

## **HOSTING WORDPRESS WEBSITE USING ELASTIC BEANSTALK**

## **ABSTRACT**

The AWS Cloud provides a broad set of infrastructure services, such as computing power, storage options, networking and databases that are delivered as a utility: on-demand, available in seconds, with pay-as-you-go pricing. From data warehousing to deployment tools, directories to content delivery, over 90 AWS services are available.

In our project, we are using EC2 ,RDS and elastic beanstalk in tandem to launch and host a wordpress powered website. EC2 is used to host a PHP server running linux. RDS is used to deploy Mysql instance to store data. Elastic beanstalk is responsible for managing both of these services and keep the website active.

## 1.INTRODUCTION

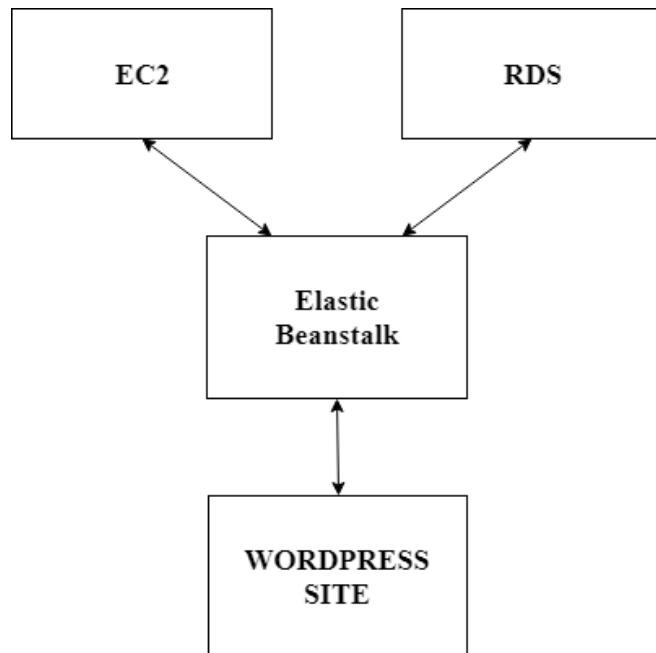
AWS Elastic Beanstalk is an orchestration service offered from Amazon Web Services for deploying infrastructure which orchestrates various AWS services, including EC2, S3, Simple Notification Service (SNS), CloudWatch, autoscaling, and Elastic Load Balancers. Elastic Beanstalk provides an additional layer of abstraction over the bare server and OS; users instead see a pre-built combination of OS and platform, such as "64bit Amazon Linux 2014.03 v1.1.0 running Ruby 2.0 (Puma)" or "64bit Debian jessie v2.0.7 running Python 3.4 (Preconfigured - Docker)". Deployment requires a number of components to be defined: an 'application' as a logical container for the project, a 'version' which is a deployable build of the application executable, a 'configuration template' that contains configuration information for both the Beanstalk environment and for the product. Finally an 'environment' combines a 'version' with a 'configuration' and deploys them.

In our project we use RDS to launch and a Mysql database which is to store and retrieve the various data required in the wordpress site. EC2 is required to launch a Linux PHP server on which our wordpress website will be installed and run on.

Elastic Beanstalk plays a very crucial role in our project it is used to maintain and deploy both the server and the database to make sure that our website is up and running under any circumstances.

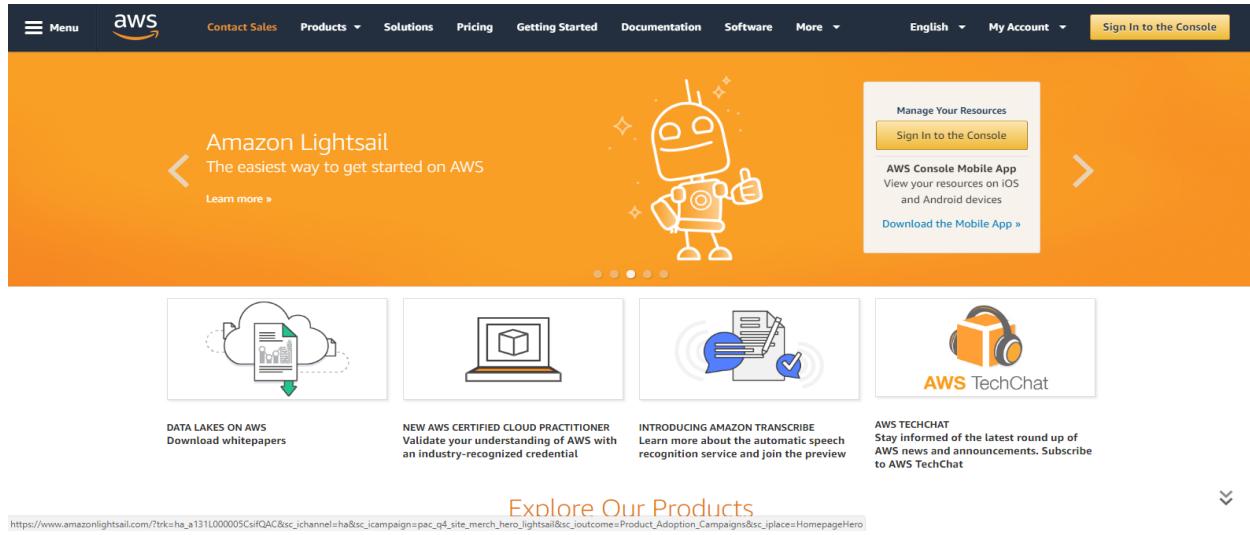
EC2 is also capable of load balancing. it is capable of varying the load on the server to ensure smooth usage .it deploys 30% of updates at one time instead of 100% so as to stop the server from being unreachable.

## 2.BLOCK DIAGRAM



### 3.GETTING STARTED WITH AMAZON WEB SERVICES

**Step 1 :** We login into Amazon web services account through Amazon portal, link is as follows. <https://aws.amazon.com/> . Using mail id and password.



**Step 2:** Select the EC2 service and choose the Amazon linux AMI 2017.09.1 (HVM),Amazon machine image

The screenshot shows the "Choose an Amazon Machine Image (AMI)" step of the EC2 instance wizard. The user is on the "Quick Start" tab, which lists several AMIs. The first item is "Amazon Linux AMI 2017.09.1 (HVM), SSD Volume Type - ami-cb9ec1b1", which is highlighted with a blue border. To its right is a "Select" button and a "64-bit" link. Below it are other options: "Amazon Linux 2 LTS Candidate AMI 2017.12.0 (HVM), SSD Volume Type - ami-2452275e" (64-bit), "SUSE Linux Enterprise Server 12 SP3 (HVM), SSD Volume Type - ami-a03869da" (64-bit), and "Red Hat Enterprise Linux 7.4 (HVM), SSD Volume Type - ami-26ebbc5c" (64-bit). At the bottom of the list is a "Default" option. The top navigation bar shows the user is in the EC2 Manager section, and the address bar shows the URL: https://console.aws.amazon.com/ec2/v2/home?region=us-east-1#LaunchInstanceWizard. The bottom of the screen shows the Windows taskbar with various icons and the system tray.

## Step 3: Select general purpose t2.micro instance type

The screenshot shows the AWS EC2 Launch Instance Wizard at Step 2: Choose an Instance Type. The user has selected the 'All instance types' filter and is viewing the 'Current generation' section. The table lists various instance types under the 'General purpose' family. The **t2.micro** instance is highlighted with a green border, indicating it is free-tier eligible. Other visible instances include t2.nano, t2.small, t2.medium, t2.large, t2.xlarge, and t2.2xlarge. The table columns include Family, Type, vCPUs, Memory (GiB), Instance Storage (GiB), EBS-Optimized Available, Network Performance, and IPv6 Support.

## Step 4: keep the default settings

The screenshot shows the AWS EC2 Launch Instance Wizard at Step 3: Configure Instance Details. The user has selected the 'Request Spot Instances' purchasing option. The configuration includes setting the number of instances to 1, choosing the vpc-1ed5b266 (default) network, and selecting the 'Shared - Run a shared hardware instance' tenancy. Other settings like IAM role, shutdown behavior, and monitoring are set to their defaults. The 'T2 Unlimited' option is also selected. The 'Review and Launch' button is visible at the bottom.

## Step 5: Allot the required storage

Step 4: Add Storage

Your instance will be launched with the following storage device settings. You can attach additional EBS volumes and instance store volumes to your instance, or edit the settings of the root volume. You can also attach additional EBS volumes after launching an instance, but not instance store volumes. [Learn more](#) about storage options in Amazon EC2.

Volume Type	Device	Snapshot	Size (GiB)	Volume Type	IOPS	Throughput (MB/s)	Delete on Termination	Encrypted
Root	/dev/xvda	snap-0e6d1e06d131ff774	8	General Purpose SSD (GP2)	100 / 3000	N/A	<input checked="" type="checkbox"/>	Not Encrypted

Free tier eligible customers can get up to 30 GB of EBS General Purpose (SSD) or Magnetic storage. [Learn more](#) about free usage tier eligibility and usage restrictions.

[Cancel](#) [Previous](#) [Review and Launch](#) [Next: Add Tags](#)

## Step 6: add the required tags

Step 5: Add Tags

A tag consists of a case-sensitive key-value pair. For example, you could define a tag with key = Name and value = Webserver. A copy of a tag can be applied to volumes, instances or both. Tags will be applied to all instances and volumes. [Learn more](#) about tagging your Amazon EC2 resources.

Key	(127 characters maximum)	Value	(255 characters maximum)	Instances	Volumes
ABCD		1234		<input checked="" type="checkbox"/>	<input type="checkbox"/>

Add another tag (Up to 50 tags maximum)

[Cancel](#) [Previous](#) [Review and Launch](#) [Next: Configure Security Group](#)

## Step 7: configure the security groups as defaults

The screenshot shows the AWS EC2 Launch Instance Wizard at Step 6: Configure Security Group. The page title is "Step 6: Configure Security Group". A warning message states: "A security group is a set of firewall rules that control the traffic for your instance. On this page, you can add rules to allow specific traffic to reach your instance. For example, if you want to set up a web server and allow Internet traffic to reach your instance, add rules that allow unrestricted access to the HTTP and HTTPS ports. You can create a new security group or select from an existing one below. Learn more about Amazon EC2 security groups." Below this, there are two options: "Create a new security group" (radio button selected) and "Select an existing security group". The "Security group name" field contains "launch-wizard-4" and the "Description" field contains "launch-wizard-4 created 2018-01-09T14:46:07.256+05:30". A table lists a single rule: Type: SSH, Protocol: TCP, Port Range: 22, Source: Custom 0.0.0.0/0, Description: e.g. SSH for Admin Desktop. A warning box at the bottom says: "Warning: 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." At the bottom right are "Cancel", "Previous", and "Review and Launch" buttons.

## Step 8: review the instance detail before launching

The screenshot shows the AWS EC2 Launch Instance Wizard at Step 7: Review Instance Launch. The page title is "Step 7: Review Instance Launch". The summary includes:

- AMI Details:** Amazon Linux AMI 2017.09.1 (HVM), SSD Volume Type - ami-cb9ec1b1. It is a Free tier eligible image.
- Instance Type:** t2.micro (Variable, 1 vCPU, 1 GiB Memory, EBS only storage, EBS-Optimized Available, Low to Moderate Network Performance).
- Security Groups:** launch-wizard-4 (launched 2018-01-09T14:46:07.256+05:30). A table shows a rule: Type: SSH, Protocol: TCP, Port Range: 22, Source: 0.0.0.0/0.
- Instance Details:** (Edit instance details)

At the bottom right are "Cancel", "Previous", and "Launch" buttons.

## Step 9: download the key pair for the instance for the future use

The screenshot shows the AWS EC2 Launch Instance Wizard at Step 7: Review Instance Launch. A modal window titled "Select an existing key pair or create a new key pair" is displayed. Inside the modal, there is a dropdown menu set to "Select a key pair" with an option "bibo\_10\_12\_17" listed. Below the dropdown is a checkbox that reads: "I acknowledge that I have access to the selected private key file (bibo\_10\_12\_17.pem), and that without this file, I won't be able to log into my instance." At the bottom of the modal are two buttons: "Cancel" and "Launch Instances". The background of the main EC2 wizard shows the instance configuration details, including the AMI (Amazon Linux AMI 2017.09.1 (HVM)), instance type (t2.micro), and security group (launch-wizard-4). The status bar at the bottom indicates the date and time as 09-01-2018 02:48 PM.

## Step 10: EC2 instance is launching

The screenshot shows the AWS EC2 Launch Instance Wizard at Step 10: Launch Status. A green success message box states "Your instances are now launching" and provides a link to "View launch log". Below this, a blue info message box discusses "Get notified of estimated charges" and encourages creating billing alerts. The main content area contains several sections: "How to connect to your instances" (explaining the instance state and connection steps), "Helpful resources" (listing the Amazon EC2 User Guide and Discussion Forum), and "While your instances are launching you can also" (mentioning status check alarms, EBS volumes, and security groups). The status bar at the bottom of the browser window shows the date and time as 09-01-2018 02:49 PM.

## Step 11: EC2 instance is up and running

The screenshot shows the AWS EC2 Instances page. On the left, there's a sidebar with navigation links like EC2 Dashboard, Events, Tags, Reports, Limits, Instances, Launch Templates, Spot Requests, Reserved Instances, Dedicated Hosts, Scheduled Instances, Images, AMIs, Bundle Tasks, Elastic Block Store, Volumes, Snapshots, Network & Security, Security Groups, and Elastic IPs. The main content area has tabs for Launch Instance, Connect, and Actions. A search bar at the top says "Filter by tags and attributes or search by keyword". Below it is a table with columns: Name, Instance ID, Instance Type, Availability Zone, Instance State, Status Checks, Alarm Status, Public DNS (IPv4), IPv4 Public IP, and IPv6 IP. There are two rows: one for a terminated instance (wordpressNew-env) and one for a running instance (i-097d63b293576228a). The running instance's details are expanded, showing fields like Instance ID (i-04bae9a503b109c9d), Instance state (terminated), Instance type (t2.micro), Availability zone (us-east-1b), Security groups (-), Scheduled events (-), AMI ID (aws-elasticbeanstalk-amzn-2017.09.1.x86\_64-phpp71-hvm-201801051751 (ami-9bcd9de1)), Platform (-), IAM role (-), Key pair name (-), Public DNS (IPv4) (ec2-54-204-98-227.co...), IPv4 Public IP (54.204.98.227), IPv6 IP (-), Private DNS (-), Secondary private IPs (-), VPC ID (-), Subnet ID (-), Network interfaces (-), Source/dest. check (False), T2 Unlimited (Disabled), and Owner (424179511380).

## Step 12: creating Amazon RDS instance, click on launch DB instance

The screenshot shows the AWS RDS Instances page. On the left, there's a sidebar with links for Dashboard, Instances (which is selected and highlighted in orange), Clusters, Performance Insights, Snapshots, Reserved instances, Subnet groups, Parameter groups, Option groups, Events, Event subscriptions, and Notifications. The main content area has tabs for RDS and Instances. It features a "Launch DB instance" button in orange. Below it is a table with columns: Instances (0), Instance actions, Launch DB instance, and Restore from S3. A search bar says "Filter instances". The table shows a single row: DB instance, Status, CPU, Current activity, Maintenance, Class, and VPC. The status is "No instances found". At the bottom, there's a footer with links for Feedback, English (US), Privacy Policy, Terms of Use, and a system tray showing the date and time.

## Step 13: choose mysql engine from the engine option

The screenshot shows the 'Engine options' step of the AWS RDS setup wizard. It lists several database engines: Amazon Aurora, MySQL, MariaDB, PostgreSQL, Oracle, and Microsoft SQL Server. The MySQL option is selected, highlighted with a blue border and a circular icon. Below the MySQL section, there is a detailed description of MySQL's features and benefits, such as support for up to 16 TB and automated backup.

MySQL

MySQL is the most popular open source database in the world. MySQL on RDS offers the rich features of the MySQL community edition with the flexibility to easily scale compute resources or storage capacity for your database.

- Supports database size up to 16 TB.
- Instances offer up to 32 vCPUs and 244 GiB Memory.
- Supports automated backup and point-in-time recovery.
- Supports cross-region read replicas.

Only enable options eligible for RDS Free Usage Tier [info](#)

Cancel **Next**

## Step 14: select the latest DB engine version and select DB.t2.micro as the DB instance class

The screenshot shows the 'Instance specifications' step of the AWS RDS setup wizard. It specifies the DB engine as MySQL Community Edition, the license model as general-public-license, and the DB engine version as mysql 5.6.37. A 'Known Issues/Limitations' section provides information about the Free tier, stating that it offers a single db.t2.micro instance and up to 20 GB of storage. A checkbox for enabling options eligible for RDS Free Usage Tier is checked. The DB instance class is set to db.t2.micro, which is described as having 1 vCPU and 1 GiB RAM.

Step 2  
Specify DB details

Step 3  
Configure advanced settings

**Instance specifications**

Estimate your monthly costs for the DB Instance using the AWS Simple Monthly Calculator.

DB engine  
MySQL Community Edition

License model [info](#)  
general-public-license

DB engine version [info](#)  
mysql 5.6.37

**Known Issues/Limitations**  
Review the Known Issues/Limitations to learn about potential compatibility issues with specific database versions.

**Free tier**  
The Amazon RDS Free Tier provides a single db.t2.micro instance as well as up to 20 GB of storage, allowing new AWS customers to gain hands-on experience with Amazon RDS. Learn more about the RDS Free Tier and the instance restrictions [here](#).

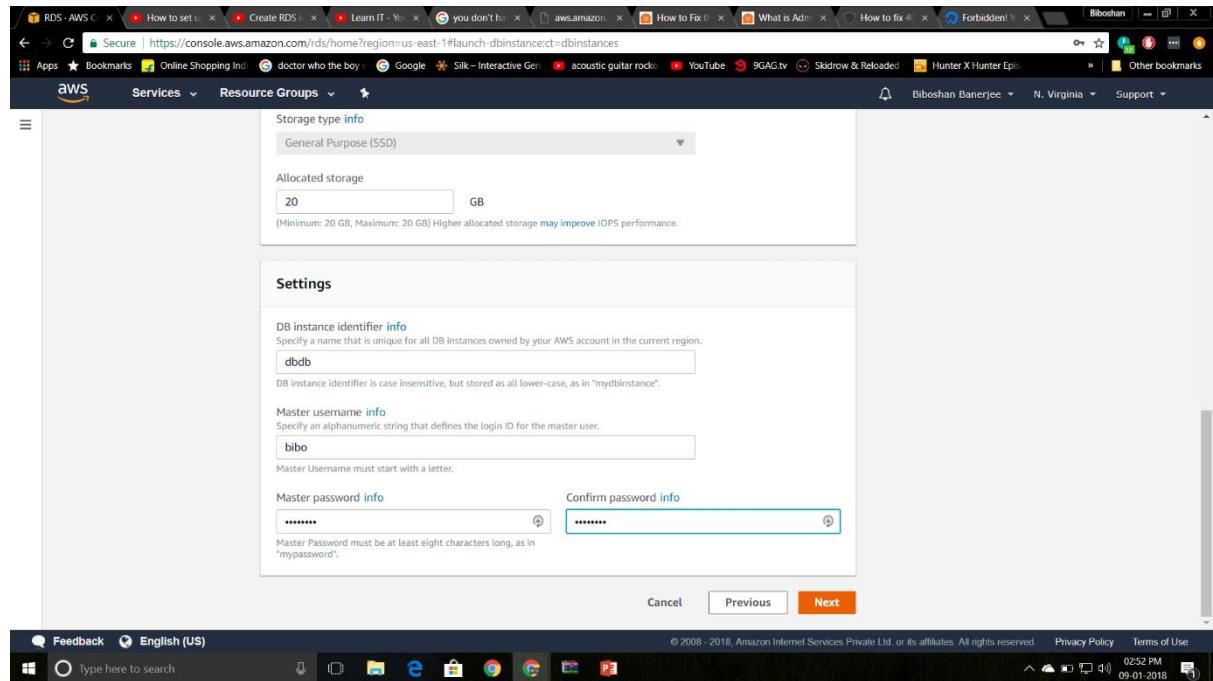
Only enable options eligible for RDS Free Usage Tier [info](#)

DB instance class [info](#)  
db.t2.micro — 1 vCPU, 1 GiB RAM

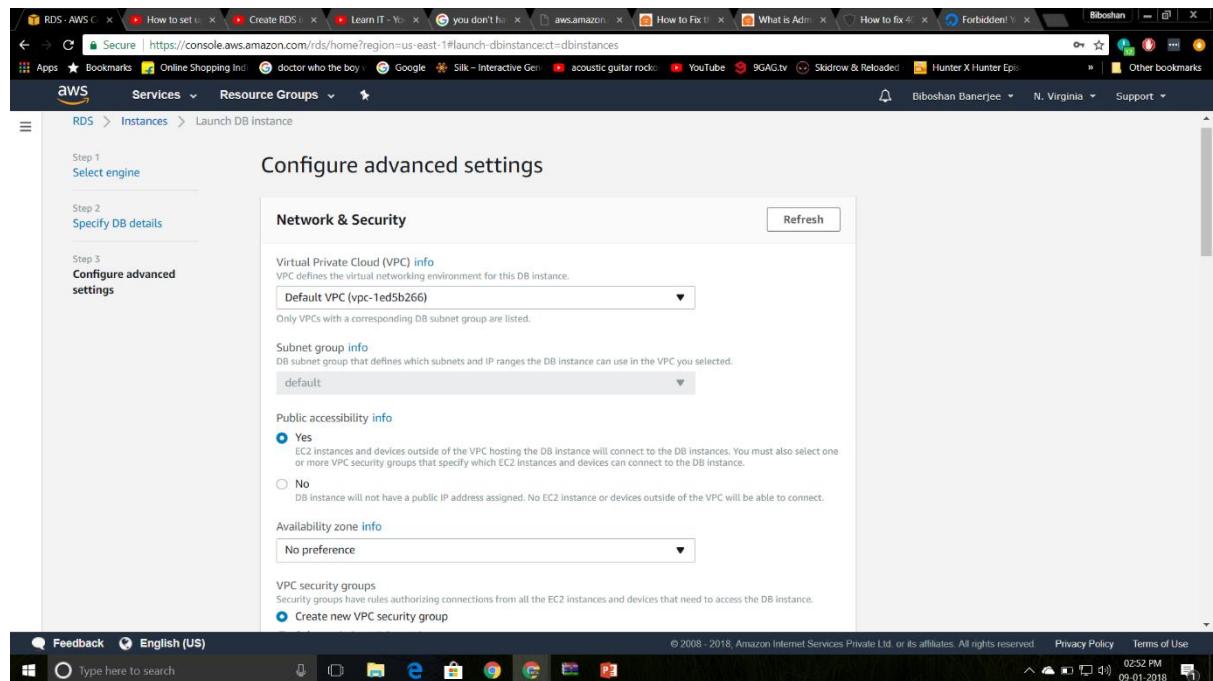
Feedback English (US)

© 2008 - 2018, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved. Privacy Policy Terms of Use

## Step 15: allocate the storage DB instance identifier ,master user name and master password



## Step 16: choose the default network and security settings



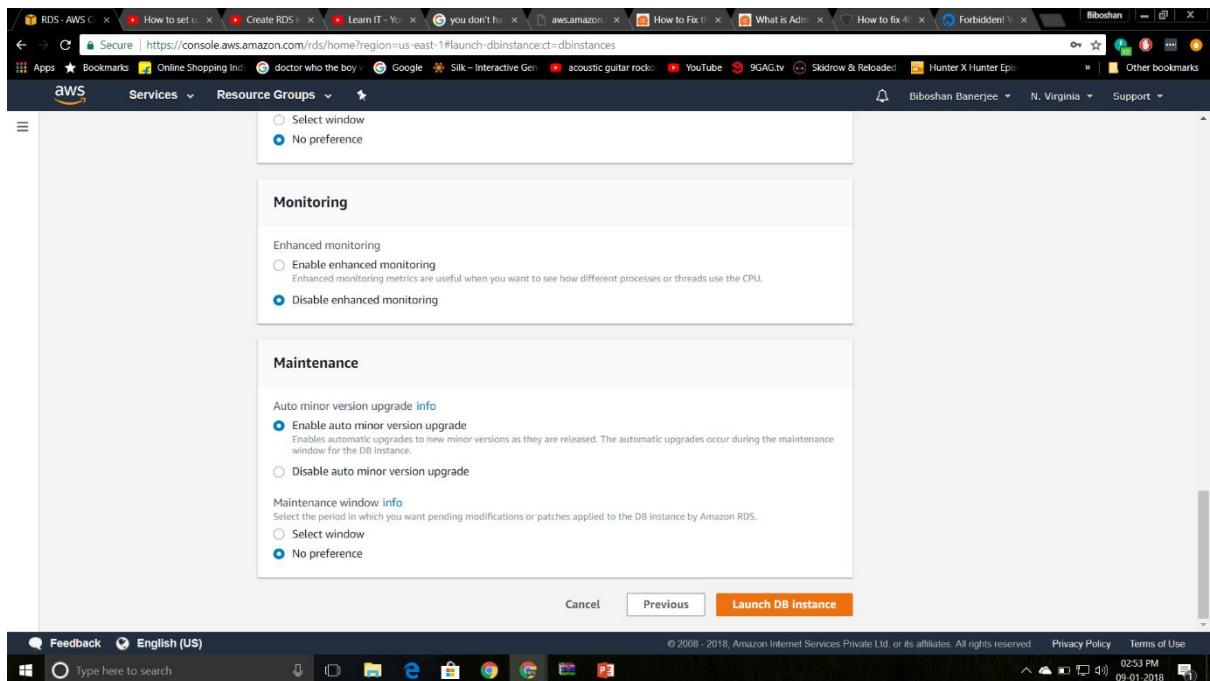
## Step 17: set the database port

The screenshot shows the AWS RDS console with the URL <https://console.aws.amazon.com/rds/home?region=us-east-1#launch-dbinstance:c=dbinstances>. The 'Services' tab is selected. In the main area, there's a 'VPC security groups' section with two options: 'Create new VPC security group' (selected) and 'Select existing VPC security groups'. Below it is the 'Database options' section, which includes a 'Database name' field containing 'dbname', a 'Database port' field containing '3306', and dropdown menus for 'DB parameter group info' (set to 'default.mysql5.6') and 'Option group info' (set to 'default:mysql-5-6'). There are also checkboxes for 'Copy tags to snapshots' and 'Enable IAM DB authentication'.

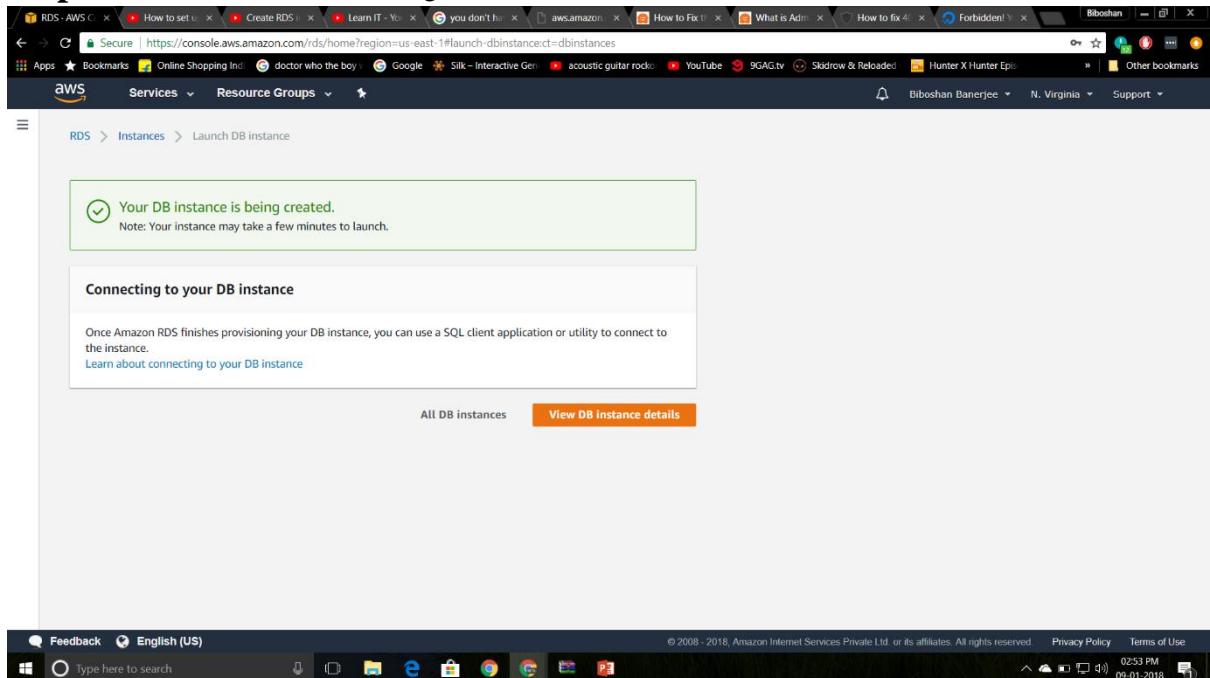
## Step 18: choose the default settings

The screenshot shows the AWS RDS console with the same URL as the previous step. The 'Encryption' section is open, showing the 'Disable Encryption' option selected. A note at the bottom states: 'The selected engine or DB instance class does not support storage encryption.' The 'Backup' section is also visible, featuring a warning message: 'Please note that automated backups are currently supported for InnoDB storage engine only. If you are using MyISAM, refer to detail [here](#)'. It includes a 'Backup retention period info' field set to '7 days' and a 'Backup window info' section where 'No preference' is selected.

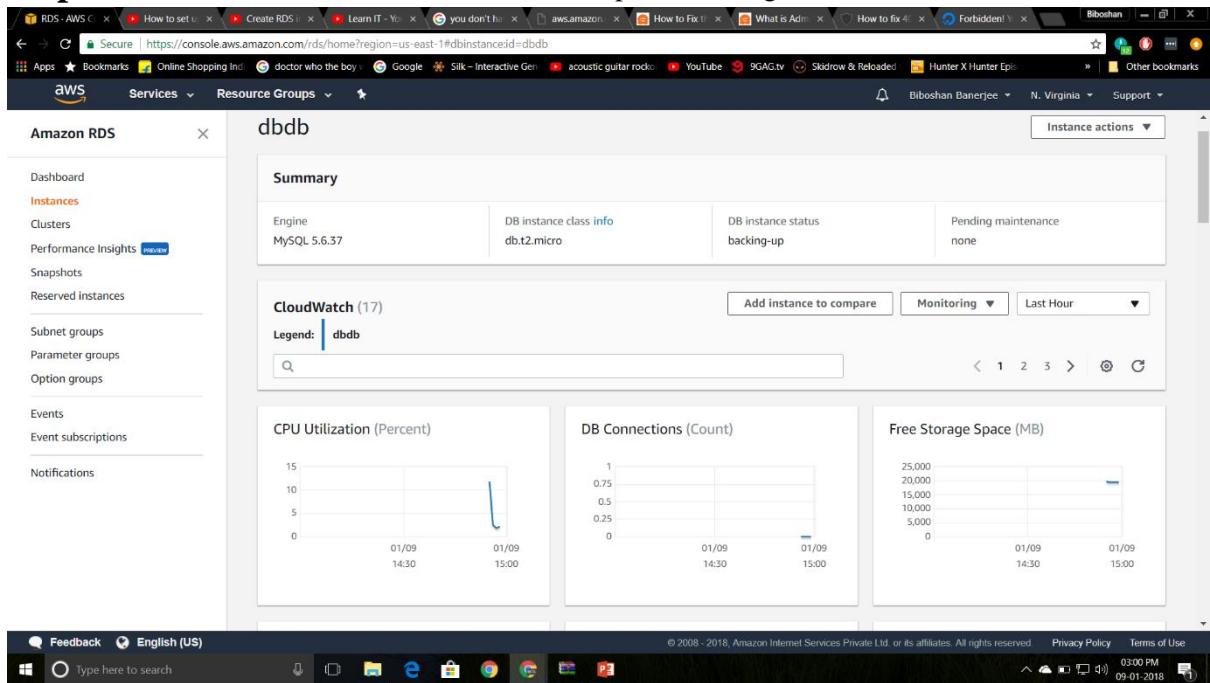
## Step 19: choose the default settings



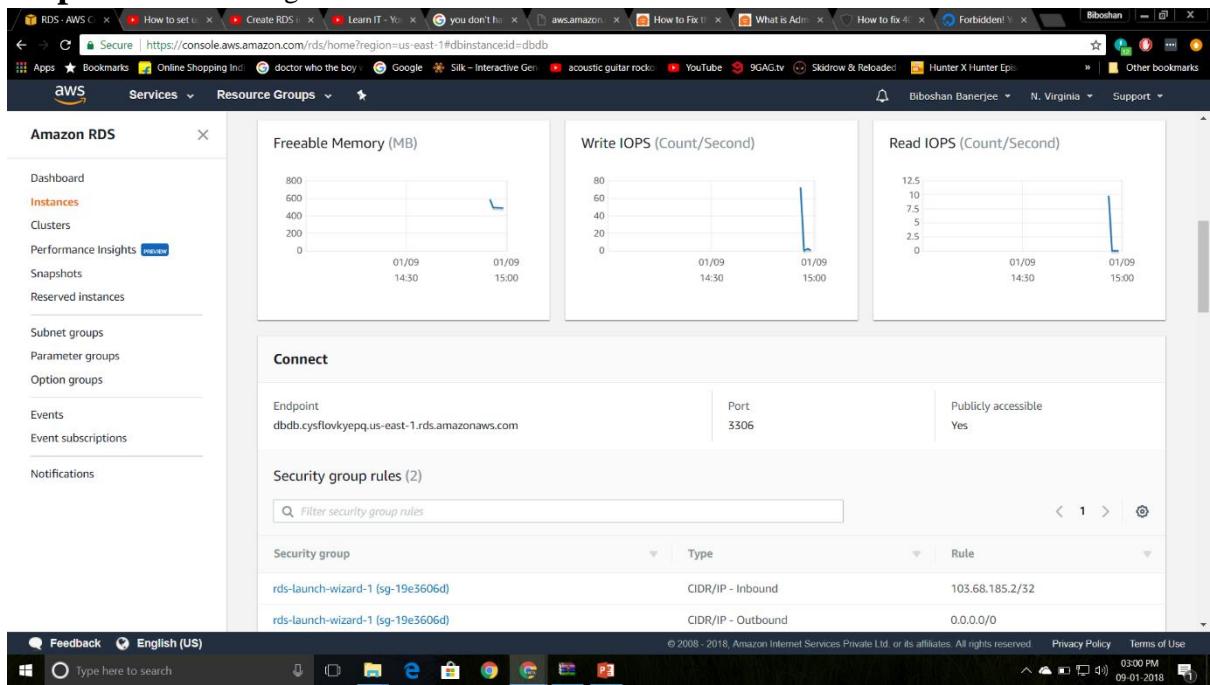
## Step 20: DB instance is launching



## Step 21: DB instance is been launched and it is up and running.



## Step 22: the below fig tells us the DB instance detail



## Step 23: the below fig tells us the DB instance detail

The screenshot shows the AWS RDS console with the following details for the database instance dbdb:

Details			
Configurations	Security and network	Instance and IOPS	Maintenance details
ARN: arnaws:rds:us-east-1:1424179511380:db:dbdb	Availability zone: us-east-1f	Instance Class: db.t2.micro	Auto minor version upgrade: Yes
Engine: MySQL 5.6.37	VPC: vpc-1ed5b266	Storage Type: General Purpose (SSD)	Maintenance window: tue:03:44-tue:04:14 UTC (GMT)
License Model: General Public License	Subnet group: default	Storage: 20 GB	Backup window: 10:23-10:53 UTC (GMT)
Created Time: Tue Jan 09 14:56:05 GMT+530 2018	Subnets: subnet-a6afc3c2, subnet-f9a5ae5, subnet-1787544a, subnet-e78e50c8, subnet-5844c307, subnet-7d66fd56	Availability and durability: DB instance status: backing-up, Multi AZ: No	Pending Modifications: None
Username: bibo	Security groups: rds-launch-wizard-1 (sg-19e3606d) (active)	Automated backups: Enabled (7 Days)	Pending maintenance: none
Option Group: default:mysql-5-6	Publicly accessible: Yes		Encryption enabled: No
Parameter group: default.mysql5.6 (in-sync)	Endpoint: dbdb.orclcloudapp.us-east-1.rds.amazonaws.com:3306		
Copy tags to snapshots: false			

## LAUNCHING ELASTIC BEANSTALK

### Step 1: choose an application name

The screenshot shows the AWS Elastic Beanstalk console with the following steps for creating a new application:

- Try the new design:** A message indicates we're testing a new design for the environment creation wizard. Opt in now to try it and let us know what you think!
- Application Info:** New Environment
- Application Information:** Application name: WordPress\_new (Must be less than 100 characters and cannot contain a /). Description: Optional.
- Next Step:** Cancel or Next

## Step 2: choose web server environment

The screenshot shows the AWS Elastic Beanstalk 'Create New Environment' wizard. The 'Web Server Environment' section is selected, displaying information about standard web servers and their port 80 processing tasks. A 'Create web server' button is visible. Below it, the 'Worker Environment' section is shown, providing details for worker applications and a 'Create worker' button. At the bottom right are 'Cancel' and 'Done' buttons.

## Step 3: select PHP as the platform

The screenshot shows the 'Environment Type' configuration screen. Under 'Environment Type', 'PHP' is selected as the predefined configuration. It notes that AWS Elastic Beanstalk will create an environment running PHP 7.1 on 64bit Amazon Linux 2017.09 v2.6.2. A 'Change platform version' link is available. The 'Environment type' dropdown shows 'Load balancing, auto scaling'. At the bottom right are 'Cancel', 'Previous', and 'Next' buttons.



## Step 4: keep the default setting

The screenshot shows the 'Application Version' configuration page. On the left, a sidebar lists options: Application Info, New Environment, Environment Type, Application Version (which is selected and highlighted in orange), Additional Resources, Configuration Details, Environment Tags, Permissions, and Review Information. The main content area is titled 'Application Version' and contains the following fields:

- Select a source for your application version:
  - Source:  Sample application
  - Upload your own ([Learn more](#))
  - Choose File No file chosen
- S3 URL:

Below this is a section titled 'Deployment Preferences' with the following settings:

- Deployment policy: Rolling [Learn more](#)
- Healthy threshold: Ok
- Ignore health check: False
- Batch size:
  - Percentage: 30 % of the fleet at a time
  - Fixed: 1 instances at a time

The status bar at the bottom indicates the date and time: 09-01-2018 03:04 PM.

## Step 5: select the application URL

The screenshot shows the 'Create New Environment' wizard. The top navigation bar includes 'Services', 'Resource Groups', and 'Create New Environment'. A message box says 'Try the new design' with the subtext 'We're testing a new design for the environment creation wizard. [Opt in now](#) to try it and let us know what you think!'. The main form is titled 'Environment Information' and contains the following fields:

- Environment name:
- Environment URL:  [Check availability](#)
- Description:  Optional: 200 character maximum

At the bottom right are 'Cancel', 'Previous', and 'Next' buttons.



## Step 6: upload the key pair in EC2 key pair location and choose the instance type as t2.micro

The screenshot shows the 'Configuration Details' section of the AWS Elastic Beanstalk console. On the left, a sidebar lists options like Application Info, New Environment, Environment Type, Application Version, Environment Info, Additional Resources, Configuration Details (which is selected), Environment Tags, RDS Configuration, Permissions, and Review Information. The main area is titled 'Configuration Details' and contains the following fields:

- Instance type:** t2.micro (selected)
- EC2 key pair:** bibo\_10\_12\_17 (selected)
- Email address:** biboshan23@gmail.com
- Application health check URL:** (empty field)
- Rolling update type:** Rolling based on Health
- Cross zone load balancing:**  (selected)
- Connection draining:**  (selected)
- Connection draining timeout:** 20 seconds

Below these settings is a 'Health Reporting' section with a 'System type:' dropdown set to 'Enhanced'.

## Step 7: keep the default settings

The screenshot shows the 'Configuration Details' section of the AWS Elastic Beanstalk console. The sidebar on the left shows the 'Configuration Details' option is still selected. The main area contains the same settings as the previous screenshot, plus a 'Root Volume (Boot Device)' section:

- Cross zone load balancing:**  (selected)
- Connection draining:**  (selected)
- Connection draining timeout:** 20 seconds
- Root Volume (Boot Device):**
  - Root volume type:** (Container default) (selected)
  - Root volume size:** 10 GiB (selected)

At the bottom right of the configuration page, there are 'Cancel', 'Previous', and 'Next' buttons. The status bar at the bottom of the browser window shows the date and time as 09-01-2018 03:05 PM.

## Step 8: add the appropriate tags for the application

The screenshot shows the AWS Elastic Beanstalk environment creation wizard at the 'Environment Tags' step. On the left, a sidebar lists options like Application Info, New Environment, Environment Type, Application Version, Environment Info, Additional Resources, Configuration Details, Environment Tags (which is selected), RDS Configuration, Permissions, and Review Information. The main area has a header 'Environment Tags' with a sub-instruction: 'You can specify tags (key-value pairs) for your Environment. You can add up to 47 unique key-value pairs for each Environment.' Below this is a table for entering tags:

Key (128 characters maximum)	Value (256 characters maximum)
1. abcd	1234
2.	

Below the table, it says '46 remaining'. At the bottom right are 'Cancel', 'Previous', and 'Next' buttons.



## Step 9: Define the RDS settings

The screenshot shows the AWS Elastic Beanstalk environment creation wizard at the 'RDS Configuration' step. The sidebar is identical to the previous step. The main area has a header 'RDS Configuration' with a sub-instruction: 'Specify your RDS settings. Learn more'. It includes fields for Snapshot (None), DB engine (mysql), DB engine version (5.6.37), Instance class (db.t2.micro), Allocated storage (5 GB), Username (bibo), Password (redacted), Retention setting (Create snapshot), and Availability (Single availability zone). A note below the retention setting says: 'Terminating your environment can permanently delete your Amazon RDS DB instance and all its data. By default, AWS Elastic Beanstalk saves a snapshot, which preserves your data but may incur backup storage charges.' At the bottom right are 'Cancel', 'Previous', and 'Next' buttons.



## Step 10: Select the default instance profile

The screenshot shows the AWS Elastic Beanstalk environment creation wizard. The current step is 'Permissions'. A message at the top says 'Try the new design' and 'We're testing a new design for the environment creation wizard. Opt in now to try it and let us know what you think!'. On the left, a sidebar lists steps: Application Info, New Environment, Environment Type, Application Version, Environment Info, Additional Resources, Configuration Details, Environment Tags, RDS Configuration, Permissions (which is selected), and Review Information. The main area is titled 'Permissions' and contains instructions: 'Select an instance profile and service role for your AWS Elastic Beanstalk environment.' It explains that an instance profile is an IAM role for EC2 instances and a service role allows the Elastic Beanstalk service to monitor resources. Below are dropdown menus for 'Instance profile:' (set to 'aws-elasticbeanstalk-ec2-role') and 'Service role:' (set to 'aws-elasticbeanstalk-service-role'). At the bottom right are 'Cancel', 'Previous', and 'Next' buttons.



## Step 11: review to check at the correct settings are been chosen

The screenshot shows the AWS Elastic Beanstalk environment creation wizard in the 'Review' step. The sidebar on the left shows steps: Application Info, New Environment, Environment Type, Application Version, Environment Info, Additional Resources, Configuration Details, Environment Tags, RDS Configuration, Permissions, and Review Information (which is selected). The main area has sections for 'Application Info' (Application name: WordPress\_new), 'New Environment' (Tier: Web Server), 'Environment Type' (Container type: 64bit Amazon Linux 2017.09 v2.6.2 running PHP 7.1, Environment type: Load balancing, auto scaling), 'Application Version' (Application source: Sample application, Deployment settings: Deploy to 30% of the fleet at a time), and 'Environment Info' (Environment name: wordpressNew-env, Environment URL: http://wordpressNew-env.us-east-1.elasticbeanstalk.com). At the bottom right are 'Cancel', 'Previous', and 'Next' buttons, with 'Next' being highlighted.

## Step 12: review to check at the correct settings are been chosen

The screenshot shows the 'Configuration Details' section of the AWS Elastic Beanstalk console. It includes fields for Instance type (t2.micro), Key pair (bibo\_10\_12\_17), Email address (biboshan23@gmail.com), Root volume type (default), Root volume size (default), Root volume IOPS (default), Application health (check URL), and Environment Tags (abcd 1234). Below this is the 'RDS Configuration' section, which lists DB engine (mysql), Engine version (5.6.37), Instance class (db.t2.micro), Allocated storage (5), and Deletion policy (Create snapshot).

## Step 13: elastic beanstalk is launching

The screenshot shows the 'All Applications > WordPress\_new > wordpressNew-env' dashboard. A central message states 'Elastic Beanstalk is launching your environment.' Below it is an 'Overview' section with tabs for Health (Pending) and Causes. To the right, there's a 'Running Version' section with a PHP logo and configuration details (64bit Amazon Linux 2017.09, v2.6.2 running PHP 7.1). At the bottom, a 'Recent Events' table lists log entries from January 9, 2018:

Time	Type	Details
2018-01-09 15:07:23 UTC+0530	INFO	Created load balancer named: awseb-e-8-AWSEBLba-1L4W3DYW/GQ83D
2018-01-09 15:07:23 UTC+0530	INFO	Created security group named: awseb-e-8dxjsh4fzm-stack-AWSEBSecurityGroup-1E8CY1HHWTBQ3
2018-01-09 15:07:23 UTC+0530	INFO	Created security group named: sg-49ee6d3d
2018-01-09 15:07:12 UTC+0530	INFO	Environment health has transitioned to Pending. Initialization in progress (running for 8 seconds). There are no instances.
2018-01-09 15:07:06 UTC+0530	INFO	Created SNS Notification Topic. ARN: arn:aws:sns:us-east-1:424179511380:ElasticBeanstalkNotifications-Environment-wordpressNew-env

## This Is The Final Wordpress Website

