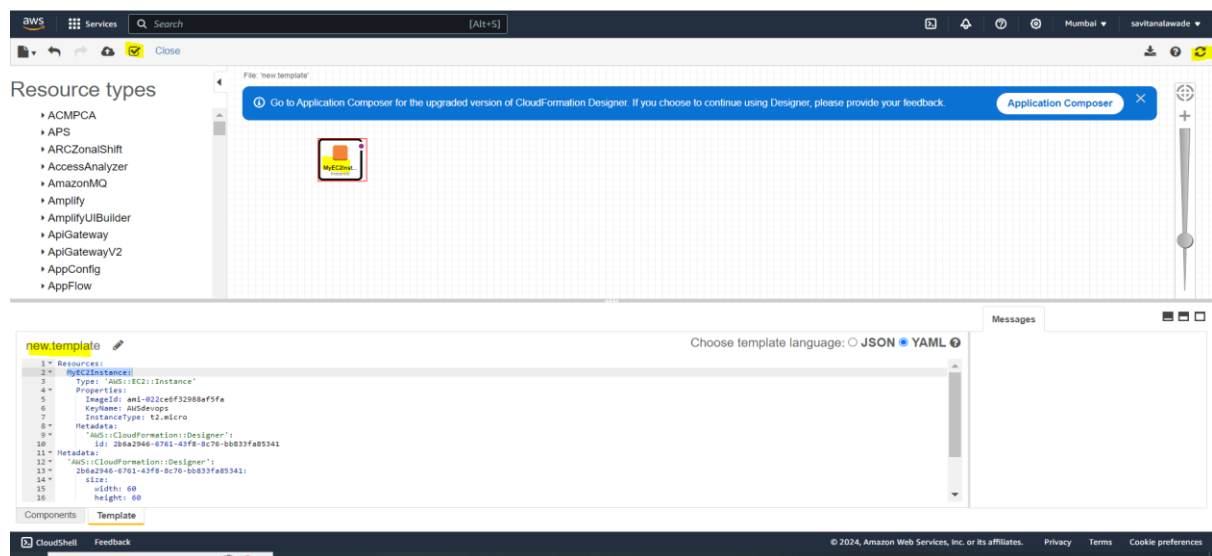


CloudFormation 27th_Sep_2024 SavitaNalawade

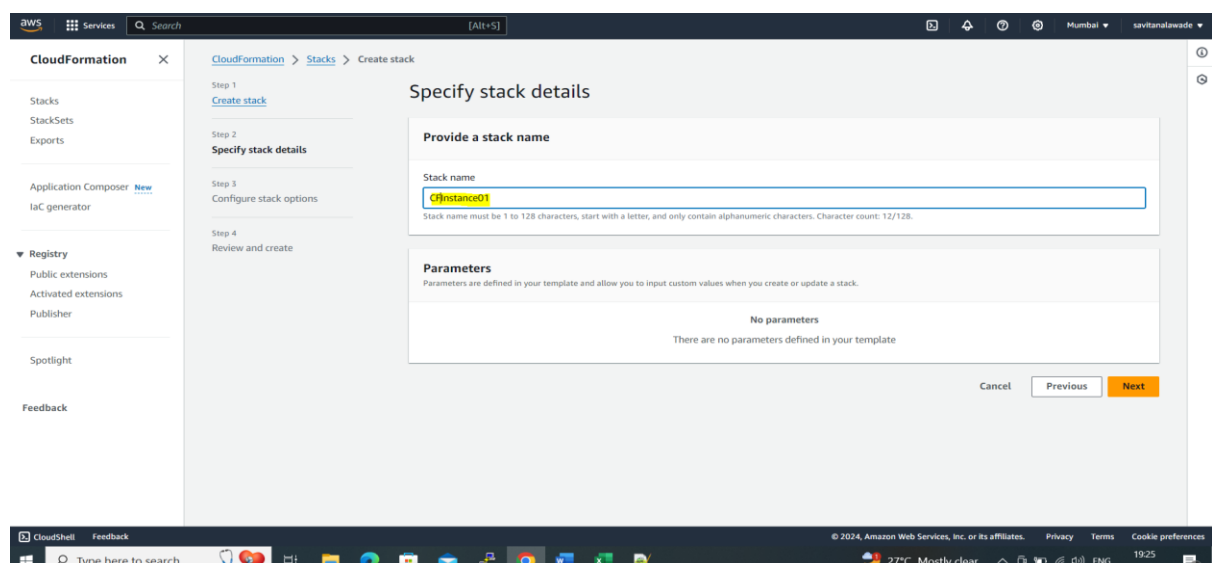
CloudFormation:

- AWS CloudFormation is a service provided by AWS that enables users to model and manage infrastructure resources in an automated and secure manner.
- Using CloudFormation, developers can define and provision AWS infrastructure resources
- In Cloudformation we can use JSON- or YAML Language to code.

1) Create instance through CF write code in template ->refresh->Validate template



Create Stack



Instance have been created

The screenshot shows the AWS CloudFormation console. On the left, the 'Stacks' list shows 'CFInstance01' with a status of 'CREATE_COMPLETE'. The main panel displays the 'Resources' tab for 'CFInstance01', showing a single resource 'MyEC2Instance' with a status of 'CREATE_COMPLETE'. The resource details include its Logical ID, Physical ID, Type (AWS::EC2::Instance), and Status.

2)Create S3 Bucket

The screenshot shows the AWS CloudFormation Designer. A new template is being created, and the 'Resources' section shows an 'S3Bucket' resource. The resource details include its Logical ID, Physical ID, Type (AWS::S3::Bucket), and Status. The designer interface includes a left sidebar with 'Resource types' and a central canvas for the template.

Bucket is created

The screenshot shows the AWS CloudFormation console. On the left, the 'Stacks' list shows 'CFInstance01' with a status of 'CREATE_COMPLETE'. The main panel displays the 'Resources' tab for 'CFInstance01', showing two resources: 'CFInstance01' and 'CFBucket01', both with a status of 'CREATE_COMPLETE'. The resource details include their Logical IDs, Physical IDs, Types (AWS::EC2::Instance and AWS::S3::Bucket), and Status.

3) Template to setup lifecycle configuration

The screenshot shows the AWS CloudFormation console. On the left, the 'Stacks' list shows three stacks: 'BucketLifecycle', 'CFS3Bucket01', and 'CFInstance01', all with a status of 'CREATE_COMPLETE'. The main panel displays the 'Template' tab for the 'BucketLifecycle' stack. The template is an AWS::S3::Bucket resource named 'DemoBucket' with a lifecycle configuration. The lifecycle configuration has two rules: 'Rule for log prefix' and 'Rule for log suffix'. The 'Rule for log prefix' has a prefix of 'logs', a status of 'Enabled', and a transition to 'STANDARD_IA' after 40 days. The 'Rule for log suffix' has a suffix of '90', a status of 'Enabled', and a transition to 'GLACIER' after 90 days. The template also includes an output for 'BucketName'.

```
AWSTemplateFormatVersion: 2010-09-09
Description: Template to setup lifecycle configuration
Parameters:
  BucketName:
    Type: String
    Description: Name of the bucket on which lifecycle configuration will apply
    Default: cfs3bucket001
Resources:
  DemoBucket:
    Type: 'AWS::S3::Bucket'
    Properties:
      BucketName: !Ref BucketName
      LifecycleConfiguration:
        Rules:
          - Id: Rule for log prefix
            Prefix: logs
            Status: Enabled
            Transitions:
              - TransitionInDays: 40
                StorageClass: STANDARD_IA
              - TransitionInDays: 90
                StorageClass: GLACIER
            ExpirationInDays: 365
Outputs:
  BucketName:
    Value: !Ref DemoBucket
    Description: Name of the sample Amazon S3 bucket with a lifecycle configuration.
```

The screenshot shows the AWS CloudFormation console with the 'Events' tab selected for the 'BucketLifecycle' stack. The events table shows the following events:

Timestamp	Logical ID	Status	Detailed status	Status reason
2024-09-29 20:04:56 UTC+0530	BucketLifecycle	CREATE_COMPLETE	-	-
2024-09-29 20:04:55 UTC+0530	DemoBucket	CREATE_COMPLETE	-	-
2024-09-29 20:04:42 UTC+0530	DemoBucket	CREATE_IN_PROGRESS	-	Resource creation Initiated
2024-09-29 20:04:41 UTC+0530	DemoBucket	CREATE_IN_PROGRESS	-	-
2024-09-29 20:04:39 UTC+0530	BucketLifecycle	CREATE_IN_PROGRESS	-	User Initiated

4) Create an EC2 instance, EIP and associate with instance

The screenshot shows the AWS CloudFormation console with the 'new.template' editor open. The editor displays a template for creating an EC2 instance, an Elastic IP (EIP), and associating the EIP with the instance. The template includes parameters for 'ImageId', 'InstanceType', 'KeyName', and 'EIP'. The resources section defines 'DemoInstance' (AWS::EC2::Instance), 'DemoEIP' (AWS::EC2::EIP), and 'DemoAssociate' (AWS::EC2::Associate). The 'DemoAssociate' resource is associated with 'DemoInstance' and 'DemoEIP'.

```
AWSTemplateFormatVersion: 2010-09-09
Description: Template to Create an EC2 Instance, EIP and associate with Instance
Parameters:
  ImageId:
    Type: String
    Description: Linux 2 AMI
    Default: ami-022ce6f32880af9fa
  InstanceType:
    Type: String
    Description: Choosing t2.micro
    Default: t2.micro
  KeyName:
    Type: String
    Description: SSH KeyPair to login to the instance
    Default: AWSDevops
Resources:
  DemoInstance:
    Type: 'AWS::EC2::Instance'
    Properties:
      ImageId: !Ref ImageId
      InstanceType: !Ref InstanceType
      KeyName: !Ref KeyName
  DemoEIP:
    Type: 'AWS::EC2::EIP'
    Properties:
      Domain: vpc
      InstanceId: !Ref DemoInstance
  DemoAssociate:
    Type: 'AWS::EC2::Associate'
    Properties:
      InstanceId: !Ref DemoInstance
      EIPId: !Ref DemoEIP
```

CloudFormation console showing the InstanceEIP stack. The stack is in the 'CREATE_COMPLETE' state. The Resources tab shows two resources: DemoElasticIP and DemoInstance, both in the 'CREATE_COMPLETE' state.

Logical ID	Physical ID	Type	Status	Module
DemoElasticIP	65.2.104.46	AWS::EC2::EIP	CREATE_COMPLETE	-
DemoInstance	i-0e417f5e2c37f56a6	AWS::EC2::Instance	CREATE_COMPLETE	-

5) EC2 Instance with EBS Volume

CloudFormation console showing the 'new.template' file. The template is valid. The Resource types section lists ACMPA, APS, ARCOZonalShift, AccessAnalyzer, and AmazonMQ. The template content is as follows:

```
1 AWSTemplateFormatVersion: 2010-09-09
2 Description: First EC2 Instance
3 Resources:
4   FirstLinuxEC2Instance:
5     Type: AWS::EC2::Instance
6     Properties:
7       AvailabilityZone: ap-south-1a
8       ImageId: ami-0b2ceaf32088af9fa
9       InstanceType: t2.micro
10      Tags:
11        - Key: Name
12          Value: Linux
13        - Key: SecurityGroupIds
14          Value: sg-0e07d179a3f1da26c
15      Volumes:
16        - Device: /dev/sdf
17          VolumeId: !Ref NewVolume
18      NewVolume:
19        Type: AWS::EC2::Volume
20        Properties:
21          Size: 2
22          AvailabilityZone: ap-south-1a
23        Tags:
24          - Key: Name
25            Value: Linux
26        DeletionPolicy: Snapshot
```

CloudFormation > **Stacks** > **EC2vmwithVolume**

Stacks (5)

Filter by stack name

Filter status: Active View nested

Stacks

- EC2vmwithVolume
 - 2024-09-29 20:20:05 UTC+05:30
 - CREATE_COMPLETE
- InstanceEIP
 - 2024-09-29 20:13:30 UTC+05:30
 - CREATE_COMPLETE
- BucketLifecycle
 - 2024-09-29 20:04:39 UTC+05:30
 - CREATE_COMPLETE
- CFS3Bucket01
 - 2024-09-29 19:49:57 UTC+05:30
 - CREATE_COMPLETE

EC2vmwithVolume

Stack info | Events | **Resources** | Outputs | Parameters | Template | Change sets | Git sync

Resources (2)

Search resources

Logical ID	Physical ID	Type	Status	Module
FirstLinuxEC2Instance	i-0cd0839400e7f54ad	AWS::EC2::Instance	CREATE_COMPLETE	-
NewVolume	vol-075a857dd643005d0	AWS::EC2::Volume	CREATE_COMPLETE	-

6) EC2 with EBS Volume + Security Group

Resource types

- ACMPICA
- APS
- ARCZonalShift
- AccessAnalyzer
- AmazonMQ
- Amplify

new.template

Choose template language: JSON **YAML**

```

6 AvailabilityZone: ap-south-1a
7 ImageId: ami-0b2eef3208a5f5a
8 InstanceType: t2.micro
9 KeyName: "AddKeyPair"
10 BlockDeviceMappings:
11   - DeviceName: "/dev/sdn"
12     Ebs:
13       VolumeType: "io1"
14       Iops: "200"
15       DeleteOnTermination: "false"
16       VolumeSize: "5"
17     SecurityGroups:
18       - !Ref InstanceSecurityGroup01
19 InstanceSecurityGroup01:
20   Type: "AWS::EC2::SecurityGroup"
21   Properties:
22     GroupDescription: "Enable SSH access via port 22"
23     SecurityGroupIngress:
24     - CidrIp: 0.0.0.0/0
25       FromPort: 22
26       IpProtocol: tcp
27       ToPort: 22
28     CidrIp: 0.0.0.0/0
29       FromPort: 3386
30       IpProtocol: tcp
31       ToPort: 3386
32
  
```

CloudFormation > **Stacks** > **EC2EBSSG01**

Stacks (7)

Filter by stack name

Filter status: Active View nested

Stacks

- EC2EBSSG01
 - 2024-09-29 20:31:53 UTC+05:30
 - DELETE_IN_PROGRESS
- EC2EBSSG
 - 2024-09-29 20:28:11 UTC+05:30
 - DELETE_IN_PROGRESS
- EC2vmwithVolume
 - 2024-09-29 20:20:05 UTC+05:30
 - CREATE_COMPLETE
- InstanceEIP
 - 2024-09-29 20:13:30 UTC+05:30
 - CREATE_COMPLETE

EC2EBSSG01

Stack info | Events | **Resources** | Outputs | Parameters | Template | Change sets | Git sync

Resources (2)

Search resources

Logical ID	Physical ID	Type	Status	Module
InstanceSecurityGroup01	ec2ebssg01-InstanceSecurityGroup01-SGAd1c0hA1UJ	AWS::EC2::SecurityGroup	CREATE_COMPLETE	-
MyInstance	i-0271abe1ae0d7d230	AWS::EC2::Instance	CREATE_COMPLETE	-

7) Create EC2 instance & update security group and Public IP address of the newly created EC2 instance

The screenshot displays the AWS CloudFormation console. The top section shows the 'new.template' editor with a YAML template for an EC2 instance. The template includes parameters for image ID, instance type, and availability zone, and resources for the EC2 instance, security group, and IP address. The bottom section shows the 'EC2VMPublicIP' stack in the 'Resources' tab, listing the EC2 instance, security group, IP address, and IP association, all with a status of 'CREATE_COMPLETE'.

new.template

```
1 AWSTemplateFormatVersion: 2010-09-09
2 Description: Hello
3 Parameters:
4   - ImageId:
5     Type: String
6     Description: Linux 2 AMI
7     Default: ami-022ce8f32988af5fa
8   - InstanceType:
9     Type: String
10    Description: Linux 2 AMI
11    Default: t2.micro
12 Resources:
13   EC2Instance:
14     Type: AWS::EC2::Instance
15     Properties:
16       AvailabilityZone: ap-south-1a
17       ImageId: !Ref ImageId
18       InstanceType: !Ref InstanceType
19       KeyName: AWSDefault
20     SecurityGroups:
21       - !Ref InstanceSecurityGroup
22   InstanceSecurityGroup:
23     Type: AWS::EC2::SecurityGroup
24     Properties:
25       GroupDescription: "Enable SSH access via port 22"
26     SecurityGroupIngress:
27       - CidrIp: 0.0.0.0/0
28         FromPort: 22
29         IpProtocol: tcp
30         ToPort: 22
```

EC2VMPublicIP

Logical ID	Physical ID	Type	Status	Module
EC2Instance	i-0d0443f7074c41176	AWS::EC2::Instance	CREATE_COMPLETE	-
InstanceSecurityGroup	EC2VMPublicIP-InstanceSecurityGroup-rwmvB0bzc5Ah	AWS::EC2::SecurityGroup	CREATE_COMPLETE	-
IPAddress	13.202.155.24	AWS::EC2::EIP	CREATE_COMPLETE	-
IPAssoc	eipassoc-0f51e243bf87f09b6	AWS::EC2::EIPAssociation	CREATE_COMPLETE	-

8) Create VPC

The image displays two screenshots from the AWS CloudFormation console, illustrating the creation and deployment of a VPC architecture.

Top Screenshot: New Template Editor

The top screenshot shows the "new.template" editor in the AWS CloudFormation console. The left sidebar lists "Resource types" including ACMPCA, APS, ARCAZonalShift, AccessAnalyzer, AmazonMQ, Amplify, AmplifyUIBuilder, ApiGateway, and ApiGatewayV2. The main area displays a visual diagram of a VPC architecture with two subnets (PublicSub and PrivateSub) connected to an InternetGateway. The bottom pane shows the YAML template code for the VPC stack.

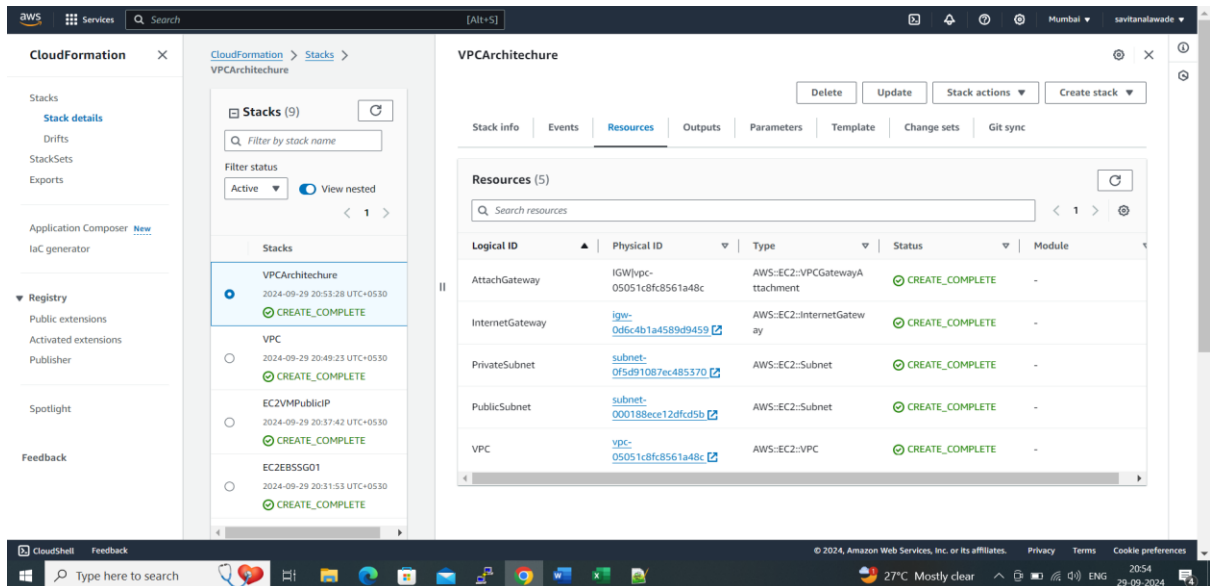
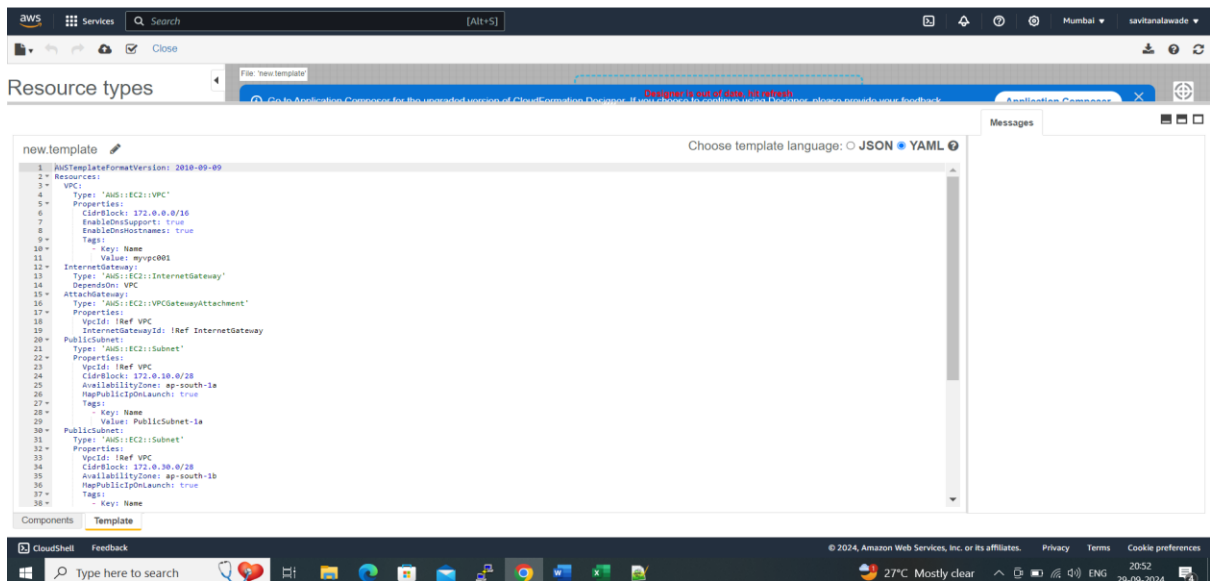
```
1: AWSTemplateFormatVersion: 2010-09-09
2: Resources:
3:   VPC:
4:     Type: 'AWS::EC2::VPC'
5:     Properties:
6:       CidrBlock: 10.0.0.0/16
7:       EnableDnsSupport: true
8:       EnableDnsHostnames: true
9:       Tags:
10:        - Key: Name
11:          Value: myvpc001
12:   InternetGateway:
13:     Type: 'AWS::EC2::InternetGateway'
14:     DependsOn: VPC
15:   AttachGateway:
16:     Type: 'AWS::EC2::VPCGatewayAttachment'
17:     Properties:
18:       VpcId: !Ref VPC
19:       InternetGatewayId: !Ref InternetGateway
20:   PublicSubnet:
21:     Type: 'AWS::EC2::Subnet'
22:     Properties:
23:       VpcId: !Ref VPC
```

Bottom Screenshot: CloudFormation Stack Details

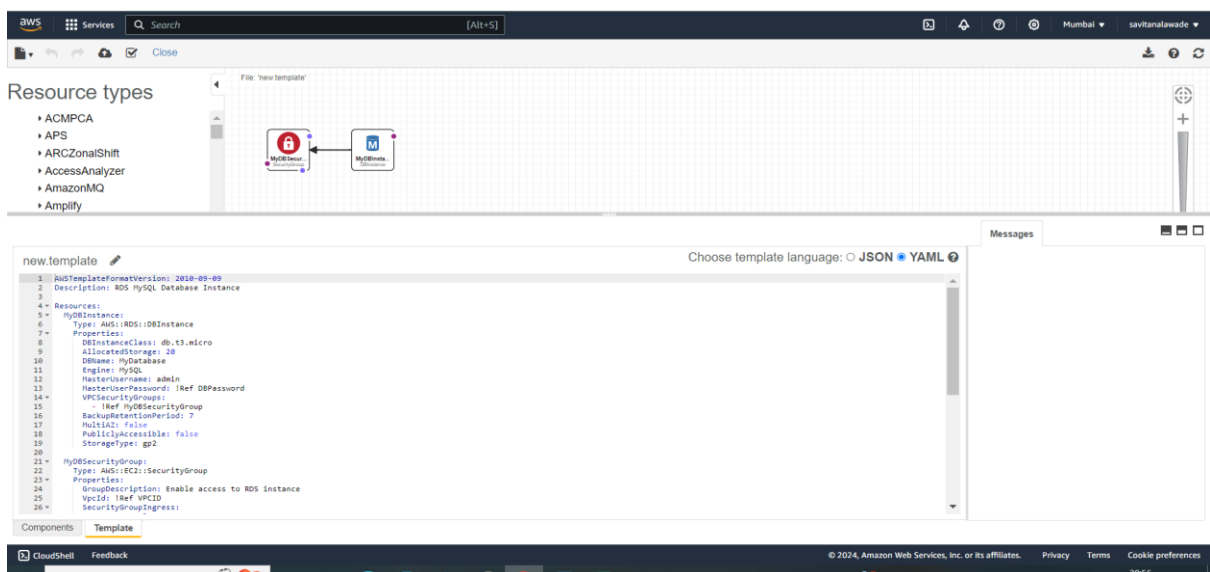
The bottom screenshot shows the "CloudFormation" console with the "Stacks" tab selected. The "Stacks (8)" list shows several stacks, including "VPC", "EC2VMPublicIP", "EC2BSSG01", "EC2vmwithVolume", and "InstanceEIP". The "VPC" stack is highlighted, and its details are shown in the right pane. The "Resources (5)" section lists the resources created by the VPC stack:

Logical ID	Physical ID	Type	Status	Module
AttachGateway	IGW/vpc-06329c0af6766b9df	AWS-EC2::VPCGatewayAttachment	CREATE_COMPLETE	-
InternetGateway	igw-01a22496a0ec55c9c	AWS-EC2::InternetGateway	CREATE_COMPLETE	-
PrivateSubnet	subnet-0526f70a9f9fduof	AWS-EC2::Subnet	CREATE_COMPLETE	-
PublicSubnet	subnet-070513856615bbba8	AWS-EC2::Subnet	CREATE_COMPLETE	-
VPC	vpc-06329c0af6766b9df	AWS-EC2::VPC	CREATE_COMPLETE	-

9)VPC Architecture



10) Create RDS MySQL Database Instance



Specify stack details

Specify stack details

Provide a stack name

Stack name
RDMySQLDBInstance
Stack name must be 1 to 128 characters, start with a letter, and only contain alphanumeric characters. Character count: 18/128.

Parameters
Parameters are defined in your template and allow you to input custom values when you create or update a stack.

DBPassword
The password for the DB admin account.

VPCID
The VPC ID for the database.
vpc-01ff4ab41ecc0386a

Cancel Previous Next

DBInstance and Security group are created

RDMySQLDBInstance

Stack info | Events | **Resources** | Outputs | Parameters | Template | Change sets | Git syn

Resources (2)

Logical ID	Physical ID	Type	Status
MyDBInstance	rdsmysqlbinstance-mydbinstance-3i5rmhcu79pm	AWS::RDS::DBInstance	CREATE_COMPLETE
MyDBSecurityGroup	sg-068d6abeed7bb07df	AWS::EC2::SecurityGroup	CREATE_COMPLETE

Stacks

Stack details

Drifts

StackSets

Exports

Application Composer

laC generator

Registry

Public extensions

Activated extensions

Publisher

Spotlight

Feedback

CloudFormation

Stacks

RDSMySQLDBInstance

Stacks (1)

Filter by stack name

Filter status

Active

View nested

Stacks

RDSMySQLDBInstance

2024-09-29 20:58:11 UTC+0530

CREATE_COMPLETE

Stack info

Events

Resources

Outputs

Parameters

Template

Change sets

Git sync

Events (8)

Detect root cause

Search events

Timestamp	Logical ID	Status	Detailed status	Status reason
2024-09-29 21:08:33 UTC+0530	RDSMySQLDBInstance	CREATE_COMPLETE	-	-
2024-09-29 21:08:32 UTC+0530	MyDBInstance	CREATE_COMPLETE	-	-
2024-09-29 20:58:19 UTC+0530	MyDBInstance	CREATE_IN_PROGRESS	-	Resource creation Initiated
2024-09-29 20:58:17 UTC+0530	MyDBInstance	CREATE_IN_PROGRESS	-	-
2024-09-29 20:58:16 UTC+0530	MyDBSecurityGroup	CREATE_COMPLETE	-	-
2024-09-29 20:58:16 UTC+0530	MyDBSecurityGroup	CREATE_IN_PROGRESS	-	Resource creation Initiated
2024-09-29 20:58:13 UTC+0530	MyDBSecurityGroup	CREATE_IN_PROGRESS	-	-
2024-09-29 20:58:11 UTC+0530	RDSMySQLDBInstance	CREATE_IN_PROGRESS	-	User Initiated

Thank You!