



Prashant Lakhera

Mar 3, 2019 · 3 min read · Listen



# 100 Days of DevOps — Day 21- MySQL RDS Database Creation using Terraform

Welcome to Day 21 of 100 Days of DevOps, Let continue our journey with terraform and today we are going to create MySql database using terraform

## What Is Amazon Relational Database Service (Amazon RDS)?

Amazon Relational Database Service (Amazon RDS) is a web service that makes it easier to set up, operate, and scale a relational database in the cloud. It provides cost-efficient, resizable capacity for an industry-standard relational database and manages common database administration tasks.

### Step1: Create a DB subnet group

- In order to create a new MySql database we first need to create a subnet group and assign at least two subnets to it.

```
1 resource "aws_db_subnet_group" "rds-private-subnet" {
2   name = "my-rds-sg"
3   subnet_ids = ["${var.rds_subnet1}", "${var.rds_subnet2}"]
4 }
```

rds\_db\_subnet.tf hosted with by GitHub [view raw](#)

### Step2: Create a Security Group to allow mysql port 3306

```
1 resource "aws_security_group" "rds-sg" {
2   name = "my-rds-sg"
3   vpc_id = "${var.vpc_id}"
4 }
5
6
7 # Ingress Security Port 3306
8 resource "aws_security_group_rule" "mysql_inbound_access" {
9   from_port = 3306
10  protocol = "tcp"
11  security_group_id = "${aws_security_group.rds-sg.id}"
12  to_port = 3306
13  type = "ingress"
14  cidr_blocks = ["0.0.0.0/0"]
15 }
```

mysql\_rds\_port.tf hosted with by GitHub [view raw](#)

### Step3: Next step is to create MySQL resource

```
1 resource "aws_db_instance" "my_test_mysql" {
2   allocated_storage = 20
3   storage_type = "gp2"
4   engine = "mysql"
5   engine_version = "5.7"
6   instance_class = "${var.db_instance}"
7   name = "myrds_testmysql"
8   username = "admin"
9   password = "admin123"
10  parameter_group_name = "default.mysql5.7"
11  db_subnet_group_name = "${aws_db_subnet_group.rds-private-subnet.name}"
12  vpc_security_group_ids = ["${aws_security_group.rds-sg.id}"]
13  allow_major_version_upgrade = true
14  auto_minor_version_upgrade = true
15  backup_retention_period = 35
16  backup_window = "22:00-23:00"
17  maintenance_window = "Sat:00:00-Sat:03:00"
18  multi_az = true
19  skip_final_snapshot = true
20 }
```

rds\_mysql.tf hosted with by GitHub [view raw](#)

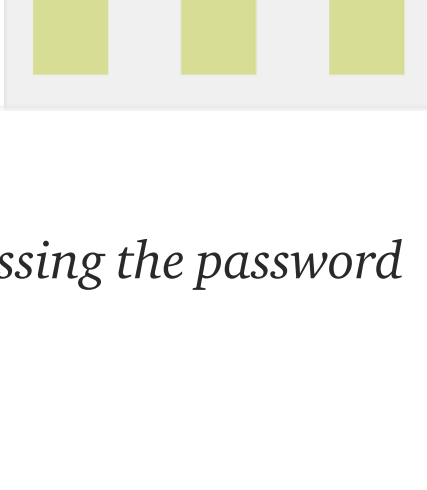
**\* allocated\_storage:** This is the amount in GB  
**\* storage\_type:** Type of storage we want to allocate(options available "standard" (magnetic), "gp2" (general purpose SSD), or "io1" (provisioned IOPS SSD)  
**\* engine:** Database engine(for supported values check [https://docs.aws.amazon.com/AmazonRDS/latest/APIReference/API\\_CreateDBInstance.html](https://docs.aws.amazon.com/AmazonRDS/latest/APIReference/API_CreateDBInstance.html)) eg: Oracle, Amazon Aurora,Postgres  
**\* engine\_version:** engine version to use  
**\* instance\_class:** instance type for rds instance  
**\* name:** The name of the database to create when the DB instance is created.  
**\* username:** Username for the master DB user.  
**\* password:** Password for the master DB user  
**\* db\_subnet\_group\_name:** DB instance will be created in the VPC associated with the DB subnet group. If unspecified, will be created in the default VPC  
**\* vpc\_security\_group\_ids:** List of VPC security groups to associate.  
**\* allows\_major\_version\_upgrade:** Indicates that major version upgrades are allowed. Changing this parameter does not result in an outage and the change is asynchronously applied as soon as possible.  
**\* auto\_minor\_version\_upgrade:**Indicates that minor engine upgrades will be applied automatically to the DB instance during the maintenance window. Defaults to true.  
**\* backup\_retention\_period:** The days to retain backups for. Must be between 0 and 35. When creating a Read Replica the value must be greater than 0  
**\* backup\_window:** The daily time range (in UTC) during which automated backups are created if they are enabled. Must not overlap with maintenance window  
**\* maintenance\_window:** The window to perform maintenance in. Syntax: "ddd:hh24:mi-ddd:hh24:mi".  
**\* multi\_az:** Specifies if the RDS instance is multi-AZ  
**\* skip\_final\_snapshot:** Determines whether a final DB snapshot is created before the DB instance is deleted. If true is specified, no DBSnapshot is created. If false is specified, a DB snapshot is created before the DB instance is deleted, using the value from final\_snapshot\_identifier. Default is false

### GitHub Link for Complete Code

**100daysofdevops/100daysofdevops**

Contribute to 100daysofdevops/100daysofdevops development by creating an account on GitHub.

github.com



- One of the clear issues I see in the above code is that we are passing the password in the plain text inside the terraform code

- Now to encrypt that password we can use KMS

### Step1: First Create KMS Keys

```
1 resource "aws_kms_key" "rds-key" {
2   description = "key to encrypt rds password"
3   tags {
4     Name = "my-rds-kms-key"
5   }
6 }
7
8 resource "aws_kms_alias" "rds-kms-alias" {
9   target_key_id = "${aws_kms_key.rds-key.id}"
10  name = "alias/rds-kms-key"
11 }
```

kms\_keys.tf hosted with by GitHub [view raw](#)

### Step2: Now use that key to encrypt a secret on a command line

```
aws kms encrypt --key-id <kms key id> --plaintext admin123 --output text --query CiphertextBlob
```

### Step3: Now the encoded string we got pass it as a payload in your terraform code

```
1 data "aws_kms_secret" "rds-secret" {
2   "secret" {
3     name = "master_password"
4     payload = "payload value here"
5   }
6 }
7
8 resource "aws_db_instance" "my_test_mysql" {
9   allocated_storage = 20
10  storage_type = "gp2"
11  engine = "mysql"
12  engine_version = "5.7"
13  instance_class = "${var.db_instance}"
14  name = "myrds_testmysql"
15  username = "admin"
16  password = "${data.aws_kms_secret.rds-secret.master_password}"
17  parameter_group_name = "default.mysql5.7"
18  db_subnet_group_name = "${aws_db_subnet_group.rds-private-subnet.name}"
19  vpc_security_group_ids = ["${aws_security_group.rds-sg.id}"]
20  allow_major_version_upgrade = true
21  auto_minor_version_upgrade = true
22  backup_retention_period = 35
23  backup_window = "22:00-23:00"
24  maintenance_window = "Sat:00:00-Sat:03:00"
25  multi_az = true
26  skip_final_snapshot = true
27 }
```

kms\_rds\_password.tf hosted with by GitHub [view raw](#)

- Now why I didn't put this solution in first place and the reason for that, because of the below-mentioned error and I want to present a working solution

```
1 $ terraform plan
2 Refreshing Terraform state in-memory prior to plan...
3 The refreshed state will be used to calculate this plan, but will not be
4 persisted to local or remote state storage.
5
6 aws_kms_key.rds-key: Refreshing state... (ID: 9731dd04-5859-430b-aa92-c27c517ecb10)
7 data.aws_kms_secret.rds: Refreshing state...
8 data.aws_availability_zones.available: Refreshing state...
9 aws_kms_alias.rds-kms-alias: Refreshing state... (ID: alias/rds-kms-key)
10
11 Error: Error refreshing state: 1 error(s) occurred:
12
13 * data.aws_kms_secret.rds: 1 error(s) occurred:
14
15 * data.aws_kms_secret.rds: data.aws_kms_secret.rds: This data source has been replaced
```

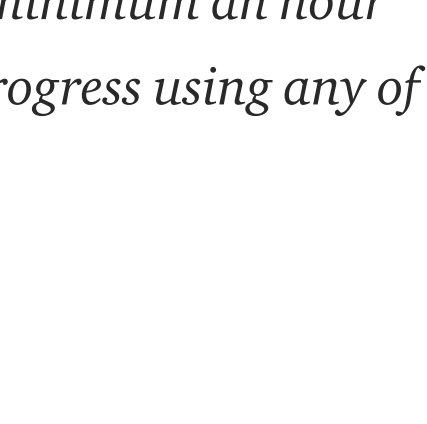
terraform\_plan\_rds hosted with by GitHub [view raw](#)

- There is already a bug opened for this issue

**data-source/aws\_kms\_secret: Soft remove data source type with removal message by bfriad - Pull...**

References: #5144 #5195 The aws\_kms\_secret data source uses dynamic attribute functionality which is not supported in...

github.com



Looking forward from you guys to join this journey and spend a minimum an hour every day for the next 100 days on DevOps work and post your progress using any of the below medium.

- Twitter: [@100daysofdevops](#) OR [@lakhera2015](#)

- Facebook: <https://www.facebook.com/groups/795382630808645/>

- Medium: <https://medium.com/@devopslearning>

- Slack: <https://devops-myworld.slack.com/messages/CF41EFG49/>

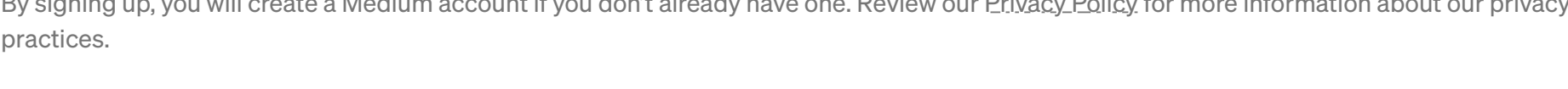
- GitHub Link:<https://github.com/100daysofdevops>

## Reference

**100 Days of DevOps — Day 0**

D-day is just one day away and finally, this is a continuation of the post(I posted a month earlier)

medium.com



Get an email whenever Prashant Lakhera publishes.

Subscribe

By signing up, you will create a Medium account if you don't already have one. Review our [Privacy Policy](#) for more information about our privacy practices.

## More from Prashant Lakhera

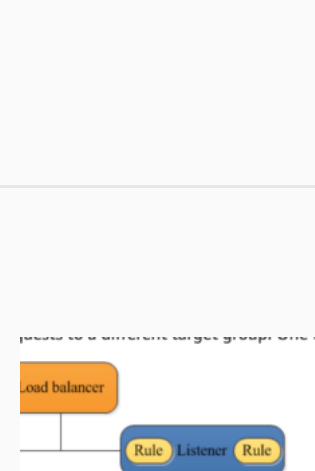
AWS Community Builder, Ex-Redhat, Author, Blogger, YouTuber, RHCA, RHCTS, RHCE, Docker Certified,4XAWS, CCNA, MCP, Certified Jenkins, Terraform Certified, 1XGCP

Mar 2, 2019

### 100 Days of DevOps — Day 20— Auto-Scaling Group using Terraform

Welcome to Day 20 of 100 Days of DevOps, Let continue our terraform journey, yesterday we created an application load balancer, let expand th...

AWS · 4 min read



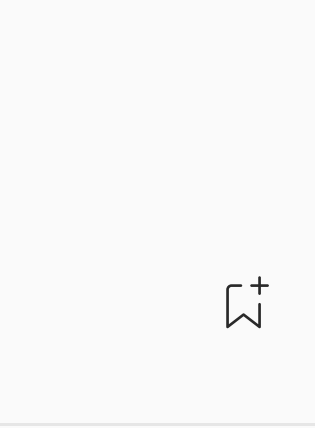
Share your ideas with millions of readers. [Write on Medium](#)

Mar 1, 2019

### 100 Days of DevOps — Day 19 - Application Load Balancer using Terraform

Welcome to Day 19 of 100 Days of DevOps, Let continue our terraform journey and see how can we create Application Load Balancer using...

Cloud Computing · 4 min read



Feb 28, 2019

### 100 Days of DevOps — Day 18-Add monitoring to these instances using Terraform(CloudWatch and SNS)

Check the updated 101 Days of DevOps Course Course Registration link:

<https://www.101daysofdevops.com/register/> Course Link:...

AWS · 3 min read

Feb 27, 2019

### 100 Days of DevOps — Day 17- Creating EC2 Instance using Terraform

Check the updated 101 Days of DevOps Course Course Registration link:

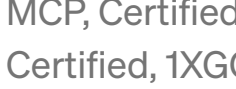
<https://www.101daysofdevops.com/register/> Course Link:...

AWS · 5 min read



Love podcasts or audiobooks? Learn on the go with our new app. [Try Knowable](#)

[Get started](#) [Sign In](#)




Prashant Lakhera

4.7K Followers

AWS Community Builder, Ex-Redhat, Author, Blogger, YouTuber, RHCA, RHCTS, RHCE, Docker Certified,4XAWS, CCNA, MCP, Certified Jenkins, Terraform Certified, 1XGCP

[Follow](#)


## More from Medium

 Br... in Open De...

Create a CloudFront distribution with...

 Sandun Dayana...  
Installing Docker on an AWS EC2 instance



 Apoti... in Dev G...  
Let's talk about K8S Certifications



 Başak Nisan İvg...  
Creating Linux Server with Using AWS



Help Status Writers Blog Careers  
Privacy Terms About Knowable