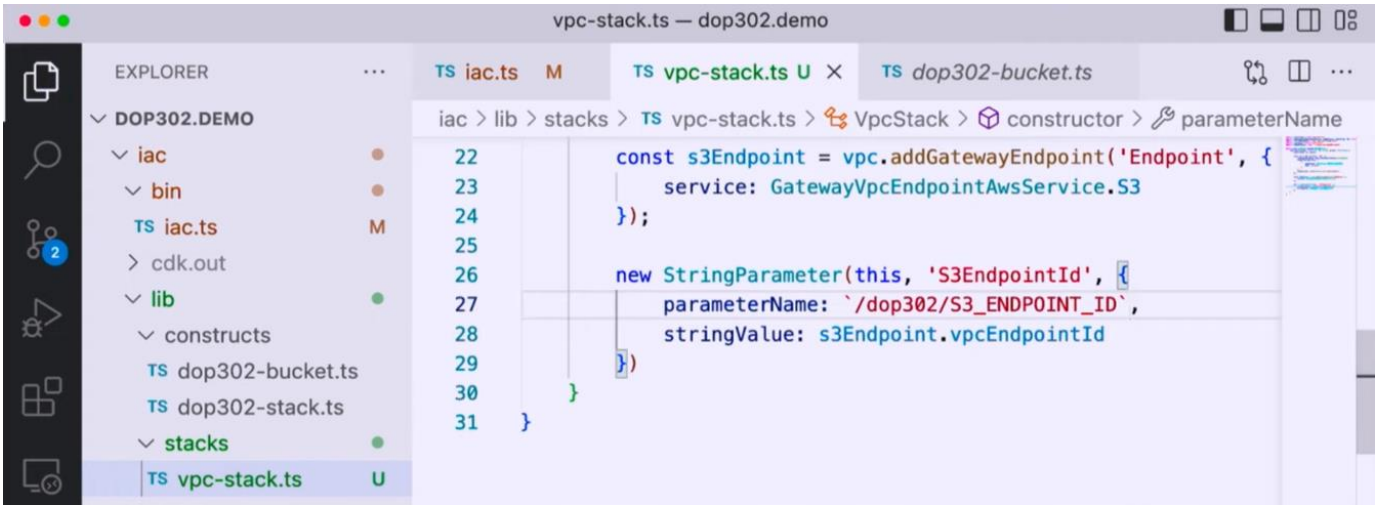

vpc-stack.ts — dop302.demo
EXPLORER
DOP302.DEMO
iac
bin
iac.ts
cdk.out
lib
constructs
stacks
vpc-stack.ts U
node_modules
test
.gitignore
.npmignore
cdk.json
iac.ts
lib > stacks > vp
(property) Stack.availabilityZones: string[]
Returns the list of AZs that are available in the AWS environment (account/region) associated with this stack.
If the stack is environment-agnostic (either account and/or region are tokens), this property will return an array with 2 tokens that will resolve at deploy-time to the first two availability zones returned from CloudFormation's Fn::GetAZs intrinsic function.
If they are not available in the context, returns a set of dummy values and reports them as missing, and let the CLI resolve them by calling EC2 DescribeAvailabilityZones on the target environment.
To specify a different strategy for selecting availability zones override this method.

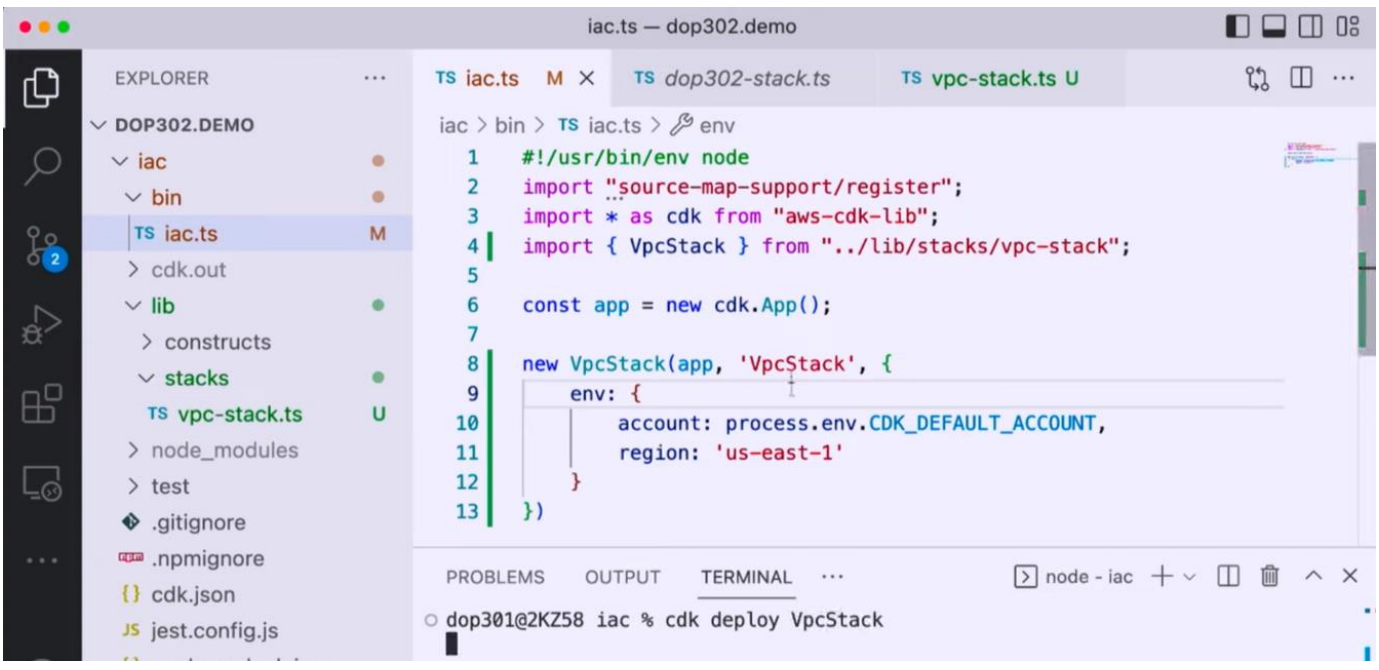

vpc-stack.ts — dop302.demo
EXPLORER
DOP302.DEMO
iac
bin
iac.ts
cdk.out
lib
constructs
stacks
vpc-stack.ts U
node_modules
test
.gitignore
.npmignore
iac.ts
dop302-stack.ts
vpc-stack.ts U
iac > lib > stacks > vpc-stack.ts > VpcStack > constructor > vpc
ort { StackProps } from "aws-cdk-lib";
ort { Vpc } from "aws-cdk-lib/aws-ec2";
ort { Construct } from "constructs";
ort { Dop302Stack } from "../constructs/dop302-stack";
ort class VpcStack extends Dop302Stack {
constructor(scope: Construct, id: string, props?: StackProps) {
super(scope, id, props);
const vpc = new Vpc(this, 'Vpc', {
availabilityZones: this.availabilityZones.slice(0,2),
})
}


vpc-stack.ts — dop302.demo
EXPLORER
DOP302.DEMO
iac
bin
iac.ts
cdk.out
lib
constructs
stacks
vpc-stack.ts U
node_modules
test
.gitignore
.npmignore
cdk.json
jest.config.js
iac.ts
dop302-stack.ts
vpc-stack.ts U
iac > lib > stacks > vpc-stack.ts > VpcStack > constructor > vpc
ort { StackProps } from "aws-cdk-lib";
ort { SubnetType, Vpc } from "aws-cdk-lib/aws-ec2";
ort { Construct } from "constructs";
ort { Dop302Stack } from "../constructs/dop302-stack";
ort class VpcStack extends Dop302Stack {
constructor(scope: Construct, id: string, props?: StackProps)
super(scope, id, props);
const vpc = new Vpc(this, 'Vpc', {
availabilityZones: this.availabilityZones.slice(0,2),
subnetConfiguration: [
{
subnetType: SubnetType.PRIVATE_ISOLATED,
name: 'private'
}
],


vpc-stack.ts — dop302.demo
EXPLORER
DOP302.DEMO
iac
bin
iac.ts
cdk.out
lib
constructs
stacks
vpc-stack.ts U
node_modules
test
iac.ts
dop302-stack.ts
vpc-stack.ts U
ac > lib > stacks > vpc-stack.ts > VpcStack > constructor > vpc > ipAddresses
const vpc = new Vpc(this, 'Vpc', {
availabilityZones: this.availabilityZones.slice(0,2),
subnetConfiguration: [
{
cidr(cidrBlock: string): IIpAddresses
Used to provide local Ip Address Management services for your VPC
VPC Cidr is supplied at creation and subnets are calculated locally
}
],
ipAddresses: IpAddresses.cidr('10.0.0.0/16')
})
}

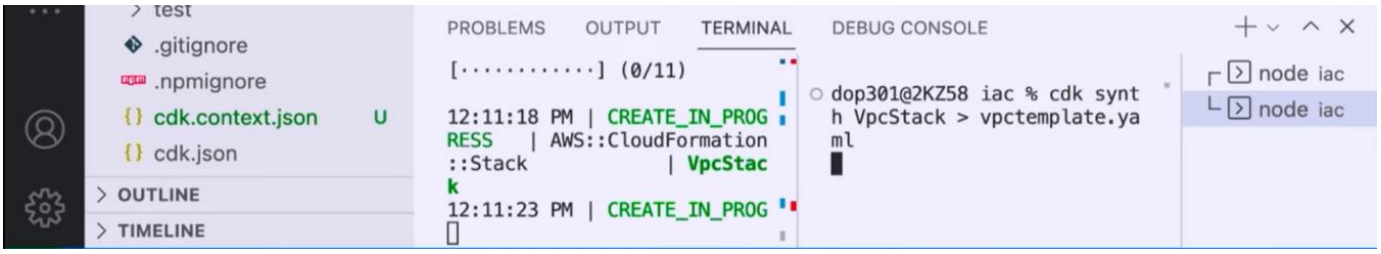

vpc-stack.ts — dop302.demo
EXPLORER
DOP302.DEMO
iac
bin
iac.ts
cdk.out
lib
constructs
stacks
vpc-stack.ts U
node_modules
test
.gitignore
.npmignore
cdk.json
jest.config.js
package-lock.json
iac.ts
dop302-stack.ts
vpc-stack.ts U
iac > lib > stacks > vpc-stack.ts > VpcStack
subnetConfiguration: [
{
subnetType: SubnetType.PRIVATE_ISOLATED,
name: 'private'
}
],
ipAddresses: IpAddresses.cidr('10.0.0.0/16')
});
const s3Endpoint = vpc.addGatewayEndpoint('Endpoint', {
service: GatewayVpcEndpointAwsService.S3
});
new StringParameter(this, 'S3EndpointId', {
parameterName: 'dop302/S3_ENDPOINT_ID',
stringValue: s3Endpoint.vpcEndpointId
})
}

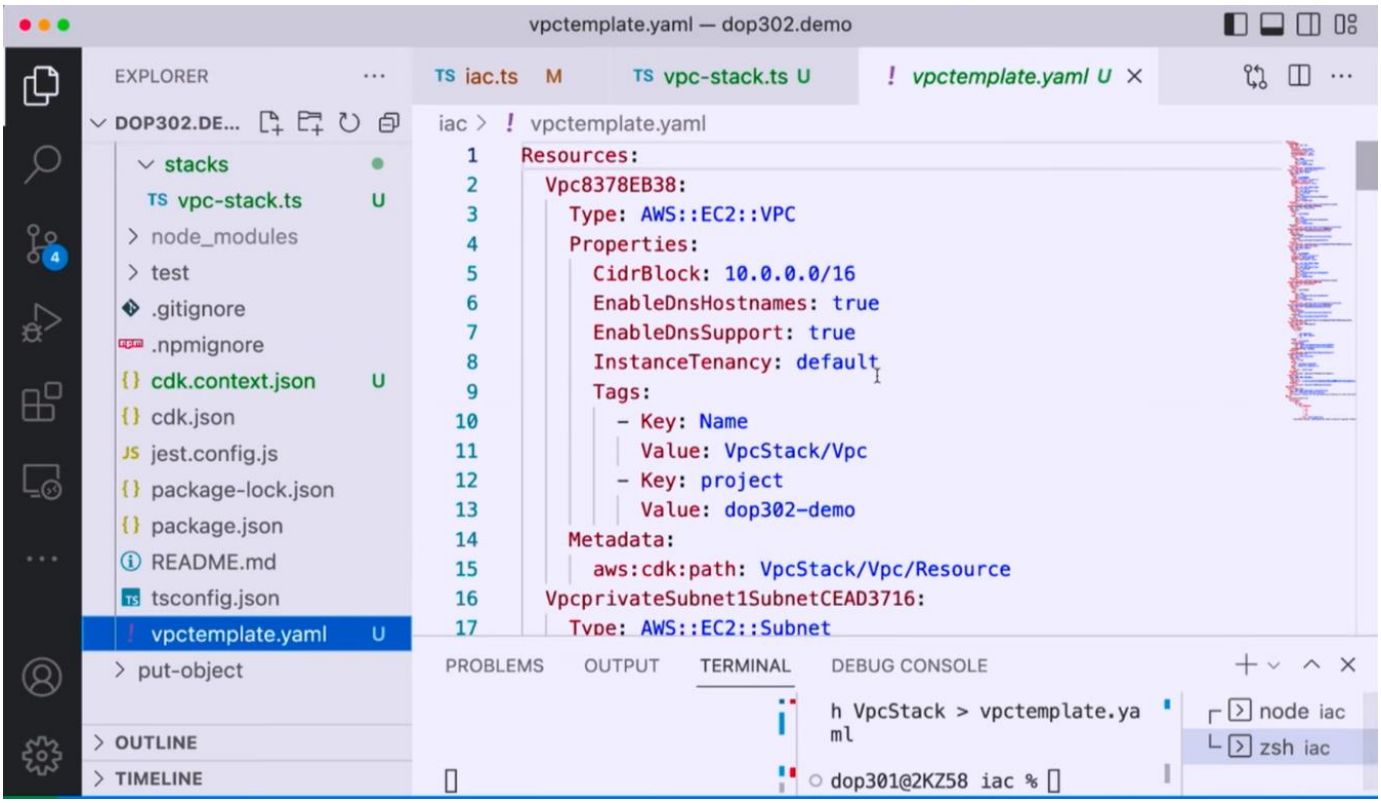

vpc-stack.ts — dop302.demo
EXPLORER
DOP302.DEMO
iac
bin
iac.ts
M
cdk.out
lib
constructs
dop302-bucket.ts
dop302-stack.ts
stacks
vpc-stack.ts U
iac.ts M
vpc-stack.ts U
dop302-bucket.ts
iac > lib > stacks > vpc-stack.ts > VpcStack > constructor > parameterName
const s3Endpoint = vpc.addGatewayEndpoint('Endpoint', {
service: GatewayVpcEndpointAwsService.S3
});
new StringParameter(this, 'S3EndpointId', {
parameterName: `/dop302/S3_ENDPOINT_ID`,
stringValue: s3Endpoint.vpcEndpointId
})
}
}

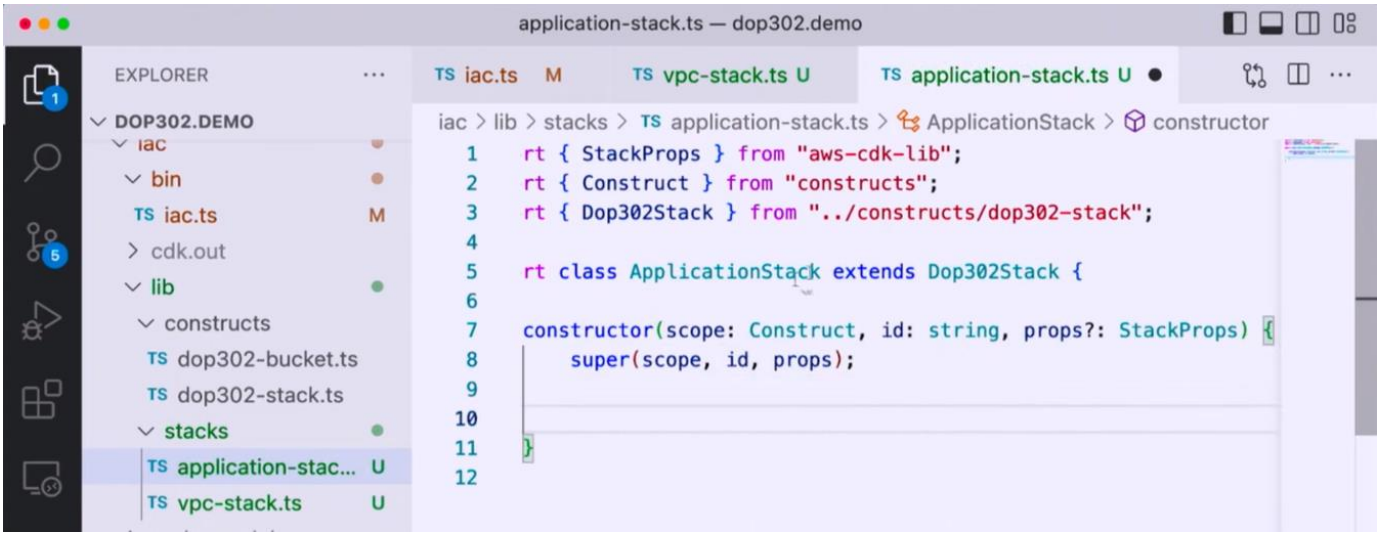

iac.ts — dop302.demo
EXPLORER
DOP302.DEMO
iac
bin
iac.ts
M
cdk.out
lib
constructs
stacks
vpc-stack.ts
U
node_modules
test
.gitignore
.npmignore
cdk.json
jest.config.js
iac.ts M
dop302-stack.ts
vpc-stack.ts U
iac > bin > iac.ts > env
#!/usr/bin/env node
import "source-map-support/register";
import * as cdk from "aws-cdk-lib";
import { VpcStack } from "../lib/stacks/vpc-stack";
const app = new cdk.App();
new VpcStack(app, 'VpcStack', {
env: {
account: process.env.CDK_DEFAULT_ACCOUNT,
region: 'us-east-1'
}
})
PROBLEMS
OUTPUT
TERMINAL
node - iac
dop301@2KZ58 iac % cdk deploy VpcStack


test
.gitignore
.npmignore
cdk.context.json U
cdk.json
OUTLINE
TIMELINE
PROBLEMS OUTPUT TERMINAL ...
node - iac
7eadf250:389171335348-us-east-1
VpcStack: assets built
VpcStack: deploying...
[0%] start: Publishing ee0e3163f688c46979cccd4f74e528d42d151ce560e7d88cda4bc23
37eadf250:389171335348-us-east-1

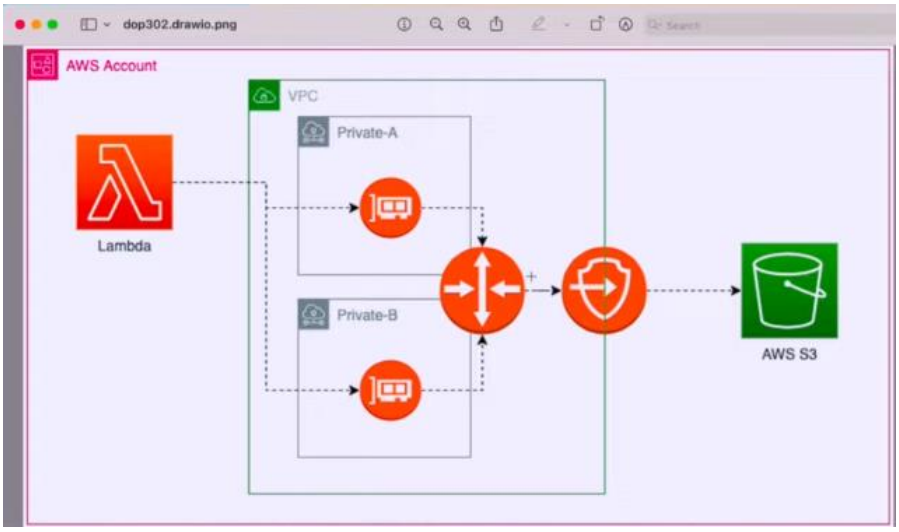

test
.gitignore
.npmignore
cdk.context.json U
cdk.json
OUTLINE
TIMELINE
PROBLEMS OUTPUT TERMINAL ...
node - iac
VpcStack: deploying...
[0%] start: Publishing ee0e3163f688c46979cccd4f74e528d42d151ce560e7d88cda4bc23
37eadf250:389171335348-us-east-1
[100%] success: Published ee0e3163f688c46979cccd4f74e528d42d151ce560e7d88cda4b
c2337eadf250:389171335348-us-east-1
VpcStack: creating CloudFormation changeset...

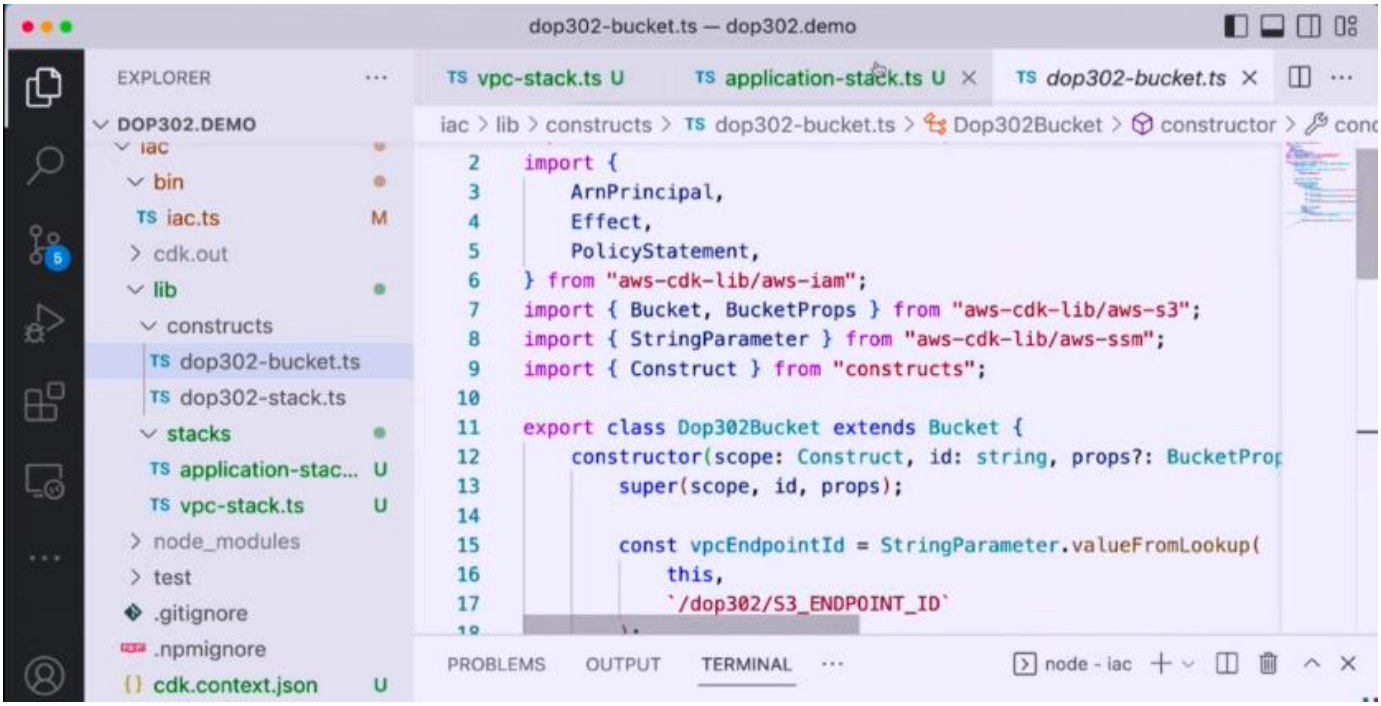

test
.gitignore
.npmignore
cdk.context.json U
cdk.json
OUTLINE
TIMELINE
PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE
[.....] (0/11)
12:11:18 PM | CREATE_IN_PROGRESS | AWS::CloudFormation::Stack | VpcStack
12:11:23 PM | CREATE_IN_PROG
dop301@2KZ58 iac % cdk synth VpcStack > vpctemplate.yaml
node iac
node iac


vpctemplate.yaml — dop302.demo
EXPLORER
DOP302.DE...
stacks
vpc-stack.ts U
node_modules
test
.gitignore
.npmignore
cdk.context.json U
cdk.json
jest.config.js
package-lock.json
package.json
README.md
tsconfig.json
vpctemplate.yaml U
put-object
OUTLINE
TIMELINE
iac.ts M
vpc-stack.ts U
vpctemplate.yaml U
iac > vpctemplate.yaml
Resources:
Vpc8378EB38:
Type: AWS::EC2::VPC
Properties:
CidrBlock: 10.0.0.0/16
EnableDnsHostnames: true
EnableDnsSupport: true
InstanceTenancy: default
Tags:
- Key: Name
Value: VpcStack/Vpc
- Key: project
Value: dop302-demo
Metadata:
aws:cdk:path: VpcStack/Vpc/Resource
VpcprivateSubnet1SubnetCEAD3716:
Type: AWS::EC2::Subnet
PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE
h VpcStack > vpctemplate.yaml
dop301@2KZ58 iac %
node iac
zsh iac



Next, let's build our stack to deploy our application into the created infrastructure we now have

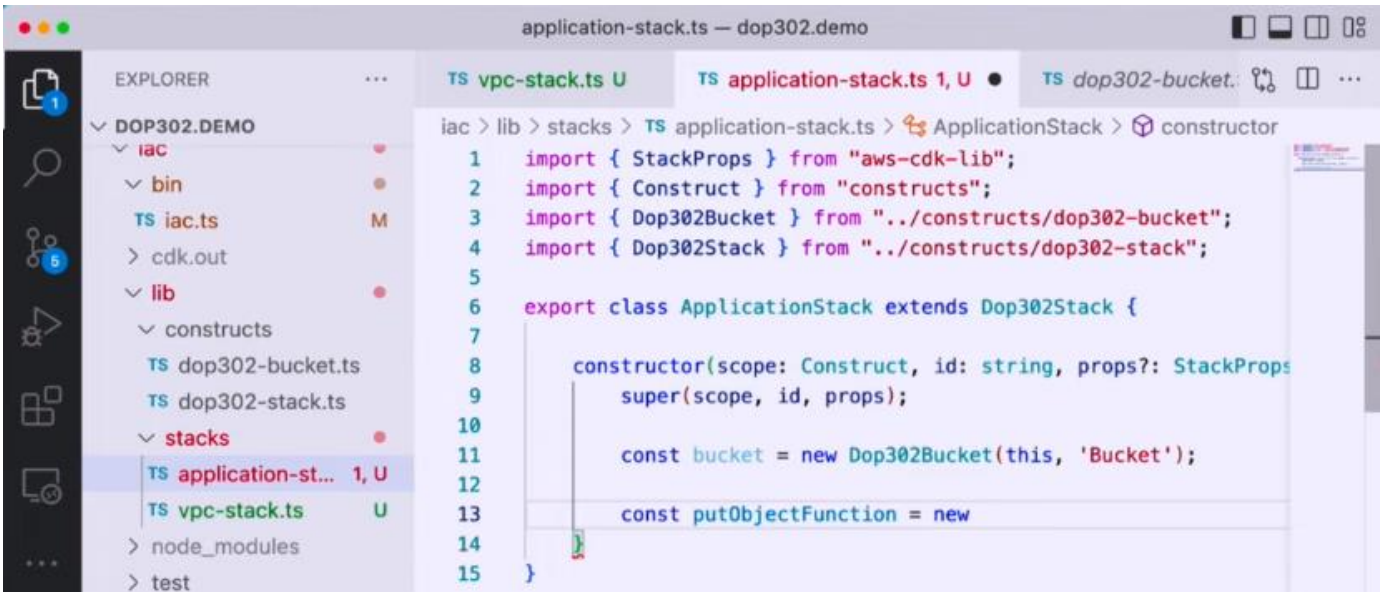


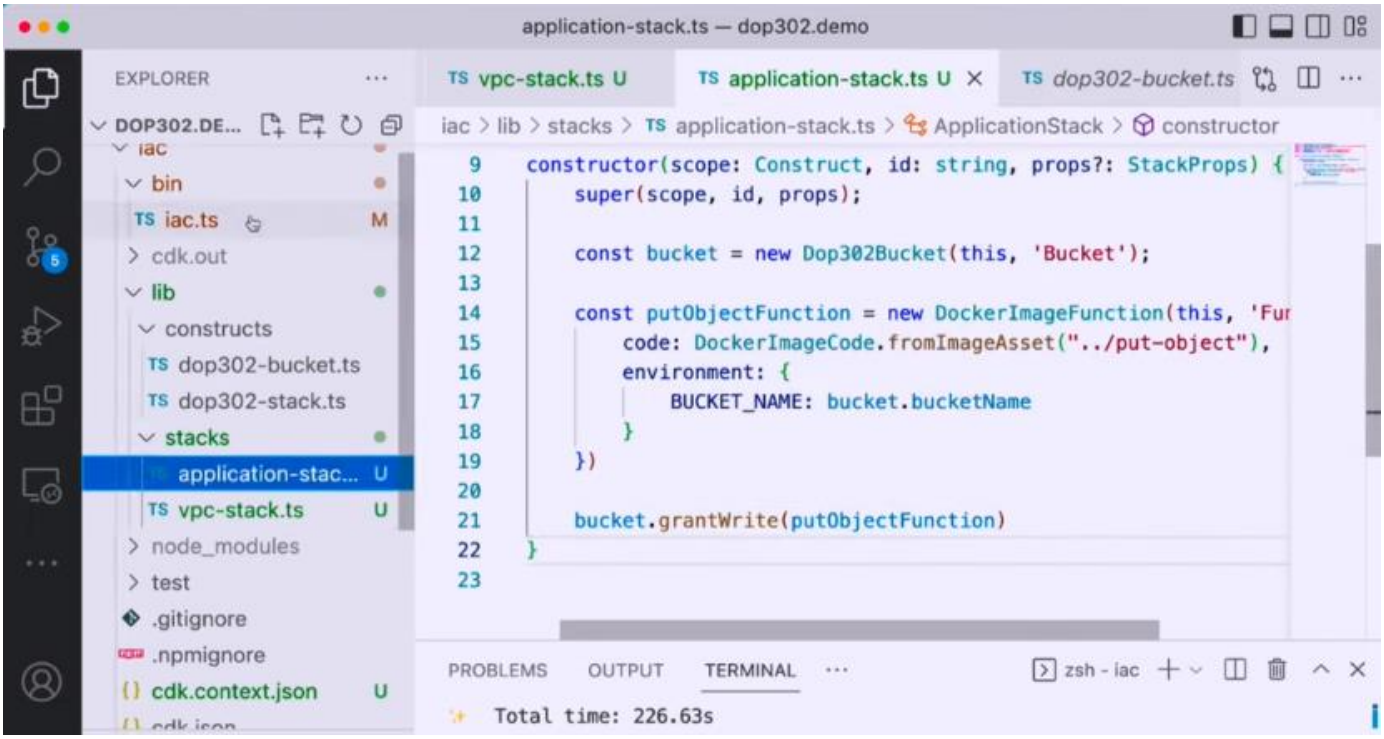


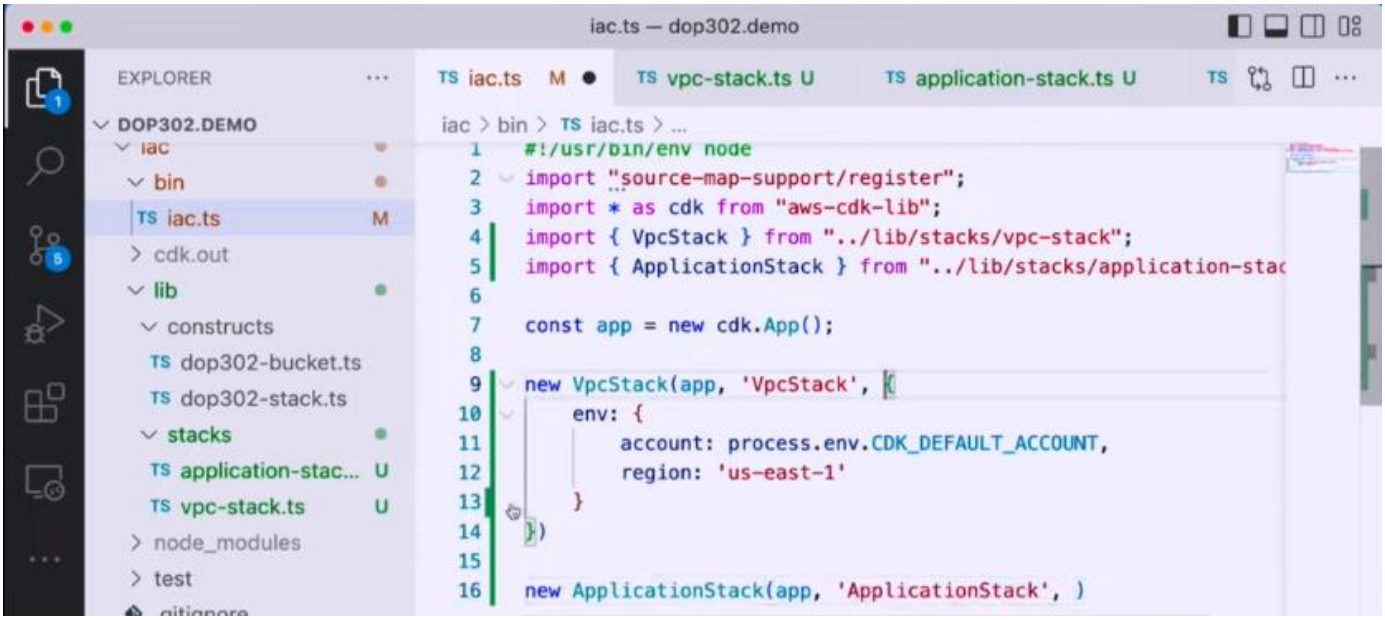
This is where we do bucket policy stuff to allow only traffic from our VPC to the application

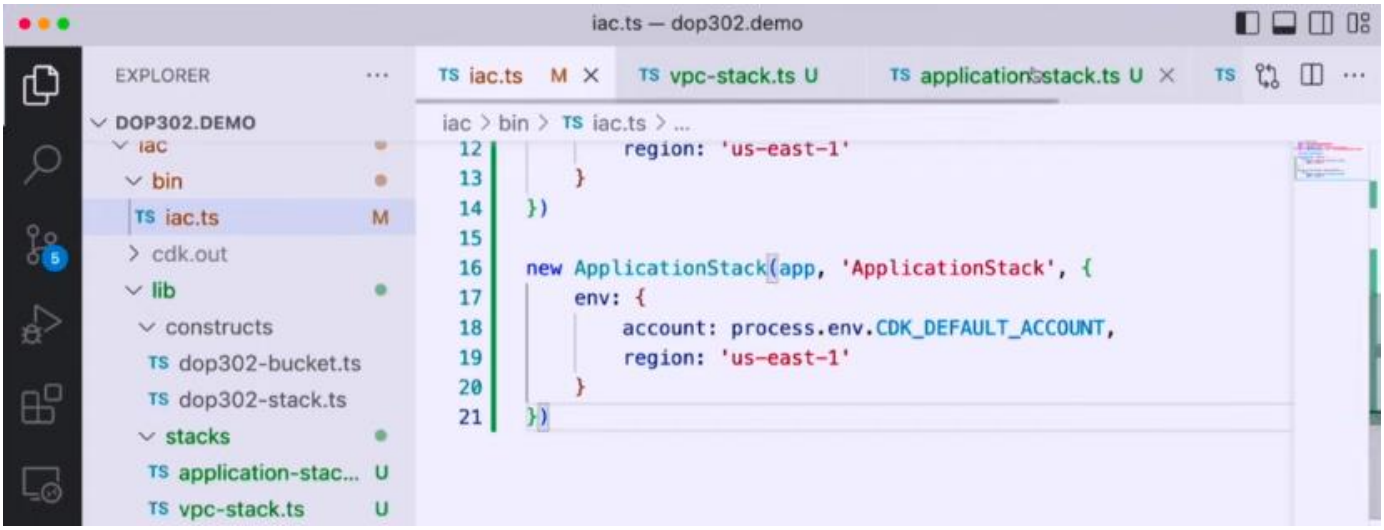

dop302-bucket.ts — dop302.demo
EXPLORER
DOP302.DEMO
iac
bin
iac.ts M
cdk.out
lib
constructs
dop302-bucket.ts
dop302-stack.ts
stacks
application-stac... U
vpc-stack.ts U
node_modules
test
.gitignore
vpc-stack.ts U
application-stack.ts U
dop302-bucket.ts
iac > lib > constructs > dop302-bucket.ts > Dop302Bucket > constructor
const stack = Stack.of(this);
this.addToResourcePolicy(
new PolicyStatement({
notPrincipals: [
new ArnPrincipal(
`arn:aws:iam::${stack.account}:user/etha
),
new ArnPrincipal(
`arn:aws:iam::${stack.account}:role/cdk-
),
new ArnPrincipal(
`arn:aws:sts::${stack.account}:assumed-r
),
],
effect: Effect.DENY,

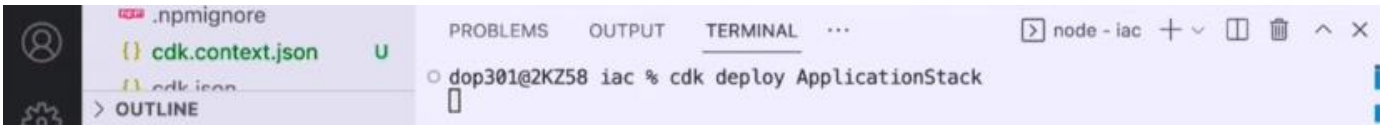

dop302-bucket.ts — dop302.demo
EXPLORER
DOP302.DEMO
iac
bin
iac.ts M
cdk.out
lib
constructs
dop302-bucket.ts
dop302-stack.ts
stacks
application-stac... U
vpc-stack.ts U
node_modules
test
vpc-stack.ts U
application-stack.ts U
dop302-bucket.ts
iac > lib > constructs > dop302-bucket.ts > Dop302Bucket > constructor
),
],
effect: Effect.DENY,
actions: ["*"],
conditions: {
StringNotEquals: {
"aws:SourceVpce": vpcEndpointId,
},
},
resources: [this.bucketArn, this.arnForObjects(
})
);
}
}

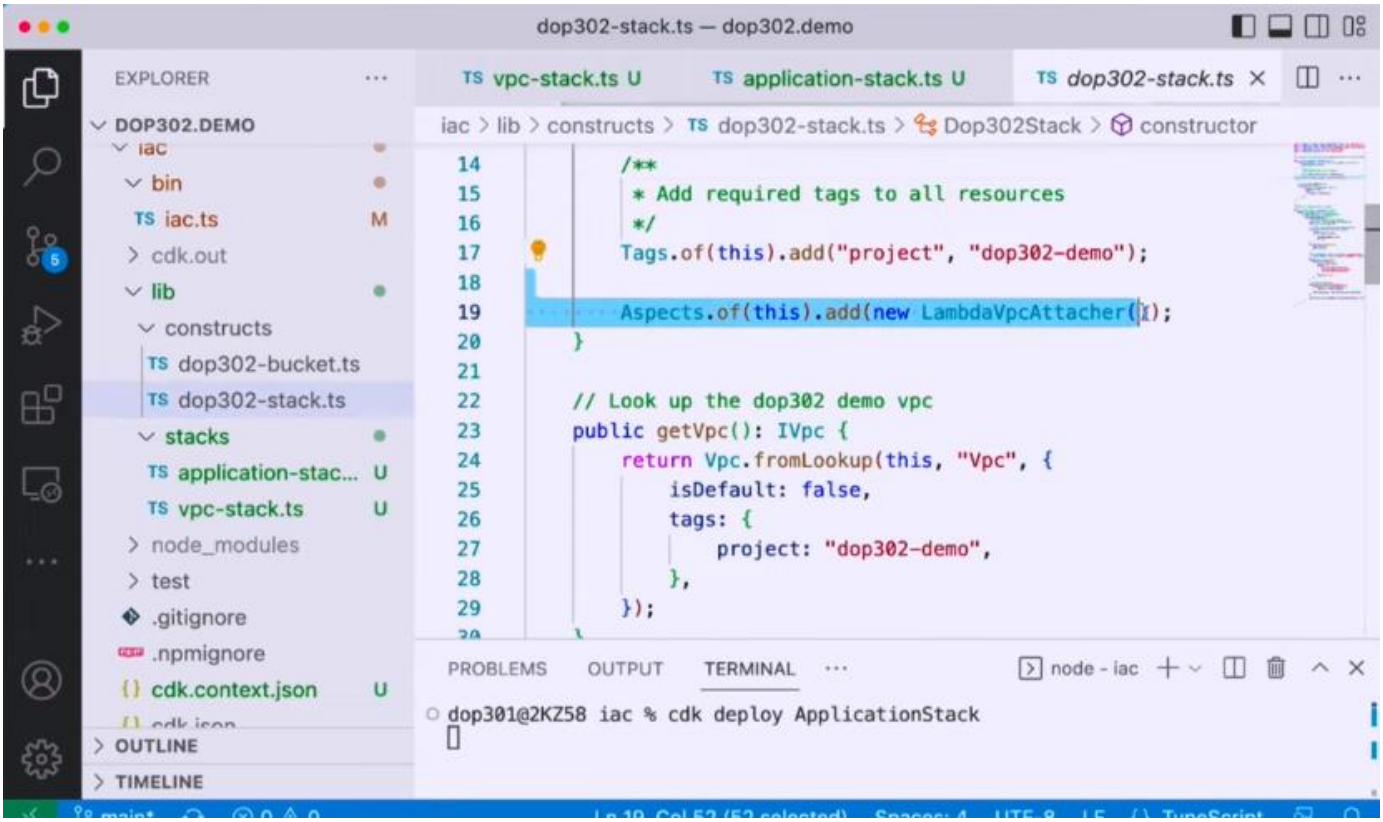

application-stack.ts — dop302.demo
EXPLORER
DOP302.DEMO
iac
bin
iac.ts M
cdk.out
lib
constructs
dop302-bucket.ts
dop302-stack.ts
stacks
application-st... 1, U
vpc-stack.ts U
node_modules
test
vpc-stack.ts U
application-stack.ts 1, U
dop302-bucket.
iac > lib > stacks > application-stack.ts > ApplicationStack > constructor
import { StackProps } from "aws-cdk-lib";
import { Construct } from "constructs";
import { Dop302Bucket } from "../constructs/dop302-bucket";
import { Dop302Stack } from "../constructs/dop302-stack";
export class ApplicationStack extends Dop302Stack {
constructor(scope: Construct, id: string, props?: StackProps
super(scope, id, props);
const bucket = new Dop302Bucket(this, 'Bucket');
const putObjectFunction = new
}
}

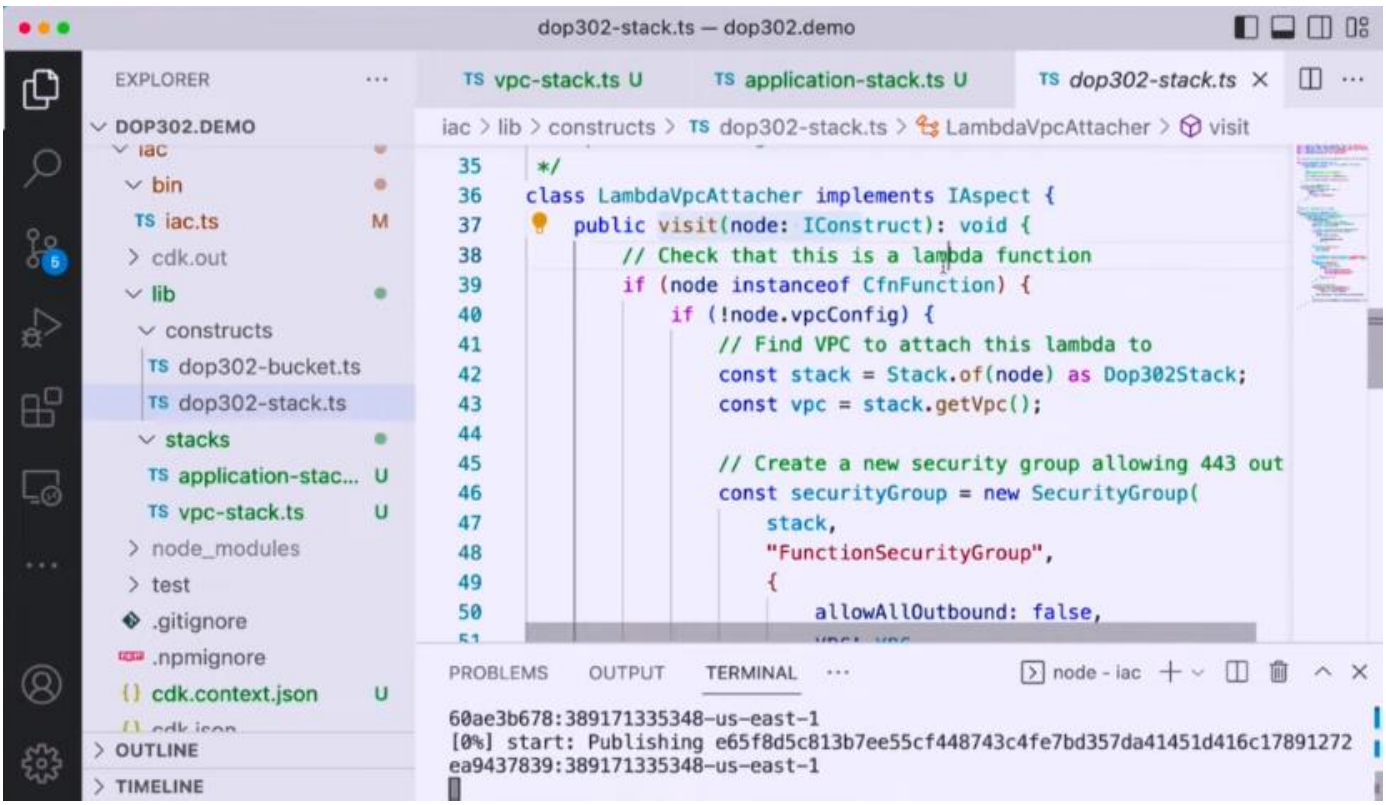

application-stack.ts — dop302.demo
EXPLORER
DOP302.DE...
iac
bin
iac.ts M
cdk.out
lib
constructs
dop302-bucket.ts
dop302-stack.ts
stacks
application-stac... U
vpc-stack.ts U
node_modules
test
.gitignore
.npmignore
cdk.context.json U
vpc-stack.ts U
application-stack.ts U
dop302-bucket.ts
iac > lib > stacks > application-stack.ts > ApplicationStack > constructor
constructor(scope: Construct, id: string, props?: StackProps) {
super(scope, id, props);
const bucket = new Dop302Bucket(this, 'Bucket');
const putObjectFunction = new DockerImageFunction(this, 'Fur
code: DockerImageCode.fromImageAsset("../put-object"),
environment: {
BUCKET_NAME: bucket.bucketName
}
})
bucket.grantWrite(putObjectFunction)
}
PROBLEMS
OUTPUT
TERMINAL
zsh - iac
Total time: 226.63s

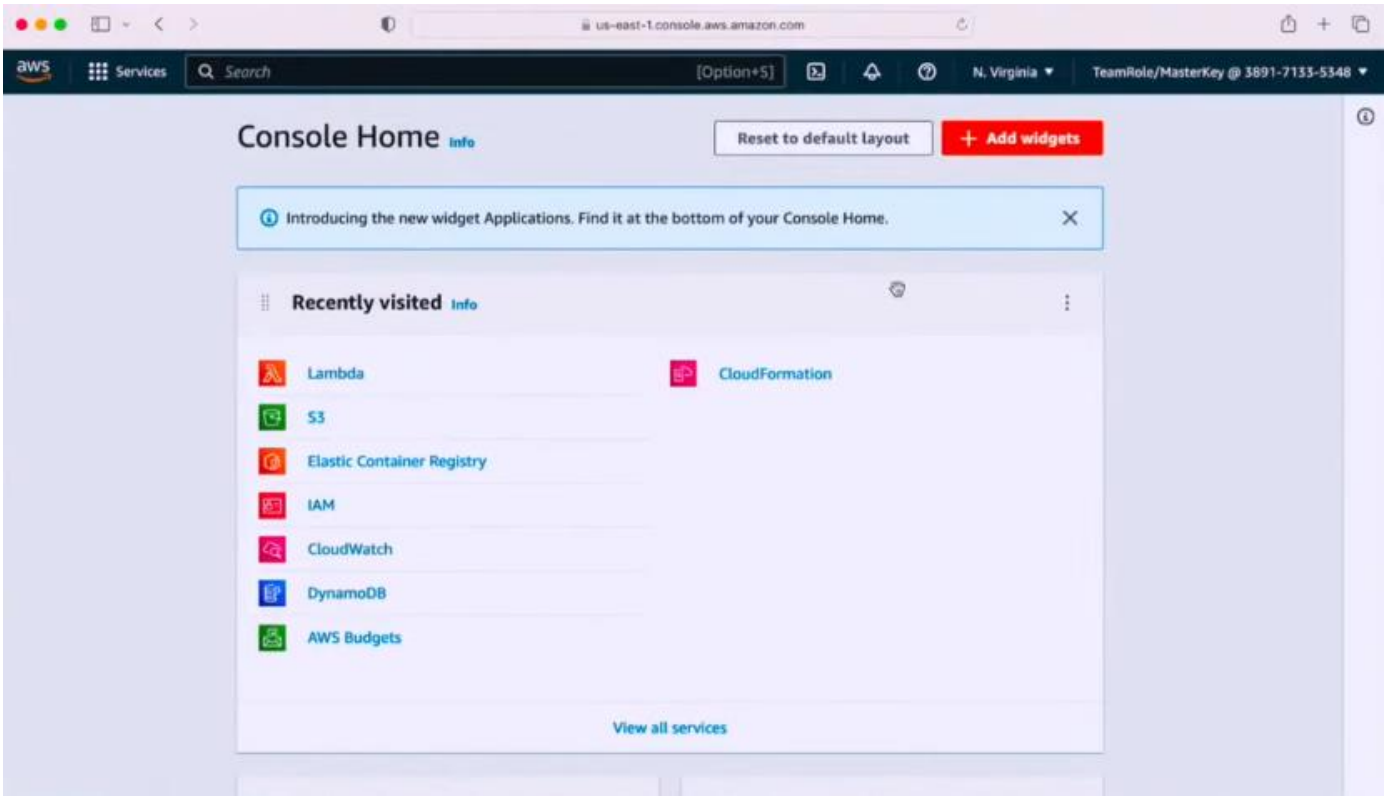

iac.ts — dop302.demo
EXPLORER
DOP302.DEMO
iac
bin
iac.ts M
cdk.out
lib
constructs
dop302-bucket.ts
dop302-stack.ts
stacks
application-stac... U
vpc-stack.ts U
node_modules
test
iac.ts M
vpc-stack.ts U
application-stack.ts U
iac > bin > iac.ts > ...
#!/usr/bin/env node
import "source-map-support/register";
import * as cdk from "aws-cdk-lib";
import { VpcStack } from "../lib/stacks/vpc-stack";
import { ApplicationStack } from "../lib/stacks/application-stac
const app = new cdk.App();
new VpcStack(app, 'VpcStack', {
env: {
account: process.env.CDK_DEFAULT_ACCOUNT,
region: 'us-east-1'
}
})
new ApplicationStack(app, 'ApplicationStack',)

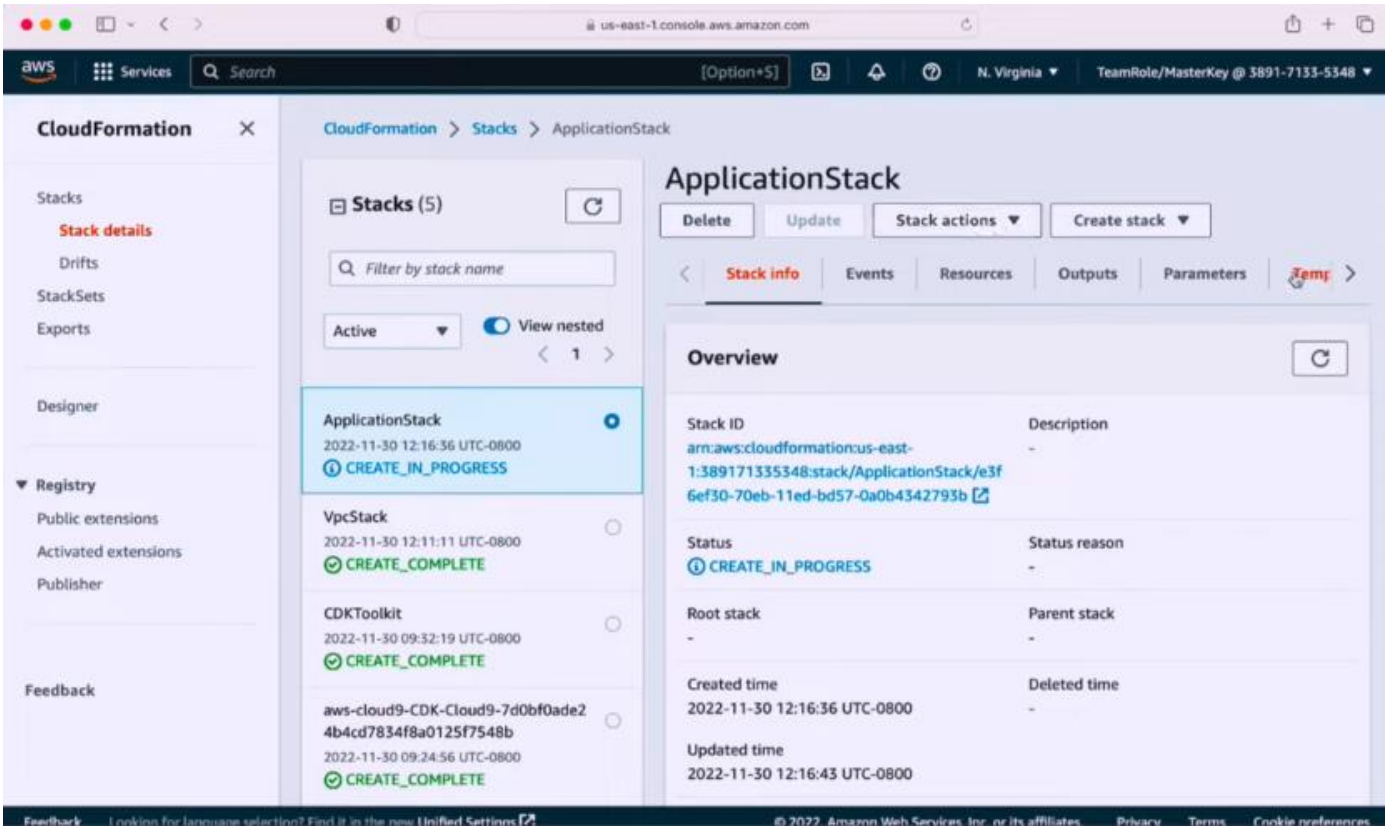

iac.ts — dop302.demo
EXPLORER
DOP302.DEMO
iac
bin
iac.ts M
cdk.out
lib
constructs
dop302-bucket.ts
dop302-stack.ts
stacks
application-stac... U
vpc-stack.ts U
iac.ts M
vpc-stack.ts U
application-stack.ts U
iac > bin > iac.ts > ...
region: 'us-east-1'
}
})
new ApplicationStack(app, 'ApplicationStack', {
env: {
account: process.env.CDK_DEFAULT_ACCOUNT,
region: 'us-east-1'
}
})

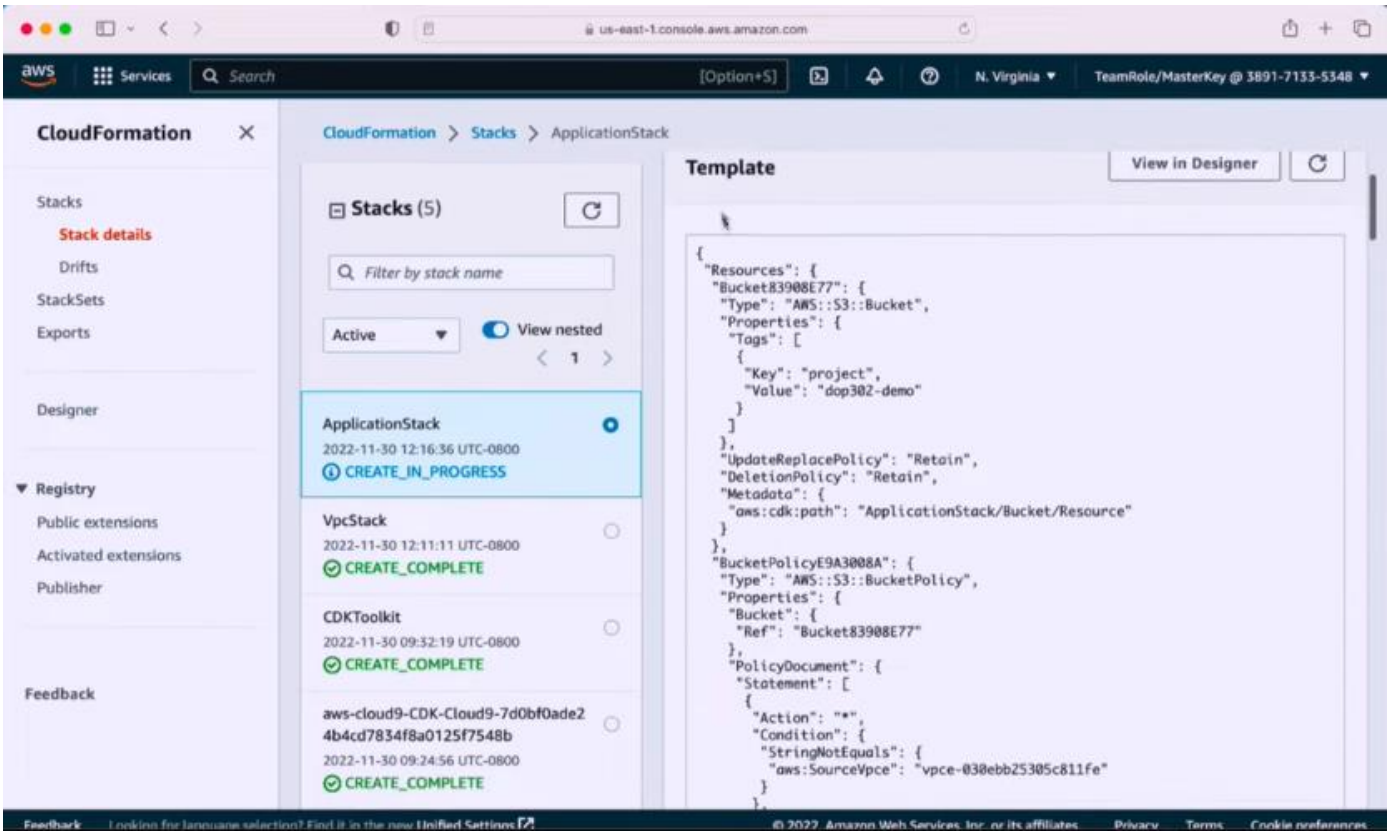

.npmignore
cdk.context.json U
OUTLINE
PROBLEMS
OUTPUT
TERMINAL
node - iac
dop301@2KZ58 iac % cdk deploy ApplicationStack

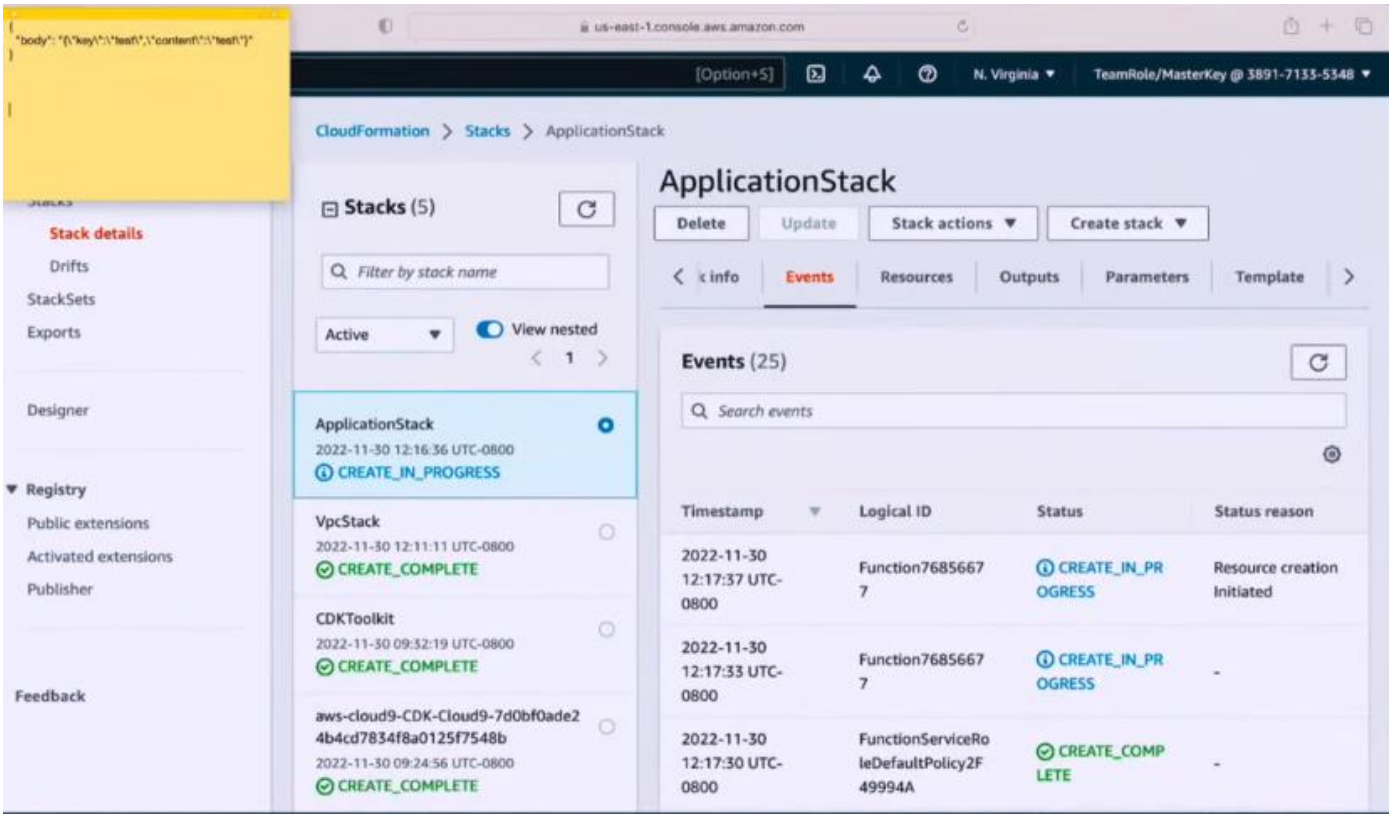

dop302-stack.ts — dop302.demo
EXPLORER
DOP302.DEMO
iac
bin
iac.ts M
cdk.out
lib
constructs
dop302-bucket.ts
dop302-stack.ts
stacks
application-stac... U
vpc-stack.ts U
node_modules
test
.gitignore
.npmignore
cdk.context.json U
OUTLINE
TIMELINE
vpc-stack.ts U
application-stack.ts U
dop302-stack.ts
iac > lib > constructs > dop302-stack.ts > Dop302Stack > constructor
/**
* Add required tags to all resources
*/
Tags.of(this).add("project", "dop302-demo");
Aspects.of(this).add(new LambdaVpcAttacher());
}
// Look up the dop302 demo vpc
public getVpc(): IVpc {
return Vpc.fromLookup(this, "Vpc", {
isDefault: false,
tags: {
project: "dop302-demo",
},
});
PROBLEMS
OUTPUT
TERMINAL
node - iac
dop301@2KZ58 iac % cdk deploy ApplicationStack

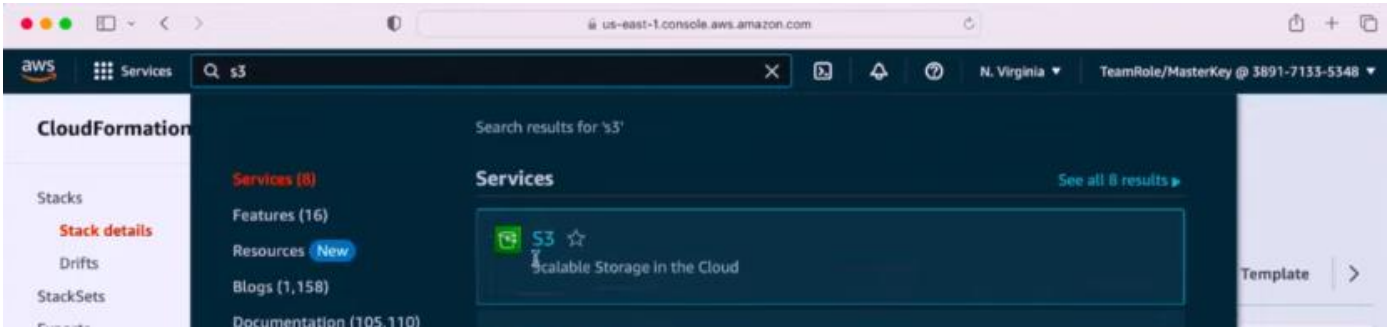

dop302-stack.ts — dop302.demo
EXPLORER
DOP302.DEMO
iac
bin
iac.ts M
cdk.out
lib
constructs
dop302-bucket.ts
dop302-stack.ts
stacks
application-stac... U
vpc-stack.ts U
node_modules
test
.gitignore
.npmignore
cdk.context.json U
OUTLINE
TIMELINE
vpc-stack.ts U
application-stack.ts U
dop302-stack.ts
iac > lib > constructs > dop302-stack.ts > LambdaVpcAttacher > visit
*/
class LambdaVpcAttacher implements IAspect {
public visit(node: IConstruct): void {
// Check that this is a lambda function
if (node instanceof CfnFunction) {
if (!node.vpcConfig) {
// Find VPC to attach this lambda to
const stack = Stack.of(node) as Dop302Stack;
const vpc = stack.getVpc();
// Create a new security group allowing 443 out
const securityGroup = new SecurityGroup(
stack,
"FunctionSecurityGroup",
{
allowAllOutbound: false,
PROBLEMS
OUTPUT
TERMINAL
node - iac
60ae3b678:389171335348-us-east-1
[0%] start: Publishing e65f8d5c813b7ee55cf448743c4fe7bd357da41451d416c17891272ea9437839:389171335348-us-east-1

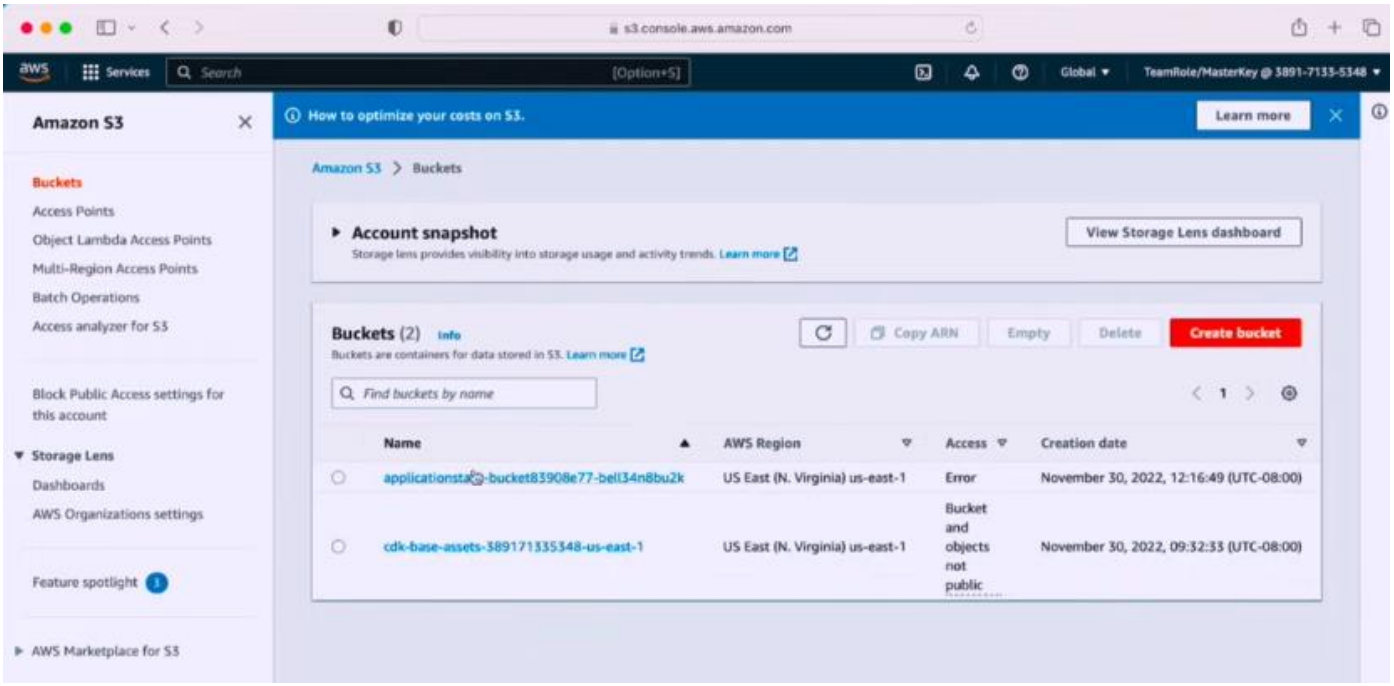

us-east-1.console.aws.amazon.com
aws
Services
Search
[Option+S]
N. Virginia
TeamRole/MasterKey @ 3891-7133-5348
Console Home Info
Reset to default layout
+ Add widgets
Introducing the new widget Applications. Find it at the bottom of your Console Home.
Recently visited Info
Lambda
CloudFormation
S3
Elastic Container Registry
IAM
CloudWatch
DynamoDB
AWS Budgets
View all services

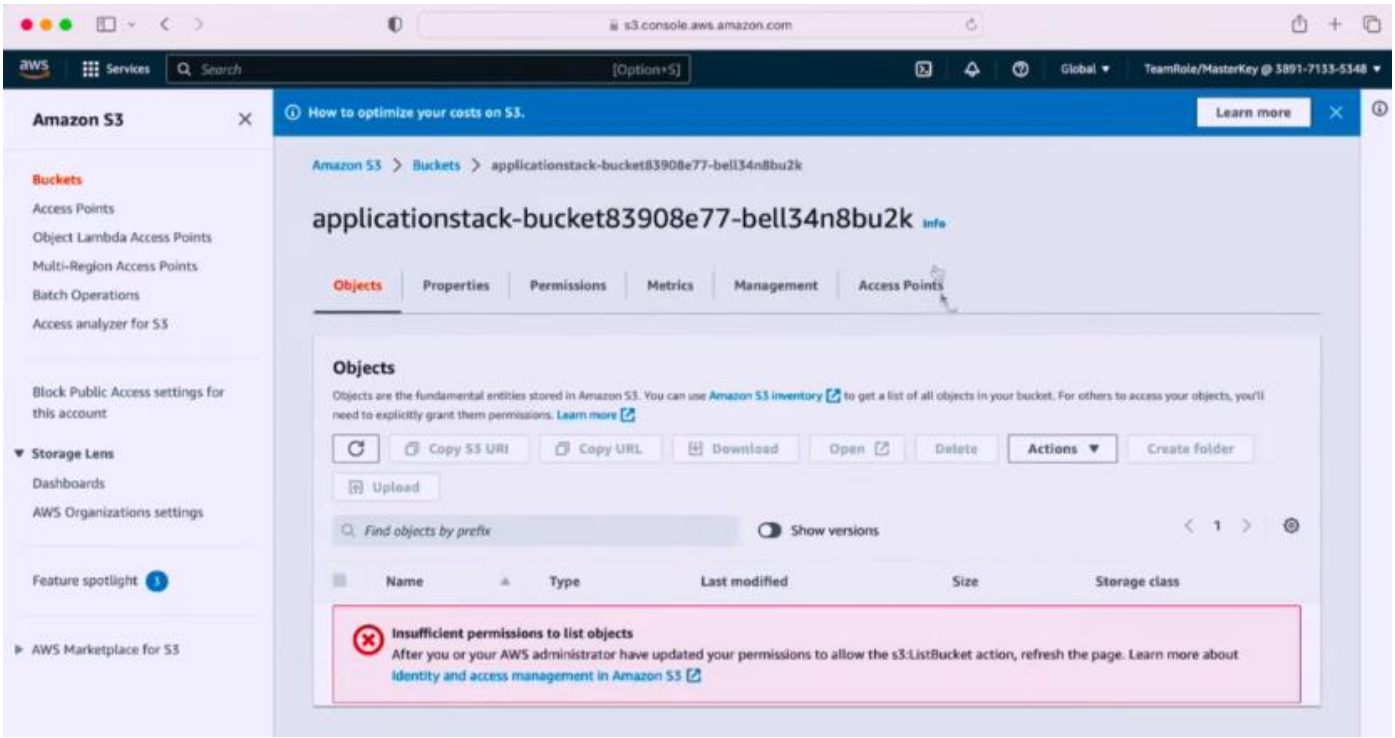

us-east-1.console.aws.amazon.com
aws
Services
Search
[Option+S]
N. Virginia
TeamRole/MasterKey @ 3891-7133-5348
CloudFormation
Stacks
Stack details
Drifts
StackSets
Exports
Designer
Registry
Public extensions
Activated extensions
Publisher
Feedback
CloudFormation > Stacks > ApplicationStack
Stacks (5)
Filter by stack name
Active
View nested
1
ApplicationStack
2022-11-30 12:16:36 UTC-0800
CREATE_IN_PROGRESS
VpcStack
2022-11-30 12:11:11 UTC-0800
CREATE_COMPLETE
CDKToolkit
2022-11-30 09:32:19 UTC-0800
CREATE_COMPLETE
aws-cloud9-CDK-Cloud9-7d0bf0ade24b4cd7834f8a0125f7548b
2022-11-30 09:24:56 UTC-0800
CREATE_COMPLETE
ApplicationStack
Delete
Update
Stack actions
Create stack
Stack info
Events
Resources
Outputs
Parameters
Overview
Stack ID
arn:aws:cloudformation:us-east-1:389171335348:stack/ApplicationStack/e3f6ef30-70eb-11ed-bd57-0a0b4342793b
Description
-
Status
CREATE_IN_PROGRESS
Status reason
-
Root stack
-
Parent stack
-
Created time
2022-11-30 12:16:36 UTC-0800
Deleted time
-
Updated time
2022-11-30 12:16:43 UTC-0800
Feedback
Looking for language selection? Find it in the new Unified Settings
© 2022, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

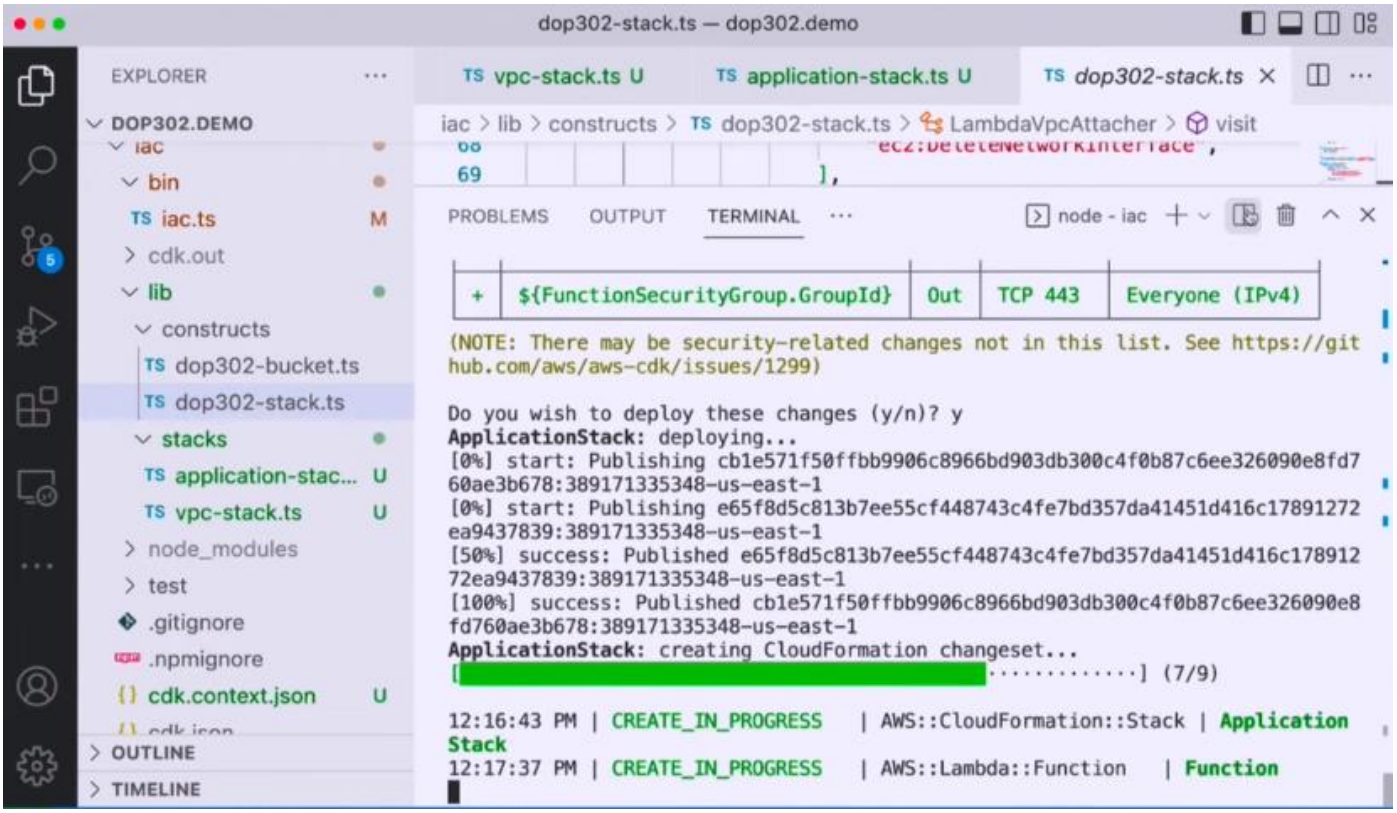

us-east-1.console.aws.amazon.com
Services
Search
[Option+S]
N. Virginia
TeamRole/MasterKey @ 3891-7133-5348
CloudFormation
Stacks
Stack details
Drifts
StackSets
Exports
Designer
Registry
Public extensions
Activated extensions
Publisher
Feedback
CloudFormation > Stacks > ApplicationStack
Stacks (5)
Filter by stack name
Active
View nested
1
ApplicationStack
2022-11-30 12:16:36 UTC-0800
CREATE_IN_PROGRESS
VpcStack
2022-11-30 12:11:11 UTC-0800
CREATE_COMPLETE
CDKToolkit
2022-11-30 09:32:19 UTC-0800
CREATE_COMPLETE
aws-cloud9-CDK-Cloud9-7d0bf0ade24b4cd7834f8a0125f7548b
2022-11-30 09:24:56 UTC-0800
CREATE_COMPLETE
Template
View in Designer
{
"Resources": {
"Bucket83908E77": {
"Type": "AWS::S3::Bucket",
"Properties": {
"Tags": [
{
"Key": "project",
"Value": "dop302-demo"
}
]
},
"UpdateReplacePolicy": "Retain",
"DeletionPolicy": "Retain",
"Metadata": {
"aws:cdk:path": "ApplicationStack/Bucket/Resource"
}
},
"BucketPolicyE9A3008A": {
"Type": "AWS::S3::BucketPolicy",
"Properties": {
"Bucket": {
"Ref": "Bucket83908E77"
},
"PolicyDocument": {
"Statement": [
{
"Action": "*",
"Condition": {
"StringNotEquals": {
"aws:SourceVpce": "vpce-030ebb25305c811fe"
}
},
© 2022, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

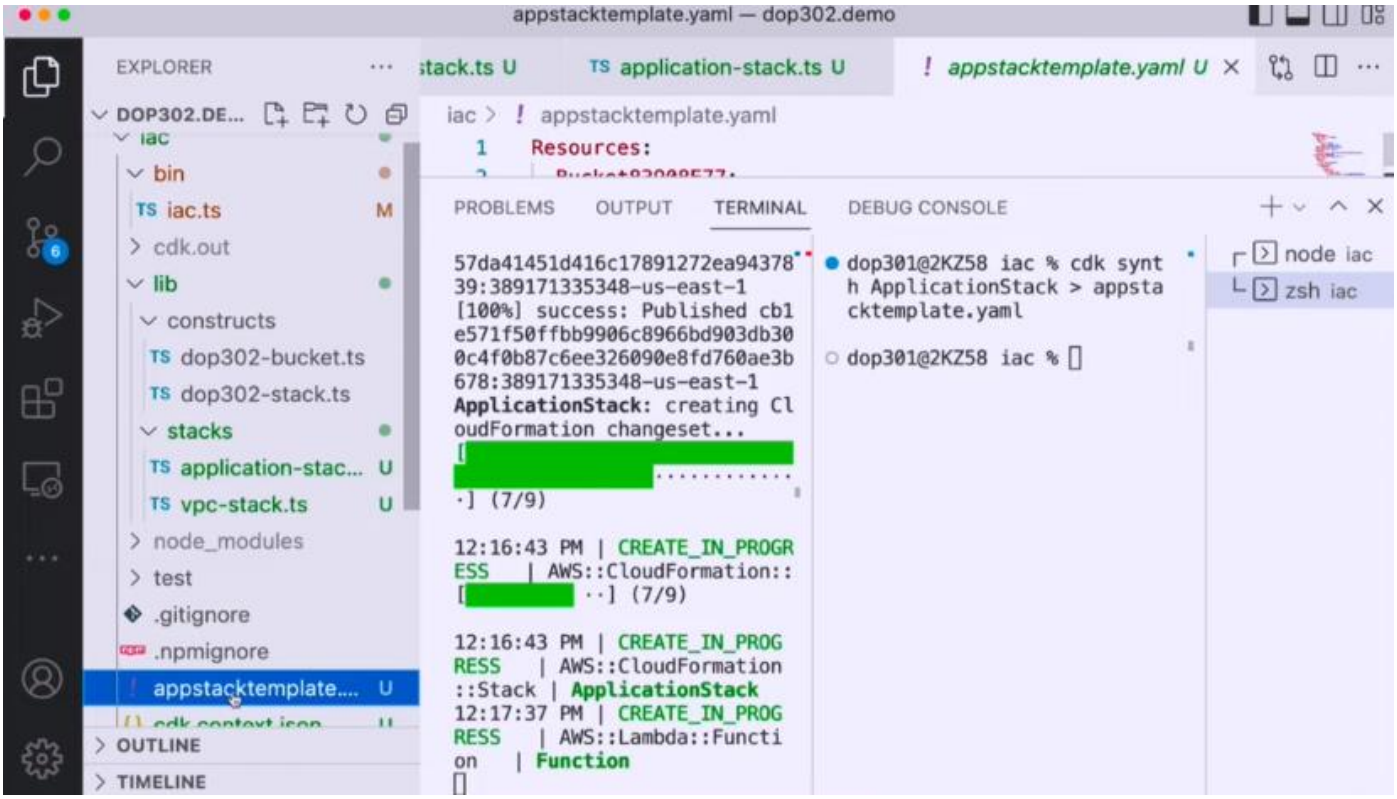

{
"body": "{\"key\":\"test\",\"content\":\"test\"}"
}
us-east-1.console.aws.amazon.com
[Option+S]
N. Virginia
TeamRole/MasterKey @ 3891-7133-5348
CloudFormation > Stacks > ApplicationStack
Stacks (5)
Filter by stack name
Active
View nested
1
ApplicationStack
2022-11-30 12:16:36 UTC-0800
CREATE_IN_PROGRESS
VpcStack
2022-11-30 12:11:11 UTC-0800
CREATE_COMPLETE
CDKToolkit
2022-11-30 09:32:19 UTC-0800
CREATE_COMPLETE
aws-cloud9-CDK-Cloud9-7d0bf0ade24b4cd7834f8a0125f7548b
2022-11-30 09:24:56 UTC-0800
CREATE_COMPLETE
ApplicationStack
Delete
Update
Stack actions
Create stack
info
Events
Resources
Outputs
Parameters
Template
Events (25)
Search events
Timestamp
Logical ID
Status
Status reason
2022-11-30 12:17:37 UTC-0800
Function76856677
CREATE_IN_PROGRESS
Resource creation Initiated
2022-11-30 12:17:33 UTC-0800
Function76856677
CREATE_IN_PROGRESS
-
2022-11-30 12:17:30 UTC-0800
FunctionServiceRoleDefaultPolicy2F49994A
CREATE_COMPLETE
-

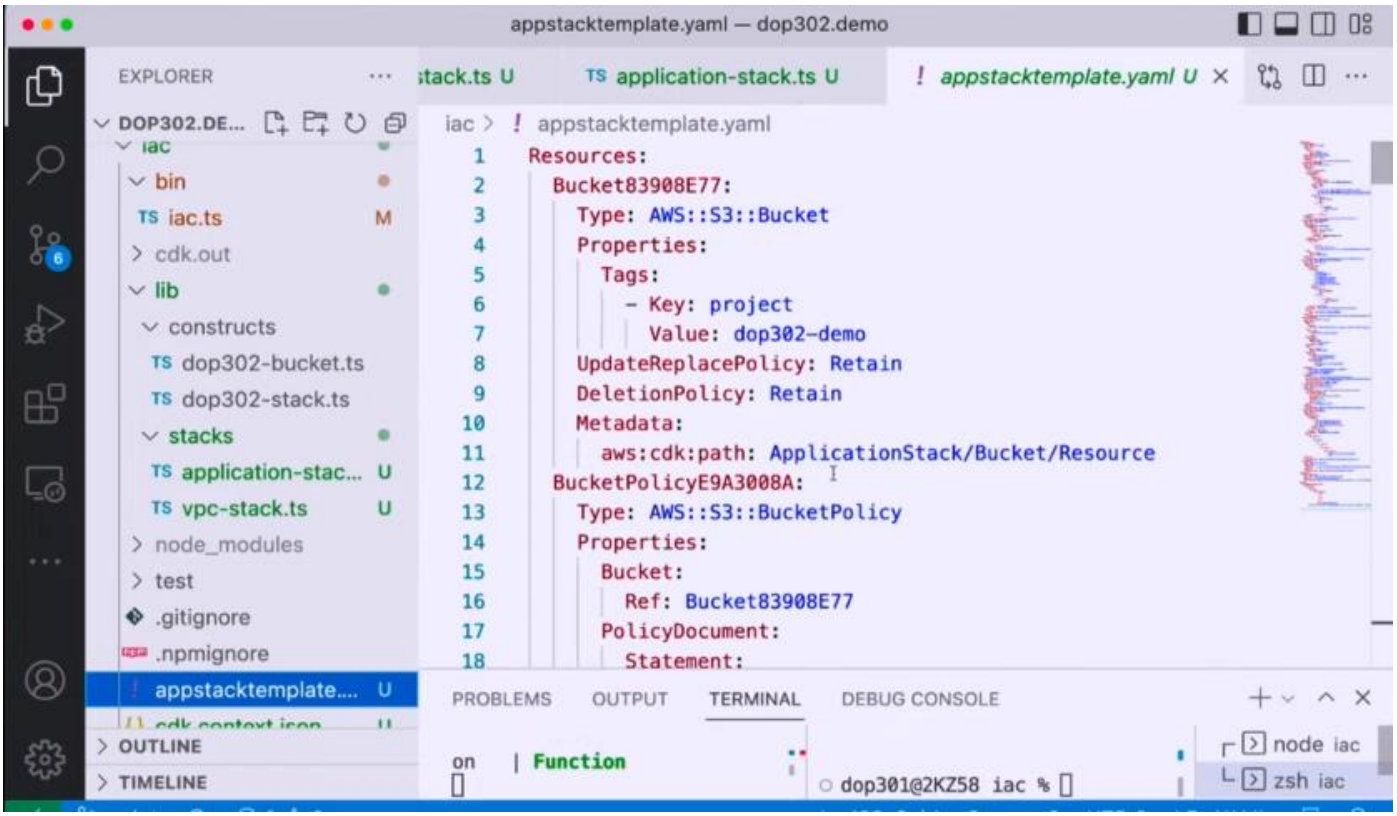

us-east-1.console.aws.amazon.com
Services
s3
N. Virginia
TeamRole/MasterKey @ 3891-7133-5348
CloudFormation
Search results for 's3'
Services (8)
Features (16)
Resources New
Blogs (1,158)
Services
See all 8 results
S3
Scalable Storage in the Cloud
Stacks
Stack details
Drifts
StackSets
Template

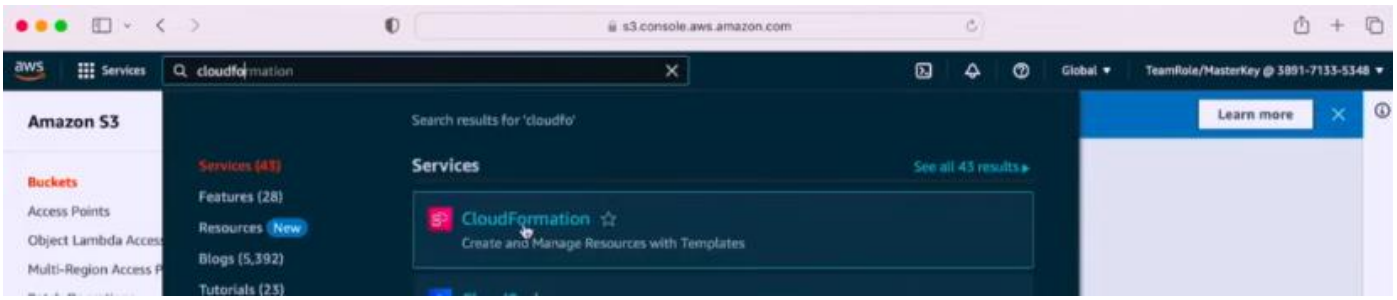

s3.console.aws.amazon.com
Search
[Option+S]
Global
TeamRole/MasterKey @ 3891-7133-5348
Amazon S3
How to optimize your costs on S3.
Learn more
Buckets
Access Points
Object Lambda Access Points
Multi-Region Access Points
Batch Operations
Access analyzer for S3
Block Public Access settings for this account
Storage Lens
Dashboards
AWS Organizations settings
Feature spotlight
AWS Marketplace for S3
Amazon S3 > Buckets
Account snapshot
Storage lens provides visibility into storage usage and activity trends. Learn more
View Storage Lens dashboard
Buckets (2) Info
Buckets are containers for data stored in S3. Learn more
Copy ARN
Empty
Delete
Create bucket
Find buckets by name
Name
AWS Region
Access
Creation date
applicationsta-bucket83908e77-bell34n8bu2k
US East (N. Virginia) us-east-1
Error
November 30, 2022, 12:16:49 (UTC-08:00)
cdk-base-assets-389171335348-us-east-1
US East (N. Virginia) us-east-1
Bucket and objects not public
November 30, 2022, 09:32:33 (UTC-08:00)

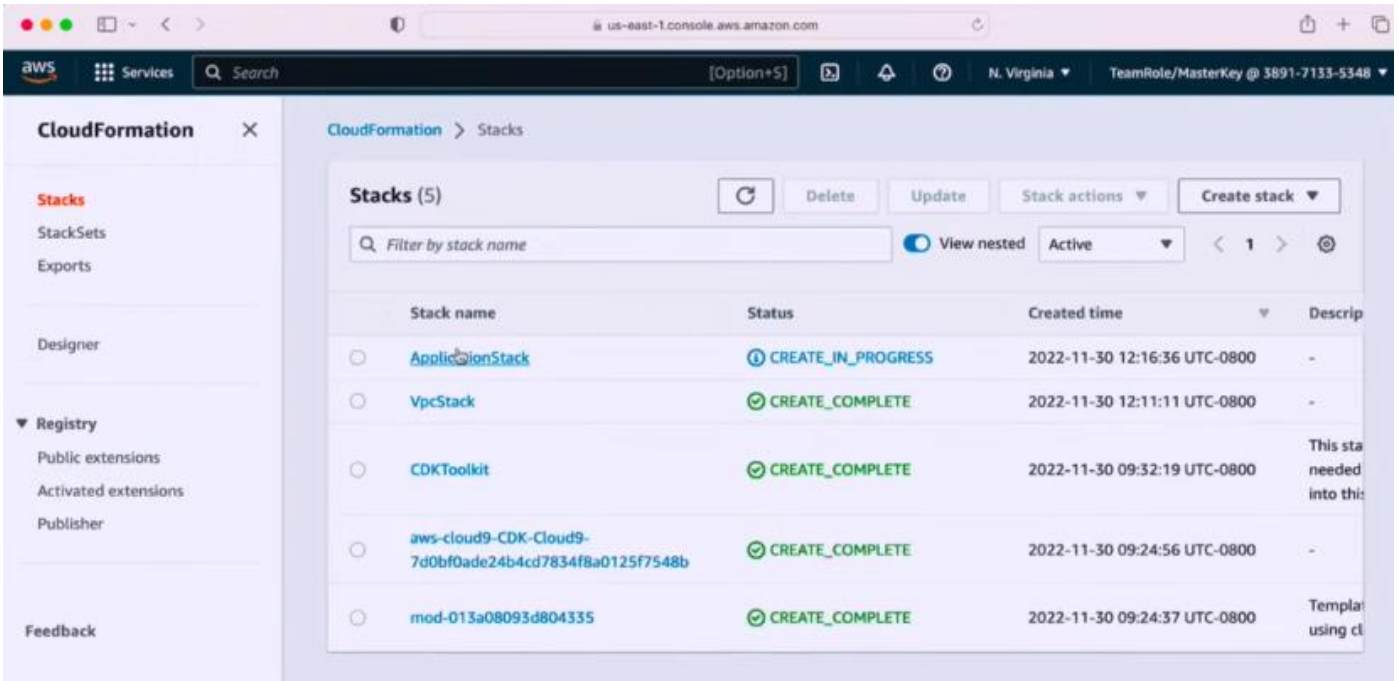

s3.console.aws.amazon.com
Search
[Option+S]
Global
TeamRole/MasterKey @ 3891-7133-5348
Amazon S3
How to optimize your costs on S3.
Learn more
Buckets
Access Points
Object Lambda Access Points
Multi-Region Access Points
Batch Operations
Access analyzer for S3
Block Public Access settings for this account
Storage Lens
Dashboards
AWS Organizations settings
Feature spotlight
AWS Marketplace for S3
Amazon S3 > Buckets > applicationstack-bucket83908e77-bell34n8bu2k
applicationstack-bucket83908e77-bell34n8bu2k Info
Objects
Properties
Permissions
Metrics
Management
Access Points
Objects
Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. Learn more
Copy S3 URI
Copy URL
Download
Open
Delete
Actions
Create folder
Upload
Find objects by prefix
Show versions
Name
Type
Last modified
Size
Storage class
Insufficient permissions to list objects
After you or your AWS administrator have updated your permissions to allow the s3:ListBucket action, refresh the page. Learn more about identity and access management in Amazon S3

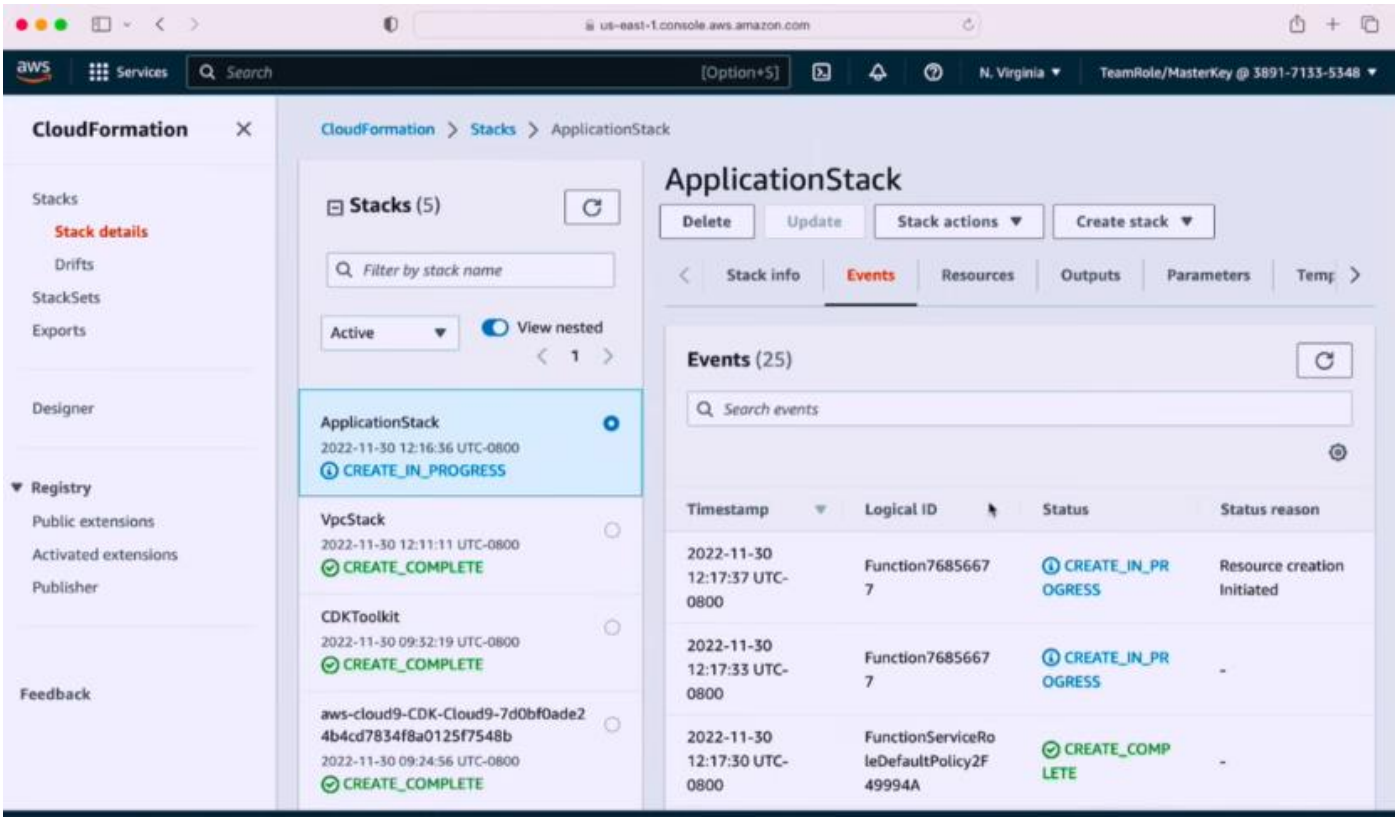

dop302-stack.ts — dop302.demo
EXPLORER
DOP302.DEMO
iac
bin
iac.ts
cdk.out
lib
constructs
dop302-bucket.ts
dop302-stack.ts
stacks
application-stac...
vpc-stack.ts
node_modules
test
.gitignore
.npmignore
cdk.context.json
OUTLINE
TIMELINE
vpc-stack.ts U
application-stack.ts U
dop302-stack.ts
iac > lib > constructs > dop302-stack.ts > LambdaVpcAttacher > visit
PROBLEMS
OUTPUT
TERMINAL
node - iac
${FunctionSecurityGroup.GroupId}
Out
TCP 443
Everyone (IPv4)
(NOTE: There may be security-related changes not in this list. See https://github.com/aws/aws-cdk/issues/1299)
Do you wish to deploy these changes (y/n)? y
ApplicationStack: deploying...
[0%] start: Publishing cb1e571f50ffbb9906c8966bd903db300c4f0b87c6ee326090e8fd760ae3b678:389171335348-us-east-1
[0%] start: Publishing e65f8d5c813b7ee55cf448743c4fe7bd357da41451d416c17891272ea9437839:389171335348-us-east-1
[50%] success: Published e65f8d5c813b7ee55cf448743c4fe7bd357da41451d416c17891272ea9437839:389171335348-us-east-1
[100%] success: Published cb1e571f50ffbb9906c8966bd903db300c4f0b87c6ee326090e8fd760ae3b678:389171335348-us-east-1
ApplicationStack: creating CloudFormation changeset...
(7/9)
12:16:43 PM | CREATE_IN_PROGRESS | AWS::CloudFormation::Stack | ApplicationStack
12:17:37 PM | CREATE_IN_PROGRESS | AWS::Lambda::Function | Function

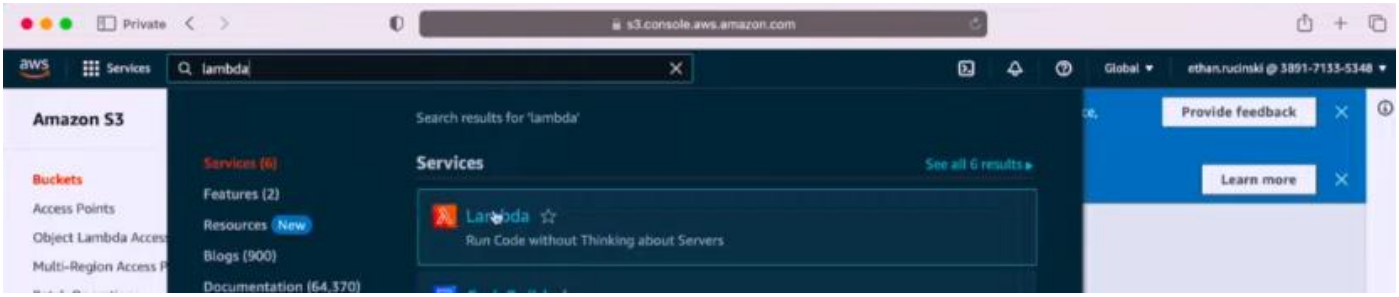

appstacktemplate.yaml — dop302.demo
EXPLORER
iac
bin
iac.ts
cdk.out
lib
constructs
dop302-bucket.ts
dop302-stack.ts
stacks
application-stac...
vpc-stack.ts
node_modules
test
.gitignore
.npmignore
appstacktemplate....
OUTLINE
TIMELINE
application-stack.ts U
appstacktemplate.yaml U
iac > appstacktemplate.yaml
1 Resources:
PROBLEMS
OUTPUT
TERMINAL
DEBUG CONSOLE
57da41451d416c17891272ea9437839:389171335348-us-east-1
[100%] success: Published cb1e571f50ffbb9906c8966bd903db300c4f0b87c6ee326090e8fd760ae3b678:389171335348-us-east-1
ApplicationStack: creating CloudFormation changeset...
(7/9)
12:16:43 PM | CREATE_IN_PROGRESS | AWS::CloudFormation::
(7/9)
12:16:43 PM | CREATE_IN_PROGRESS | AWS::CloudFormation::Stack | ApplicationStack
12:17:37 PM | CREATE_IN_PROGRESS | AWS::Lambda::Function | Function
dop301@2KZ58 iac % cdk synth ApplicationStack > appstacktemplate.yaml
dop301@2KZ58 iac %
node iac
zsh iac

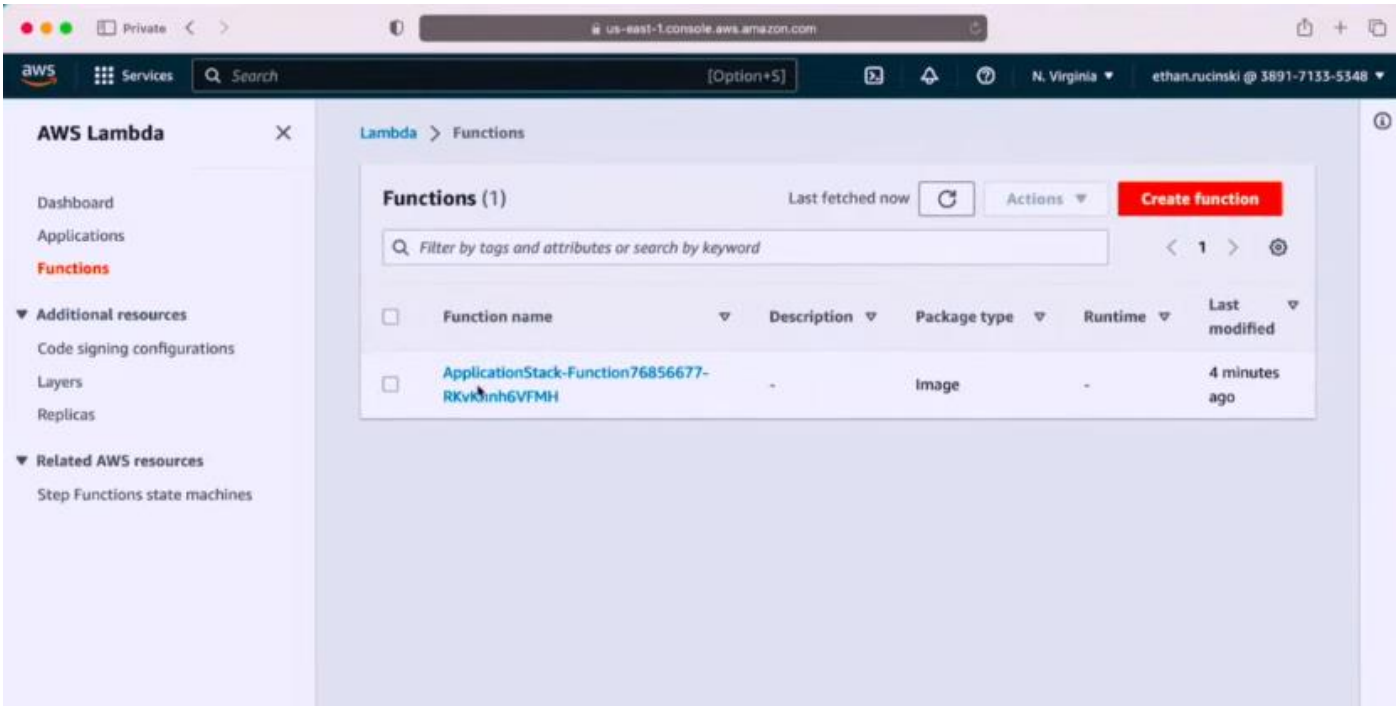

appstacktemplate.yaml — dop302.demo
EXPLORER
application-stack.ts U
appstacktemplate.yaml U
DOP302.DE...
iac
bin
iac.ts
cdk.out
lib
constructs
dop302-bucket.ts
dop302-stack.ts
stacks
application-stac... U
vpc-stack.ts U
node_modules
test
.gitignore
.npmignore
appstacktemplate.... U
OUTLINE
TIMELINE
iac > appstacktemplate.yaml
Resources:
Bucket83908E77:
Type: AWS::S3::Bucket
Properties:
Tags:
- Key: project
Value: dop302-demo
UpdateReplacePolicy: Retain
DeletionPolicy: Retain
Metadata:
aws:cdk:path: ApplicationStack/Bucket/Resource
BucketPolicyE9A3008A:
Type: AWS::S3::BucketPolicy
Properties:
Bucket:
Ref: Bucket83908E77
PolicyDocument:
Statement:
PROBLEMS
OUTPUT
TERMINAL
DEBUG CONSOLE
on | Function
dop301@2KZ58 iac %
node iac
zsh iac

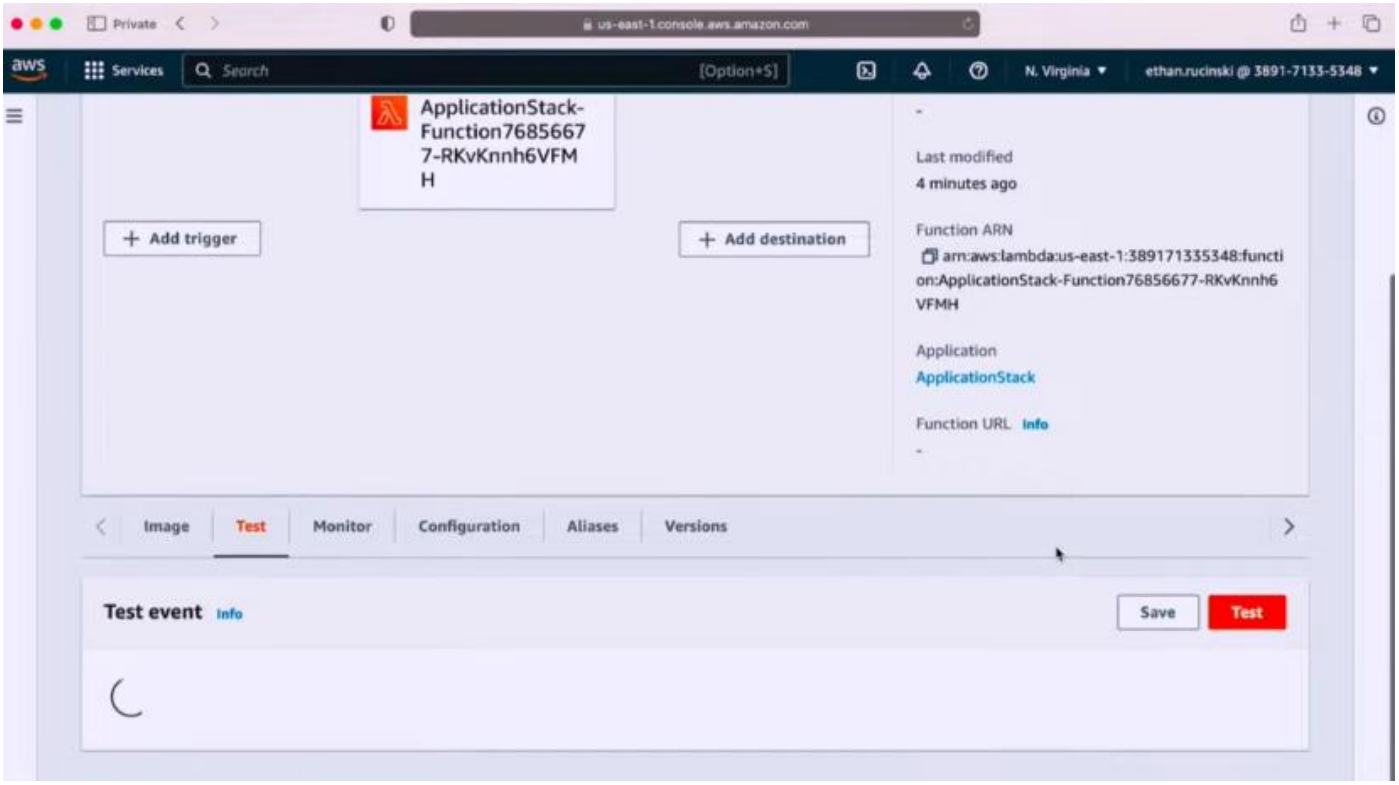

s3.console.aws.amazon.com
Services
cloudformation
Global
TeamRole/MasterKey @ 3891-7133-5348
Amazon S3
Buckets
Access Points
Search results for 'cloudfo'
Services (41)
Features (28)
Resources New
Blogs (5,392)
Tutorials (23)
Services
See all 43 results
CloudFormation
Create and Manage Resources with Templates
Learn more

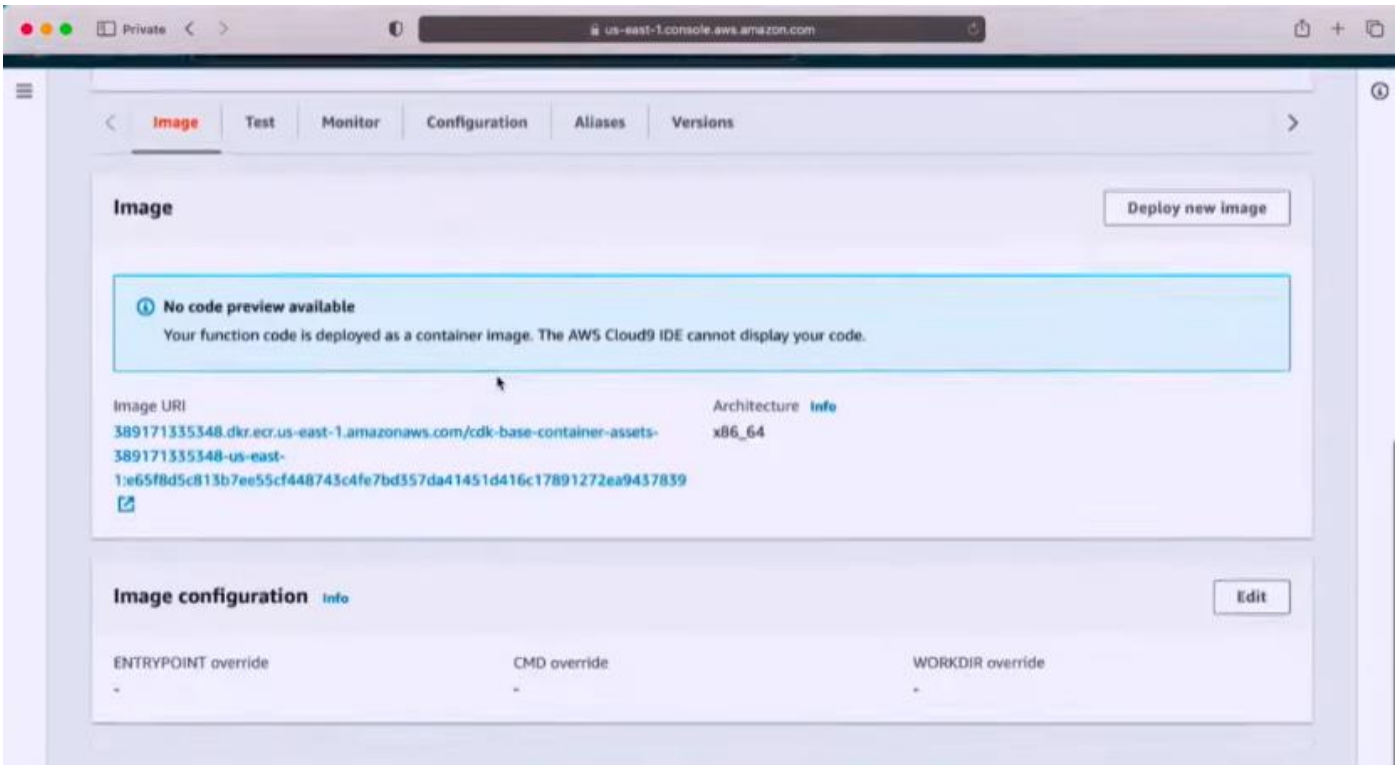

us-east-1.console.aws.amazon.com
Services
Search
[Option+S]
N. Virginia
TeamRole/MasterKey @ 3891-7133-5348
CloudFormation
Stacks
StackSets
Exports
Designer
Registry
Public extensions
Activated extensions
Publisher
Feedback
CloudFormation > Stacks
Stacks (5)
Delete
Update
Stack actions
Create stack
Filter by stack name
View nested
Active
1
Stack name
Status
Created time
ApplicationStack
CREATE_IN_PROGRESS
2022-11-30 12:16:36 UTC-0800
VpcStack
CREATE_COMPLETE
2022-11-30 12:11:11 UTC-0800
CDKToolkit
CREATE_COMPLETE
2022-11-30 09:32:19 UTC-0800
aws-cloud9-CDK-Cloud9-7d0bf0ade24b4cd7834f8a0125f7548b
CREATE_COMPLETE
2022-11-30 09:24:56 UTC-0800
mod-013a08093d804335
CREATE_COMPLETE
2022-11-30 09:24:37 UTC-0800

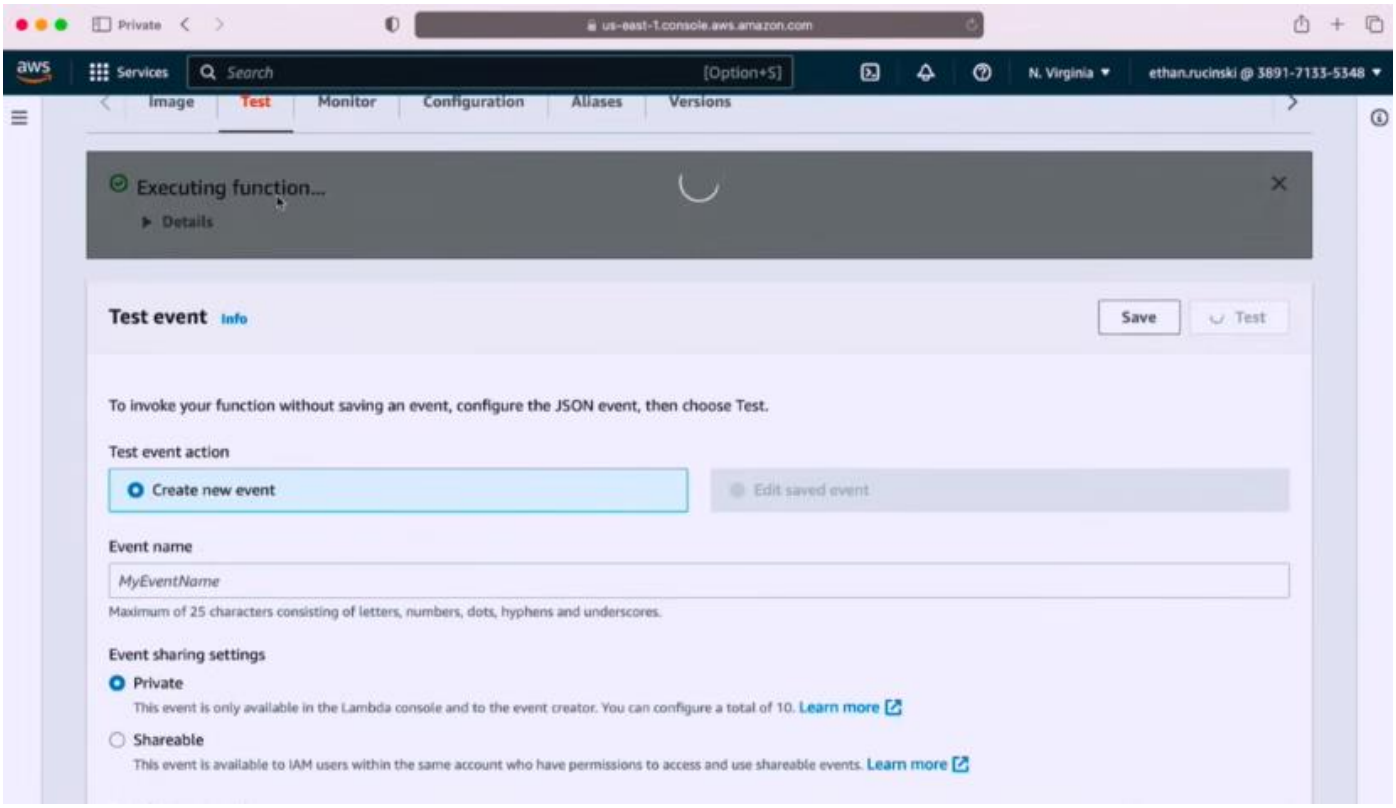

us-east-1.console.aws.amazon.com
Services
Search
[Option+S]
N. Virginia
TeamRole/MasterKey @ 3891-7133-5348
CloudFormation
Stacks
Stack details
Drifts
StackSets
Exports
Designer
Registry
Public extensions
Activated extensions
Publisher
Feedback
CloudFormation > Stacks > ApplicationStack
Stacks (5)
Filter by stack name
Active
View nested
ApplicationStack
2022-11-30 12:16:36 UTC-0800
CREATE_IN_PROGRESS
VpcStack
2022-11-30 12:11:11 UTC-0800
CREATE_COMPLETE
CDKToolkit
2022-11-30 09:32:19 UTC-0800
CREATE_COMPLETE
aws-cloud9-CDK-Cloud9-7d0bf0ade24b4cd7834f8a0125f7548b
2022-11-30 09:24:56 UTC-0800
CREATE_COMPLETE
ApplicationStack
Delete
Update
Stack actions
Create stack
Stack info
Events
Resources
Outputs
Parameters
Events (25)
Search events
Timestamp
Logical ID
Status
Status reason
2022-11-30 12:17:37 UTC-0800
Function76856677
CREATE_IN_PROGRESS
Resource creation Initiated
2022-11-30 12:17:33 UTC-0800
Function76856677
CREATE_IN_PROGRESS
-
2022-11-30 12:17:30 UTC-0800
FunctionServiceRoleDefaultPolicy2F49994A
CREATE_COMPLETE
-

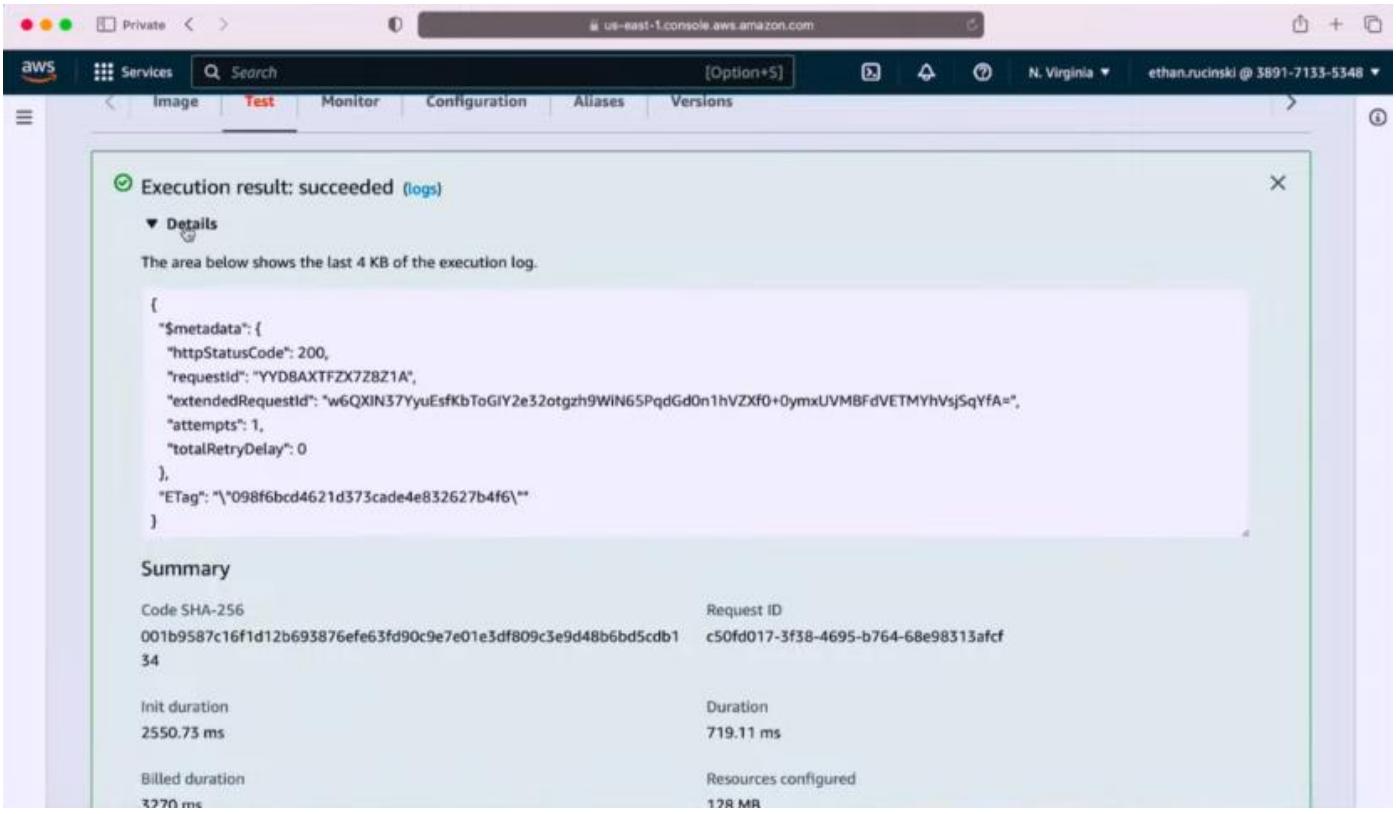

Private
s3.console.aws.amazon.com
Services
lambda
Global
ethan.rucinski @ 3891-7133-5348
Amazon S3
Buckets
Access Points
Search results for 'lambda'
Services (6)
Features (2)
Resources New
Blogs (900)
Services
See all 6 results
Run Code without Thinking about Servers
Provide feedback
Learn more

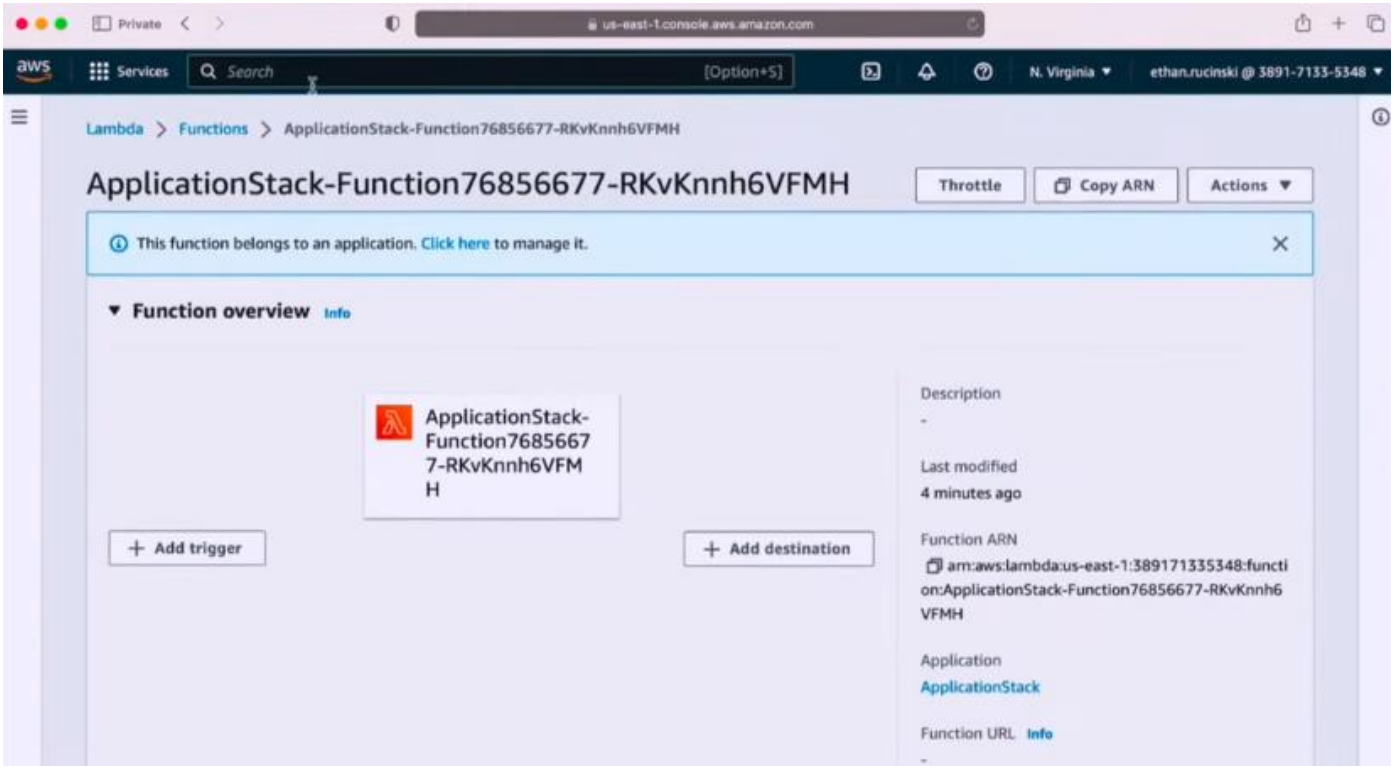

Private
us-east-1.console.aws.amazon.com
Services
Search
[Option+S]
N. Virginia
ethan.rucinski @ 3891-7133-5348
AWS Lambda
Dashboard
Applications
Functions
Additional resources
Code signing configurations
Layers
Replicas
Related AWS resources
Step Functions state machines
Lambda > Functions
Functions (1)
Last fetched now
Actions
Create function
Filter by tags and attributes or search by keyword
Function name
Description
Package type
Runtime
Last modified
ApplicationStack-Function76856677-RKvK8nh6VFMH
Image
4 minutes ago

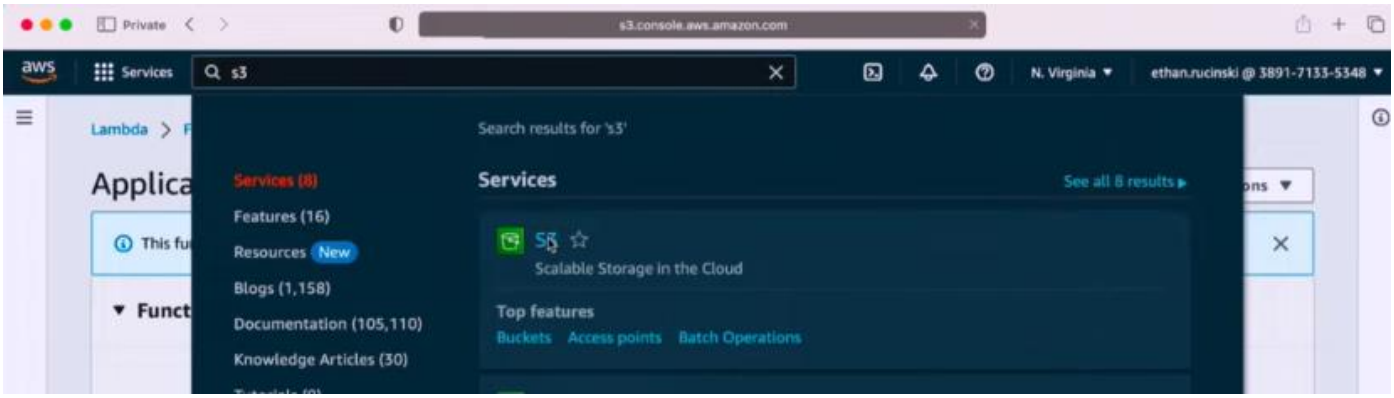

Private
us-east-1.console.aws.amazon.com
aws
Services
Search
[Option+S]
N. Virginia
ethan.rucinski @ 3891-7133-5348
ApplicationStack-Function76856677-RKvKnnh6VFMH
Last modified
4 minutes ago
+ Add trigger
+ Add destination
Function ARN
arn:aws:lambda:us-east-1:389171335348:function:ApplicationStack-Function76856677-RKvKnnh6VFMH
Application
ApplicationStack
Function URL Info
Image
Test
Monitor
Configuration
Aliases
Versions
Test event Info
Save
Test

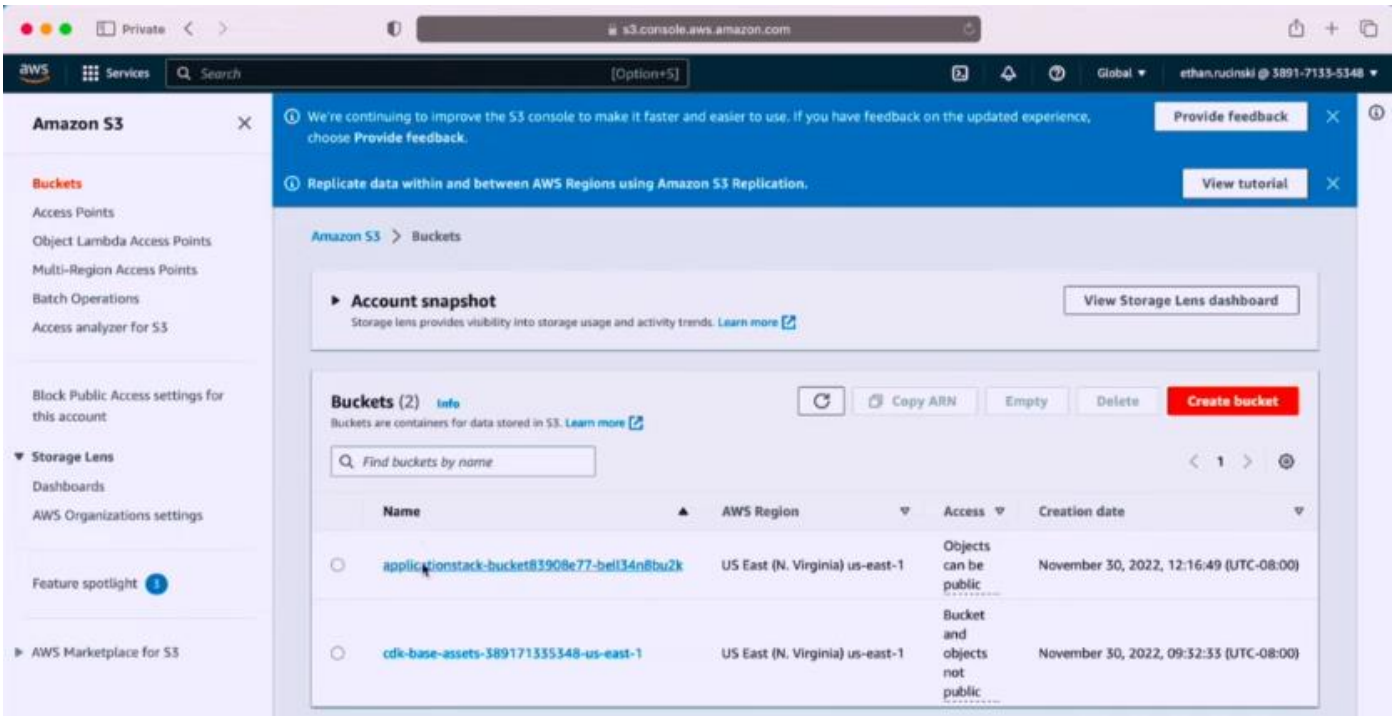

Private
us-east-1.console.aws.amazon.com
Image
Test
Monitor
Configuration
Aliases
Versions
Image
Deploy new image
No code preview available
Your function code is deployed as a container image. The AWS Cloud9 IDE cannot display your code.
Image URI
389171335348.dkr.ecr.us-east-1.amazonaws.com/cdk-base-container-assets-389171335348-us-east-1:e65f8d5c813b7ee55cf448743c4fe7bd357da41451d416c17891272ea9437839
Architecture Info
x86_64
Image configuration Info
Edit
ENTRYPOINT override
-
CMD override
-
WORKDIR override
-

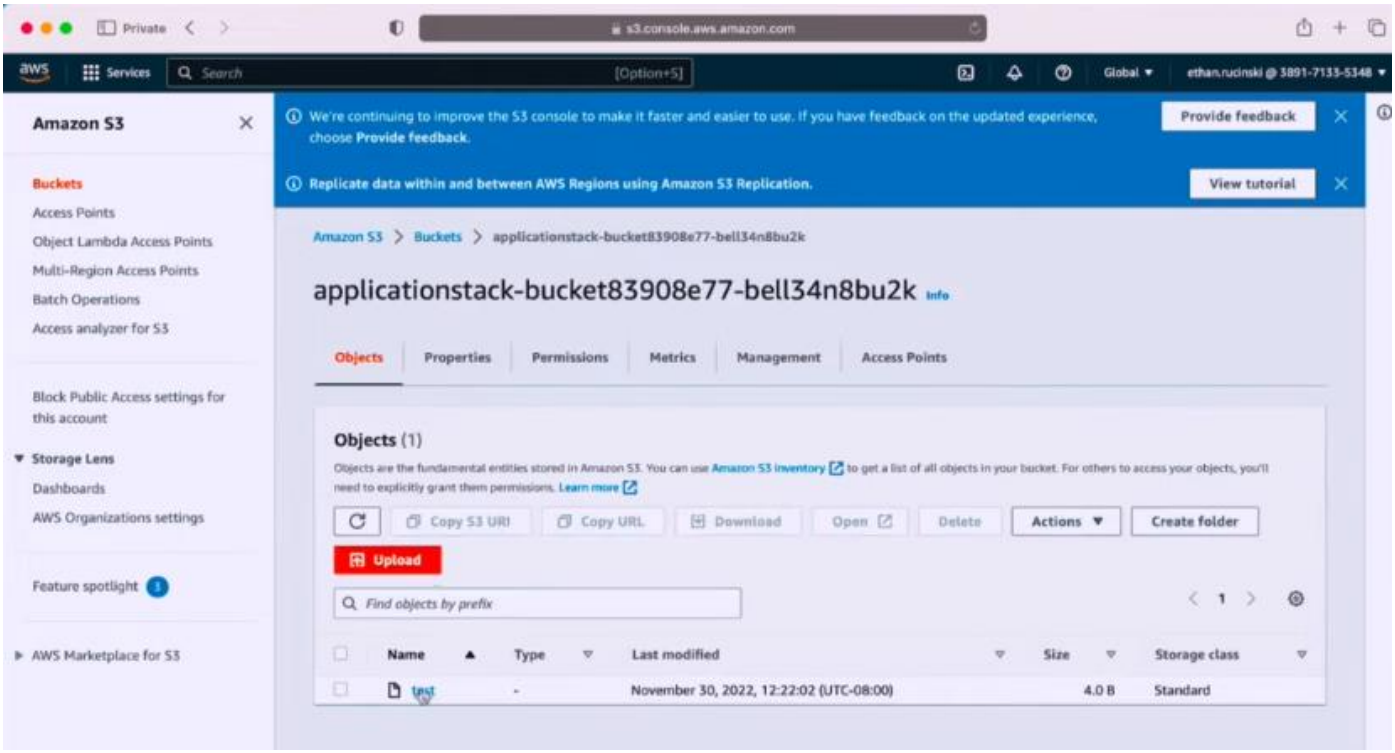

Private
us-east-1.console.aws.amazon.com
aws
Services
Search
[Option+S]
N. Virginia
ethan.rucinski @ 3891-7133-5348
Image
Test
Monitor
Configuration
Aliases
Versions
Executing function...
Details
Test event Info
Save
Test
To invoke your function without saving an event, configure the JSON event, then choose Test.
Test event action
Create new event
Edit saved event
Event name
MyEventName
Maximum of 25 characters consisting of letters, numbers, dots, hyphens and underscores.
Event sharing settings
Private
This event is only available in the Lambda console and to the event creator. You can configure a total of 10. Learn more
Shareable
This event is available to IAM users within the same account who have permissions to access and use shareable events. Learn more

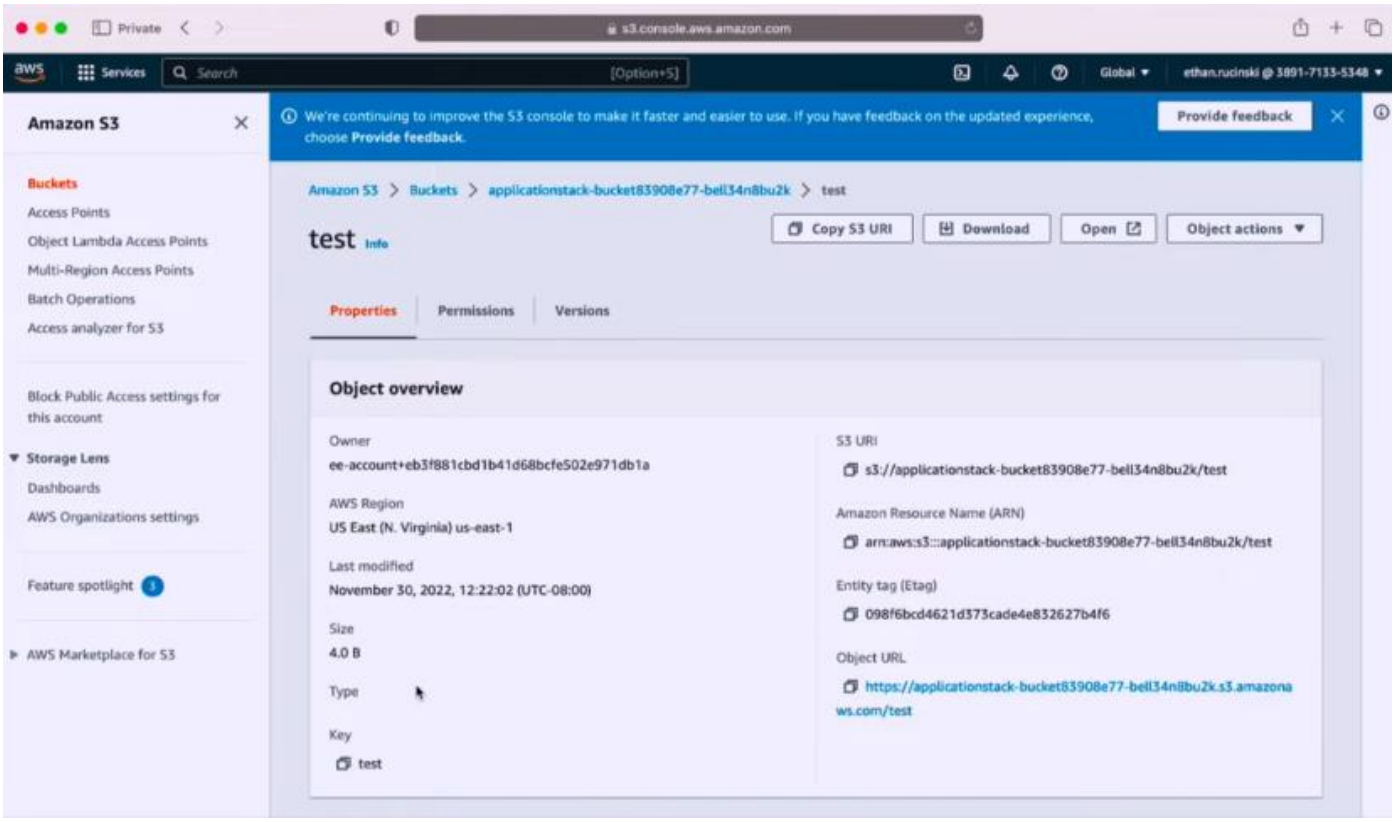

Private
us-east-1.console.aws.amazon.com
aws
Services
Search
[Option+S]
N. Virginia
ethan.rucinski @ 3891-7133-5348
Image
Test
Monitor
Configuration
Aliases
Versions
Execution result: succeeded (logs)
Details
The area below shows the last 4 KB of the execution log.
{
"$metadata": {
"httpStatusCode": 200,
"requestId": "YYD8AXTFZX7Z8Z1A",
"extendedRequestId": "w6QXIN37YyuEsfKbToGIY2e32otgzh9WiN65PqdGd0n1hVZXf0+0ymxUVMBFdVETMYhVsjSqYfA=",
"attempts": 1,
"totalRetryDelay": 0
},
"ETag": "\"098f6bcd4621d373cade4e832627b4f6\""
}
Summary
Code SHA-256
001b9587c16f1d12b693876efe63fd90c9e7e01e3df809c3e9d48b6bd5cdb134
Request ID
c50fd017-3f38-4695-b764-68e98313afcf
Init duration
2550.73 ms
Duration
719.11 ms
Billed duration
3270 ms
Resources configured
128 MB


us-east-1.console.aws.amazon.com
Services
Search
[Option+S]
N. Virginia
ethan.rucinski @ 3891-7133-5348
Lambda > Functions > ApplicationStack-Function76856677-RKvKnnh6VFMH
ApplicationStack-Function76856677-RKvKnnh6VFMH
Throttle
Copy ARN
Actions
This function belongs to an application. Click here to manage it.
Function overview Info
ApplicationStack-Function76856677-RKvKnnh6VFMH
Add trigger
Add destination
Description
-
Last modified
4 minutes ago
Function ARN
arn:aws:lambda:us-east-1:389171335348:function:ApplicationStack-Function76856677-RKvKnnh6VFMH
Application
ApplicationStack
Function URL Info
-


s3.console.aws.amazon.com
Services
s3
N. Virginia
ethan.rucinski @ 3891-7133-5348
Search results for 's3'
Services (8)
Features (16)
Resources New
Blogs (1,158)
Documentation (105,110)
Knowledge Articles (30)
Services
See all 8 results
S3
Scalable Storage in the Cloud
Top features
Buckets
Access points
Batch Operations


s3.console.aws.amazon.com
Services
Search
[Option+S]
Global
ethan.rucinski @ 3891-7133-5348
Amazon S3
Buckets
Access Points
Object Lambda Access Points
Multi-Region Access Points
Batch Operations
Access analyzer for S3
Block Public Access settings for this account
Storage Lens
Dashboards
AWS Organizations settings
Feature spotlight
AWS Marketplace for S3
We're continuing to improve the S3 console to make it faster and easier to use. If you have feedback on the updated experience, choose Provide feedback.
Provide feedback
Replicate data within and between AWS Regions using Amazon S3 Replication.
View tutorial
Amazon S3 > Buckets
Account snapshot
Storage lens provides visibility into storage usage and activity trends. Learn more
View Storage Lens dashboard
Buckets (2) Info
Buckets are containers for data stored in S3. Learn more
Copy ARN
Empty
Delete
Create bucket
Find buckets by name
Name
AWS Region
Access
Creation date
applicationstack-bucket83908e77-bell34n8bu2k
US East (N. Virginia) us-east-1
Objects can be public
November 30, 2022, 12:16:49 (UTC-08:00)
cdk-base-assets-389171335348-us-east-1
US East (N. Virginia) us-east-1
Bucket and objects not public
November 30, 2022, 09:32:33 (UTC-08:00)


s3.console.aws.amazon.com
Services
Search
[Option+S]
Global
ethan.rucinski @ 3891-7133-5348
Amazon S3
We're continuing to improve the S3 console to make it faster and easier to use. If you have feedback on the updated experience, choose Provide feedback.
Provide feedback
Replicate data within and between AWS Regions using Amazon S3 Replication.
View tutorial
Buckets
Access Points
Object Lambda Access Points
Multi-Region Access Points
Batch Operations
Access analyzer for S3
Block Public Access settings for this account
Storage Lens
Dashboards
AWS Organizations settings
Feature spotlight
AWS Marketplace for S3
Amazon S3 > Buckets > applicationstack-bucket83908e77-bell34n8bu2k
applicationstack-bucket83908e77-bell34n8bu2k Info
Objects
Properties
Permissions
Metrics
Management
Access Points
Objects (1)
Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. Learn more
Copy S3 URI
Copy URL
Download
Open
Delete
Actions
Create folder
Upload
Find objects by prefix
Name
Type
Last modified
Size
Storage class
test
November 30, 2022, 12:22:02 (UTC-08:00)
4.0 B
Standard


s3.console.aws.amazon.com
Services
Search
[Option+S]
Global
ethan.rucinski @ 3891-7133-5348
Amazon S3
We're continuing to improve the S3 console to make it faster and easier to use. If you have feedback on the updated experience, choose Provide feedback.
Provide feedback
Buckets
Access Points
Object Lambda Access Points
Multi-Region Access Points
Batch Operations
Access analyzer for S3
Block Public Access settings for this account
Storage Lens
Dashboards
AWS Organizations settings
Feature spotlight
AWS Marketplace for S3
Amazon S3 > Buckets > applicationstack-bucket83908e77-bell34n8bu2k > test
test Info
Copy S3 URI
Download
Open
Object actions
Properties
Permissions
Versions
Object overview
Owner
ee-account+eb3f881cbd1b41d68bcfe502e971db1a
AWS Region
US East (N. Virginia) us-east-1
Last modified
November 30, 2022, 12:22:02 (UTC-08:00)
Size
4.0 B
Type
Key
test
S3 URI
s3://applicationstack-bucket83908e77-bell34n8bu2k/test
Amazon Resource Name (ARN)
arn:aws:s3:::applicationstack-bucket83908e77-bell34n8bu2k/test
Entity tag (Etag)
098f6bcd4621d373cade4e832627b4f6
Object URL
https://applicationstack-bucket83908e77-bell34n8bu2k.s3.amazonaws.com/test

“Good programmers know what to write. Great ones know what to rewrite (and reuse).”

Eric S. Raymond

American software developer, open-source software advocate

CloudFormation Community Extensions