Set Up a WordPress Instance for Your Organization

DESCRIPTION

You are hired as a cloud architect in a global media company. You have been asked to set up a WordPress instance to publish blogs for your company per the defined specifications.

Background of the problem statement:

Your organization publishes blogs and provides documentation services for other businesses and technologies. You have been asked to:

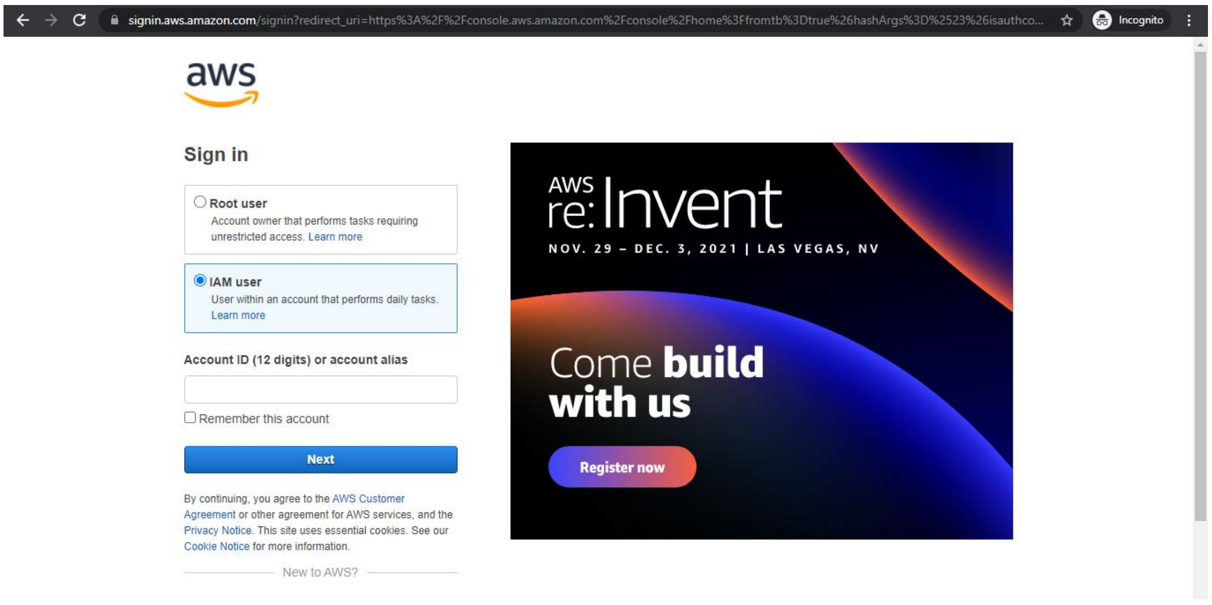
- Set up a live WordPress instance to publish blogs
- Set up a WordPress instance that can be used for development and testing purposes so that any work done on this instance will not impact the live blog
- Configure the WordPress instance for development and testing purposes, which will be available only for the business hours (9 AM–6 PM)
- Give access of the WordPress instance to the blogging team for development and testing purposes

You must use the following:

- Amazon CloudFormation
- AWS Auto Scaling
- Amazon Machine Images
- Amazon Identity and Access Management

The following requirements should be met:

- Ensure that the live WordPress instance has public read access from anywhere over the internet
- Ensure that you block the public access to the WordPress instance for development and testing purposes
- Ensure that only the blogging team has access to the WordPress instance for development and testing purposes
- Ensure that the WordPress instance for development and testing purposes gets shut down after 6 PM every day



signin.aws.amazon.com/signin?redirect_uri=https%3A%2F%2Fconsole.aws.amazon.com%2Fconsole%2Fhome%3Ffromtb%3Dtrue%26hashArgs%3D%2523%26isauthco...

Sign in

( ) Root user
Account owner that performs tasks requiring unrestricted access. Learn more

(●) IAM user
User within an account that performs daily tasks. Learn more

Account ID (12 digits) or account alias

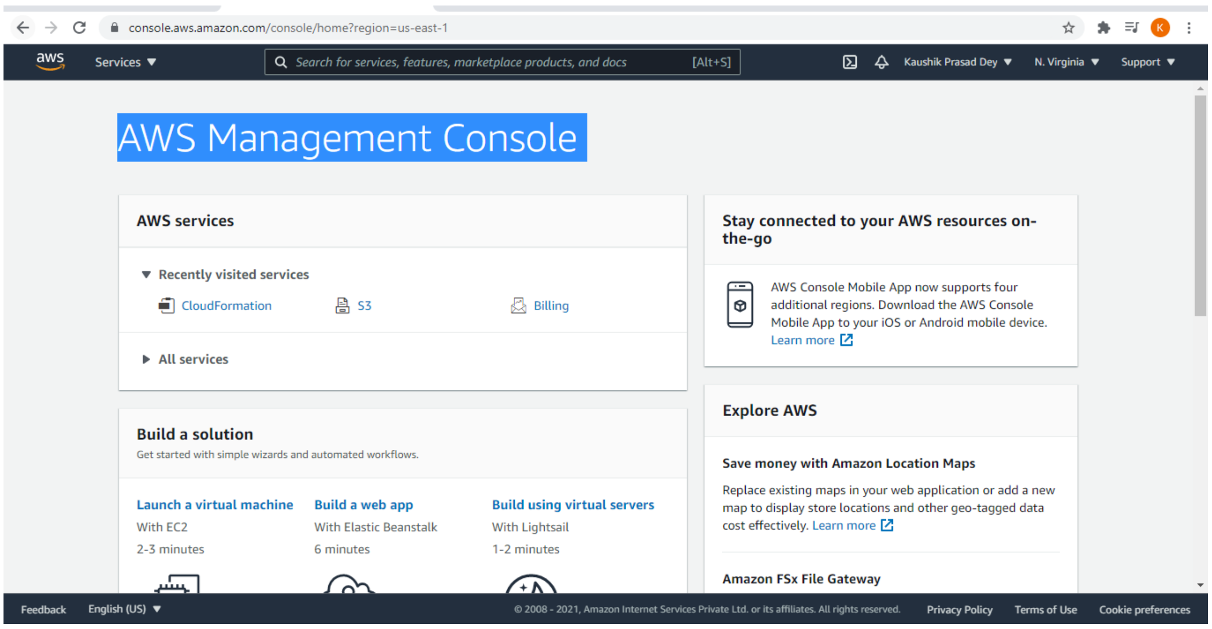
☐ Remember this account

Next

By continuing, you agree to the AWS Customer Agreement or other agreement for AWS services, and the Privacy Notice. This site uses essential cookies. See our Cookie Notice for more information.

New to AWS?

AWS re:Invent
NOV. 29 – DEC. 3, 2021 | LAS VEGAS, NV
Come build with us
Register now



console.aws.amazon.com/console/home?region=us-east-1

Services
Search for services, features, marketplace products, and docs [Alt+S]
Kaushik Prasad Dey
N. Virginia
Support

AWS Management Console

AWS services

Recently visited services
CloudFormation S3 Billing

All services

Build a solution
Get started with simple wizards and automated workflows.

Launch a virtual machine	Build a web app	Build using virtual servers
With EC2	With Elastic Beanstalk	With Lightsail
2-3 minutes	6 minutes	1-2 minutes

Stay connected to your AWS resources on-the-go

AWS Console Mobile App now supports four additional regions. Download the AWS Console Mobile App to your iOS or Android mobile device. Learn more

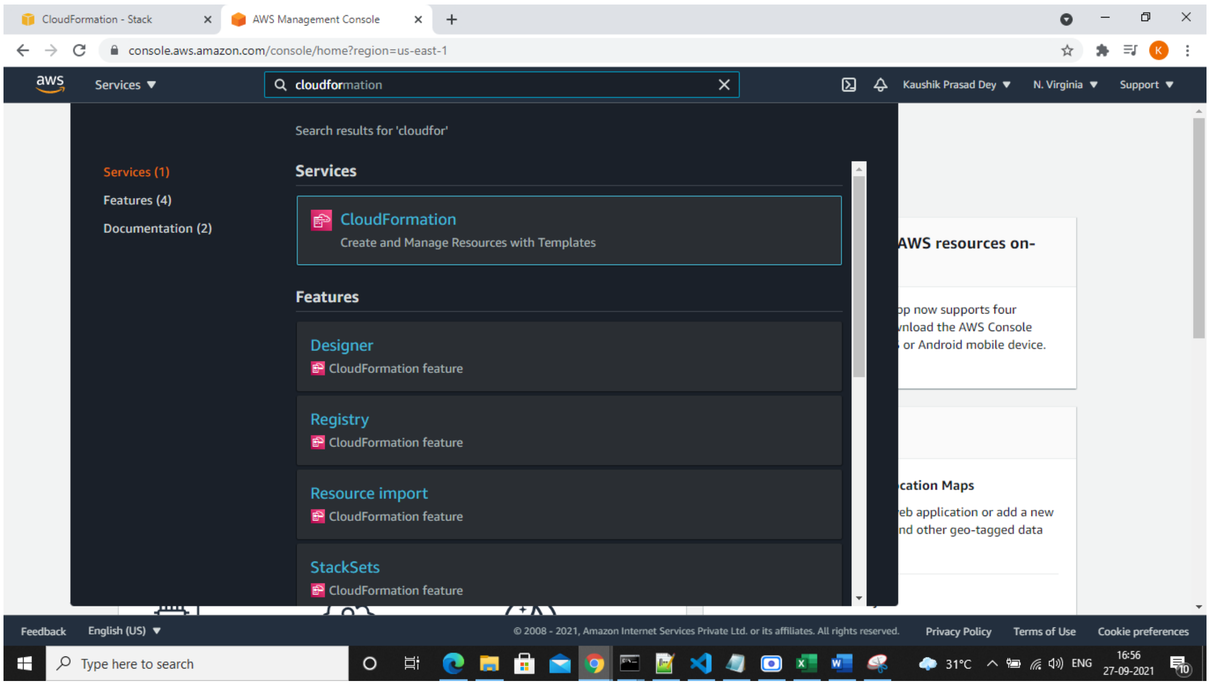
Explore AWS

Save money with Amazon Location Maps
Replace existing maps in your web application or add a new map to display store locations and other geo-tagged data cost effectively. Learn more

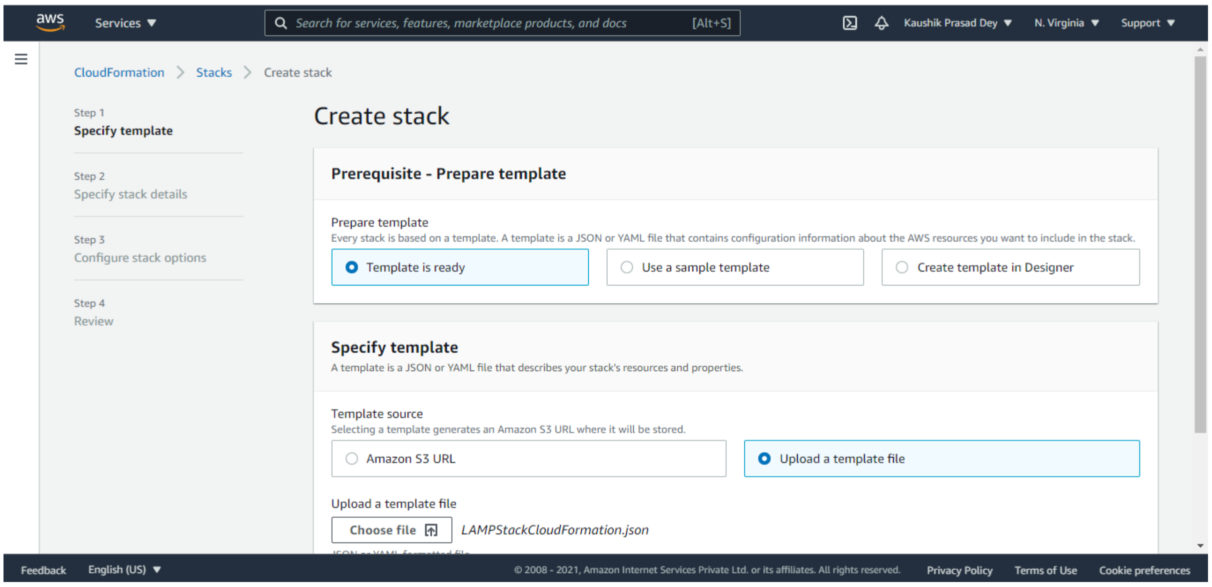
Amazon FSx File Gateway

Feedback English (US) © 2008 - 2021, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved. Privacy Policy Terms of Use Cookie preferences

Set Up a WordPress Instance for Your Organization



Description | Parameters | Mappings | Conditions | Resources | Outputs


Services
Search for services, features, marketplace products, and docs [Alt+S]
Kaushik Prasad Dey
N. Virginia
Support
CloudFormation > Stacks > Create stack
Step 1
Specify template
Step 2
Specify stack details
Step 3
Configure stack options
Step 4
Review
Feedback
English (US)
© 2008 - 2021, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use
Cookie preferences

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.

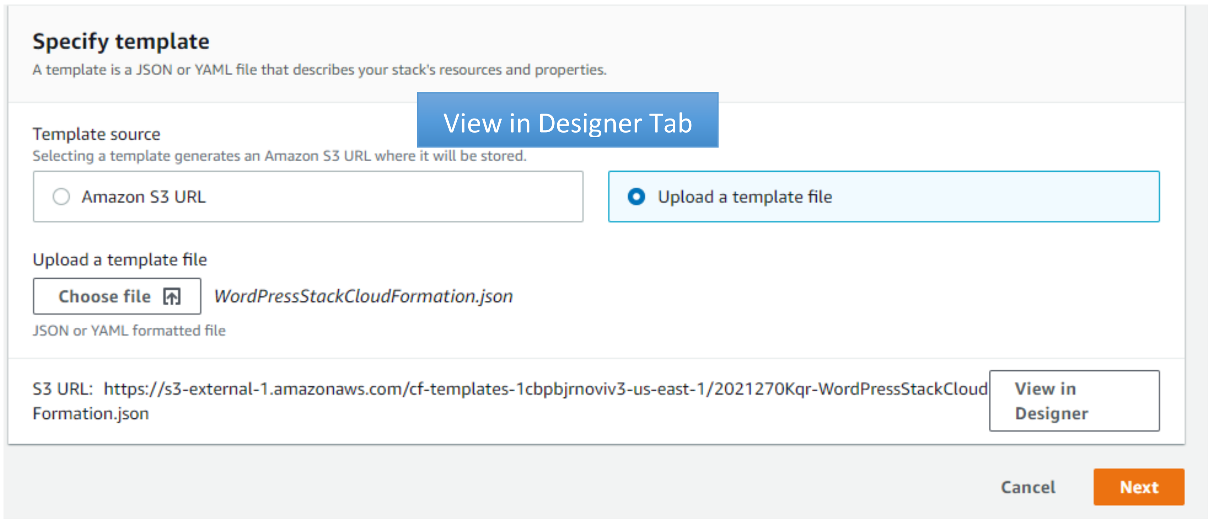
[✓ Template is ready] Use a sample template Create template in Designer

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 *LAMPStackCloudFormation.json*



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

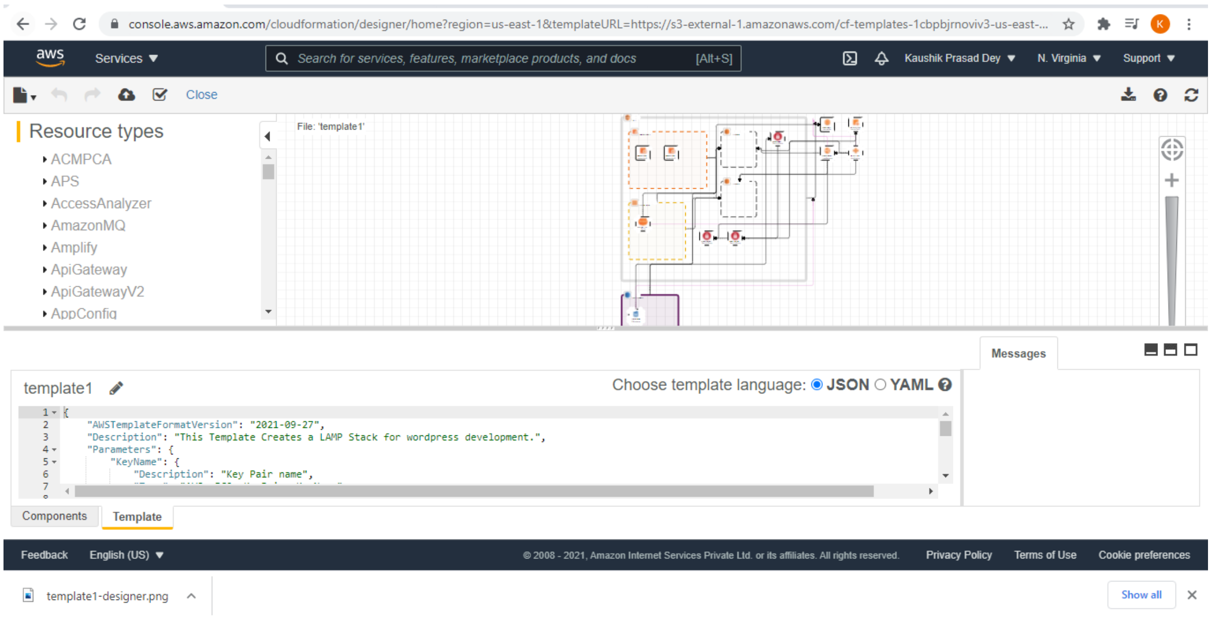
Choose file *WordPressStackCloudFormation.json*

JSON or YAML formatted file

S3 URL: https://s3-external-1.amazonaws.com/cf-templates-1cbpbjrnoviv3-us-east-1/2021270Kqr-WordPressStackCloudFormation.json

```
{
    "AWSTemplateFormatVersion": "2010-09-09",
    "Description": "This Template Creates a LAMP Stack for wordpress developme
nt.",
    "Parameters": {
        "KeyName": {
            "Description": "Key Pair name",
            "Type": "AWS::EC2::KeyPair::KeyName",
            "Default": "mykey"
        }
    },
    "Mappings": {
        "EC2RegionMap": {
            "ap-northeast-1": {"AmazonLinuxAMIHVMEBSBacked64bit": "ami-
cbf90ecb"},
            "ap-south-1": {"AmazonLinuxAMIHVMEBSBacked64bit": "ami-f9daac96"},
            "ap-south": {"AmazonLinuxAMIHVMEBSBacked64bit": "ami-f9daac96"},
            "eu-central-1": {"AmazonLinuxAMIHVMEBSBacked64bit": "ami-
a8221fb5"},
            "eu-west-1": {"AmazonLinuxAMIHVMEBSBacked64bit": "ami-a10897d6"},
            "sa-east-1": {"AmazonLinuxAMIHVMEBSBacked64bit": "ami-b52890a8"},
            "us-east-1": {"AmazonLinuxAMIHVMEBSBacked64bit": "ami-1ecae776"},
            "us-west-1": {"AmazonLinuxAMIHVMEBSBacked64bit": "ami-d114f295"},
            "us-west-2": {"AmazonLinuxAMIHVMEBSBacked64bit": "ami-e7527ed7"}
        }
    },
    "Resources": {
        "VPC": {
            "Type": "AWS::EC2::VPC",
            "Properties": {
                "CidrBlock": "172.31.0.0/16",
                "EnableDnsHostnames": "true"
            }
        },
        "InternetGateway": {
            "Type": "AWS::EC2::InternetGateway",
            "Properties": {
            }
        },
        "VPCGatewayAttachment": {
            "Type": "AWS::EC2::VPCGatewayAttachment",
            "Properties": {
                "VpcId": {"Ref": "VPC"},
                "InternetGatewayId": {"Ref": "InternetGateway"}
            }
        },
        "SubnetA": {
            "Type": "AWS::EC2::Subnet",
            "Properties": {
                "AvailabilityZone": {"Fn::Select": ["0", {"Fn::GetAZs": ""}]},
                "CidrBlock": "172.31.38.0/24",
                "VpcId": {"Ref": "VPC"}
            }
        },
        "SubnetB": {
            "Type": "AWS::EC2::Subnet",
            "Properties": {
                "AvailabilityZone": {"Fn::Select": ["1", {"Fn::GetAZs": ""}]},
                "CidrBlock": "172.31.37.0/24",
                "VpcId": {"Ref": "VPC"}
            }
        },
        "RouteTable": {
            "Type": "AWS::EC2::RouteTable",
            "Properties": {
                "VpcId": {"Ref": "VPC"}
            }
        },
        "RouteTableAssociationA": {
            "Type": "AWS::EC2::SubnetRouteTableAssociation",
            "Properties": {
                "SubnetId": {"Ref": "SubnetA"},
                "RouteTableId": {"Ref": "RouteTable"}
            }
        },
        "RouteTableAssociationB": {
            "Type": "AWS::EC2::SubnetRouteTableAssociation",
            "Properties": {
                "SubnetId": {"Ref": "SubnetB"},
                "RouteTableId": {"Ref": "RouteTable"}
            }
        },
        "RoutePublicNATToInternet": {
            "Type": "AWS::EC2::Route",
            "Properties": {
                "RouteTableId": {"Ref": "RouteTable"},
                "DestinationCidrBlock": "0.0.0.0/0",
                "GatewayId": {"Ref": "InternetGateway"}
            },
            "DependsOn": "VPCGatewayAttachment"
        },
        "NetworkAcl": {
            "Type": "AWS::EC2::NetworkAcl",
            "Properties": {
                "VpcId": {"Ref": "VPC"}
            }
        },
        "SubnetNetworkAclAssociationA": {
            "Type": "AWS::EC2::SubnetNetworkAclAssociation",
            "Properties": {
                "SubnetId": {"Ref": "SubnetA"},
                "NetworkAclId": {"Ref": "NetworkAcl"}
            }
        },
        "SubnetNetworkAclAssociationB": {
            "Type": "AWS::EC2::SubnetNetworkAclAssociation",
            "Properties": {
                "SubnetId": {"Ref": "SubnetB"},
                "NetworkAclId": {"Ref": "NetworkAcl"}
            }
        },
        "NetworkAclEntryIngress": {
            "Type": "AWS::EC2::NetworkAclEntry",
            "Properties": {
                "NetworkAclId": {"Ref": "NetworkAcl"},
                "RuleNumber": "100",
                "Protocol": "-1",
                "RuleAction": "allow",
                "Egress": "false",
                "CidrBlock": "0.0.0.0/0"
            }
        },
        "NetworkAclEntryEgress": {
            "Type": "AWS::EC2::NetworkAclEntry",
            "Properties": {
                "NetworkAclId": {"Ref": "NetworkAcl"},
                "RuleNumber": "100",
                "Protocol": "-1",
                "RuleAction": "allow",
                "Egress": "true",
                "CidrBlock": "0.0.0.0/0"
            }
        },
        "LoadBalancer": {
            "Type": "AWS::ElasticLoadBalancing::LoadBalancer",
            "Properties": {
                "Subnets": [{"Ref": "SubnetA"}, {"Ref": "SubnetB"}],
                "LoadBalancerName": "aws-simplearn-elb",
                "Listeners": [{
                    "InstancePort": "80",
                    "InstanceProtocol": "HTTP",
                    "LoadBalancerPort": "80",
                    "Protocol": "HTTP"
                }],
                "HealthCheck": {
                    "HealthyThreshold": "2",
                    "Interval": "5",
                    "Target": "TCP:80",
                    "Timeout": "3",
                    "UnhealthyThreshold": "2"
                },
                "SecurityGroups": [{"Ref": "LoadBalancerSecurityGroup"}],
                "Scheme": "internet-facing"
            },
            "DependsOn": "VPCGatewayAttachment"
        },
        "LoadBalancerSecurityGroup": {
            "Type": "AWS::EC2::SecurityGroup",
            "Properties": {
                "GroupDescription": "aws-simplearn-elb-sg",
                "VpcId": {"Ref": "VPC"},
                "SecurityGroupIngress": [{
                    "CidrIp": "0.0.0.0/0",
                    "FromPort": 80,
                    "IpProtocol": "tcp",
                    "ToPort": 80
                }]
            }
        },
        "WebServerSecurityGroup": {
            "Type": "AWS::EC2::SecurityGroup",
            "Properties": {
                "GroupDescription": "aws-simplearn-sg",
                "VpcId": {"Ref": "VPC"},
                "SecurityGroupIngress": [{
                    "CidrIp": "0.0.0.0/0",
                    "FromPort": 22,
                    "IpProtocol": "tcp",
                    "ToPort": 22
                }, {
                    "FromPort": 80,
                    "IpProtocol": "tcp",
                    "SourceSecurityGroupId": {"Ref": "LoadBalancerSecurityGrou
p"},
                    "ToPort": 80
                }]
            }
        },
        "DatabaseSecurityGroup": {
            "Type": "AWS::EC2::SecurityGroup",
            "Properties": {
                "GroupDescription": "aws-simplearn-db-sg",
                "VpcId": {"Ref": "VPC"},
                "SecurityGroupIngress": [{
                    "IpProtocol": "tcp",
                    "FromPort": "3306",
                    "ToPort": "3306",
                    "SourceSecurityGroupId": {"Ref": "WebServerSecurityGroup"}
                }]
            }
        },
        "Database": {
            "Type": "AWS::RDS::DBInstance",
            "Properties": {
                "AllocatedStorage": "5",
                "BackupRetentionPeriod": "0",
                "DBInstanceClass": "db.t2.micro",
                "DBInstanceIdentifier": "aws-simplearn-db",
                "DBName": "wordpress",
                "Engine": "MySQL",
                "MasterUsername": "wordpress",
                "MasterUserPassword": "wordpress",
                "VPCSecurityGroups": [{"Fn::GetAtt": ["DatabaseSecurityGroup",
 "GroupId"]}],
                "DBSubnetGroupName": {"Ref": "DBSubnetGroup"}
            },
            "DependsOn": "VPCGatewayAttachment"
        },
        "DBSubnetGroup" : {
            "Type" : "AWS::RDS::DBSubnetGroup",
            "Properties" : {
                "DBSubnetGroupDescription" : "DB subnet group",
                "SubnetIds": [{"Ref": "SubnetA"}, {"Ref": "SubnetB"}]
            }
        },
        "LaunchConfiguration": {
            "Type": "AWS::AutoScaling::LaunchConfiguration",
            "Metadata": {
                "AWS::CloudFormation::Init": {
                    "config": {
                        "packages": {
                            "yum": {
                                "php": [],
                                "php-mysql": [],
                                "mysql": [],
                                "httpd": []
                            }
                        },
                        "sources": {
                            "/var/www/html": "https://wordpress.org/wordpress-
4.7.2.tar.gz"
                        },
                        "files": {
                            "/tmp/config": {
                                "content": {"Fn::Join": ["", [
                                    "#!/bin/bash -ex\n",
                                    "cp /var/www/html/wordpress/wp-config-
sample.php /var/www/html/wordpress/wp-config.php\n",
                                    "sed -
i \"s/'database_name_here'/'wordpress'/g\" wp-config.php\n",
                                    "sed -
i \"s/'username_here'/'wordpress'/g\" wp-config.php\n",
                                    "sed -
i \"s/'password_here'/'wordpress'/g\" wp-config.php\n",
                                    "sed -
i \"s/'localhost'/'", {"Fn::GetAtt": ["Database", "Endpoint.Address"]}, "'/g\"
 wp-config.php\n",
                                    "chmod -R 777 wp-content/ \n"
                                ]]},
                                "mode": "000500",
                                "owner": "root",
                                "group": "root"
                            }
                        },
                        "commands": {
                            "01_config": {
                                "command": "/tmp/config",
                                "cwd": "/var/www/html/wordpress"
                            }
                        },
                        "services": {
                            "sysvinit": {
                                "httpd": {
                                    "enabled": "true",
                                    "ensureRunning": "true"
                                }
                            }
                        }
                    }
                }
            },
            "Properties": {
                "EbsOptimized": false,
                "ImageId": {"Fn::FindInMap": ["EC2RegionMap", {"Ref": "AWS::Re
gion"}, "AmazonLinuxAMIHVMEBSBacked64bit"]},
                "InstanceType": "t2.micro",
                "SecurityGroups": [{"Ref": "WebServerSecurityGroup"}],
                "KeyName": {"Ref": "KeyName"},
                "AssociatePublicIpAddress": true,
                "UserData": {"Fn::Base64": {"Fn::Join": ["", [
                    "#!/bin/bash -ex\n",
                    "yum update -y aws-cfn-bootstrap\n",
                    "/opt/aws/bin/cfn-init -v --
stack ", {"Ref": "AWS::StackName"}, " --resource LaunchConfiguration --
region ", {"Ref": "AWS::Region"}, "\n",
                    "/opt/aws/bin/cfn-signal -e $? --
stack ", {"Ref": "AWS::StackName"}, " --resource AutoScalingGroup --
region ", {"Ref": "AWS::Region"}, "\n"
                ]]}}
            }
        },
        "AutoScalingGroup": {
            "Type": "AWS::AutoScaling::AutoScalingGroup",
            "Properties": {
                "LoadBalancerNames": [{"Ref": "LoadBalancer"}],
                "LaunchConfigurationName": {"Ref": "LaunchConfiguration"},
                "MinSize": "2",
                "MaxSize": "2",
                "DesiredCapacity": "2",
                "VPCZoneIdentifier": [{"Ref": "SubnetA"}, {"Ref": "SubnetB"}]
            },
            "CreationPolicy": {
                "ResourceSignal": {
                    "Timeout": "PT10M"
                }
            },
            "DependsOn": "VPCGatewayAttachment"
        }
    },
    "Outputs": {
        "URL": {
            "Value": {"Fn::Join": ["", ["http://", {"Fn::GetAtt": ["LoadBalanc
er", "DNSName"]}, "/wordpress"]]},
            "Description": "Wordpress URL"
        }
    }
}
```

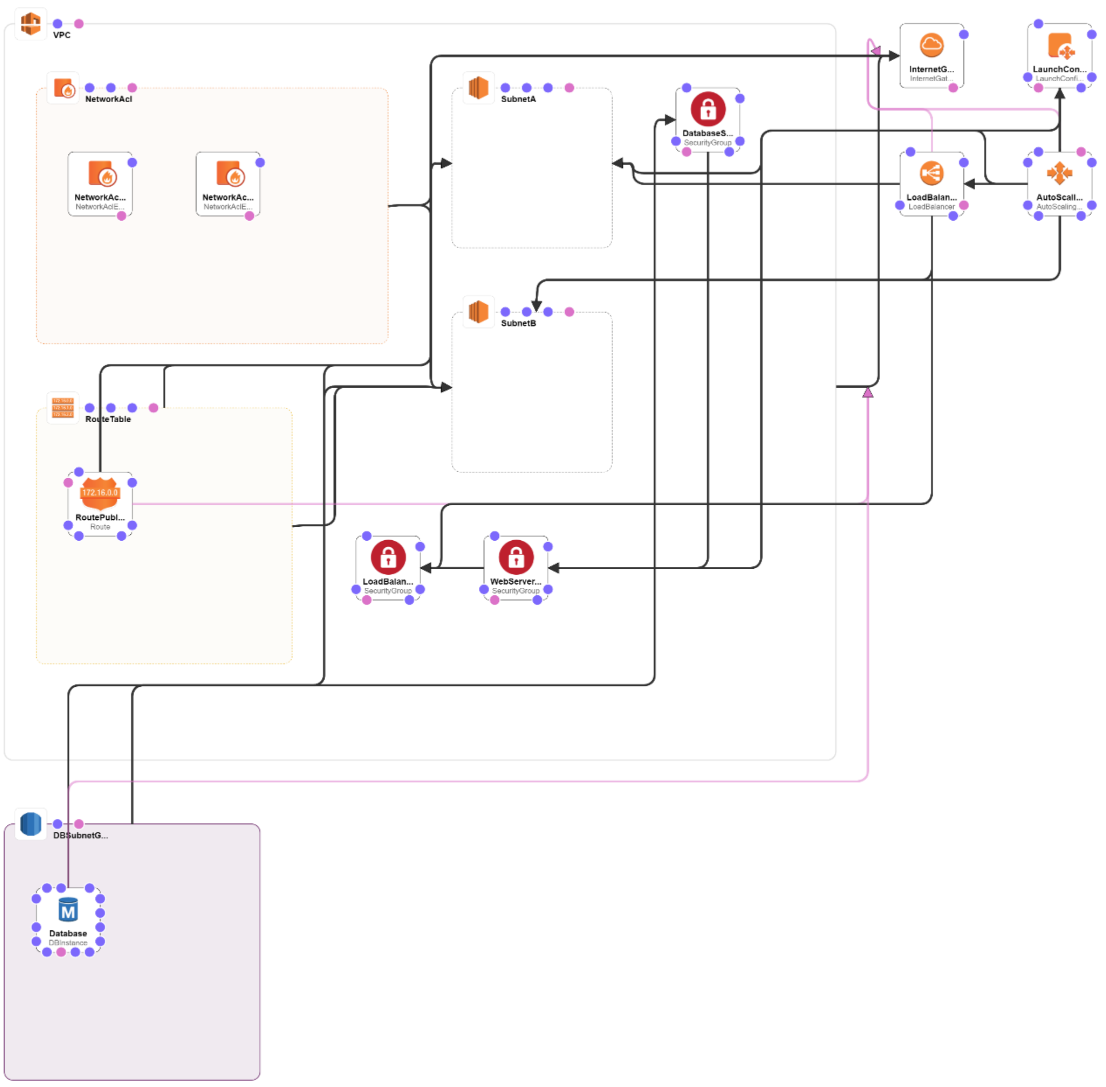
Set Up a WordPress Instance for Your Organization

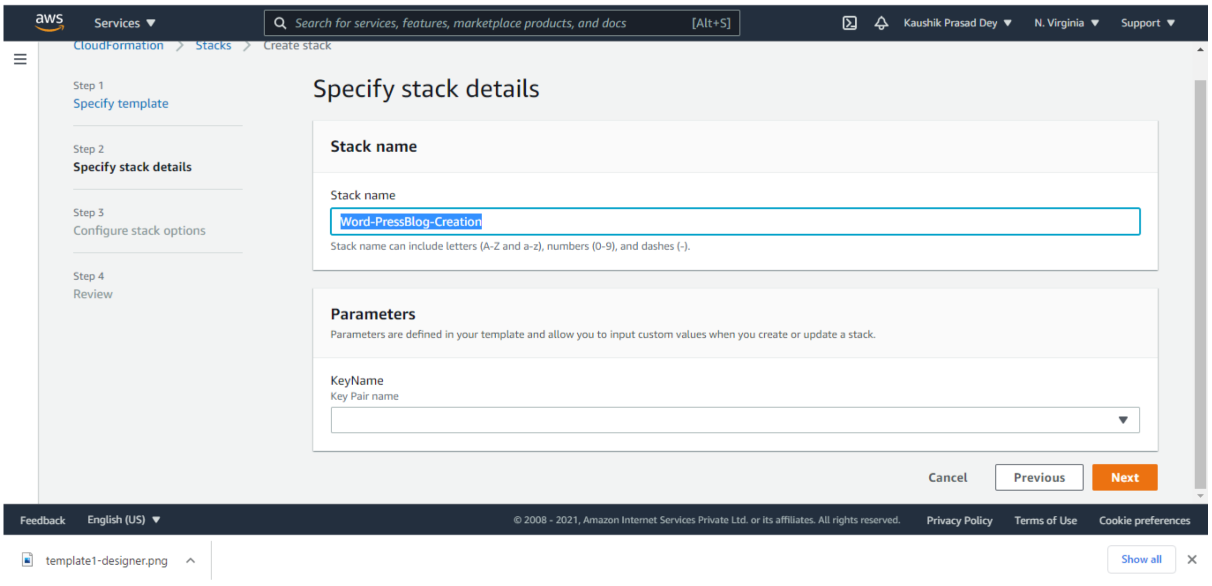

console.aws.amazon.com/cloudformation/designer/home?region=us-east-1&templateURL=https://s3-external-1.amazonaws.com/cf-templates-1cbpbjrnoviv3-us-east-...
aws
Services ▼
Search for services, features, marketplace products, and docs [Alt+S]
Kaushik Prasad Dey ▼
N. Virginia ▼
Support ▼
Close
Resource types
ACMPCA
APS
AccessAnalyzer
AmazonMQ
Amplify
ApiGateway
ApiGatewayV2
AppConfig
File: 'template1'
Messages
Components
Template
Feedback
English (US) ▼
© 2008 - 2021, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use
Cookie preferences
template1-designer.png
Show all

template1

Choose template language: [✓ JSON] YAML

```
{
    "AWSTemplateFormatVersion": "2021-09-27",
    "Description": "This Template Creates a LAMP Stack for wordpress development.",
    "Parameters": {
        "KeyName": {
            "Description": "Key Pair name",
```





Specify stack details

Stack name

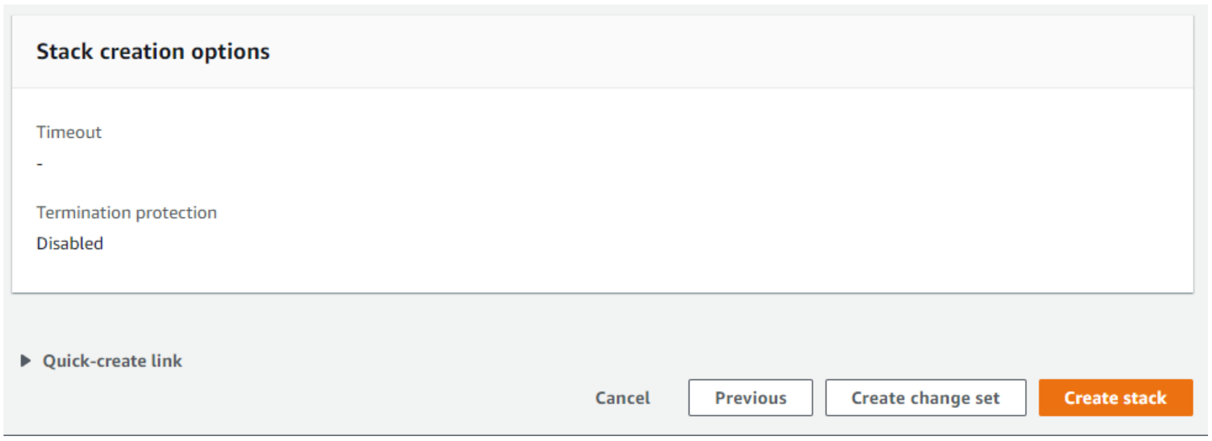
Stack name: Word-PressBlog-Creation

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.

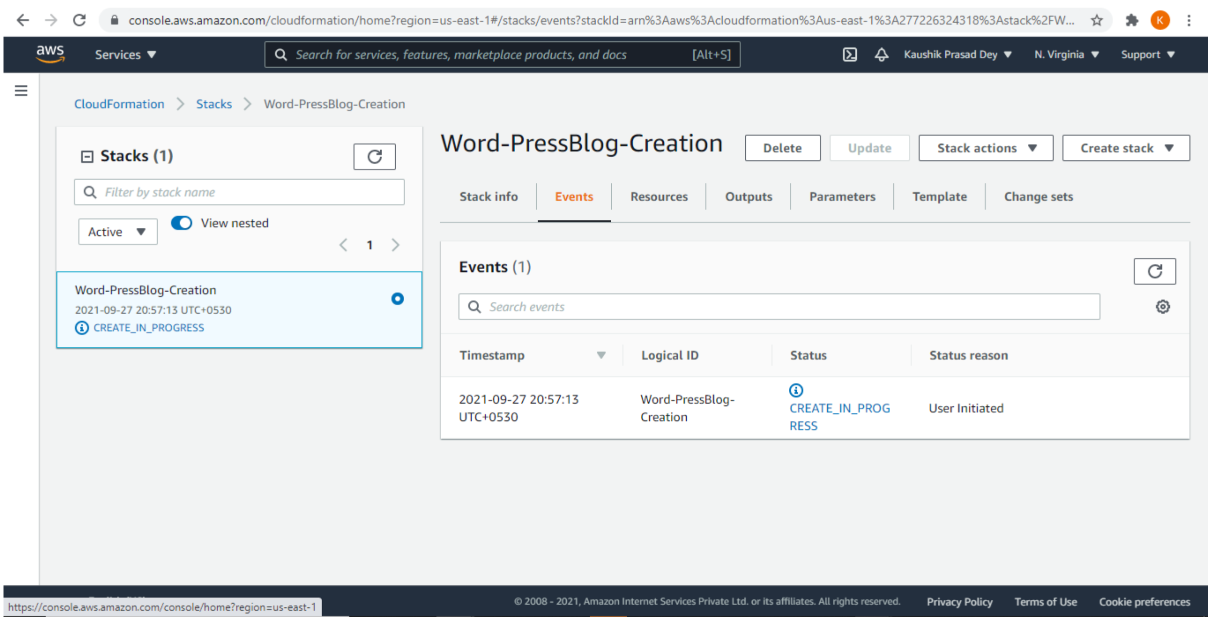
KeyName
Key Pair name



Stack creation options

Timeout
-

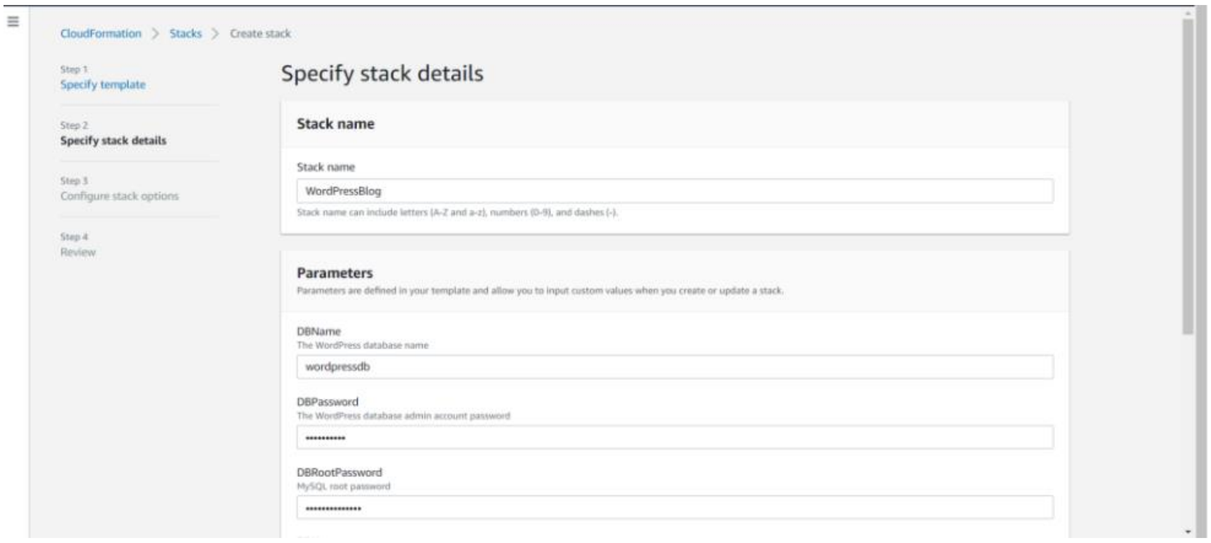
Termination protection
Disabled



Word-PressBlog-Creation

Events (1)

Timestamp	Logical ID	Status	Status reason
2021-09-27 20:57:13 UTC+0530	Word-PressBlog-Creation	CREATE_IN_PROGRESS	User Initiated



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

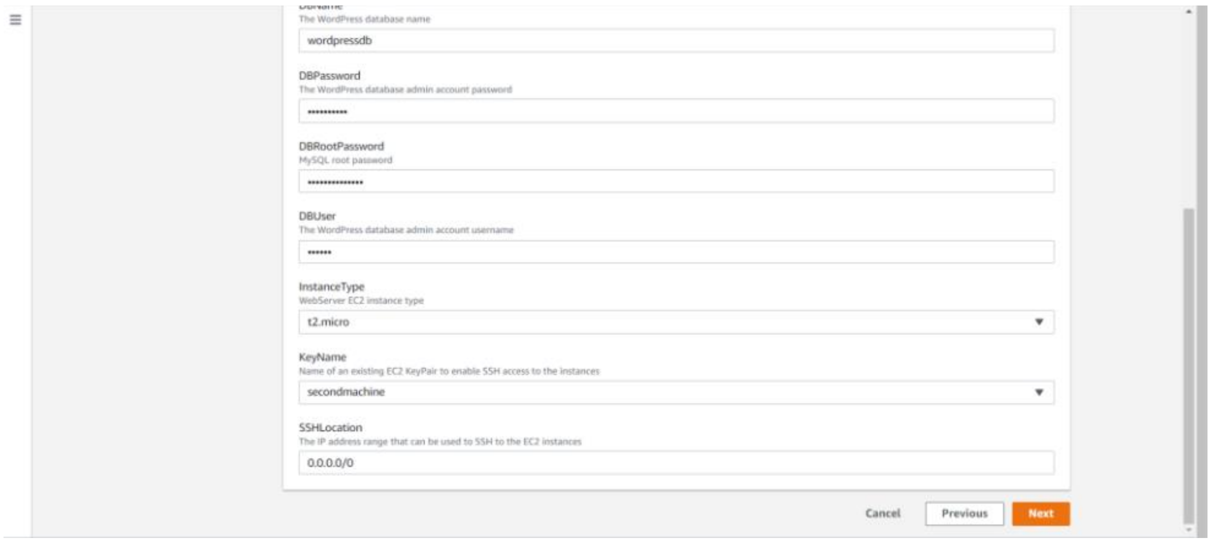
Stack name
WordPressBlog
Stack name can include letters (A-Z and a-z), numbers (0-9), and dashes (-).

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

DBName
The WordPress database name
wordpressdb

DBPassword
The WordPress database admin account password
••••••••••

DBRootPassword
MySQL root password
••••••••••••••



The WordPress database name
wordpressdb

DBPassword
The WordPress database admin account password
••••••••••

DBRootPassword
MySQL root password
••••••••••••••

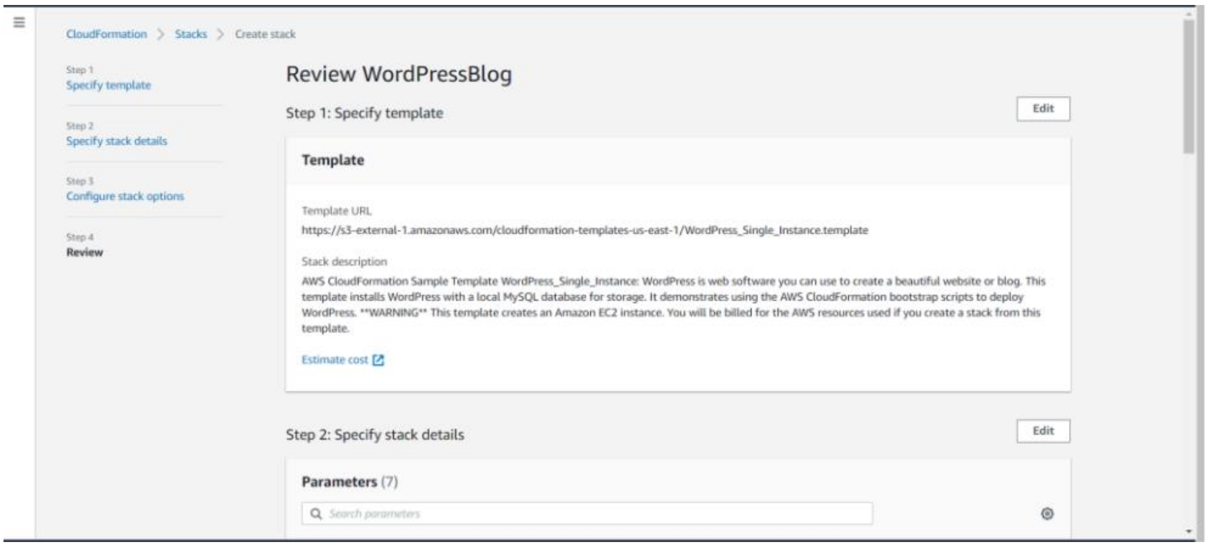
DBUser
The WordPress database admin account username
••••••

InstanceType
WebServer EC2 instance type
t2.micro

KeyName
Name of an existing EC2 KeyPair to enable SSH access to the instances
secondmachine

SSHLocation
The IP address range that can be used to SSH to the EC2 instances
0.0.0.0/0

Cancel | Previous | Next



CloudFormation > Stacks > Create stack

Step 1
Specify template

Step 2
Specify stack details

Step 3
Configure stack options

Step 4
Review

Review WordPressBlog

Step 1: Specify template — Edit

Template

Template URL
https://s3-external-1.amazonaws.com/cloudformation-templates-us-east-1/WordPress_Single_Instance.template

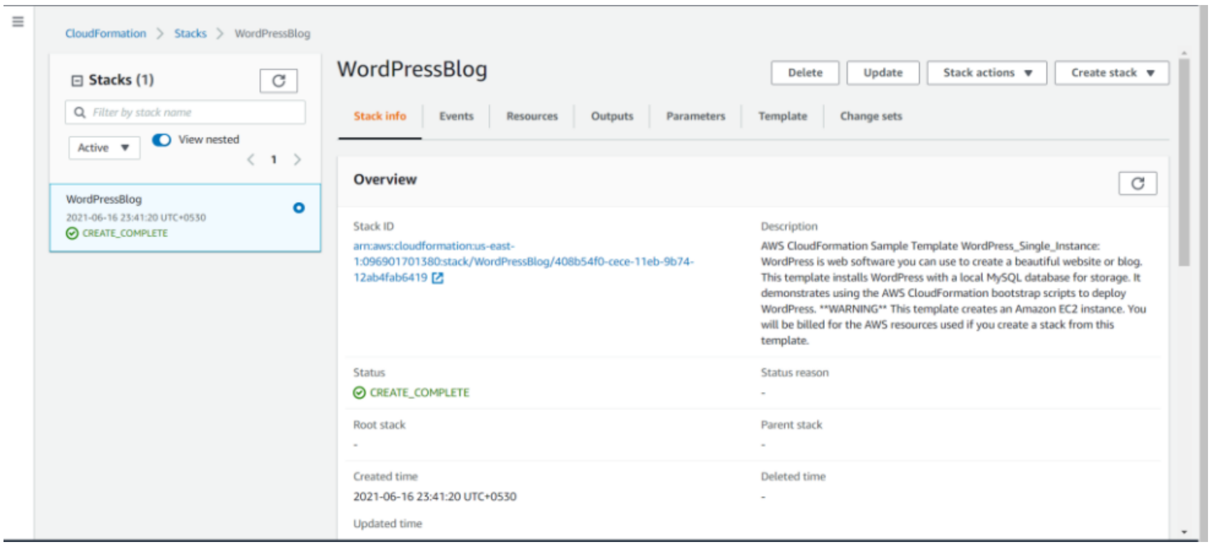
Stack description
AWS CloudFormation Sample Template WordPress_Single_Instance: WordPress is web software you can use to create a beautiful website or blog. This template installs WordPress with a local MySQL database for storage. It demonstrates using the AWS CloudFormation bootstrap scripts to deploy WordPress. **WARNING** This template creates an Amazon EC2 instance. You will be billed for the AWS resources used if you create a stack from this template.

Estimate cost

Step 2: Specify stack details — Edit

Parameters (7)

Search parameters



WordPressBlog
2021-06-16 23:41:20 UTC+0530
CREATE_COMPLETE

WordPressBlog

Overview

Stack ID
arn:aws:cloudformation:us-east-1:096901701380:stack/WordPressBlog/408b54f0-cece-11eb-9b74-12ab4fab6419

Description
AWS CloudFormation Sample Template WordPress_Single_Instance: WordPress is web software you can use to create a beautiful website or blog. This template installs WordPress with a local MySQL database for storage. It demonstrates using the AWS CloudFormation bootstrap scripts to deploy WordPress. **WARNING** This template creates an Amazon EC2 instance. You will be billed for the AWS resources used if you create a stack from this template.

Status
CREATE_COMPLETE

Status reason
-

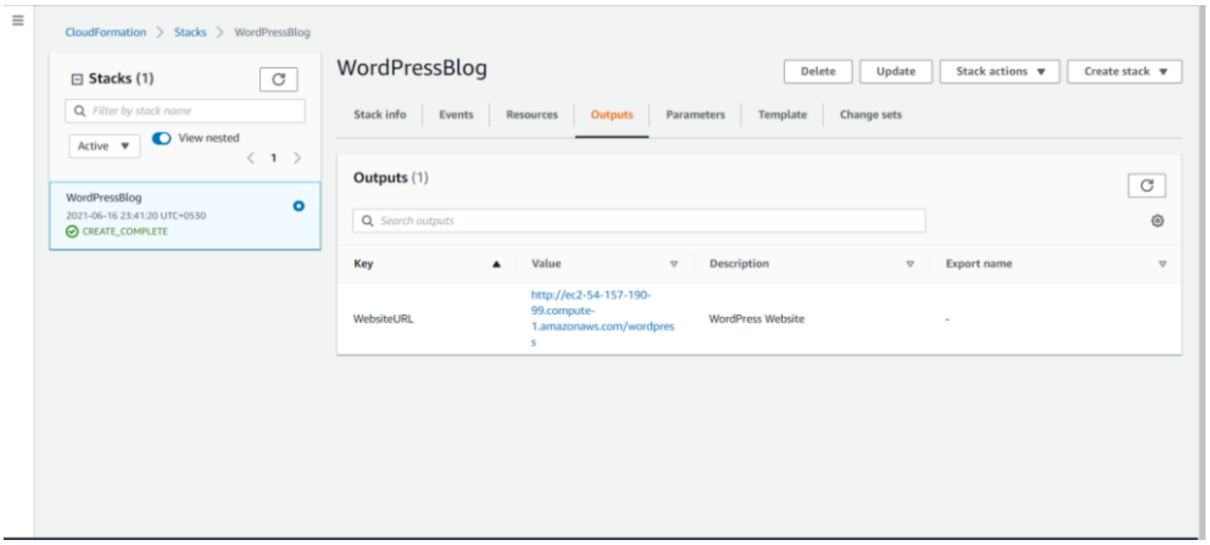
Root stack
-

Parent stack
-

Created time
2021-06-16 23:41:20 UTC+0530

Deleted time
-

Updated time

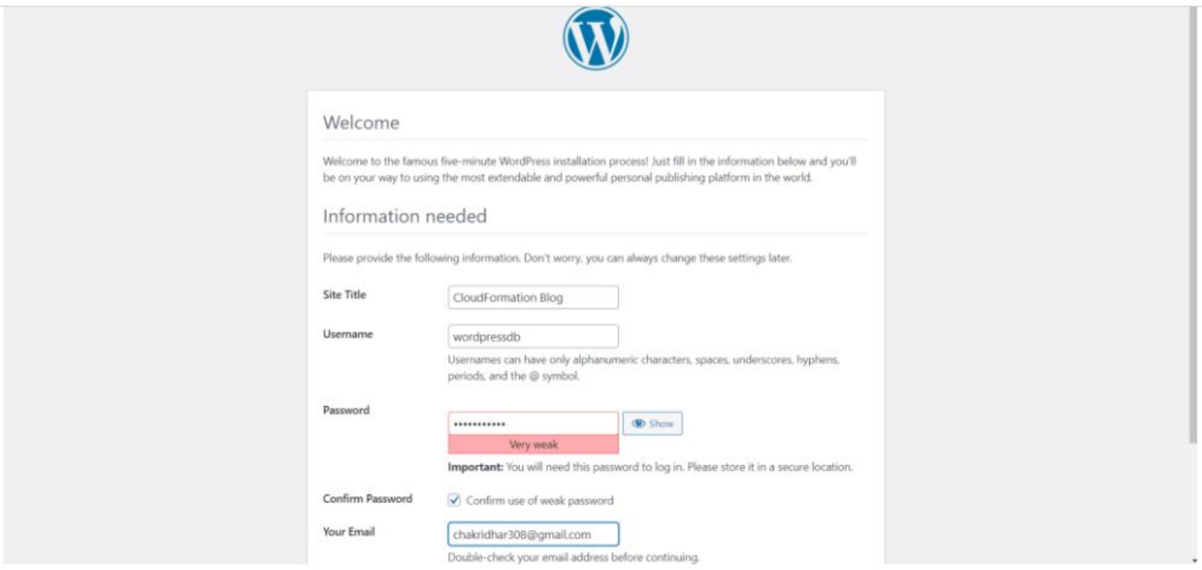


WordPressBlog
2021-06-16 23:41:20 UTC+0530
CREATE_COMPLETE

WordPressBlog

Outputs (1)

Key	Value	Description	Export name
WebsiteURL	http://ec2-54-157-190-99.compute-1.amazonaws.com/wordpress	WordPress Website	-



Welcome

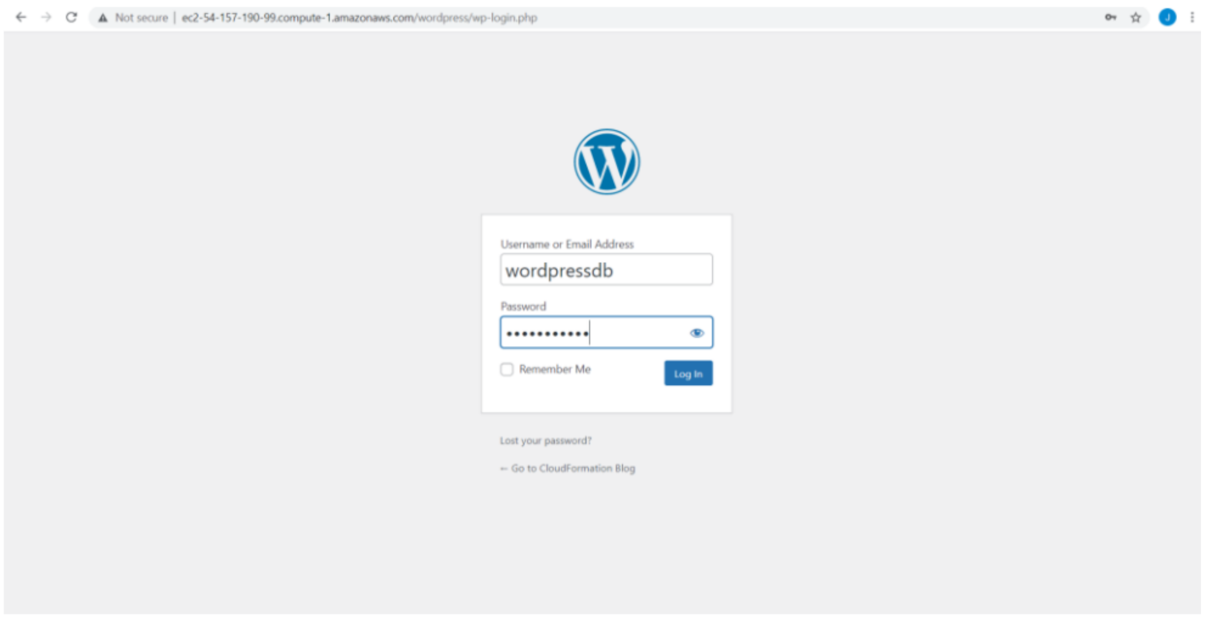
Welcome to the famous five-minute WordPress installation process! Just fill in the information below and you'll be on your way to using the most extendable and powerful personal publishing platform in the world.

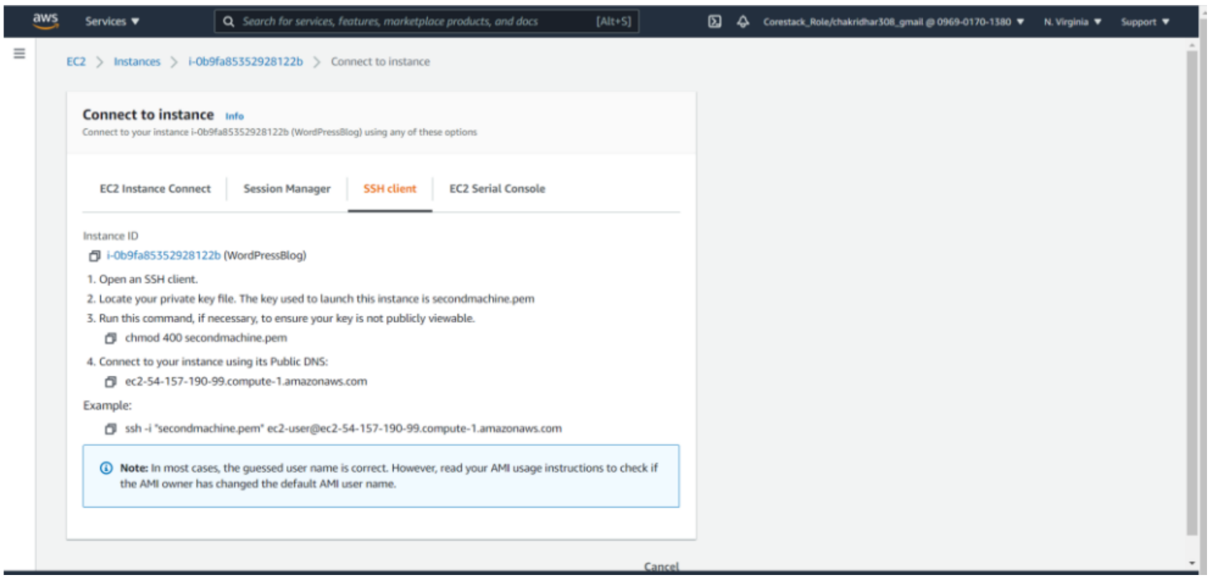
Information needed

Please provide the following information. Don't worry, you can always change these settings later.

Site Title	CloudFormation Blog
Username	wordpressdb Usernames can have only alphanumeric characters, spaces, underscores, hyphens, periods, and the @ symbol.
Password	•••••••••••• Show Very weak **Important:** You will need this password to log in. Please store it in a secure location.
Confirm Password	✓ Confirm use of weak password
Your Email	chakridhar308@gmail.com Double-check your email address before continuing.

Installing WordPress


Not secure | ec2-54-157-190-99.compute-1.amazonaws.com/wordpress/wp-login.php
Username or Email Address
wordpressdb
Password
Remember Me
Log In
Lost your password?
← Go to CloudFormation Blog


Services
Search for services, features, marketplace products, and docs [Alt+S]
Corestack_Role/chakridhar308_gmail @ 0969-0170-1380
N. Virginia
Support
EC2 > Instances > i-0b9fa85352928122b > Connect to instance
EC2 Instance Connect
Session Manager
SSH client
EC2 Serial Console
Cancel

Connect to instance Info

Connect to your instance i-0b9fa85352928122b (WordPressBlog) using any of these options

Instance ID

i-0b9fa85352928122b (WordPressBlog)

1. Open an SSH client.
2. Locate your private key file. The key used to launch this instance is secondmachine.pem
3. Run this command, if necessary, to ensure your key is not publicly viewable.

```
chmod 400 secondmachine.pem
```

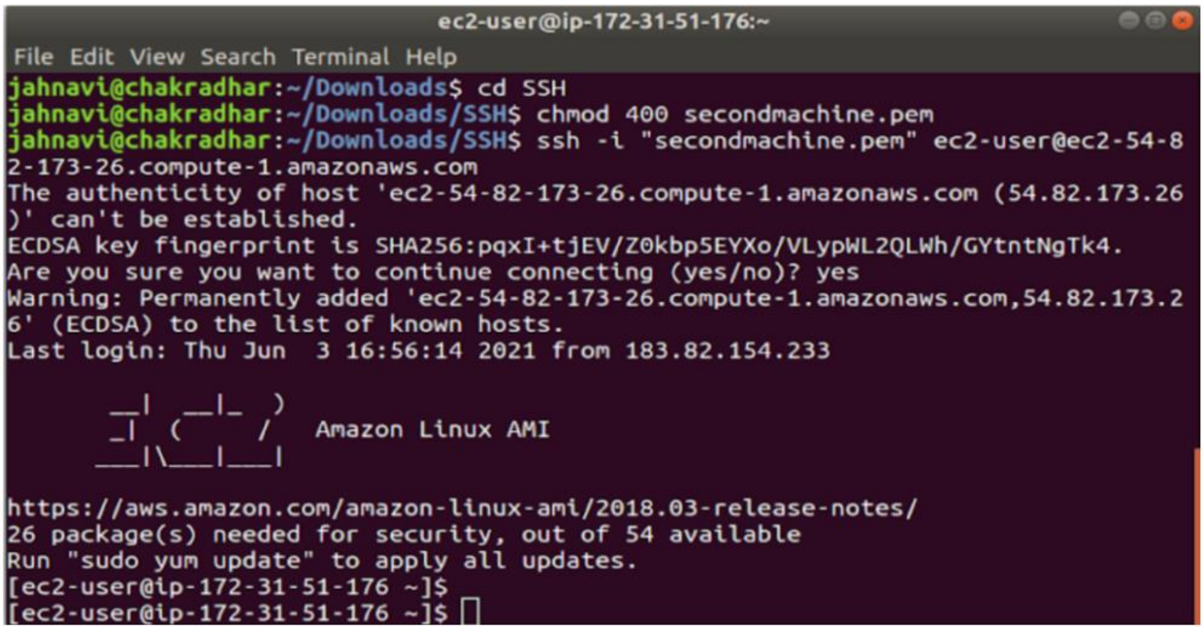
4. Connect to your instance using its Public DNS:

```
ec2-54-157-190-99.compute-1.amazonaws.com
```

Example:

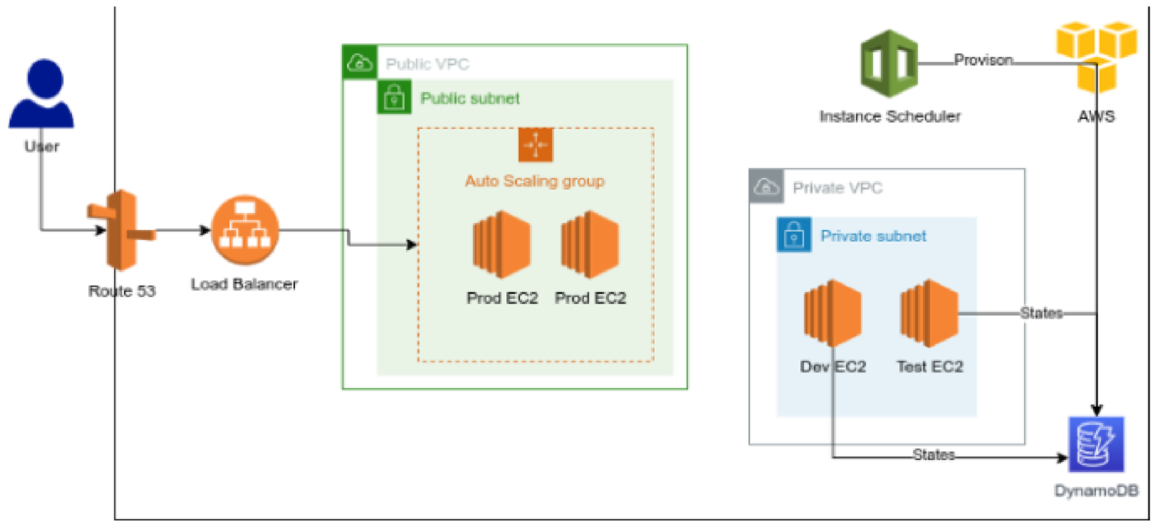
```
ssh -i "secondmachine.pem" ec2-user@ec2-54-157-190-99.compute-1.amazonaws.com
```

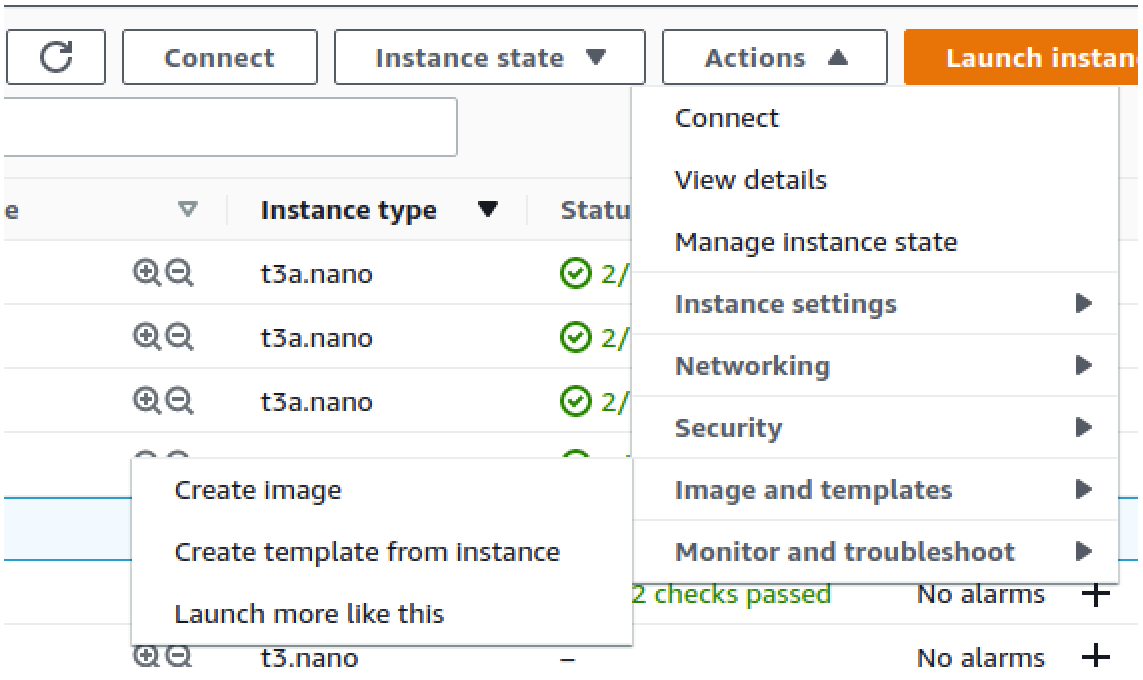
Note: In most cases, the guessed user name is correct. However, read your AMI usage instructions to check if the AMI owner has changed the default AMI user name.

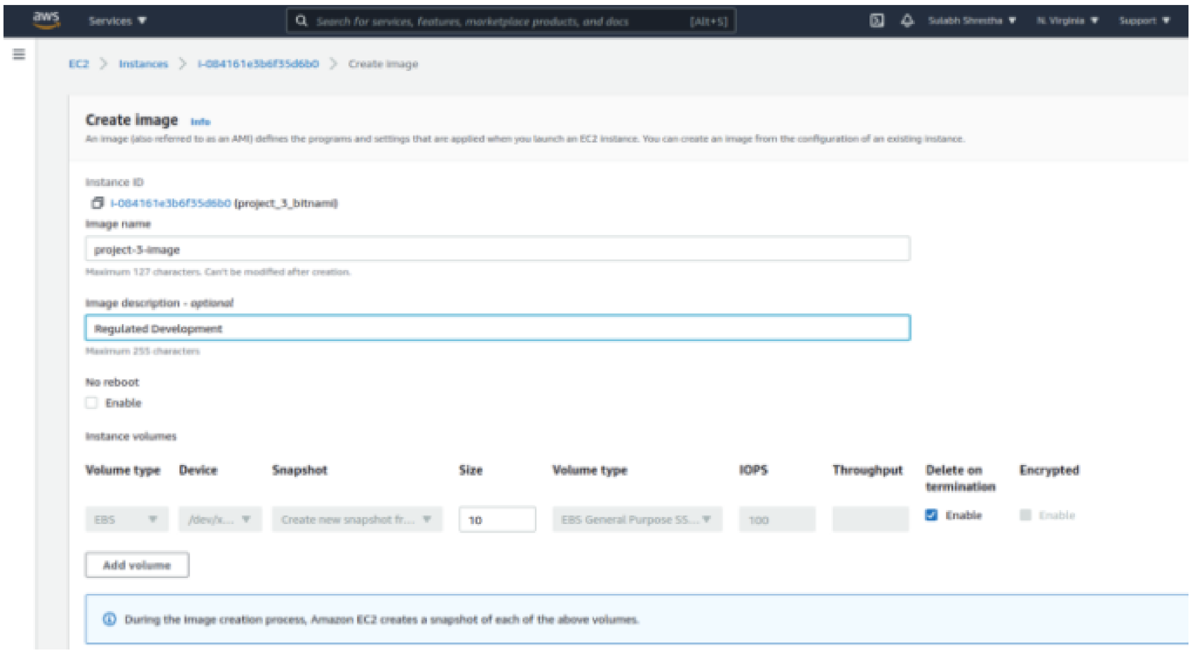


```
ec2-user@ip-172-31-51-176:~
File  Edit  View  Search  Terminal  Help
jahnavi@chakradhar:~/Downloads$ cd SSH
jahnavi@chakradhar:~/Downloads/SSH$ chmod 400 secondmachine.pem
jahnavi@chakradhar:~/Downloads/SSH$ ssh -i "secondmachine.pem" ec2-user@ec2-54-8
2-173-26.compute-1.amazonaws.com
The authenticity of host 'ec2-54-82-173-26.compute-1.amazonaws.com (54.82.173.26
)' can't be established.
ECDSA key fingerprint is SHA256:pqxI+tjEV/Z0kbp5EYXo/VLypWL2QLWh/GYtntNgTk4.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'ec2-54-82-173-26.compute-1.amazonaws.com,54.82.173.2
6' (ECDSA) to the list of known hosts.
Last login: Thu Jun  3 16:56:14 2021 from 183.82.154.233

       __|  __|_  )
       _|  (     /   Amazon Linux AMI
      ___|\___|___|

https://aws.amazon.com/amazon-linux-ami/2018.03-release-notes/
26 package(s) needed for security, out of 54 available
Run "sudo yum update" to apply all updates.
[ec2-user@ip-172-31-51-176 ~]$
[ec2-user@ip-172-31-51-176 ~]$ 
```

Architecture:






aws
Services
Search for services, features, marketplace products, and docs
[Alt+S]
Sulabh Shrestha
N. Virginia
Support
EC2 > Instances > i-084161e3b6f35d6b0 > Create image

Create image Info

An image (also referred to as an AMI) defines the programs and settings that are applied when you launch an EC2 instance. You can create an image from the configuration of an existing instance.

Instance ID
i-084161e3b6f35d6b0 (project_3_bitnami)

Image name
project-3-image
Maximum 127 characters. Can't be modified after creation.

Image description - *optional*
Regulated Development
Maximum 255 characters

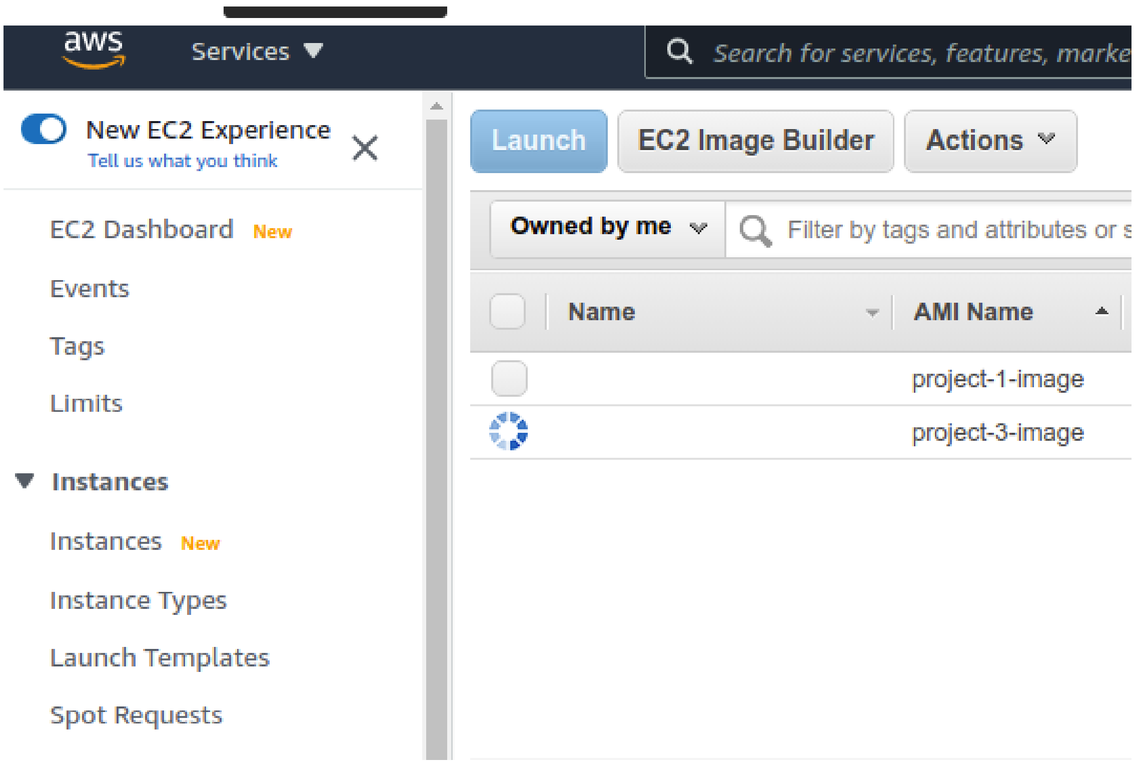
No reboot
☐ Enable

Instance volumes

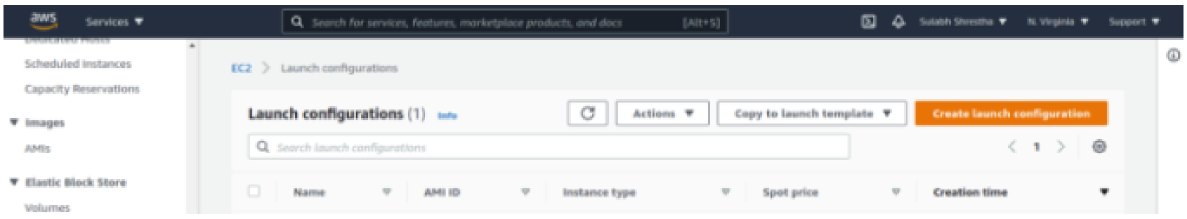
Volume type	Device	Snapshot	Size	Volume type	IOPS	Throughput	Delete on termination	Encrypted
EBS	/dev/x...	Create new snapshot fr...	10	EBS General Purpose SS...	100		✓ Enable	☐ Enable

Add volume

During the image creation process, Amazon EC2 creates a snapshot of each of the above volumes.

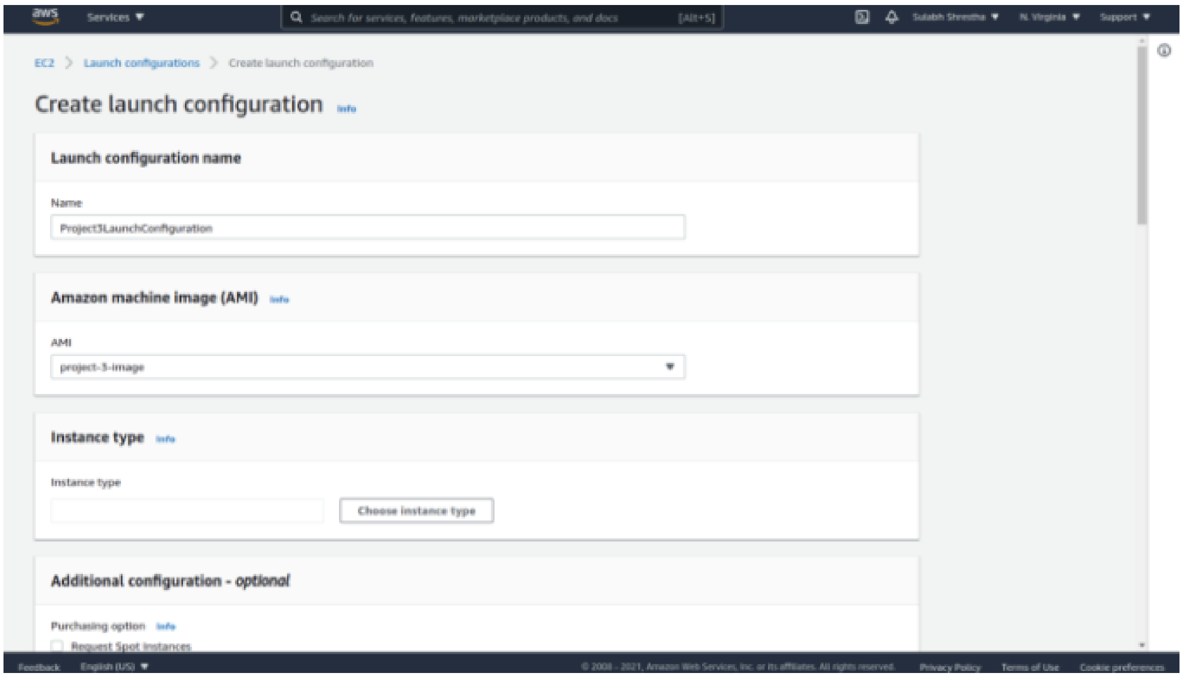


Name	AMI Name
	project-1-image
	project-3-image



Launch configurations (1)

Name	AMI ID	Instance type	Spot price	Creation time


aws
Services
Search for services, features, marketplace products, and docs [Alt+S]
Sulabh Shrestha
N. Virginia
Support
EC2 > Launch configurations > Create launch configuration
Info
Choose instance type
Feedback
English (US)
© 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use
Cookie preferences

Create launch configuration

Launch configuration name

Name: Project3LaunchConfiguration

Amazon machine image (AMI)

AMI: project-3-image

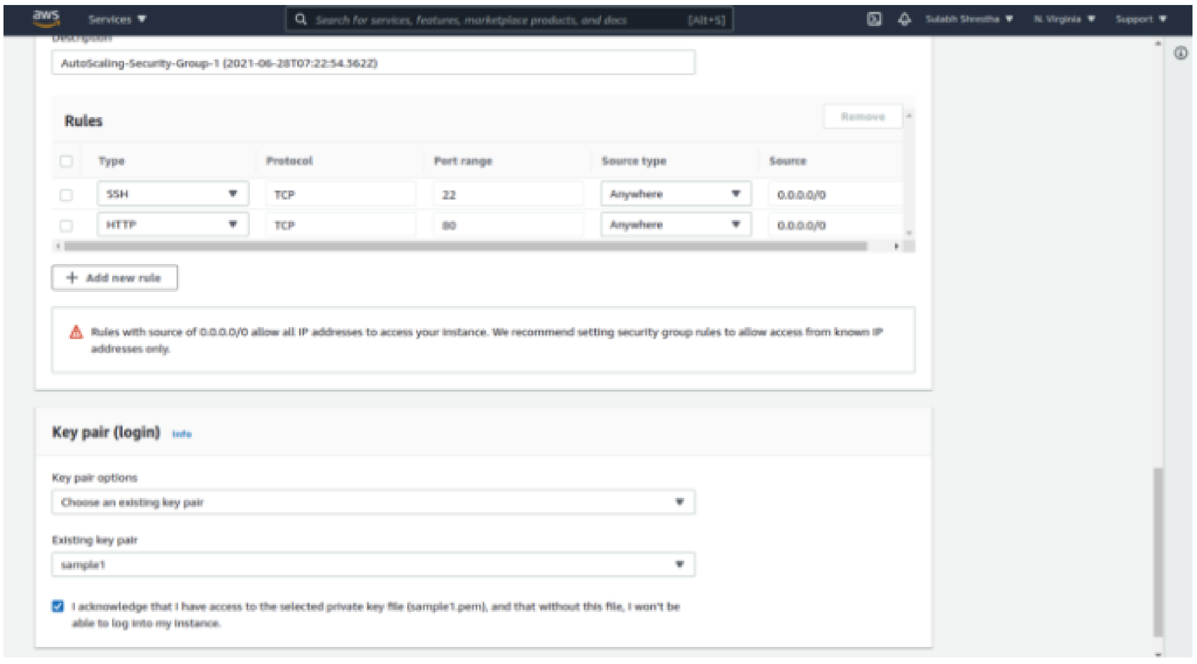
Instance type

Instance type:

Additional configuration - *optional*

Purchasing option

☐ Request Spot Instances


Services
Search for services, features, marketplace products, and docs [Alt+S]
Sulabh Shrestha
N. Virginia
Support
Remove
Add new rule
Info

Description

AutoScaling-Security-Group-1 (2021-06-28T07:22:54.362Z)

Rules

	Type	Protocol	Port range	Source type	Source
☐	SSH	TCP	22	Anywhere	0.0.0.0/0
☐	HTTP	TCP	80	Anywhere	0.0.0.0/0

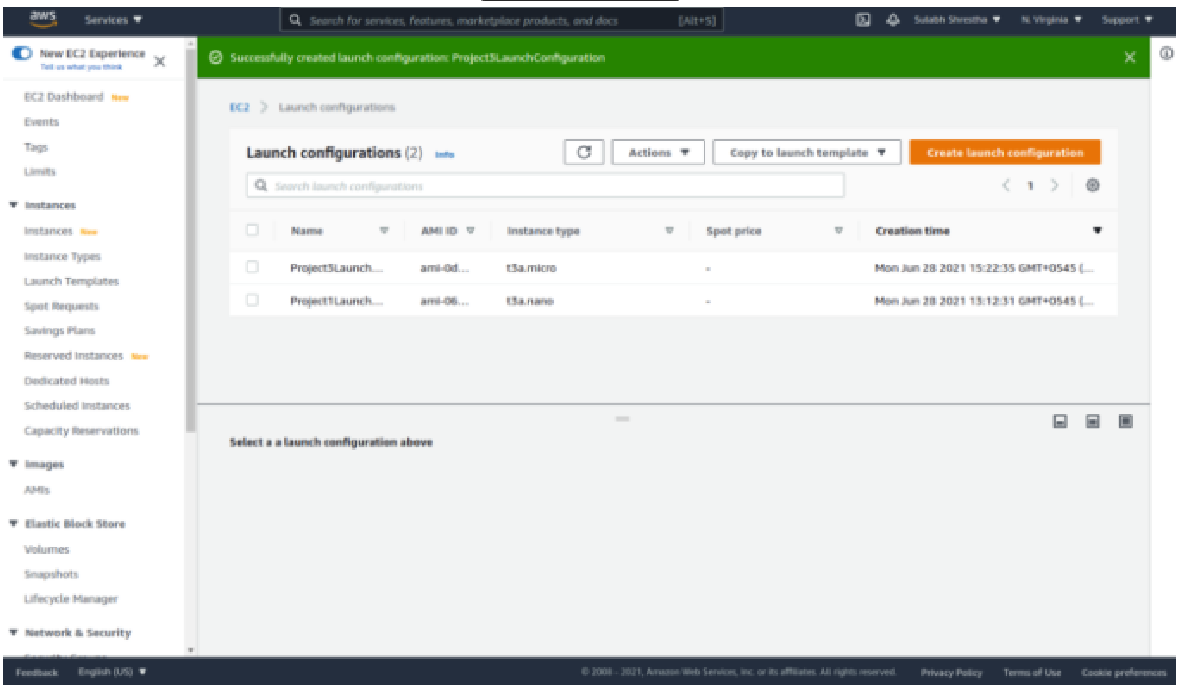
Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.

Key pair (login)

Key pair options: Choose an existing key pair

Existing key pair: sample1

☑ I acknowledge that I have access to the selected private key file (sample1.pem), and that without this file, I won't be able to log into my instance.



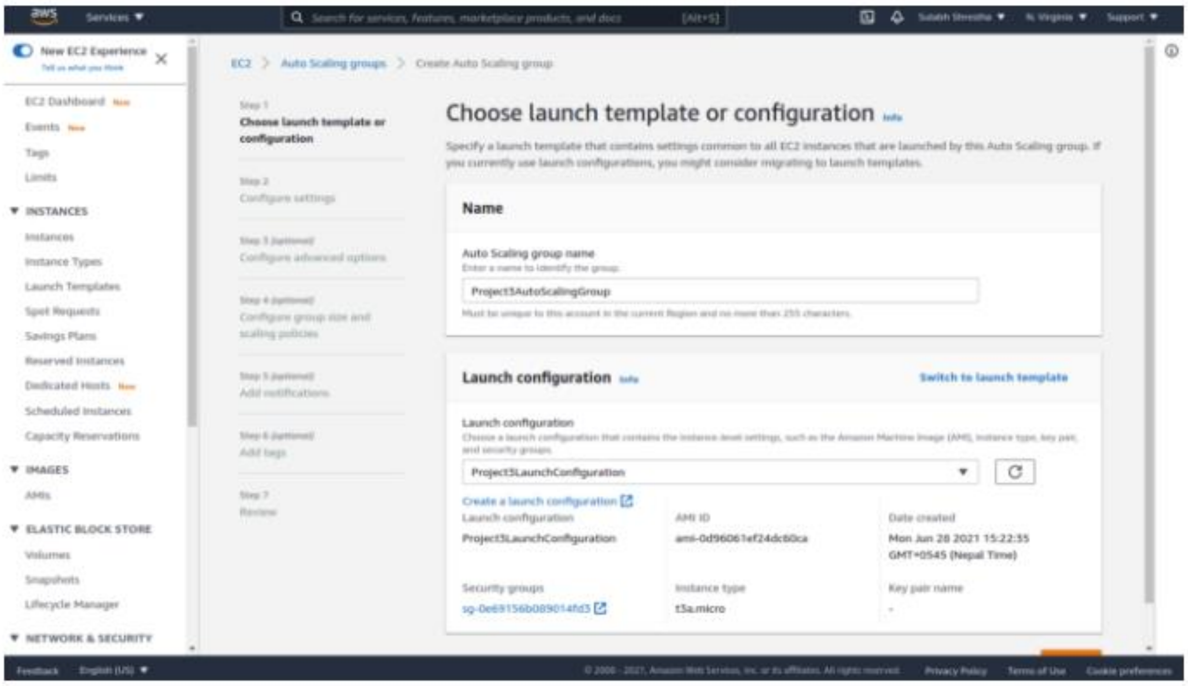
Successfully created launch configuration: Project3LaunchConfiguration

Launch configurations (2)

	Name	AMI ID	Instance type	Spot price	Creation time
☐	Project3Launch...	ami-0d...	t3a.micro	-	Mon Jun 28 2021 15:22:35 GMT+0545 (...
☐	Project1Launch...	ami-06...	t3a.nano	-	Mon Jun 28 2021 13:12:31 GMT+0545 (...

Select a a launch configuration above

© 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.



EC2 > Auto Scaling groups > Create Auto Scaling group

Step 1
Choose launch template or configuration

Step 2
Configure settings

Step 3 (optional)
Configure advanced options

Step 4 (optional)
Configure group size and scaling policies

Step 5 (optional)
Add notifications

Step 6 (optional)
Add tags

Step 7
Review

Choose launch template or configuration Info

Specify a launch template that contains settings common to all EC2 instances that are launched by this Auto Scaling group. If you currently use launch configurations, you might consider migrating to launch templates.

Name

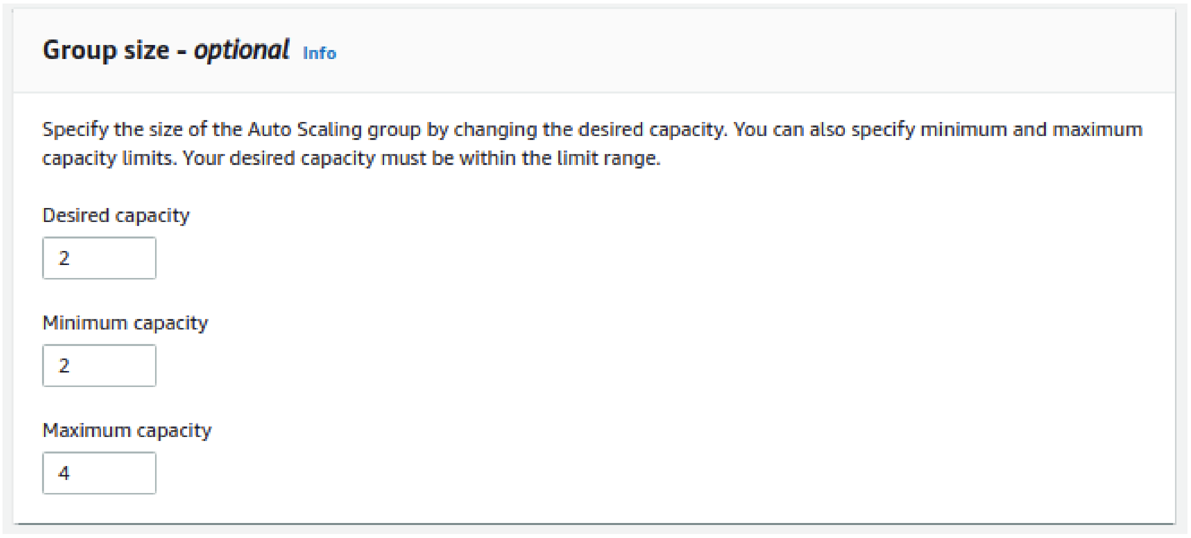
Auto Scaling group name
Enter a name to identify the group.
Project3AutoScalingGroup
Must be unique to this account in the current Region and no more than 255 characters.

Launch configuration Info — Switch to launch template

Launch configuration
Choose a launch configuration that contains the instance-level settings, such as the Amazon Machine Image (AMI), instance type, key pair, and security groups.
Project3LaunchConfiguration

Create a launch configuration

Launch configuration	AMI ID	Date created
Project3LaunchConfiguration	ami-0d96061ef24dc60ca	Mon Jun 28 2021 15:22:35 GMT+0545 (Nepal Time)
Security groups	**Instance type**	**Key pair name**
sg-0e69156b089014fd3	t3a.micro	-



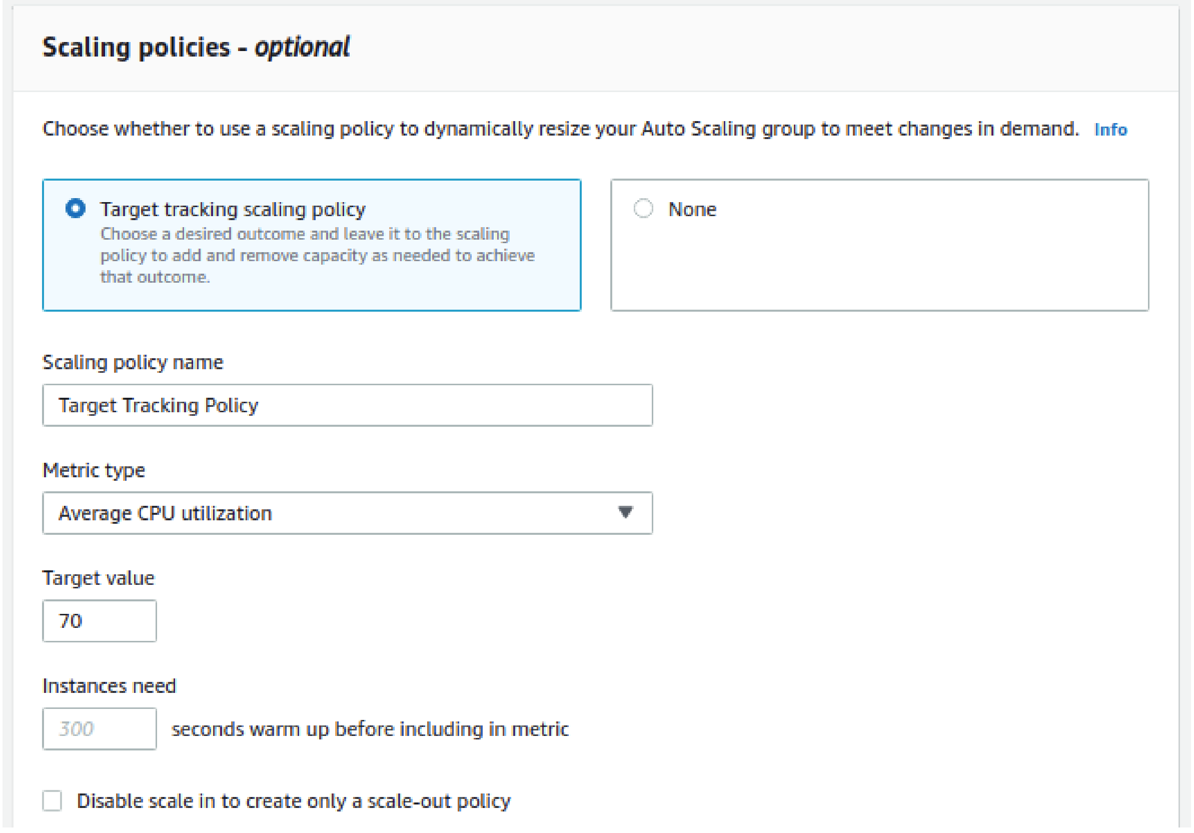
Group size - *optional* Info

Specify the size of the Auto Scaling group by changing the desired capacity. You can also specify minimum and maximum capacity limits. Your desired capacity must be within the limit range.

Desired capacity
2

Minimum capacity
2

Maximum capacity
4



Scaling policies - *optional*

Choose whether to use a scaling policy to dynamically resize your Auto Scaling group to meet changes in demand. Info

[✓ Target tracking scaling policy] Choose a desired outcome and leave it to the scaling policy to add and remove capacity as needed to achieve that outcome.

None

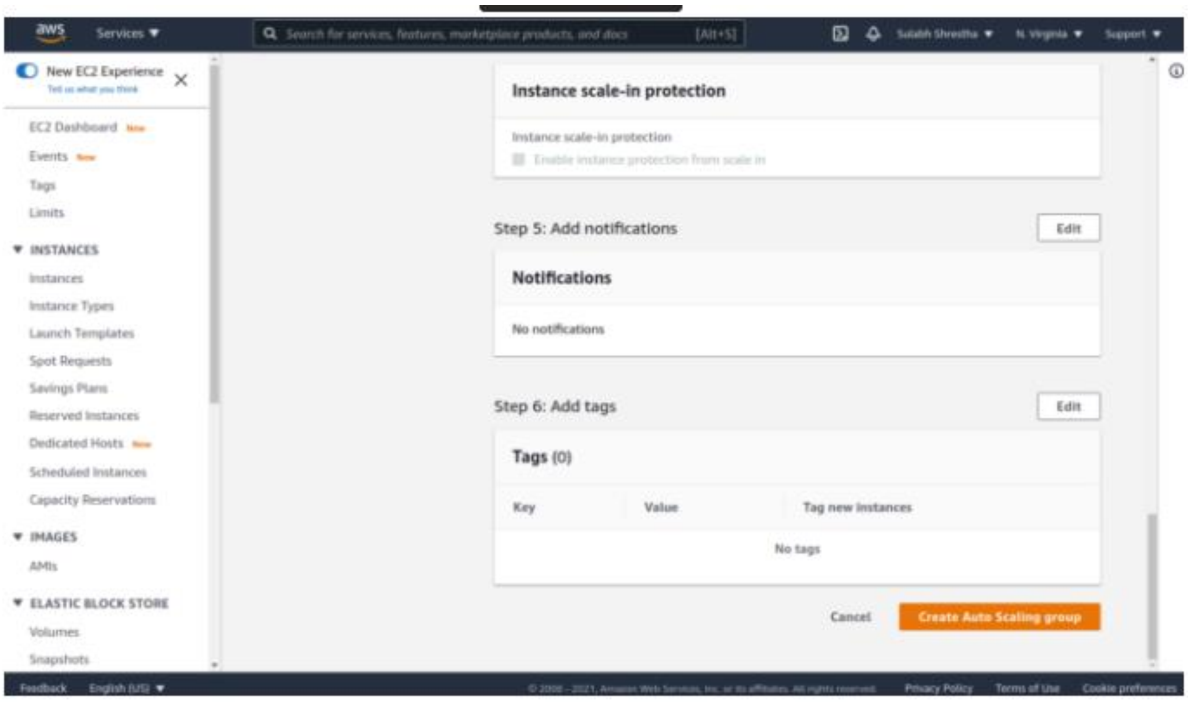
Scaling policy name: Target Tracking Policy

Metric type: Average CPU utilization

Target value: 70

Instances need 300 seconds warm up before including in metric

☐ Disable scale in to create only a scale-out policy



Instance scale-in protection

Instance scale-in protection

☐ Enable instance protection from scale in

Step 5: Add notifications — Edit

Notifications

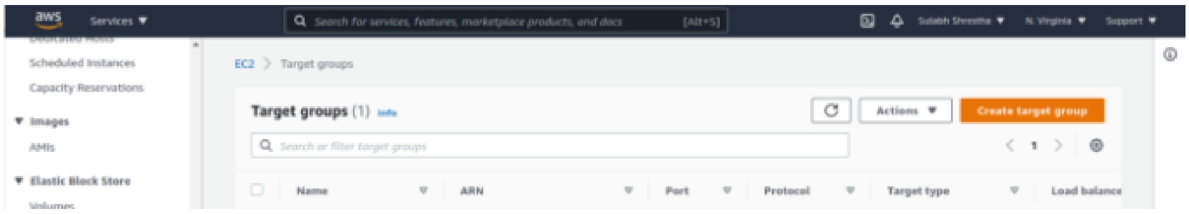
No notifications

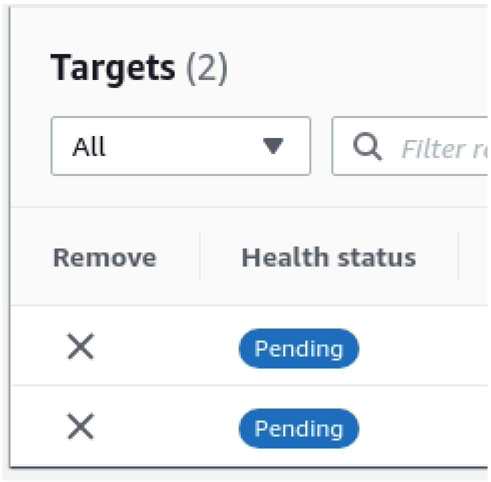
Step 6: Add tags — Edit

Tags (0)

Key	Value	Tag new instances
No tags		

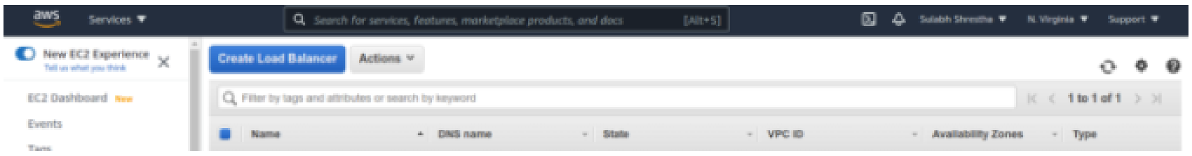
Cancel — Create Auto Scaling group

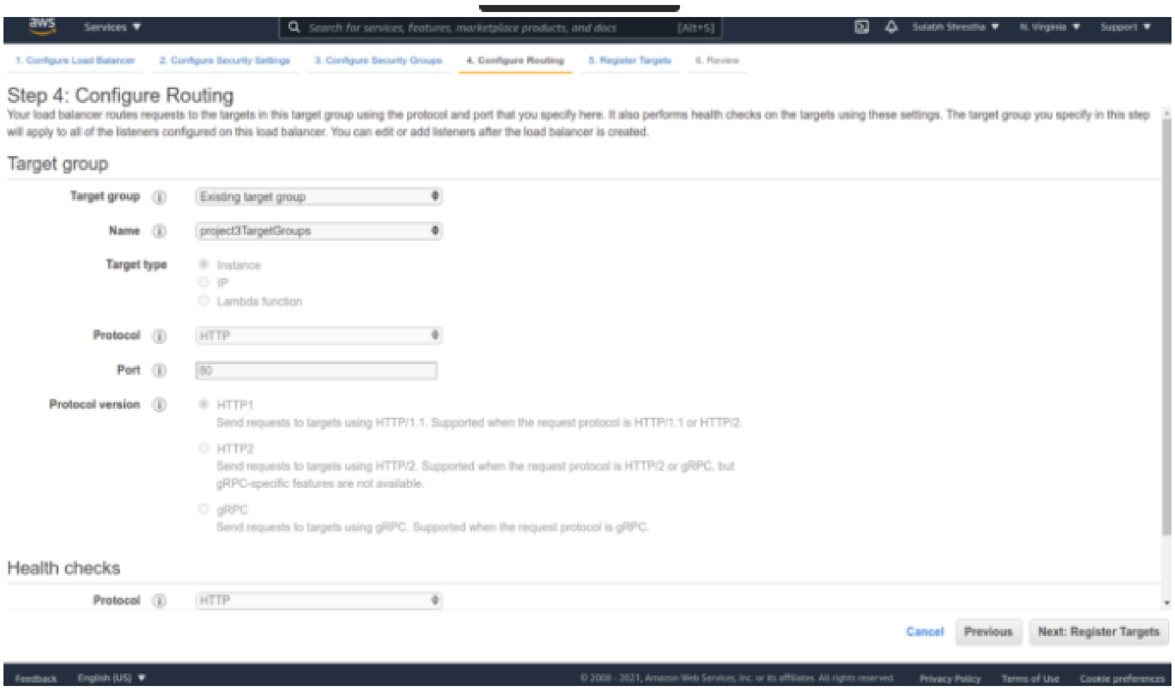




Targets (2)

Remove	Health status
✕	Pending
✕	Pending





Services
Search for services, features, marketplace products, and docs [Alt+S]
Sulabh Shrestha
N. Virginia
Support
1. Configure Load Balancer
2. Configure Security Settings
3. Configure Security Groups
4. Configure Routing
5. Register Targets
6. Review

Step 4: Configure Routing

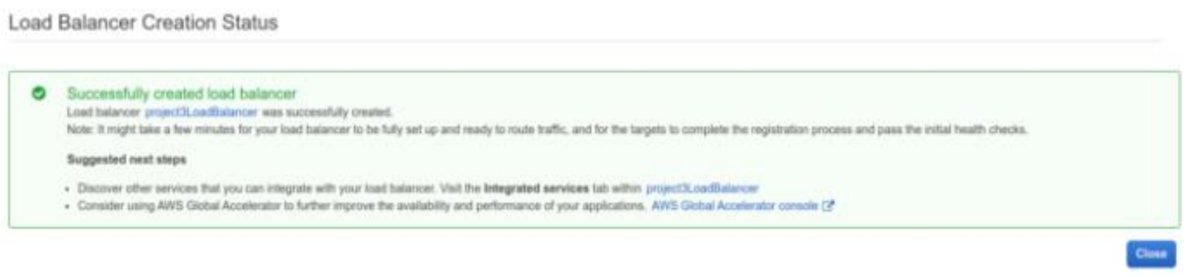
Your load balancer routes requests to the targets in this target group using the protocol and port that you specify here. It also performs health checks on the targets using these settings. The target group you specify in this step will apply to all of the listeners configured on this load balancer. You can edit or add listeners after the load balancer is created.

Target group

Target group	Existing target group
Name	project3TargetGroups
Target type	[✓ Instance] IP Lambda function
Protocol	HTTP
Port	80
Protocol version	[✓ HTTP1] Send requests to targets using HTTP/1.1. Supported when the request protocol is HTTP/1.1 or HTTP/2. HTTP2 Send requests to targets using HTTP/2. Supported when the request protocol is HTTP/2 or gRPC, but gRPC-specific features are not available. gRPC Send requests to targets using gRPC. Supported when the request protocol is gRPC.

Health checks

Protocol	HTTP

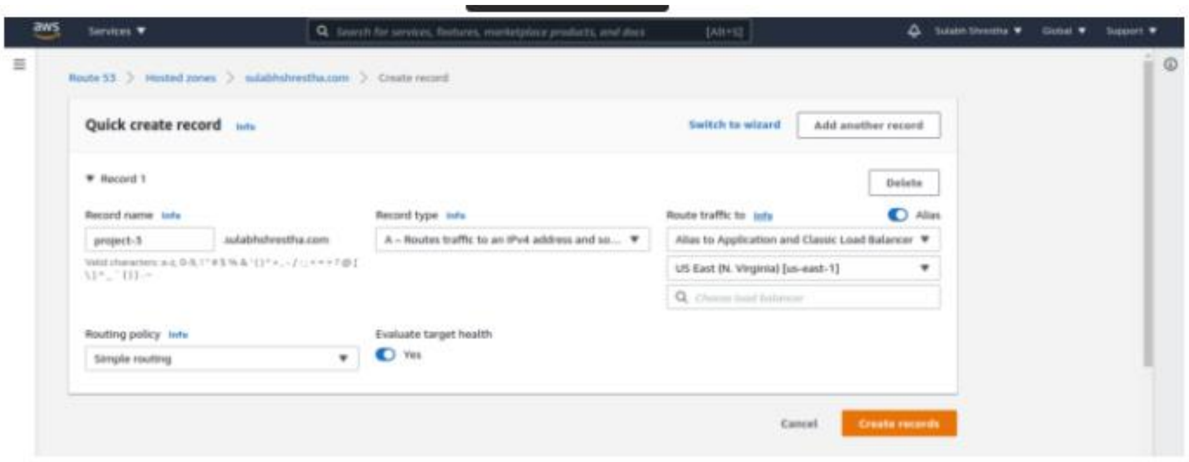


Load Balancer Creation Status

Successfully created load balancer
Load balancer project3LoadBalancer was successfully created.
Note: It might take a few minutes for your load balancer to be fully set up and ready to route traffic, and for the targets to complete the registration process and pass the initial health checks.

Suggested next steps

- Discover other services that you can integrate with your load balancer. Visit the **Integrated services** tab within project3LoadBalancer
- Consider using AWS Global Accelerator to further improve the availability and performance of your applications. AWS Global Accelerator console

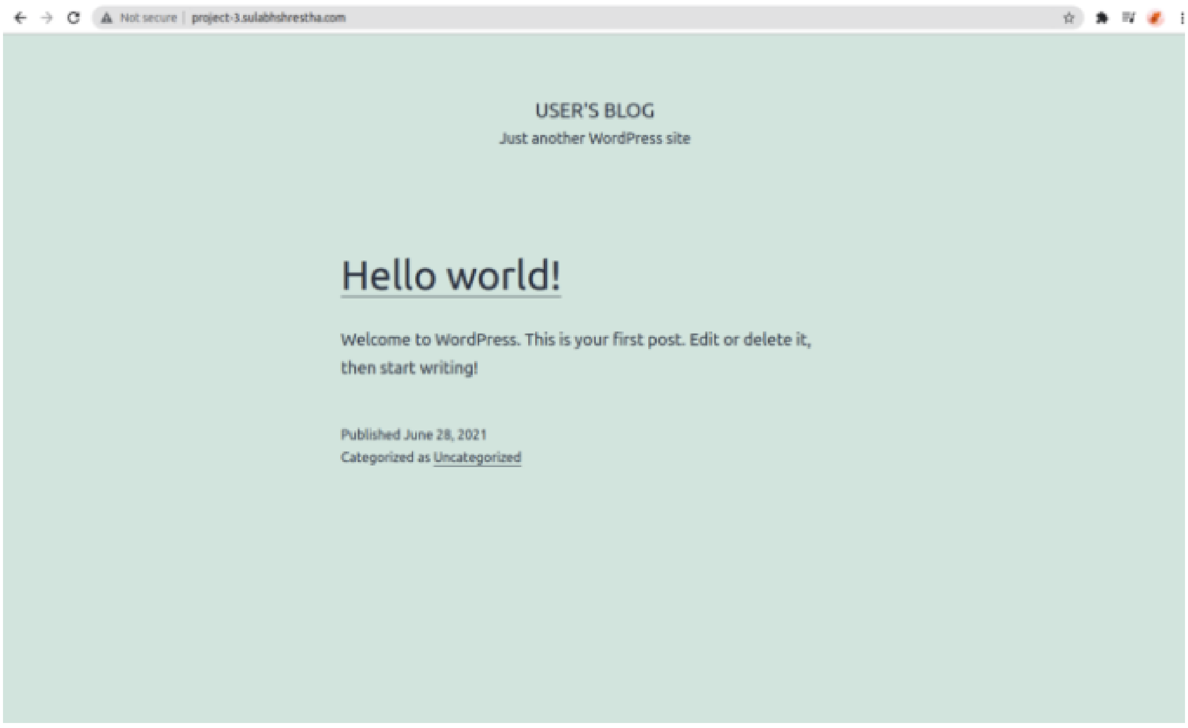


Route 53 > Hosted zones > sulabhshrestha.com > Create record

Quick create record

Record 1

Record name: project-3 .sulabhshrestha.com	Record type: A – Routes traffic to an IPv4 address and so...	Route traffic to: Alias — Alias to Application and Classic Load Balancer; US East (N. Virginia) [us-east-1]; Choose load balancer
Routing policy: Simple routing	Evaluate target health: Yes	



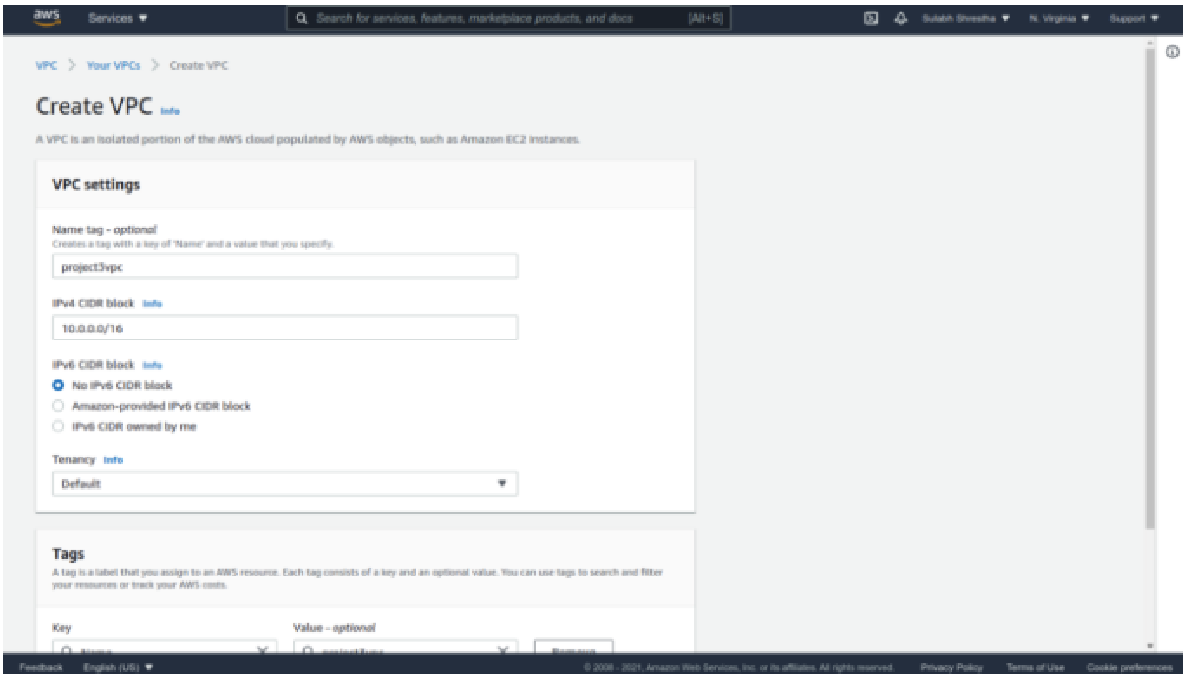
Not secure | project-3.sulabhshrestha.com

USER'S BLOG
Just another WordPress site

Hello world!

Welcome to WordPress. This is your first post. Edit or delete it, then start writing!

Published June 28, 2021
Categorized as Uncategorized



VPC > Your VPCs > Create VPC

Create VPC Info

A VPC is an isolated portion of the AWS cloud populated by AWS objects, such as Amazon EC2 instances.

VPC settings

Name tag - *optional*
Creates a tag with a key of 'Name' and a value that you specify.
project3vpc

IPv4 CIDR block Info
10.0.0.0/16

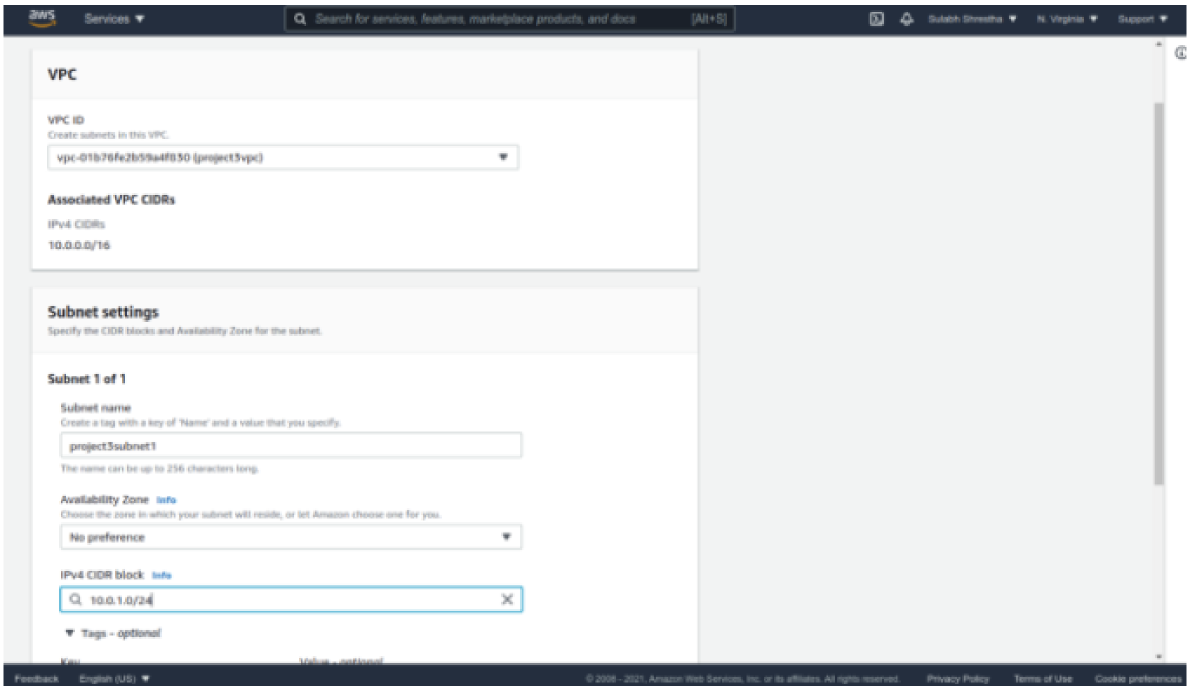
IPv6 CIDR block Info
[✓ No IPv6 CIDR block]
Amazon-provided IPv6 CIDR block
IPv6 CIDR owned by me

Tenancy Info
Default

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*

Services
Search for services, features, marketplace products, and docs [Alt+S]
Sulabh Shrestha
N. Virginia
Support
Feedback
English (US)
© 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use
Cookie preferences



VPC

VPC ID
Create subnets in this VPC.
vpc-01b76fe2b59a4f030 (project3vpc)

Associated VPC CIDRs
IPv4 CIDRs
10.0.0.0/16

Subnet settings
Specify the CIDR blocks and Availability Zone for the subnet.

Subnet 1 of 1

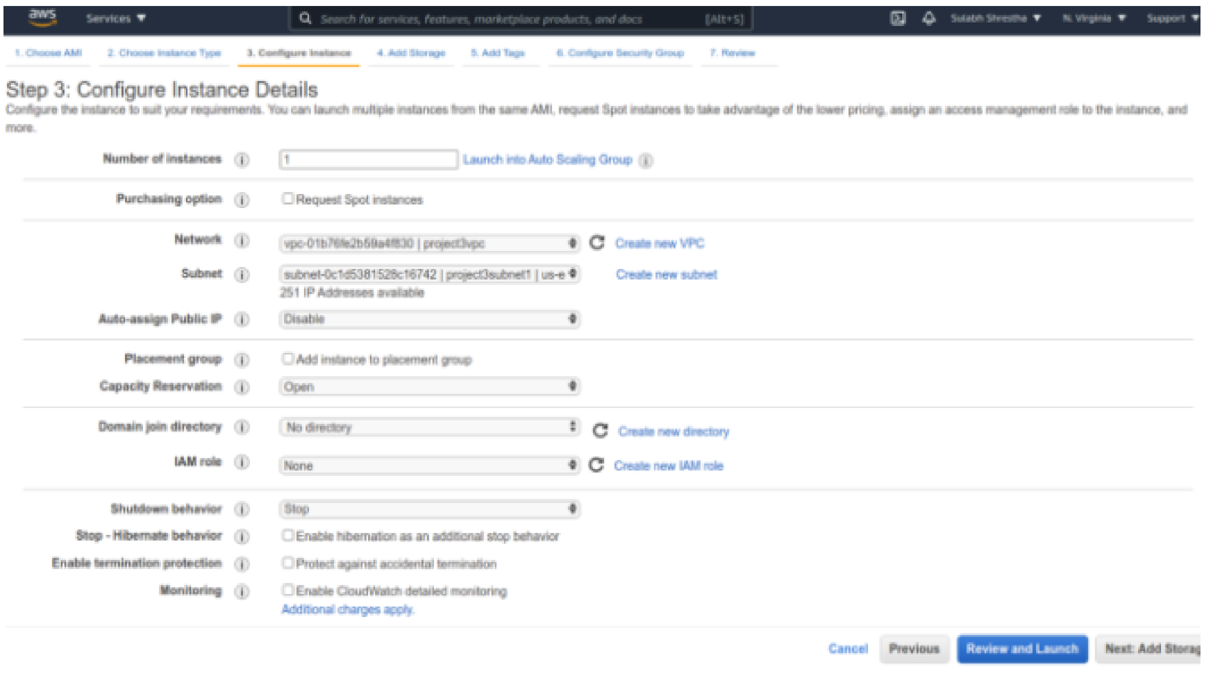
Subnet name
Create a tag with a key of 'Name' and a value that you specify.
project3subnet1
The name can be up to 256 characters long.

Availability Zone Info
Choose the zone in which your subnet will reside, or let Amazon choose one for you.
No preference

IPv4 CIDR block Info
10.0.1.0/24

Tags - *optional*

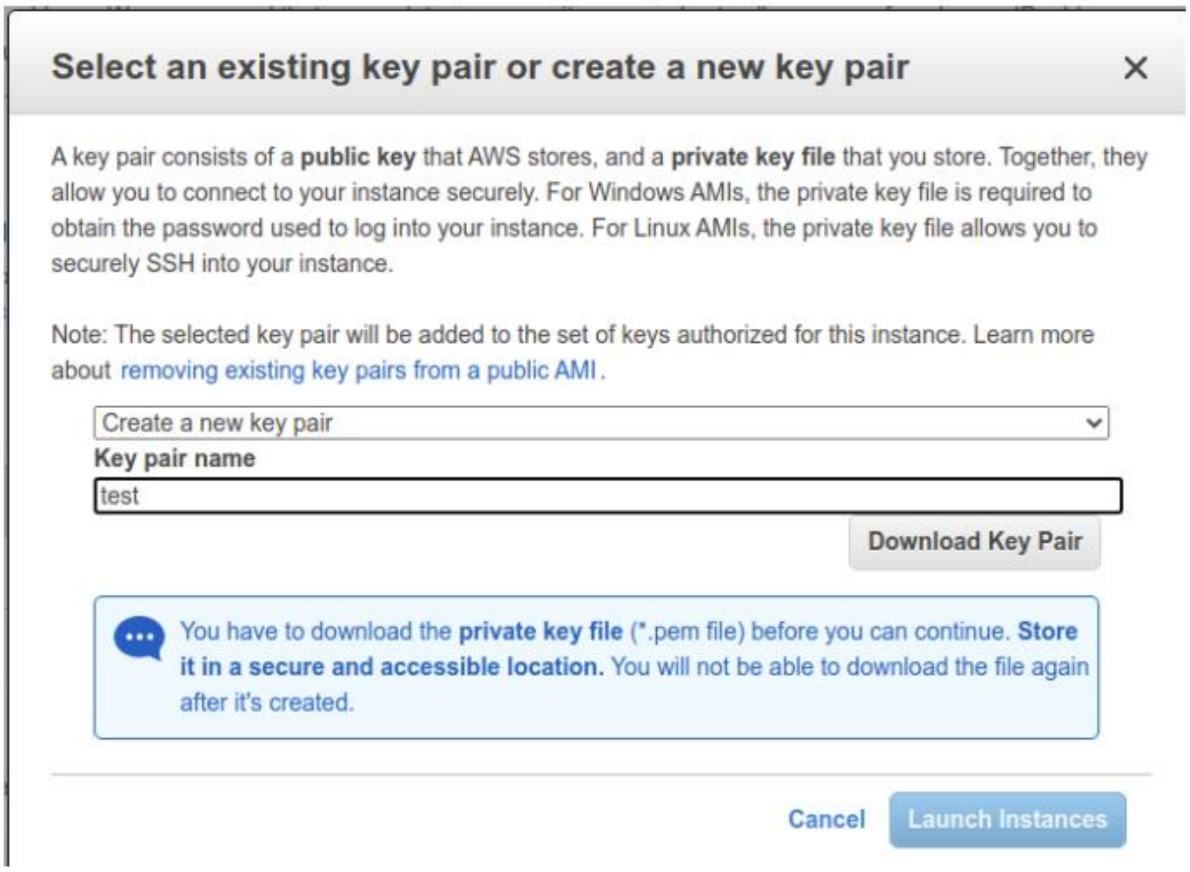
Services
Search for services, features, marketplace products, and docs [Alt+S]
Sulabh Shrestha
N. Virginia
Support
Feedback
English (US)
© 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use
Cookie preferences



Step 3: Configure Instance Details

Configure the instance to suit your requirements. You can launch multiple instances from the same AMI, request Spot instances to take advantage of the lower pricing, assign an access management role to the instance, and more.

Number of instances	1 — Launch into Auto Scaling Group
Purchasing option	☐ Request Spot instances
Network	vpc-01b76fe2b59a4f830 | project3vpc — Create new VPC
Subnet	subnet-0c1d5381528c16742 | project3subnet1 | us-e — Create new subnet 251 IP Addresses available
Auto-assign Public IP	Disable
Placement group	☐ Add instance to placement group
Capacity Reservation	Open
Domain join directory	No directory — Create new directory
IAM role	None — Create new IAM role
Shutdown behavior	Stop
Stop - Hibernate behavior	☐ Enable hibernation as an additional stop behavior
Enable termination protection	☐ Protect against accidental termination
Monitoring	☐ Enable CloudWatch detailed monitoring Additional charges apply.



Select an existing key pair or create a new key pair

A key pair consists of a **public key** that AWS stores, and a **private key file** that you store. Together, they allow you to connect to your instance securely. For Windows AMIs, the private key file is required to obtain the password used to log into your instance. For Linux AMIs, the private key file allows you to securely SSH into your instance.

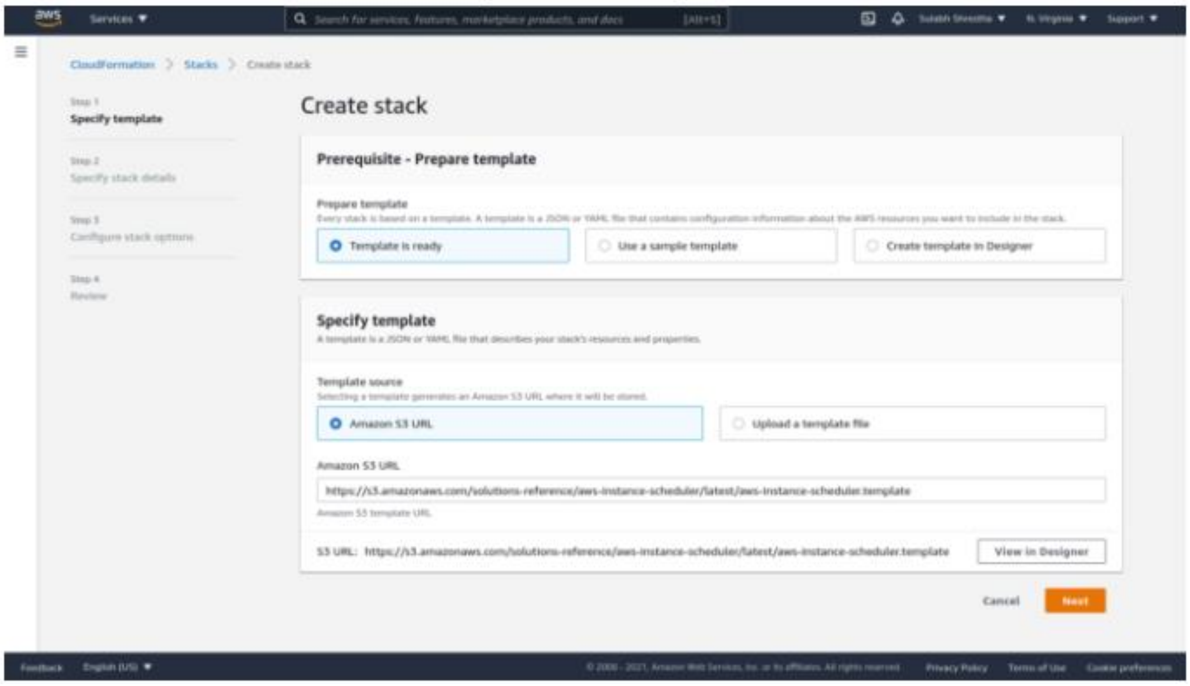
Note: The selected key pair will be added to the set of keys authorized for this instance. Learn more about removing existing key pairs from a public AMI.

Create a new key pair

Key pair name

test

You have to download the **private key file** (*.pem file) before you can continue. **Store it in a secure and accessible location.** You will not be able to download the file again after it's created.


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

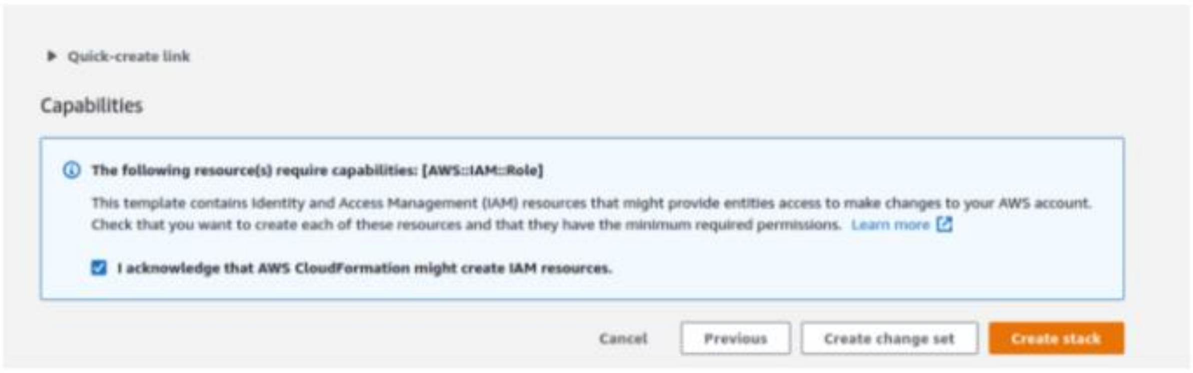
Specify template
A template is a JSON or YAML file that describes your stack's resources and properties.

Template source
Selecting a template generates an Amazon S3 URL where it will be stored.

[✓ Amazon S3 URL] Upload a template file

Amazon S3 URL
https://s3.amazonaws.com/solutions-reference/aws-instance-scheduler/latest/aws-instance-scheduler.template
Amazon S3 template URL

S3 URL: https://s3.amazonaws.com/solutions-reference/aws-instance-scheduler/latest/aws-instance-scheduler.template


▶ Quick-create link

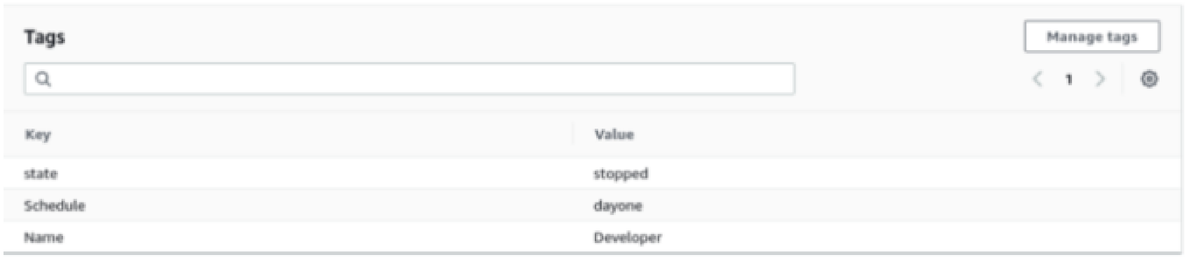
Capabilities

The following resource(s) require capabilities: [AWS::IAM::Role]

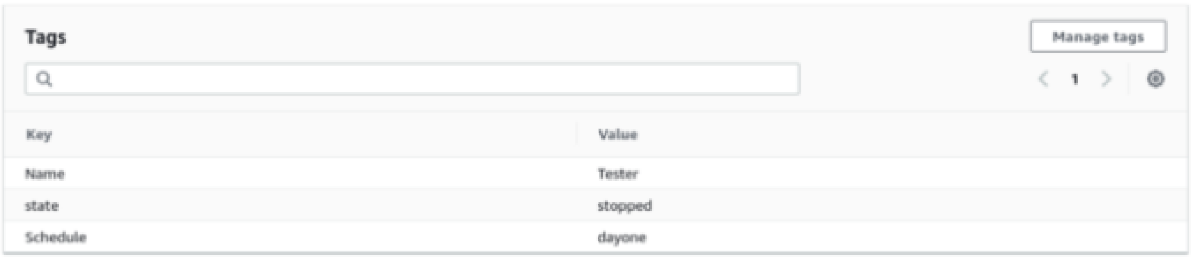
This template contains Identity and Access Management (IAM) resources that might provide entities access to make changes to your AWS account. Check that you want to create each of these resources and that they have the minimum required permissions. Learn more

☑ **I acknowledge that AWS CloudFormation might create IAM resources.**

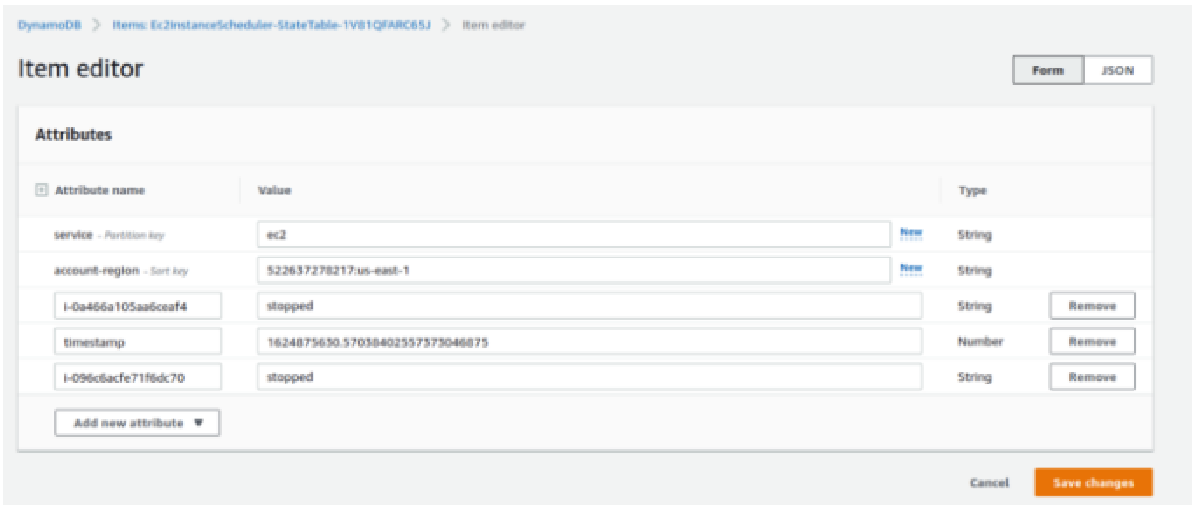
```
wget https://s3.amazonaws.com/solutions-reference/aws-instance-
scheduler/latest/scheduler-cli.zip
unzip scheduler-cli.zip
cd scheduler-cli
python setup.py install
scheduler-cli create-period --stack Ec2instanceScheduler --
region us-east-1 --name firstdayofmonth --begintime 09:00 --
endtime 06:00 --monthdays 1
scheduler-cli create-schedule --stack Ec2instanceScheduler --
name dayone --region us-east-1 --periods firstdayofmonth --
timezone UTC
```


Tags

Key	Value
state	stopped
Schedule	dayone
Name	Developer



Key	Value
Name	Tester
state	stopped
Schedule	dayone



DynamoDB > Items: Ec2InstanceScheduler-StateTable-1V81QFARC65J > Item editor
Form
JSON
Add new attribute ▼
Cancel
Save changes

Item editor

Attributes

Attribute name	Value		Type	
service - Partition key	ec2	New	String	
account-region - Sort key	522637278217:us-east-1	New	String	
i-0a466a105aa6ceaf4	stopped		String	Remove
timestamp	1624875630.57038402557373046875		Number	Remove
i-096c6acfe71f6dc70	stopped		String	Remove