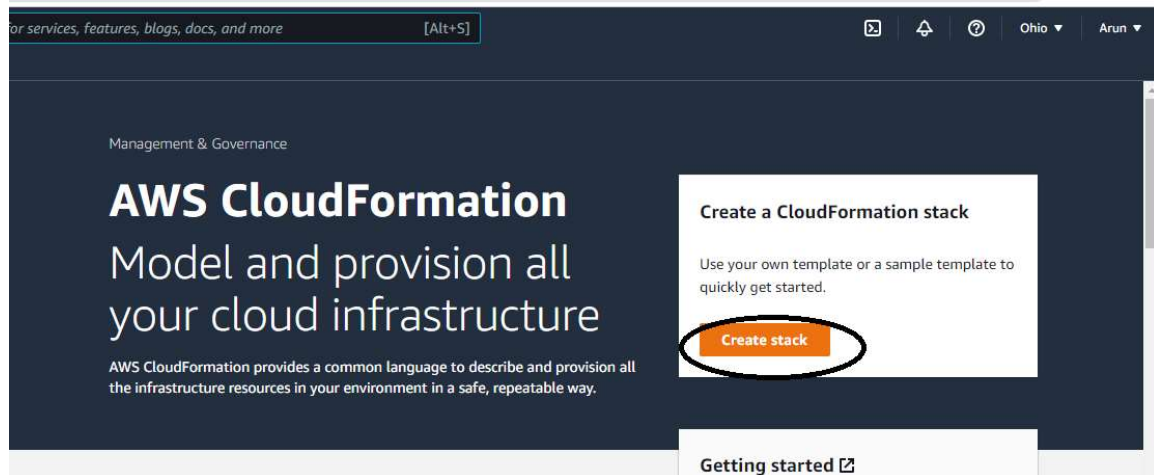


# RDS with CloudFormation

**Amazon Relational Database Service (RDS) is a collection of managed services that makes it simple to set up, operate, and scale databases in the cloud. Choose from seven popular engines — Amazon Aurora with MySQL compatibility, Amazon Aurora with PostgreSQL compatibility, MySQL, MariaDB, PostgreSQL, Oracle, and SQL Server — and deploy on-premises with Amazon RDS on AWS Outposts.**

Uploading Template file :

1. Create stack :



2. prepare template upload it to to stack

features, blogs, docs, and more [Alt+S]

**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 No file chosen  
JSON or YAML formatted file

S3 URL: Will be generated when template file is uploaded

View in Designer

Cancel **Next**

### 3. Enter Stack name and Next

res, blogs, docs, and more [Alt+S]

**Specify stack details**

**Stack name**

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

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

**No parameters**  
There are no parameters defined in your template

Cancel Previous **Next**

### 4. Tags

ures, blogs, docs, and more [Alt+S]

ck

## Configure stack options

### Tags

You can specify tags (key-value pairs) to apply to resources in your stack. You can add up to 50 unique tags for each stack. [Learn more](#)

Key	Value	Remove

Add tag

### Permissions

Choose an IAM role to explicitly define how CloudFormation can create, modify, or delete resources in the stack. If you don't choose a role, CloudFormation uses permissions based on your user credentials. [Learn more](#)

IAM role - optional  
Choose the IAM role for CloudFormation to use for all operations performed on the stack.

iamRoleName	Remove

## 5.click Next

features, blogs, docs, and more [Alt+S]

Preserves the state of successfully provisioned resources, while rolling back failed resources to the last known stable state. Resources without a last known stable state will be deleted upon the next stack operation.

## Advanced options

You can set additional options for your stack, like notification options and a stack policy. [Learn more](#)

► **Stack policy**  
Defines the resources that you want to protect from unintentional updates during a stack update.

► **Rollback configuration**  
Specify alarms for CloudFormation to monitor when creating and updating the stack. If the operation breaches an alarm threshold, CloudFormation rolls it back. [Learn more](#)

► **Notification options**

► **Stack creation options**

Cancel Previous **Next**

## 5. Review and Acknowledge

more [Alt+S] Ohio Arun

**Stack creation options**

Timeout  
-

Termination protection  
Disabled

► 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.

Cancel Previous Create change set **Create stack**

### Yaml Code :

Resources:

RDS:

Type: AWS::RDS::DBInstance

Properties:

DBName: MyRDS

MasterUsername: rdsuser

MasterUserPassword: PaSsWoRd

Engine : MySQL

DBInstanceClass: db.t2.micro

StorageType: gp2

PubliclyAccessible: True

AllocatedStorage: "20"

DBInstanceIdentifier: !Join ["-",["MyDbInstance", !Ref "AWS::Region"]]

The screenshot shows the AWS CloudFormation console for an RDS stack. The 'Events' tab is selected, displaying a list of events. The first event is 'CREATE\_IN\_PROGRESS' with a status reason of 'User Initiated'.

Timestamp	Logical ID	Status	Status reason
2022-04-21 13:43:33 UTC+0530	RDS	CREATE_IN_PROGRESS	User Initiated

The screenshot shows the AWS RDS console 'Databases' page. A table lists the database instances. The first instance, 'mydbinstance-us-east-2', is highlighted with a red box. It is an 'Instance' of 'MySQL Community' engine, located in 'us-east-2a' region, with a size of 'db.t2.micro' and a status of 'Available'.

DB identifier	Role	Engine	Region & AZ	Size	Status	CPU
mydbinstance-us-east-2	Instance	MySQL Community	us-east-2a	db.t2.micro	Available	

aws.amazon.com/rds/home?region=us-east-2#database:id=mydbinstance-us-east-2;is-cluster=false

services, features, blogs, docs, and more [Alt+S]

Ohio Arun

### Summary

DB identifier mydbinstance-us-east-2	CPU 5.08%	Status Backing-up	Class db.t2.micro
Role Instance	Current activity 0 Connections	Engine MySQL Community	Region & AZ us-east-2a

Connectivity & security Monitoring Logs & events Configuration Maintenance & backups Tags

### Connectivity & security

Endpoint & port Endpoint mydbinstance-us-east-2.cizna11kcpdg.us-east-2.rds.amazonaws.com Port 3306	Networking Availability Zone us-east-2a VPC vpc-0150e11be96adb6ad Subnet group default Subnets	Security VPC security groups default (sg-01e1bc4646e97f351) Active Publicly accessible Yes Certificate authority rds-ca-2019
--	---	---