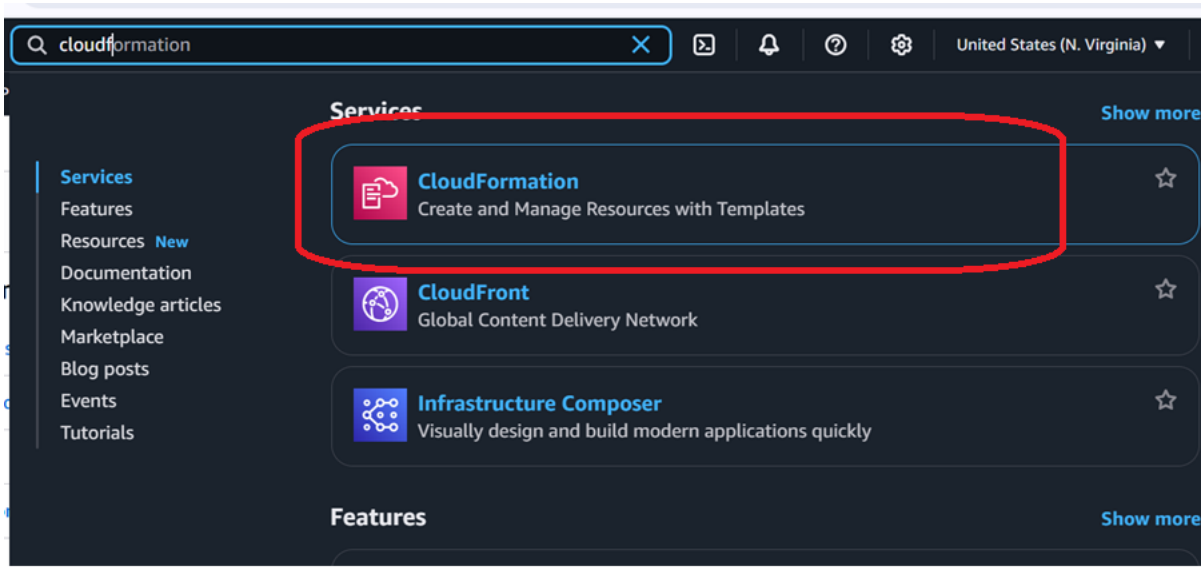
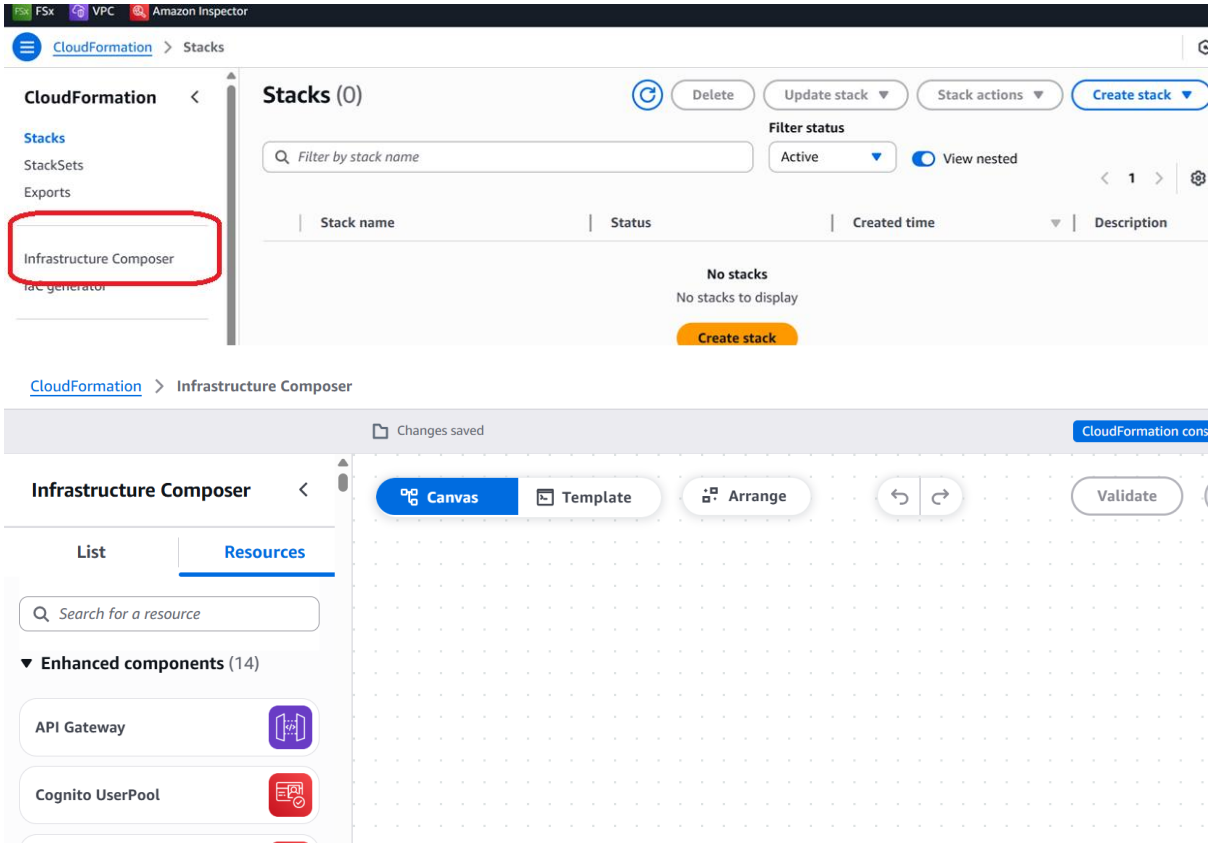
CloudFormation

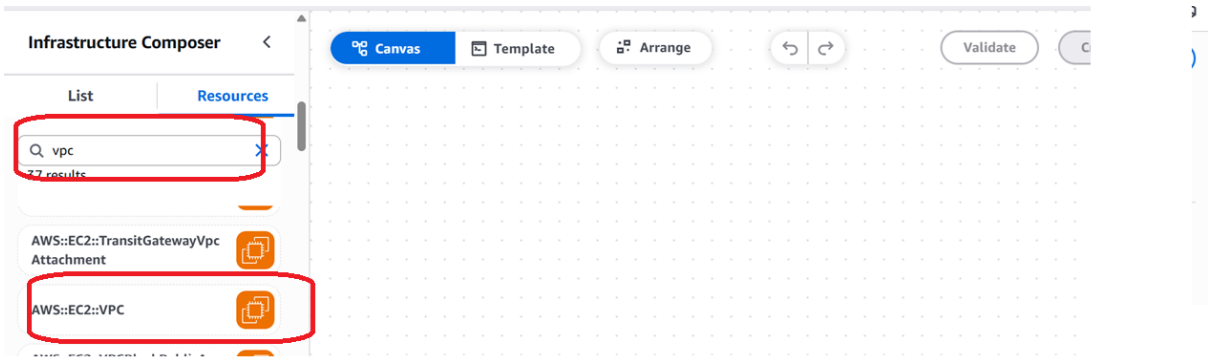
In aws console, select cloudformation:



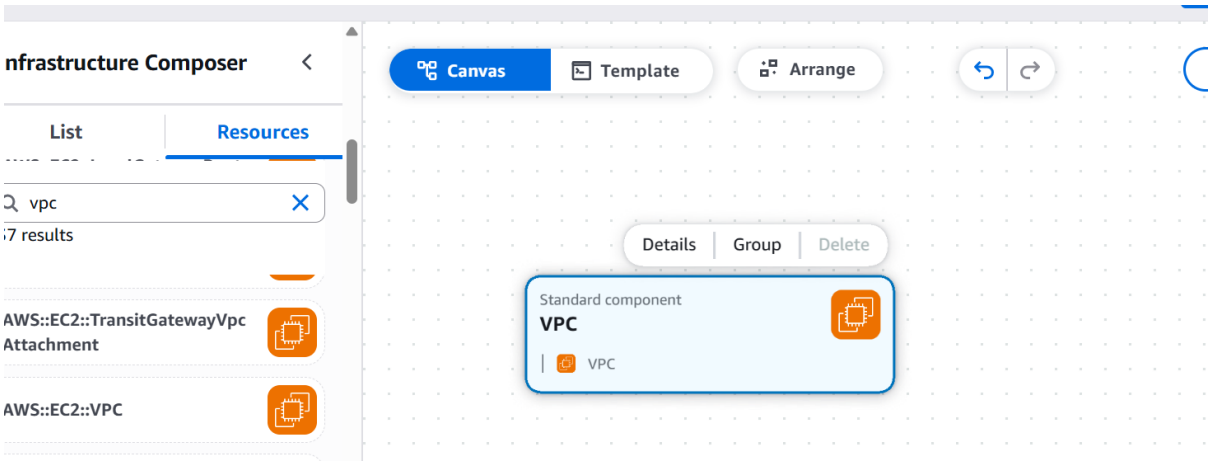
Select infrastructure composer



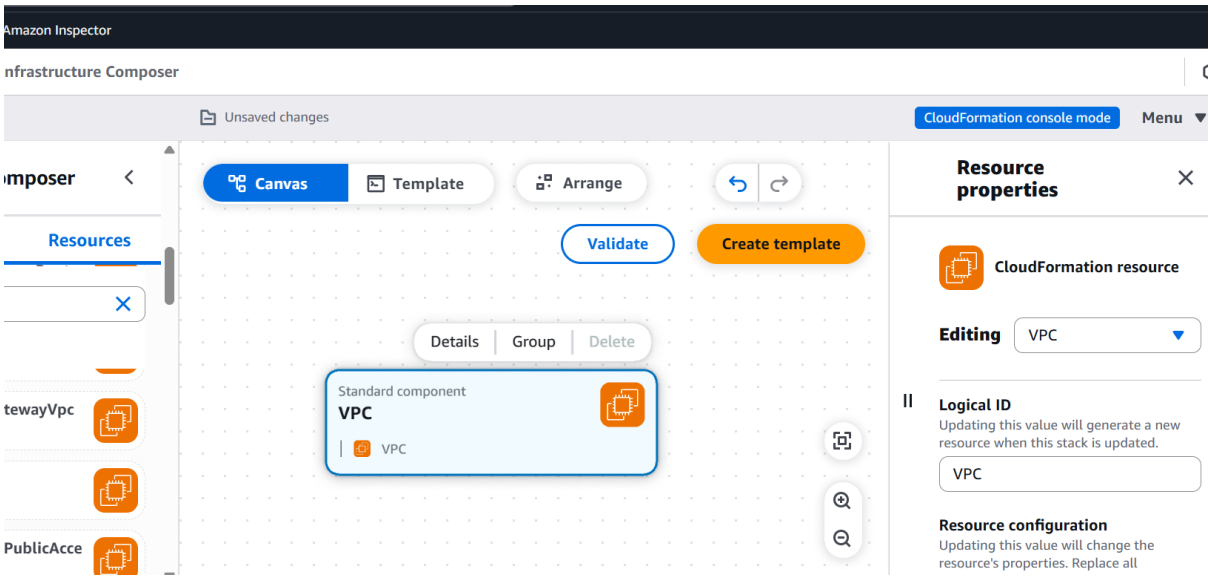
Search for VPC



Drag and drop vpc

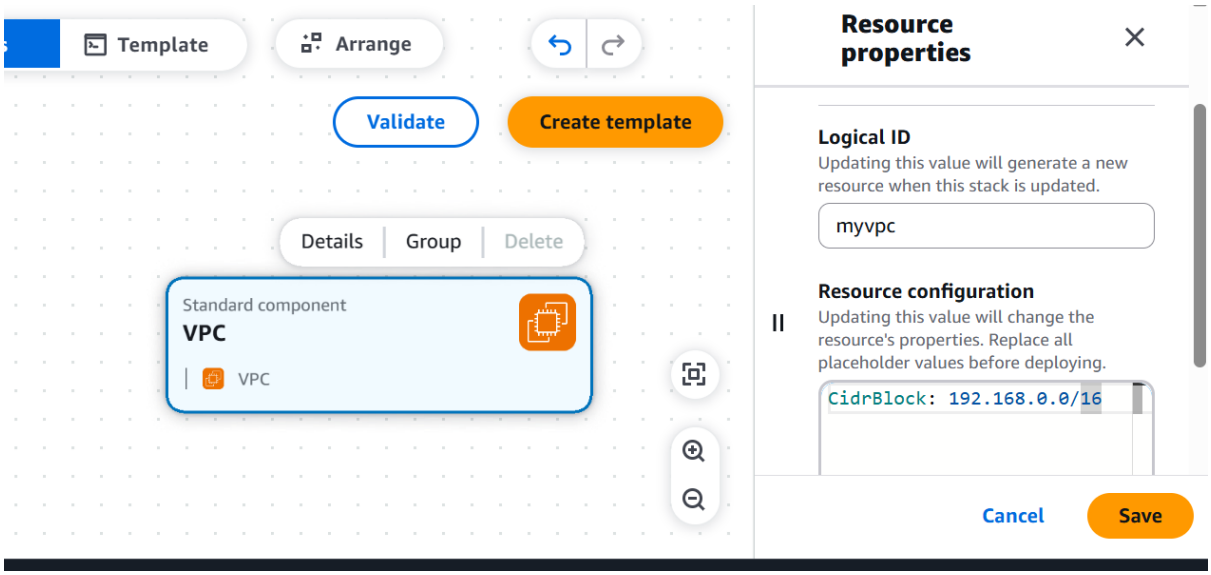


Double click on VPC



Right-hand side you will find Resource configuration

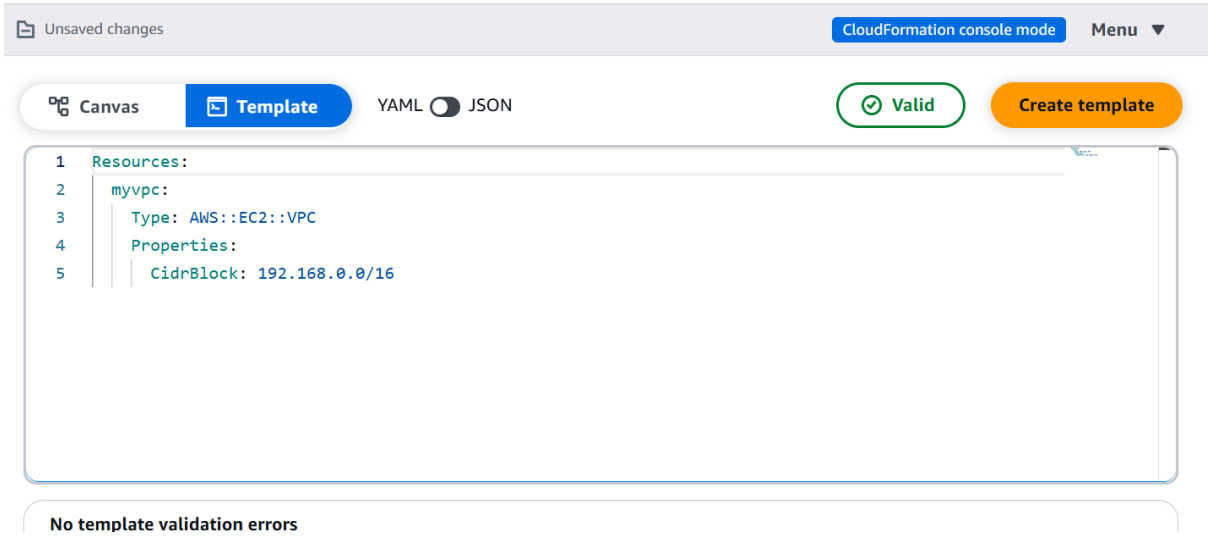
Add code in Resource configuration and save



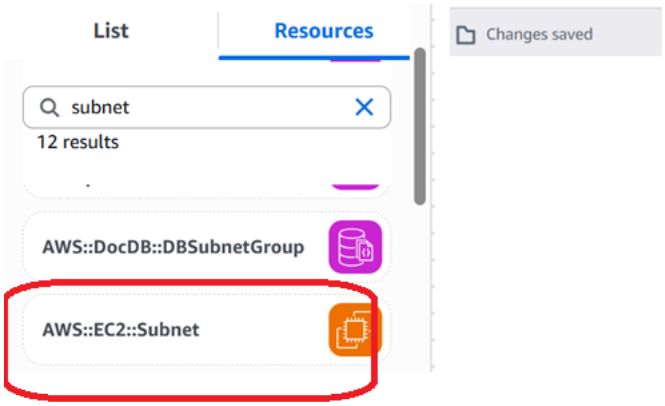
Add following:

CidrBlock: 192.168.0.0/16

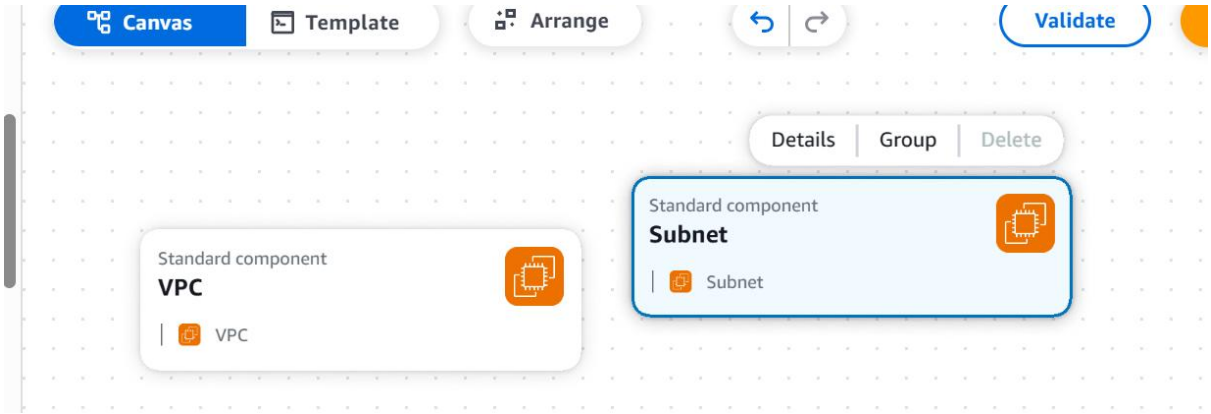
Click on Template you will find the code and click on validate and check



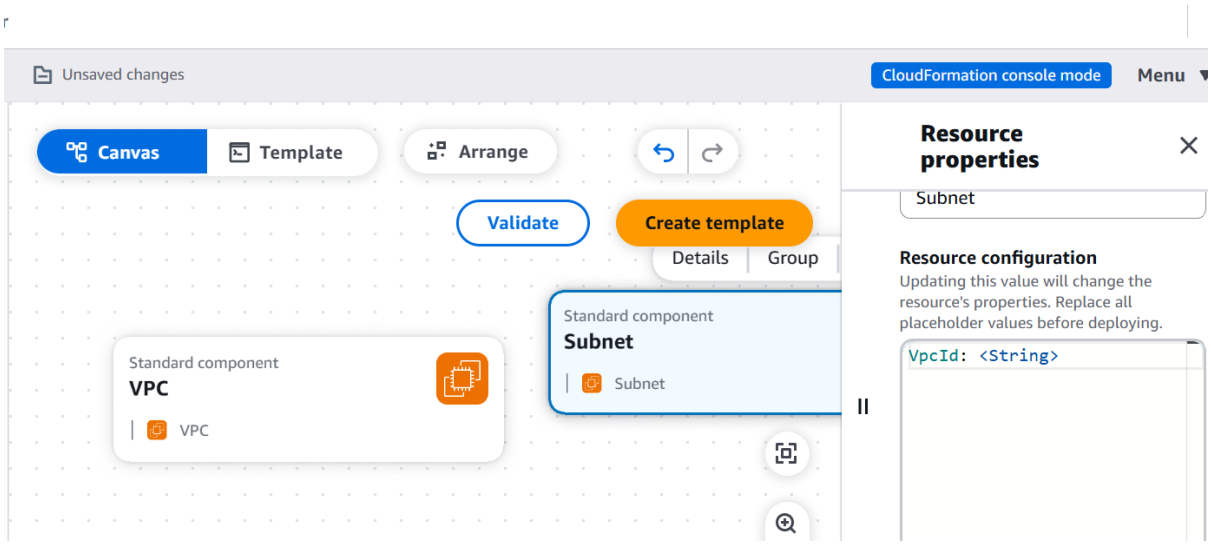
Check for subnet



Drag and drop on AWS::EC2::Subnet



Double click on Subnet and find resource Configration

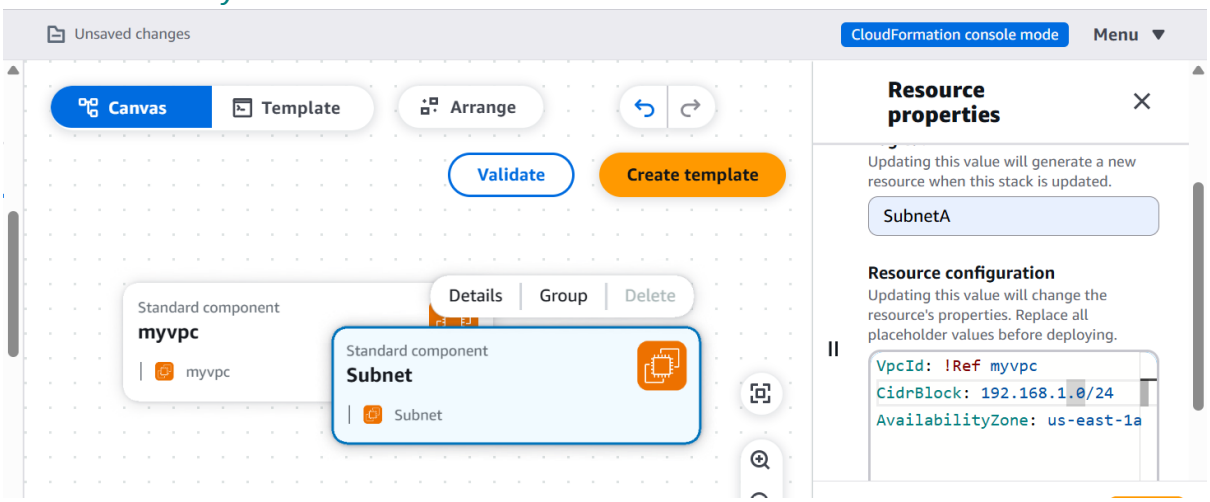


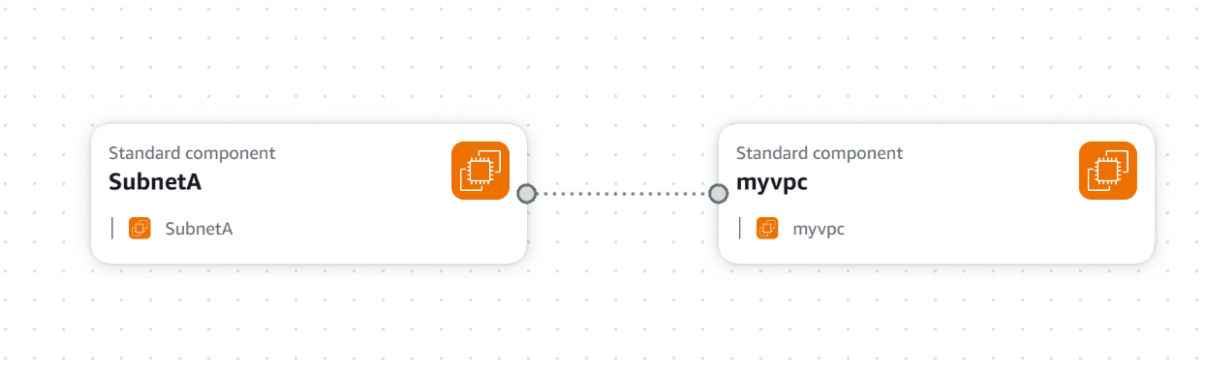
In **AWS CloudFormation**, !Ref (or Ref) is an **intrinsic function** used to **reference a resource, parameter, or pseudo parameter** within a CloudFormation template.

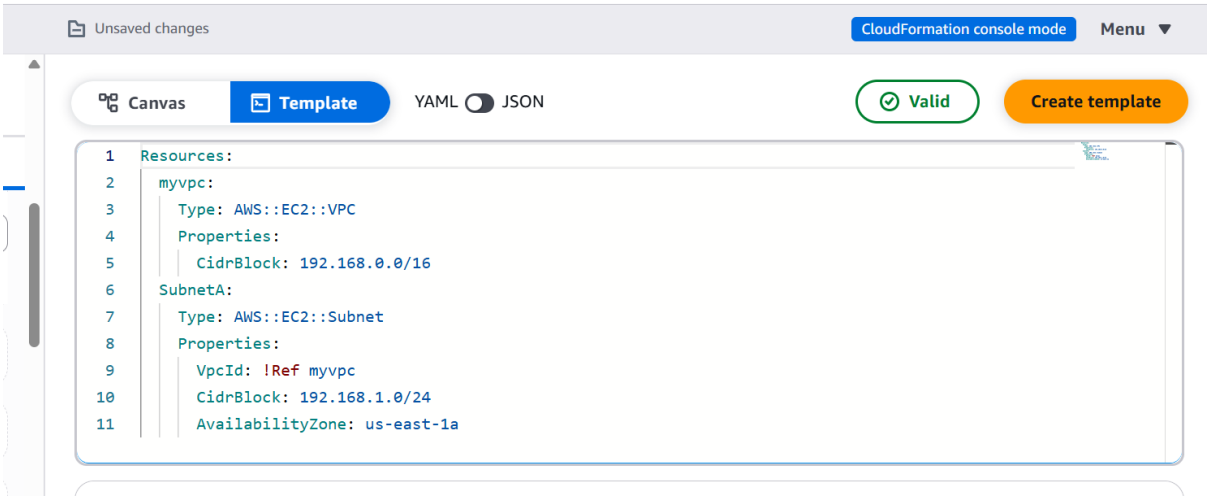
What does !Ref do?

- It **returns a value** that's associated with the specified resource or parameter.
- The returned value depends on what is being referenced.

```
VpcId: !Ref myvpc
CidrBlock: 192.168.1.0/24
AvailabilityZone: us-east-1a
```





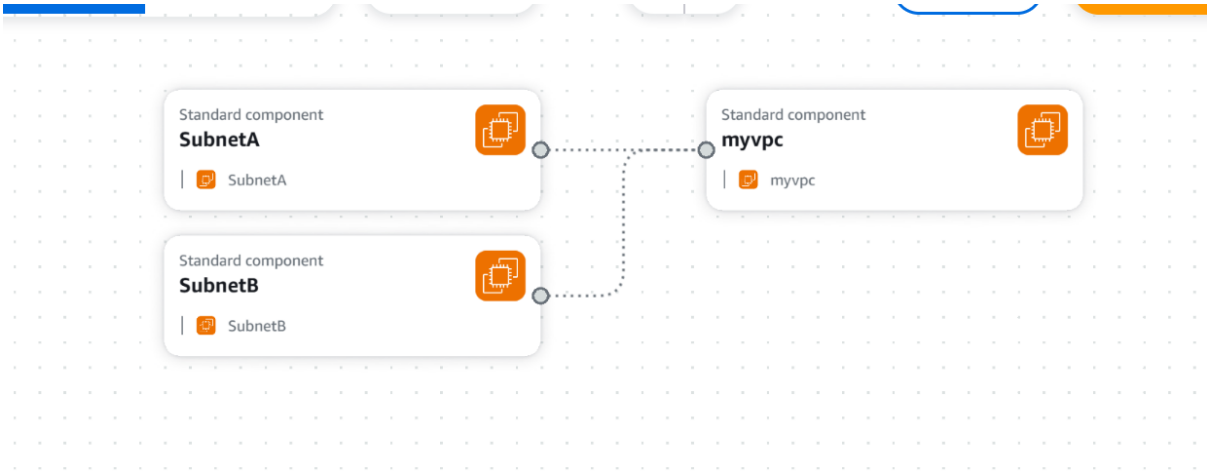


Assignment Create one more subnet

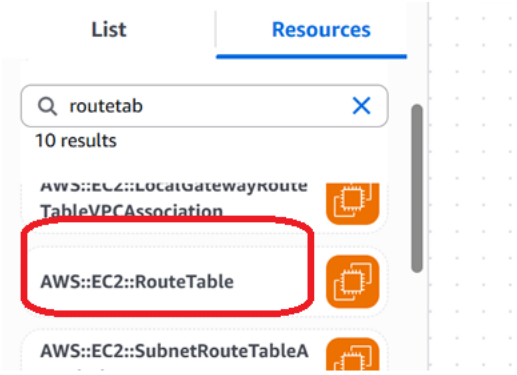
VpcId: !Ref myvpc

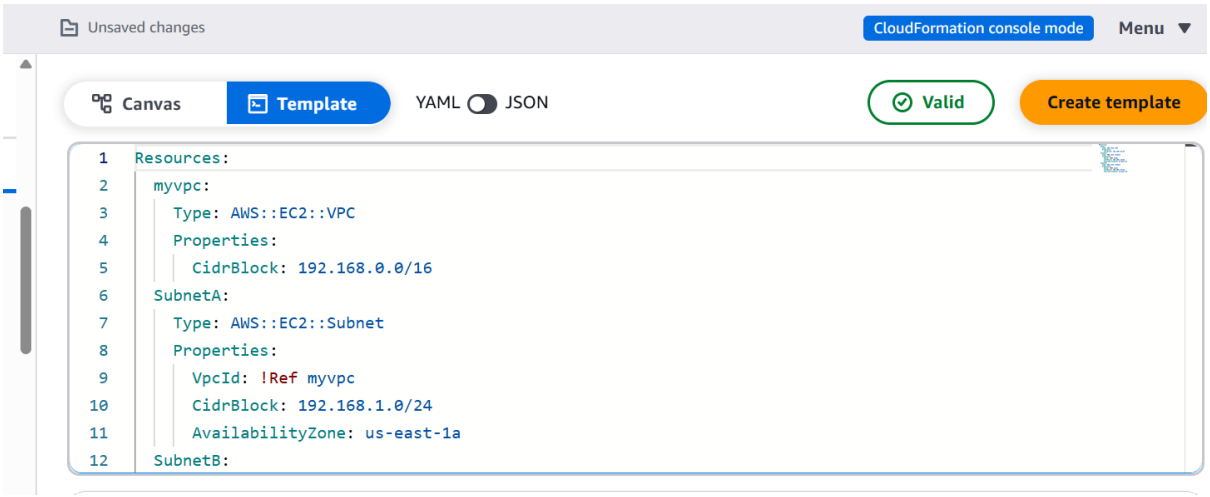
CidrBlock: 192.168.2.0/24

AvailabilityZone: us-east-1b



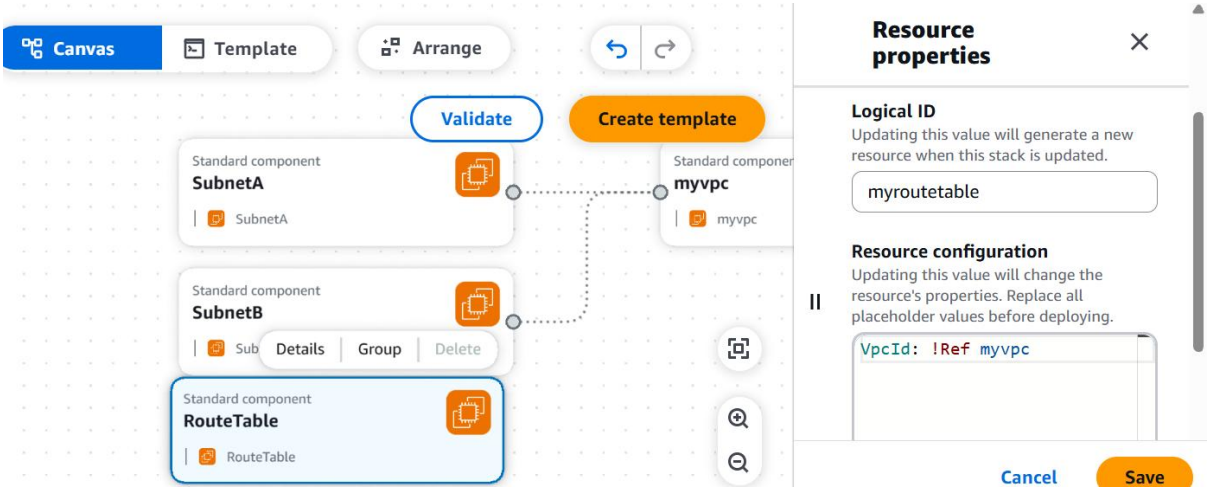
Create RouteTable



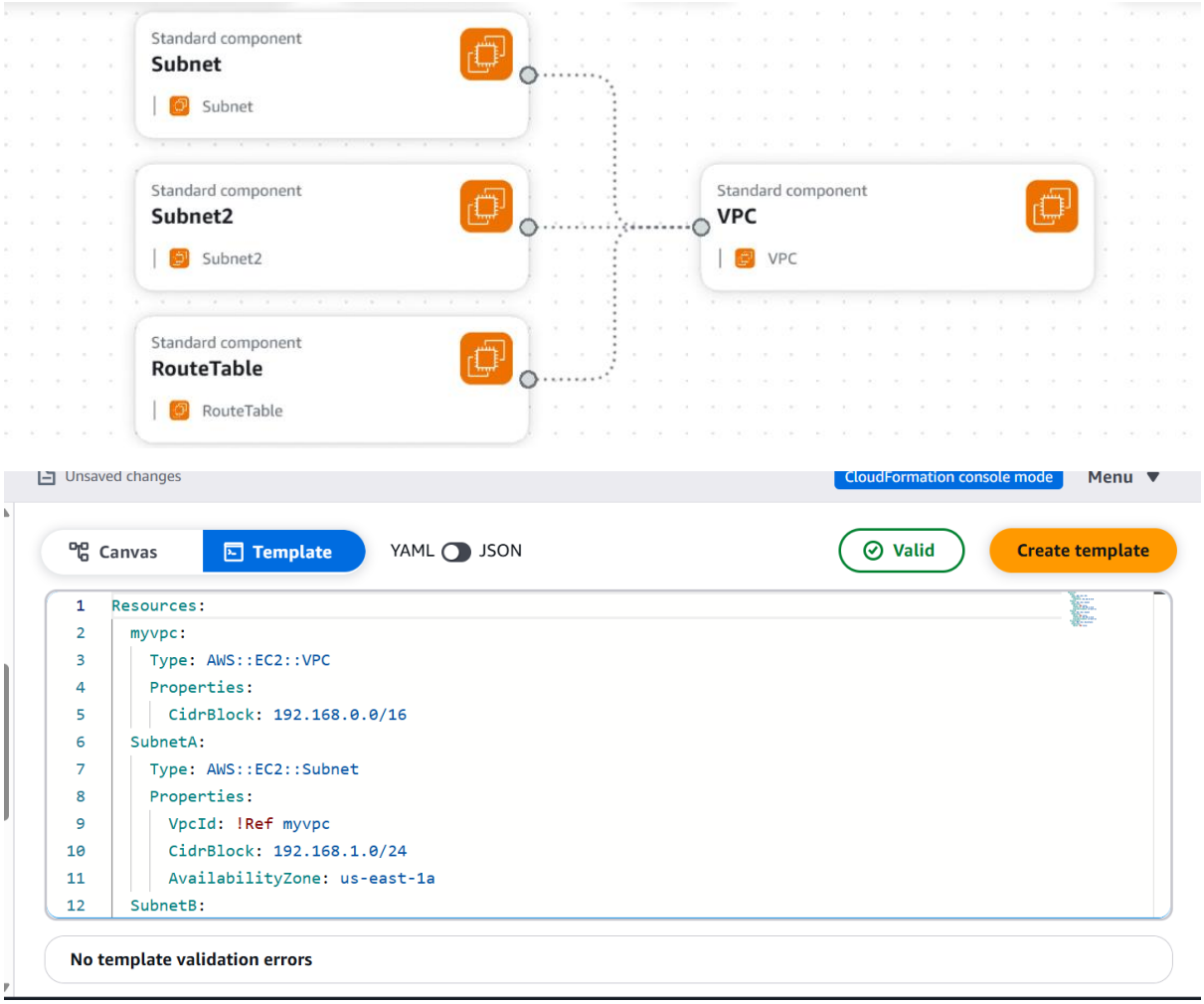


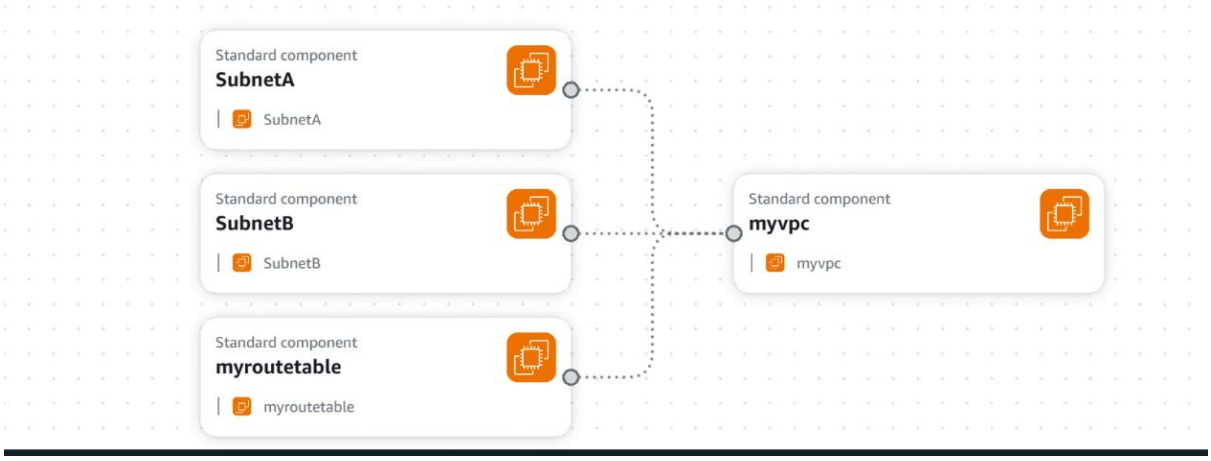
Drag and drop

VpcId: !Ref myvpc

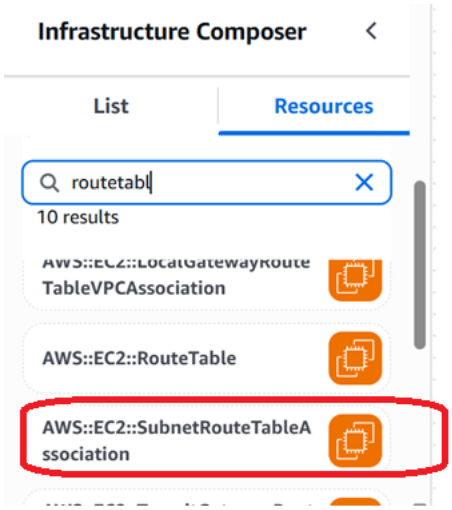


Save it

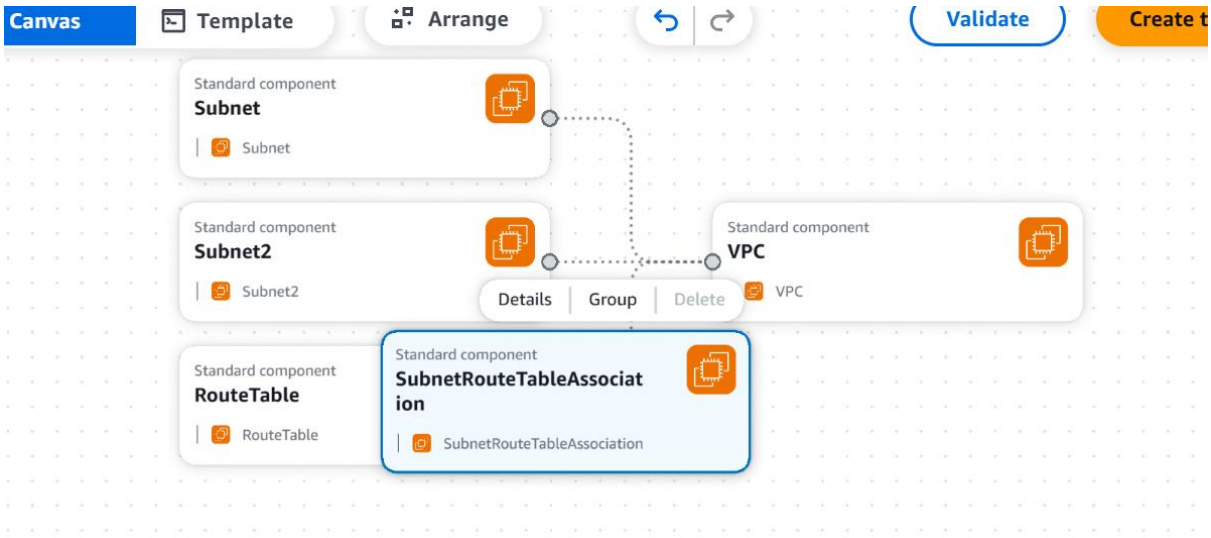


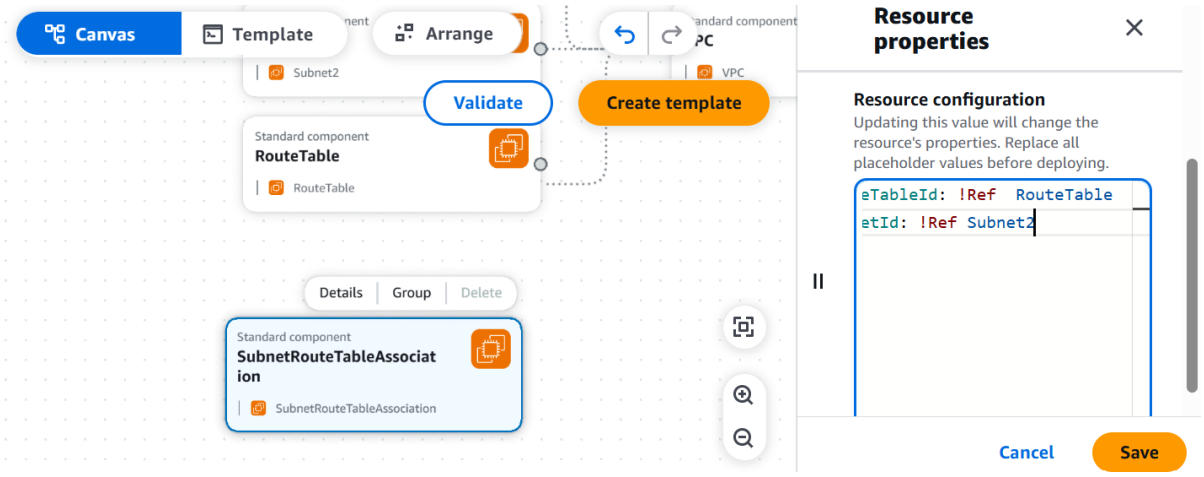


Check for subnetroutetableassociation



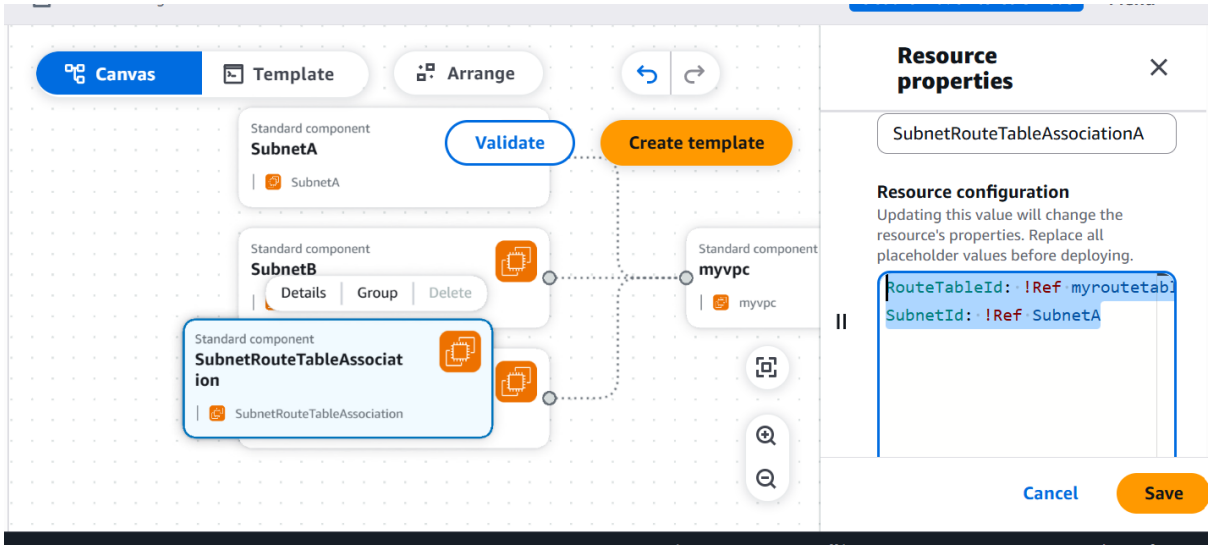
Drag and drop

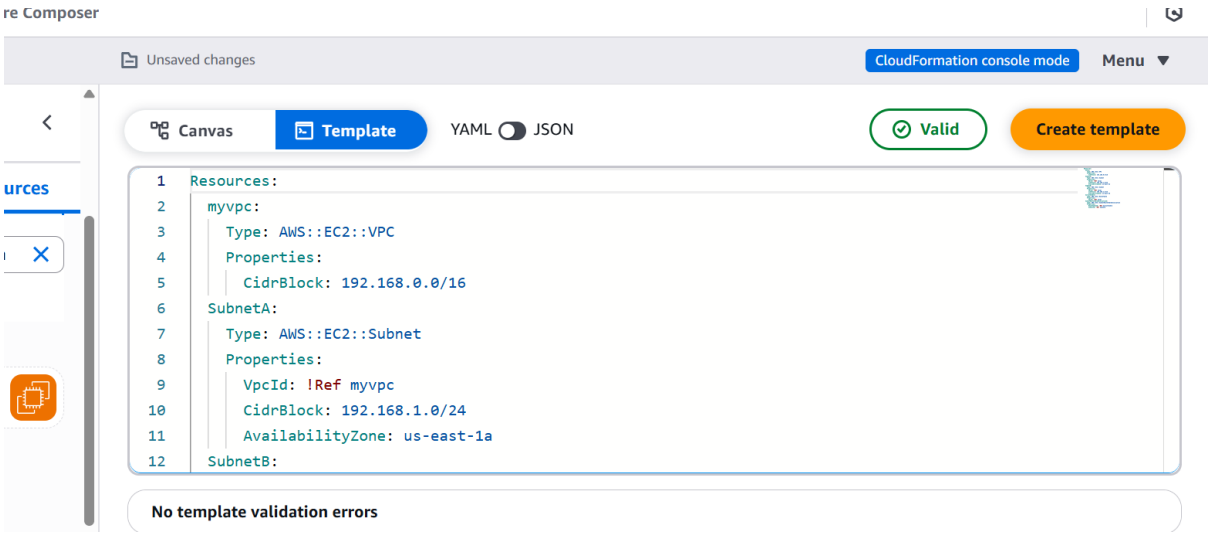


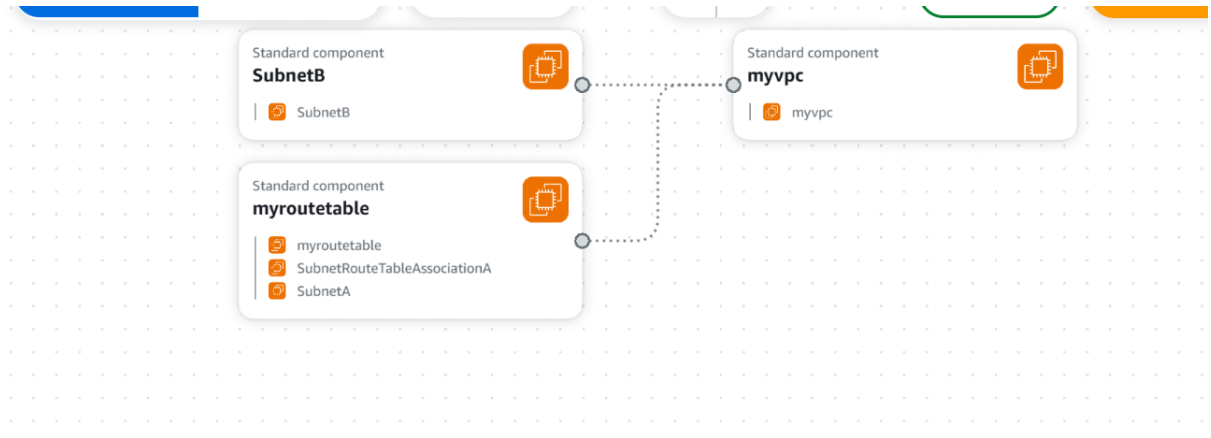


RouteTableId: !Ref myroutetable

SubnetId: !Ref SubnetA



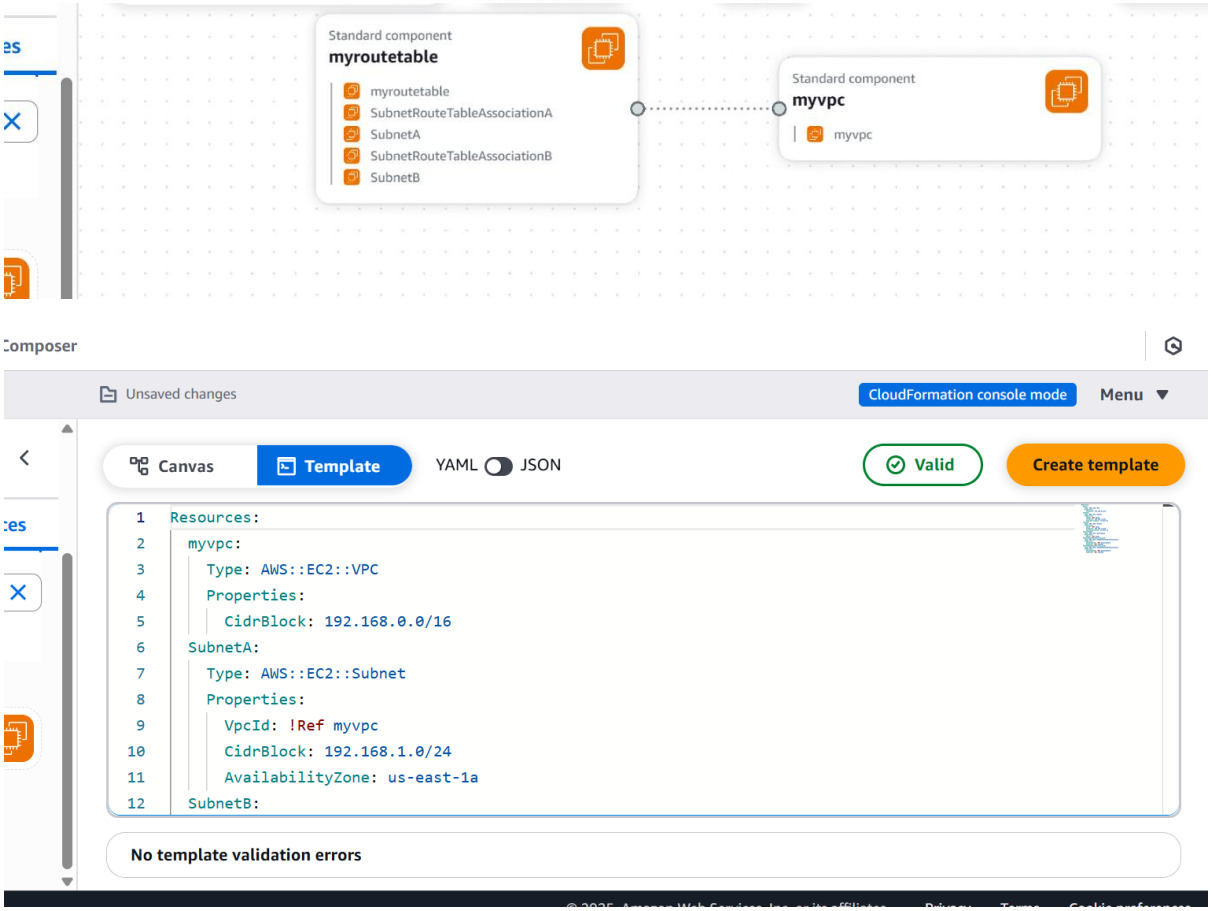




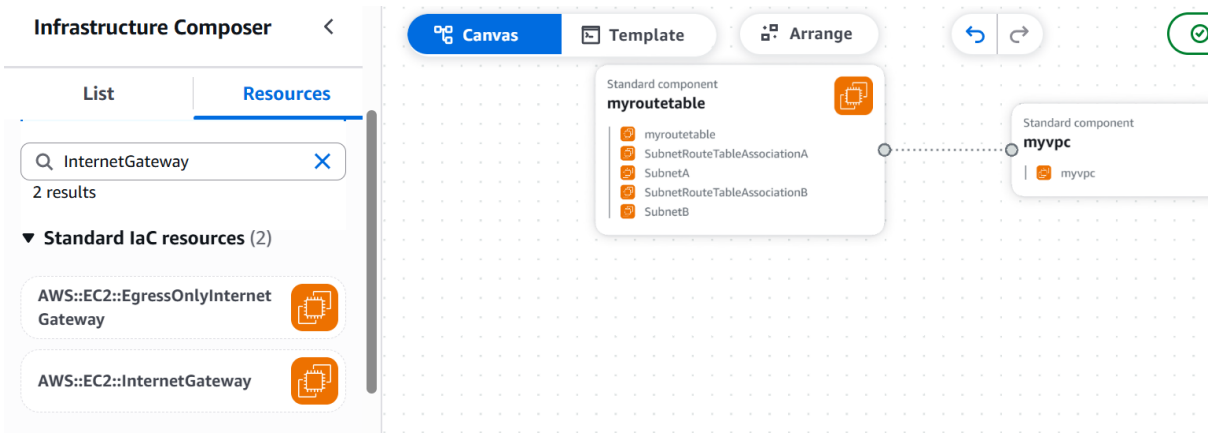
Assignment Add subnetB as well: using SubnetRouteTableAssociation

RouteTableId: !Ref myroutetable

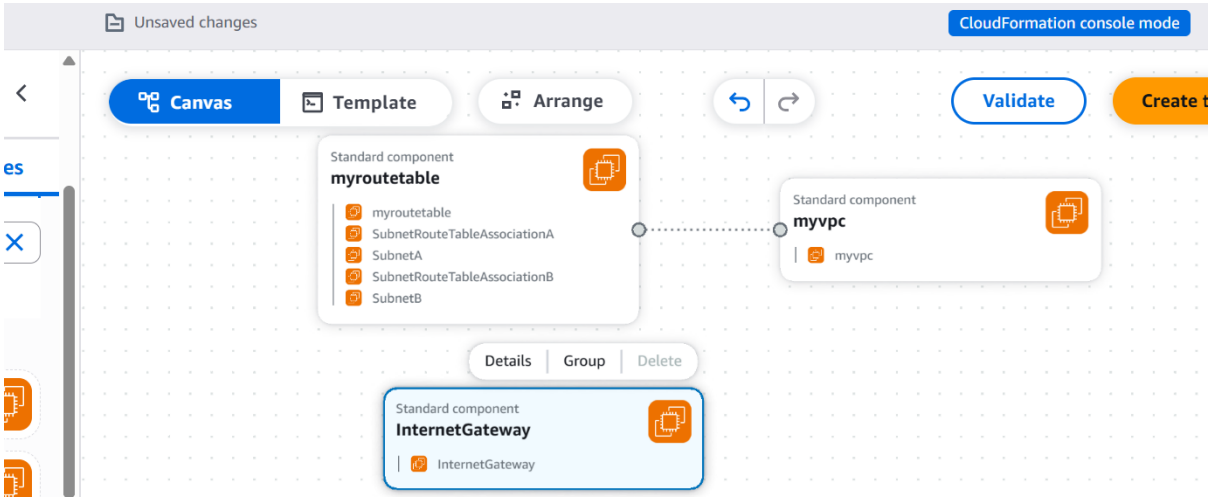
SubnetId: !Ref SubnetB

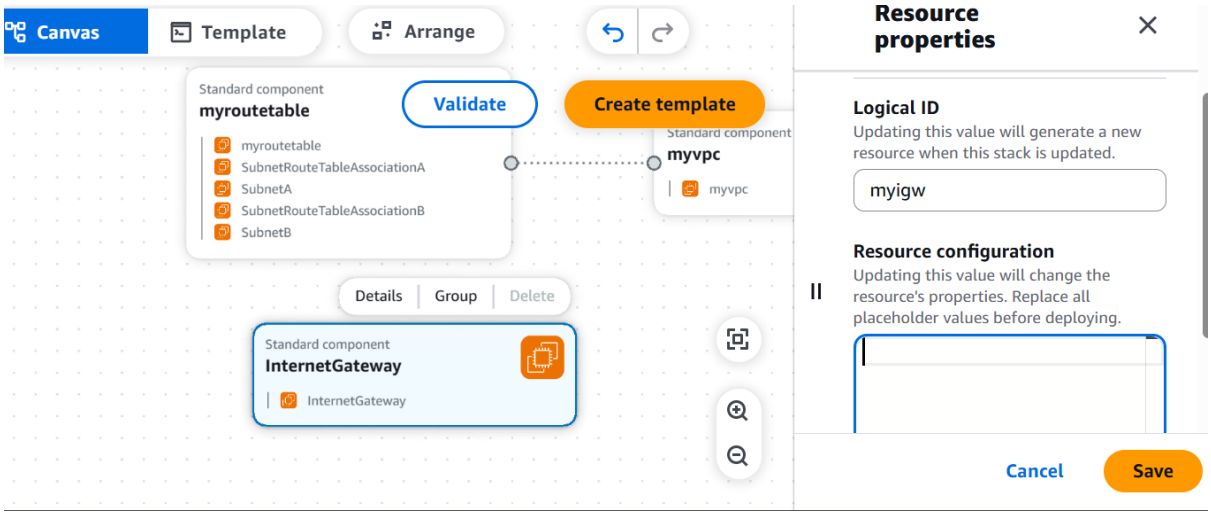


Internetgateway:

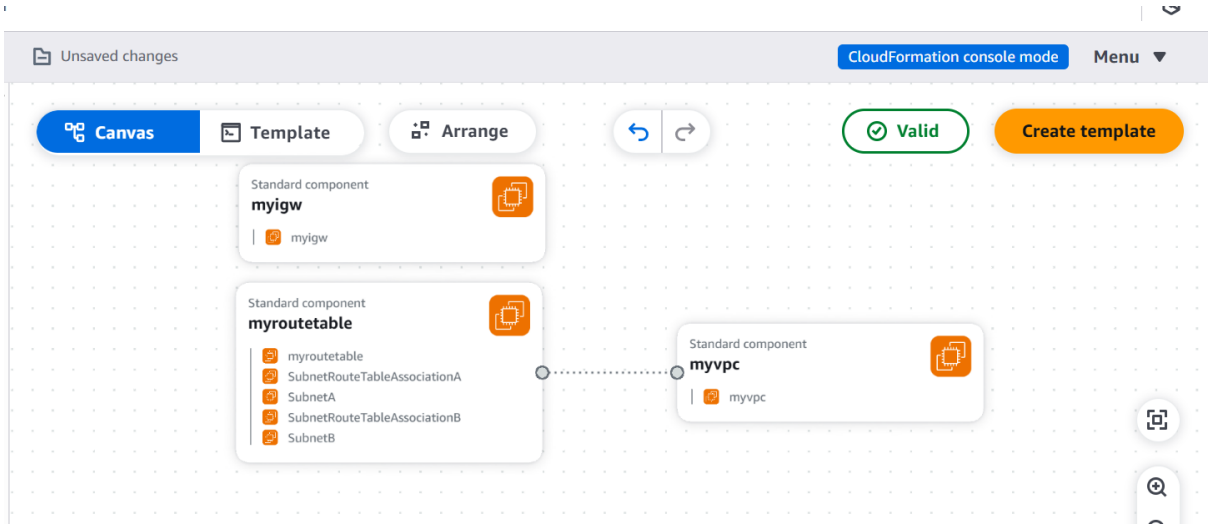


Drag and drop internetgateway

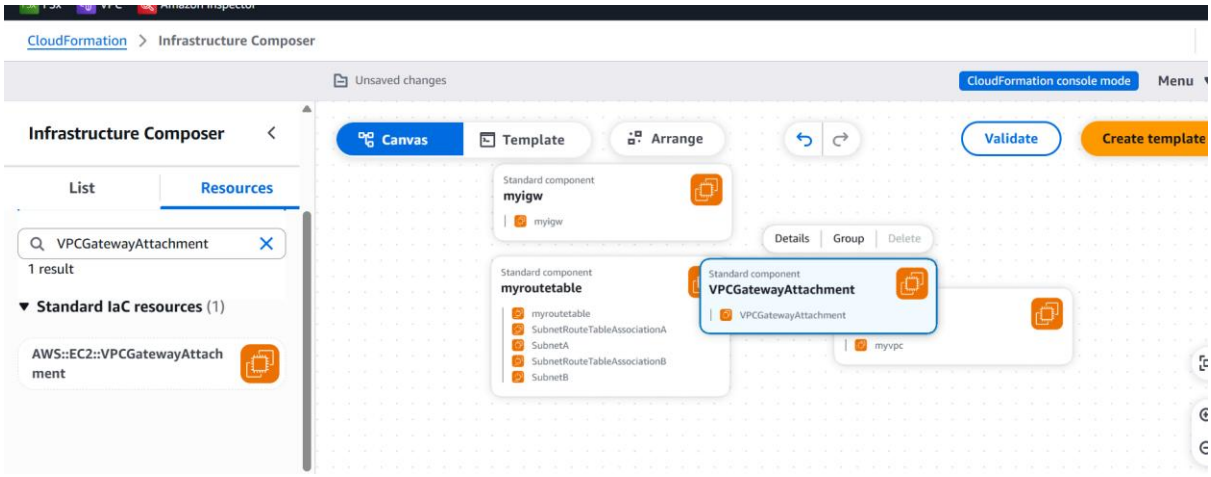


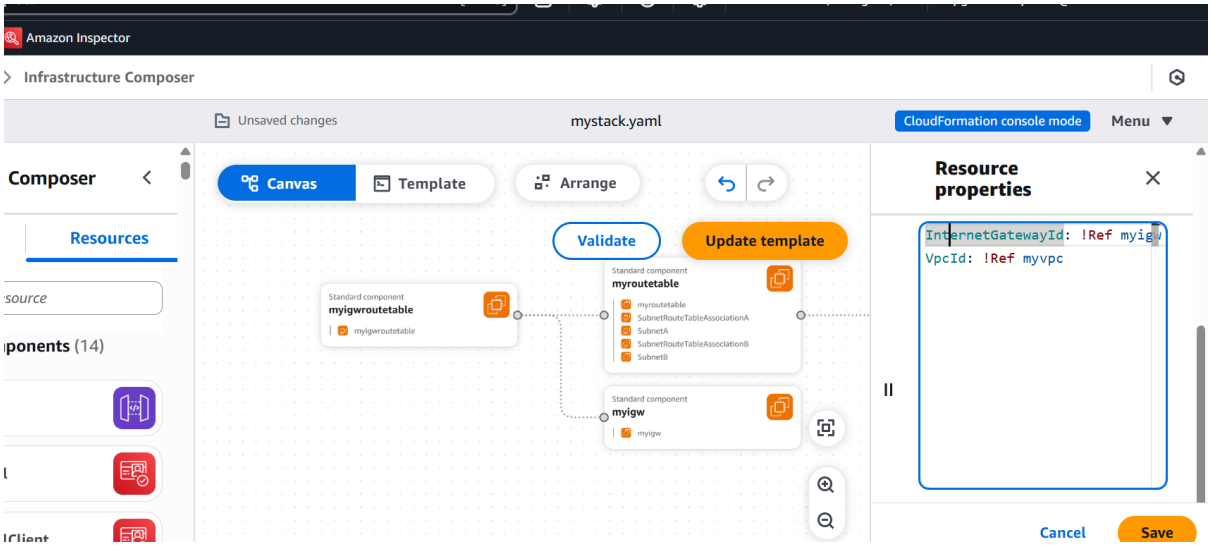


Validate



Check for VPCGatewayAttachment:



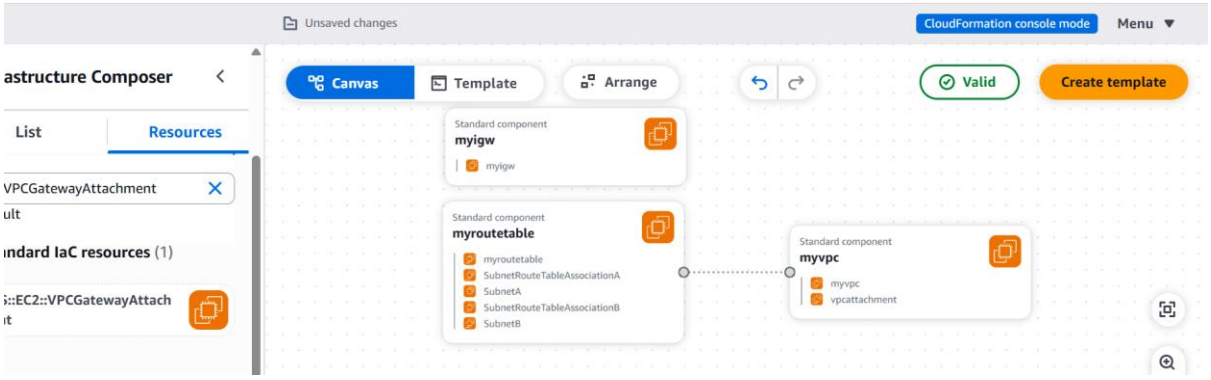


Code :

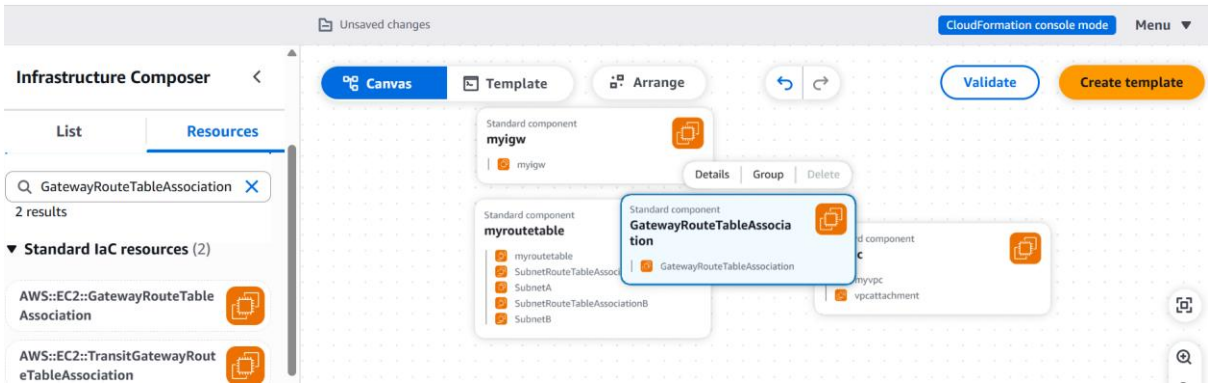
Int ernetGatewayId: !Ref myigw

VpcId: !Ref myvpc

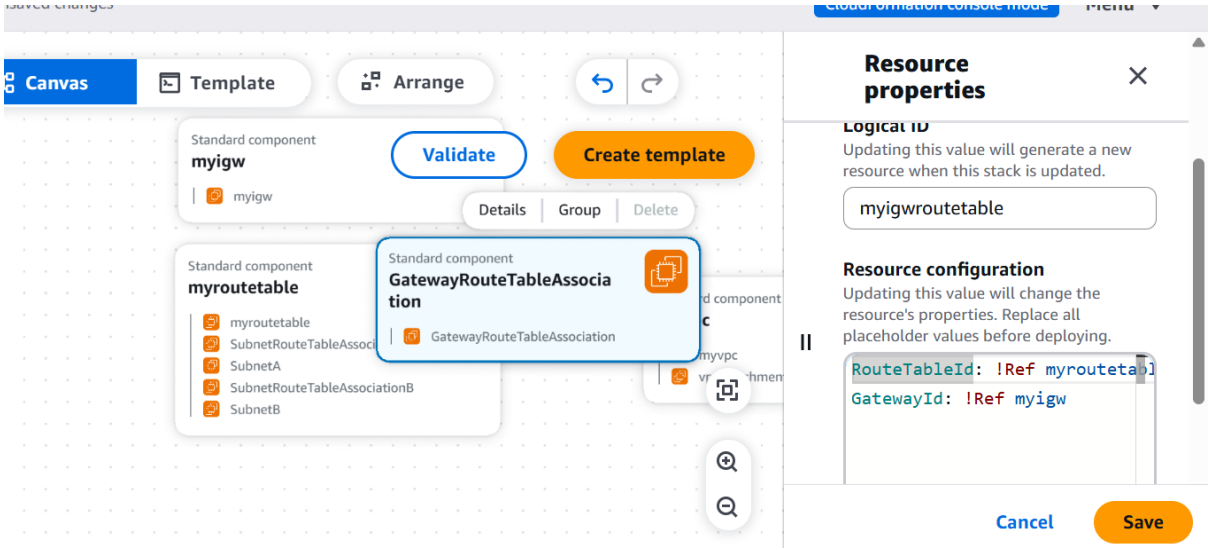
Save and validate



Next search for GatewayRouteTableAssociation



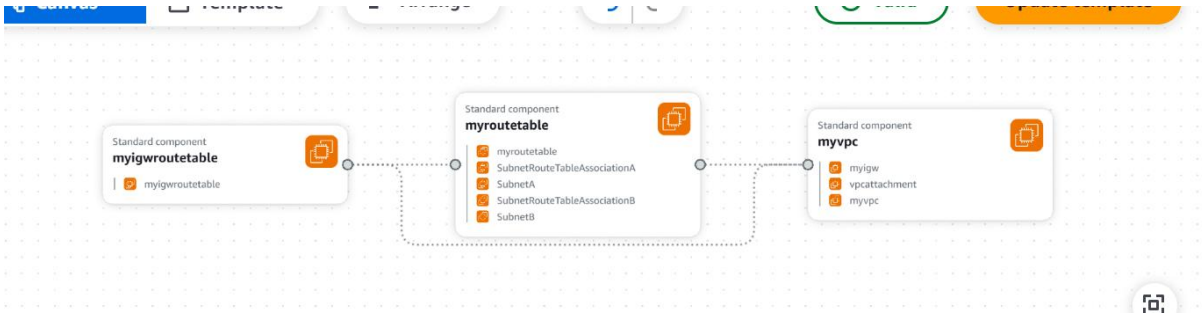
Double click and provide details



Code

RouteTableId: !Ref myroutetable

GatewayId: !Ref myigw



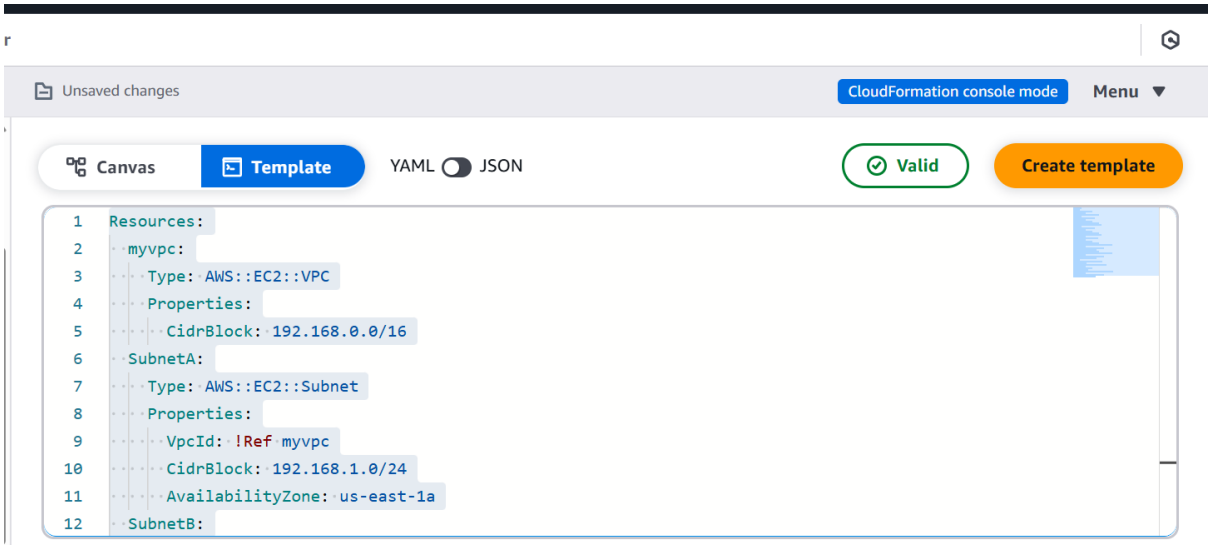
Click on validate

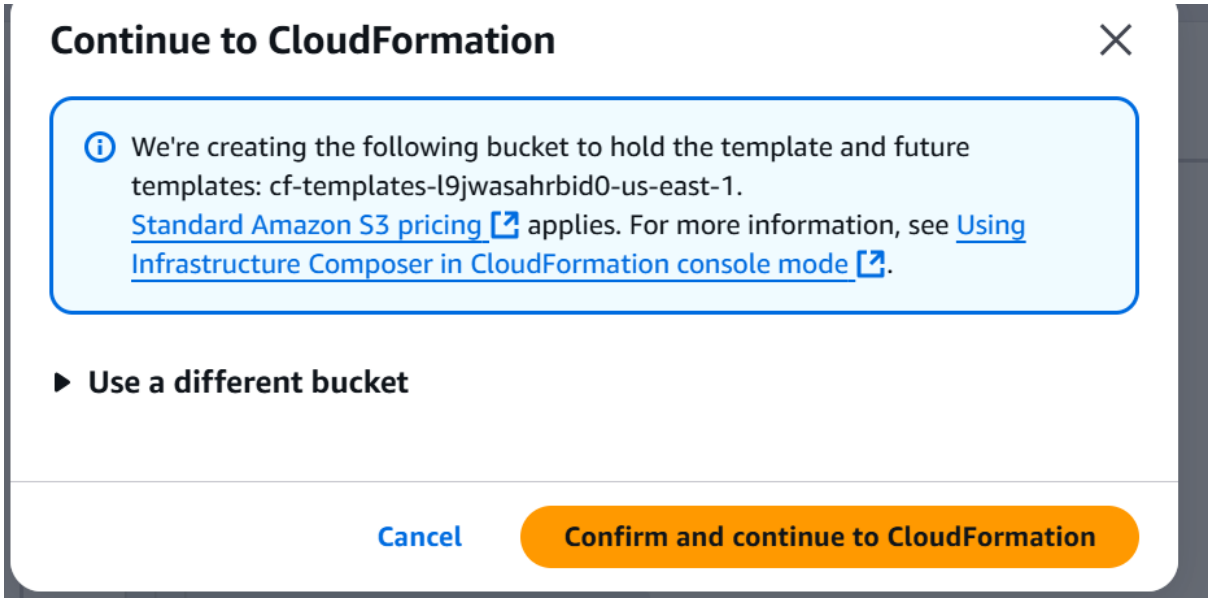
Code:

```
Resources:
  myvpc:
    Type: AWS::EC2::VPC
    Properties:
      CidrBlock: 192.168.0.0/16
  SubnetA:
    Type: AWS::EC2::Subnet
    Properties:
      VpcId: !Ref myvpc
      CidrBlock: 192.168.1.0/24
      AvailabilityZone: us-east-1a
  SubnetB:
    Type: AWS::EC2::Subnet
    Properties:
      VpcId: !Ref myvpc
      CidrBlock: 192.168.2.0/24
      AvailabilityZone: us-east-1b
  myroutetable:
    Type: AWS::EC2::RouteTable
    Properties:
      VpcId: !Ref myvpc
```

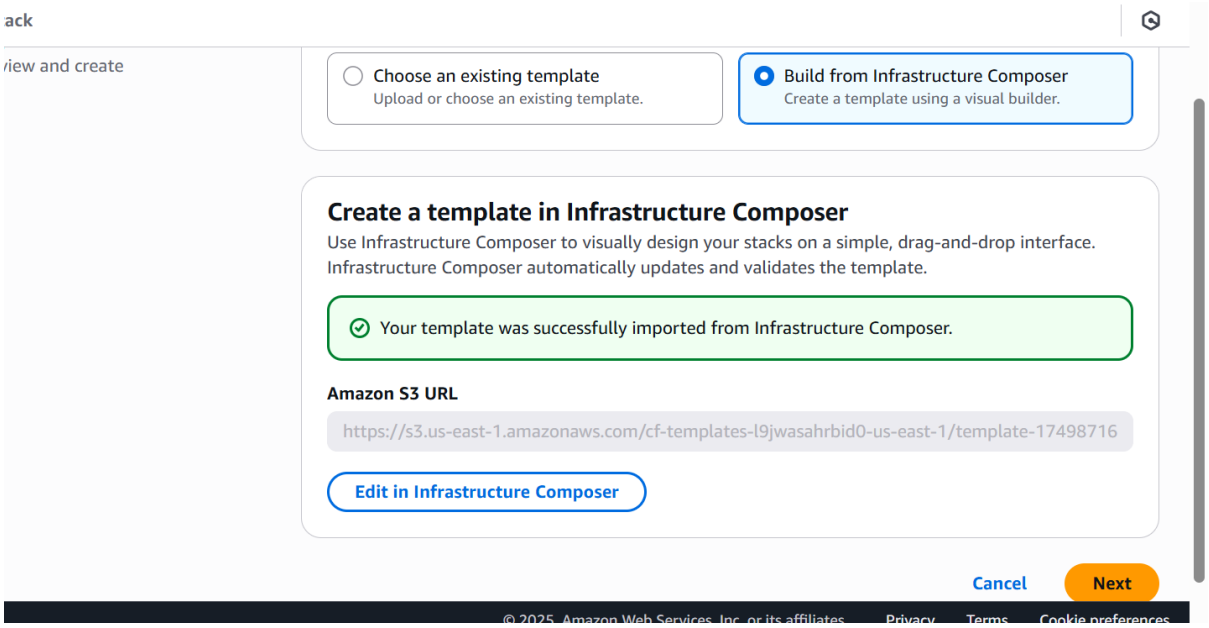
```
SubnetRouteTableAssociationA:
  Type: AWS::EC2::SubnetRouteTableAssociation
  Properties:
    RouteTableId: !Ref myroutetable
    SubnetId: !Ref SubnetA
SubnetRouteTableAssociationB:
  Type: AWS::EC2::SubnetRouteTableAssociation
  Properties:
    RouteTableId: !Ref myroutetable
    SubnetId: !Ref SubnetB
myigw:
  Type: AWS::EC2::InternetGateway
vpcattachment:
  Type: AWS::EC2::VPCGatewayAttachment
  Properties:
    Int ernetGatewayId: !Ref myigw
    VpcId: !Ref myvpc
myigwroutetable:
  Type: AWS::EC2::GatewayRouteTableAssociation
  Properties:
    RouteTableId: !Ref myroutetable
    GatewayId: !Ref myigw
```

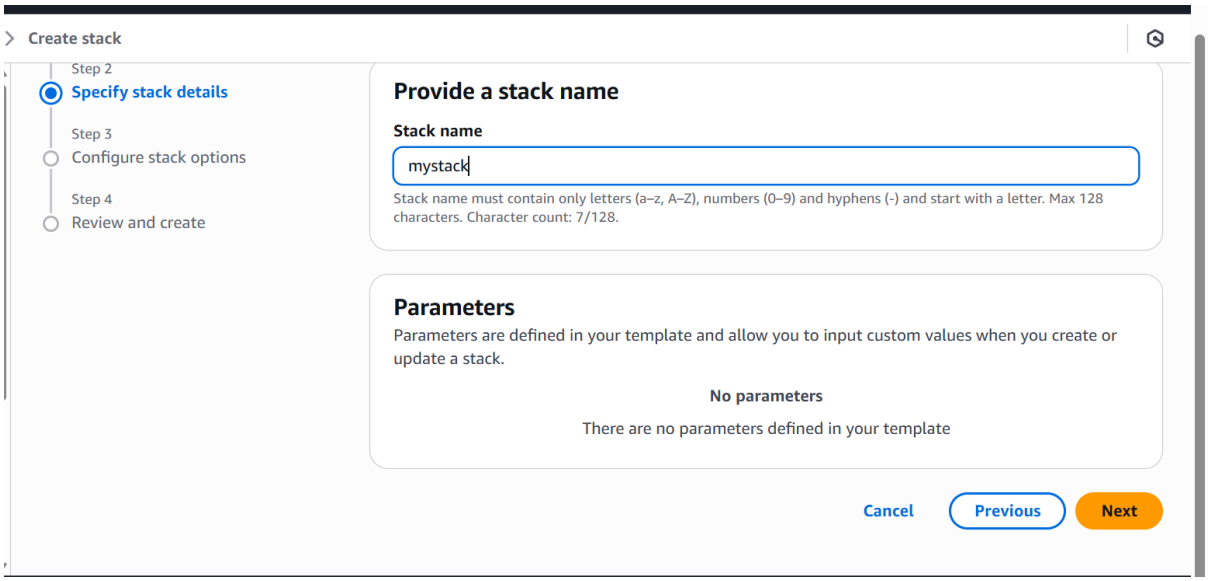
Click on create template

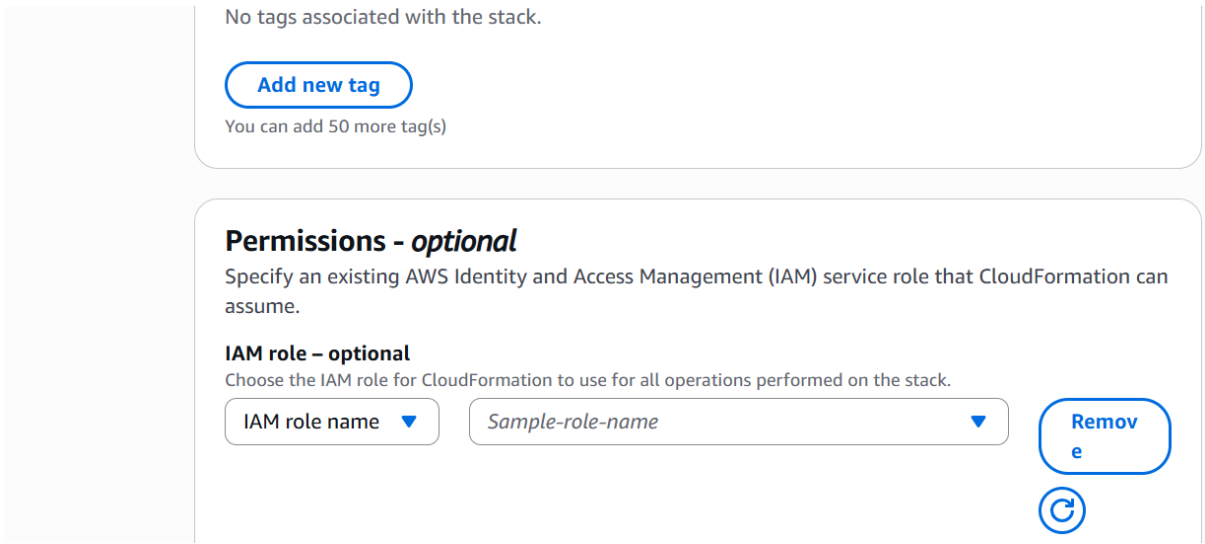


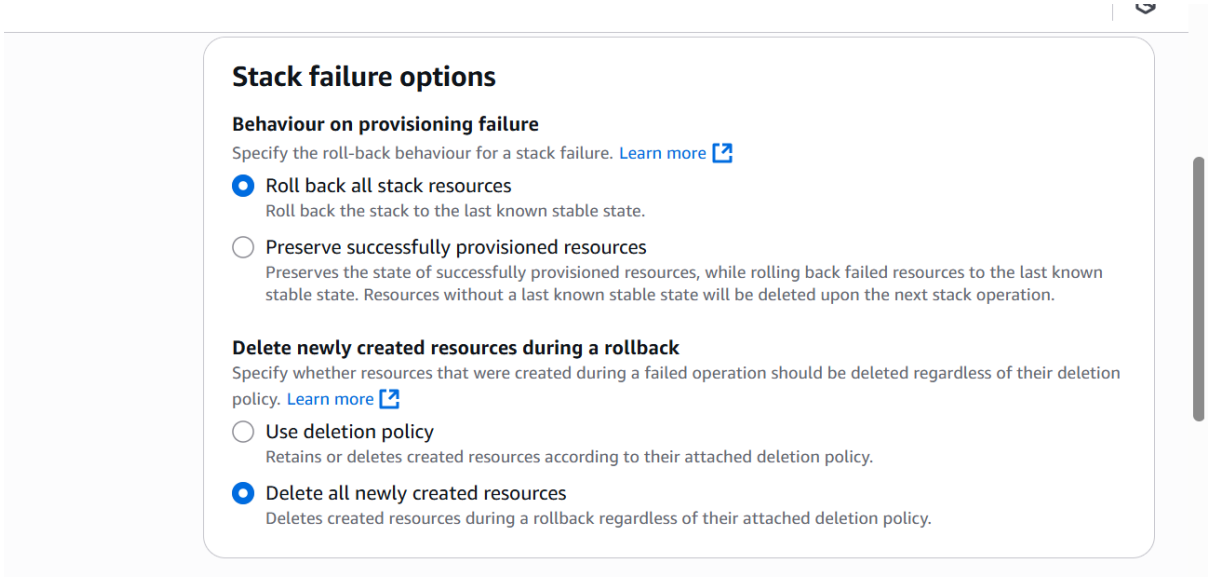


Click on next

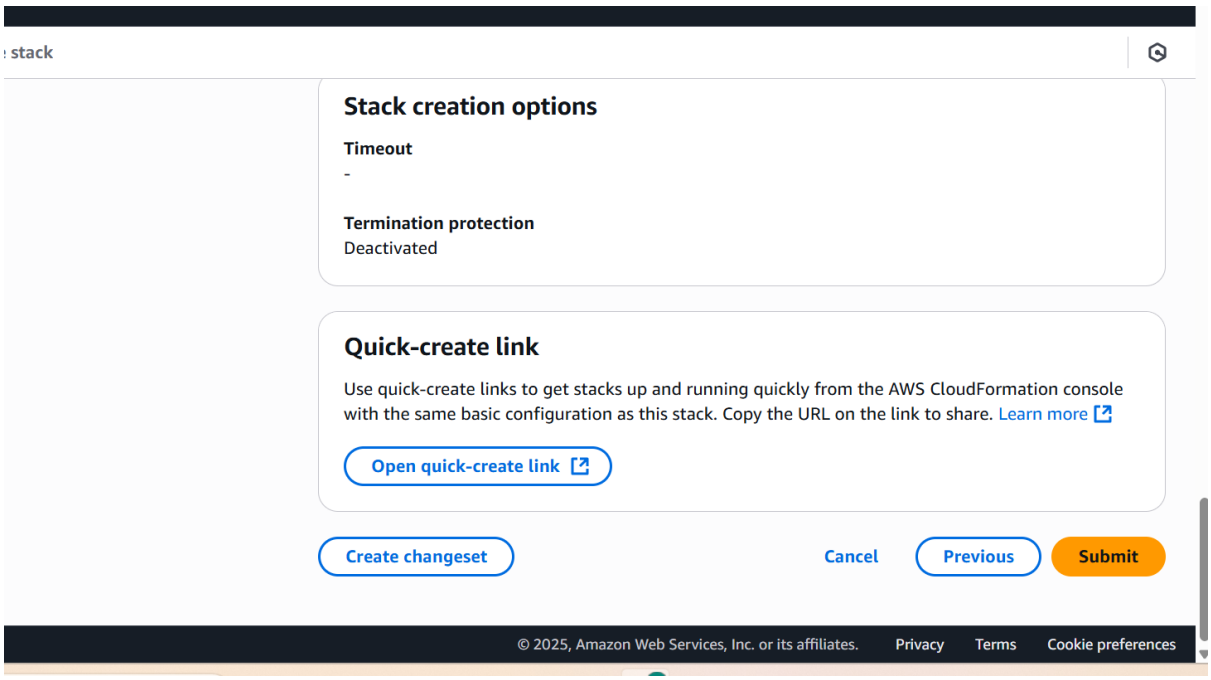

ack
iew and create
Choose an existing template
Upload or choose an existing template.
Build from Infrastructure Composer
Create a template using a visual builder.
Create a template in Infrastructure Composer
Use Infrastructure Composer to visually design your stacks on a simple, drag-and-drop interface. Infrastructure Composer automatically updates and validates the template.
Your template was successfully imported from Infrastructure Composer.
Amazon S3 URL
https://s3.us-east-1.amazonaws.com/cf-templates-l9jwasahrbid0-us-east-1/template-17498716
Edit in Infrastructure Composer
Cancel
Next
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

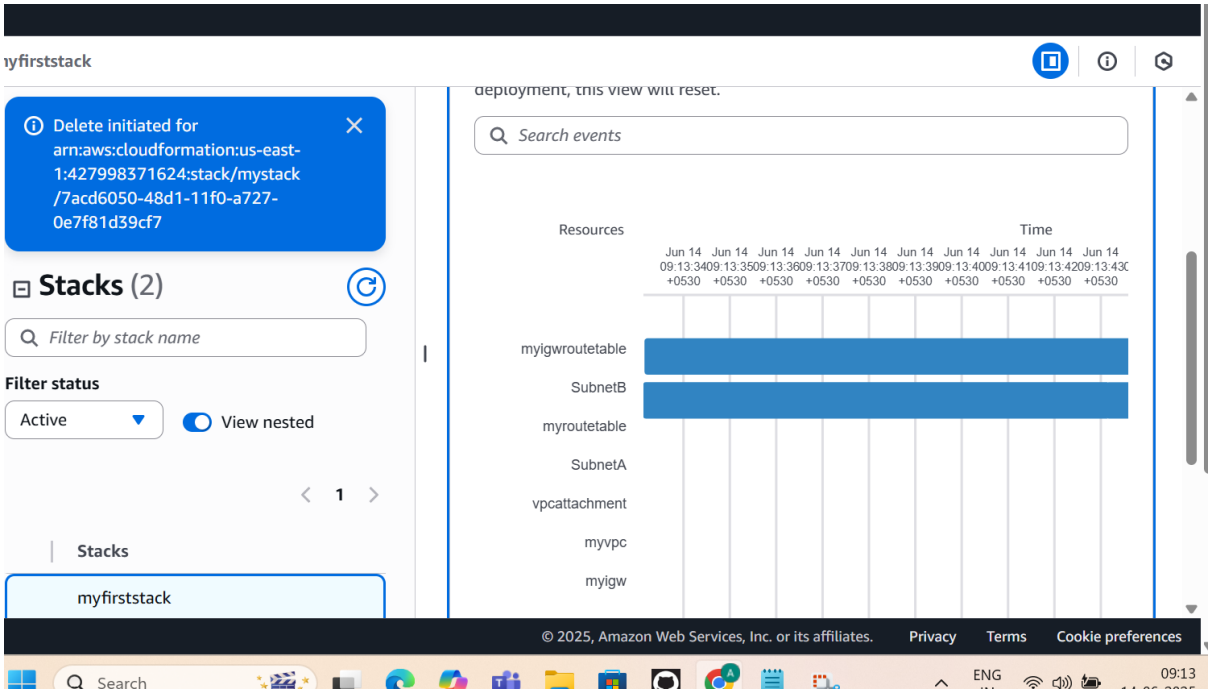

Create stack
Step 2
Specify stack details
Step 3
Configure stack options
Step 4
Review and create
Provide a stack name
Stack name
mystack
Stack name must contain only letters (a–z, A–Z), numbers (0–9) and hyphens (-) and start with a letter. Max 128 characters. Character count: 7/128.
Parameters
Parameters 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


No tags associated with the stack.
Add new tag
You can add 50 more tag(s)
Permissions - optional
Specify an existing AWS Identity and Access Management (IAM) service role that CloudFormation can assume.
IAM role – optional
Choose the IAM role for CloudFormation to use for all operations performed on the stack.
IAM role name
Sample-role-name
Remove



Click on submit




myfirststack
Delete initiated for arn:aws:cloudformation:us-east-1:427998371624:stack/mystack/7acd6050-48d1-11f0-a727-0e7f81d39cf7
Stacks (2)
Filter by stack name
Filter status
Active
View nested
1
Stacks
myfirststack
deployment, this view will reset.
Search events
Resources
Time
myigwroutetable
SubnetB
myroutetable
SubnetA
vpcattachment
myvpc
myigw
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences
Search
ENG
09:13