A CLOUD GURU

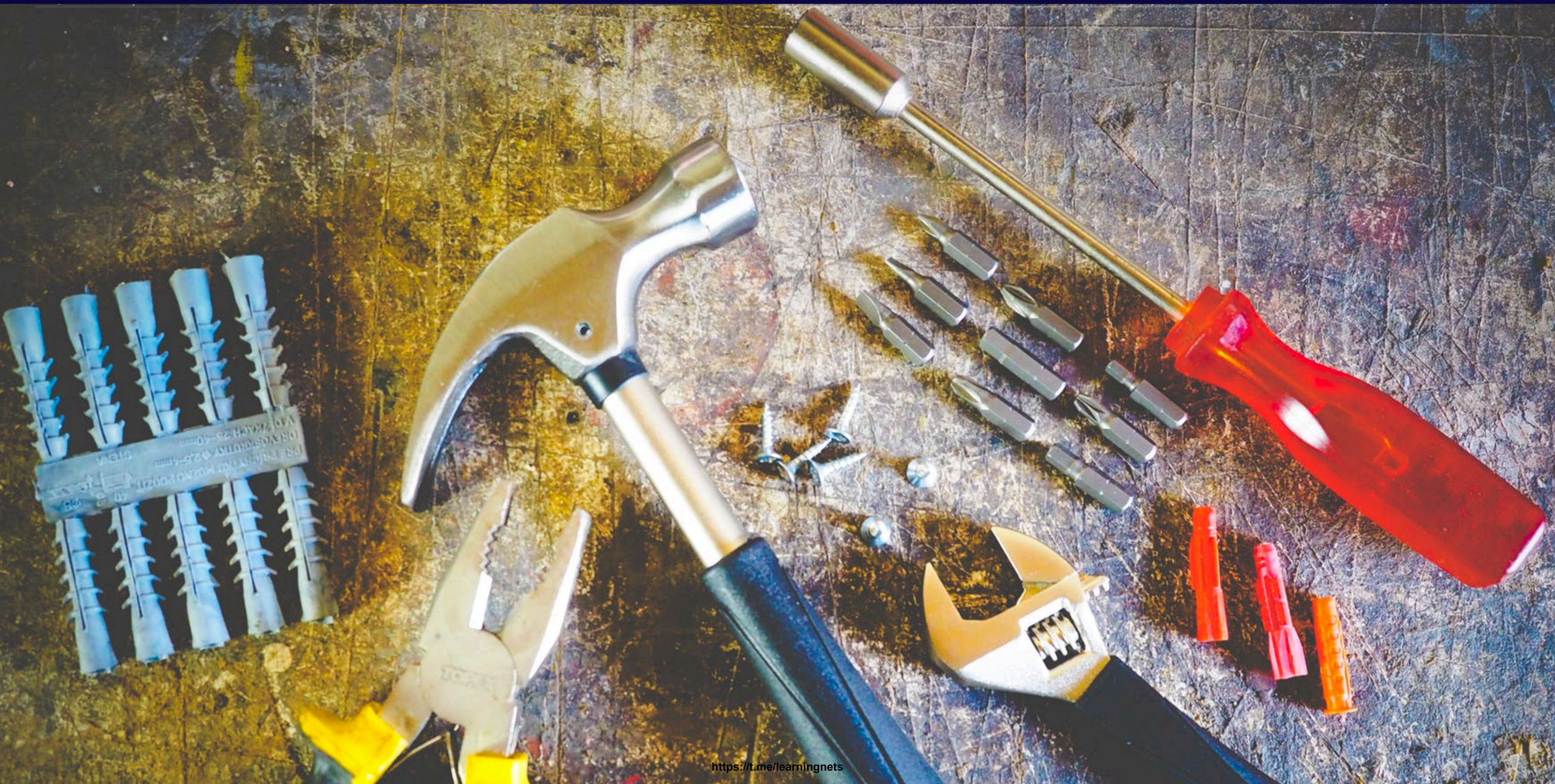
CloudFormation Part 1

Brock Tubre
TECHNICAL INSTRUCTOR

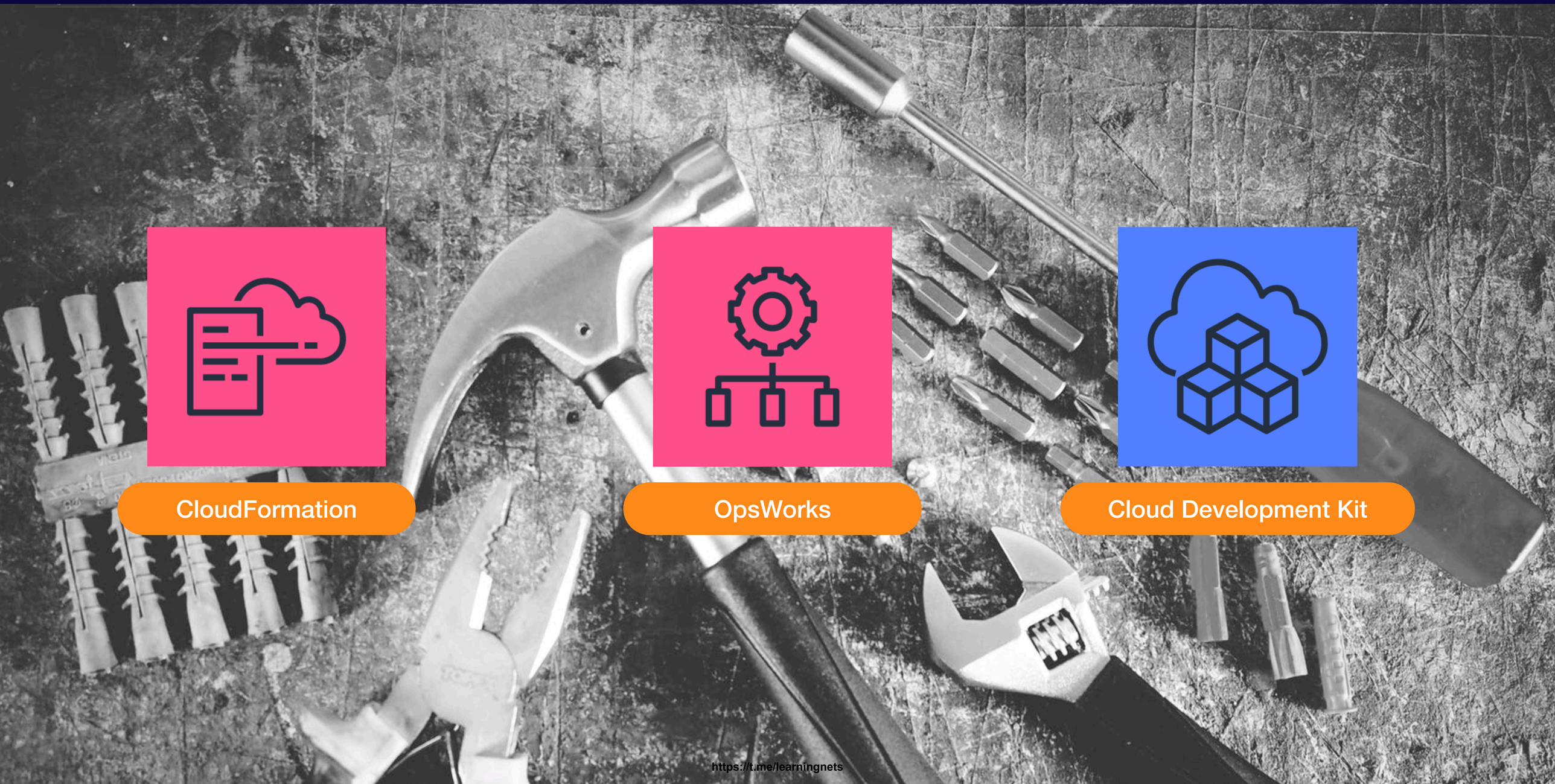
CLOUDFORMATION

Automation Tool Set

A CLOUD GURU

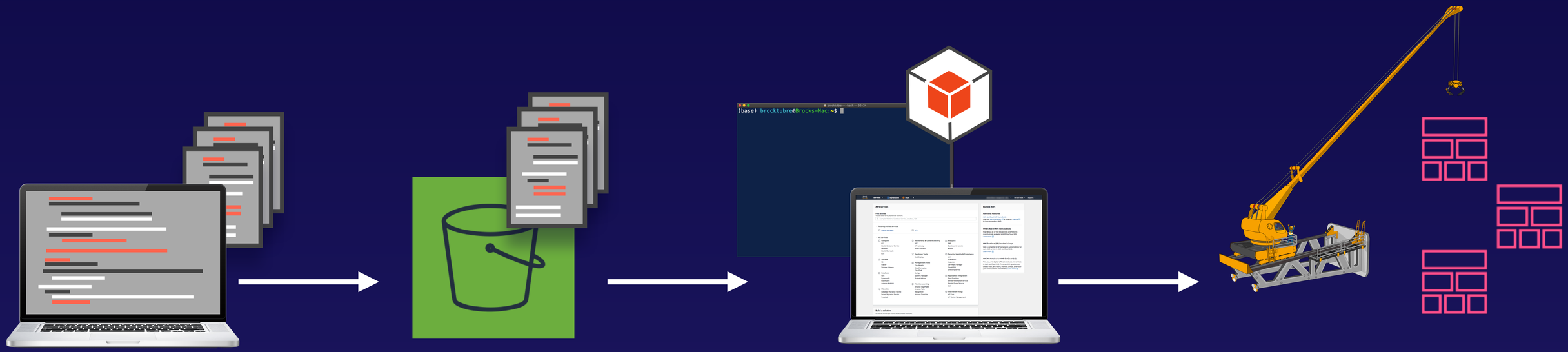


Automation Tool Set



CloudFormation

A CLOUD GURU



Code templates in JSON or YAML.

Upload templates from local files or from S3.

Create a stack using API, CLI, or console.

Stacks, infrastructure, and resources are created.

CloudFormation Components

1

Templates

The documents that describe the environment that you want to create and launch in AWS. Templates are written in JSON or YAML.

2

Stacks

A stack is deployed from a template and represents the resources and infrastructure that are connected together and belong to an application.

3

Change Sets

Once a stack has been successfully launched, we can use change sets to perform updates on the deployed infrastructure.

Template Sections

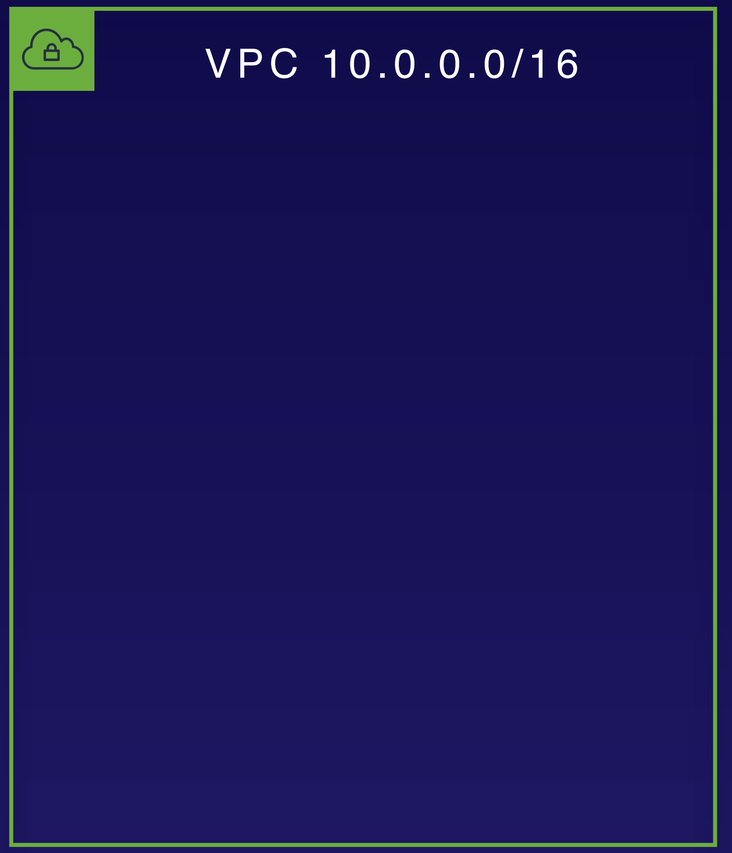
Section	Purpose
`AWSTemplateFormatVersion`	Template version.
`Description`	Describes what the template does.
`Metadata`	Allows us to pass metadata to services and instances at the time of creation.
`Parameters`	These values can be used as “variables” in the `Resource` section. These can be defaulted or users can select from several options you specify.
`Mappings`	Used to map parameters to the environment or region.

Template Sections

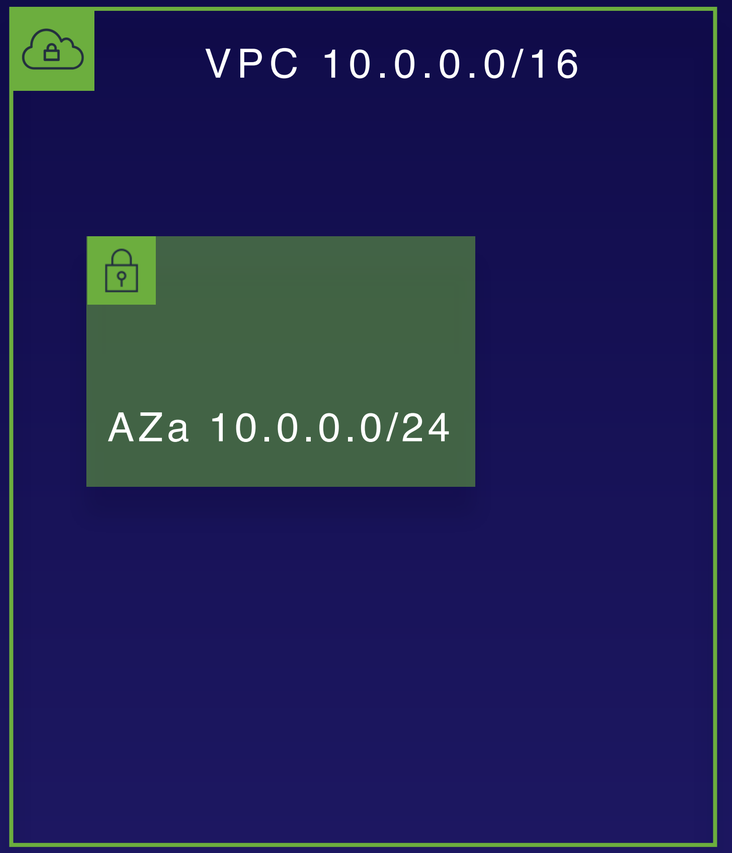
Section	Purpose
Conditions	Used to specify conditional operations, mappings, or whether resources will be created.
Transform	Used to define macros to process the template.
★ Resources	The only required section of a template. It defines the actual resources that will be created when the template is launched.
Outputs	Returns any piece of information we want after the deployment is complete.

Creating Network Services


```
Resources:
  LnCustomVPC:
    Type: AWS::EC2::VPC
    Properties:
      CidrBlock: '10.0.0.0/16'
```

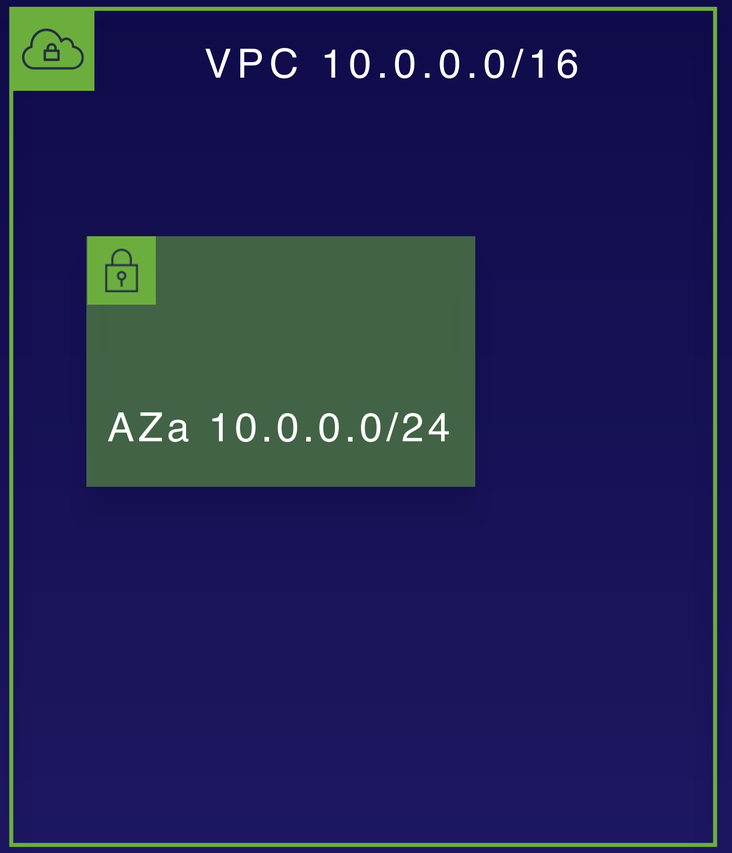


Creating Network Services



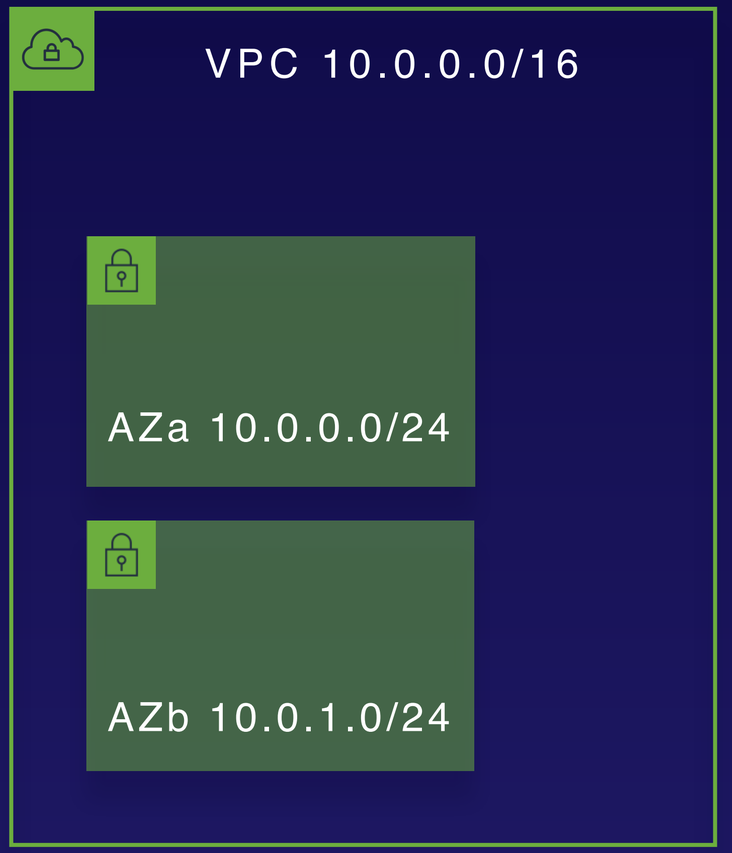
Creating Network Services


```yaml
Resources:
  LnCustomVPC:
    Type: AWS::EC2::VPC
    Properties:
      CidrBlock: '10.0.0.0/16'
  LnSubnet1a:
    Type: AWS::EC2::Subnet
    Properties:
      CidrBlock: '10.0.0.0/24'
      MapPublicIpOnLaunch: true
      AvailabilityZone: 'us-east-1a'
      VpcId: !Ref LnCustomVPC
```



Creating Network Services


```yaml
Resources:
  LnCustomVPC:
    Type: AWS::EC2::VPC
    Properties:
      CidrBlock: '10.0.0.0/16'
  LnSubnet1a:
    Type: AWS::EC2::Subnet
    Properties:
      CidrBlock: '10.0.0.0/24'
      MapPublicIpOnLaunch: true
      AvailabilityZone: 'us-east-1a'
      VpcId: !Ref LnCustomVPC
  LnSubnet1b:
    Type: AWS::EC2::Subnet
    Properties:
      CidrBlock: '10.0.1.0/24'
      MapPublicIpOnLaunch: true
      AvailabilityZone: 'us-east-1b'
      VpcId: !Ref LnCustomVPC
```

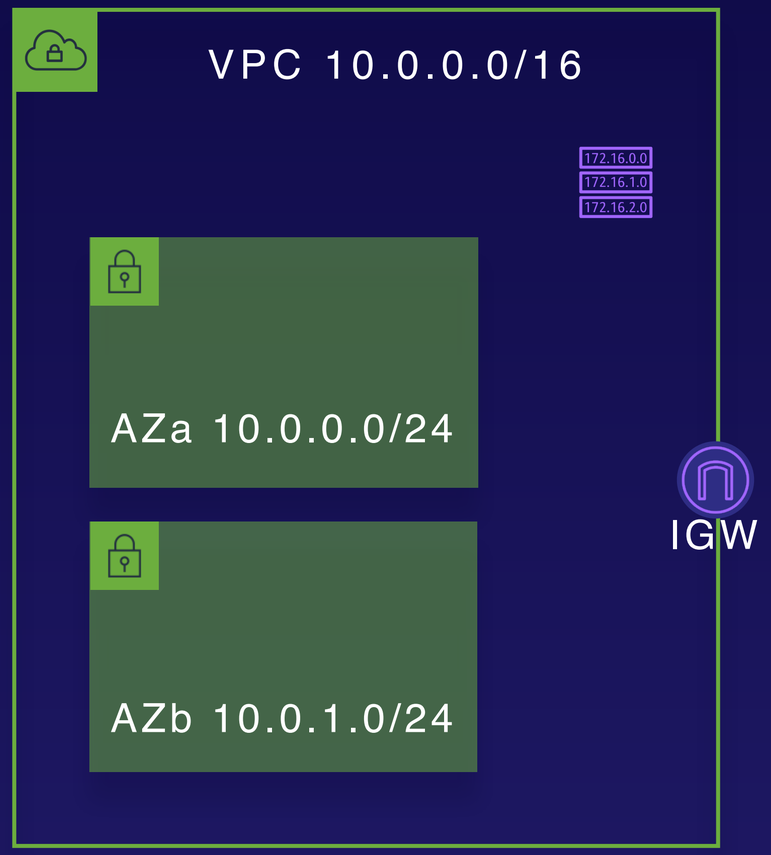


Creating Network Services


```
Resources:
  …
  LnCustomInternetGateway:
    Type: AWS::EC2::InternetGateway
  LnAttachIGtoVPC:
    Type: AWS::EC2::VPCGatewayAttachment
    Properties:
      InternetGatewayId: !Ref LnCustomInternetGateway
      VpcId: !Ref LnCustomVPC
```

Creating Network Services

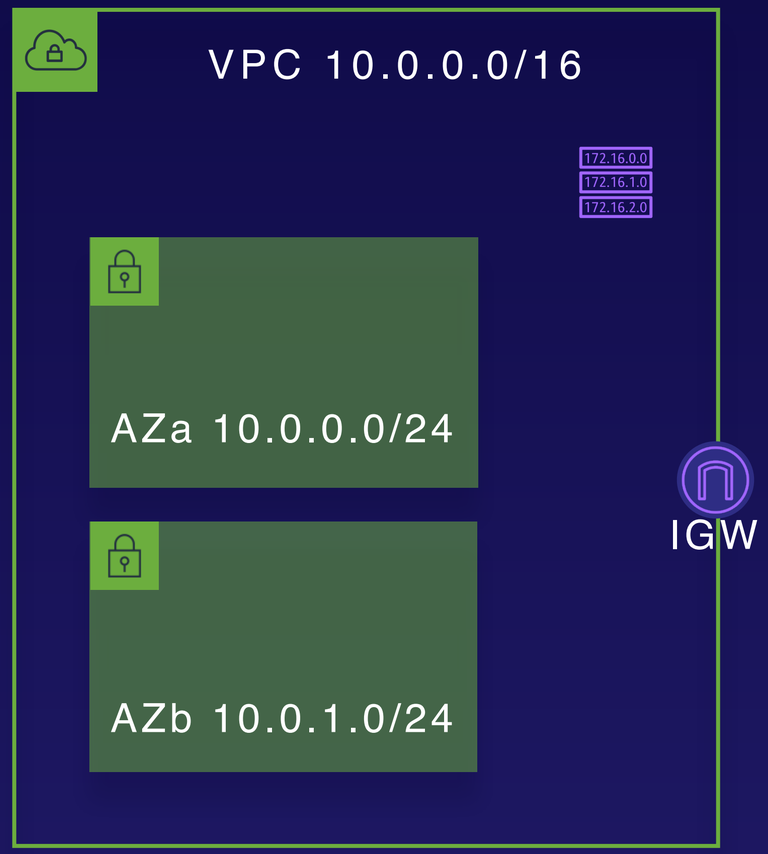

```
Resources:
  …
 LnCustomPublicRouteTable:
    Type: AWS::EC2::RouteTable
    Properties:
      VpcId: !Ref LnCustomVPC
```



Creating Network Services

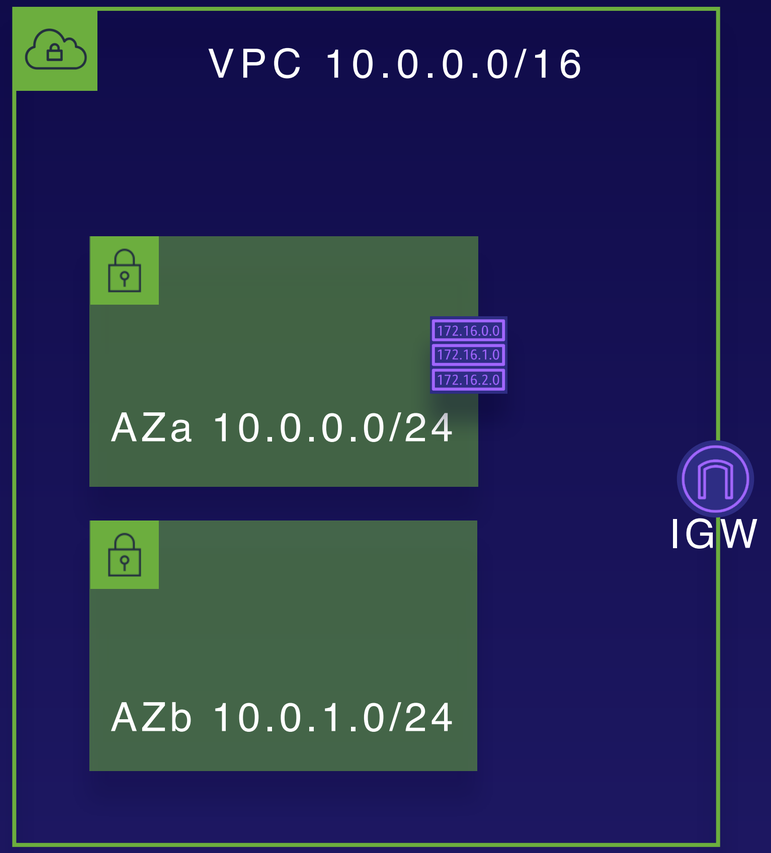

```
Resources:
  …
 LnCustomPublicRouteTable:
    Type: AWS::EC2::RouteTable
    Properties:
      VpcId: !Ref LnCustomVPC
  LnAddIGInternetRule:
    Type: AWS::EC2::Route
    Properties:
      RouteTableId: !Ref LnCustomPublicRouteTable
      DestinationCidrBlock: 0.0.0.0/0
      GatewayId: !Ref LnCustomInternetGateway
```

Destination	Target
10.0.0.0/16	local
0.0.0.0/0	IGW



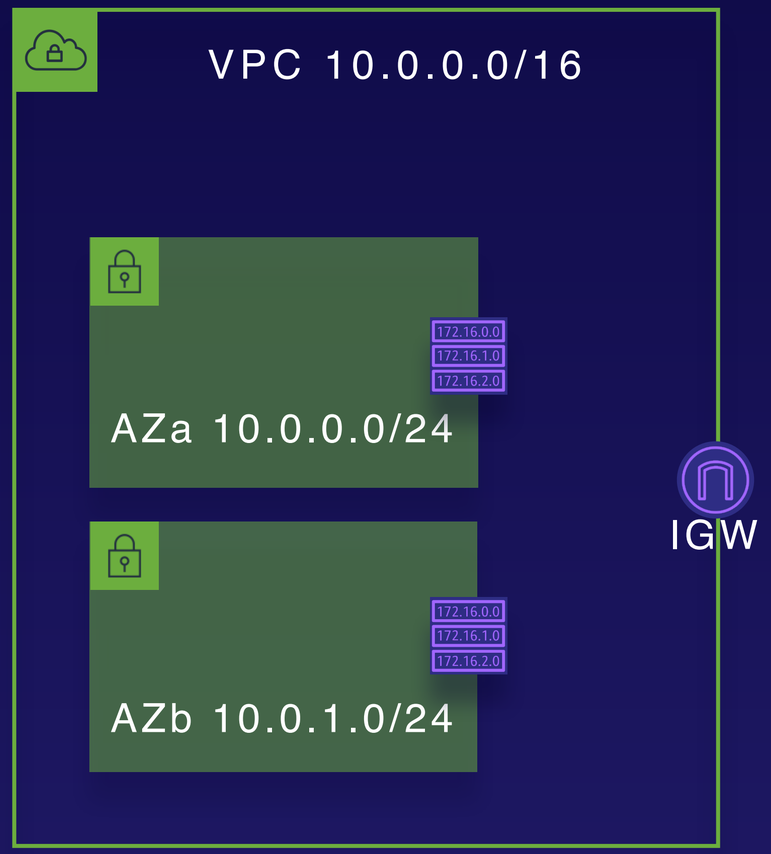
Creating Network Services


```
Resources:
  …
 LnCustomPublicRouteTable:
    Type: AWS::EC2::RouteTable
    Properties:
      VpcId: !Ref LnCustomVPC
  LnAddIGInternetRule:
    Type: AWS::EC2::Route
    Properties:
      RouteTableId: !Ref LnCustomPublicRouteTable
      DestinationCidrBlock: 0.0.0.0/0
      GatewayId: !Ref LnCustomInternetGateway
  LnAssociatePublicSubnetAWithRouteTable:
    Type: AWS::EC2::SubnetRouteTableAssociation
    Properties:
      RouteTableId: !Ref LnCustomPublicRouteTable
      SubnetId: !Ref LnSubnet1a
```



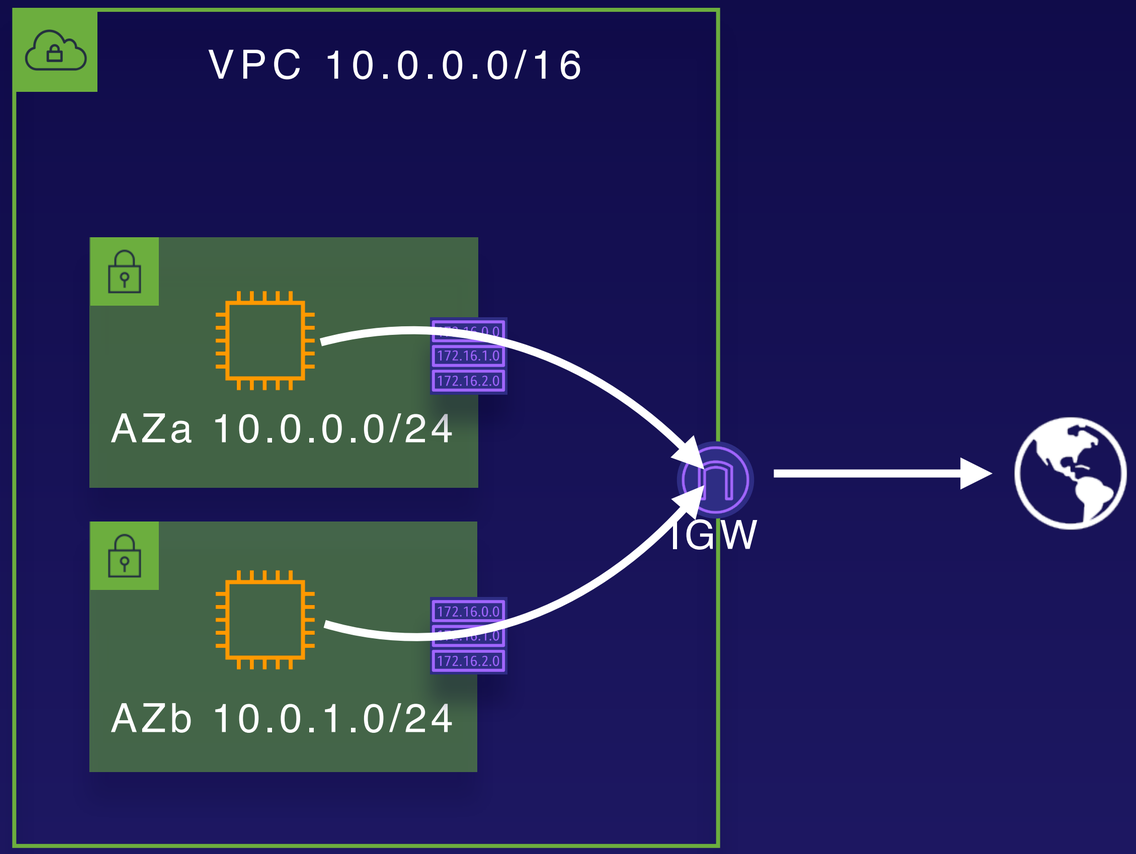
Creating Network Services


```
Resources:
  …
 LnCustomPublicRouteTable:
    Type: AWS::EC2::RouteTable
    Properties:
      VpcId: !Ref LnCustomVPC
  LnAddIGInternetRule:
    Type: AWS::EC2::Route
    Properties:
      RouteTableId: !Ref LnCustomPublicRouteTable
      DestinationCidrBlock: 0.0.0.0/0
      GatewayId: !Ref LnCustomInternetGateway
  LnAssociatePublicSubnetAWithRouteTable:
    Type: AWS::EC2::SubnetRouteTableAssociation
    Properties:
      RouteTableId: !Ref LnCustomPublicRouteTable
      SubnetId: !Ref LnSubnet1a
  LnAssociatePublicSubnetBWithRouteTable:
    Type: AWS::EC2::SubnetRouteTableAssociation
    Properties:
      RouteTableId: !Ref LnCustomPublicRouteTable
      SubnetId: !Ref LnSubnet1b
```



Creating Network Services


```
Resources:
  …
 LnCustomPublicRouteTable:
    Type: AWS::EC2::RouteTable
    Properties:
      VpcId: !Ref LnCustomVPC
  LnAddIGInternetRule:
    Type: AWS::EC2::Route
    Properties:
      RouteTableId: !Ref LnCustomPublicRouteTable
      DestinationCidrBlock: 0.0.0.0/0
      GatewayId: !Ref LnCustomInternetGateway
  LnAssociatePublicSubnetAWithRouteTable:
    Type: AWS::EC2::SubnetRouteTableAssociation
    Properties:
      RouteTableId: !Ref LnCustomPublicRouteTable
      SubnetId: !Ref LnSubnet1a
  LnAssociatePublicSubnetBWithRouteTable:
    Type: AWS::EC2::SubnetRouteTableAssociation
    Properties:
      RouteTableId: !Ref LnCustomPublicRouteTable
      SubnetId: !Ref LnSubnet1b
```



A CLOUD GURU

EXAM TIP

Determining Free IP Ranges

Fn::Cidr

The **Fn::Cidr** intrinsic function returns an array of CIDR block ranges that you can use when creating subnets.

Using Fn::Cidr

A CLOUD GURU

YAML

```
!Cidr [ '10.0.0.0/16', 6, 8 ]
```

JSON

```
{ "Fn::Cidr" : [ "10.0.0.0/16", "6", "8"] }
```

Number of CIDR generated.

Using Fn::Cidr

A CLOUD GURU

YAML

```
!Cidr [ '10.0.0.0/16', 6, 8 ]
```

JSON

```
{ "Fn::Cidr" : [ "10.0.0.0/16", "6", "8"] }
```


The subnet mask bits.
32 - *this_number* = subnet mask

Using Fn::Cidr

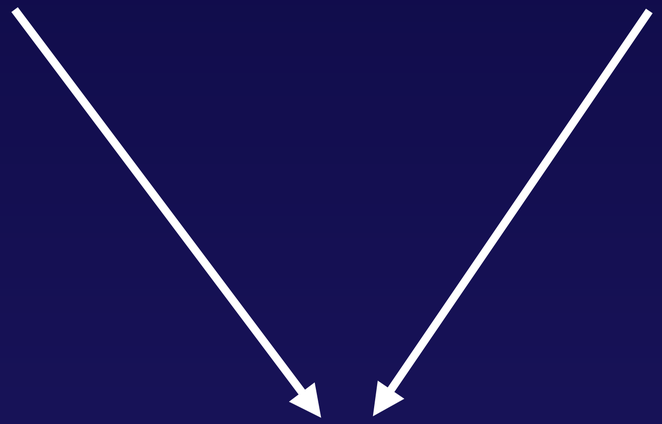
A CLOUD GURU

YAML

```
!Cidr [ '10.0.0.0/16', 6, 8 ]
```

JSON

```
{ "Fn::Cidr" : [ "10.0.0.0/16", "6", "8"] }
```



```
[
    10.0.0.0/24,
    10.0.1.0/24,
    10.0.2.0/24,
    10.0.3.0/24,
    10.0.4.0/24,
    10.0.5.0/24
]
```

Fast Takeaways

Understand the capabilities of CloudFormation and the different template sections.

Know how to use Fn::Cidr to specify CIDR ranges for subnets.