INSY 5345 & INSY 4307

Dr. Santoso Budiman

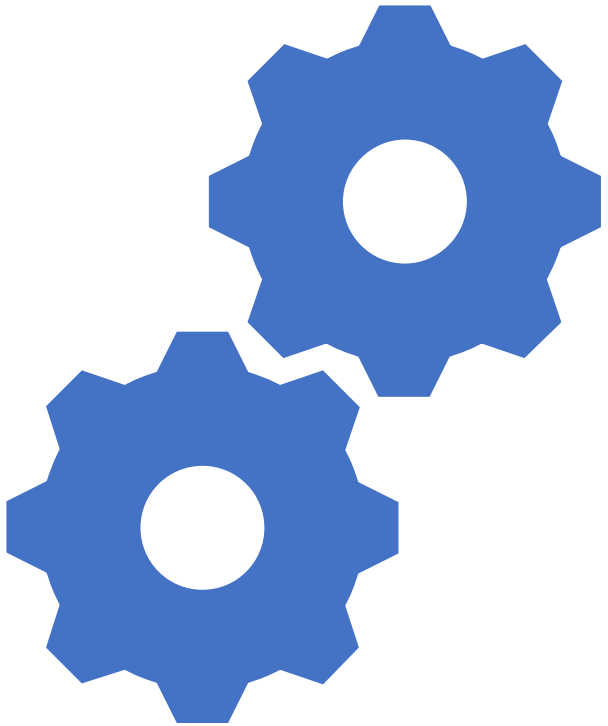
Cloud Computing Theory and Practice

Network Connections

Section 4

Topics

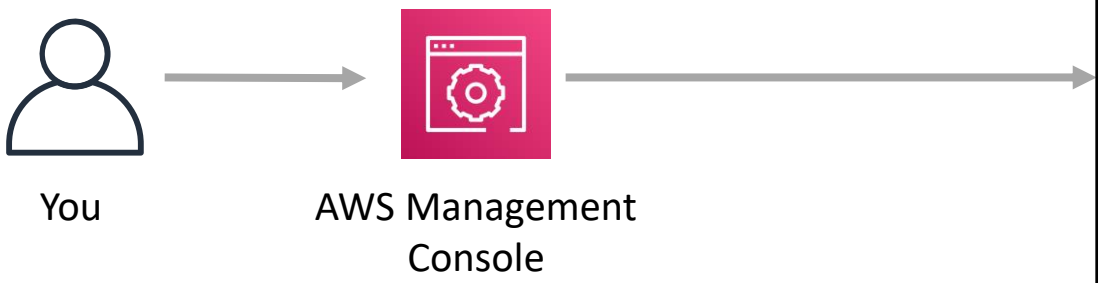
- IaC
  - CloudFormation
- Connections Between Networks
  - VPC Peering
  - Transit Gateway
- Multiple Accounts
- Homework

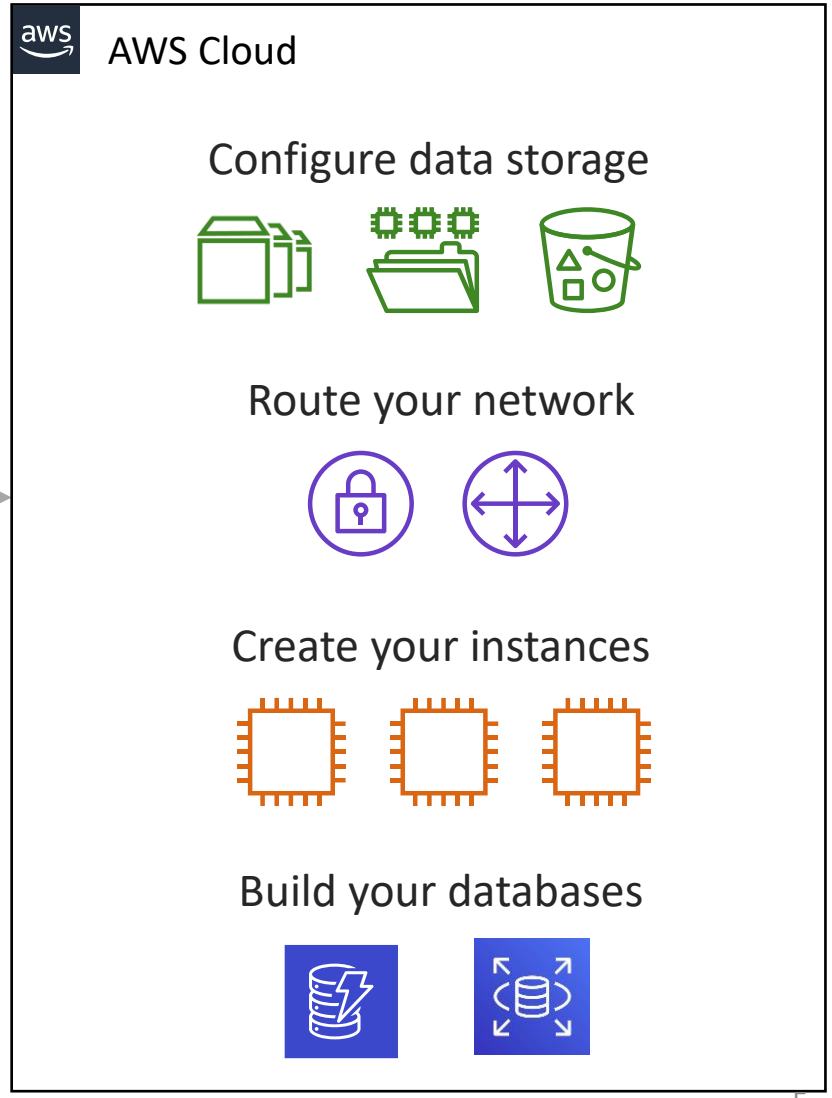


Reasons to automate

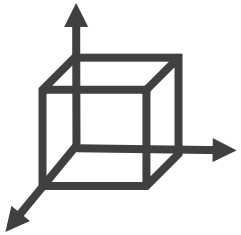
Without automation

Long *manual process* to build an architecture





Risks from manual processes



Does not support repeatability at scale

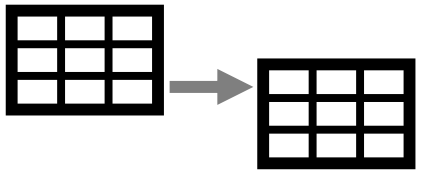
- How will you replicate deployments to multiple Regions?

No version control

- How will you roll back the production environment to a prior version?

Lack of audit trails

- How will you ensure compliance? How will you track changes to configuration details at the resource level?



Inconsistent data management

- For example, how will you ensure matching configurations across multiple Amazon Elastic Compute Cloud (Amazon EC2) instances?

Complying with AWS Well-Architected Framework principles

- Operational excellence design principles
  - Perform operations as code
  - Make frequent, small, reversible changes
- Reliability pillar design principles
  - Manage change in automation

Creating and maintaining AWS resources and deployments by following a manual approach does not enable you to meet these guidelines.

AWS Automation Tools

IaC

CloudFormation & CDK

Automatic Deployment

Systems Manager

Configuration Management (Application Management Service)

OpsWorks

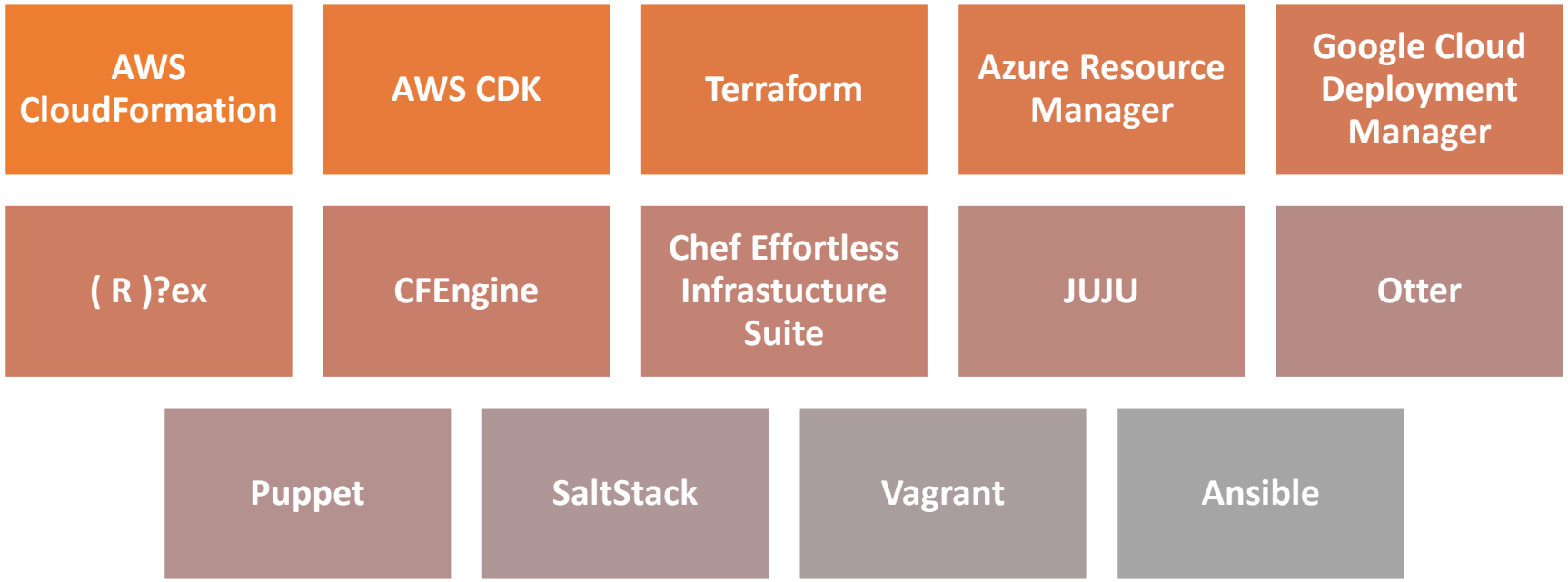
Web Applications deployment

Elastic Beanstalk

Infrastructure as Code (IaC)

- **Infrastructure as code** (**IaC**) is the managing and provisioning of infrastructure through code instead of through manual processes (physical hardware configuration or interactive configuration tools).
- The IT infrastructure managed by this process can comprise of **physical equipment, virtual machines, and Cloud resources**.
- Can be used as a part of CI/CD process.

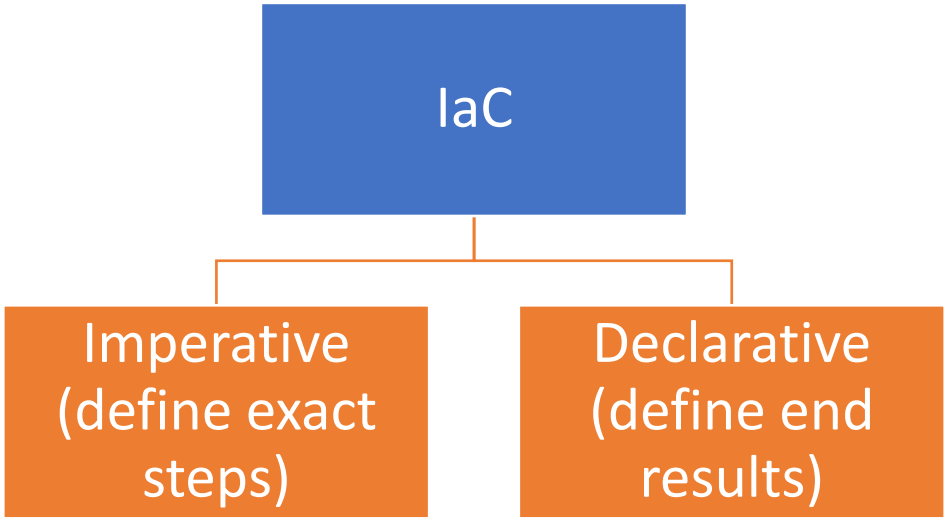
IaC tools



Cloud Native:
AWS CloudFormation & CDK
Azure ARM
Google Deployment Manager

On-premise & cross-platform:
Terraform
Ansible
Cloud Foundry

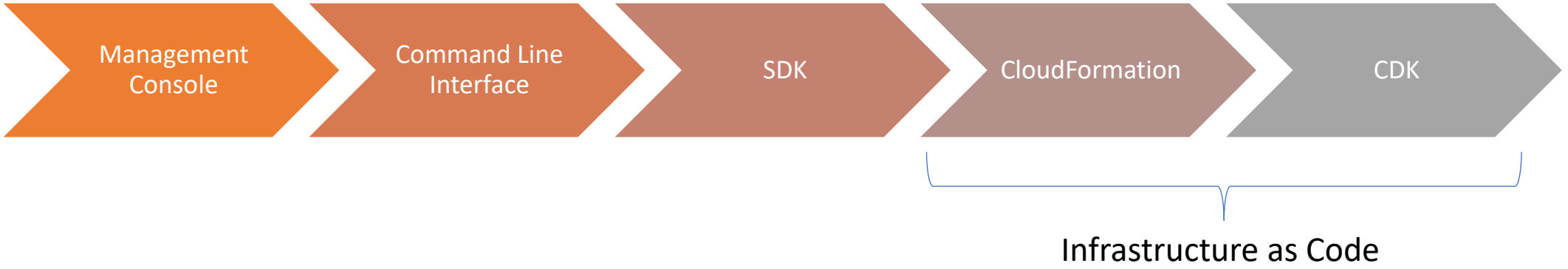
Tool types



Imperative	Declarative
Define exact steps - Each command to run	Define end result -resources and properties
Teardown in steps	Teardown with a single command
	Ex. AWS CloudFormation and Terraform

Managing AWS Resources Lifecycle

Lifecyle: Create, Update, Delete resources



AWS IaC

- IaC has become a main process to automatically provision and manage cloud resources.
- AWS provides 2 IaC options
  - AWS CloudFormation
    - The original AWS IaC
    - **NOTE: eksctl uses CloudFormation but only for EKS**
  - AWS Cloud Development Kit (CDK).
    - an abstraction on top of CloudFormation.
    - enables developers build cloud application in their own IDE.
    - Can use Python, TypeScript, Java, JavaScript, C#
    - AWS CDK can do more than CloudFormation.

AWS CDK Vs. AWS SDK

- **CDK:** Is a framework to model and provision infrastructure or stack.
  - a software development framework for defining cloud infrastructure using familiar programming languages: Python, TypeScript, Java, JavaScript, C#
  - A **stack** is a **collection of AWS resources created** that can be managed as **a single unit.**
- **SDK:** These are the code libraries provided by Amazon in various languages, like Java, Python, PHP, Javascript, Typescript etc.
  - These libraries help interact with AWS services (like creating data in DynamoDB) which you either create through CDK or console.
  - SDK is not an IaC
  - Ex: Boto3 for Python; sdk for Java; etc.

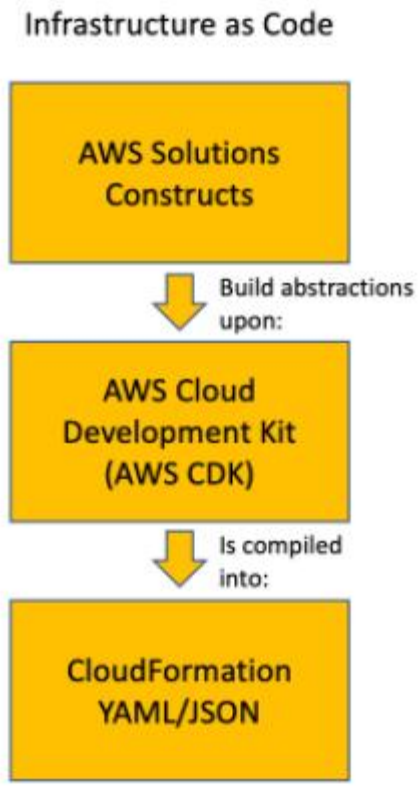
AWS CDK and AWS CloudFormation

AWS CDK

Imperative language: Python, TypeScript, Java, JavaScript, C#

AWS CloudFormation

Declarative language: Json/Yaml



When AWS CDK applications are run, they compile down to fully formed CloudFormation JSON/YAML templates that are then submitted to the CloudFormation service for provisioning.

AWS Cloud Development Kit (CDK) Features

- Constructs for generating AWS infrastructure
- Construct is a cloud component you want to create (one or multiple aws resources)
- Customize, share, and reuse constructs
- Powered by AWS CloudFormation
- Use familiar programming languages, tools, and workflows
- Deploy infrastructure and runtime code together
- Developer-friendly command line interface (CLI)
- https://aws.amazon.com/cdk/features/
- https://www.youtube.com/watch?v=MgfzMTKoQYY

CDK example

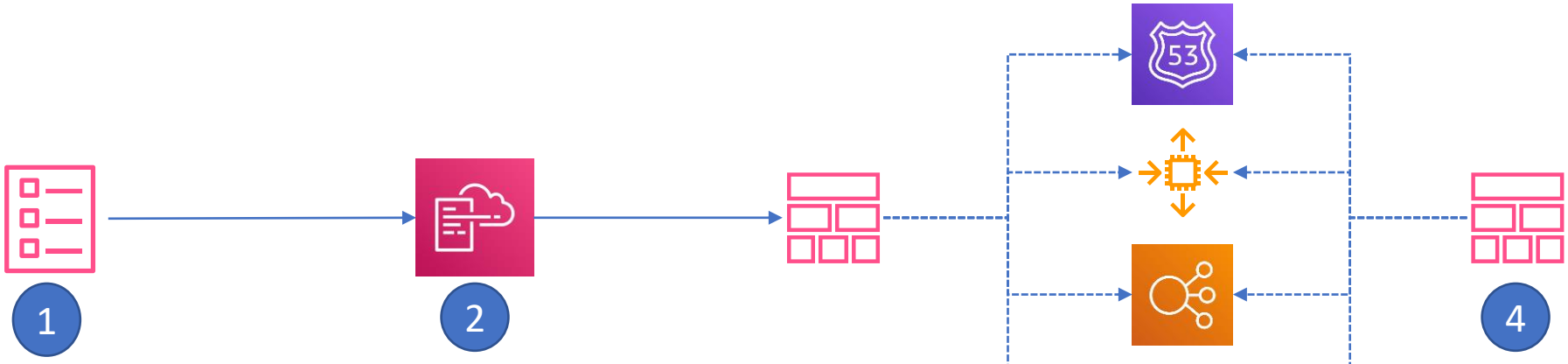
```
import * as cdk from '@aws-cdk/core';
import { FargateDemoStack } from '../lib/fargate';
import { CloudfrontDemoStack } from '../lib/cloudfront';

const app = new cdk.App();

// FargateDemoStack
new FargateDemoStack(app, "FargateDemoStack", {
  env: { account: "123456789", region: "eu-central-1" },
});

// CloudfrontDemoStack
new CloudfrontDemoStack(app, "CloudfrontDemoStack", {
  stage: "prod",
  env: { account: "123456789", region: "eu-central-1" },
});
```

AWS CloudFormation

Automating your infrastructure

AWS CloudFormation

- AWS CloudFormation provides a simplified way to model, create, and manage a collection of AWS resources
  - Collection of resources is called an AWS CloudFormation stack
  - No extra charge (pay only for resources you create)
- Can create, update, and delete stacks
- Enables orderly and predictable provisioning and updating of resources
- Enables version control of AWS resource deployments



Define your resources in a **template** *or* use a pre-built template.

Upload the template to AWS CloudFormation *or* point to a template stored in an S3 bucket.

Run a **create stack** action. Resources are created across multiple services in your AWS account as a running environment.

The stack retains control of the resources that are created. You can later **update stack**, **detect drift** or **delete stack**.

Template - Stack

Template

- An AWS CloudFormation template is a **JSON** or **YAML** formatted text file.
  - can save these files with any extension, such as .json, .yaml, .template, or .txt.
- AWS CloudFormation uses these templates as blueprints for building AWS resources.
  - For example, in a template, can describe an EC2 instance, such as the instance type, the AMI ID, block device mappings, and its Amazon EC2 key pair name.

Stack

- A **stack** is a **collection of AWS resources created** that can be managed as **a single unit**.
  - can create, update, or delete stacks.
- All the **resources in a stack** are **defined** by the stack's **AWS CloudFormation template**.
- A stack, for instance, can include all the resources required to run a web application, such as a web server, a database, and networking rules.
- If a stack is deleted, all of its related resources are deleted.

YAML

- **YAML** = YAML Ain't Markup Language
  - a human-readable **data serialization standard (to transfer data)** that can be used in conjunction with programming languages and is often used to write **configuration files**.
  - object-based data format
    - Key-value pairs (Hash) – string, Boolean, int, float, list, date and time (ISO 8601)
  - intended to be read and written in streams
  - Document extension: .yml or .yaml (.yaml is preferred)
  - use cases:
    - configuration files,
    - messages between applications, and
    - saving application state
  - Editor (there are others) – use this to check https://onlineyamltools.com/edit-yaml
- https://yaml.org/
- https://onlineyamltools.com/highlight-yaml
- https://yaml.org/spec/1.2/spec.html
- YAML - Basics (tutorialspoint.com)
- https://docs.ansible.com/ansible/latest/reference_appendices/YAMLSyntax.html

YAML Syntax

https://yaml.org/spec/1.2/spec.html

YAML's block collections use **indentation for scope** and begin each entry on its own line.

- **Don't use tab** – use spaces (suggested 2 but YAML will follow consistent spacing)
- Must align

Mappings use a **colon and space** (": ") to mark each key: value pair (**must have space after :**).

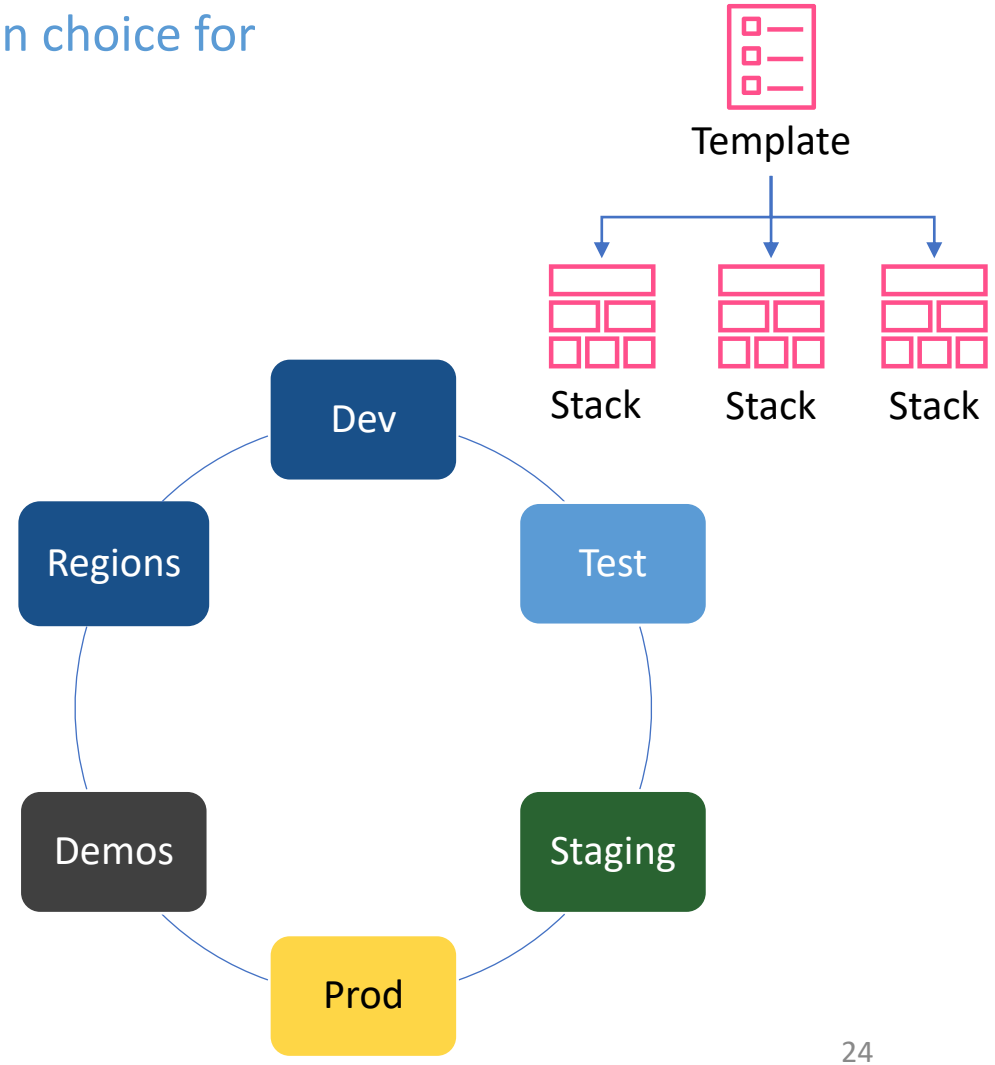
Case sensitive.

Block sequences indicate each entry with a **dash and space** ("- ").

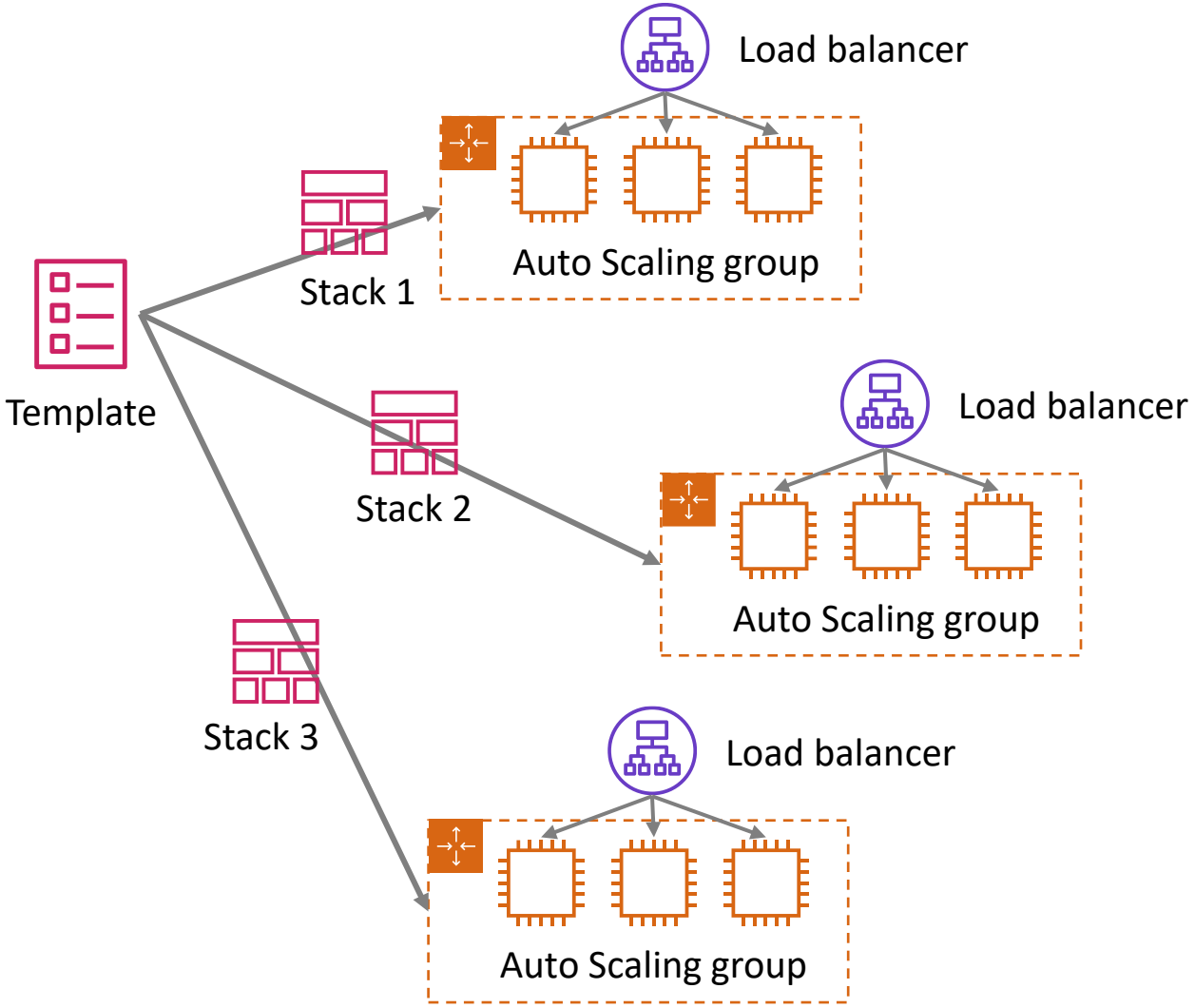
Comments begin "#".

For AWS Cloud development, the built-in choice for IaC is AWS CloudFormation.

- IaC is the process of provisioning and managing your cloud resources by writing a template file that is –
  - Human readable
  - Machine consumable
- It is infrastructure you can replicate, re-deploy, re-purpose
- You can roll back to the last good state on failures



Infrastructure as code: Benefits



Reduce multiple matching environments

- Rapid deployment of complex environments
- Provides configuration consistency
- Simple clean up when wanted (deleting the stack deletes the resources created)
- Easy to propagate a change to all stacks
  - Modify the template, run update stack on all stacks

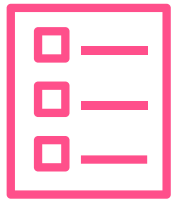
Benefits

- Reusability
- Repeatability
- Maintainability

AWS CloudFormation template syntax

AWS CloudFormation templates

- Author in JavaScript Object Notation (JSON) or YAML Ain't Markup Language (YAML) – **YAML is more common**.
- YAML advantages –
  - Less verbose (no {},"", characters)
  - Supports embedded comments
- JSON advantages –
  - More widely used by other computer systems (for example, APIs)
- Recommendation – Treat templates as source code
  - Store them in a code repository



```
{
 "AWSTemplateFormatVersion": "2010-09-09",
 "Resources" : {
 "awsexamplebucket1" : {
 "Type" : "AWS::S3::Bucket"
 }
 }
}
```

JSON example

```
AWSTemplateFormatVersion: 2010-09-09
Resources:
 awsexamplebucket1:
 Type: AWS::S3::Bucket
```

YAML example

Templates can also be authored in the AWS CloudFormation Designer—a graphical design interface in the AWS Management Console.
https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/template-reference.html

Simple template: Create an EC2 instance

```
{
 "AWSTemplateFormatVersion": "2010-09-09",
 "Description": "Create EC2 instance",
 "Parameters": {
 "KeyPair": {
 "Description": "SSH Key Pair",
 "Type": "String"}},
 "Resources": {
 "Ec2Instance": {
 "Type": "AWS::EC2::Instance",
 "Properties": {
 "ImageId": "ami-9d23aeea",
 "InstanceType": "m3.medium",
 "KeyName": {"Ref": "KeyPair}

       }},
 "Outputs": {
 "InstanceId": {
 "Description": "InstanceId",
 "Value": {"Ref": "Ec2Instance"}
 }
 }
}
```

Parameters (Optional) – Specify what values can be set at runtime when you create the stack

- Example uses: Region-specific settings, or production versus test environment settings

Resources – Define what needs to be created in the AWS account

- Example: Create all components of a virtual private cloud (VPC) in a Region, and then create EC2 instances in the VPC
- Can reference parameters

Outputs (Optional) – Specify values returned after the stack is created

- Example use: Return the instanceId or the public IP address of an EC2 instance

Parameters

- An optional section to input custom values to create/update a stack.
- https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/parameters-section-structure.html

```
Parameters:
  InstanceTypeParameter:
    Type: String
    Default: t2.micro
    AllowedValues:
      - t2.micro
      - m1.small
      - m1.large
    Description: Enter t2.micro, m1.small, or m1.large. Default is t2.micro.
```

Referencing a parameter within a template

```
Ec2Instance:
  Type: AWS::EC2::Instance
  Properties:
    InstanceType:
      Ref: InstanceTypeParameter
    ImageId: ami-0ff8a91507f77f867
```

Resources

```
Resources:
  Logical ID:
    Type: Resource type
    Properties:
      Set of properties
```

https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/resources-section-structure.html

- Logical ID must be alphanumeric (A-Za-z0-9) and unique within the template.
  - Use the logical name to reference the resource in other parts of the template.
- Physical ID is the actual assigned name for that resource, such as an EC2 instance ID or an S3 bucket name.
  - Use the physical IDs to identify resources outside of AWS CloudFormation templates after the resources have been created.

Resource Attributes

- Type
- Properties
- DependsOn
- CreationPolicy
- DeletionPolicy
- UpdatePolicy
- UpdateReplacePolicy

https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/aws-template-resource-type-ref.html

https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/aws-product-attribute-reference.html

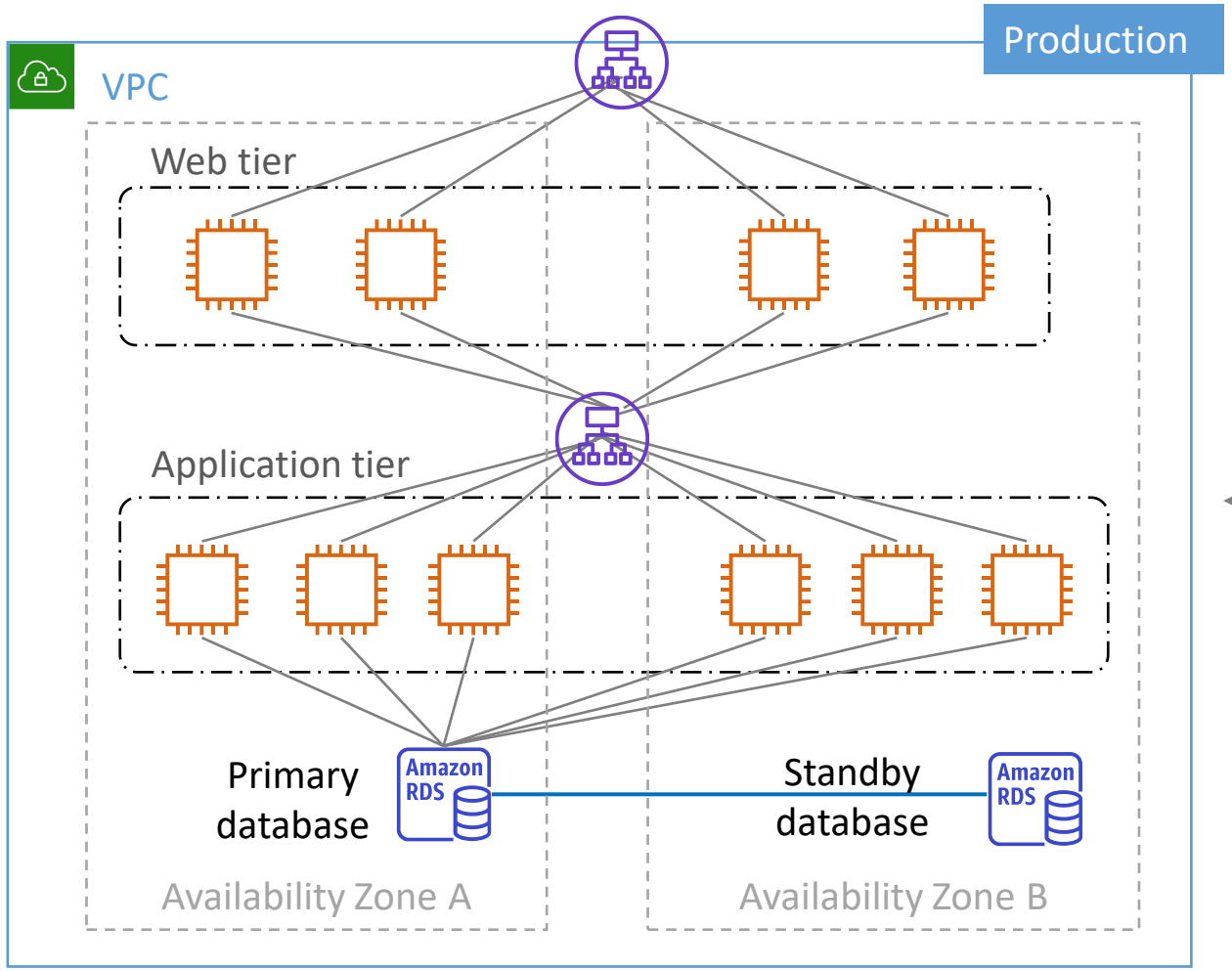
CloudFormation Resource Creation Order

- Independent resources are created in parallel.
- Implicitly dependent resources – (Using the Ref function, the referenced resource will be created before the referencing one).
- Explicit dependency using the DependsOn attribute.

  In the below example, the SNS Topic resource will be created after the EC2 Instance.

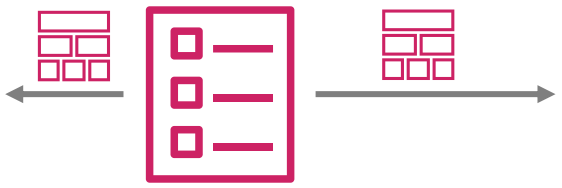
```
SNSTopic:
  Type: AWS::SNS::Topic
  DependsOn: EC2Instance
```

AWS CloudFormation

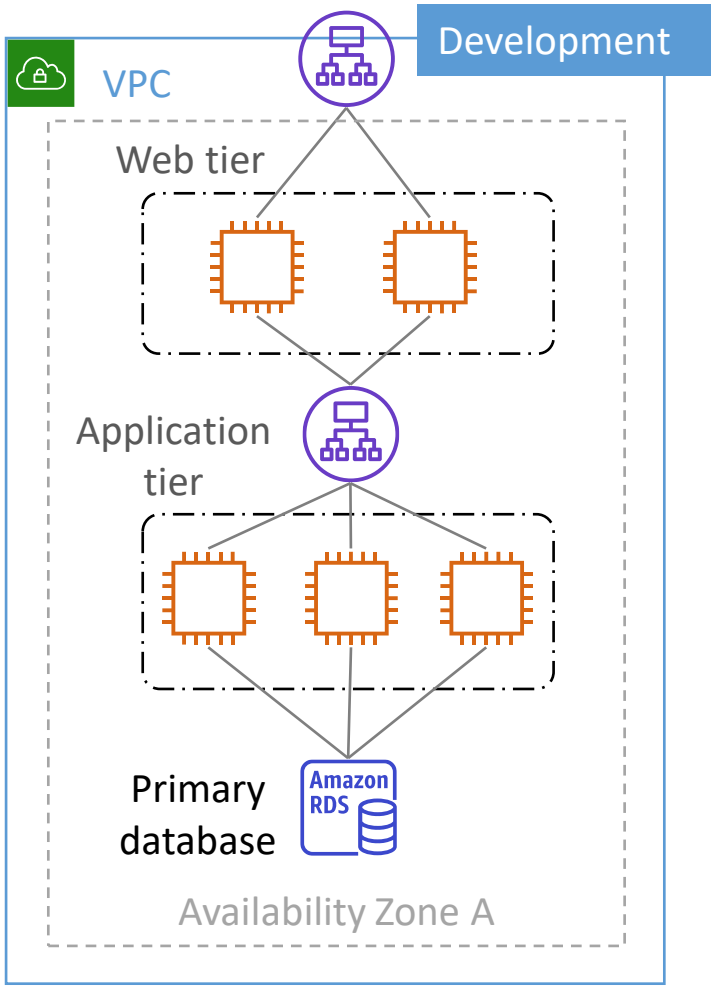


Need to update a configuration?

Edit the template details, then update the stacks.

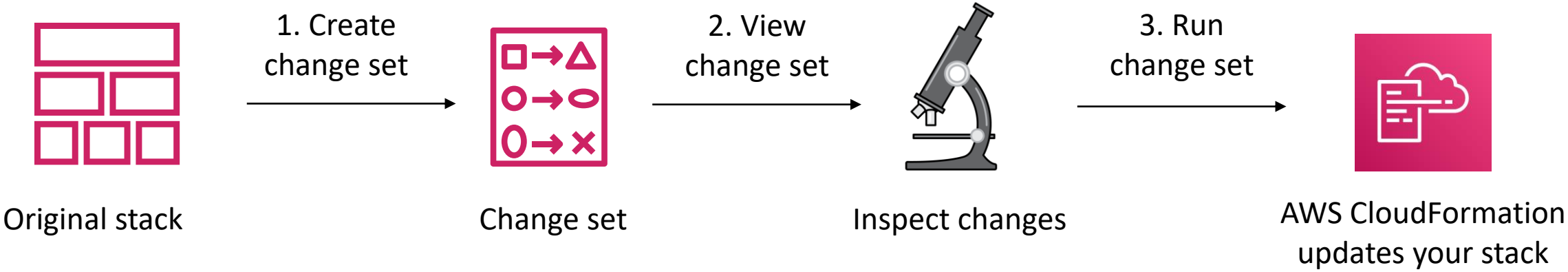


A single AWS CloudFormation template with defined **conditions** can deploy a different application infrastructure to different environments.



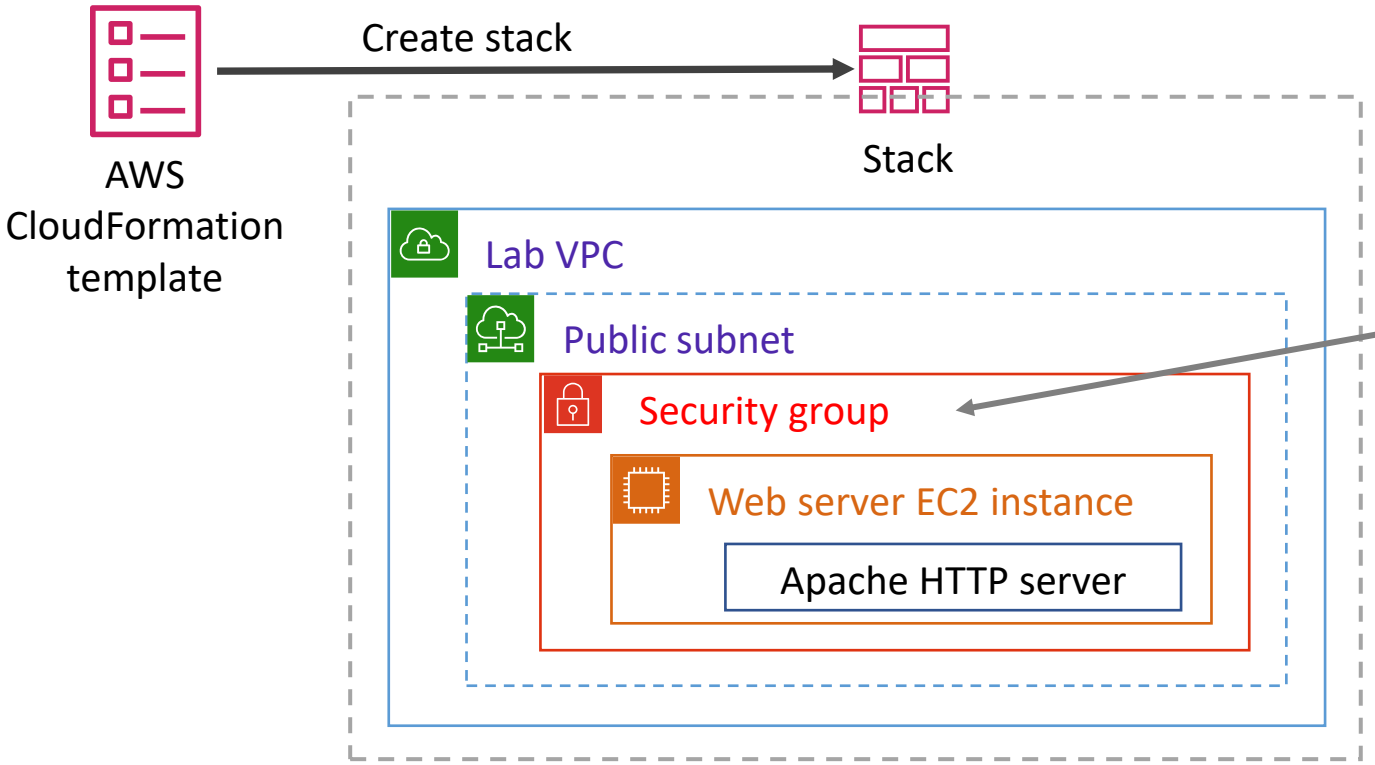
AWS CloudFormation change sets

Change sets enable you to preview changes before you implement them.



Use the DeletionPolicy attribute to preserve or backup a resource when its stack is deleted or updated.

Drift detection



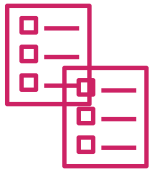
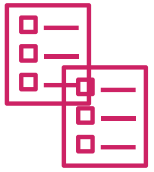
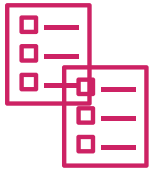
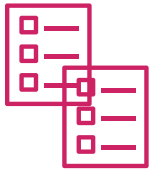
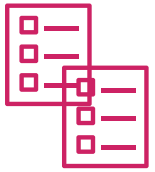
Scenario:

1. An application environment is created by an AWS CloudFormation stack.
2. Later, someone ***manually modifies*** **the security group** and opens a new inbound TCP port.
3. Drift detection is run on the stack.
4. All resources except the security group show the result IN_SYNC, but the security group shows a status of MODIFIED, with details.

Question: In this scenario, what would be a better approach if the team wants to modify the security group setting?

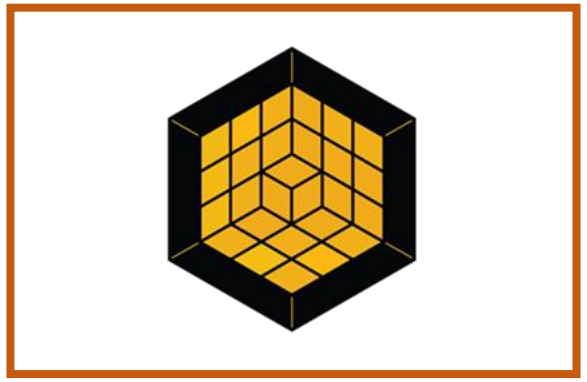
Answer: Modify the AWS CloudFormation template security group settings. Then, run Update Stack. AWS CloudFormation will update the security group. Keeps the *model* deployment synchronized with the actual deployment.

Scoping and organizing templates

Frontend services		Web interfaces, mobile access, analytics dashboard
Backend services		Search, payments, reviews, recommendations
Shared services		Customer relationship management (CRM) databases, common monitoring, alarms, subnets, security groups
Network		VPCs, internet gateways, virtual private networks (VPNs), Network Address Translation (NAT) devices
Security		AWS Identity and Access Management (IAM) policies, users, groups, and roles

AWS Quick Starts

AWS Quick Starts

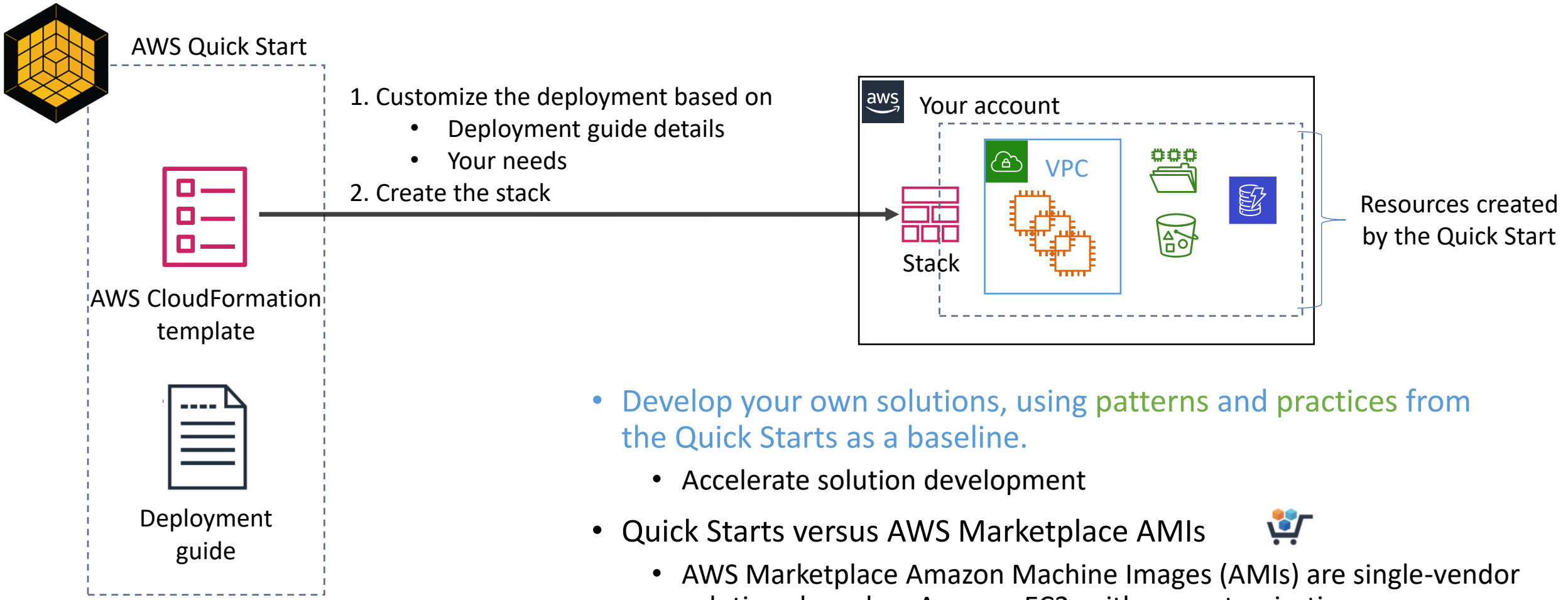


AWS CloudFormation templates built by AWS solutions architects

- Are gold-standard deployments
- Are based on AWS best practices for security and high availability
- Can be used create entire architectures with one click in less than an hour
- Can be used for experimentation and as the basis for your own architectures

https://aws.amazon.com/quickstart/?solutions-all.sort-by=item.additionalFields.sortDate&solutions-all.sort-order=desc&awsf.filter-content-type=*all&awsf.filter-tech-category=*all&awsf.filter-industry=*all

Using AWS Quick Starts



- Develop your own solutions, using patterns and practices from the Quick Starts as a baseline.
  - Accelerate solution development
- Quick Starts versus AWS Marketplace AMIs
  - AWS Marketplace Amazon Machine Images (AMIs) are single-vendor solutions based on Amazon EC2, with no customization
  - Quick Starts are modular and customizable

Cross Stack Reference

To reference resources from a different stack:

- use the export field to flag the value of a resource output for export.
- use the Fn::ImportValue intrinsic function to import the value in any stack within the same AWS Region and account.
- AWS CloudFormation identifies exported values by the names specified in the template.
  - These names must be unique to your AWS Region and account.

- https://aws.amazon.com/id/premiumsupport/knowledge-center/cloudformation-reference-resource/#:~:text=To%20create%20a%20cross%2Dstack%20reference%2C%20use%20the%20export%20field,names%20specified%20in%20the%20template.

CloudFormation – Outputs section

- The optional Outputs section declares output values (Values exported from the created stack) to:
  - Values exported from the created stack
  - import into other stacks (to create cross-stack references),
  - return in response (to describe stack calls), or
  - view on the AWS CloudFormation console.

```
Outputs:
  Logical ID:
    Description: Information about the value
    Value: Value to return
    Export:
      Name: Name of resource to export
```

Intrinsic function reference (Fn:: xxx)

CloudFormation provides several built-in functions to manage stacks

- https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/intrinsic-function-reference.html

Some intrinsic functions:

- Fn::ImportValue returns the value of an output exported by another stack.
  - https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/intrinsic-function-reference-importvalue.html
- Fn::Sub substitutes variables in an input string with values that you specify.
  - YAML
    - full function name Fn::Sub:
    - Short form !Sub
  - A string with variables that AWS CloudFormation substitutes with their associated values at runtime (${MyVarName}).
  - https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/intrinsic-function-reference-sub.html
- Ref returns the value of the specified **parameter** or **resource**.

- AWS resource and property types reference
- https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/aws-template-resource-type-ref.html
- Intrinsic Functions Fn:: (!xx)
- https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/intrinsic-function-reference.html
- Pseudo Parameters (parameters that are predefined by AWS CloudFormation, such as AWS::Stackname)
- https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/pseudo-parameter-reference.html

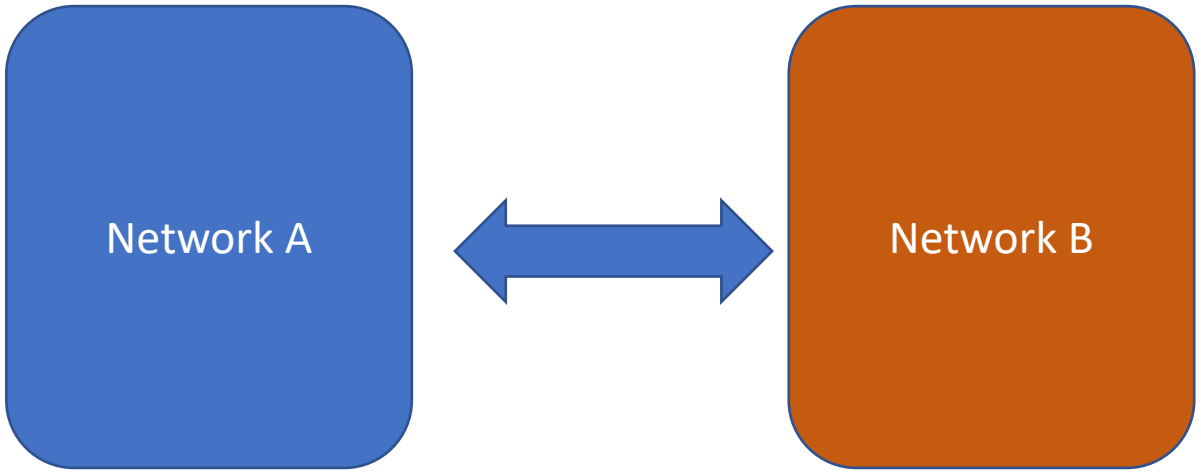
CloudFormation key takeaways

- AWS CloudFormation is an infrastructure as code (IaC) service that enables you to model, create, and manage a collection of AWS resources
- AWS CloudFormation IaC is defined in templates that are authored in JSON or YAML
- A stack is what you create when you use a template to create AWS resources
- Actions that are available on an existing stack include update stack, detect drift, and delete stack
- AWS Quick Starts provide AWS CloudFormation templates that are built by solutions architects that that reflect AWS best practices
- Template Resource Types
  - Format: *service-provider::service-name::data-type-name*
  - Ex: Type: AWS::EC2::Instance
  - https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/aws-template-resource-type-ref.html

Connections Between Networks

Connectivity Between Networks

- Amazon VPC provides multiple network connectivity options depending on user's current network designs and requirements.
- https://docs.aws.amazon.com/whitepapers/latest/aws-vpc-connectivity-options/introduction.html



- **As if** in the **same network**
- **Each host must have unique private IP address**
- Can be on-premise and VPC/ VPC and VPC

AWS Options

- Network to Amazon VPC connectivity options
  - On-premise network to AWS VPC
- Amazon VPC to Amazon VPC
  - AWS VPC to AWS VPC
- Software remote access to Amazon VPC connectivity
  - Remote user to AWS VPC
- Transit VPC option
  - a common strategy for connecting multiple, geographically disperse VPCs and remote networks in order to create a global network transit center

Connecting to your remote network with AWS Site-to-Site VPN

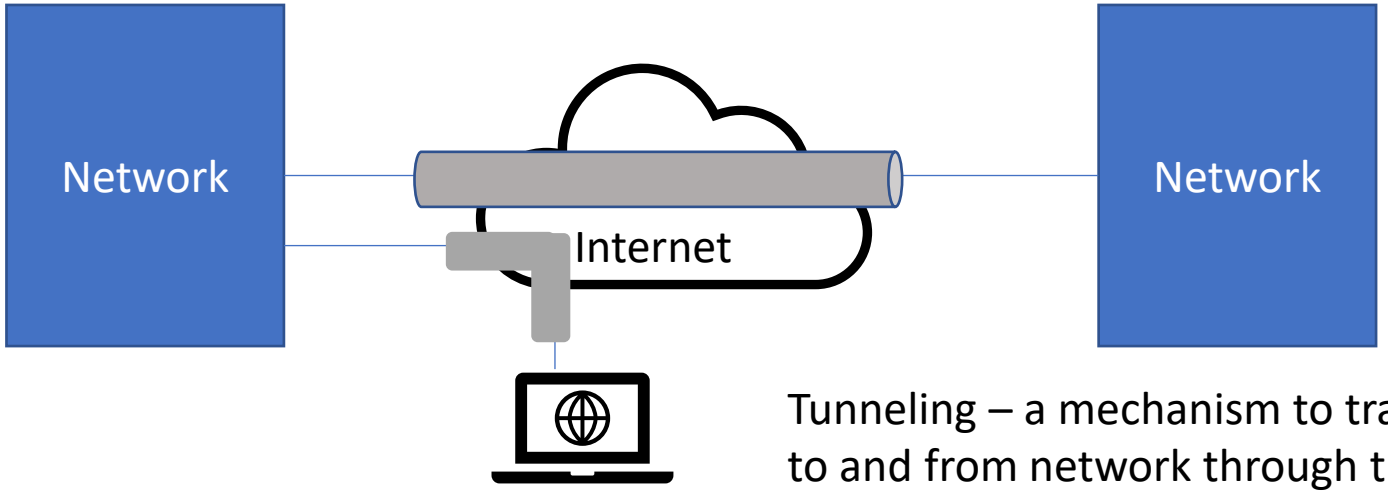
Virtual Private Network (VPN)

Virtual Private Network **(VPN) -** a **private network across the internet**, it enables users to exchange data across shared or public networks as if they are in the same network.

- Security is provided by encapsulating packets

VPN main types:

- **Remote Access** - connects a user to a secure remote server in order to access a private network.
  - For remote workers
- **Site-to-site** - a connection set up between multiple networks.



Tunneling – a mechanism to transfer data securely to and from network through the public internet.

Remote Access VPN

- User starts the VPN software and login to the VPN Gateway
- A VPN interface is created

- VPN Gateway assigns an IP address to the VPN interface (DHCP)
- Establish the tunnel

Site-To-Site VPN

IPSec VPN Vs SSL/TLS VPN

- **Internet Protocol Security** (IPsec) is a suite of secure network protocols that authenticates and encrypts data packets to provide secure communication between computers over the internet.
  - Typically used for Site-to-site VPN
  - OSI Layer 3 (network layer)
  - Can also be used for Remote Access (but TLS is more common)
- **Transport Layer Security (TLS)** is a cryptographic protocol designed to provide communications security over a computer network.
  - Typically used for Remote Access VPN
  - OSI Layer 4-7
  - Also used in HTTPS

AWS Site-to-Site VPN

AWS
Site-to-Site VPN

AWS Site-to-Site is a highly available solution that enables you to securely connect your on-premises network or branch office site to your VPC.

- Uses internet protocol security (IPSec) communications to create encrypted virtual private network (VPN) tunnels
- Provides two encrypted tunnels per VPN connection
- Charged per VPN connection-hour

Static and dynamic routing

Static routing

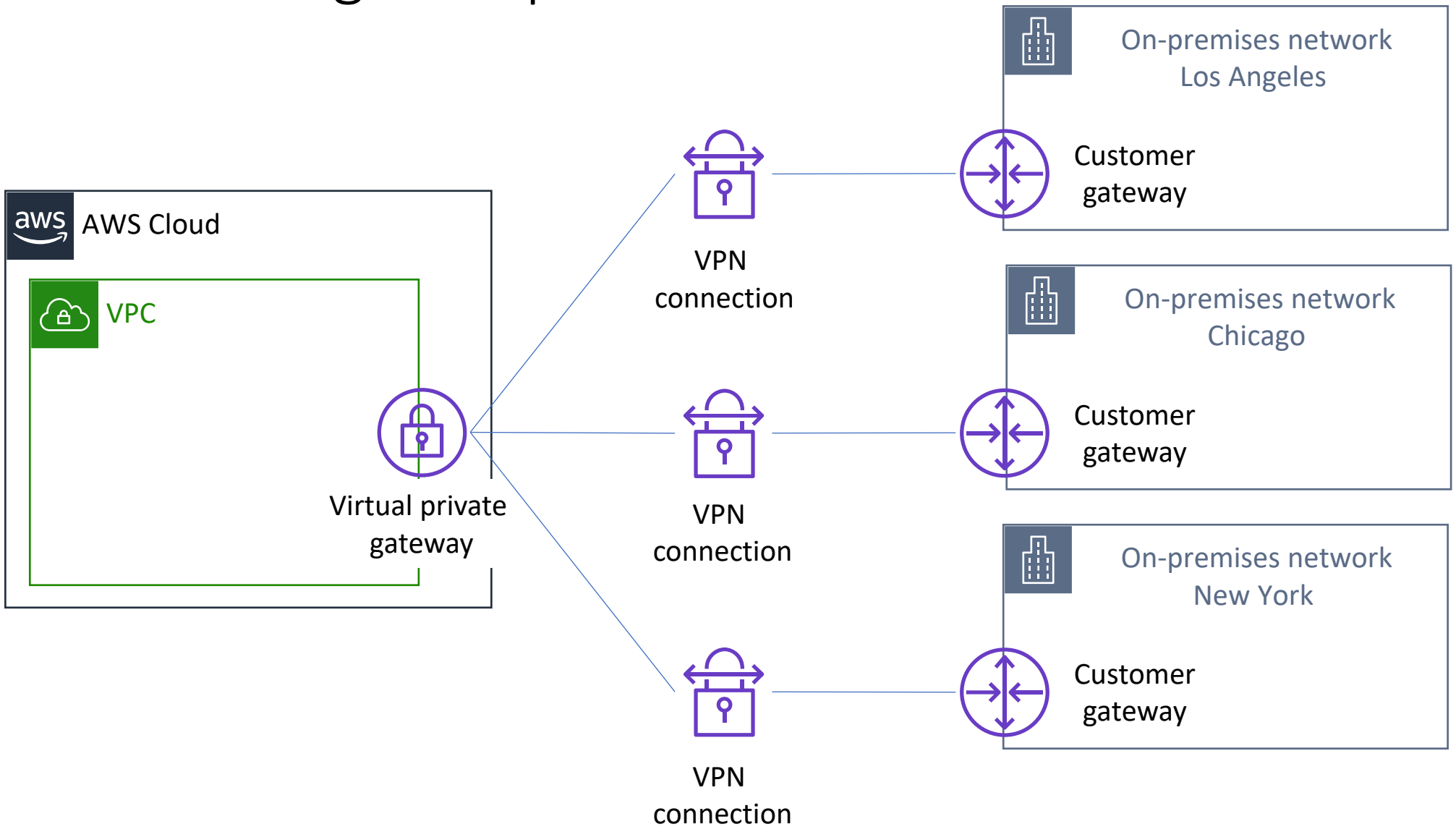
- Requires you to specify all routes (IP prefixes)
- Specify *static routing* if your customer gateway device does not support BGP

Dynamic routing

- Uses the Border Gateway Protocol (BGP) to advertise its routes to the virtual private gateway
- Specify *dynamic routing* if your customer gateway device supports BGP*

*We recommend that you use BGP-capable devices because the BGP protocol offers robust liveness detection checks.

Connecting multiple VPNs



Site-to-site VPN key takeaways

- AWS Site-to-Site VPN is a highly available solution that enables you to securely connect your on-premises network or branch office site to your VPC
- AWS Site-to-Site VPN supports both static and dynamic routing
- You can establish multiple VPN connections from multiple customer gateway devices to a single virtual private gateway

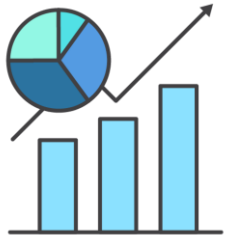
Connecting to your remote network with AWS Direct Connect

AWS Direct Connect (DX)

AWS Direct Connect

AWS Direct Connect (which is also known as DX) provides you with a dedicated, private network connection capacity of either 1 Gbps or 10 Gbps.

Reduces data transfer costs



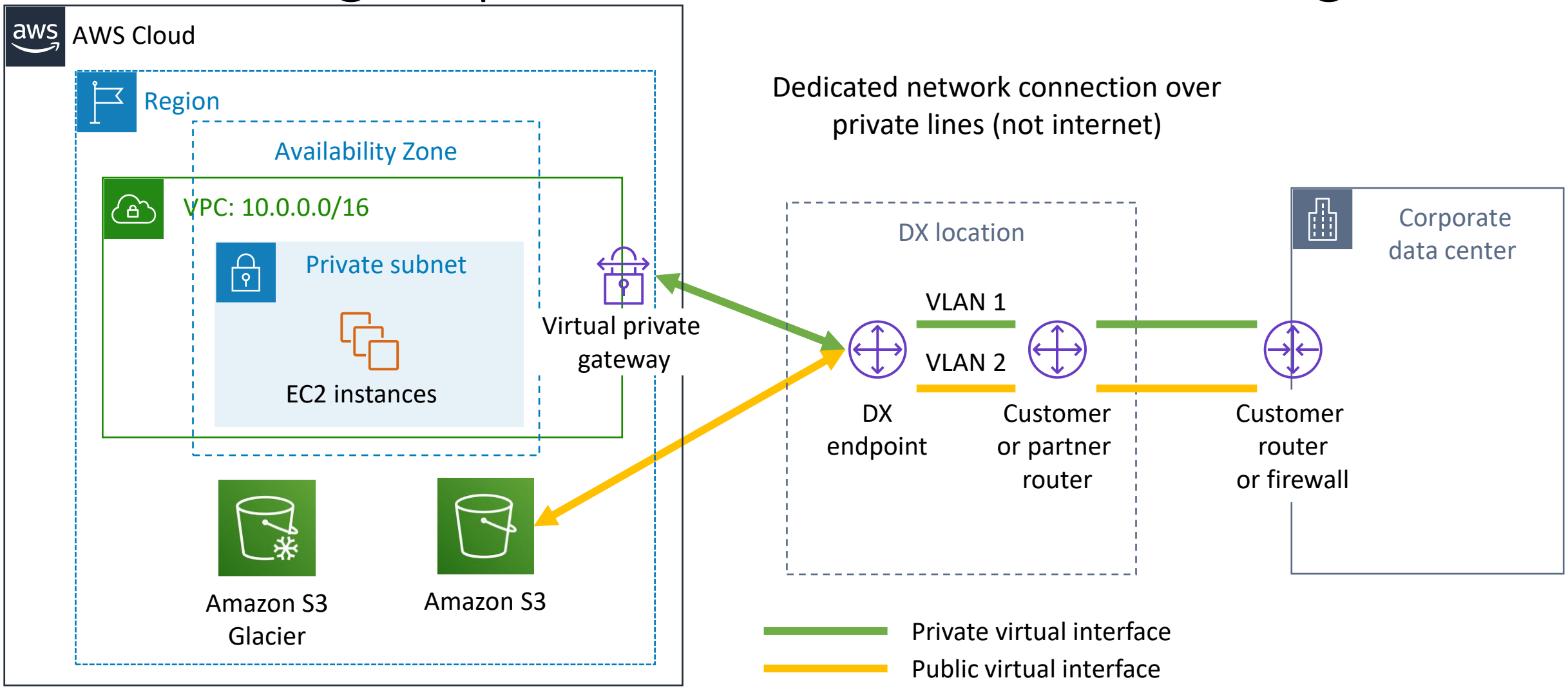
Improves application performance with predictable metrics

DX use cases

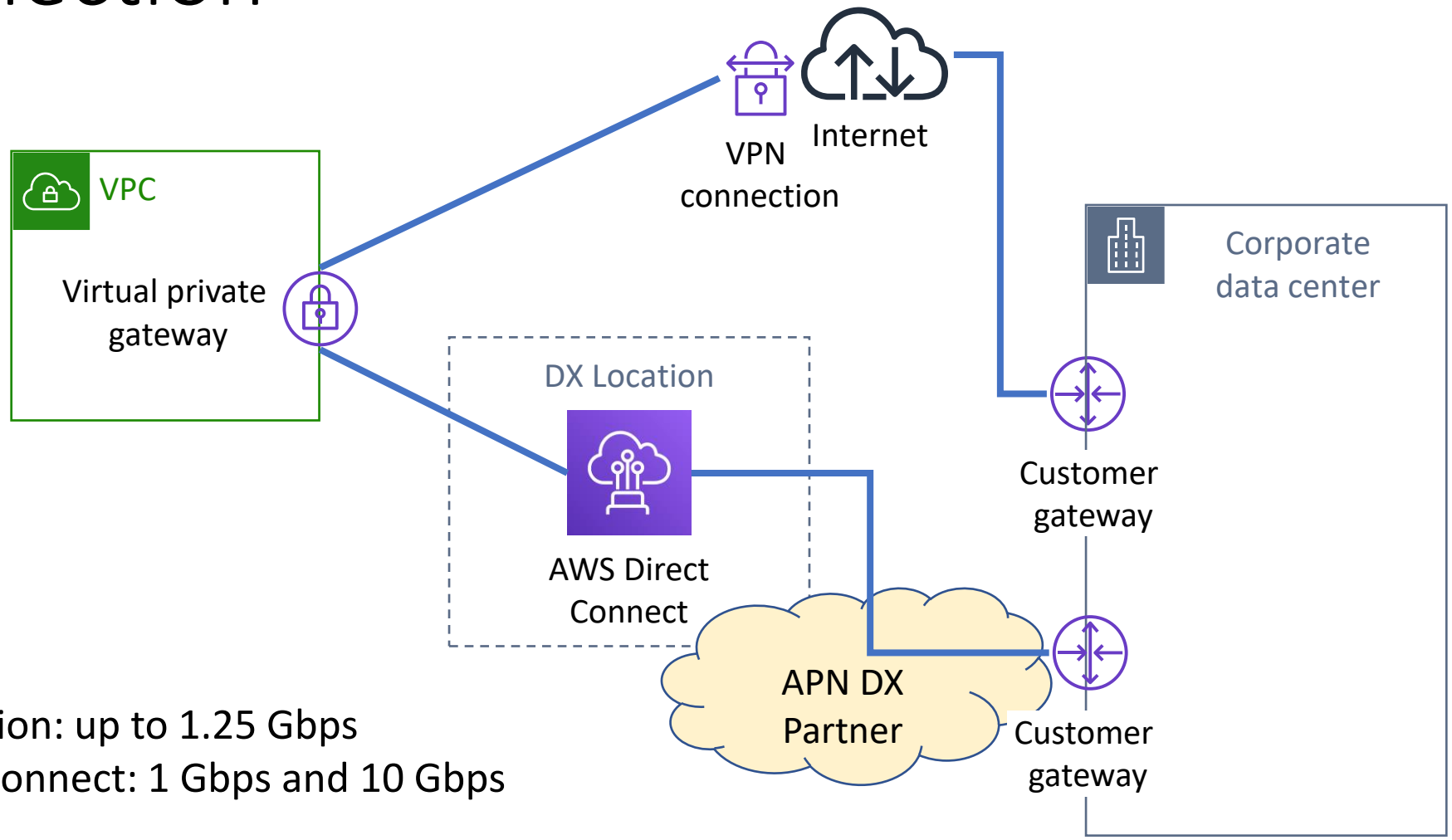
AWS Direct Connect

- Hybrid environments
- Transferring large datasets
- Network performance predictability
- Security and compliance

Extending on-premises network to AWS using DX

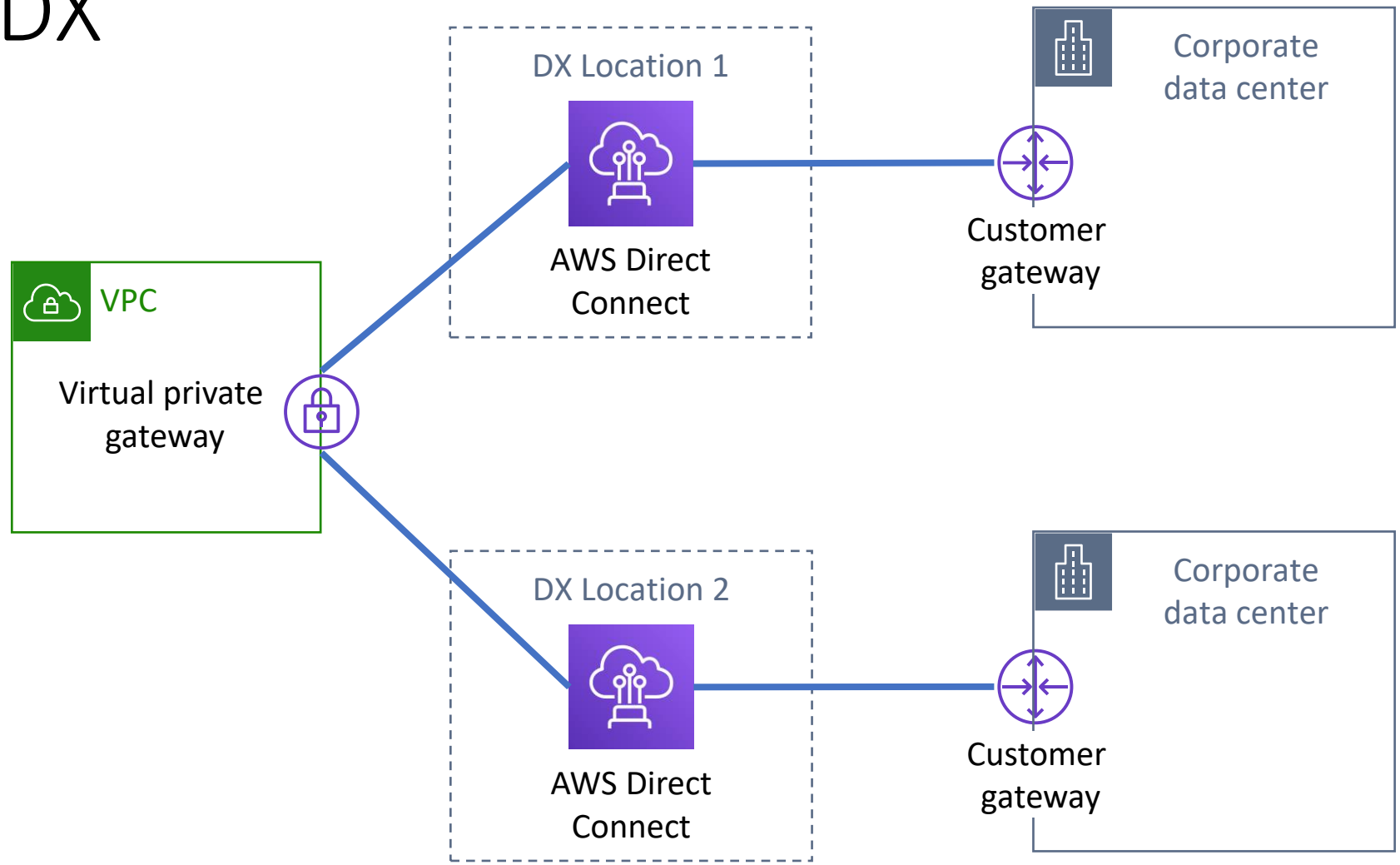


Enabling high availability: DX with backup VPN connection



- VPN connection: up to 1.25 Gbps
- AWS Direct Connect: 1 Gbps and 10 Gbps

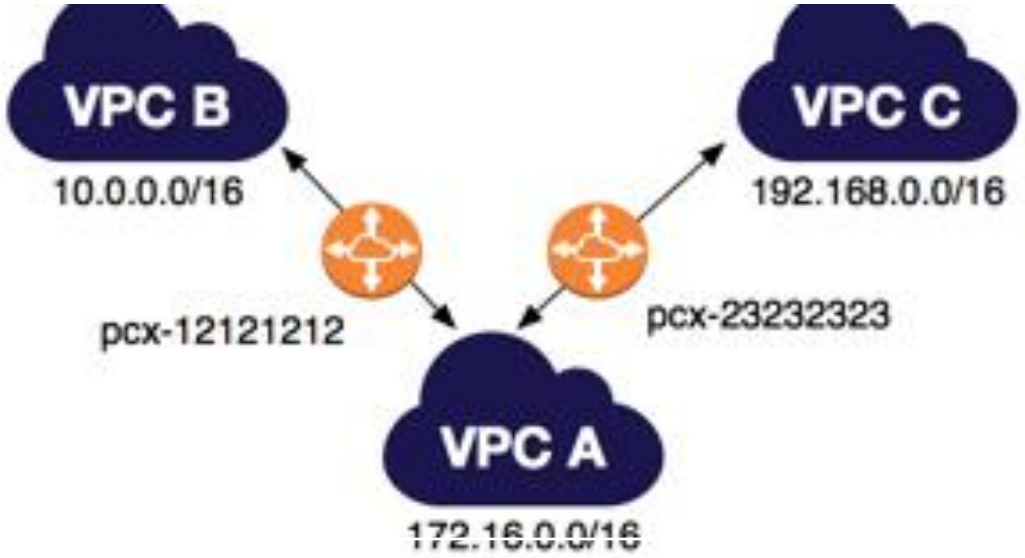
Enabling high resiliency for critical workloads with DX

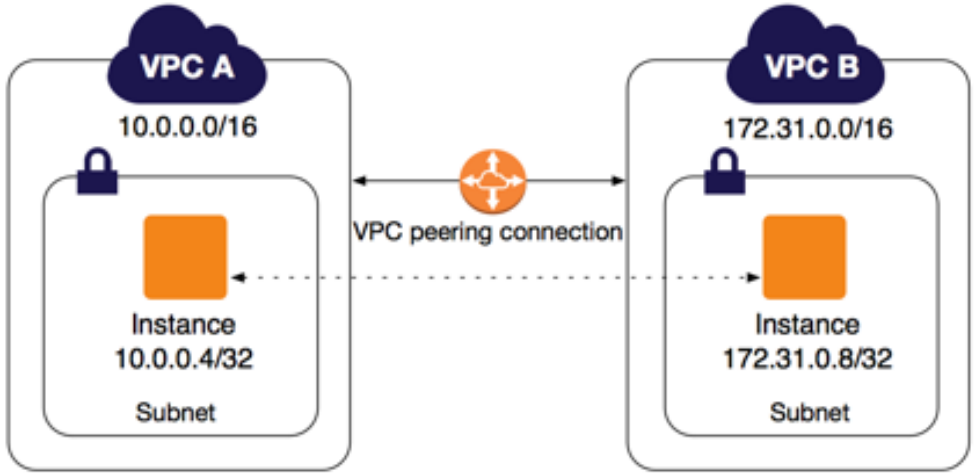


Direct Connect key takeaways

- AWS Direct Connect uses open standard 802.1q VLANs that enable you to establish a dedicated, private network connection from your premises to AWS
- You can access any VPC or public AWS service in any Region (except China) from any supported DX location
- You can implement highly available connectivity between your data centers and your VPC by coupling one or more DX connections that you use for primary connectivity with a lower-cost, backup VPN connection
- To implement a highly resilient, fault-tolerant architecture, connect to your AWS network from multiple data centers so you can have physical location redundancy

Connecting VPCs in AWS with VPC peering

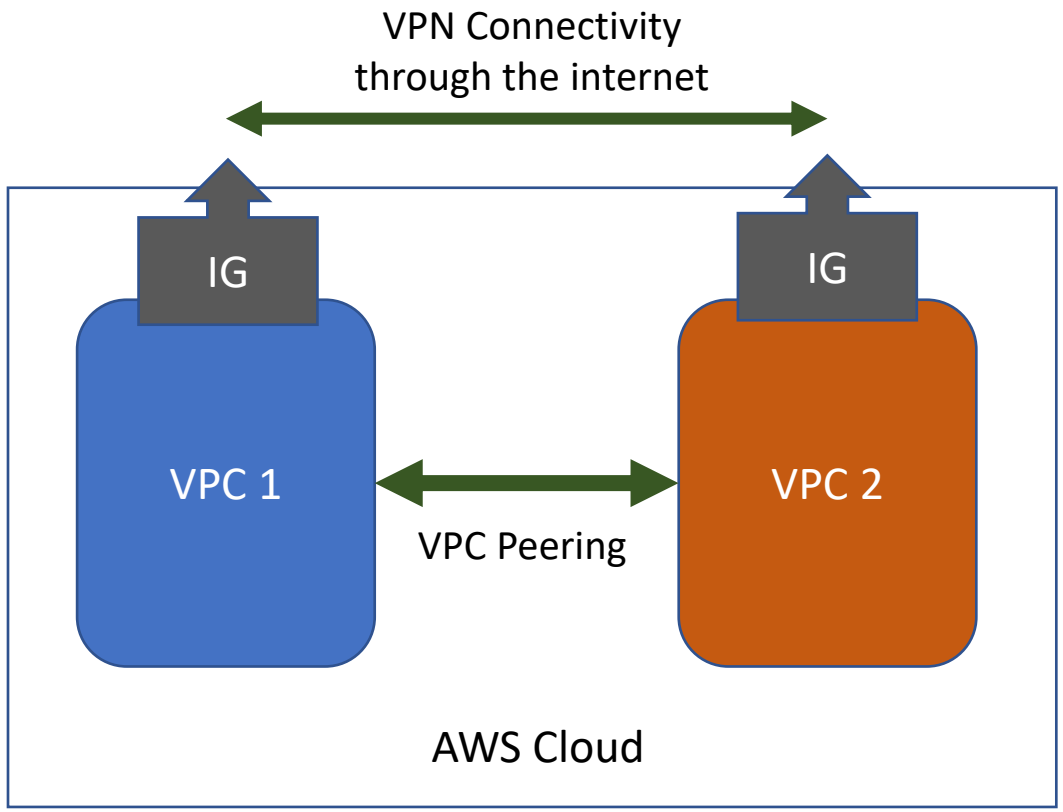




Amazon VPC-to-Amazon VPC connectivity options – VPC Peering

- Only 1 VPC to another VPC connection
- Internal AWS infrastructure

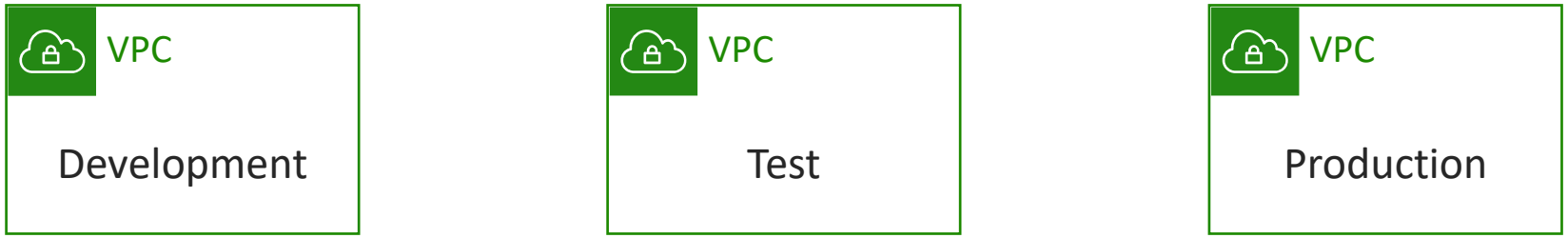
VPN Connectivity Vs. VPC Peering



- **VPN -** a private network across **the internet**, it enables users to exchange data across shared or public networks.
  - VPN connectivity through internet
  - Latency
  - Hardware throughput limitations (such as Internet Gateway)
    - AWS site-to site VPN
    - Third VPN software
    - CloudHub
- VPC Peering:
  - Use AWS Backbone
  - No bandwidth limitation
  - Cross account
  - Limited inter-region

Connecting VPCs

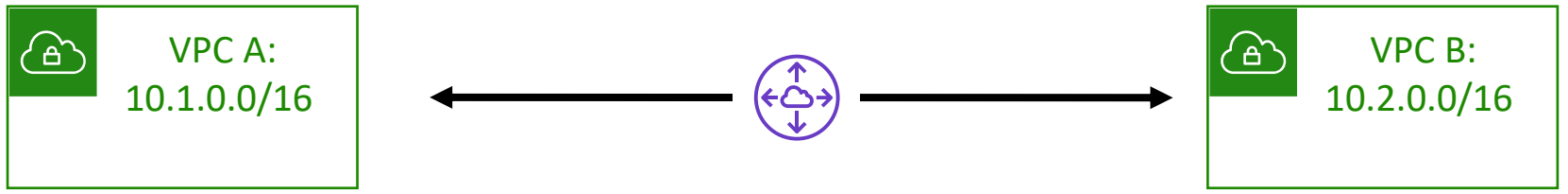
- Isolating some of your workloads is generally a good practice
- However, you might need to transfer data between two or more VPCs



VPC peering

- One-to-one networking connection between two VPCs
- No gateways, VPN connections, and separate network appliances needed
- Highly available connections
- No single point of failure or bandwidth bottleneck
- Traffic always stays on the global AWS backbone

Establishing VPC peering



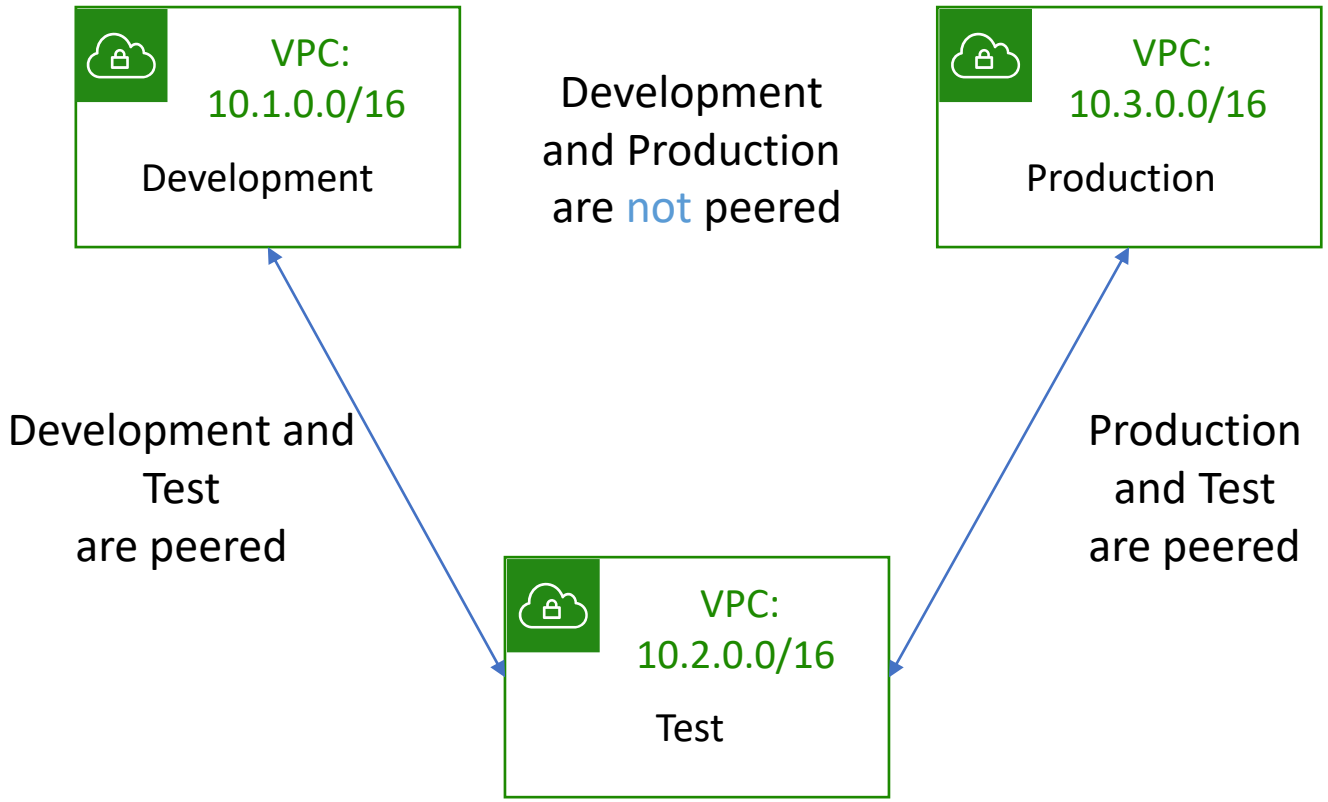
Route table VPC A

Destination	Target
10.1.0.0/16	local
10.2.0.0/16	PCX-1

Route table VPC B

Destination	Target
10.2.0.0/16	local
10.1.0.0/16	PCX-1

VPC peering connection restrictions



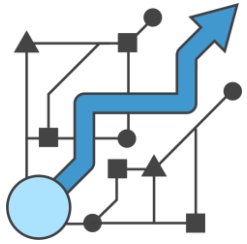
- Use private IP addresses
- Can be established between different AWS accounts
- Cannot have overlapping CIDR blocks
- Can have only one peering resource between any two VPCs
- Do not support transitive peering relationships
- Can be established across regions (inter-region VPC peering connection).

Considerations for peering multiple VPCs

When you connect multiple VPCs, consider these network design principles:

Only connect essential VPCs

Make sure your solution can scale

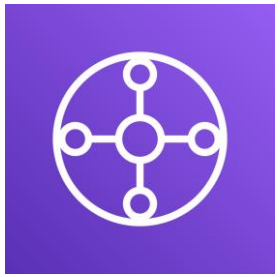


VPC Peering key takeaways

- VPC peering is a one-to-one networking connection between two VPCs that enables you to route traffic between them privately
- You can establish peering relationships between VPCs across different AWS Regions
- VPC peering connections –
  - Use private IP addresses
  - Can be established between different AWS accounts
  - Cannot have overlapping CIDR blocks
  - Can have only one peering resource between any two VPCs
  - Can be established across regions (inter-region VPC peering connection)
  - Do not support transitive peering relationships

Scaling your VPC network with AWS Transit Gateway

AWS Transit Gateway



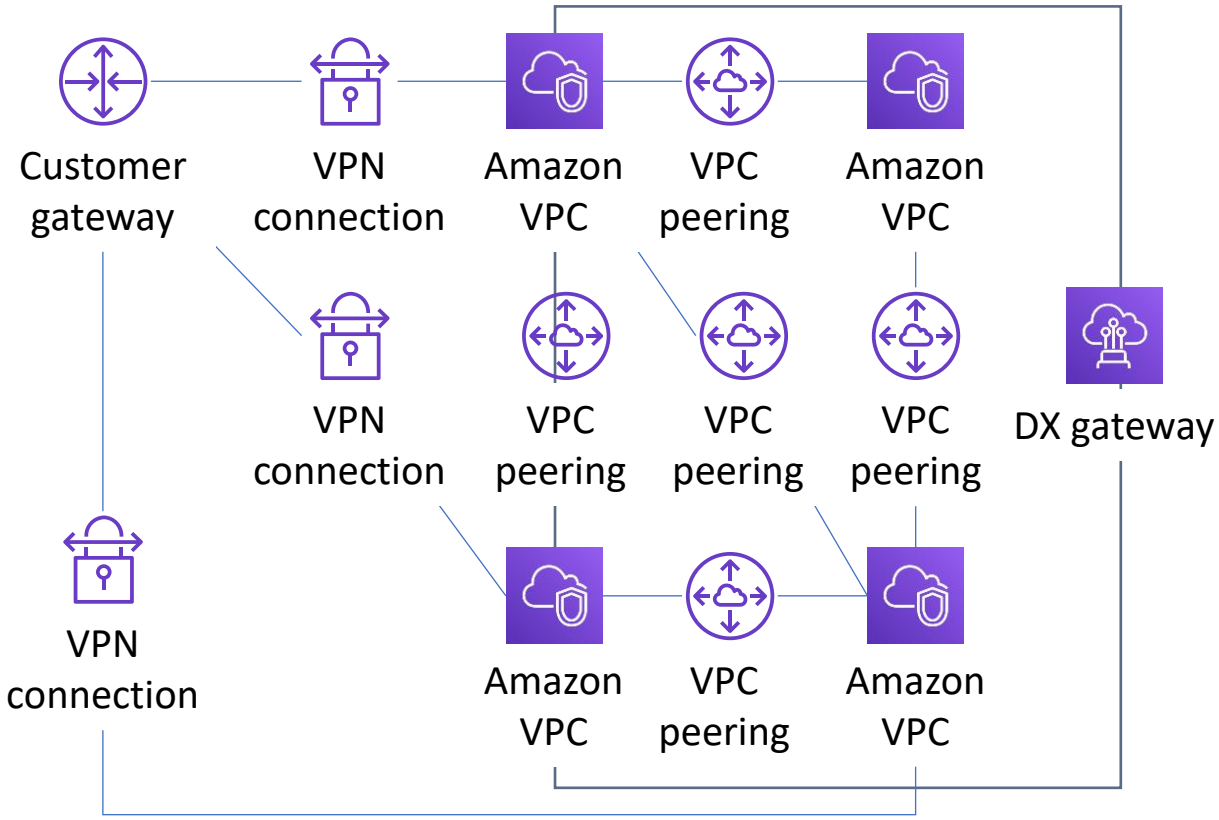
AWS Transit Gateway

AWS Transit Gateway is a service that enables you to connect your VPCs and on-premises networks to a single gateway.

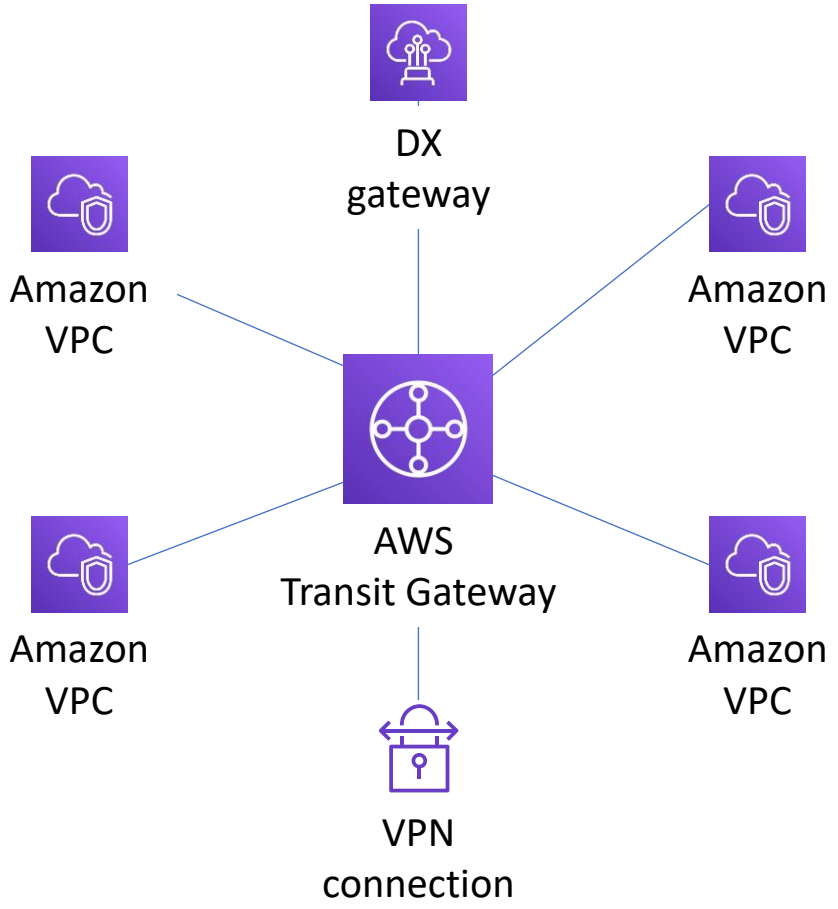
- Fully managed, highly available, flexible routing service
- Acts as a hub for all traffic to flow through between your networks
- Connects up to 5,000 VPCs and on-premises environments with a single gateway

Need to scale networks across multiple VPCs

From this…

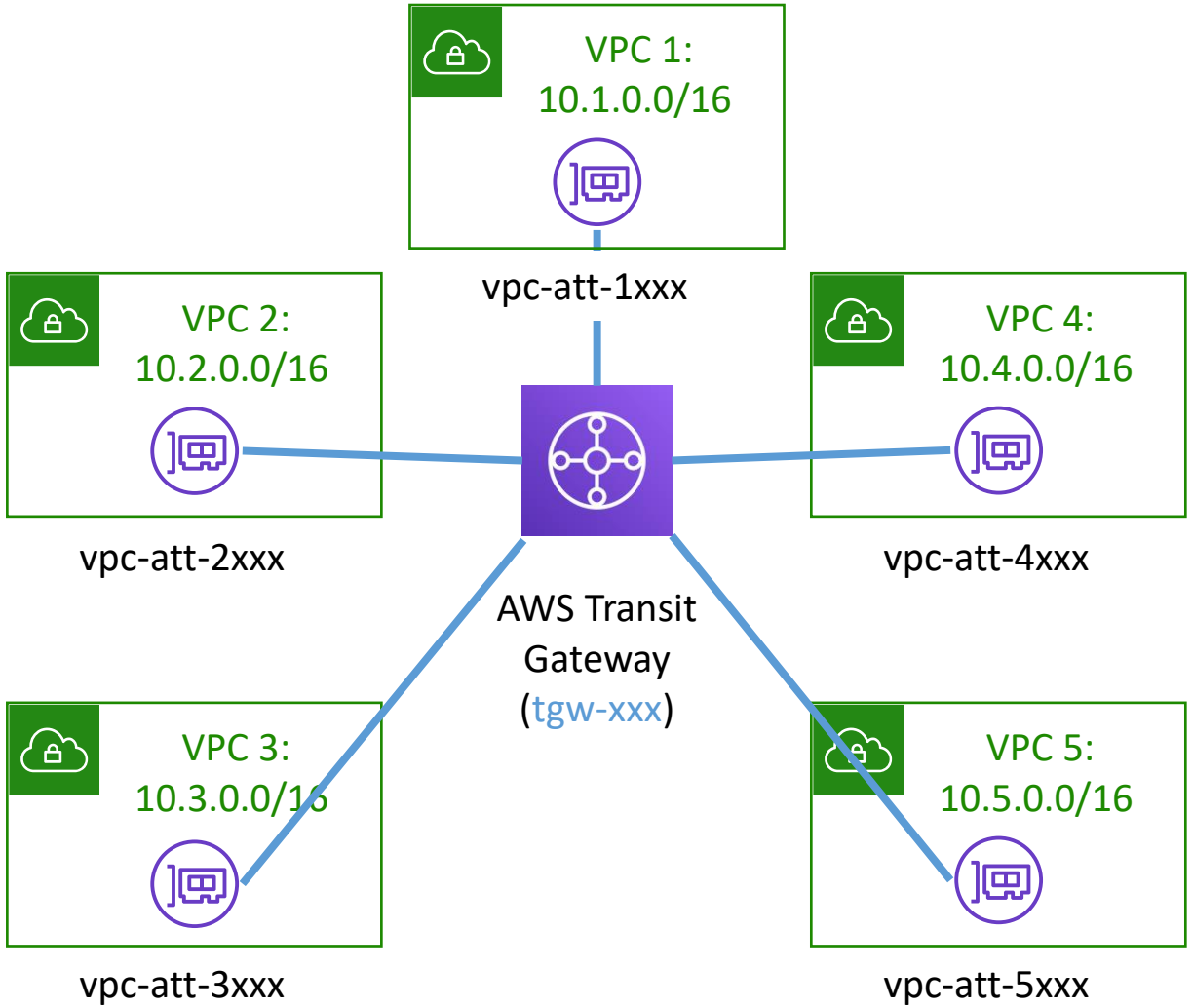


… to this



AWS Transit Gateway: Challenge

Scenario: How do you connect these five VPCs?



VPC 3 route table

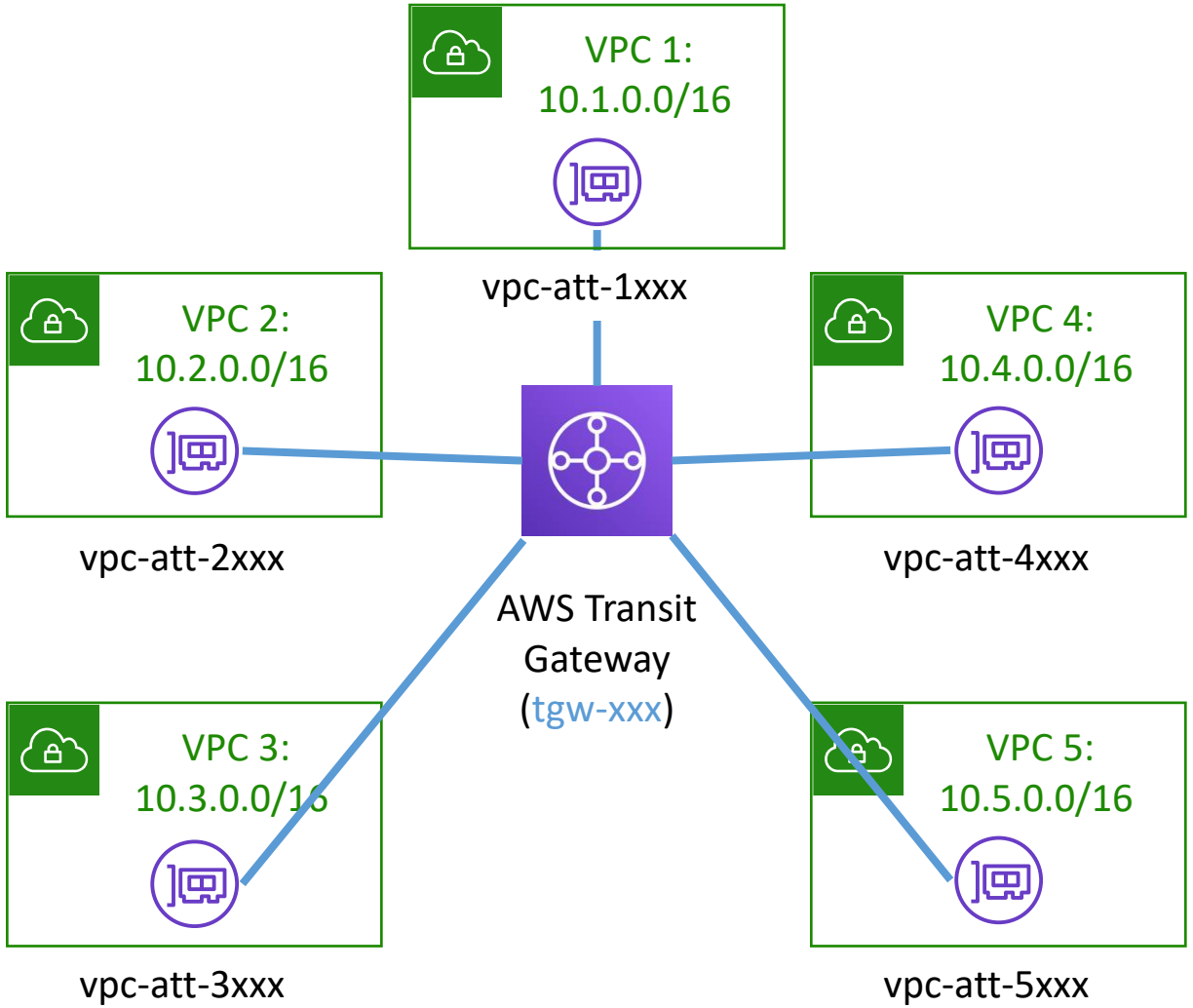
Destination	Target
10.#.0.0/16	local
?	?

Transit gateway route table

Destination	Target
?	?

AWS Transit Gateway activity: Solution

Scenario: How do you connect these five VPCs?



VPC 3 route table

Destination	Target
10.3.0.0/16	local
10.0.0.0/8	tgw-xxx

Transit gateway route table

Destination	Target
10.1.0.0/16	vpc-att-1xxx
10.2.0.0/16	vpc-att-2xxx
10.3.0.0/16	vpc-att-3xxx
10.4.0.0/16	vpc-att-4xxx
10.5.0.0/16	vpc-att-5xxx

Transit Gateway key takeaways

- AWS Transit Gateway enables you to connect your VPCs and on-premises networks to a single gateway (called a transit gateway)
- AWS Transit Gateway uses a hub-and-spoke model to simplify VPC management and reduce operational costs

Multiple accounts

One account or multiple accounts?

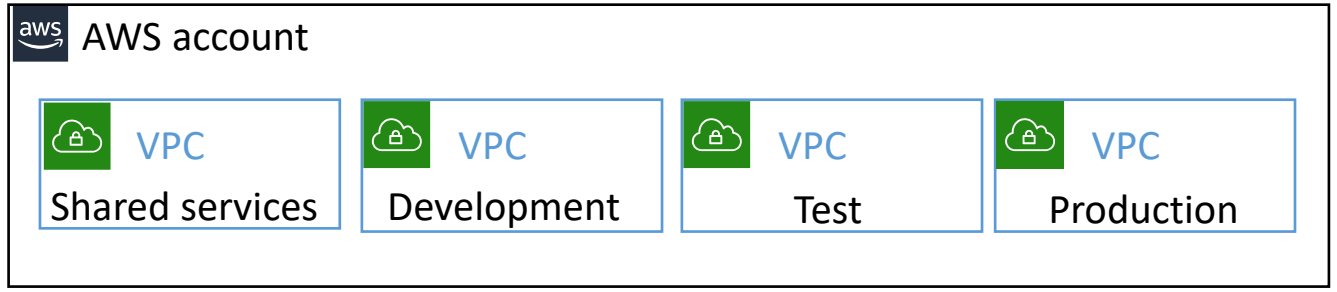
Two architectural patterns

- Most organizations choose to create multiple accounts

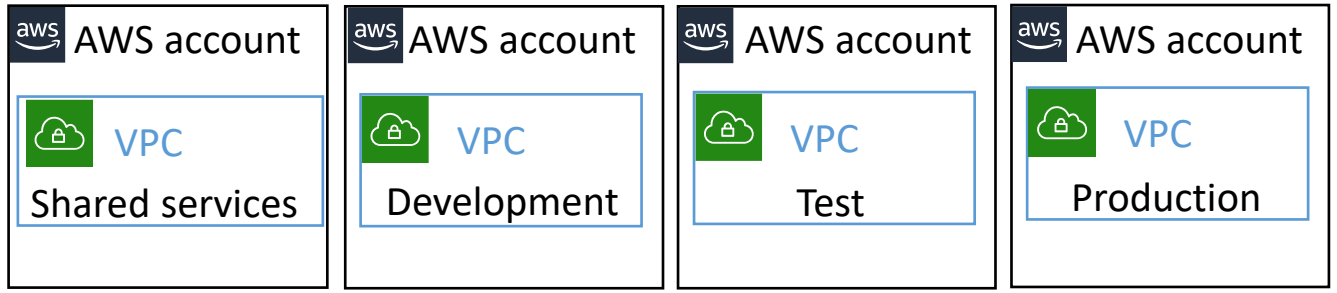
Advantages of multiple accounts

- Isolate business units or departments
- Isolate development, test, and production environments
- Isolate auditing data, recovery data
- Separate accounts for regulated workloads
- Easier to trigger cost alerts for each business unit's consumption

Multiple VPCs in a single account
architectural pattern



Multiple accounts, a VPC in each account
architectural pattern



Challenges for managing multiple accounts

- Security management across accounts
  - IAM policy replication
- Creating new accounts
  - Involves many manual processes
- Billing consolidation
- Centralized governance is needed to ensure consistency
- In AWS, a policy is an object that, when associated with an entity or resource, defines their permissions.

Manage multiple accounts with AWS Organizations

Centrally manage and enforce policies across multiple AWS accounts.

- Group-based account management
- Policy-based access to AWS services
- Automated account creation and management
- Consolidated billing
- API-based

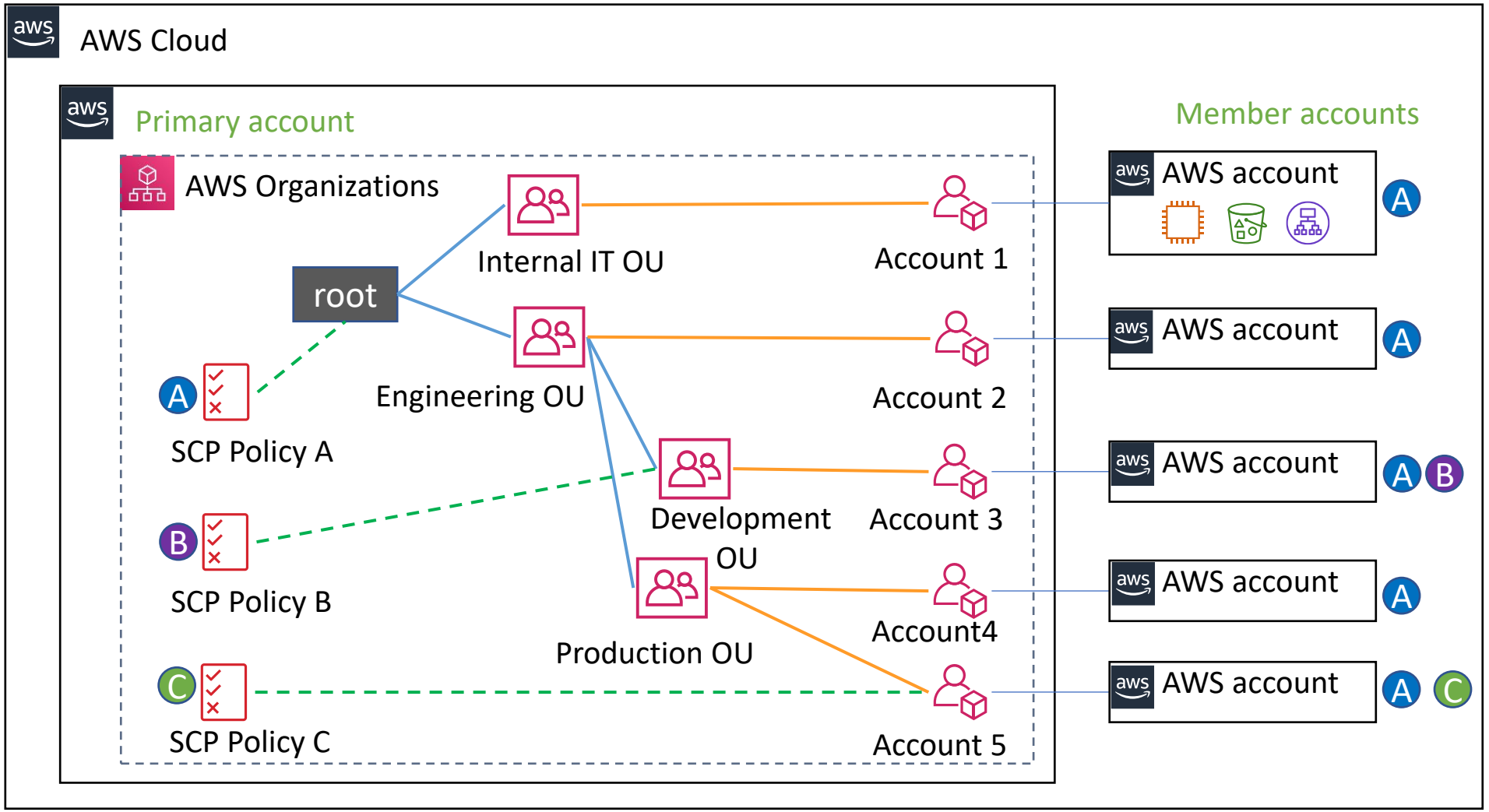
AWS Organizations: Illustrated

Which accounts does each SCP apply to?

In the AWS Organizations primary account:

1. Create a hierarchy of organizational units (OUs)
2. Assign accounts to OUs as member accounts
3. Define service control policies (SCPs) that apply permissions restrictions to specific member accounts
4. Attach the SPCs to root, OUs, or accounts

An account can be a member of only 1 OU at a time.



Example uses of SCPs

- Characteristics of **service control policies** (SCPs)
  - They enable you to control which services are accessible to IAM users in member accounts
  - SCPs cannot be overridden by the local administrator
  - IAM policies that are defined in individual accounts still apply

- Example uses of SCPs
  - Create a policy that *blocks* service access or specific actions

    Example: Deny users from disabling AWS CloudTrail in all member accounts
  - Create a policy that *allows* full access to specific services

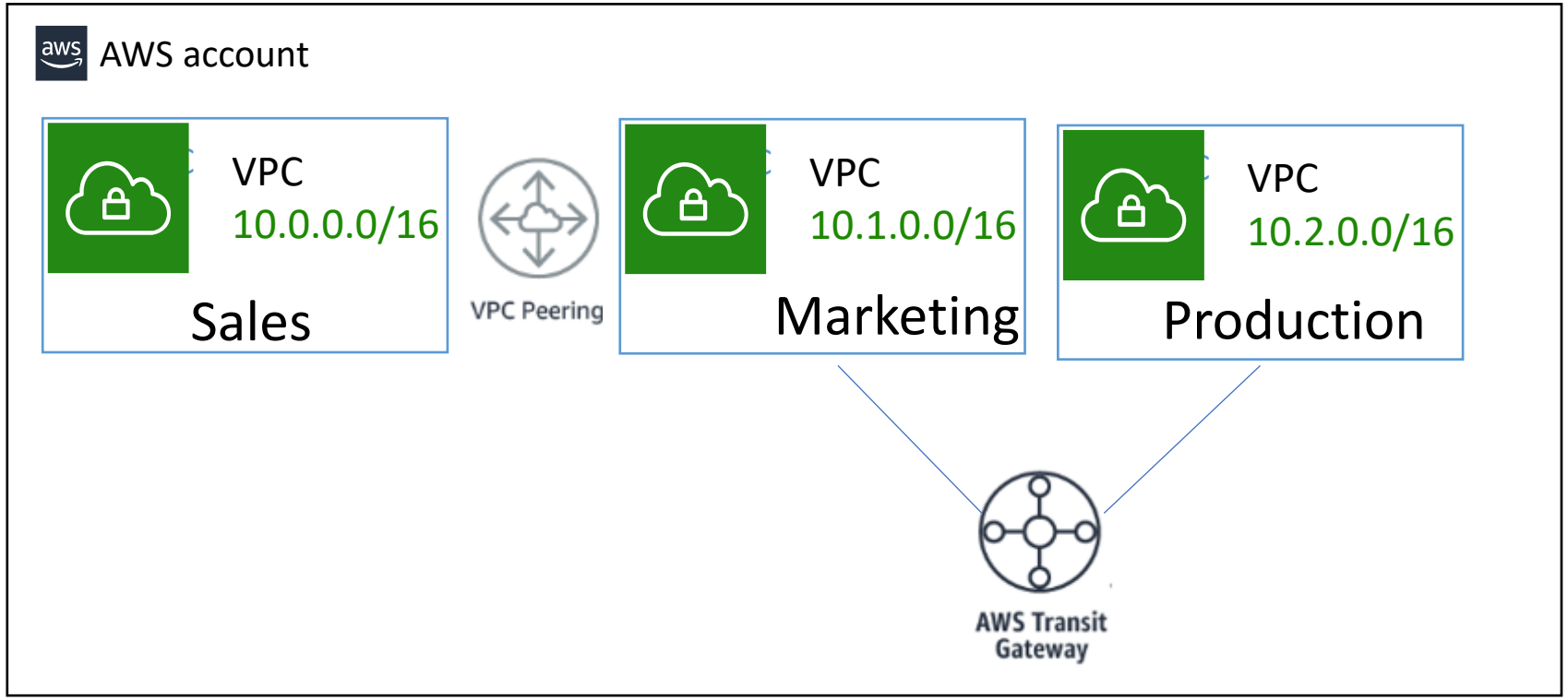
    Example: Allow full access to Amazon EC2 and CloudWatch
  - Create a policy that *enforces the tagging* of resources

key takeaways

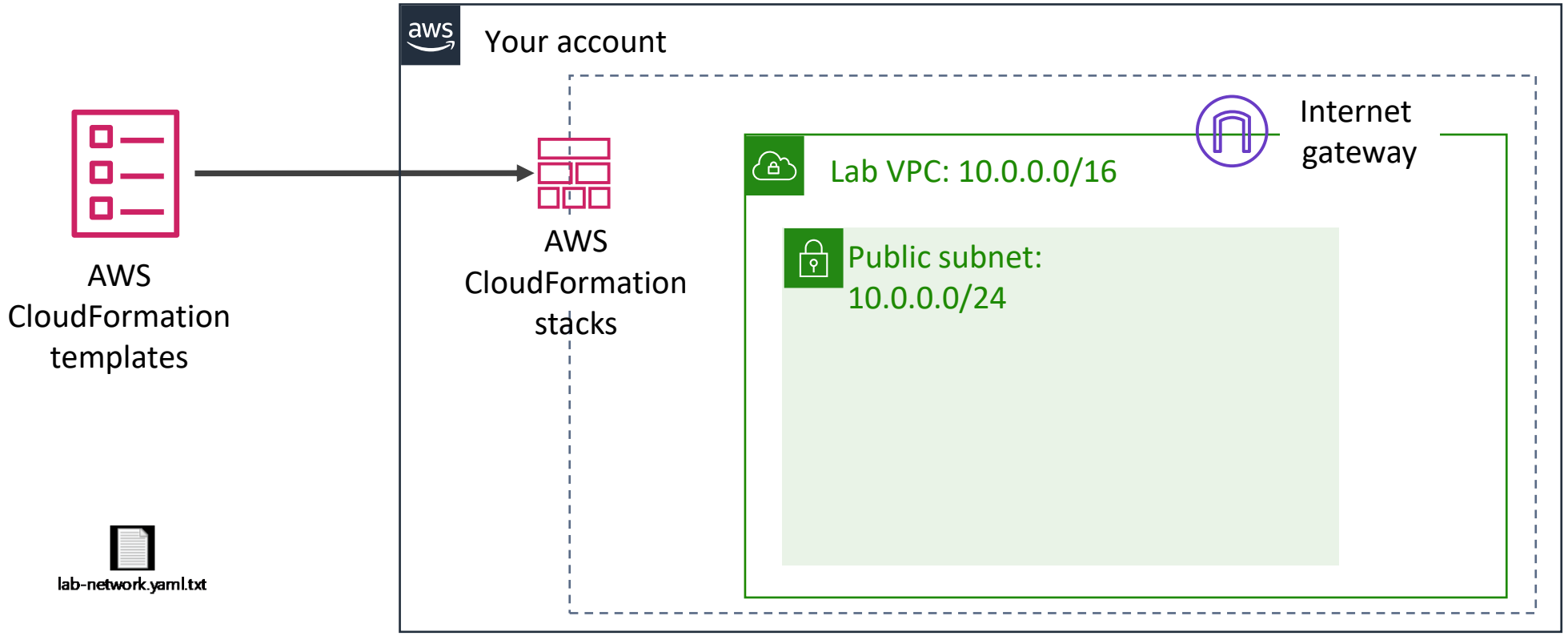
- You can use multiple AWS accounts to isolate business units, development and test environments, regulated workloads, and auditing data
- AWS Organizations enables you to configure automated account creation and consolidated billing
- You can configure access controls across accounts by using service control policies (SCPs)

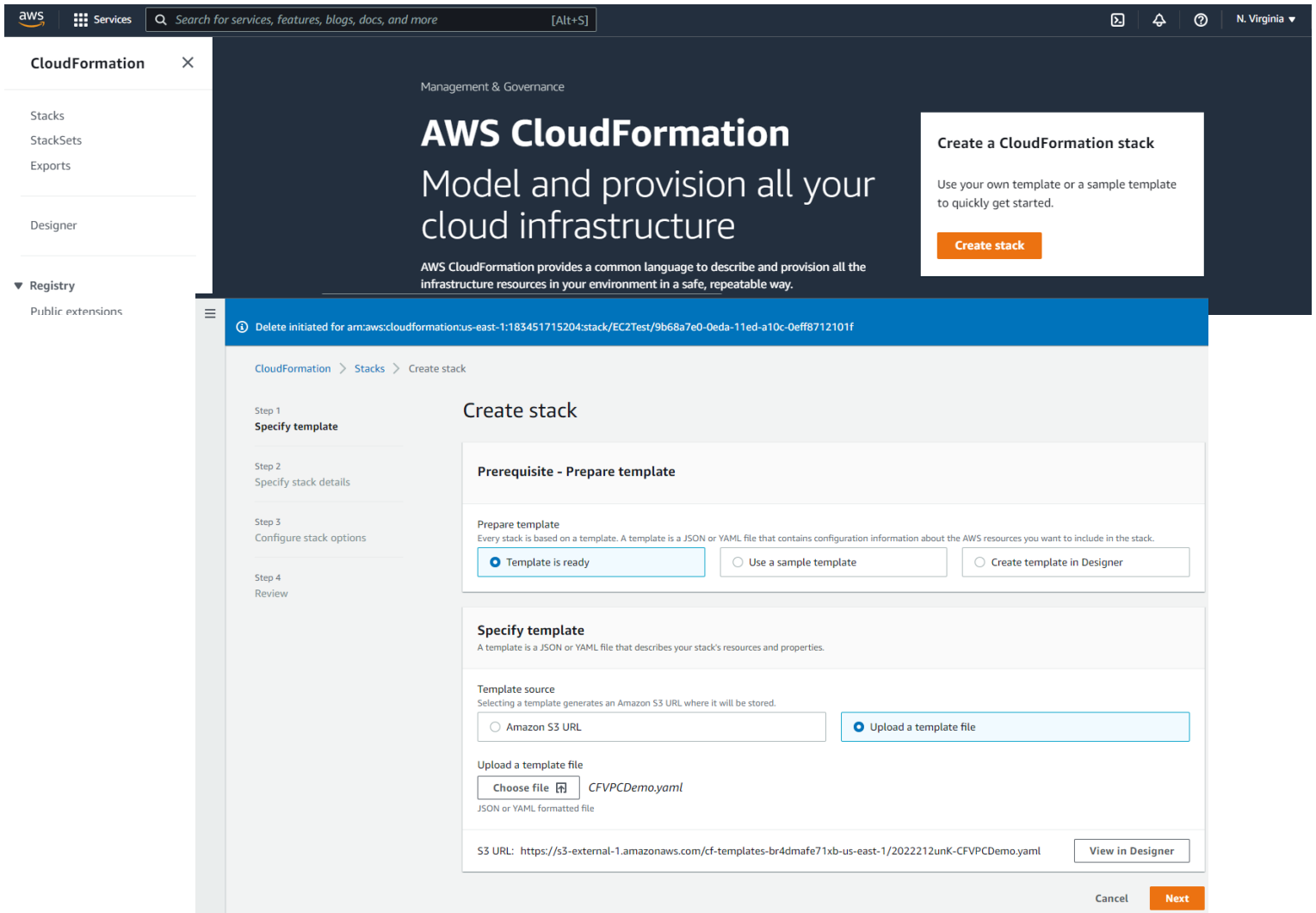
Demos

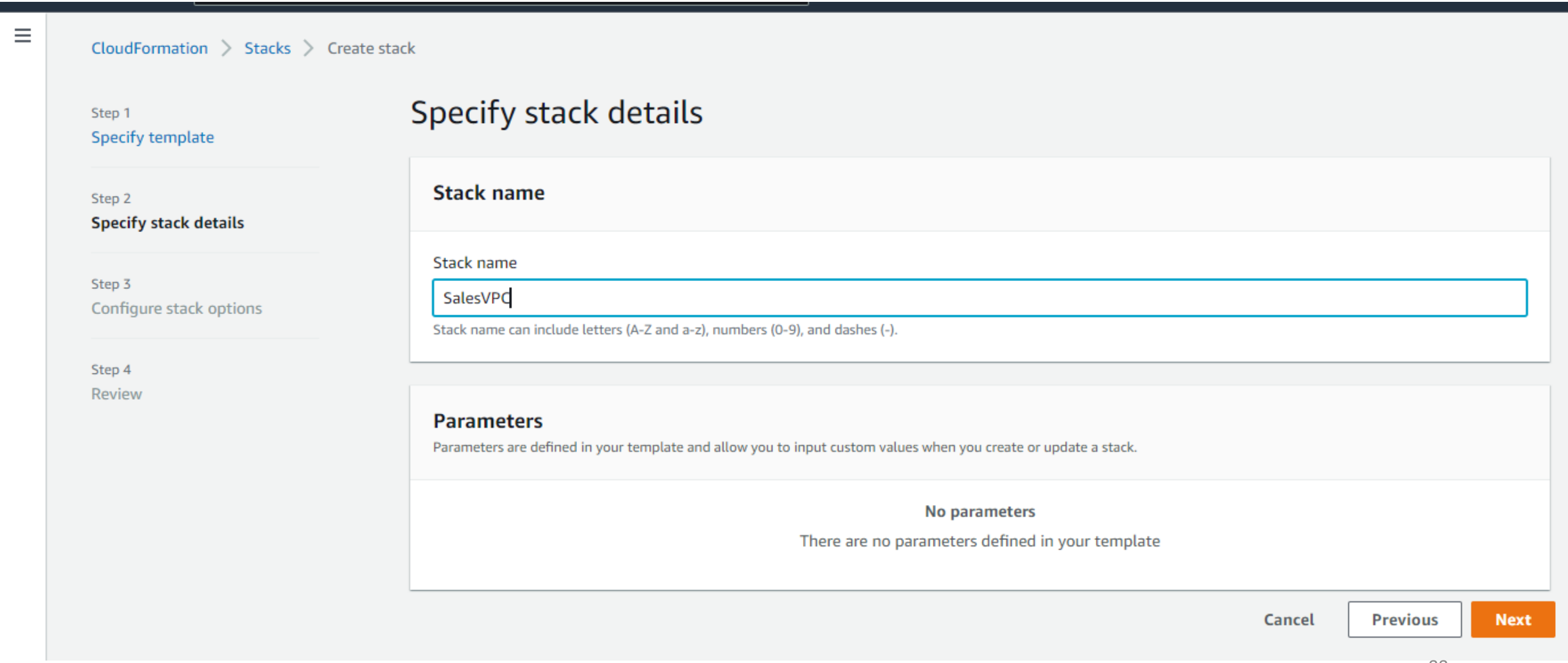
- Demo 1: Create a VPC stack
- Demo 2: Drift Detection
- Demo 3: Create an EC2 stack on the VPC Stack
- Demo 4: Create Multiple VPC + EC2 stacks
- Demo 5: Create VPC Peering
- Demo 6: Interconnect VPCs using Transit Gateway

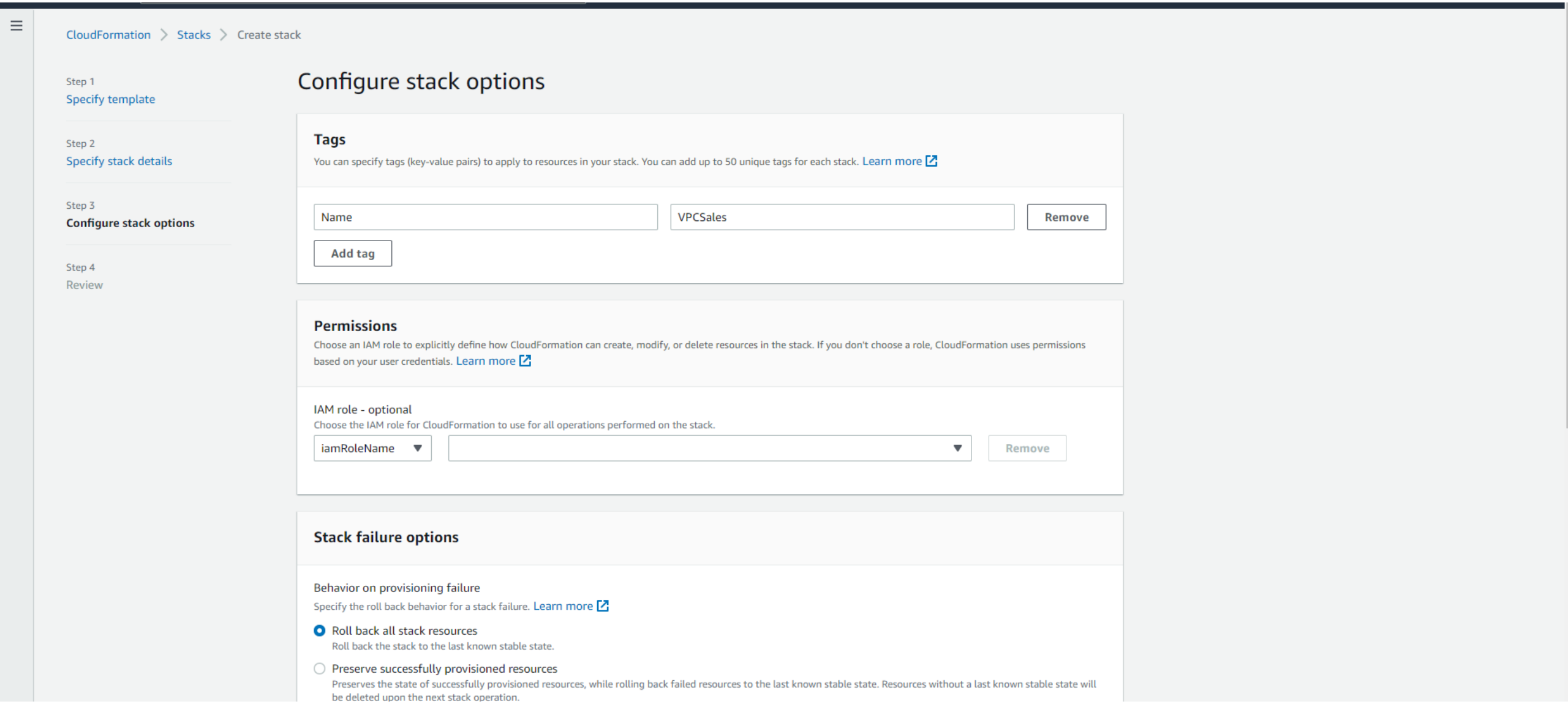

AWS account
VPC
10.0.0.0/16
Sales
VPC Peering
VPC
10.1.0.0/16
Marketing
VPC
10.2.0.0/16
Production
AWS Transit Gateway

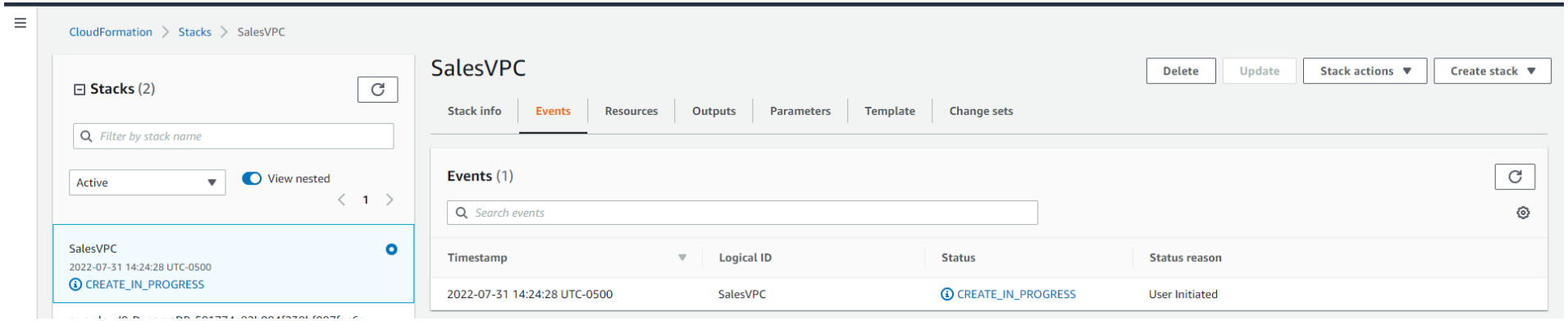
Demo 1 Create a VPC Stack (Sales)

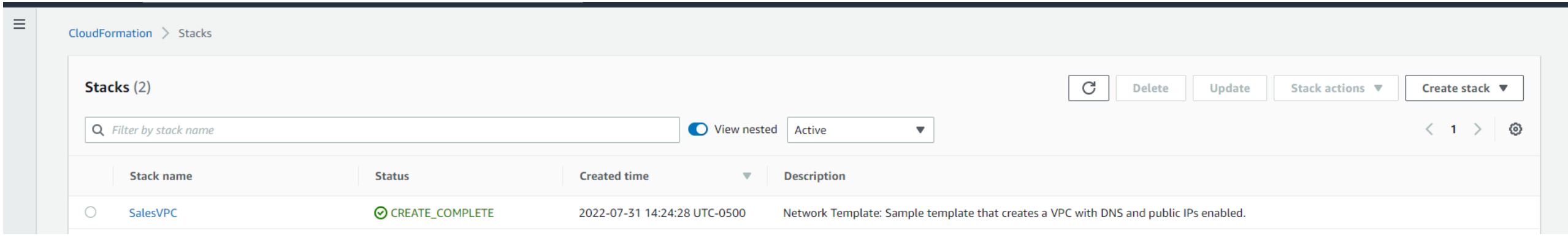


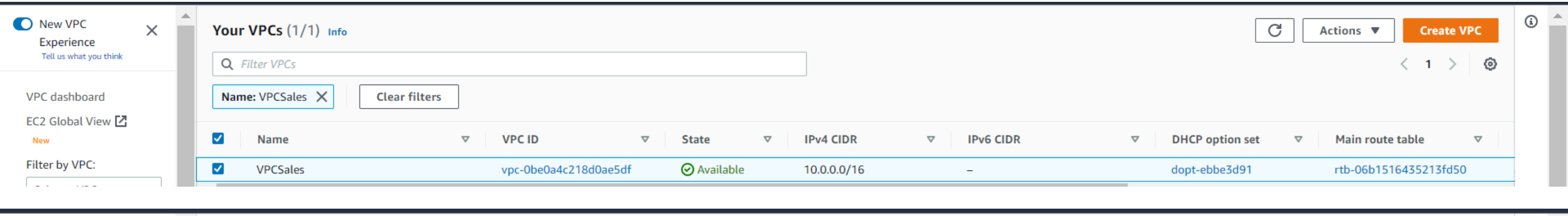

aws
Services
Search for services, features, blogs, docs, and more
[Alt+S]
N. Virginia
CloudFormation
Stacks
StackSets
Exports
Designer
Registry
Public extensions
Management & Governance
AWS CloudFormation
Model and provision all your cloud infrastructure
AWS CloudFormation provides a common language to describe and provision all the infrastructure resources in your environment in a safe, repeatable way.
Create a CloudFormation stack
Use your own template or a sample template to quickly get started.
Create stack
Delete initiated for arn:aws:cloudformation:us-east-1:183451715204:stack/EC2Test/9b68a7e0-0eda-11ed-a10c-0eff8712101f
CloudFormation > Stacks > Create stack
Step 1
Specify template
Step 2
Specify stack details
Step 3
Configure stack options
Step 4
Review
Create stack
Prerequisite - Prepare template
Prepare template
Every stack is based on a template. A template is a JSON or YAML file that contains configuration information about the AWS resources you want to include in the stack.
Template is ready
Use a sample template
Create template in Designer
Specify template
A template is a JSON or YAML file that describes your stack's resources and properties.
Template source
Selecting a template generates an Amazon S3 URL where it will be stored.
Amazon S3 URL
Upload a template file
Upload a template file
Choose file
CFVPCDemo.yaml
JSON or YAML formatted file
S3 URL: https://s3-external-1.amazonaws.com/cf-templates-br4dmafe71xb-us-east-1/2022212unK-CFVPCDemo.yaml
View in Designer
Cancel
Next

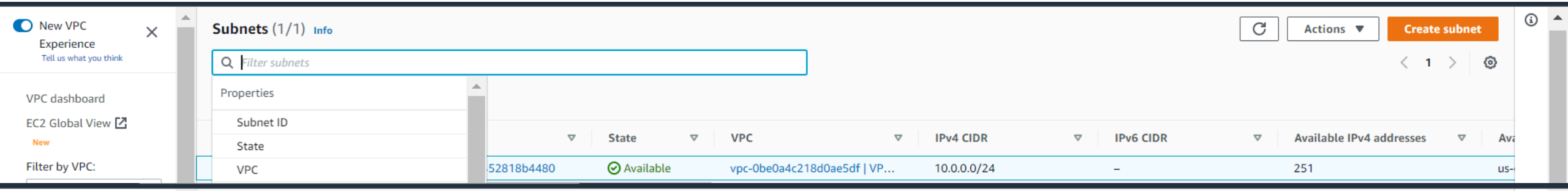

CloudFormation
Stacks
Create stack
Step 1
Specify template
Step 2
Specify stack details
Step 3
Configure stack options
Step 4
Review
Specify stack details
Stack name
Stack name
SalesVPC
Stack name can include letters (A-Z and a-z), numbers (0-9), and dashes (-).
Parameters
Parameters are defined in your template and allow you to input custom values when you create or update a stack.
No parameters
There are no parameters defined in your template
Cancel
Previous
Next

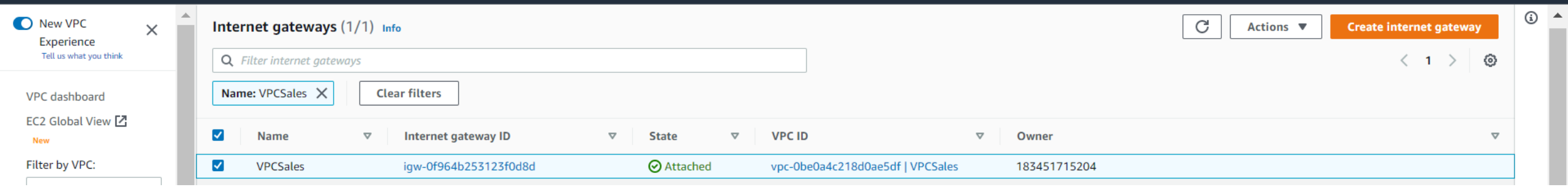

CloudFormation > Stacks > Create stack
Step 1
Specify template
Step 2
Specify stack details
Step 3
Configure stack options
Step 4
Review
Configure stack options
Tags
You can specify tags (key-value pairs) to apply to resources in your stack. You can add up to 50 unique tags for each stack. Learn more
Name
VPCSales
Remove
Add tag
Permissions
Choose an IAM role to explicitly define how CloudFormation can create, modify, or delete resources in the stack. If you don't choose a role, CloudFormation uses permissions based on your user credentials. Learn more
IAM role - optional
Choose the IAM role for CloudFormation to use for all operations performed on the stack.
iamRoleName
Remove
Stack failure options
Behavior on provisioning failure
Specify the roll back behavior for a stack failure. Learn more
Roll back all stack resources
Roll back the stack to the last known stable state.
Preserve successfully provisioned resources
Preserves the state of successfully provisioned resources, while rolling back failed resources to the last known stable state. Resources without a last known stable state will be deleted upon the next stack operation.

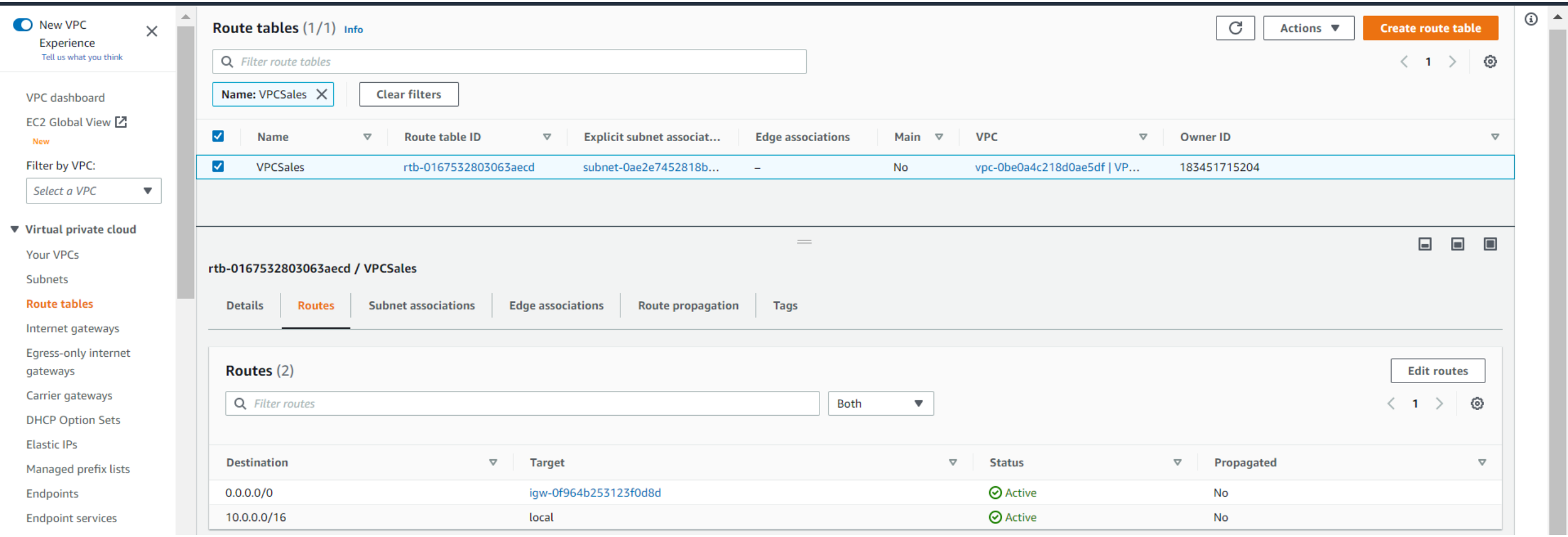

CloudFormation > Stacks > SalesVPC
Stacks (2)
Filter by stack name
Active
View nested
1
SalesVPC
2022-07-31 14:24:28 UTC-0500
CREATE_IN_PROGRESS
SalesVPC
Delete
Update
Stack actions
Create stack
Stack info
Events
Resources
Outputs
Parameters
Template
Change sets
Events (1)
Search events
Timestamp
Logical ID
Status
Status reason
2022-07-31 14:24:28 UTC-0500
SalesVPC
CREATE_IN_PROGRESS
User Initiated


CloudFormation > Stacks
Stacks (2)
Delete
Update
Stack actions
Create stack
Filter by stack name
View nested
Active
1
Stack name
Status
Created time
Description
SalesVPC
CREATE_COMPLETE
2022-07-31 14:24:28 UTC-0500
Network Template: Sample template that creates a VPC with DNS and public IPs enabled.


New VPC Experience
Tell us what you think
VPC dashboard
EC2 Global View
New
Filter by VPC:
Your VPCs (1/1) Info
Actions
Create VPC
Filter VPCs
1
Name: VPCSales
Clear filters
Name
VPC ID
State
IPv4 CIDR
IPv6 CIDR
DHCP option set
Main route table
VPCSales
vpc-0be0a4c218d0ae5df
Available
10.0.0.0/16
–
dopt-ebbe3d91
rtb-06b1516435213fd50

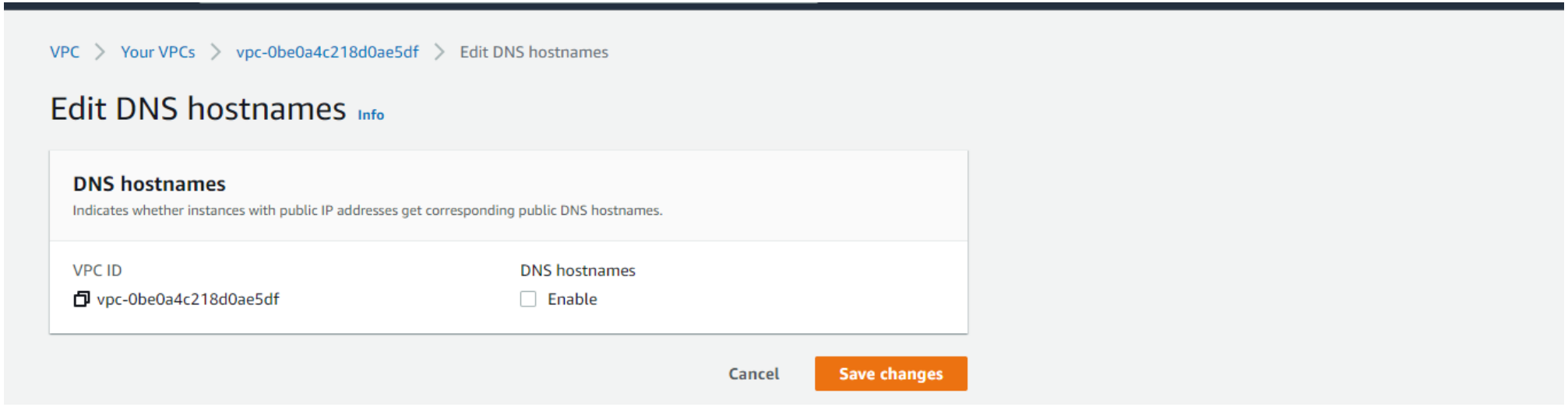

New VPC Experience
Tell us what you think
VPC dashboard
EC2 Global View
New
Filter by VPC:
Subnets (1/1) Info
Actions
Create subnet
Filter subnets
1
Properties
Subnet ID
State
VPC
State
VPC
IPv4 CIDR
IPv6 CIDR
Available IPv4 addresses
Available
vpc-0be0a4c218d0ae5df | VP...
10.0.0.0/24
–
251

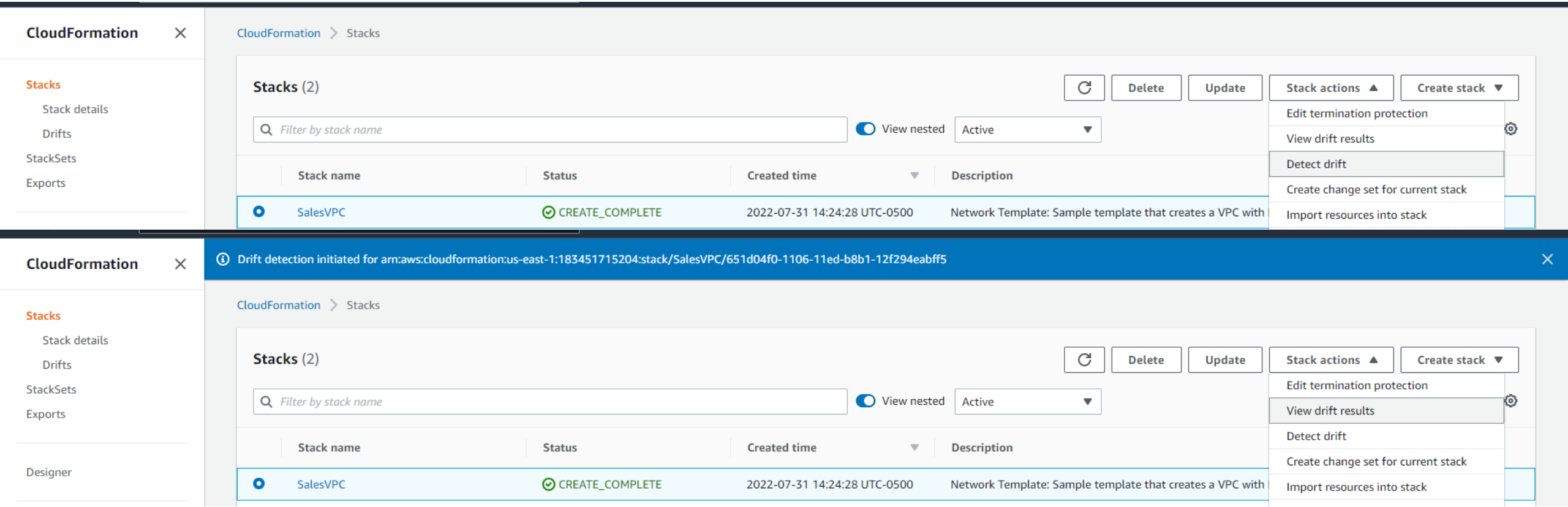

New VPC Experience
Tell us what you think
VPC dashboard
EC2 Global View
New
Filter by VPC:
Internet gateways (1/1) Info
Actions
Create internet gateway
Filter internet gateways
1
Name: VPCSales
Clear filters
Name
Internet gateway ID
State
VPC ID
Owner
VPCSales
igw-0f964b253123f0d8d
Attached
vpc-0be0a4c218d0ae5df | VPCSales
183451715204


New VPC Experience
Tell us what you think
VPC dashboard
EC2 Global View
New
Filter by VPC:
Select a VPC
Virtual private cloud
Your VPCs
Subnets
Route tables
Internet gateways
Egress-only internet gateways
Carrier gateways
DHCP Option Sets
Elastic IPs
Managed prefix lists
Endpoints
Endpoint services
Route tables (1/1) Info
Actions
Create route table
Filter route tables
Name: VPCSales
Clear filters
1
Name
Route table ID
Explicit subnet associat...
Edge associations
Main
VPC
Owner ID
VPCSales
rtb-0167532803063aecd
subnet-0ae2e7452818b...
–
No
vpc-0be0a4c218d0ae5df | VP...
183451715204
rtb-0167532803063aecd / VPCSales
Details
Routes
Subnet associations
Edge associations
Route propagation
Tags
Routes (2)
Edit routes
Filter routes
Both
1
Destination
Target
Status
Propagated
0.0.0.0/0
igw-0f964b253123f0d8d
Active
No
10.0.0.0/16
local
Active
No

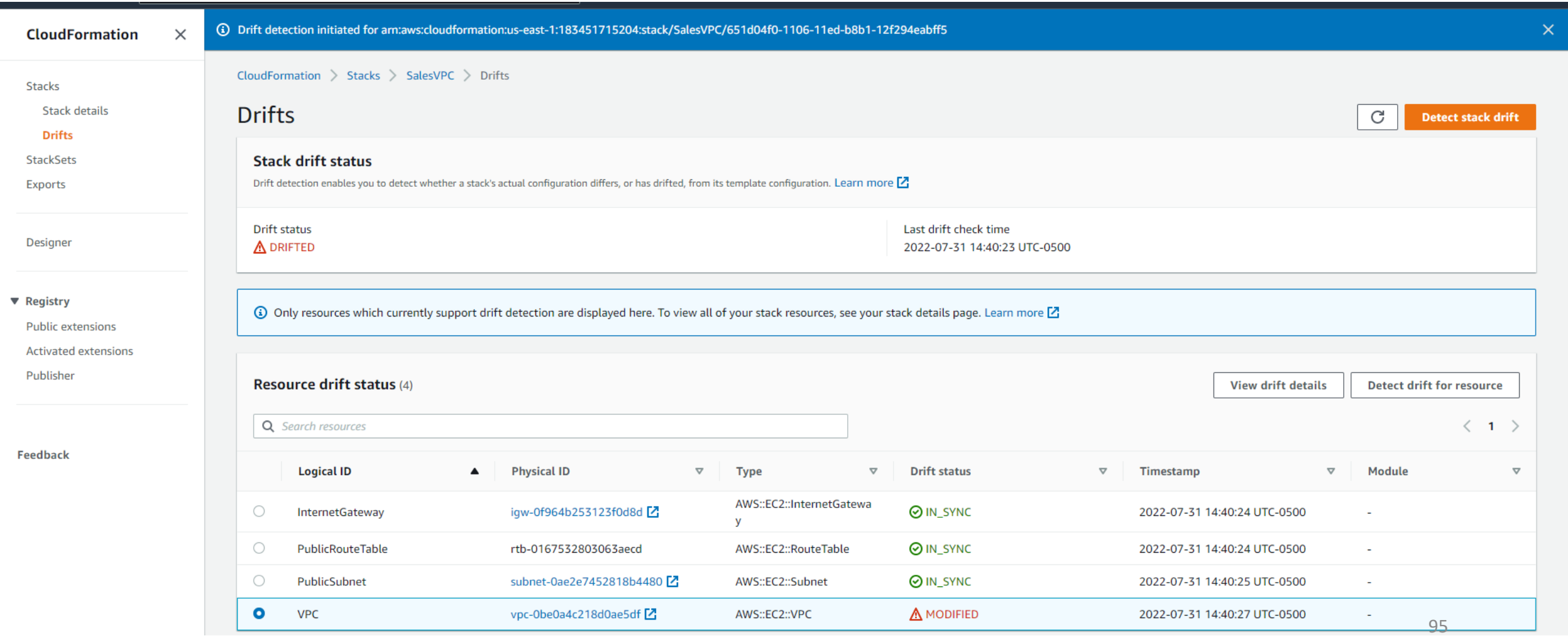
Demo 2: Drift Detection

- Let’s say someone changed configuration. You can use Drift Detection to check.
- The SalesVPC stack is with EnableDnsHostnames: true and it was changed to false.
- We will run Drift Detction

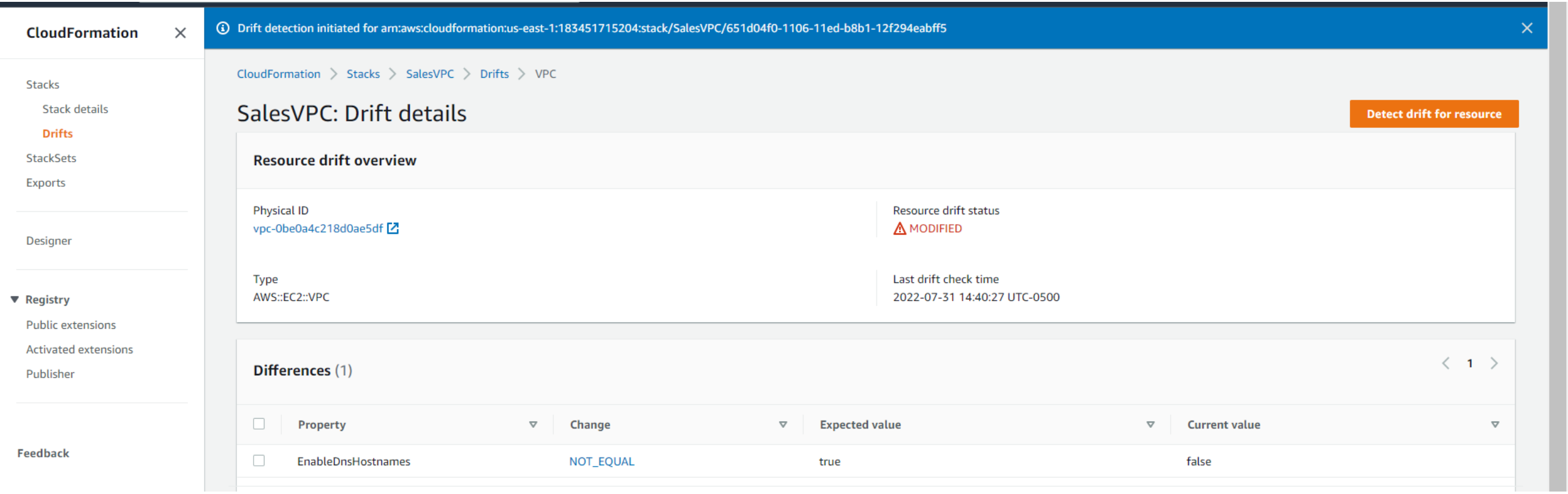



CloudFormation
Stacks
Stack details
Drifts
StackSets
Exports
CloudFormation > Stacks
Stacks (2)
Delete
Update
Stack actions
Create stack
Edit termination protection
View drift results
Detect drift
Create change set for current stack
Import resources into stack
Filter by stack name
View nested
Active
Stack name
Status
Created time
Description
SalesVPC
CREATE_COMPLETE
2022-07-31 14:24:28 UTC-0500
Network Template: Sample template that creates a VPC with

CloudFormation
Drift detection initiated for arn:aws:cloudformation:us-east-1:183451715204:stack/SalesVPC/651d04f0-1106-11ed-b8b1-12f294eabff5
Stacks
Stack details
Drifts
StackSets
Exports
Designer
CloudFormation > Stacks
Stacks (2)
Delete
Update
Stack actions
Create stack
Edit termination protection
View drift results
Detect drift
Create change set for current stack
Import resources into stack
Filter by stack name
View nested
Active
Stack name
Status
Created time
Description
SalesVPC
CREATE_COMPLETE
2022-07-31 14:24:28 UTC-0500
Network Template: Sample template that creates a VPC with

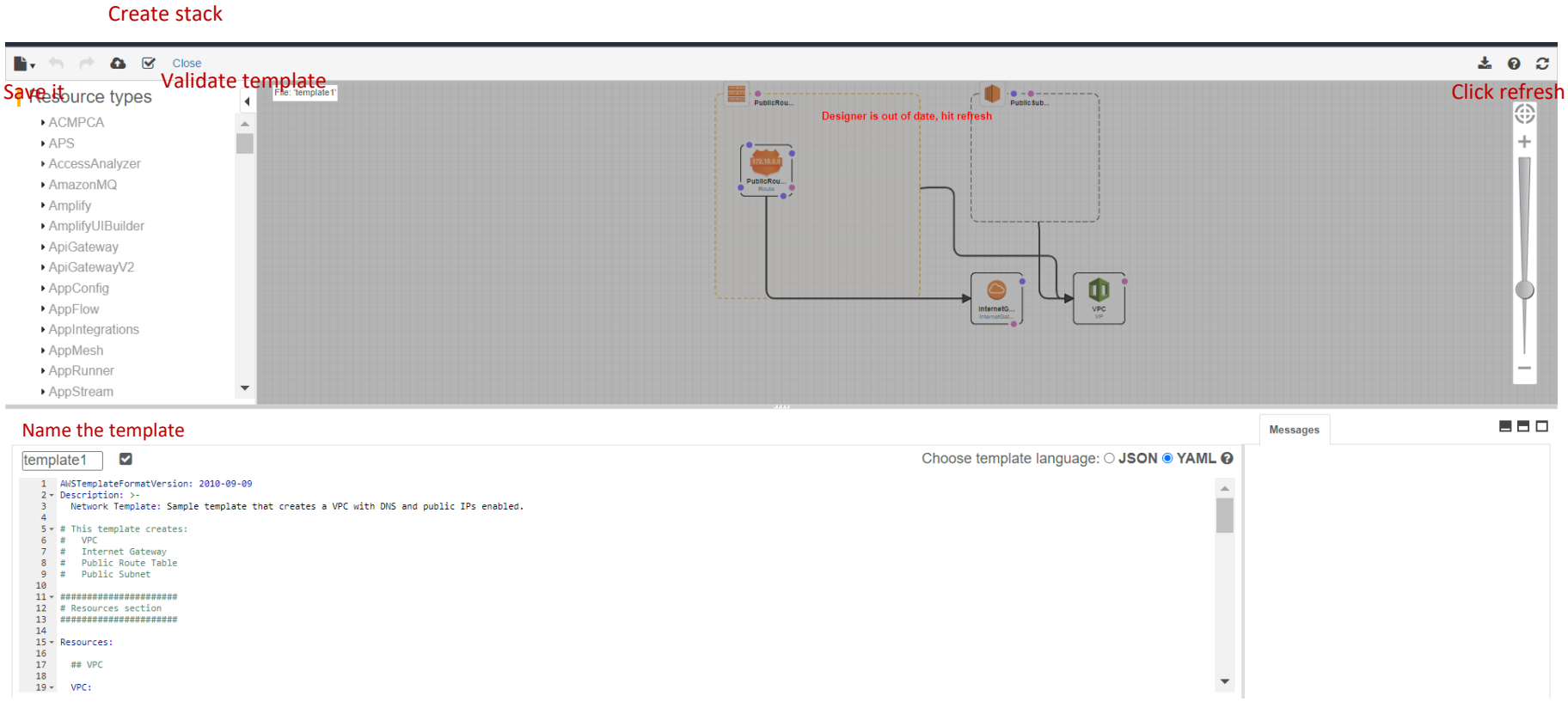

CloudFormation
Stacks
Stack details
Drifts
StackSets
Exports
Designer
Registry
Public extensions
Activated extensions
Publisher
Feedback
Drift detection initiated for arn:aws:cloudformation:us-east-1:183451715204:stack/SalesVPC/651d04f0-1106-11ed-b8b1-12f294eabff5
CloudFormation > Stacks > SalesVPC > Drifts
Drifts
Detect stack drift
Stack drift status
Drift detection enables you to detect whether a stack's actual configuration differs, or has drifted, from its template configuration. Learn more
Drift status
DRIFTED
Last drift check time
2022-07-31 14:40:23 UTC-0500
Only resources which currently support drift detection are displayed here. To view all of your stack resources, see your stack details page. Learn more
Resource drift status (4)
View drift details
Detect drift for resource
Search resources
1
Logical ID
Physical ID
Type
Drift status
Timestamp
Module
InternetGateway
igw-0f964b253123f0d8d
AWS::EC2::InternetGateway
IN_SYNC
2022-07-31 14:40:24 UTC-0500
-
PublicRouteTable
rtb-0167532803063aecd
AWS::EC2::RouteTable
IN_SYNC
2022-07-31 14:40:24 UTC-0500
-
PublicSubnet
subnet-0ae2e7452818b4480
AWS::EC2::Subnet
IN_SYNC
2022-07-31 14:40:25 UTC-0500
-
VPC
vpc-0be0a4c218d0ae5df
AWS::EC2::VPC
MODIFIED
2022-07-31 14:40:27 UTC-0500
-

It will show you the deviations



- I will set it back to the right value to continue with the demo.

If Using Designer

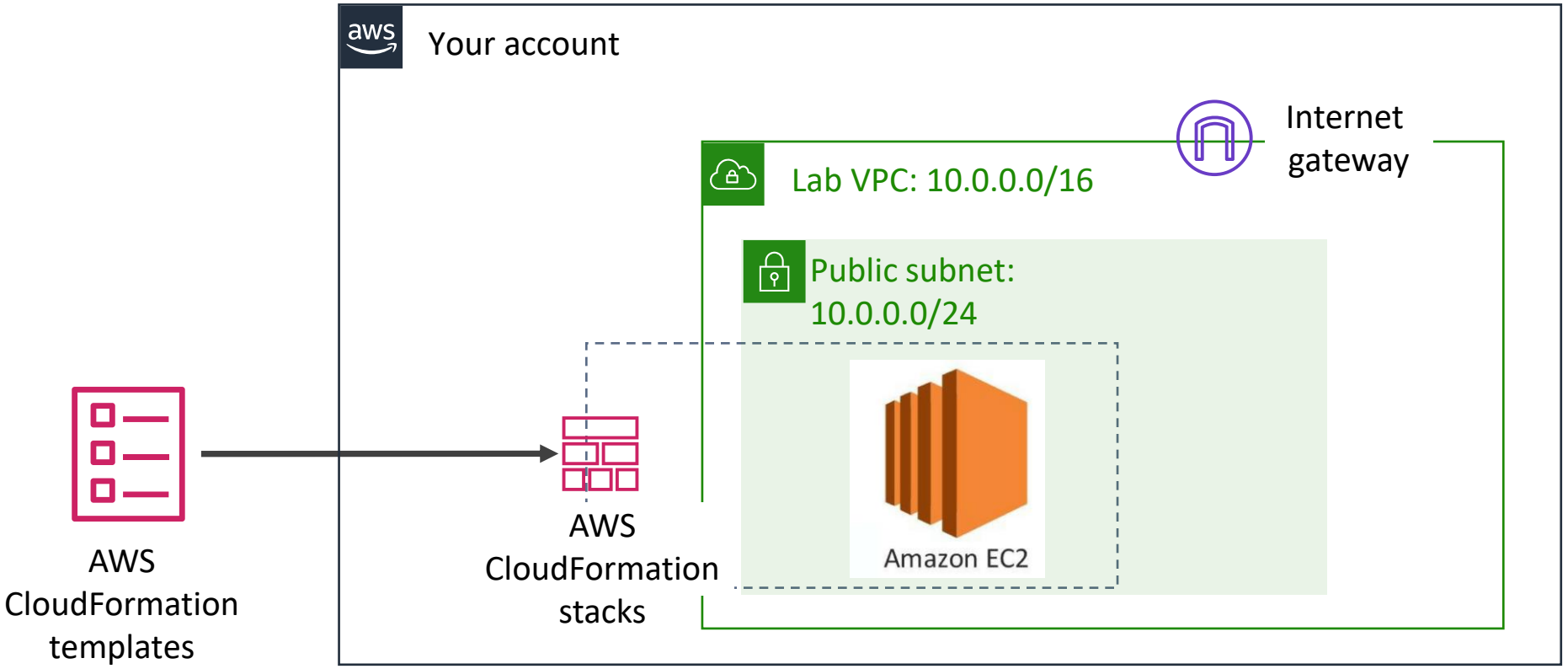


When you upload a template or start from scratch, you can use Designer to visually see the configuration.

If make changes:

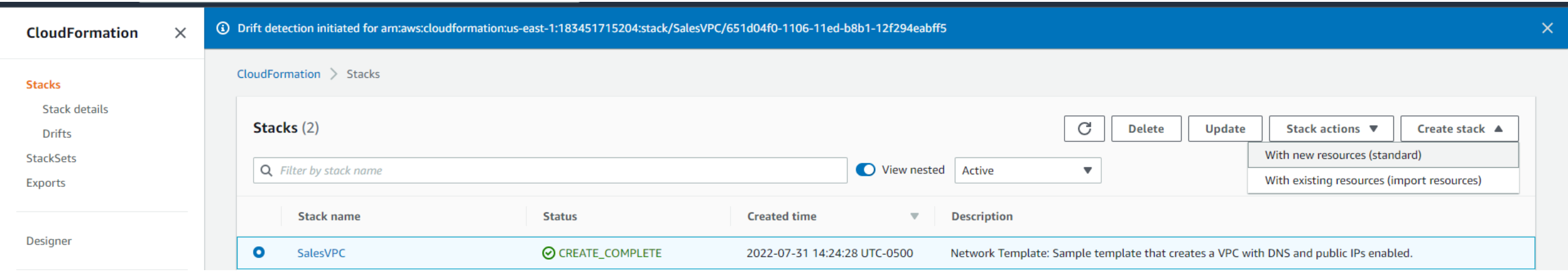
1. Name the template
2. Click refresh
3. Validate template for syntax error
4. Save it (in Download if local)
5. Create stack

Demo 3: Create an EC2 stack (EC2Sales) on the VPC Stack (SalesVPC)

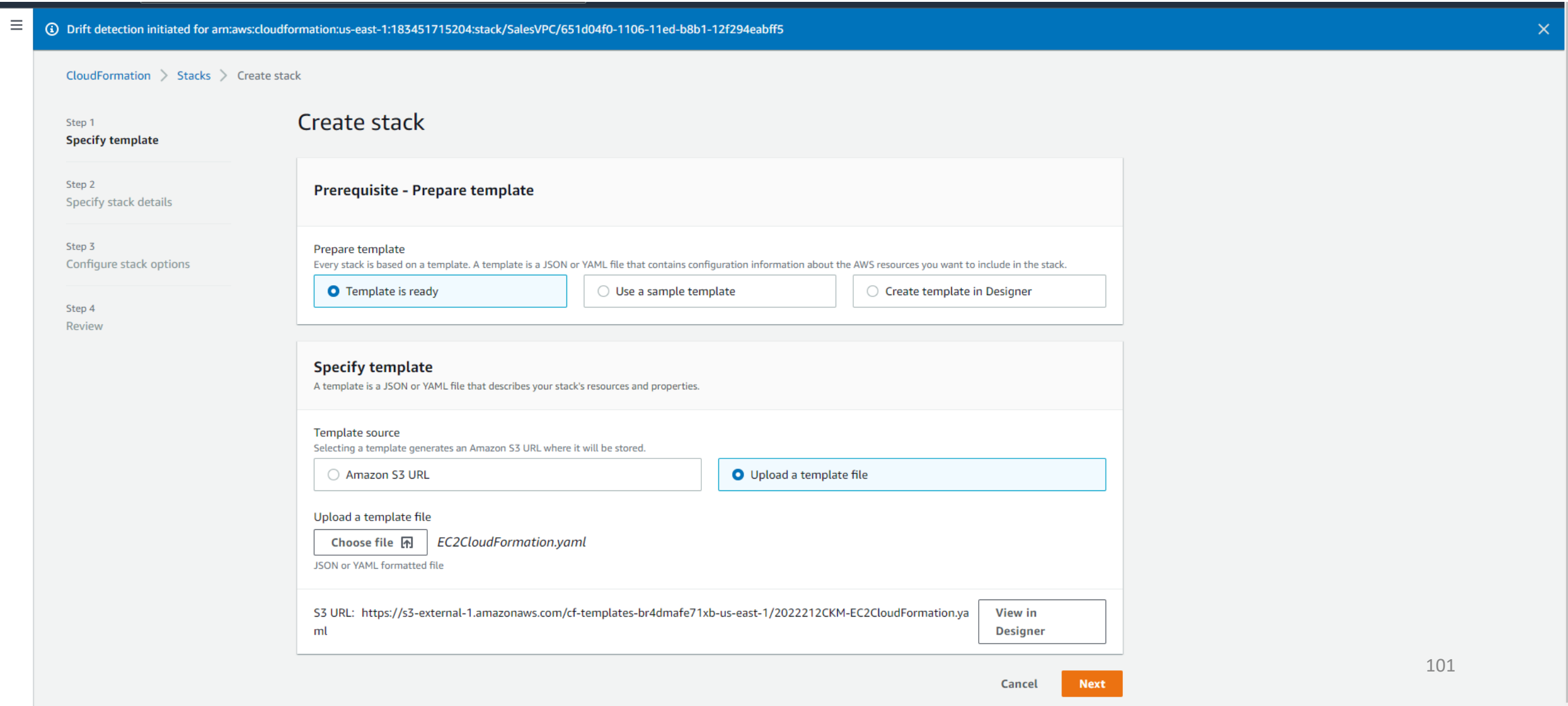


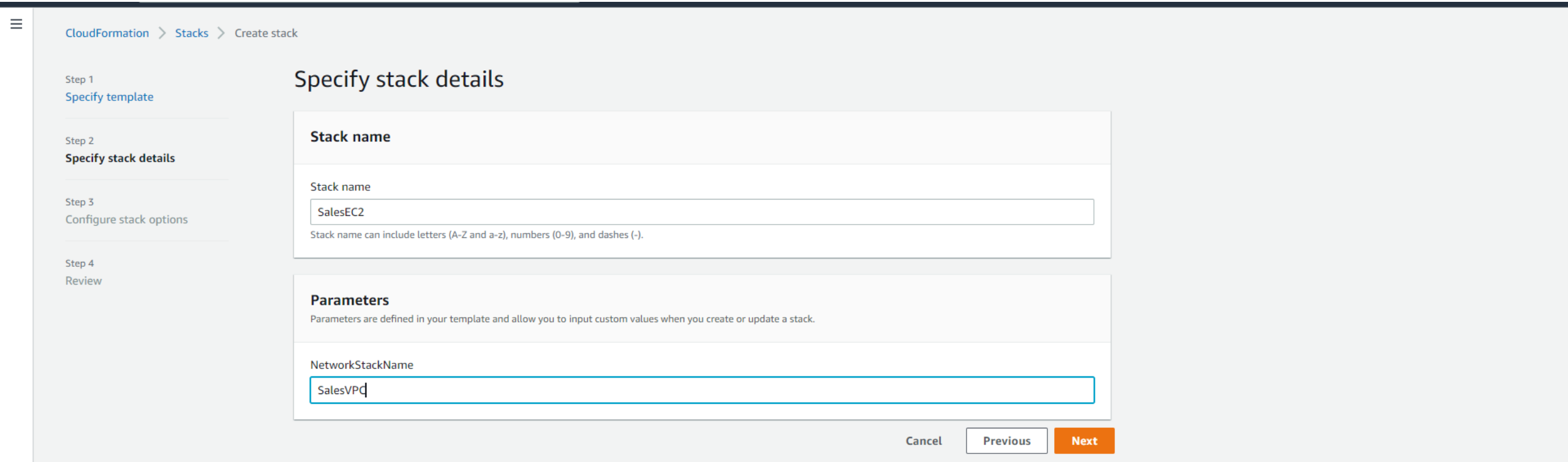
```
Parameters:

  NetworkStackName:
    Type: String


######################
# Resources section
######################

Resources:

  EC2ServerInstance:
    Type: AWS::EC2::Instance
    Properties:
      InstanceType: t2.micro
      ImageId: ami-052efd3df9dad4825
      KeyName: NVirginiaF2021
      NetworkInterfaces:
        - GroupSet:
            - !Ref StackSecurityGroup
          AssociatePublicIpAddress: tru
          DeviceIndex: 0
          DeleteOnTermination: true
          SubnetId:
            Fn::ImportValue:
              !Sub ${NetworkStackName}-SubnetID

  StackSecurityGroup:
    Type: AWS::EC2::SecurityGroup
    Properties:
      GroupDescription: Enable HTTP ingress, ssh, and icmp
      VpcId:
        Fn::ImportValue:
          !Sub ${NetworkStackName}-VPCID
      SecurityGroupIngress:
        - IpProtocol: tcp
          FromPort: 80
          ToPort: 80
          CidrIp: 0.0.0.0/0
        - IpProtocol: tcp
          FromPort: 22
          ToPort: 22
          CidrIp: 0.0.0.0/0
        - IpProtocol: icmp
          FromPort: -1
          ToPort: -1
          CidrIp: 0.0.0.0/0
```

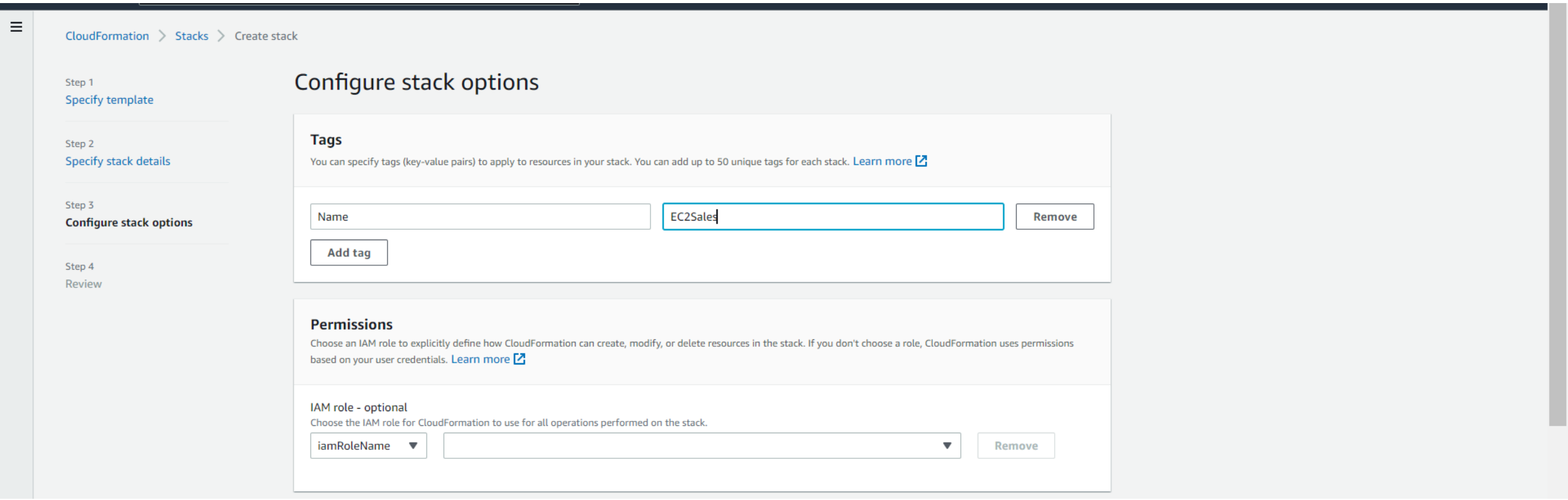
Create Stack 2 (EC2) on Stack 1

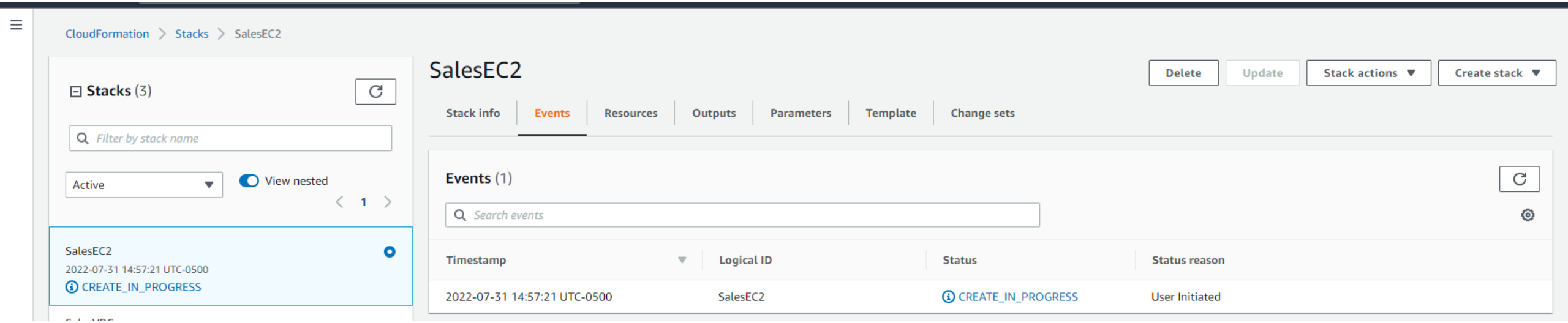


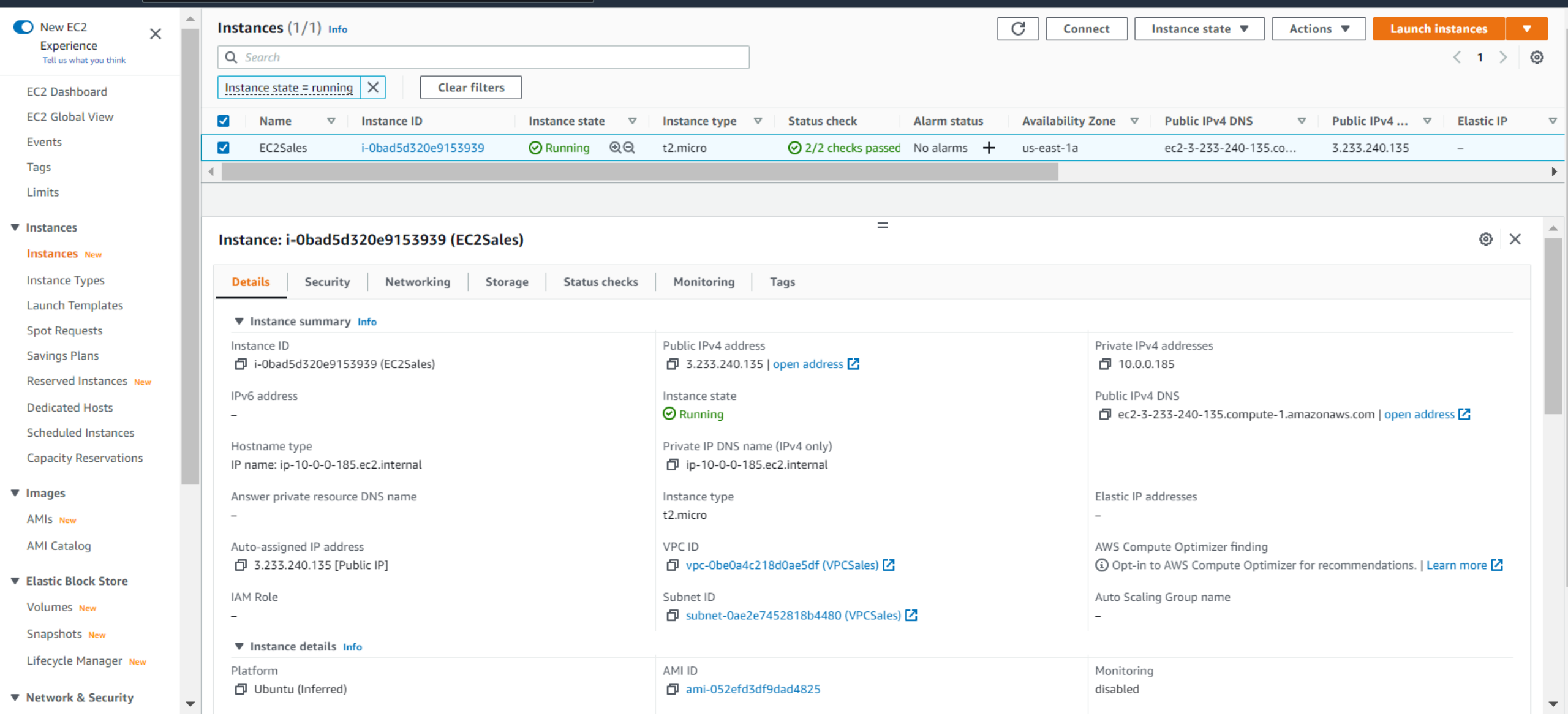
Upload Template

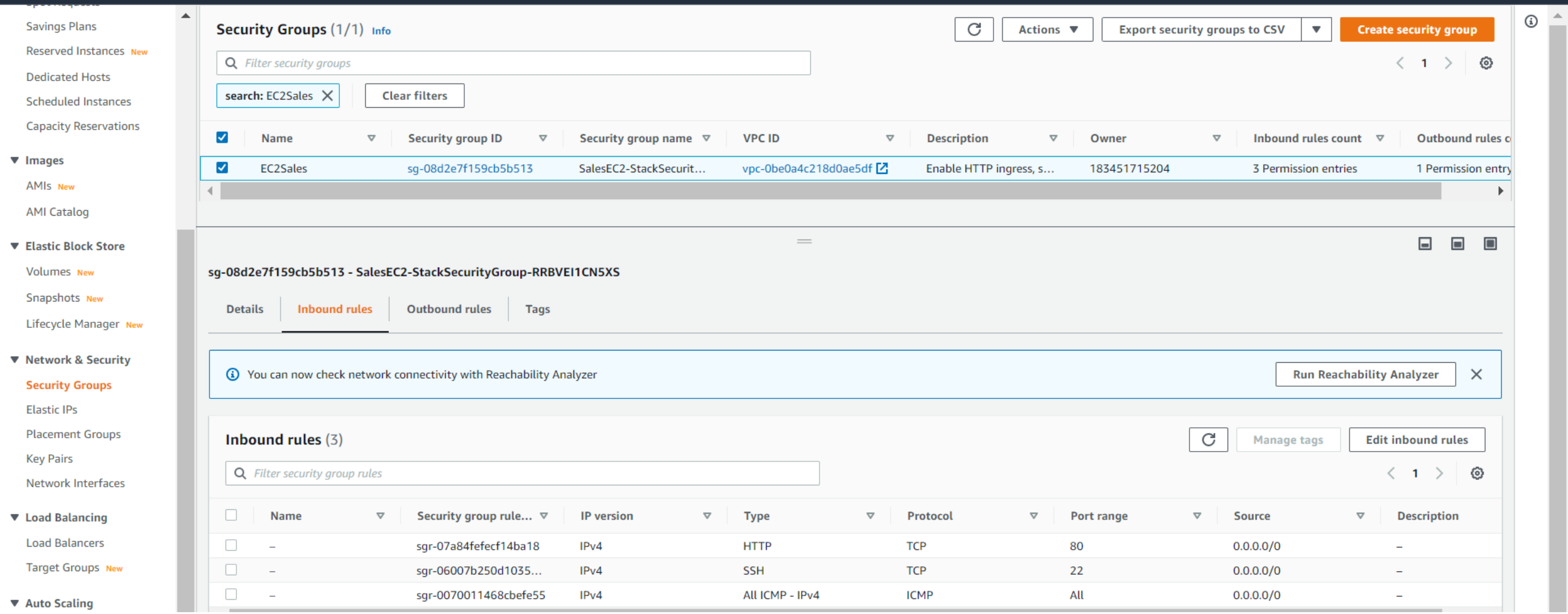



CloudFormation > Stacks > Create stack
Step 1
Specify template
Step 2
Specify stack details
Step 3
Configure stack options
Step 4
Review
Specify stack details
Stack name
Stack name
SalesEC2
Stack name can include letters (A-Z and a-z), numbers (0-9), and dashes (-).
Parameters
Parameters are defined in your template and allow you to input custom values when you create or update a stack.
NetworkStackName
SalesVPC
Cancel
Previous
Next


CloudFormation
Stacks
Create stack
Step 1
Specify template
Step 2
Specify stack details
Step 3
Configure stack options
Step 4
Review
Configure stack options
Tags
You can specify tags (key-value pairs) to apply to resources in your stack. You can add up to 50 unique tags for each stack. Learn more
Name
EC2Sales
Remove
Add tag
Permissions
Choose an IAM role to explicitly define how CloudFormation can create, modify, or delete resources in the stack. If you don't choose a role, CloudFormation uses permissions based on your user credentials. Learn more
IAM role - optional
Choose the IAM role for CloudFormation to use for all operations performed on the stack.
iamRoleName
Remove


CloudFormation
Stacks
SalesEC2
Stacks (3)
Filter by stack name
Active
View nested
1
SalesEC2
2022-07-31 14:57:21 UTC-0500
CREATE_IN_PROGRESS
SalesEC2
Delete
Update
Stack actions
Create stack
Stack info
Events
Resources
Outputs
Parameters
Template
Change sets
Events (1)
Search events
Timestamp
Logical ID
Status
Status reason
2022-07-31 14:57:21 UTC-0500
SalesEC2
CREATE_IN_PROGRESS
User Initiated

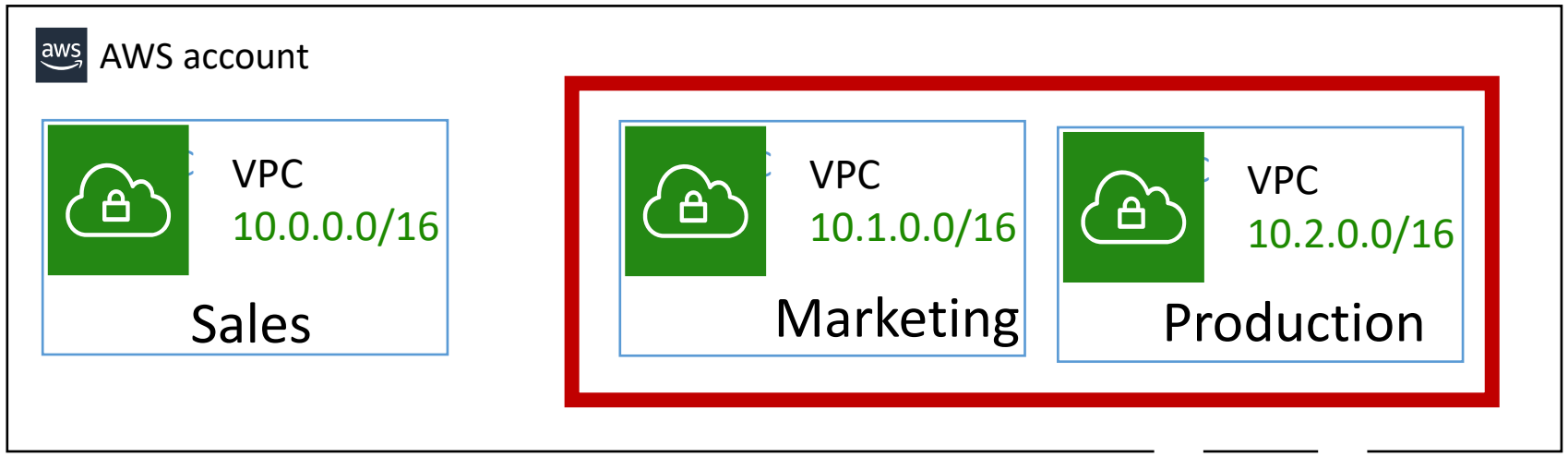

New EC2 Experience
Tell us what you think
EC2 Dashboard
EC2 Global View
Events
Tags
Limits
Instances
Instances New
Instance Types
Launch Templates
Spot Requests
Savings Plans
Reserved Instances New
Dedicated Hosts
Scheduled Instances
Capacity Reservations
Images
AMIs New
AMI Catalog
Elastic Block Store
Volumes New
Snapshots New
Lifecycle Manager New
Network & Security
Instances (1/1) Info
Search
Connect
Instance state
Actions
Launch instances
Instance state = running
Clear filters
Name
Instance ID
Instance state
Instance type
Status check
Alarm status
Availability Zone
Public IPv4 DNS
Public IPv4 ...
Elastic IP
EC2Sales
i-0bad5d320e9153939
Running
t2.micro
2/2 checks passed
No alarms
us-east-1a
ec2-3-233-240-135.co...
3.233.240.135
Instance: i-0bad5d320e9153939 (EC2Sales)
Details
Security
Networking
Storage
Status checks
Monitoring
Tags
Instance summary Info
Instance ID
i-0bad5d320e9153939 (EC2Sales)
IPv6 address
–
Hostname type
IP name: ip-10-0-0-185.ec2.internal
Answer private resource DNS name
–
Auto-assigned IP address
3.233.240.135 [Public IP]
IAM Role
–
Public IPv4 address
3.233.240.135 | open address
Instance state
Running
Private IP DNS name (IPv4 only)
ip-10-0-0-185.ec2.internal
Instance type
t2.micro
VPC ID
vpc-0be0a4c218d0ae5df (VPCSales)
Subnet ID
subnet-0ae2e7452818b4480 (VPCSales)
Private IPv4 addresses
10.0.0.185
Public IPv4 DNS
ec2-3-233-240-135.compute-1.amazonaws.com | open address
Elastic IP addresses
–
AWS Compute Optimizer finding
Opt-in to AWS Compute Optimizer for recommendations. | Learn more
Auto Scaling Group name
–
Instance details Info
Platform
Ubuntu (Inferred)
AMI ID
ami-052efd3df9dad4825
Monitoring
disabled




```
Pinging 3.233.240.135 with 32 bytes of data:
Reply from 3.233.240.135: bytes=32 time=41ms TTL=45
Reply from 3.233.240.135: bytes=32 time=45ms TTL=45
Reply from 3.233.240.135: bytes=32 time=43ms TTL=45
Reply from 3.233.240.135: bytes=32 time=41ms TTL=45

Ping statistics for 3.233.240.135:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 41ms, Maximum = 45ms, Average = 42ms
```

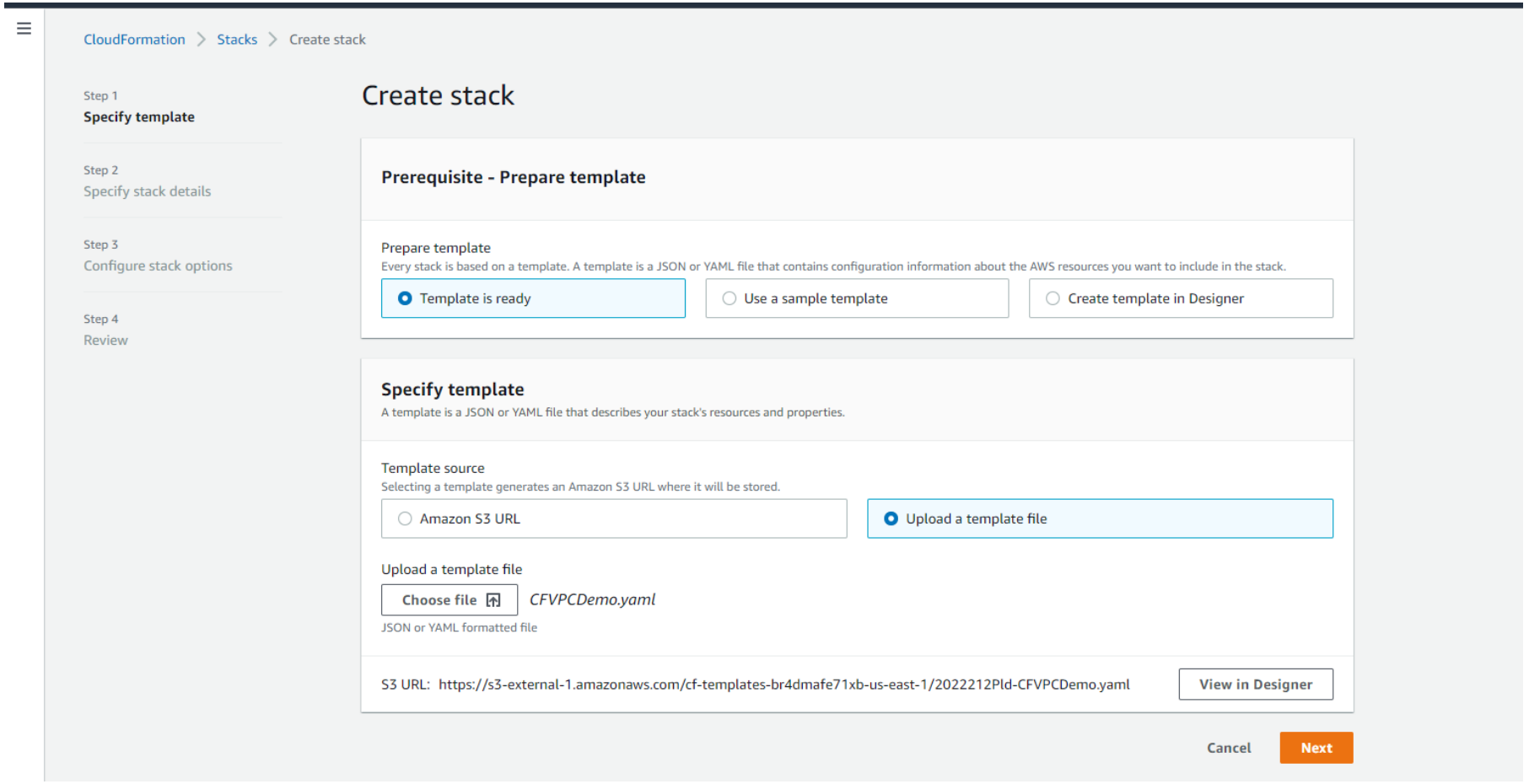
Check if you can make connection

Demo 4: Create Multiple VPCs + EC2s

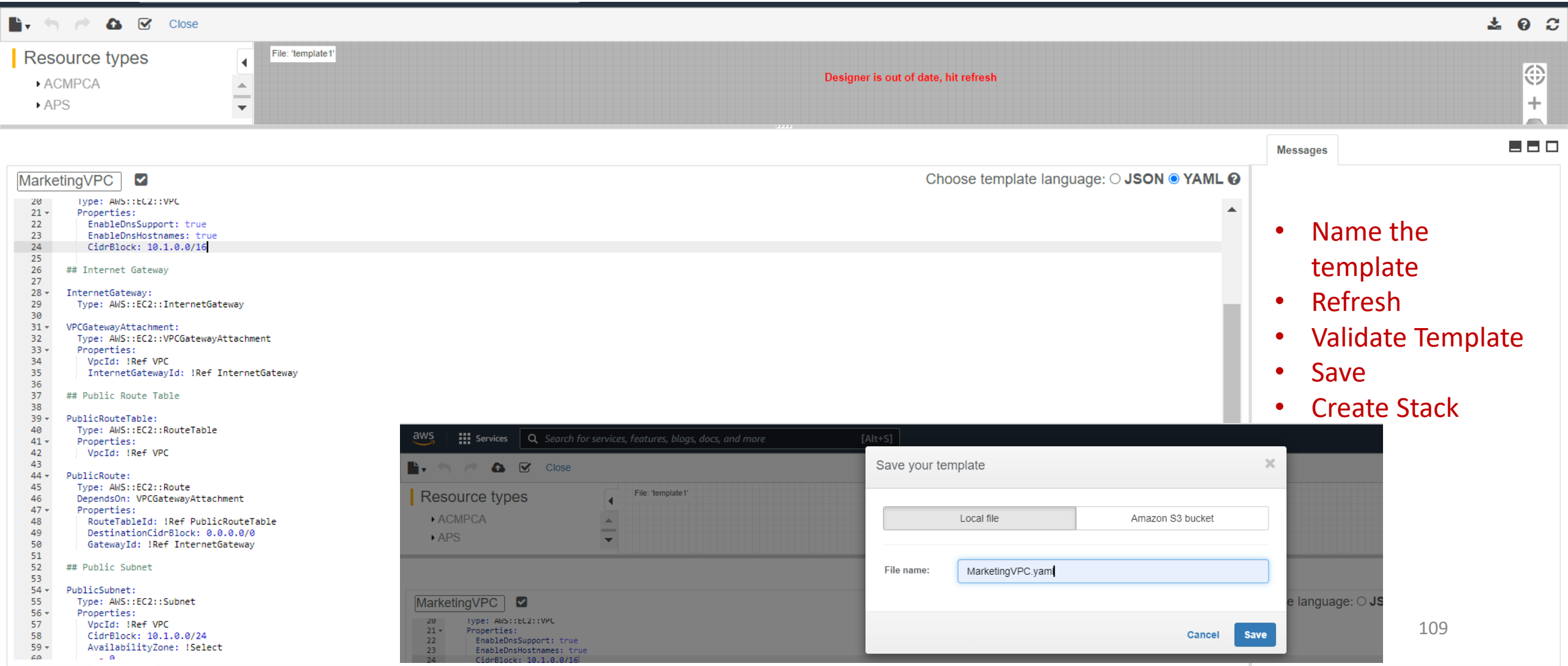


Create the 2 VPCs and an EC2 in each using the CloudFormation. You can either create the yaml documents or use Designer. I am going to use Designer.

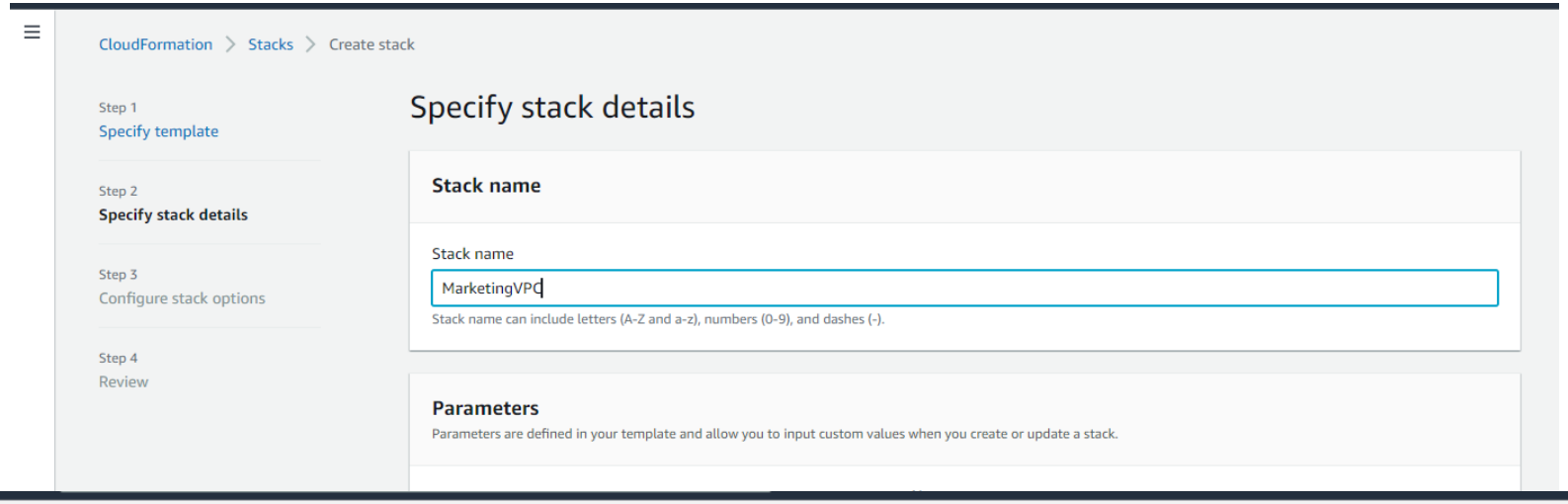
Creating Marketing VPC: Upload the original VPC template and edit using Designer

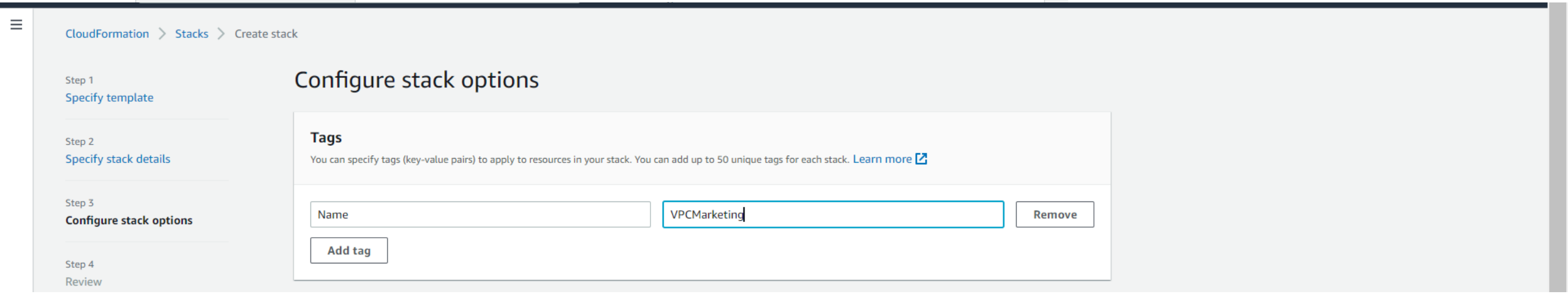


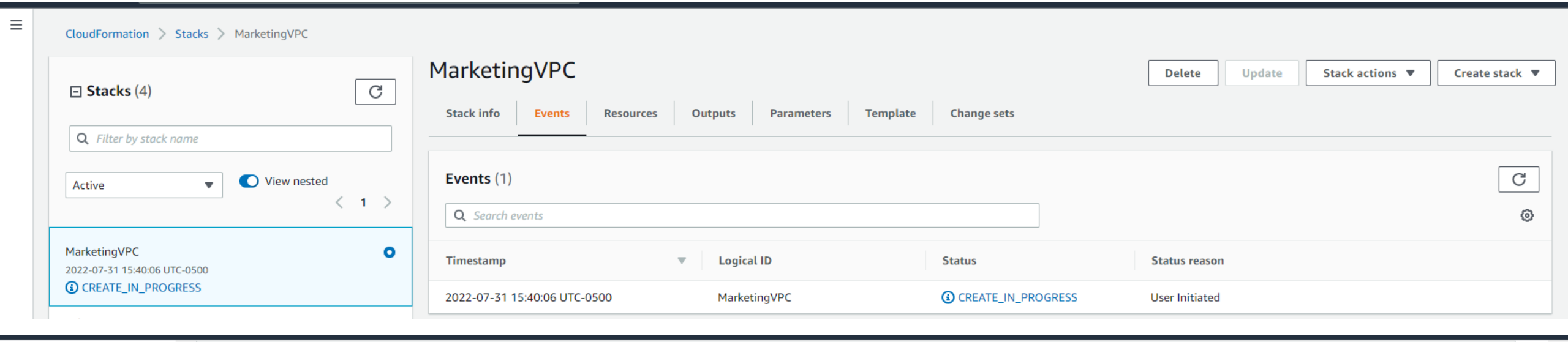
Update the VPC and Subnet CIDR blocks

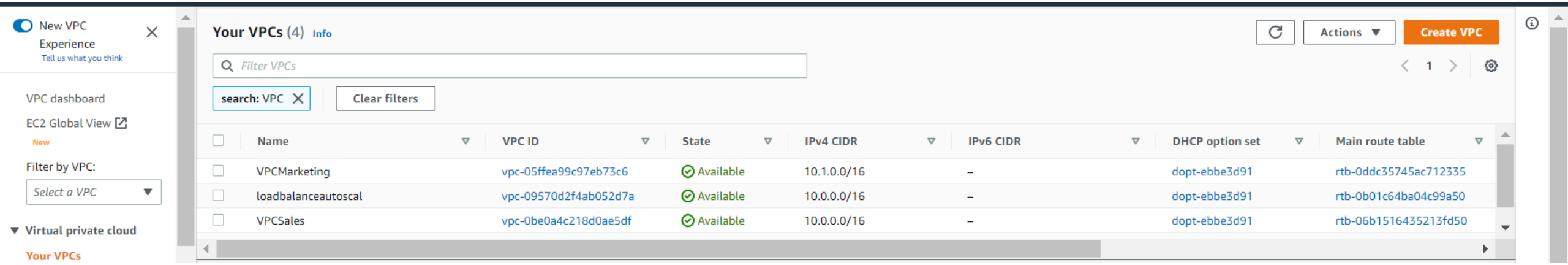


- Name the template
- Refresh
- Validate Template
- Save
- Create Stack

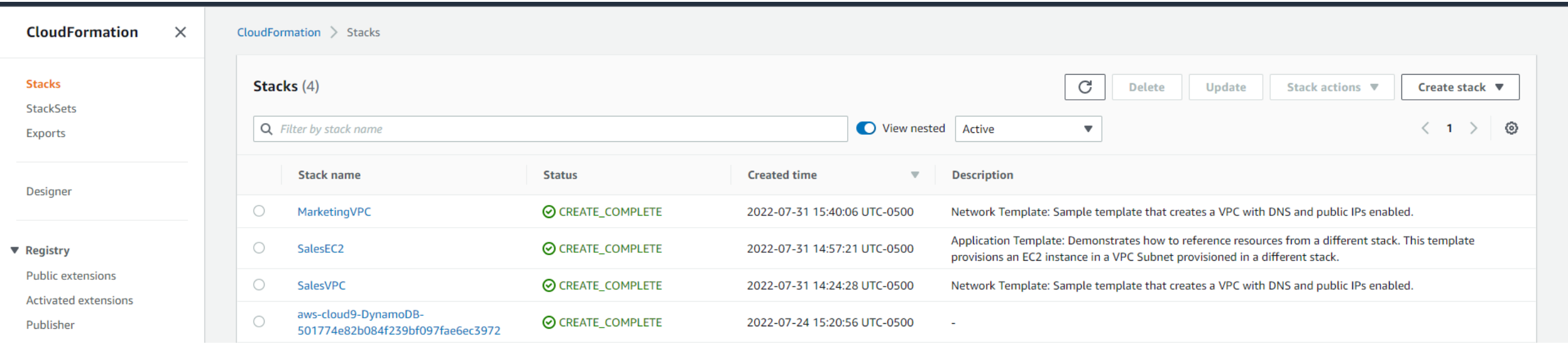

CloudFormation > Stacks > Create stack
Step 1
Specify template
Step 2
Specify stack details
Step 3
Configure stack options
Step 4
Review
Specify stack details
Stack name
Stack name
MarketingVPC
Stack name can include letters (A-Z and a-z), numbers (0-9), and dashes (-).
Parameters
Parameters are defined in your template and allow you to input custom values when you create or update a stack.

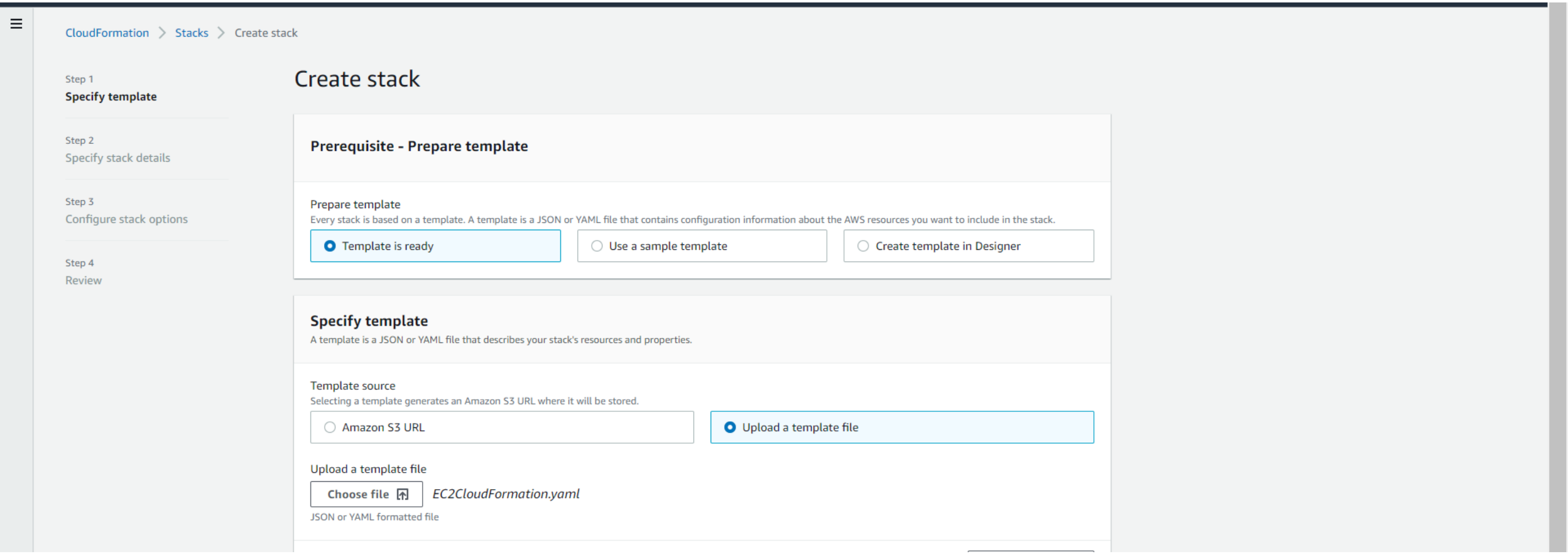

CloudFormation > Stacks > Create stack
Step 1
Specify template
Step 2
Specify stack details
Step 3
Configure stack options
Step 4
Review
Configure stack options
Tags
You can specify tags (key-value pairs) to apply to resources in your stack. You can add up to 50 unique tags for each stack. Learn more
Name
VPCMarketing
Remove
Add tag


CloudFormation > Stacks > MarketingVPC
Stacks (4)
Filter by stack name
Active
View nested
1
MarketingVPC
2022-07-31 15:40:06 UTC-0500
CREATE_IN_PROGRESS
MarketingVPC
Delete
Update
Stack actions
Create stack
Stack info
Events
Resources
Outputs
Parameters
Template
Change sets
Events (1)
Search events
Timestamp
Logical ID
Status
Status reason
2022-07-31 15:40:06 UTC-0500
MarketingVPC
CREATE_IN_PROGRESS
User Initiated

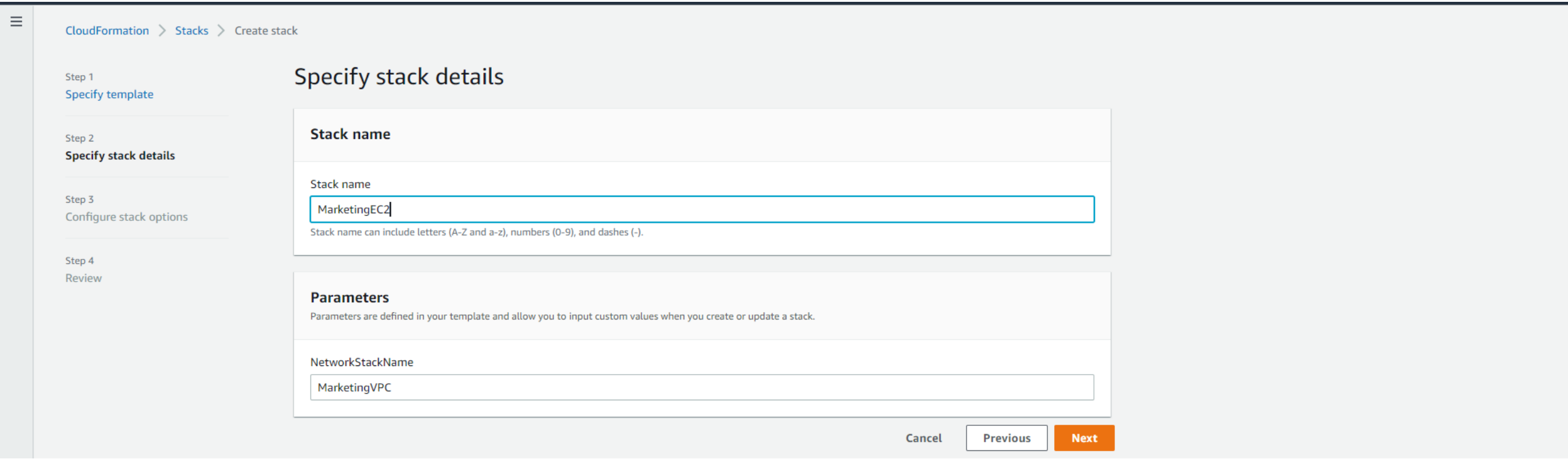

New VPC Experience
Tell us what you think
VPC dashboard
EC2 Global View
New
Filter by VPC:
Select a VPC
Virtual private cloud
Your VPCs
Your VPCs (4) Info
Actions
Create VPC
Filter VPCs
search: VPC
Clear filters
1
Name
VPC ID
State
IPv4 CIDR
IPv6 CIDR
DHCP option set
Main route table
VPCMarketing
vpc-05ffea99c97eb73c6
Available
10.1.0.0/16
–
dopt-ebbe3d91
rtb-0ddc35745ac712335
loadbalanceautoscal
vpc-09570d2f4ab052d7a
Available
10.0.0.0/16
–
dopt-ebbe3d91
rtb-0b01c64ba04c99a50
VPCSales
vpc-0be0a4c218d0ae5df
Available
10.0.0.0/16
–
dopt-ebbe3d91
rtb-06b1516435213fd50

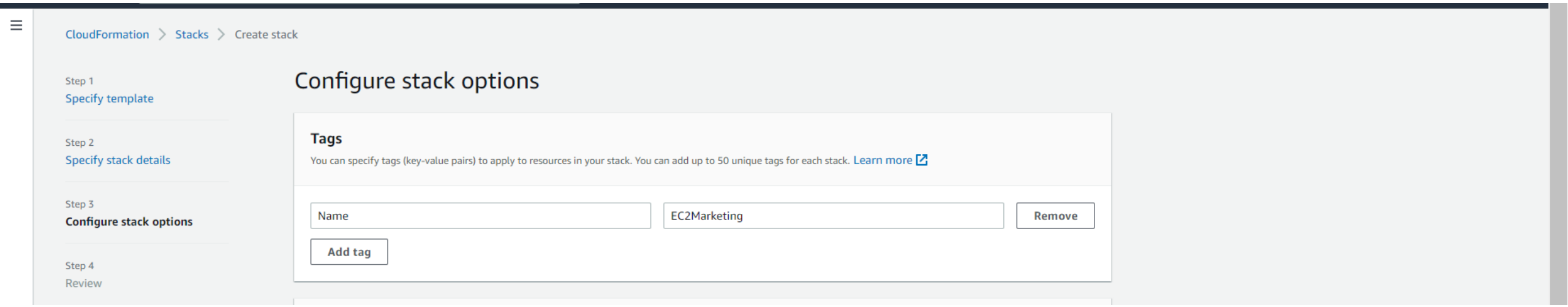
Create an EC2 in Marketing VPC using CloudFormation

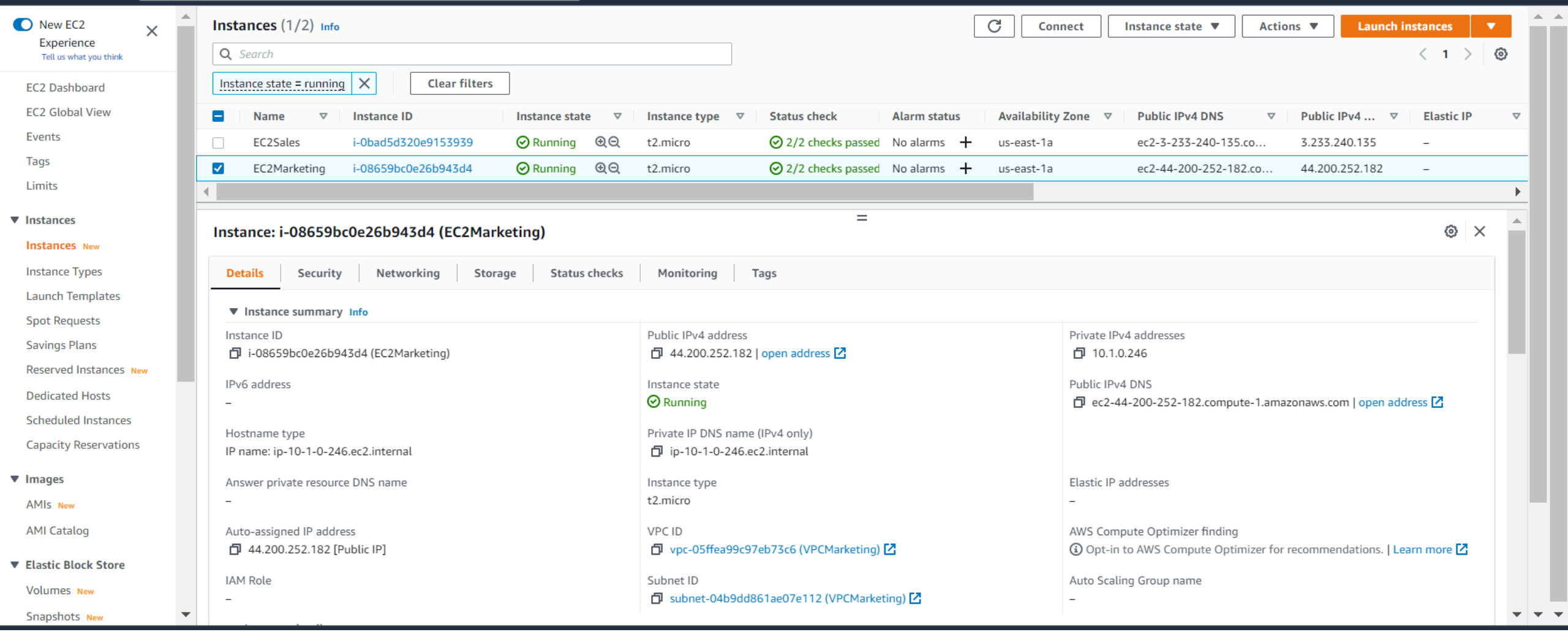




In Designer, check if anything needs to be changed. In this case, there is none so we can create the stack.


CloudFormation
Stacks
Create stack
Step 1
Specify template
Step 2
Specify stack details
Step 3
Configure stack options
Step 4
Review
Specify stack details
Stack name
Stack name
MarketingEC2
Stack name can include letters (A-Z and a-z), numbers (0-9), and dashes (-).
Parameters
Parameters are defined in your template and allow you to input custom values when you create or update a stack.
NetworkStackName
MarketingVPC
Cancel
Previous
Next

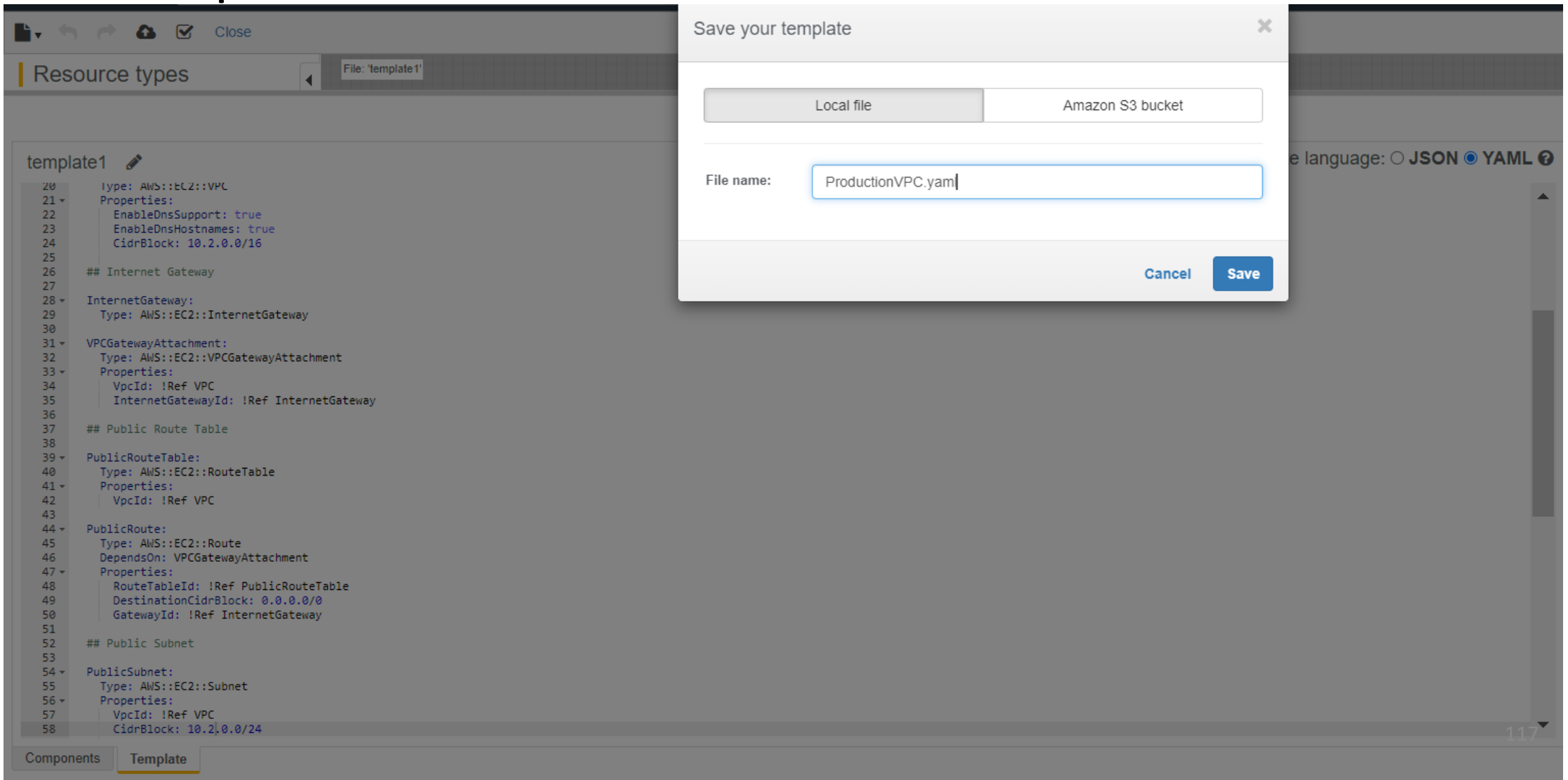

CloudFormation
Stacks
Create stack
Step 1
Specify template
Step 2
Specify stack details
Step 3
Configure stack options
Step 4
Review
Configure stack options
Tags
You can specify tags (key-value pairs) to apply to resources in your stack. You can add up to 50 unique tags for each stack. Learn more
Name
EC2Marketing
Remove
Add tag

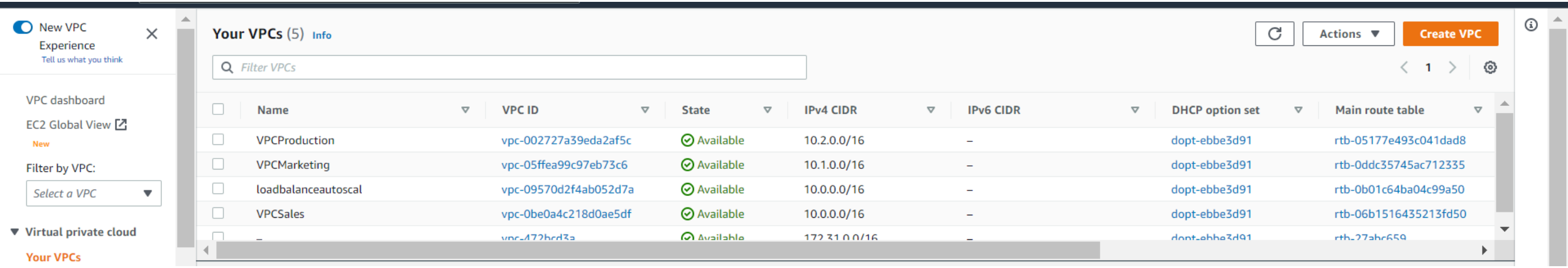


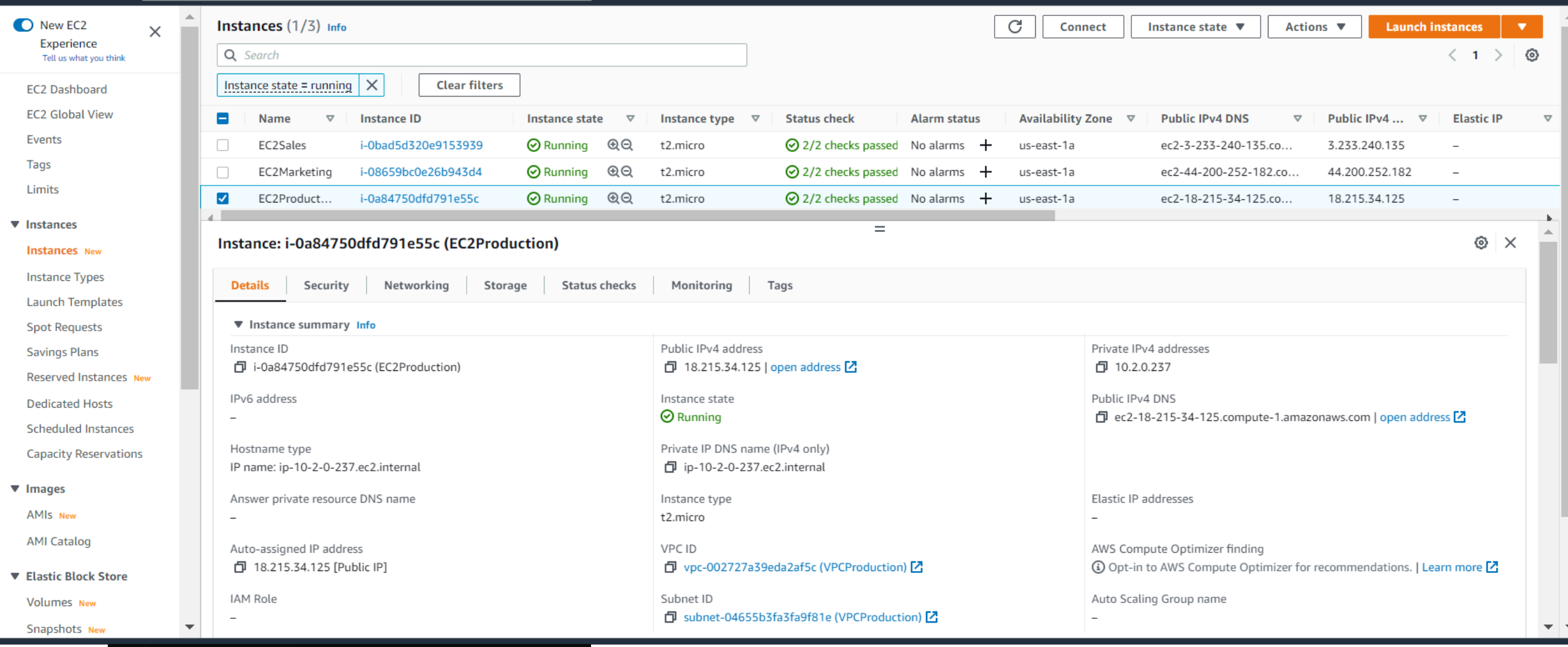

```
Pinging 44.200.252.182 with 32 bytes of data:
Reply from 44.200.252.182: bytes=32 time=58ms TTL=46
Reply from 44.200.252.182: bytes=32 time=41ms TTL=46
Reply from 44.200.252.182: bytes=32 time=41ms TTL=46
Reply from 44.200.252.182: bytes=32 time=40ms TTL=46

Ping statistics for 44.200.252.182:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 40ms, Maximum = 58ms, Average = 45ms
```

Check connectivity

Create VPC and EC2 for Production with the same process



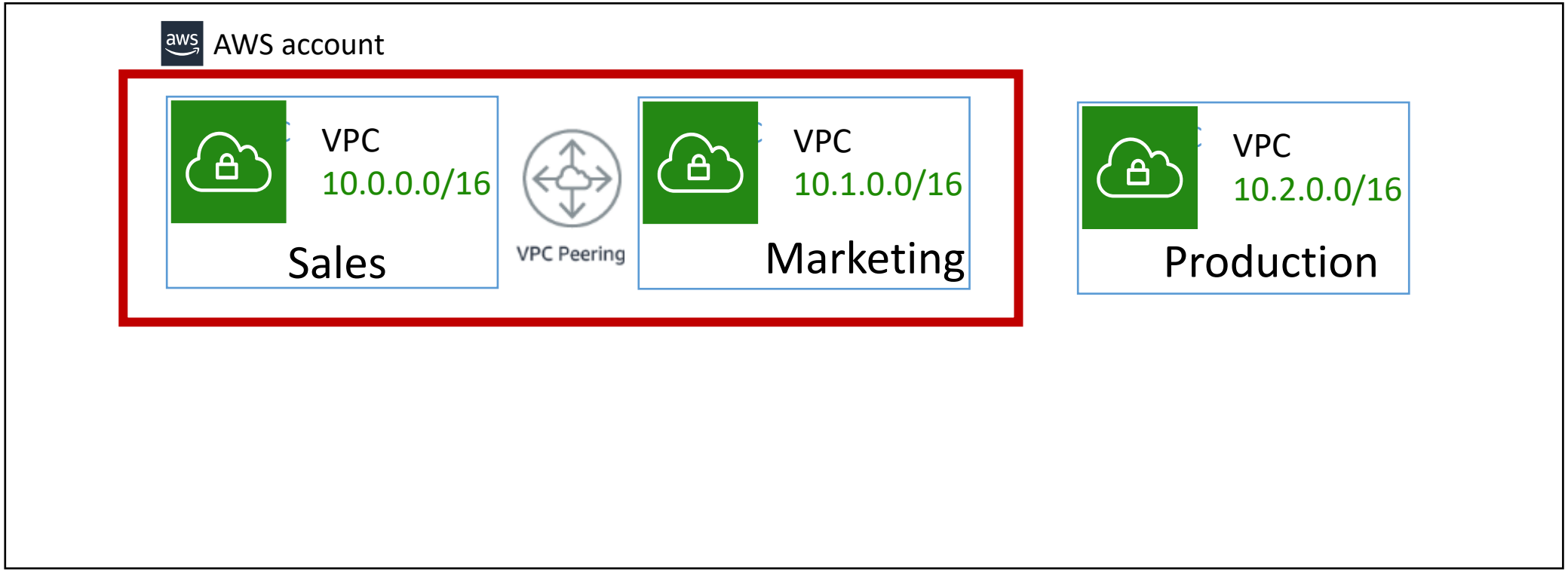

New VPC Experience
Tell us what you think
VPC dashboard
EC2 Global View
New
Filter by VPC:
Select a VPC
Virtual private cloud
Your VPCs
Your VPCs (5) Info
Actions
Create VPC
Filter VPCs
1
Name
VPC ID
State
IPv4 CIDR
IPv6 CIDR
DHCP option set
Main route table
VPCProduction vpc-002727a39eda2af5c Available 10.2.0.0/16 – dopt-ebbe3d91 rtb-05177e493c041dad8
VPCMarketing vpc-05ffea99c97eb73c6 Available 10.1.0.0/16 – dopt-ebbe3d91 rtb-0ddc35745ac712335
loadbalanceautoscal vpc-09570d2f4ab052d7a Available 10.0.0.0/16 – dopt-ebbe3d91 rtb-0b01c64ba04c99a50
VPCSales vpc-0be0a4c218d0ae5df Available 10.0.0.0/16 – dopt-ebbe3d91 rtb-06b1516435213fd50



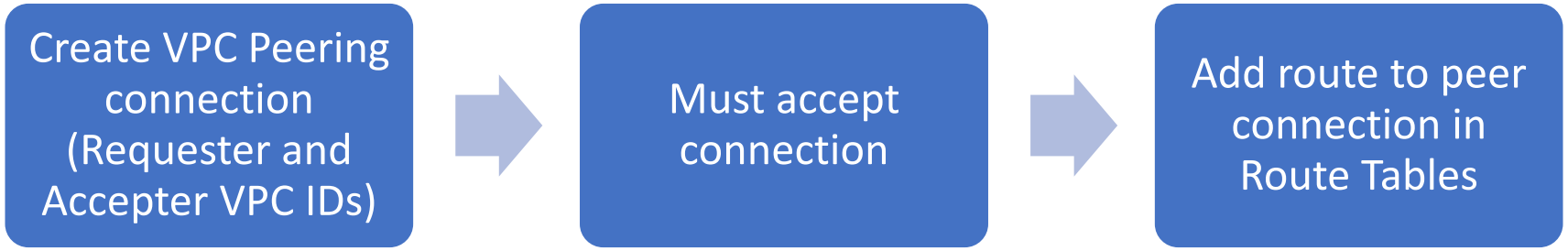

```
C:\Windows\System32>ping 18.215.34.125

Pinging 18.215.34.125 with 32 bytes of data:
Reply from 18.215.34.125: bytes=32 time=42ms TTL=46
Reply from 18.215.34.125: bytes=32 time=43ms TTL=46
Reply from 18.215.34.125: bytes=32 time=41ms TTL=46
Reply from 18.215.34.125: bytes=32 time=42ms TTL=46

Ping statistics for 18.215.34.125:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 41ms, Maximum = 43ms, Average = 42ms
```

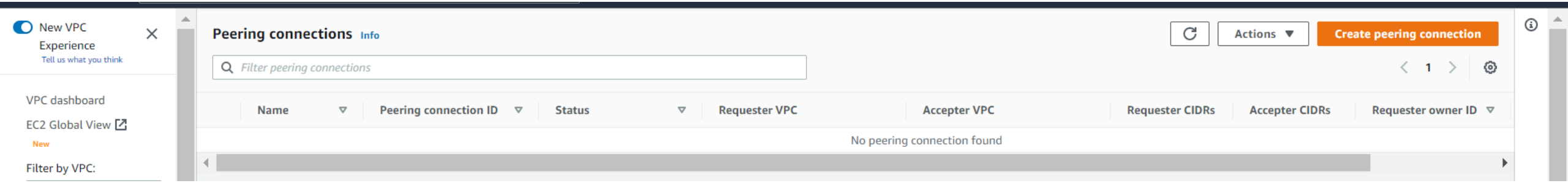
Check connectivity

Demo 5: VPC Peering



VPC Peering steps




New VPC Experience
Tell us what you think
VPC dashboard
EC2 Global View
New
Filter by VPC:
Peering connections Info
Actions
Create peering connection
Filter peering connections
1
Name
Peering connection ID
Status
Requester VPC
Accepter VPC
Requester CIDRs
Accepter CIDRs
Requester owner ID
No peering connection found

VPC > Peering connections > Create peering connection

Create peering connection

A VPC peering connection is a networking connection between two VPCs that enables you to route traffic between them privately. Info

Peering connection settings

Name - *optional*
Create a tag with a key of 'Name' and a value that you specify.

Sales-Marketing

Select a local VPC to peer with

VPC ID (Requester)

vpc-0be0a4c218d0ae5df (VPCSales)

VPC CIDRs for vpc-0be0a4c218d0ae5df (VPCSales)

CIDR	Status	Status reason
10.0.0.0/16	Associated	-

Select another VPC to peer with

Account

- (•) My account
- ( ) Another account

Region

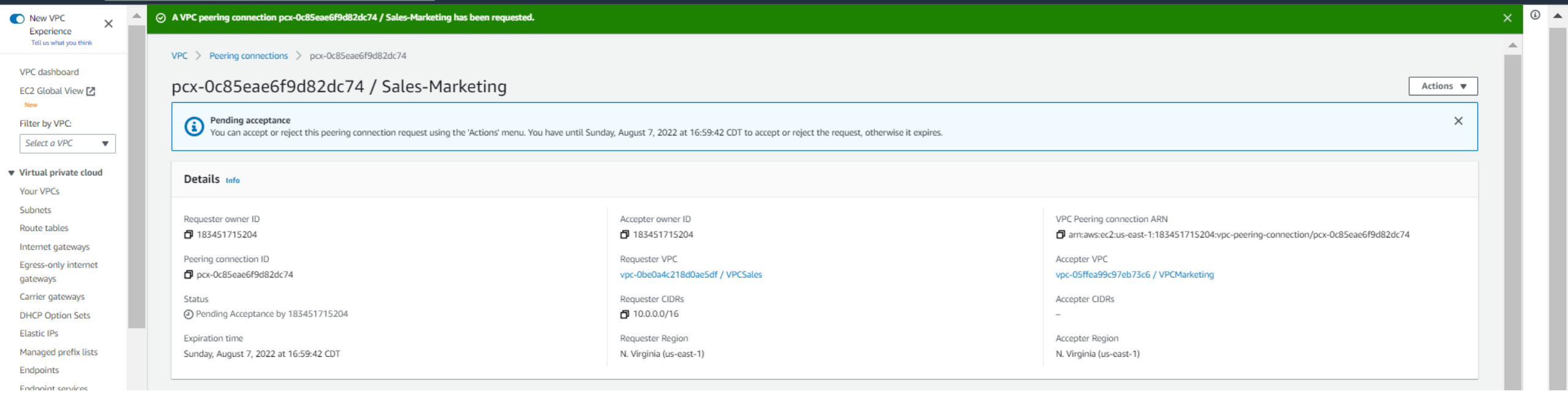
- (•) This Region (us-east-1)
- ( ) Another Region

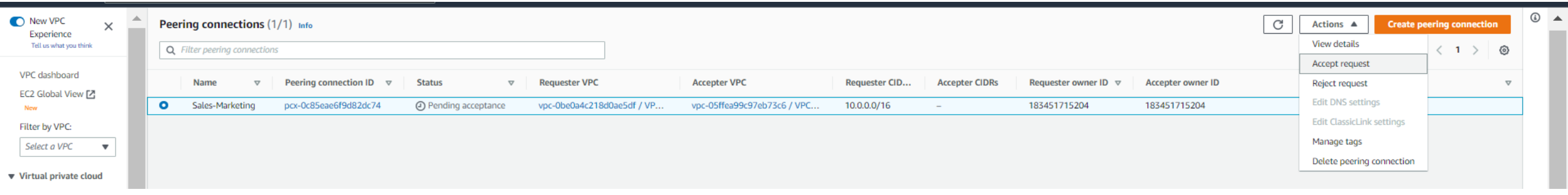
VPC ID (Accepter)

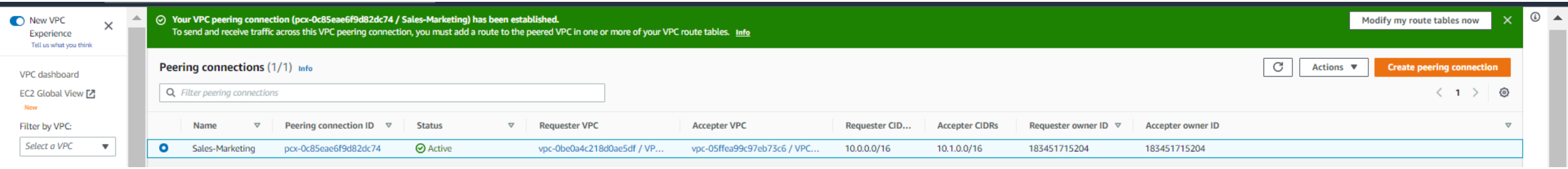
vpc-05ffea99c97eb73c6 (VPCMarketing)

VPC CIDRs for vpc-05ffea99c97eb73c6 (VPCMarketing)

CIDR	Status	Status reason
10.1.0.0/16	Associated	-


New VPC Experience
Tell us what you think
A VPC peering connection pcx-0c85eae6f9d82dc74 / Sales-Marketing has been requested.
VPC dashboard
EC2 Global View
New
Filter by VPC:
Select a VPC
Virtual private cloud
Your VPCs
Subnets
Route tables
Internet gateways
Egress-only internet gateways
Carrier gateways
DHCP Option Sets
Elastic IPs
Managed prefix lists
Endpoints
Endpoint services
VPC > Peering connections > pcx-0c85eae6f9d82dc74
pcx-0c85eae6f9d82dc74 / Sales-Marketing
Actions
Pending acceptance
You can accept or reject this peering connection request using the 'Actions' menu. You have until Sunday, August 7, 2022 at 16:59:42 CDT to accept or reject the request, otherwise it expires.
Details Info
Requester owner ID
183451715204
Peering connection ID
pcx-0c85eae6f9d82dc74
Status
Pending Acceptance by 183451715204
Expiration time
Sunday, August 7, 2022 at 16:59:42 CDT
Accepter owner ID
183451715204
Requester VPC
vpc-0be0a4c218d0ae5df / VPCSales
Requester CIDRs
10.0.0.0/16
Requester Region
N. Virginia (us-east-1)
VPC Peering connection ARN
arn:aws:ec2:us-east-1:183451715204:vpc-peering-connection/pcx-0c85eae6f9d82dc74
Accepter VPC
vpc-05ffea99c97eb73c6 / VPCMarketing
Accepter CIDRs
–
Accepter Region
N. Virginia (us-east-1)

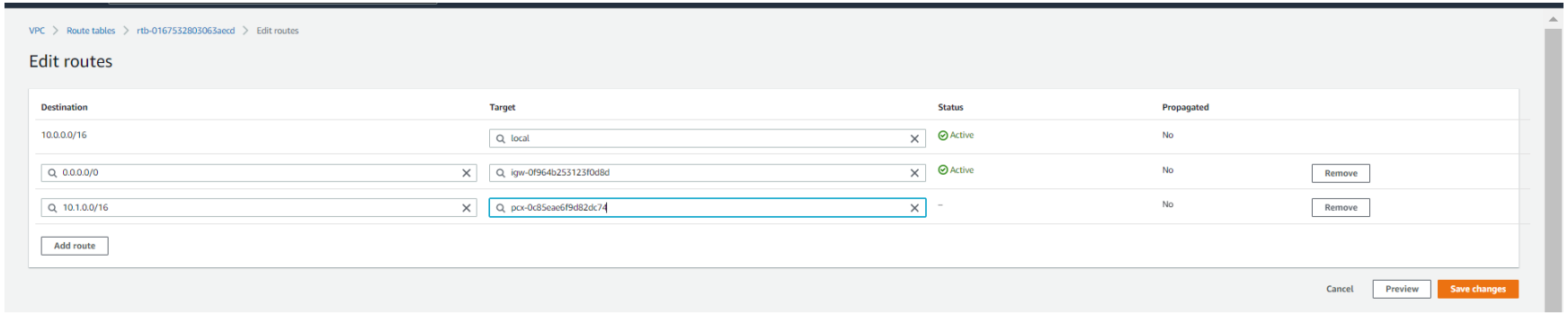

New VPC Experience
Tell us what you think
VPC dashboard
EC2 Global View
New
Filter by VPC:
Select a VPC
Virtual private cloud
Peering connections (1/1) Info
Filter peering connections
Actions
Create peering connection
View details
Accept request
Reject request
Edit DNS settings
Edit ClassicLink settings
Manage tags
Delete peering connection
Name
Peering connection ID
Status
Requester VPC
Accepter VPC
Requester CID...
Accepter CIDRs
Requester owner ID
Accepter owner ID
Sales-Marketing
pcx-0c85eae6f9d82dc74
Pending acceptance
vpc-0be0a4c218d0ae5df / VP...
vpc-05ffea99c97eb73c6 / VPC...
10.0.0.0/16
–
183451715204
183451715204


New VPC Experience
Tell us what you think
Your VPC peering connection (pcx-0c85eae6f9d82dc74 / Sales-Marketing) has been established.
To send and receive traffic across this VPC peering connection, you must add a route to the peered VPC in one or more of your VPC route tables. Info
Modify my route tables now
VPC dashboard
EC2 Global View
New
Filter by VPC:
Select a VPC
Peering connections (1/1) Info
Filter peering connections
Actions
Create peering connection
Name
Peering connection ID
Status
Requester VPC
Accepter VPC
Requester CID...
Accepter CIDRs
Requester owner ID
Accepter owner ID
Sales-Marketing
pcx-0c85eae6f9d82dc74
Active
vpc-0be0a4c218d0ae5df / VP...
vpc-05ffea99c97eb73c6 / VPC...
10.0.0.0/16
10.1.0.0/16
183451715204
183451715204

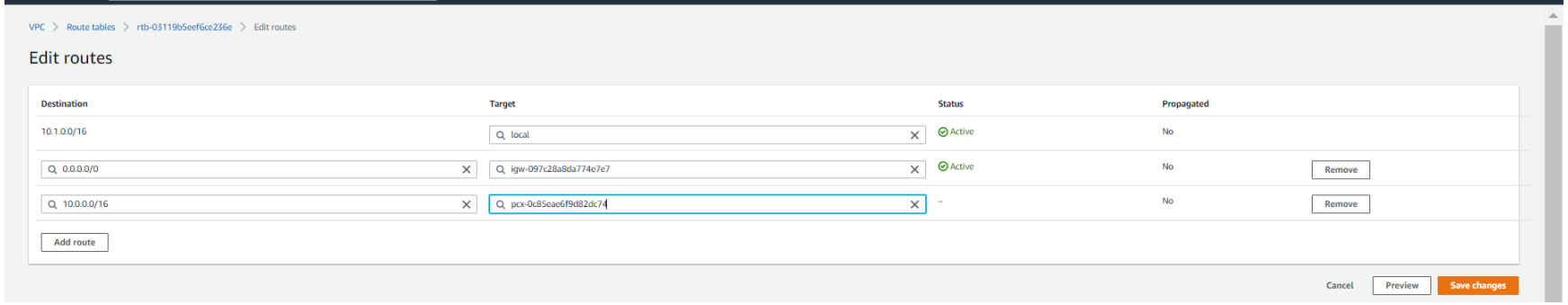
Update Route Tables

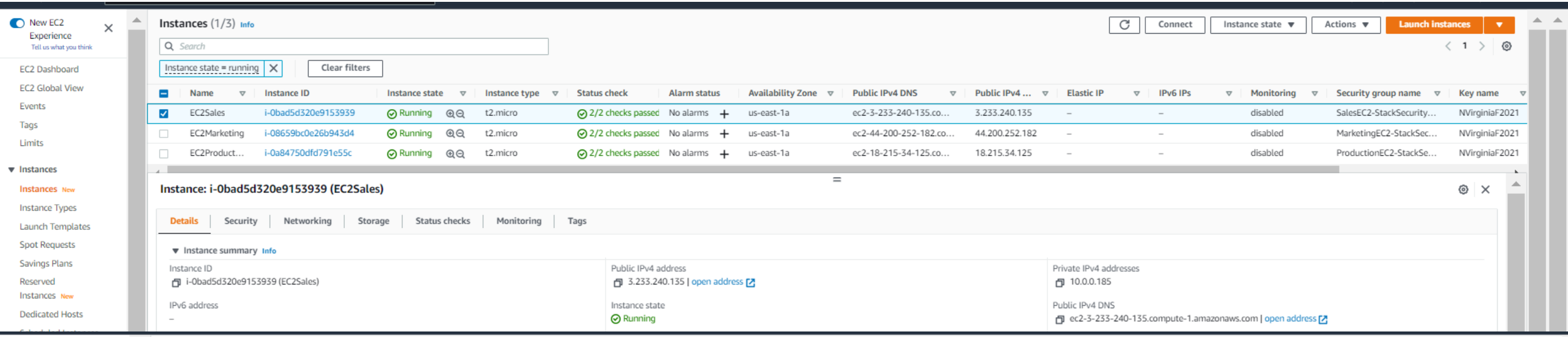
Update the route tables of each subnets to route to **the peer connection** for private IP destinations to the other VPC (VPC CIDR block).

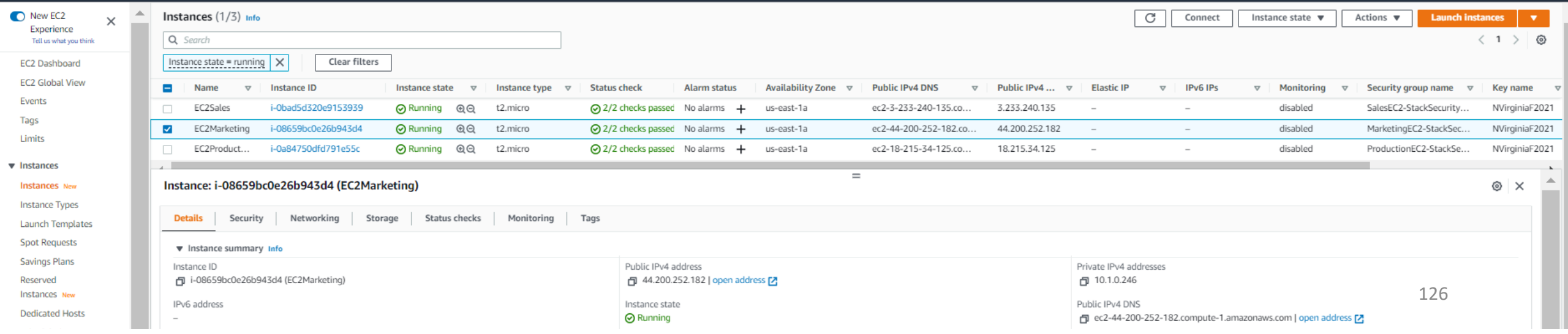
Sales to Marketing



Marketing to Sales




New EC2 Experience
Tell us what you think
EC2 Dashboard
EC2 Global View
Events
Tags
Limits
Instances
Instances New
Instance Types
Launch Templates
Spot Requests
Savings Plans
Reserved Instances New
Dedicated Hosts
Instances (1/3) Info
Search
Connect
Instance state
Actions
Launch instances
Instance state = running
Clear filters
Name
Instance ID
Instance state
Instance type
Status check
Alarm status
Availability Zone
Public IPv4 DNS
Public IPv4 ...
Elastic IP
IPv6 IPs
Monitoring
Security group name
Key name
EC2Sales
i-0bad5d320e9153939
Running
t2.micro
2/2 checks passed
No alarms
us-east-1a
ec2-3-233-240-135.co...
3.233.240.135
disabled
SalesEC2-StackSecurity...
NVirginiaF2021
EC2Marketing
i-08659bc0e26b943d4
Running
t2.micro
2/2 checks passed
No alarms
us-east-1a
ec2-44-200-252-182.co...
44.200.252.182
disabled
MarketingEC2-StackSec...
NVirginiaF2021
EC2Product...
i-0a84750dfd791e55c
Running
t2.micro
2/2 checks passed
No alarms
us-east-1a
ec2-18-215-34-125.co...
18.215.34.125
disabled
ProductionEC2-StackSe...
NVirginiaF2021
Instance: i-0bad5d320e9153939 (EC2Sales)
Details
Security
Networking
Storage
Status checks
Monitoring
Tags
Instance summary Info
Instance ID
i-0bad5d320e9153939 (EC2Sales)
IPv6 address
–
Public IPv4 address
3.233.240.135 | open address
Instance state
Running
Private IPv4 addresses
10.0.0.185
Public IPv4 DNS
ec2-3-233-240-135.compute-1.amazonaws.com | open address

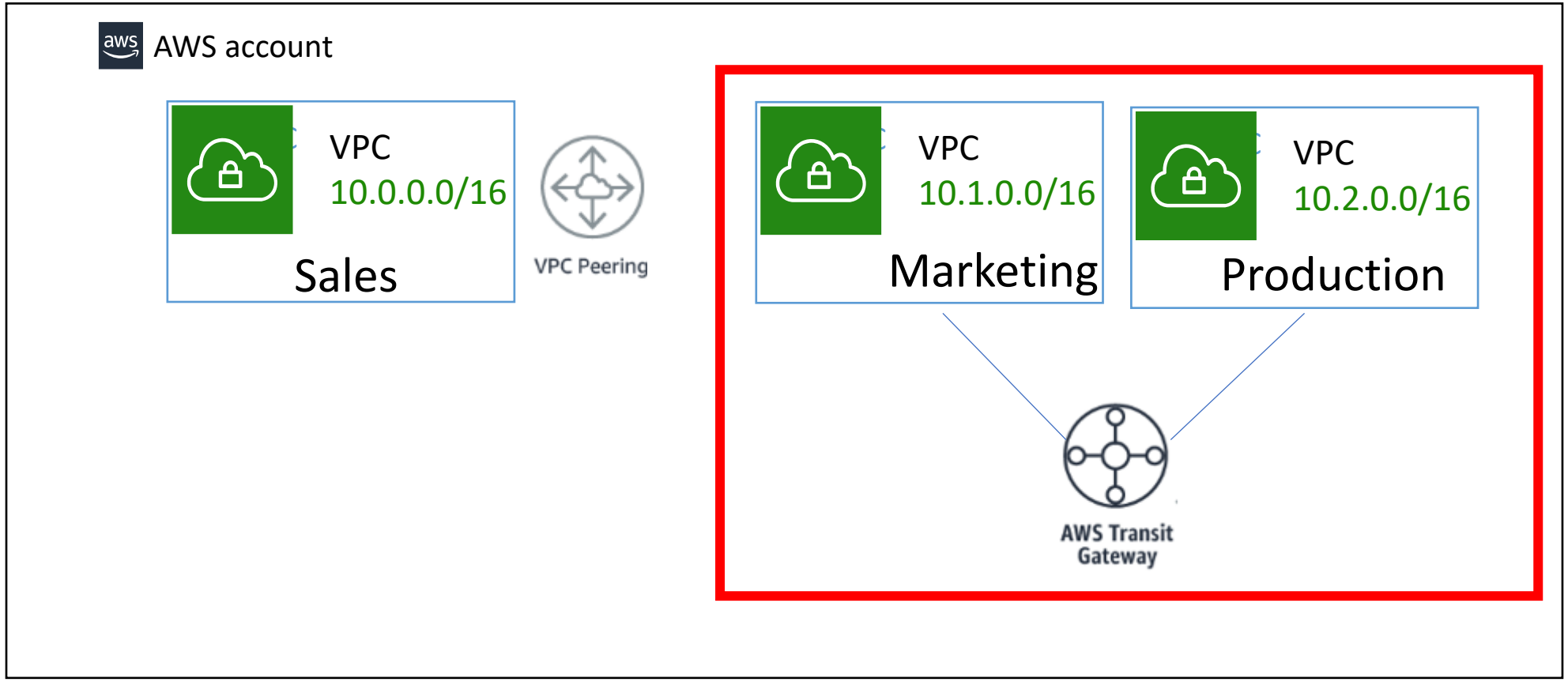

New EC2 Experience
Tell us what you think
EC2 Dashboard
EC2 Global View
Events
Tags
Limits
Instances
Instances New
Instance Types
Launch Templates
Spot Requests
Savings Plans
Reserved Instances New
Dedicated Hosts
Instances (1/3) Info
Search
Connect
Instance state
Actions
Launch instances
Instance state = running
Clear filters
Name
Instance ID
Instance state
Instance type
Status check
Alarm status
Availability Zone
Public IPv4 DNS
Public IPv4 ...
Elastic IP
IPv6 IPs
Monitoring
Security group name
Key name
EC2Sales
i-0bad5d320e9153939
Running
t2.micro
2/2 checks passed
No alarms
us-east-1a
ec2-3-233-240-135.co...
3.233.240.135
disabled
SalesEC2-StackSecurity...
NVirginiaF2021
EC2Marketing
i-08659bc0e26b943d4
Running
t2.micro
2/2 checks passed
No alarms
us-east-1a
ec2-44-200-252-182.co...
44.200.252.182
disabled
MarketingEC2-StackSec...
NVirginiaF2021
EC2Product...
i-0a84750dfd791e55c
Running
t2.micro
2/2 checks passed
No alarms
us-east-1a
ec2-18-215-34-125.co...
18.215.34.125
disabled
ProductionEC2-StackSe...
NVirginiaF2021
Instance: i-08659bc0e26b943d4 (EC2Marketing)
Details
Security
Networking
Storage
Status checks
Monitoring
Tags
Instance summary Info
Instance ID
i-08659bc0e26b943d4 (EC2Marketing)
IPv6 address
–
Public IPv4 address
44.200.252.182 | open address
Instance state
Running
Private IPv4 addresses
10.1.0.246
Public IPv4 DNS
ec2-44-200-252-182.compute-1.amazonaws.com | open address

```
ubuntu@ip-10-1-0-246:~$ ping -c3 10.0.0.185
PING 10.0.0.185 (10.0.0.185) 56(84) bytes of data.
64 bytes from 10.0.0.185: icmp_seq=1 ttl=64 time=0.572 ms
64 bytes from 10.0.0.185: icmp_seq=2 ttl=64 time=0.617 ms
64 bytes from 10.0.0.185: icmp_seq=3 ttl=64 time=0.703 ms

--- 10.0.0.185 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2051ms
rtt min/avg/max/mdev = 0.572/0.630/0.703/0.054 ms
```

```
ubuntu@ip-10-0-0-185:~$ ping -c3 10.1.0.246
PING 10.1.0.246 (10.1.0.246) 56(84) bytes of data.
64 bytes from 10.1.0.246: icmp_seq=1 ttl=64 time=0.570 ms
64 bytes from 10.1.0.246: icmp_seq=2 ttl=64 time=0.678 ms
64 bytes from 10.1.0.246: icmp_seq=3 ttl=64 time=1.29 ms

--- 10.1.0.246 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2027ms
rtt min/avg/max/mdev = 0.570/0.845/1.287/0.315 ms
ubuntu@ip-10-0-0-185:~$
```

Ping using private IP address must work as if in the same VPC.

Demo 6: Transit Gateway



Transit Gateway Steps

- Transit Gateway
  - Create Transit Gateway
  - Set attachments (1 attachment per VPC, VPN, or Peering Connection from other regions)
    - is the **connection between resources** like VPC, VPN, Direct Connect, **and the TGW**
    - A transit gateway attachment is both a source and a destination of packets.
  - Set Transit Gateway Route Tables
    - You can check the association and propagation while creating TG
    - associate a transit gateway route table with a transit gateway attachment
    - Use route propagation to add a route from an attachment to a route table.
- Update VPC Route Tables
- https://docs.aws.amazon.com/vpc/latest/tgw/what-is-transit-gateway.html

- When attaching a VPC to a transit gateway, must enable one or more Availability Zones to be used by the transit gateway to route traffic to resources in the VPC subnets.
  - To enable each Availability Zone, you specify exactly one subnet.
  - The transit gateway places a network interface in that subnet using one IP address from the subnet.
  - After an Availability Zone is enabled, traffic can be routed to all subnets in that zone, not just the specified subnet.
  - Resources that reside in Availability Zones where there is no transit gateway attachment cannot reach the transit gateway.
- Recommended to enable multiple AZs to ensure availability.

A transit gateway acts as a Regional virtual router for traffic flowing between virtual private clouds (VPCs) and on-premises networks.

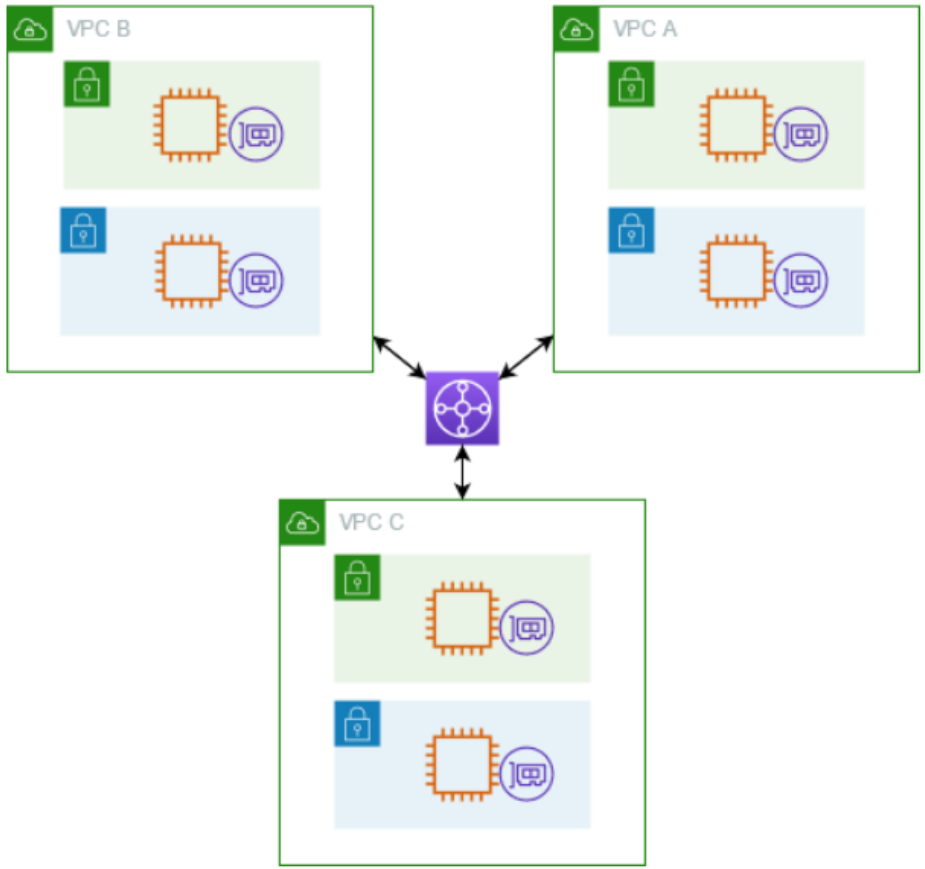
Route table association

- You can associate a transit gateway attachment with a single route table. Each route table can be associated with zero to many attachments and can forward packets to other attachments.

Route propagation

- Each attachment comes with routes that can be installed in one or more transit gateway route tables. When an attachment is propagated to a transit gateway route table, these routes are installed in the route table.
- For a VPC attachment, the CIDR blocks of the VPC are propagated to the transit gateway route table.

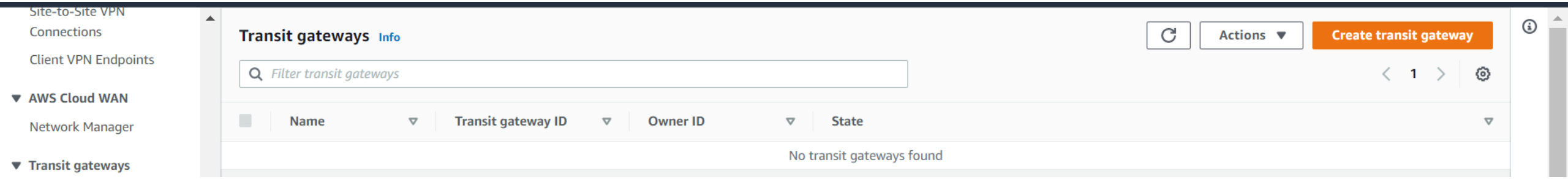
https://docs.aws.amazon.com/vpc/latest/tgw/how-transit-gateways-work.html

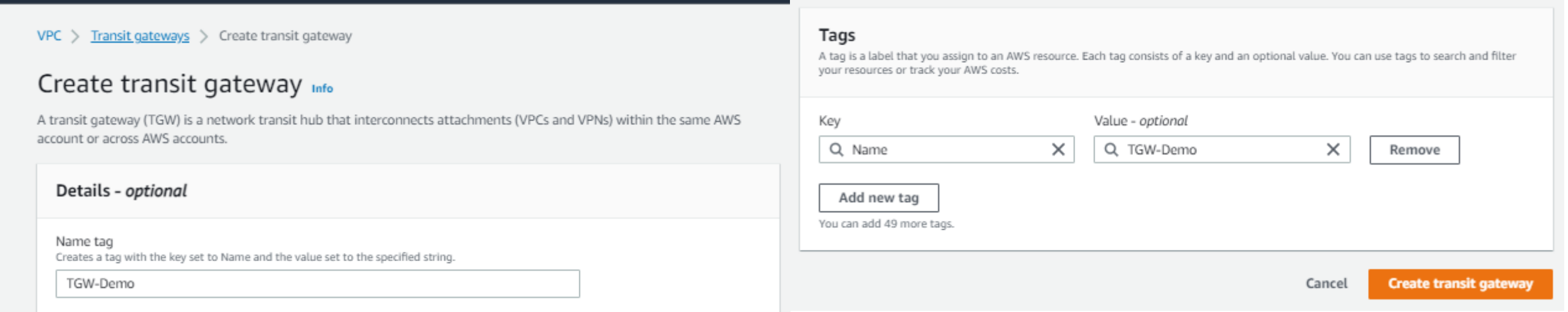


The CIDR blocks for each VPC propagate to the route table. Therefore, each attachment can route packets to the other two attachments.

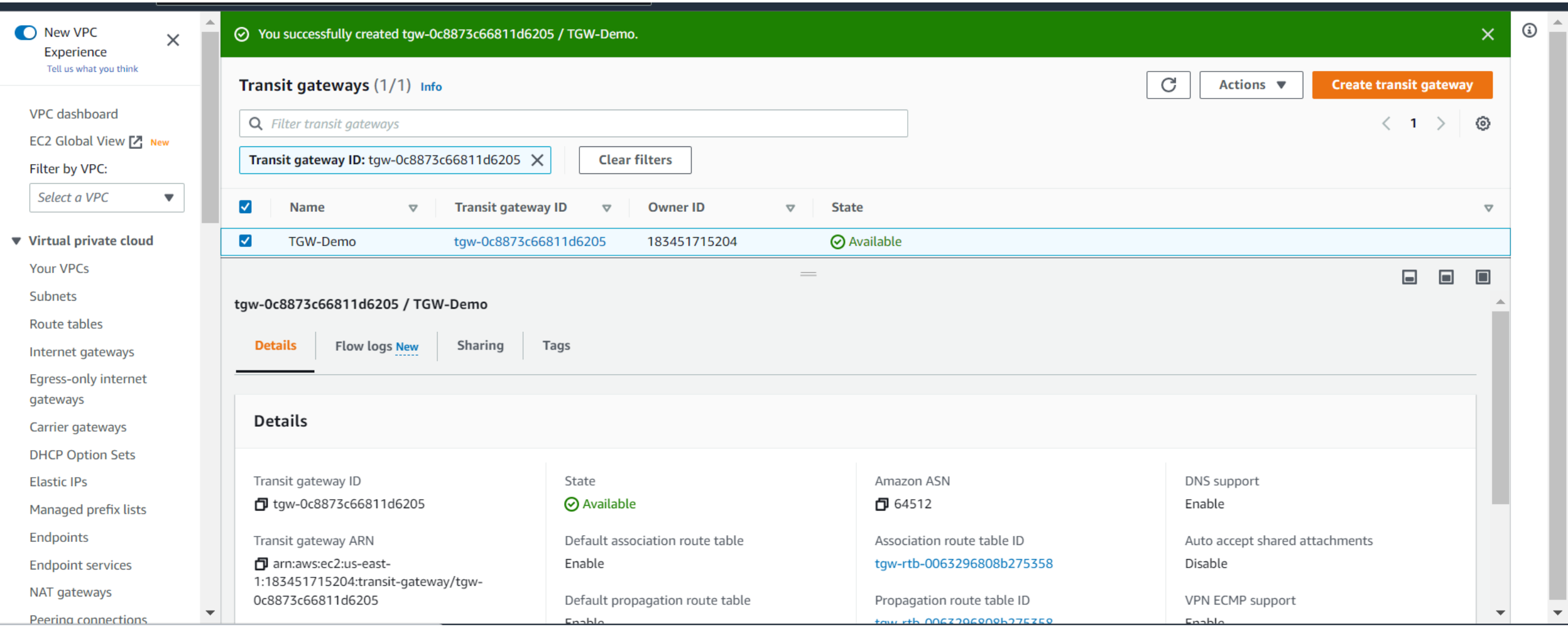
Destination	Target	Route type
VPC A CIDR	*Attachment for VPC A*	propagated
VPC B CIDR	*Attachment for VPC B*	propagated
VPC C CIDR	*Attachment for VPC C*	propagated

https://docs.aws.amazon.com/vpc/latest/tgw/how-transit-gateways-work.html#architecture-diagram


Site-to-Site VPN Connections
Client VPN Endpoints
AWS Cloud WAN
Network Manager
Transit gateways
Transit gateways Info
Actions
Create transit gateway
Filter transit gateways
1
Name
Transit gateway ID
Owner ID
State
No transit gateways found



- An Autonomous System (AS) is a collection of connected Internet Protocol (IP) routing prefixes run by one or more network operators that maintain a single, clearly-defined routing policy. Typically to identify a large network.
- Each AS is assigned an autonomous system number (ASN).
  - Private or Public (globally unique assigned by IANA)
- When you create a virtual private gateway, you can specify the **private Autonomous System Number** (ASN) for the Amazon side of the gateway. The default ASN is 64512.


New VPC Experience
Tell us what you think
VPC dashboard
EC2 Global View New
Filter by VPC:
Select a VPC
Virtual private cloud
Your VPCs
Subnets
Route tables
Internet gateways
Egress-only internet gateways
Carrier gateways
DHCP Option Sets
Elastic IPs
Managed prefix lists
Endpoints
Endpoint services
NAT gateways
Peering connections
You successfully created tgw-0c8873c66811d6205 / TGW-Demo.
Transit gateways (1/1) Info
Actions
Create transit gateway
Filter transit gateways
1
Transit gateway ID: tgw-0c8873c66811d6205
Clear filters
Name
Transit gateway ID
Owner ID
State
TGW-Demo
tgw-0c8873c66811d6205
183451715204
Available
tgw-0c8873c66811d6205 / TGW-Demo
Details
Flow logs New
Sharing
Tags
Details
Transit gateway ID
tgw-0c8873c66811d6205
Transit gateway ARN
arn:aws:ec2:us-east-1:183451715204:transit-gateway/tgw-0c8873c66811d6205
State
Available
Default association route table
Enable
Default propagation route table
Amazon ASN
64512
Association route table ID
tgw-rtb-0063296808b275358
Propagation route table ID
DNS support
Enable
Auto accept shared attachments
Disable
VPN ECMP support

Attachment-Marketing

VPC > Transit gateway attachments > Create transit gateway attachment

Create transit gateway attachment Info

A transit gateway (TGW) is a network transit hub that interconnects attachments (VPCs and VPNs) within the same AWS account or across AWS accounts.

Details

Name tag - *optional*
Creates a tag with the key set to Name and the value set to the specified string.

TGW-Marketing-Att

Transit gateway ID Info

tgw-0c8873c66811d6205 (TGW-Demo)

Attachment type Info

VPC

VPC attachment

Select and configure your VPC attachment.

DNS support Info

IPv6 support Info

VPC ID
Select the VPC to attach to the transit gateway.

vpc-05ffea99c97eb73c6 (VPCMarketing)

Subnet IDs Info
Select the subnets in which to create the transit gateway VPC attachment.

us-east-1a	subnet-04b9dd861ae07e112 (VPCMarketing)
us-east-1b	No subnet available
us-east-1c	No subnet available
us-east-1d	No subnet available
us-east-1e	No subnet available
us-east-1f	No subnet available

subnet-04b9dd861ae07e112

Tags

A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

Key	Value - *optional*	
Name	TGW-Marketing-Att	Remove

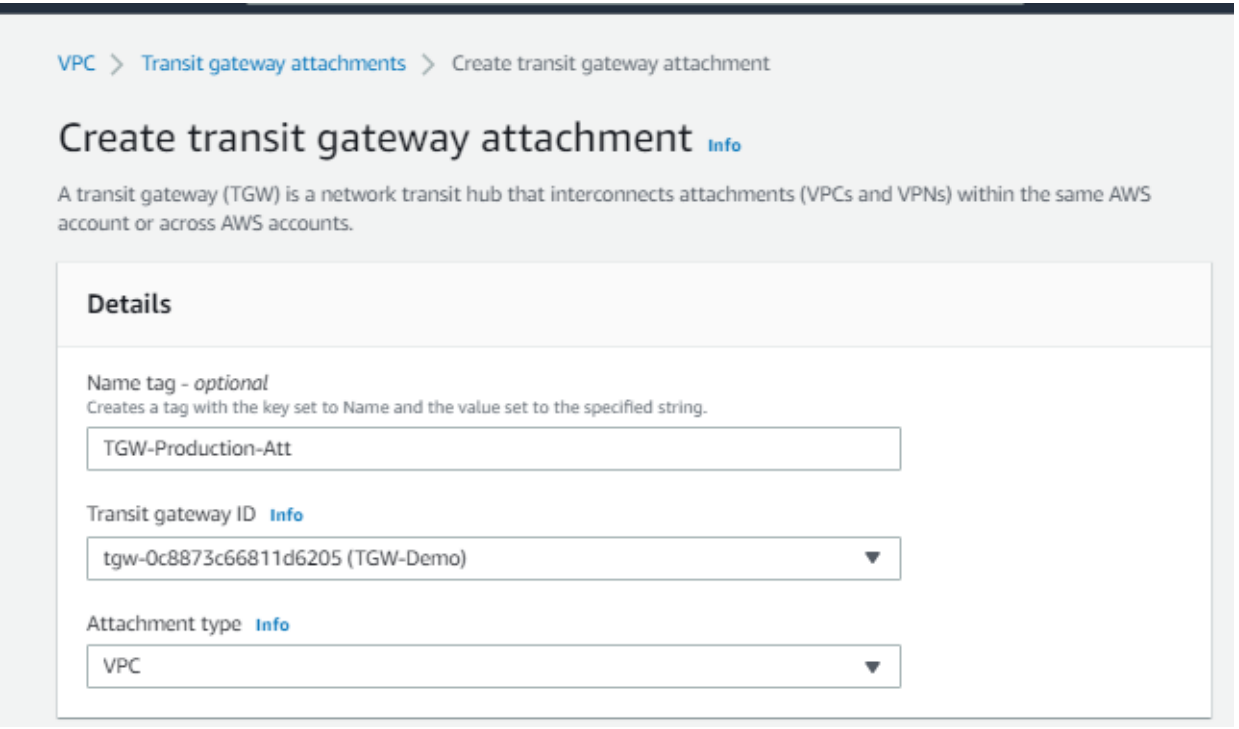
Add new tag

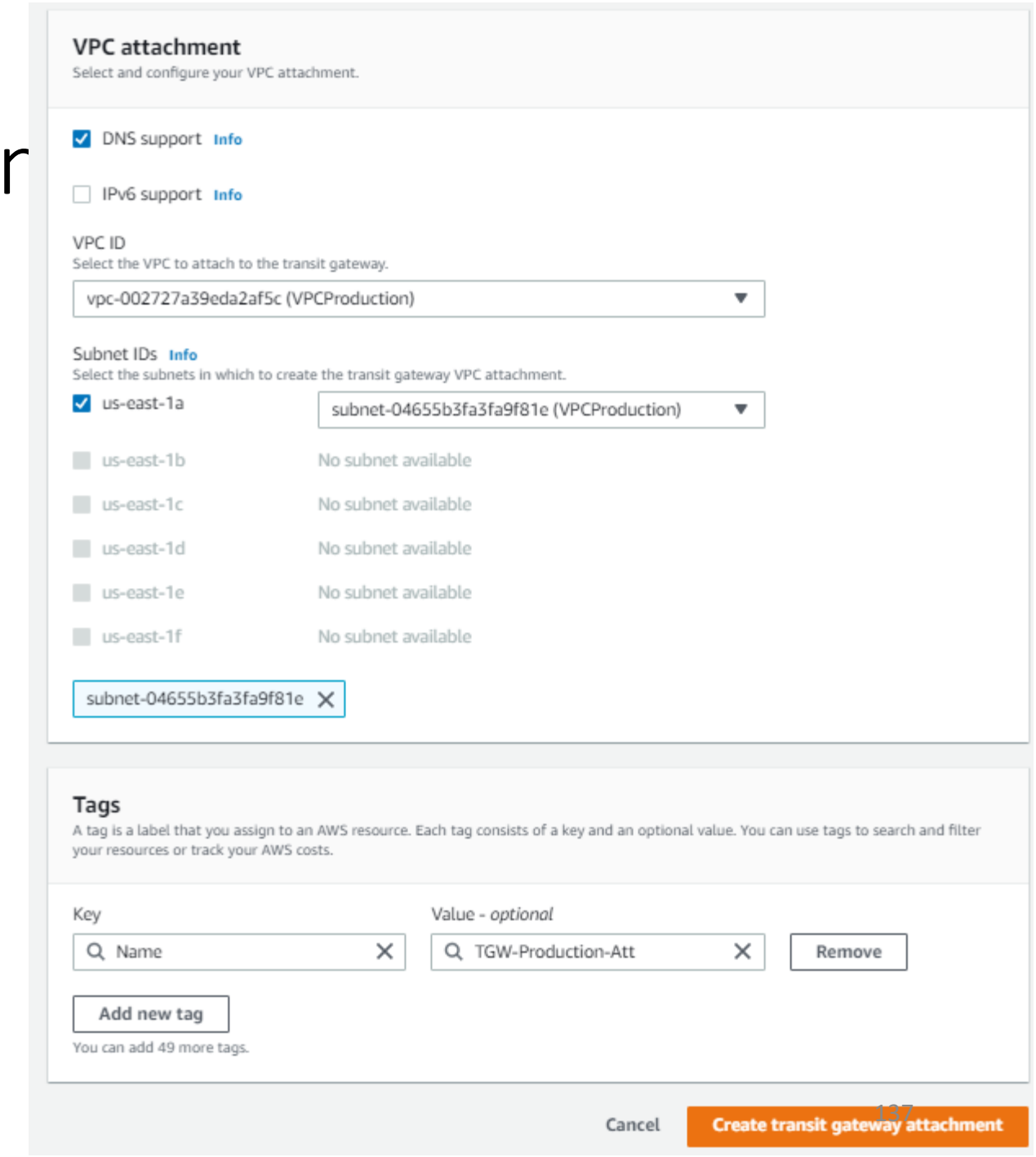
You can add 49 more tags.

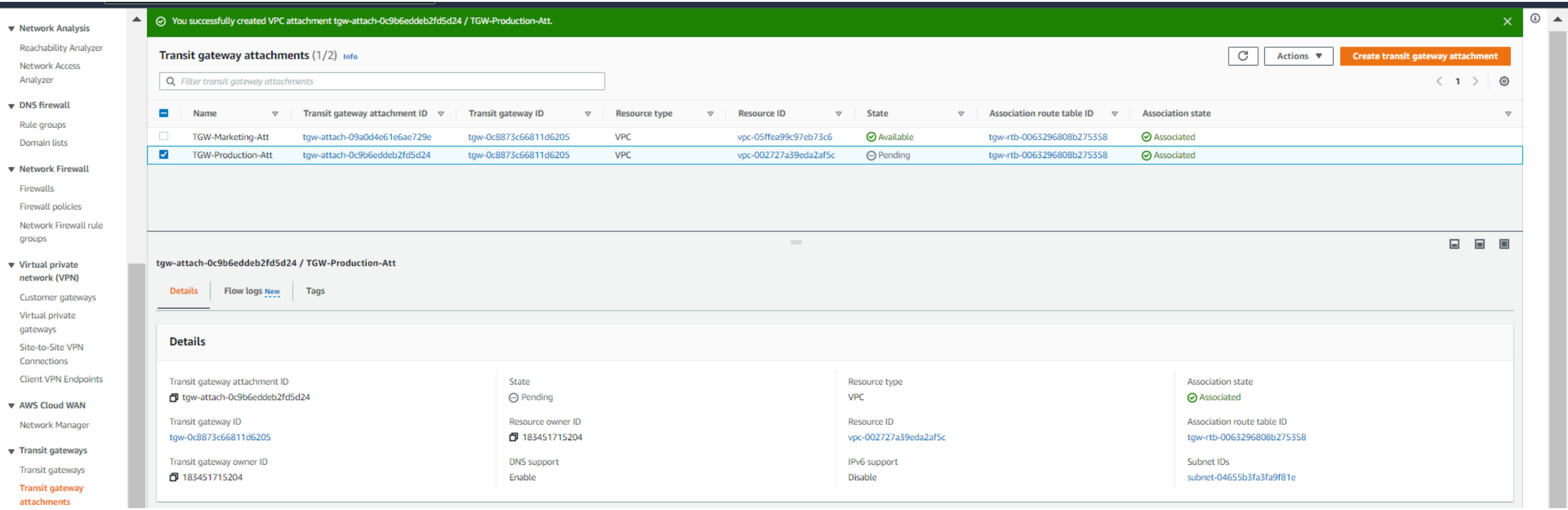
Cancel | Create transit gateway attachment

You must specify one subnet from each Availability Zone to be used by the transit gateway to route traffic. Specifying one subnet from an AZ enables traffic to reach resources in every subnet in that AZ.

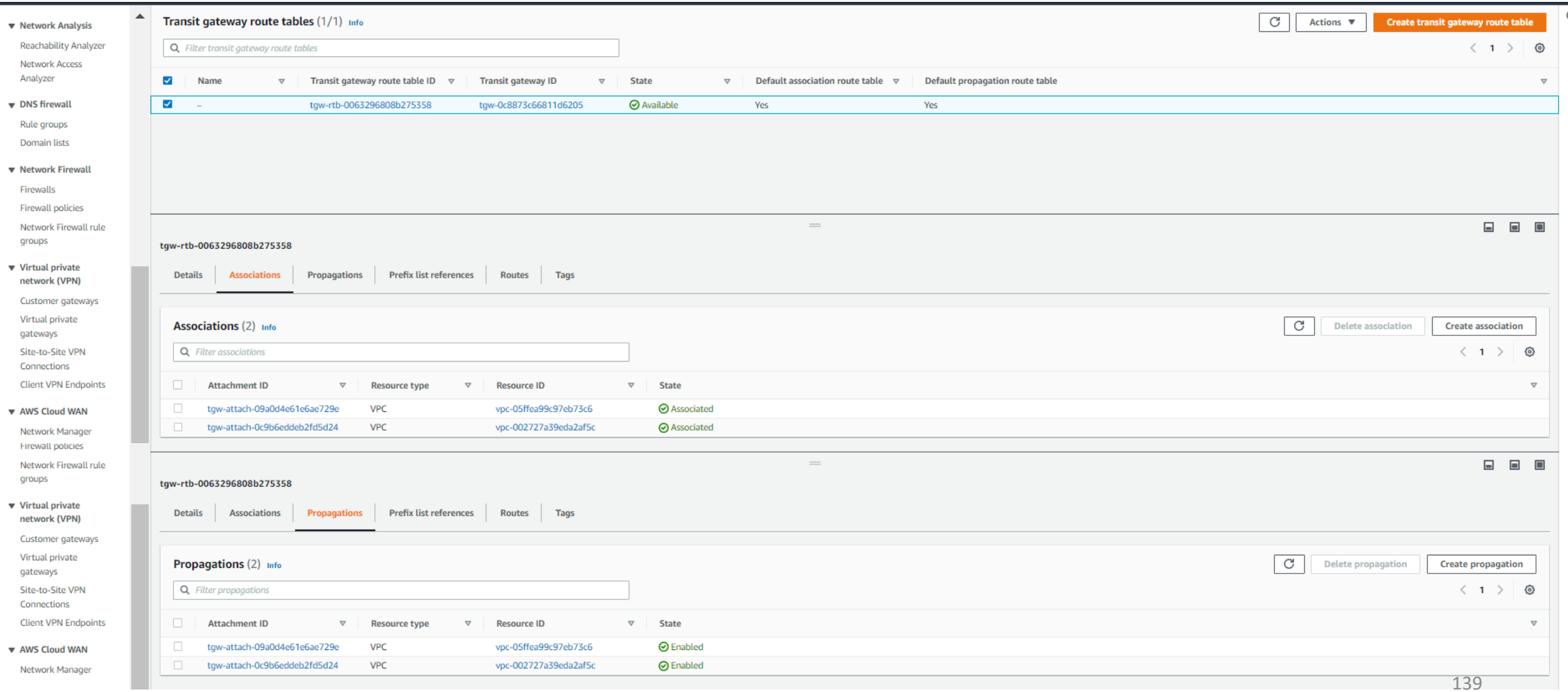
Attachment-Productior



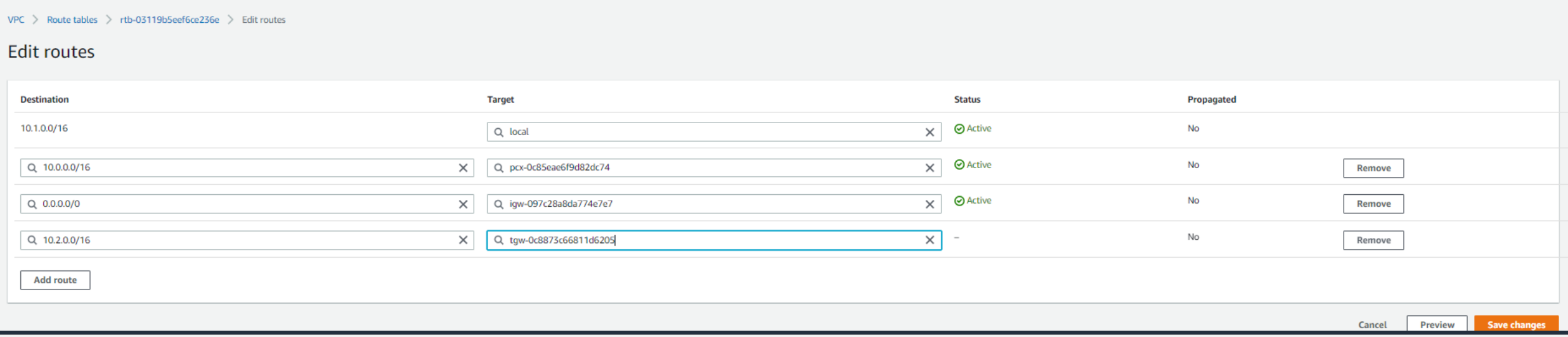


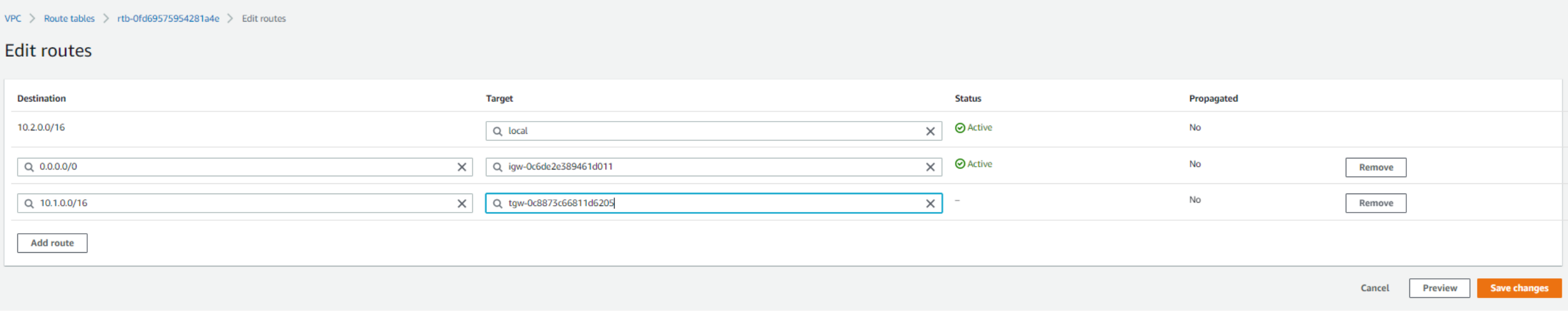

Network Analysis
Reachability Analyzer
Network Access Analyzer
DNS firewall
Rule groups
Domain lists
Network Firewall
Firewalls
Firewall policies
Network Firewall rule groups
Virtual private network (VPN)
Customer gateways
Virtual private gateways
Site-to-Site VPN Connections
Client VPN Endpoints
AWS Cloud WAN
Network Manager
Transit gateways
Transit gateways
Transit gateway attachments
You successfully created VPC attachment tgw-attach-0c9b6eddeb2fd5d24 / TGW-Production-Att.
Transit gateway attachments (1/2) Info
Actions
Create transit gateway attachment
Filter transit gateway attachments
Name
Transit gateway attachment ID
Transit gateway ID
Resource type
Resource ID
State
Association route table ID
Association state
TGW-Marketing-Att
tgw-attach-09a0d4e61e6ae729e
tgw-0c8873c66811d6205
VPC
vpc-05ffea99c97eb73c6
Available
tgw-rtb-0063296808b275358
Associated
TGW-Production-Att
tgw-attach-0c9b6eddeb2fd5d24
tgw-0c8873c66811d6205
VPC
vpc-002727a39eda2af5c
Pending
tgw-rtb-0063296808b275358
Associated
tgw-attach-0c9b6eddeb2fd5d24 / TGW-Production-Att
Details
Flow logs New
Tags
Details
Transit gateway attachment ID
tgw-attach-0c9b6eddeb2fd5d24
Transit gateway ID
tgw-0c8873c66811d6205
Transit gateway owner ID
183451715204
State
Pending
Resource owner ID
183451715204
DNS support
Enable
Resource type
VPC
Resource ID
vpc-002727a39eda2af5c
IPv6 support
Disable
Association state
Associated
Association route table ID
tgw-rtb-0063296808b275358
Subnet IDs
subnet-04655b3fa3fa9f81e

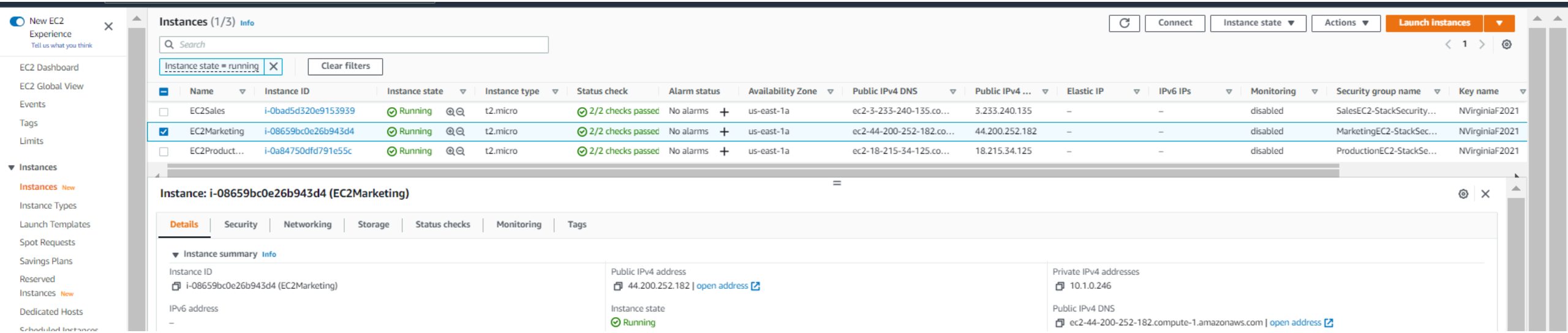
TGW Routing Table

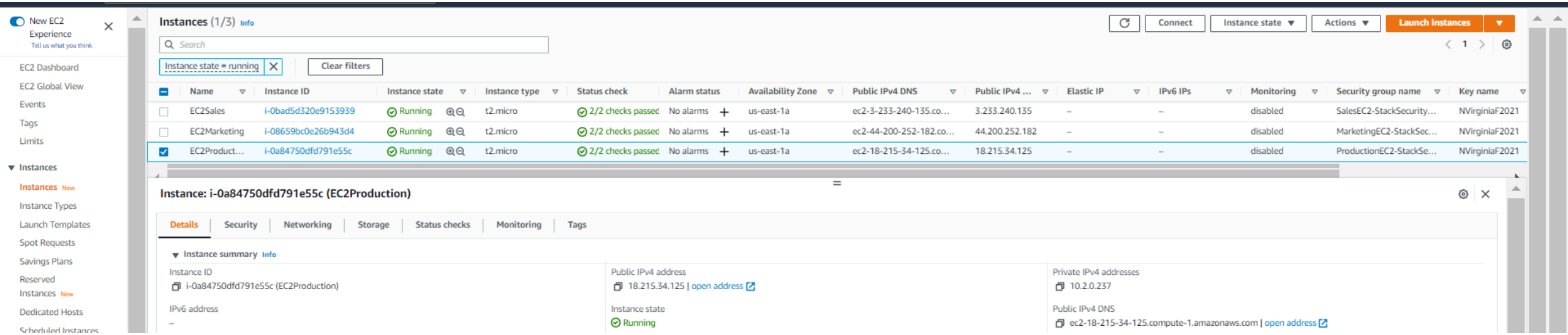


Update VPC Route Table






New EC2 Experience
Tell us what you think
EC2 Dashboard
EC2 Global View
Events
Tags
Limits
Instances
Instances New
Instance Types
Launch Templates
Spot Requests
Savings Plans
Reserved Instances New
Dedicated Hosts
Scheduled Instances
Instances (1/3) Info
Search
Instance state = running
Clear filters
Connect
Instance state
Actions
Launch instances
Name
Instance ID
Instance state
Instance type
Status check
Alarm status
Availability Zone
Public IPv4 DNS
Public IPv4 ...
Elastic IP
IPv6 IPs
Monitoring
Security group name
Key name
EC2Sales i-0bad5d320e9153939 Running t2.micro 2/2 checks passed No alarms us-east-1a ec2-3-233-240-135.co... 3.233.240.135 – – disabled SalesEC2-StackSecurity... NVirginiaF2021
EC2Marketing i-08659bc0e26b943d4 Running t2.micro 2/2 checks passed No alarms us-east-1a ec2-44-200-252-182.co... 44.200.252.182 – – disabled MarketingEC2-StackSec... NVirginiaF2021
EC2Product... i-0a84750dfd791e55c Running t2.micro 2/2 checks passed No alarms us-east-1a ec2-18-215-34-125.co... 18.215.34.125 – – disabled ProductionEC2-StackSe... NVirginiaF2021
Instance: i-08659bc0e26b943d4 (EC2Marketing)
Details
Security
Networking
Storage
Status checks
Monitoring
Tags
Instance summary Info
Instance ID
i-08659bc0e26b943d4 (EC2Marketing)
IPv6 address
–
Public IPv4 address
44.200.252.182 | open address
Instance state
Running
Private IPv4 addresses
10.1.0.246
Public IPv4 DNS
ec2-44-200-252-182.compute-1.amazonaws.com | open address


New EC2 Experience
Tell us what you think
EC2 Dashboard
EC2 Global View
Events
Tags
Limits
Instances
Instances New
Instance Types
Launch Templates
Spot Requests
Savings Plans
Reserved Instances New
Dedicated Hosts
Scheduled Instances
Instances (1/3) Info
Search
Instance state = running
Clear filters
Connect
Instance state
Actions
Launch instances
Name
Instance ID
Instance state
Instance type
Status check
Alarm status
Availability Zone
Public IPv4 DNS
Public IPv4 ...
Elastic IP
IPv6 IPs
Monitoring
Security group name
Key name
EC2Sales i-0bad5d320e9153939 Running t2.micro 2/2 checks passed No alarms us-east-1a ec2-3-233-240-135.co... 3.233.240.135 – – disabled SalesEC2-StackSecurity... NVirginiaF2021
EC2Marketing i-08659bc0e26b943d4 Running t2.micro 2/2 checks passed No alarms us-east-1a ec2-44-200-252-182.co... 44.200.252.182 – – disabled MarketingEC2-StackSec... NVirginiaF2021
EC2Product... i-0a84750dfd791e55c Running t2.micro 2/2 checks passed No alarms us-east-1a ec2-18-215-34-125.co... 18.215.34.125 – – disabled ProductionEC2-StackSe... NVirginiaF2021
Instance: i-0a84750dfd791e55c (EC2Production)
Details
Security
Networking
Storage
Status checks
Monitoring
Tags
Instance summary Info
Instance ID
i-0a84750dfd791e55c (EC2Production)
IPv6 address
–
Public IPv4 address
18.215.34.125 | open address
Instance state
Running
Private IPv4 addresses
10.2.0.237
Public IPv4 DNS
ec2-18-215-34-125.compute-1.amazonaws.com | open address

```
ubuntu@ip-10-1-0-246:~$ ping -c3 10.2.0.237
PING 10.2.0.237 (10.2.0.237) 56(84) bytes of data.
64 bytes from 10.2.0.237: icmp_seq=1 ttl=63 time=1.46 ms
64 bytes from 10.2.0.237: icmp_seq=2 ttl=63 time=0.757 ms
64 bytes from 10.2.0.237: icmp_seq=3 ttl=63 time=0.794 ms

--- 10.2.0.237 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2019ms
rtt min/avg/max/mdev = 0.757/1.002/1.455/0.320 ms
ubuntu@ip-10-1-0-246:~$
```

```
ubuntu@ip-10-2-0-237:~$ ping -c3 10.1.0.246
PING 10.1.0.246 (10.1.0.246) 56(84) bytes of data.
64 bytes from 10.1.0.246: icmp_seq=1 ttl=63 time=1.49 ms
64 bytes from 10.1.0.246: icmp_seq=2 ttl=63 time=0.859 ms
64 bytes from 10.1.0.246: icmp_seq=3 ttl=63 time=0.832 ms

--- 10.1.0.246 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2003ms
rtt min/avg/max/mdev = 0.832/1.060/1.491/0.304 ms
ubuntu@ip-10-2-0-237:~$
```

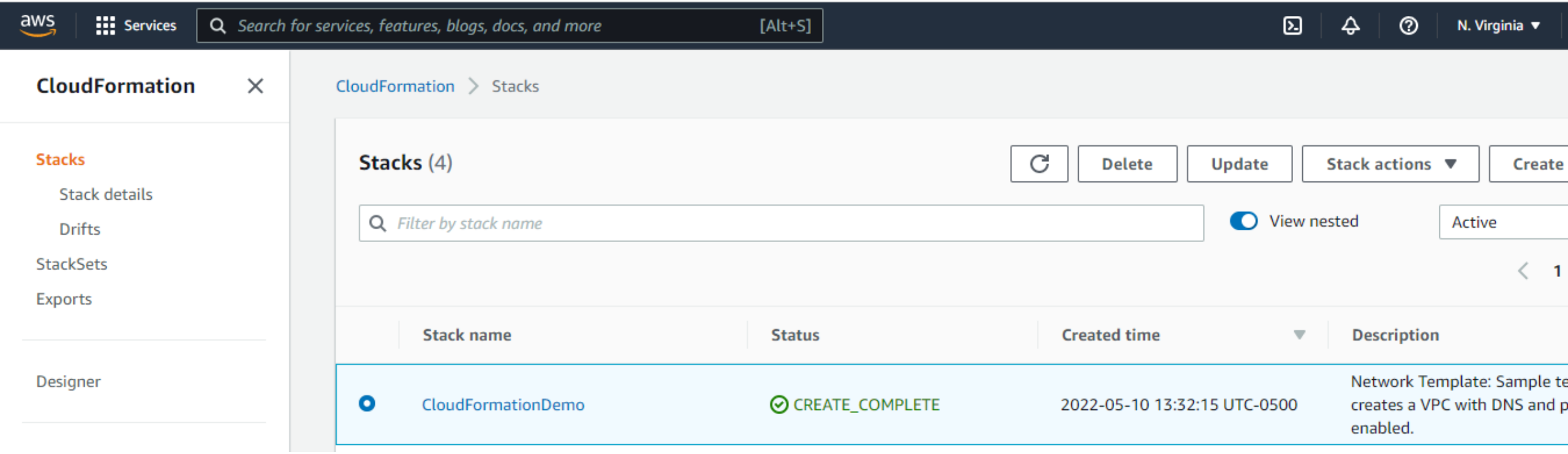
```
ubuntu@ip-10-2-0-237:~$ ping -c3 10.0.0.185
PING 10.0.0.185 (10.0.0.185) 56(84) bytes of data.

--- 10.0.0.185 ping statistics ---
3 packets transmitted, 0 received, 100% packet loss, time 2048ms

ubuntu@ip-10-2-0-237:~$
```

Fail because VPC production is not connected to VPC Sales

Deleting Order

- Delete Attachments
- Delete Transit Gateways
- Delete Peering Connections
- Delete EC2 stacks
- Delete VPC stacks

Delete Stack



End of Lecture