AWS-Relational Data Store

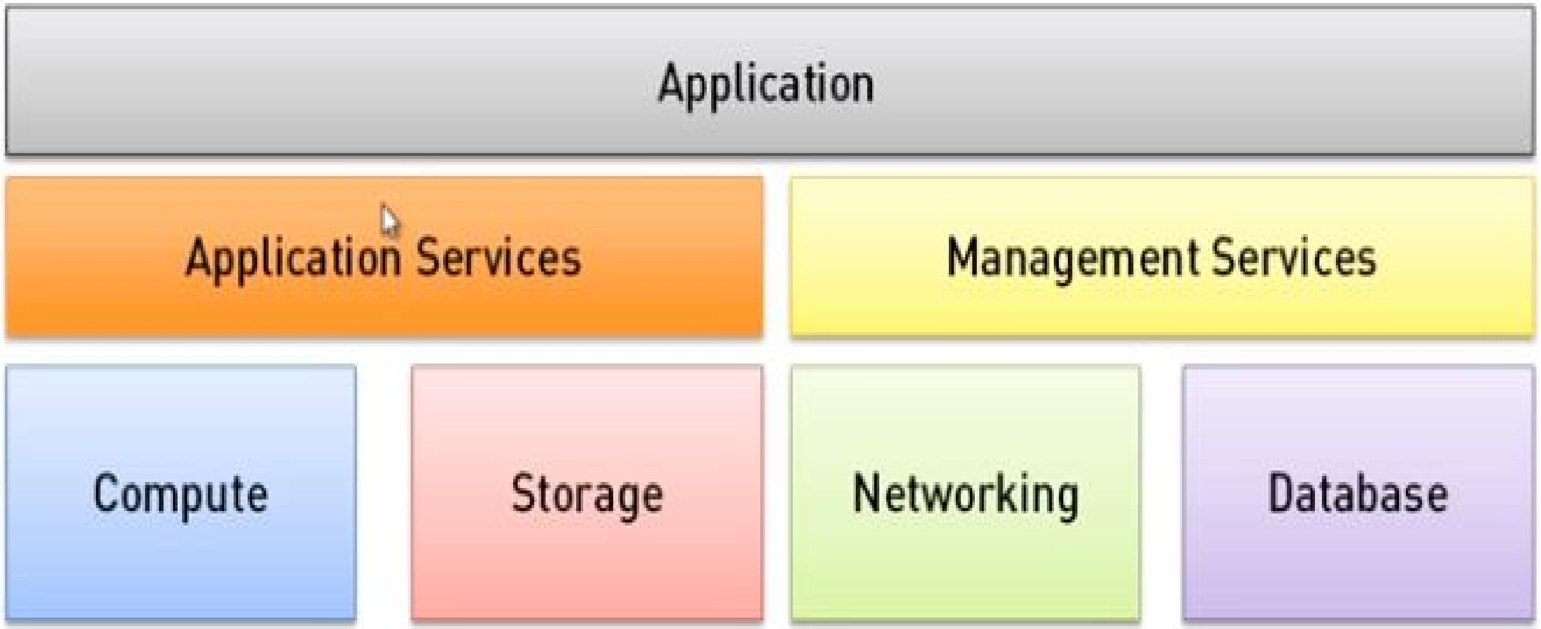
Tamal Dey

MCA,PESU

Agenda

- What powers the application?
- Dealing with databases on Cloud
- Key concepts of Amazon RDS
- Migrating the database to Amazon RDS

What Powers the application?



Application

Application Services

Management Services

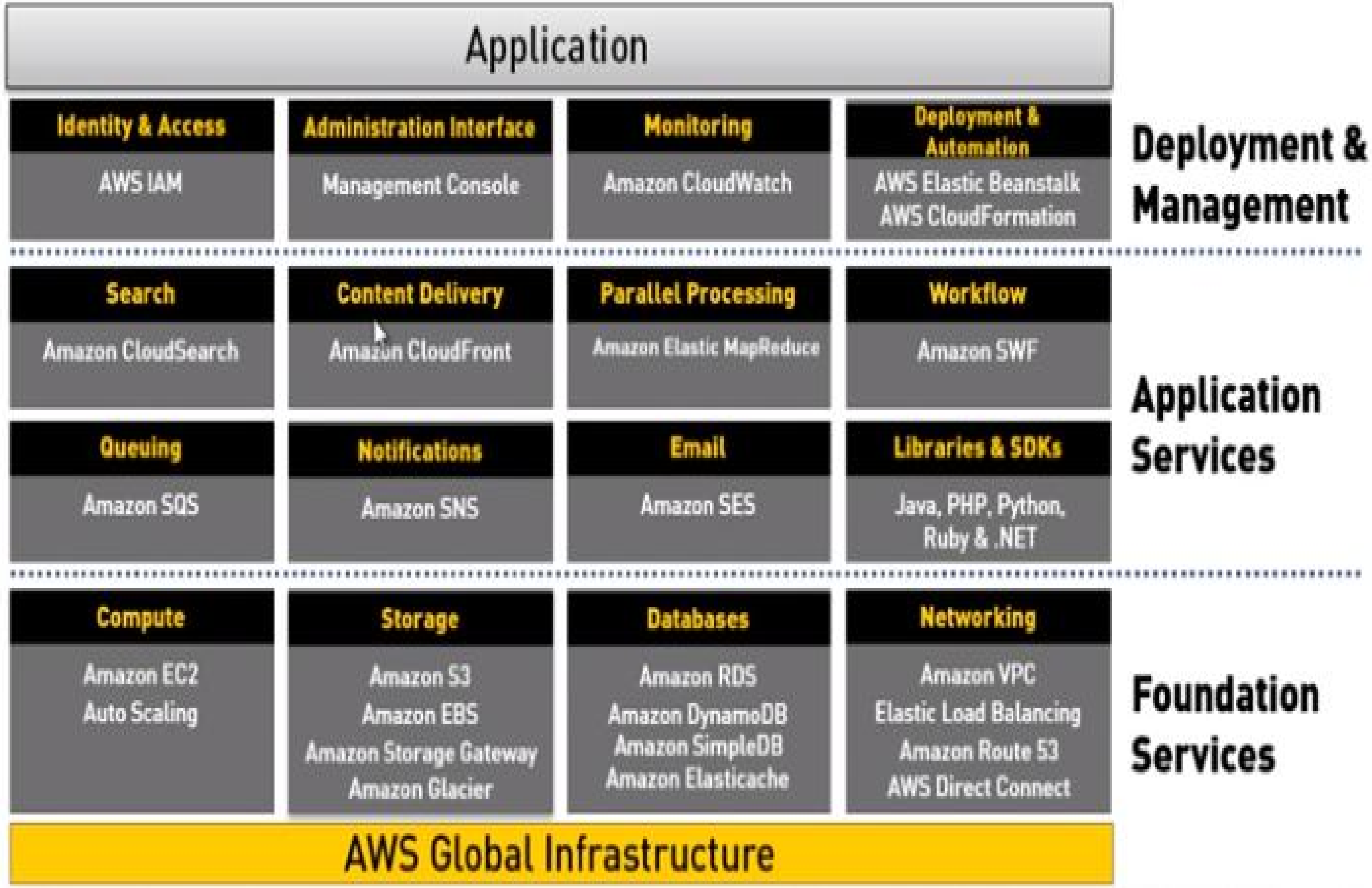
Compute

Storage

Networking

Database

Big Picture of AWS



Application

Identity & Access AWS IAM	**Administration Interface** Management Console	**Monitoring** Amazon CloudWatch	**Deployment & Automation** AWS Elastic Beanstalk AWS CloudFormation	**Deployment & Management**
Search Amazon CloudSearch	**Content Delivery** Amazon CloudFront	**Parallel Processing** Amazon Elastic MapReduce	**Workflow** Amazon SWF	**Application Services**
Queuing Amazon SQS	**Notifications** Amazon SNS	**Email** Amazon SES	**Libraries & SDKs** Java, PHP, Python, Ruby & .NET	
Compute Amazon EC2 Auto Scaling	**Storage** Amazon S3 Amazon EBS Amazon Storage Gateway Amazon Glacier	**Databases** Amazon RDS Amazon DynamoDB Amazon SimpleDB Amazon Elasticache	**Networking** Amazon VPC Elastic Load Balancing Amazon Route 53 AWS Direct Connect	**Foundation Services**

AWS Global Infrastructure

Quiz

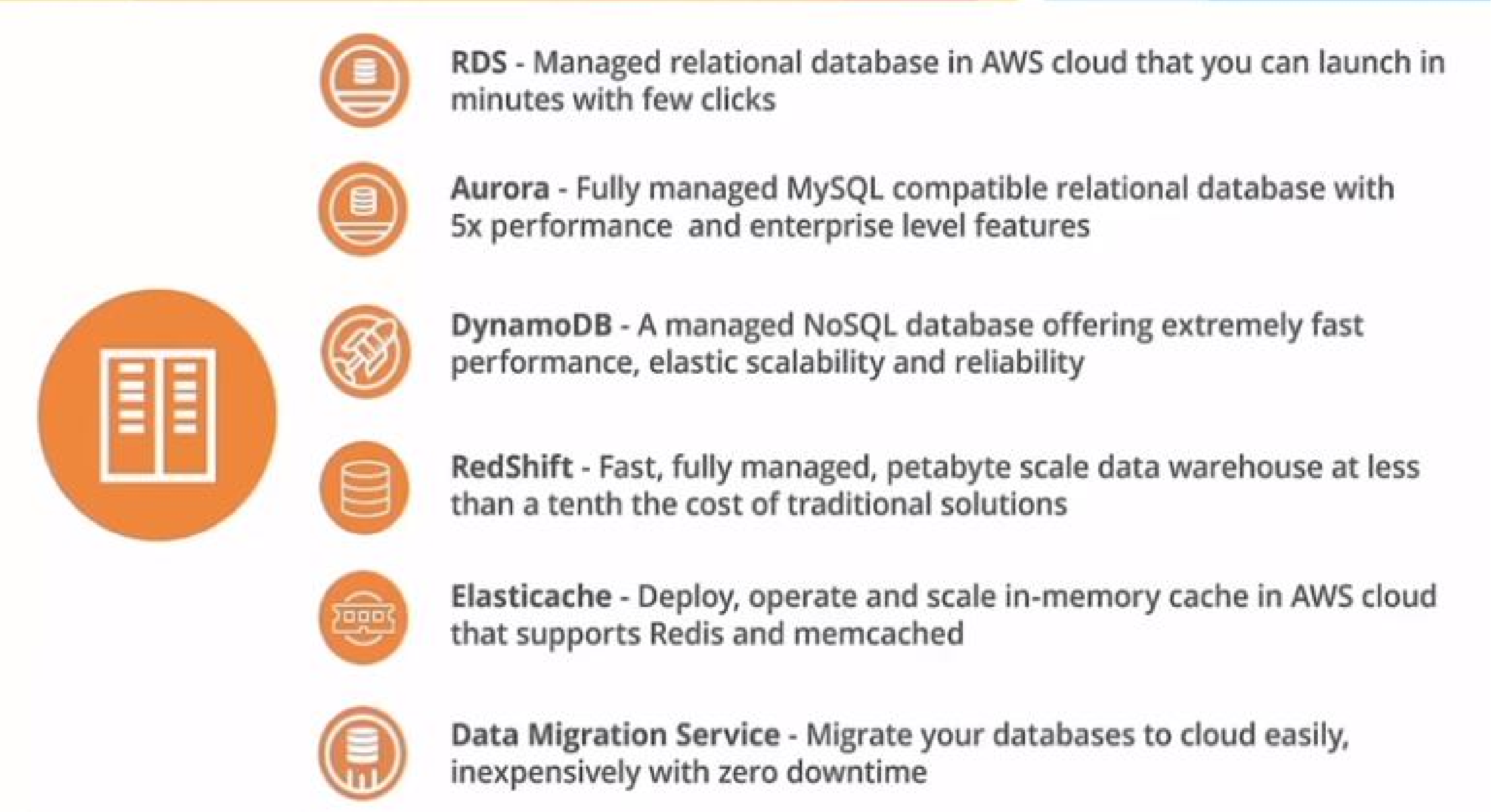
- Can we Run a Database in EC2 instance?

YES

Then why Amazon Relational Database Service ????

Amazon will manage the DB

AWS-Database Service



RDS - Managed relational database in AWS cloud that you can launch in minutes with few clicks

Aurora - Fully managed MySQL compatible relational database with 5x performance and enterprise level features

DynamoDB - A managed NoSQL database offering extremely fast performance, elastic scalability and reliability

RedShift - Fast, fully managed, petabyte scale data warehouse at less than a tenth the cost of traditional solutions

Elasticache - Deploy, operate and scale in-memory cache in AWS cloud that supports Redis and memcached

Data Migration Service - Migrate your databases to cloud easily, inexpensively with zero downtime

RDS

- Setup, operate and scale relational databases
- Access to familiar databases such as MySQL, Oracle, MS SQL Server
- Support scale out for read-heavy database workloads on MySQL
- No upfront investment ; pay-as-you-go pricing
- **Facilities:** Manage patching, backup and recovery
- **Automatic backup**
  - Daily basis & Transaction Logs
  - Retention Period- Point in time recovery1-35 Days
- **DB Snapshots**
  - User initiated from AWS Mngt. Console or using API call
  - DB Snapshot Commnad

RDS Multi AZ

- **Production Version** (Parallel Environment)
  - , Disaster Recovery (**DR**), Continuation of Business (COB)
  - **RDS Read Replica** can be also given for primary DB Source
    - MySQL, MariaDB Postgres SQL
- ~~**DB Dev / Test**~~
- ~~**Free Tier**~~

RDS Encryption Resources

- Instances are encrypted by AES-256 algorithm
  Advanced Encryption Standard (**AES**)
  - Used by gamers

RDS – Benefits and features

- Easy to administer
- Available and durable
- Secure
- Higly scalable
- Fast
- Inexpensive

Use Cases

- Web and Mobile applications
- Mobile and Online Games
- E-Commerce Applications

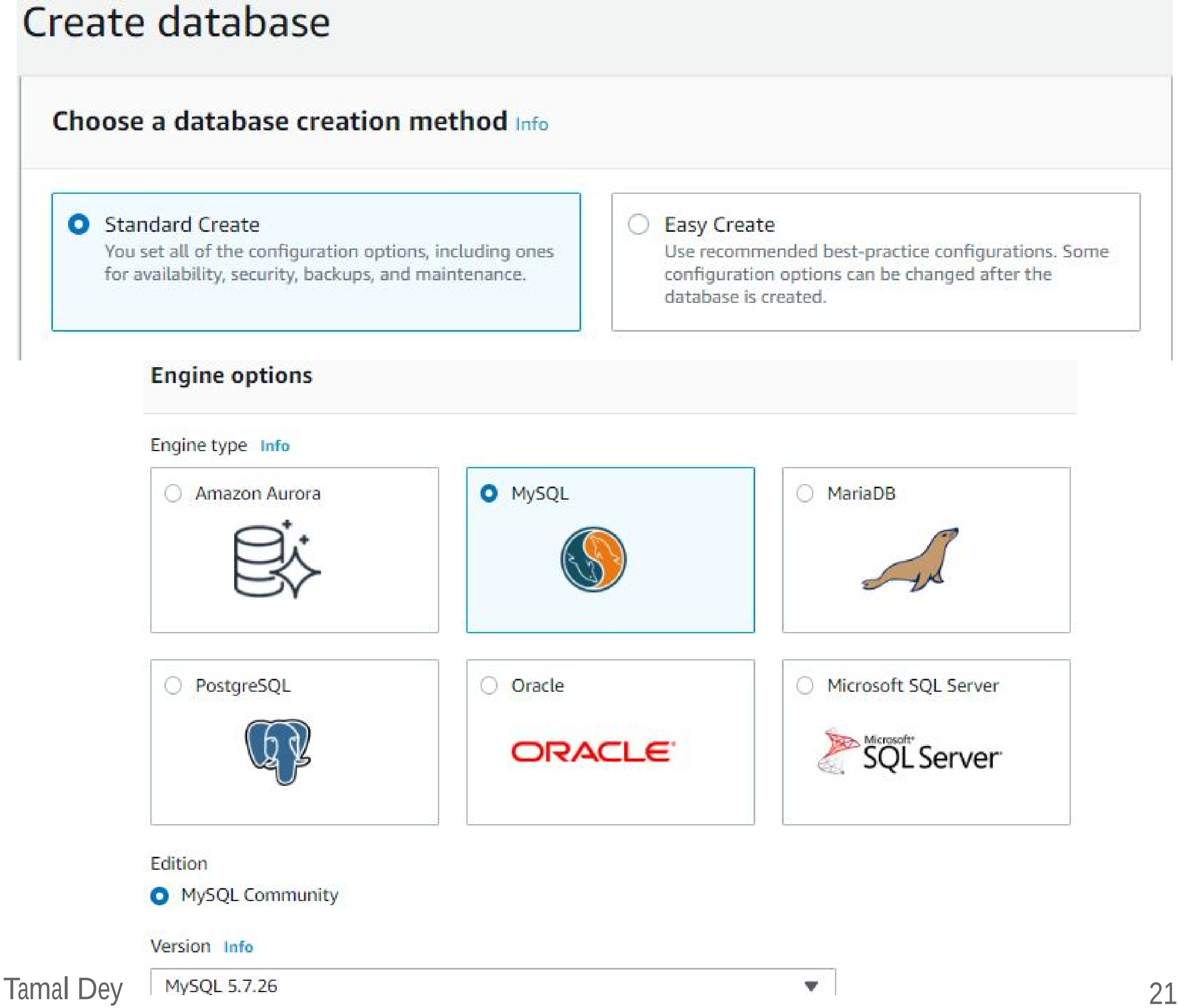
Getting started with Amazon-RDS

- Step 1 – Go to AWS console and Search RDS->Create Database
- Step 2 - Launch an Instance
- Step 3 – Authorize access
- Step 4 – Connect from the client

Getting started with Amazon-RDS

- Step 1 – Go to AWS console
- Step 2 - Launch an Instance
- Step 3 – Authorize access
- Step 4 – Connect from the client

Step 1: Select Engine



Create database

Choose a database creation method Info

● Standard Create
You set all of the configuration options, including ones for availability, security, backups, and maintenance.

○ Easy Create
Use recommended best-practice configurations. Some configuration options can be changed after the database is created.

Engine options

Engine type Info

○ Amazon Aurora

● MySQL

○ MariaDB

○ PostgreSQL

○ Oracle

○ Microsoft SQL Server

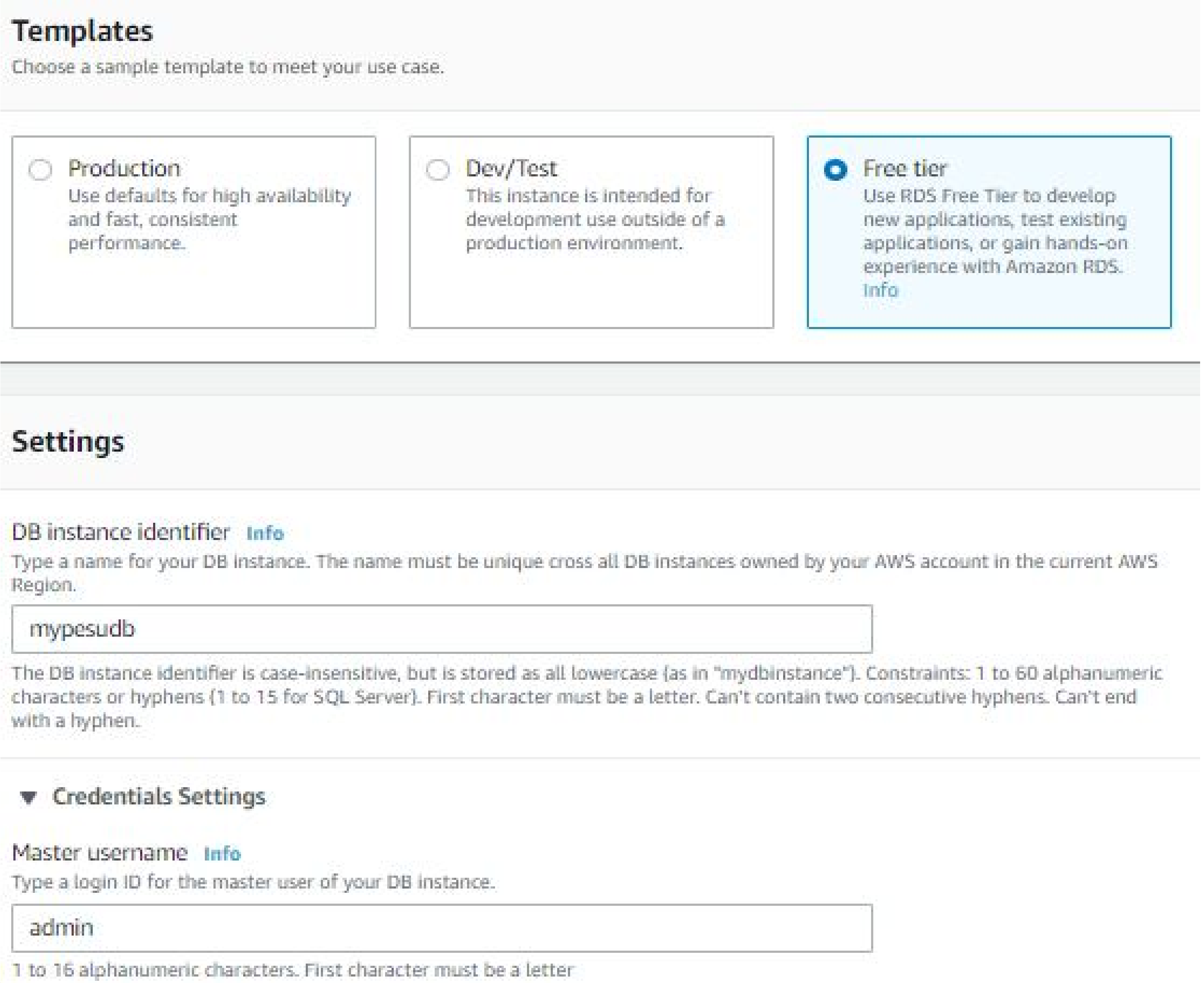
Edition

● MySQL Community

Version Info

MySQL 5.7.26

Step 2: Choose Use-Case



Templates

Choose a sample template to meet your use case.

○ Production
Use defaults for high availability and fast, consistent performance.

○ Dev/Test
This instance is intended for development use outside of a production environment.

● Free tier
Use RDS Free Tier to develop new applications, test existing applications, or gain hands-on experience with Amazon RDS.
Info

Settings

DB instance identifier Info
Type a name for your DB instance. The name must be unique cross all DB instances owned by your AWS account in the current AWS Region.

mypesudb

The DB instance identifier is case-insensitive, but is stored as all lowercase (as in "mydbinstance"). Constraints: 1 to 60 alphanumeric characters or hyphens (1 to 15 for SQL Server). First character must be a letter. Can't contain two consecutive hyphens. Can't end with a hyphen.

▼ Credentials Settings

Master username Info
Type a login ID for the master user of your DB instance.

admin

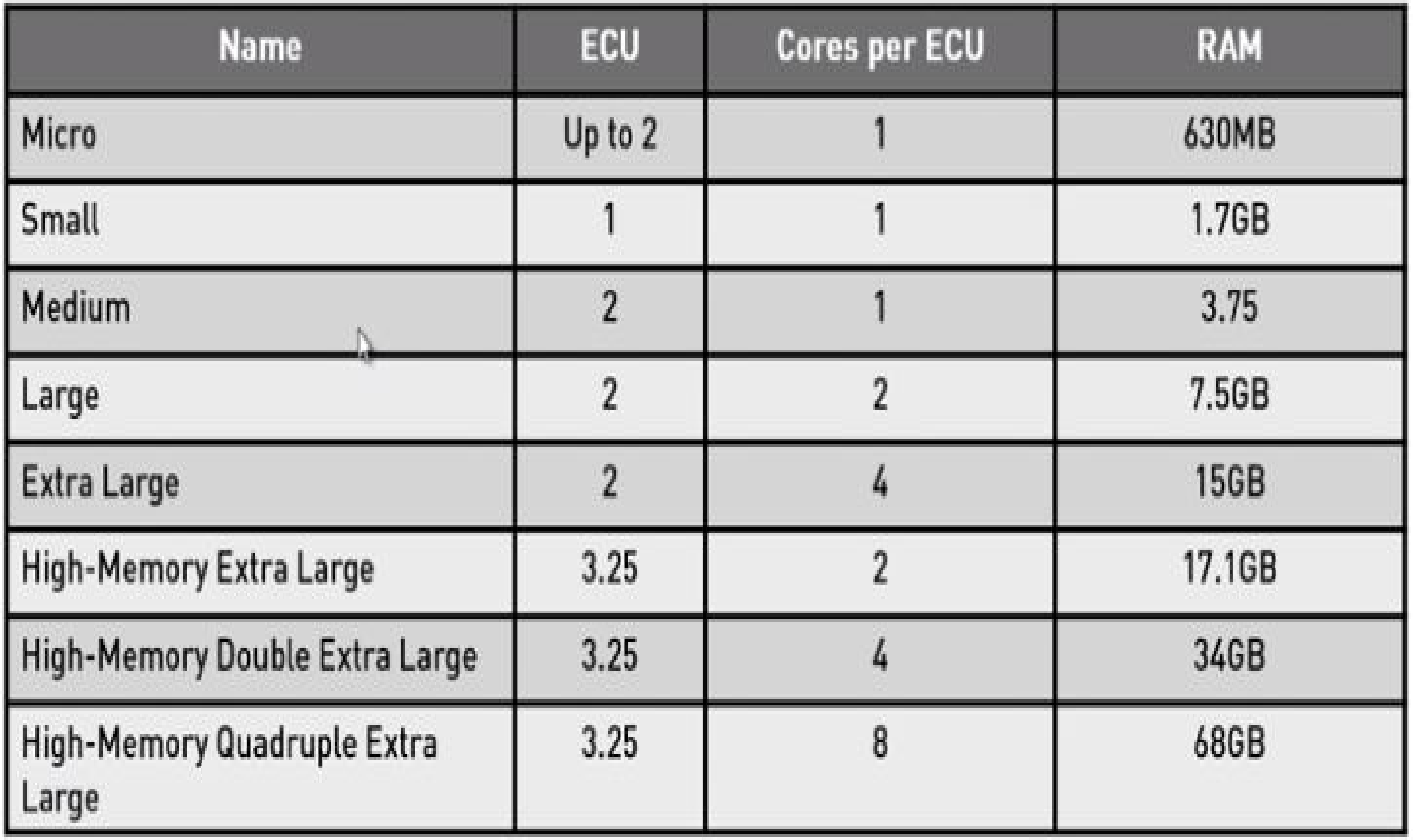
1 to 16 alphanumeric characters. First character must be a letter

Step 3: Specify db details

- Db instance – **t2 micro**
- Multi-AZ –Yes/No as required
- Specify settings
  - Db instance identifier
  - Username and password

RDS Instance Types

Elastic Compute Unit (ECU)

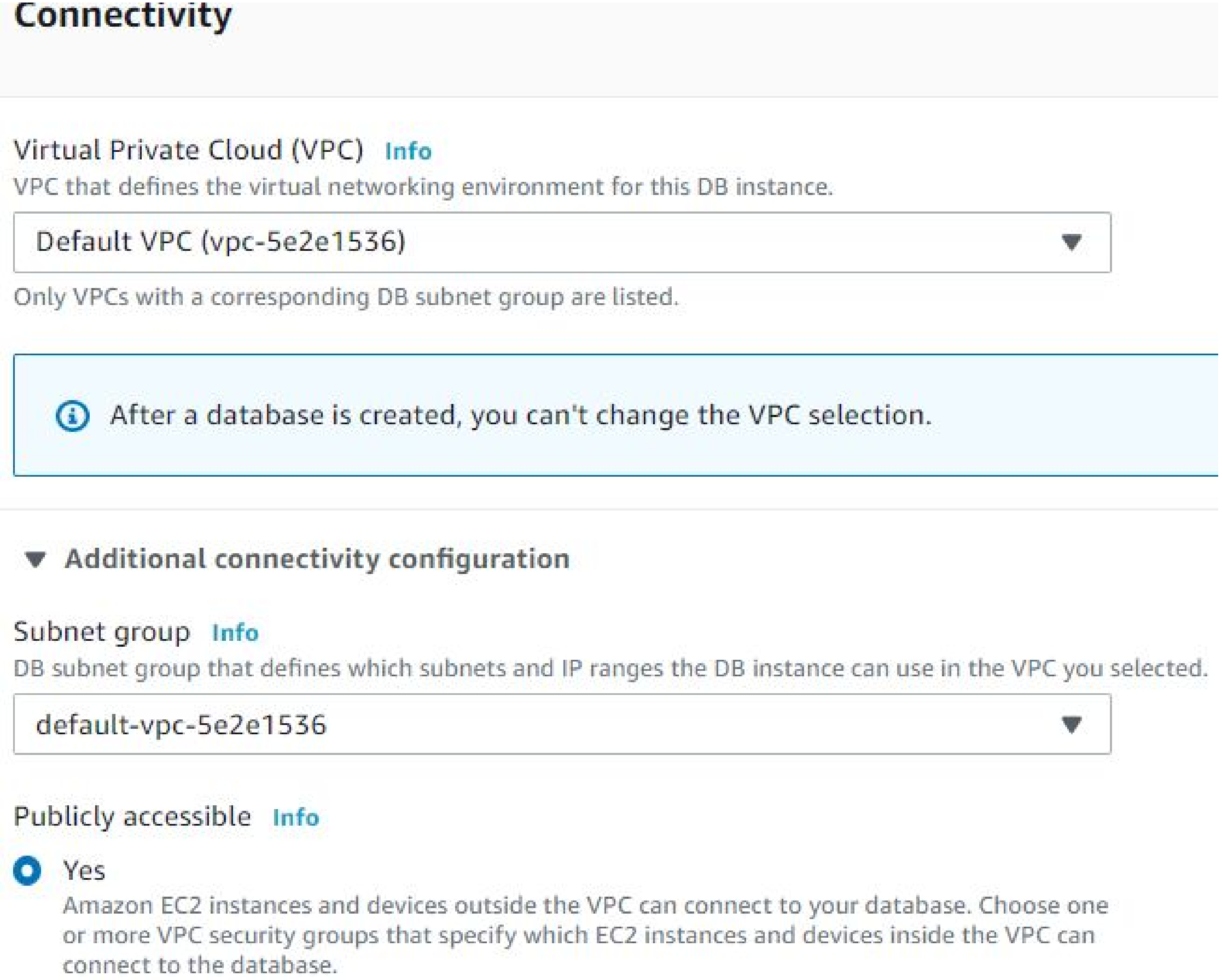


Name	ECU	Cores per ECU	RAM
Micro	Up to 2	1	630MB
Small	1	1	1.7GB
Medium	2	1	3.75
Large	2	2	7.5GB
Extra Large	2	4	15GB
High-Memory Extra Large	3.25	2	17.1GB
High-Memory Double Extra Large	3.25	4	34GB
High-Memory Quadruple Extra Large	3.25	8	68GB

Connectivity

- Connectivity->Additional connectivity configuration
- Choose default VPC and Subnet Group
- Choose existing- **Default** Security Group (Refer Next Slide)
- Choose default availability zone and port number

- Note: **Publicly accessible (yes)**
  - Amazon EC2 instances and devices outside the VPC can connect to your database. Choose one or more VPC security groups that specify which EC2 instances and devices inside the VPC can connect to the database.



Connectivity

Virtual Private Cloud (VPC)

VPC that defines the virtual networking environment for this DB instance.

Default VPC (vpc-5e2e1536) ▼

Only VPCs with a corresponding DB subnet group are listed.

ⓘ After a database is created, you can't change the VPC selection.

▼ Additional connectivity configuration

Subnet group

DB subnet group that defines which subnets and IP ranges the DB instance can use in the VPC you selected.

default-vpc-5e2e1536 ▼

Publicly accessible

● Yes

Amazon EC2 instances and devices outside the VPC can connect to your database. Choose one or more VPC security groups that specify which EC2 instances and devices inside the VPC can connect to the database.

Step 4: Configure advanced settings

- Choose retention period
- Choose public access required/not
- Choose sample db name

- Click launch the instance
- After launching wait until the endpoint is created. End point is like the DNS name in EC2.

Perform following Operations

- Take Snapshots
- Create Read Replica (Backup)
- Create Aura read replica
- Restore to point in time
- Delete the instances (Must)
- Migrate Snapshot
- Start / Stop / Reboot

Getting started with Amazon-RDS

- Step 1 – Go to AWS console
- Step 2 - Launch an Instance
- Step 3 – Authorize access
- Step 4 – Connect from the client

Authorize access

- Go to the **security group** and check for the inbound and outbound details
- In inbound -> add new rule if you want to connect through ec2 instance also (Protocol must be MySQL/Aurora) and CIDR/IP Address choose security group of your EC2 instance.
- Click "Save" to save your changes.
- In out bound check whether the all traffic is enabled

- In Amazon EC2-Linux-sudo yum update and sudo yum install mysql
- In Ubuntu EC2-sudo apt update and sudo apt install mysql-client

Connect from EC2-Linux/Ubuntu

- mysql -h <end point of DB instance> -u root/admin -p
- mysql -h mypesudb.cdbwue56chbe.ap-south-1.rds.amazonaws.com -u admin -p

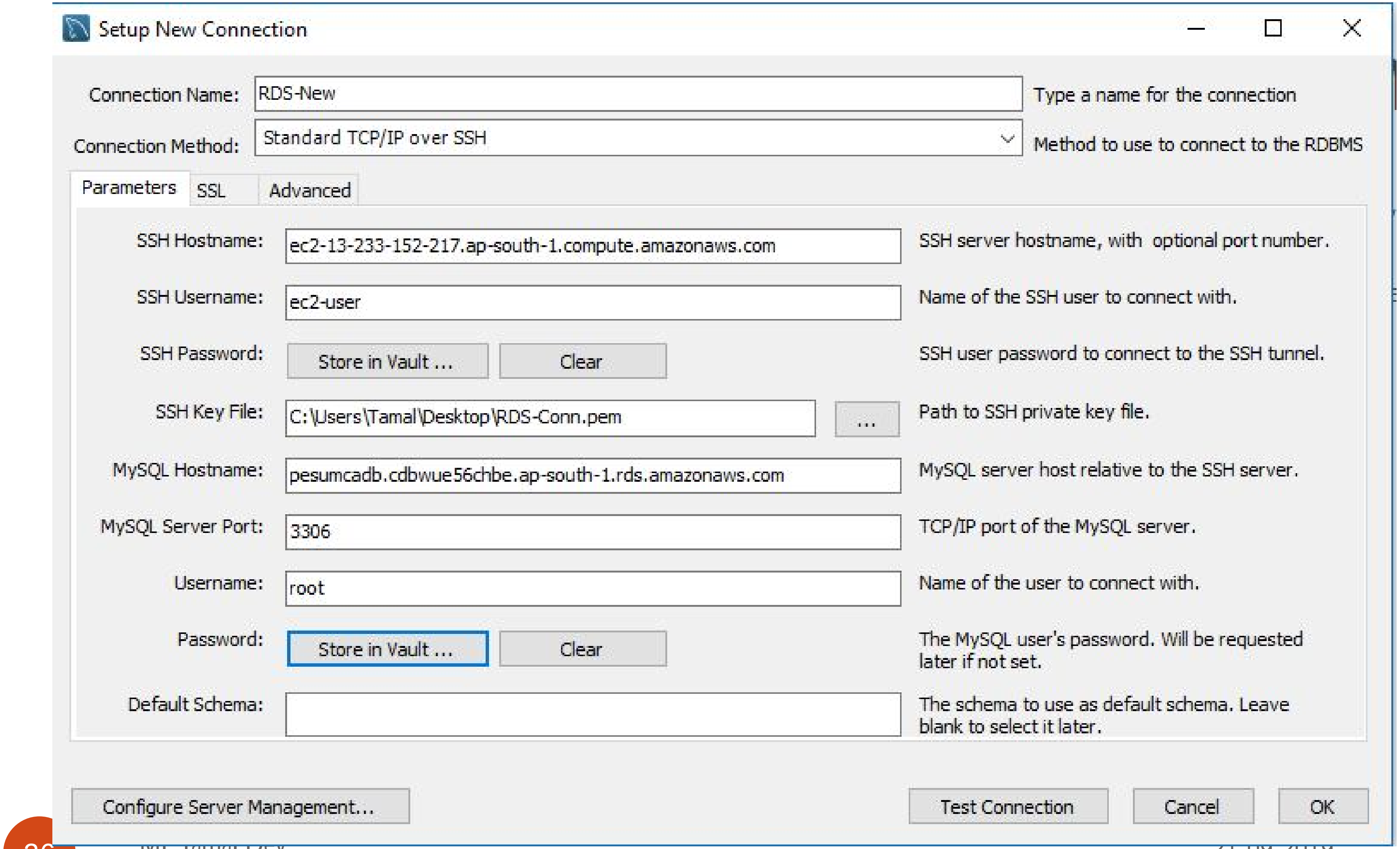
Getting started with Amazon-RDS

- Step 1 – Go to AWS console
- Step 2 - Launch an Instance
- Step 3 – Authorize access
- Step 4 – Connect from the client

Connect to the RDS

- Download and install mysql workbench (relevant for your operating system)
- Open mysql work bench
- Click on the + button to add a new mysql connection
  - Give a connection name of your choice
  - Connection method : **standard TCP / IP over SSH**
  - Fill in the values as described below which illustrated in next slide

Choose TCP/IP over SSH



Setup New Connection

Connection Name: RDS-New — Type a name for the connection

Connection Method: Standard TCP/IP over SSH — Method to use to connect to the RDBMS

SSH Hostname: ec2-13-233-152-217.ap-south-1.compute.amazonaws.com — SSH server hostname, with optional port number.

SSH Username: ec2-user — Name of the SSH user to connect with.

SSH Password: Store in Vault ... Clear — SSH user password to connect to the SSH tunnel.

SSH Key File: C:\Users\Tamal\Desktop\RDS-Conn.pem ... — Path to SSH private key file.

MySQL Hostname: pesumcadb.cdbwue56chbe.ap-south-1.rds.amazonaws.com — MySQL server host relative to the SSH server.

MySQL Server Port: 3306 — TCP/IP port of the MySQL server.

Username: root — Name of the user to connect with.

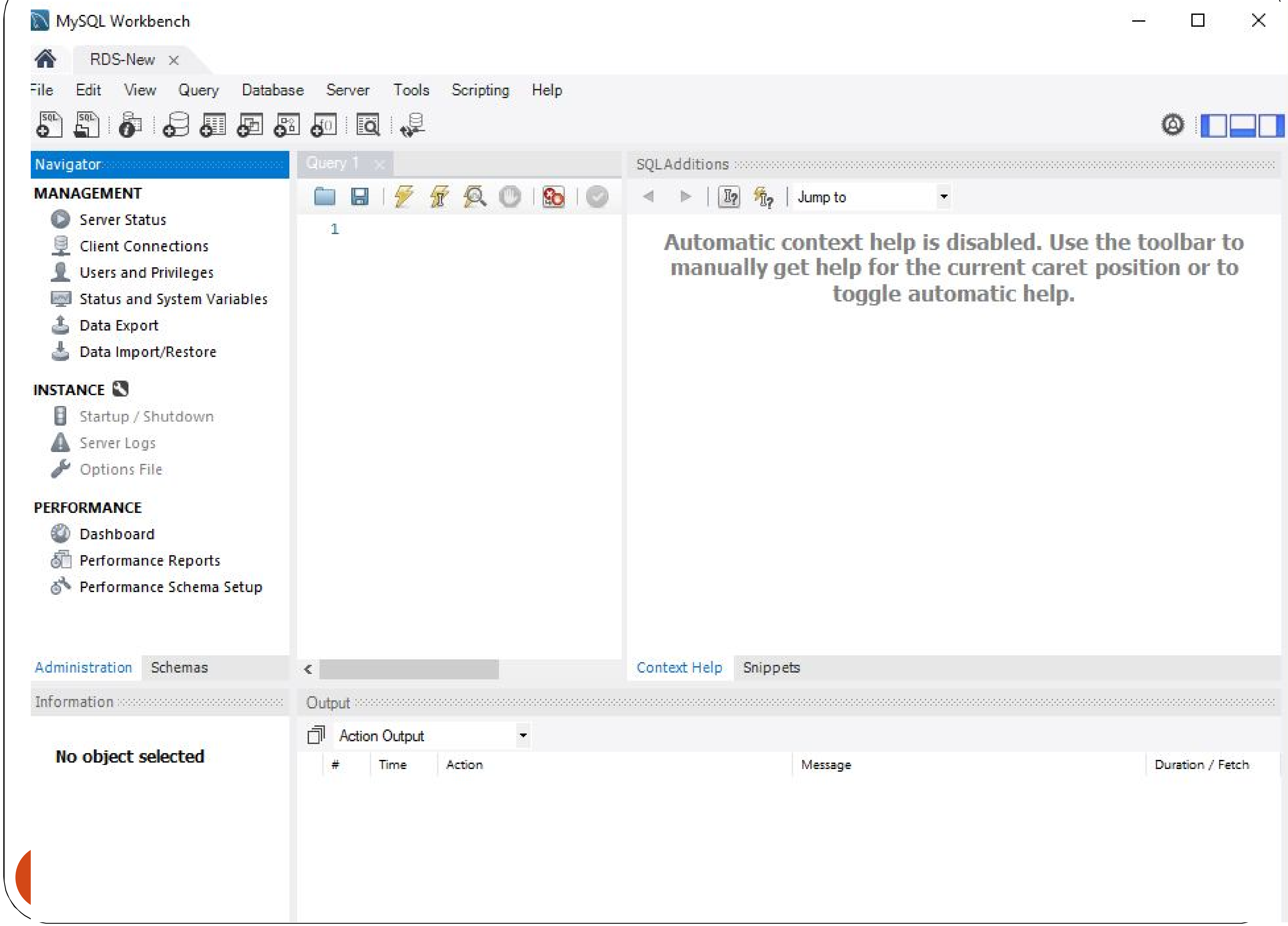
Password: Store in Vault ... Clear — The MySQL user's password. Will be requested later if not set.

Default Schema: — The schema to use as default schema. Leave blank to select it later.

Explanation-Previous Slide

- **Connection Method** – Select Standard TCP/IP over SSH from the drop down list instead of Standard(TCP/IP). Then on the appearing two tabbed pane, fill the following values found under Parameters tab.
- **SSH Hostname** – Provide the Public DNS of the Amazon EC2 instace (refer Figure 3) which will be used as the intermediate server instance used to create the connection with the DB instance.
- **SSH Username** – Provide the user of the Amazon EC2 instance (refer Figure 3) which will be used as the intermediate server instance used to create the connection with the DB instance.
- **SSH Key File** – Provide the Private Key (xxx.pem) used to connect Amazon EC2 instace via SSH
- **MySQL Hostname** – Provide the Endpoint of the DB instance created in Amazon RDS(refer Figure 5).
- **Username** – Provide the Master Username of the DB instance created in Amazon RDS(refer Figure 5). Connection Name – Provide a name to identify your connection
- **Password** – Click on Store in Keyvault. button and type the password provided while creating the DB instance in Amazon RDS.

Check you connection. Wait for some time to get connected.



MANAGEMENT

- Server Status
- Client Connections
- Users and Privileges
- Status and System Variables
- Data Export
- Data Import/Restore

INSTANCE

- Startup / Shutdown
- Server Logs
- Options File

PERFORMANCE

- Dashboard
- Performance Reports
- Performance Schema Setup

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

No object selected

#	Time	Action	Message	Duration / Fetch

For Ubuntu users

- **sudo apt install mysql-workbench**
- Click on the + button to add a new mysql connection
  - Give a connection name of your choice
  - Connection method : standard TCP/IP over SSH
  - Hostname: EC2 instance name
  - Username: ubuntu password: <leave empty>
  - Browse for key file and upload it
  - Hostname: End point of RDS
  - Username: as what you have specified in the db instance
  - Password: as what you have specified in the db instance
  - Click ok.
  - Check you connection. Wait for some time to get connected.