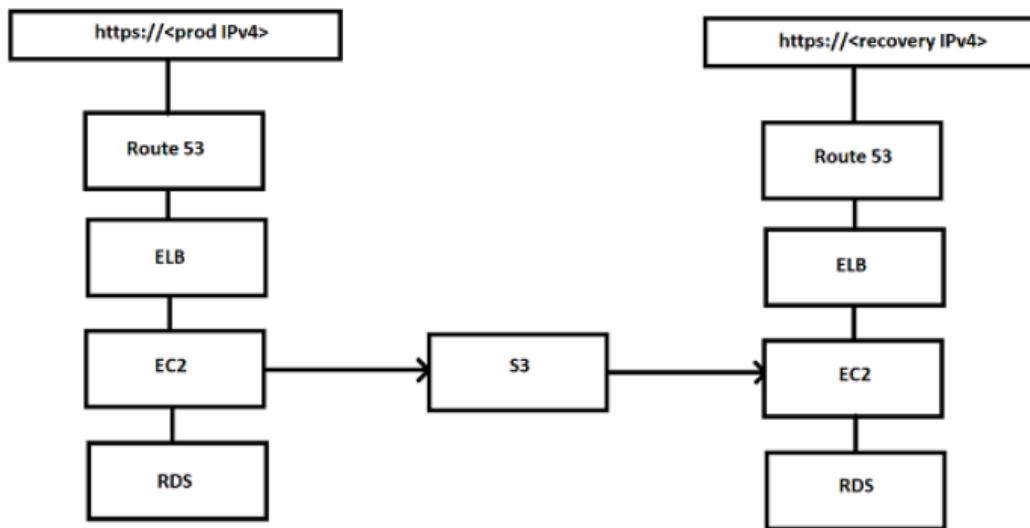


Highly availability of WordPress Application Deployment using AWS service like EC2, RDS, S3, Route-53

Project Architecture:



POV:

1. Create two EC2 instances in different Availability Zones.
2. Create an RDS database in multi-AZ mode.
3. Create an S3 bucket to store our WordPress content.
4. Create a Route 53 hosted zone for our domain name.
5. Configure EC2 instance to use the RDS database.
6. Configure our WordPress Application to use the S3 bucket.
7. Configure Route 53 to route Traffic to our EC2 instances.

Step 1: creating IAM role :

In IAM service – roles – create role – in trusted and entity type: choose AWS service – in use case: EC2 – in permission policies: choose Administrator Access – set Role name (eg : full-access – create role.

Screenshot of the AWS IAM 'Create role' wizard - Step 1: Select trusted entity.

The 'Trusted entity type' section shows the following options:

- AWS service: Allow AWS services like EC2, Lambda, or others to perform actions in this account.
- AWS account: Allow entities in other AWS accounts belonging to you or a 3rd party to perform actions in this account.
- Web identity: Allows users federated by the specified external web identity provider to assume this role to perform actions in this account.
- SAML 2.0 federation: Allow users federated with SAML 2.0 from a corporate directory to perform actions in this account.
- Custom trust policy: Create a custom trust policy to enable others to perform actions in this account.

The 'Use case' section indicates: Allow an AWS service like EC2, Lambda, or others to perform actions in this account. The 'Service or use case' dropdown is set to EC2.

Screenshot of the AWS IAM 'Create role' wizard - Step 3: Add permissions.

The 'Permissions policies' section lists the following managed policies:

Policy name	Type	Description
<input checked="" type="checkbox"/> AdministratorAccess	AWS managed - job function	AWS managed
<input type="checkbox"/> AdministratorAccess-Amplify	AWS managed	AWS managed
<input type="checkbox"/> AdministratorAccess-AWSElasticBeanst...	AWS managed	AWS managed

The screenshot shows two pages from the AWS IAM console.

Top Page (Create role | IAM | Global):

- Step 2: Add permissions**
- Role name:** full-access
- Description:** Allows EC2 Instances to call AWS services on your behalf.
- Step 1: Select trusted entities**
- Trust policy:**

```

1  [
2   "Version": "2012-10-17",
3   "Statement": [
4     {
5       "Effect": "Allow",
6       "Action": [
7         "sts:AssumeRole"
8       ],
9       "Principal": [
10        "arn:aws:iam::aws:service:ec2.amazonaws.com"
11      ]
12    }
13  ]

```

Bottom Page (Identity and Access Management (IAM)):

- Roles (5) Info:** Shows a list of roles with their details and last activity.
- Role Name:** full-access, Trusted entities: AWS Service: ec2, Last activity: -
- Other Roles:**
 - AWSServiceRoleForAmazonSSM, Trusted entities: AWS Service: ssm (Service-Linked Role), Last activity: 32 minutes ago
 - AWSServiceRoleForRDS, Trusted entities: AWS Service: rds (Service-Linked Role), Last activity: 45 minutes ago
 - AWSServiceRoleForSupport, Trusted entities: AWS Service: support (Service-Linked Role), Last activity: -
 - AWSServiceRoleForTrustedAdvisor, Trusted entities: AWS Service: trustedadvisor (Service-Linked Role), Last activity: -

Successfully created a role with administrator access.

Step 2: create route 53 hosted zone for our domain name.

In Route 53 service – create hosted zone – set Domain name: borutomokshi.world – type: public hosted zone – create hosted zone.

The screenshot shows the 'Create hosted zone' configuration page in the AWS Route 53 console. The 'Hosted zone configuration' section is open, showing the following fields:

- Domain name:** borutomokshi.world
- Description - optional:** The hosted zone is used for...
- Type:** Public hosted zone (selected)

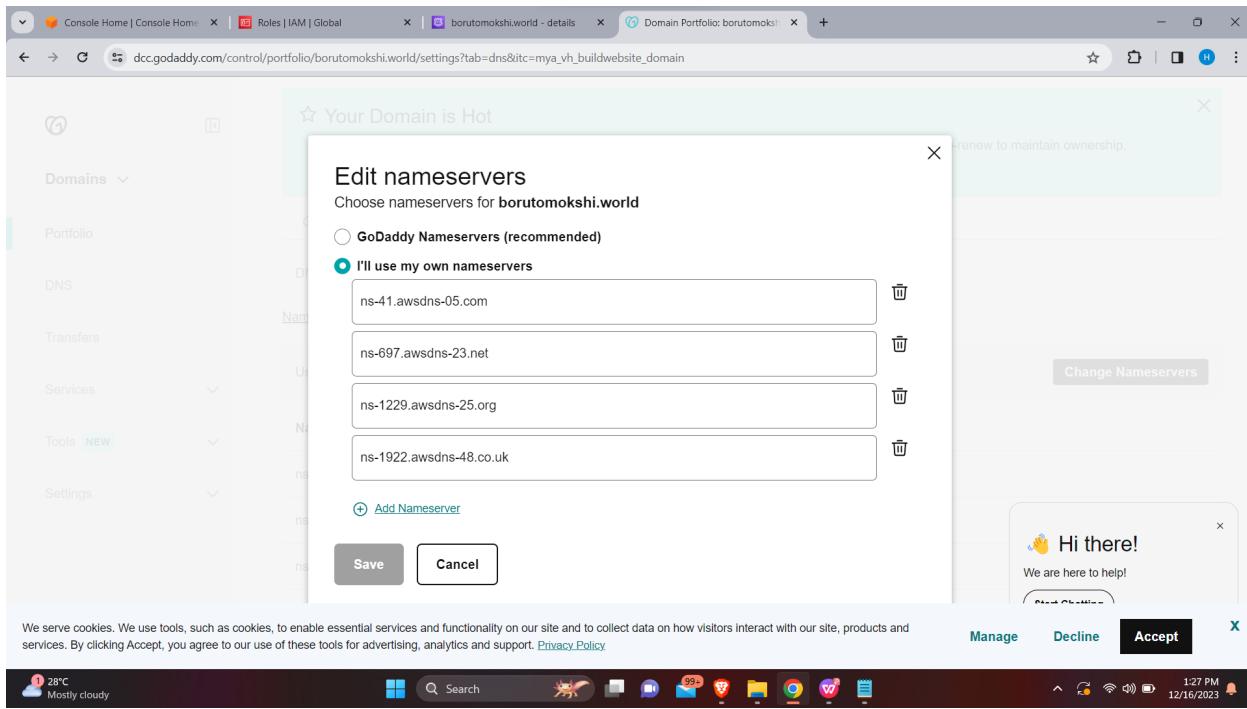
The page also includes a sidebar with navigation links like CloudShell, Feedback, and a weather widget showing 28°C Mostly cloudy. The top navigation bar shows tabs for Route 53, Hosted zones, and Create hosted zone.

In hosted zone details, name servers are created.

The screenshot shows the 'Hosted zones' details page for the domain borutomokshi.world. The 'Hosted zone details' section displays the following information:

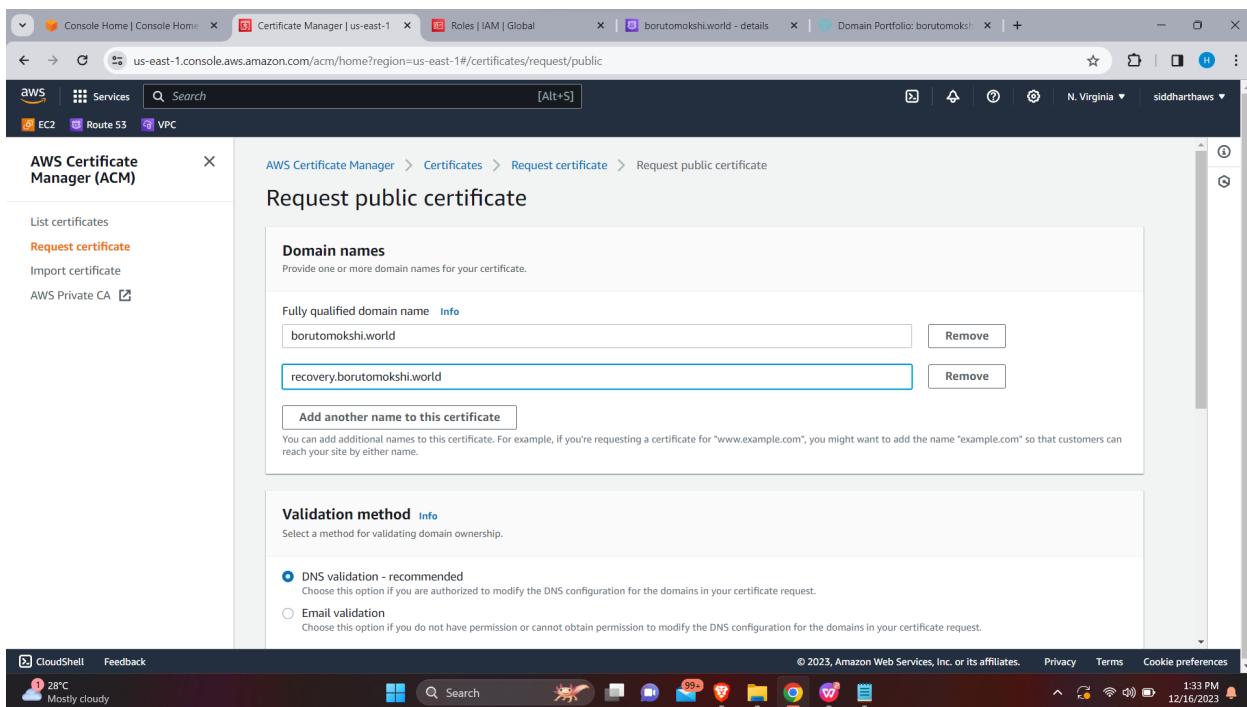
Hosted zone name	Query log	Name servers
borutomokshi.world	-	ns-41.awsdns-05.com ns-697.awsdns-23.net ns-1229.awsdns-25.org ns-1922.awsdns-48.co.uk
Hosted zone ID	Type	Record count
Z09274171OJN4XFRANYNP	Public hosted zone	2
Description		
-		

Copy the name servers & paste it in the third party domain name provider.



Step 3: request SSL certificate for our domain name.

In certificate manager service – request – request type: public – in domain name set fully qualified domain name.



In Validation method: DNS method – in key algorithm: RSA 2048 – click request.

The screenshot shows the AWS Certificate Manager interface. In the left sidebar, under 'AWS Certificate Manager (ACM)', there are options like 'List certificates', 'Request certificate', 'Import certificate', and 'AWS Private CA'. The main area displays a table titled 'Certificates (1)'. The table has columns for 'Certificate ID', 'Domain name', 'Type', 'Status', 'In use', 'Renewal eligibility', and 'Key algorithm'. One row is listed: '5e13b6e8-4852-496d-875e-f0899c004ea1' for 'borutomokshi.world' which is 'Amazon Issued' and 'Pending validation'. A large orange 'Request' button is visible at the top right.

Certificate has been requested successfully. Wait for some time for the certificate to be issued.

Go to requested certificate details – in domain: one new record is created and click create record in route 53 – create records.

The screenshot shows the 'Create DNS records in Amazon Route 53' dialog. It includes a search bar for 'Search domains', filter buttons for 'Validation status: Pending validation' and 'Is domain in Route 53? Yes', and a 'Clear filter' button. A table lists a single record: 'recovery.bor utomokshi.w orld' with 'Validation status: Pending validation' and 'Type: CNAME'. The 'CNAME name' is '_05a2381a81f18cc973fa9437017801' and the 'CNAME value' is '98.recovery.boruto mokshi.world._0e1a3e92b4df0dc72e4231...'. At the bottom are 'Cancel' and 'Create records' buttons.

In route 53 service, CNAME record is created.

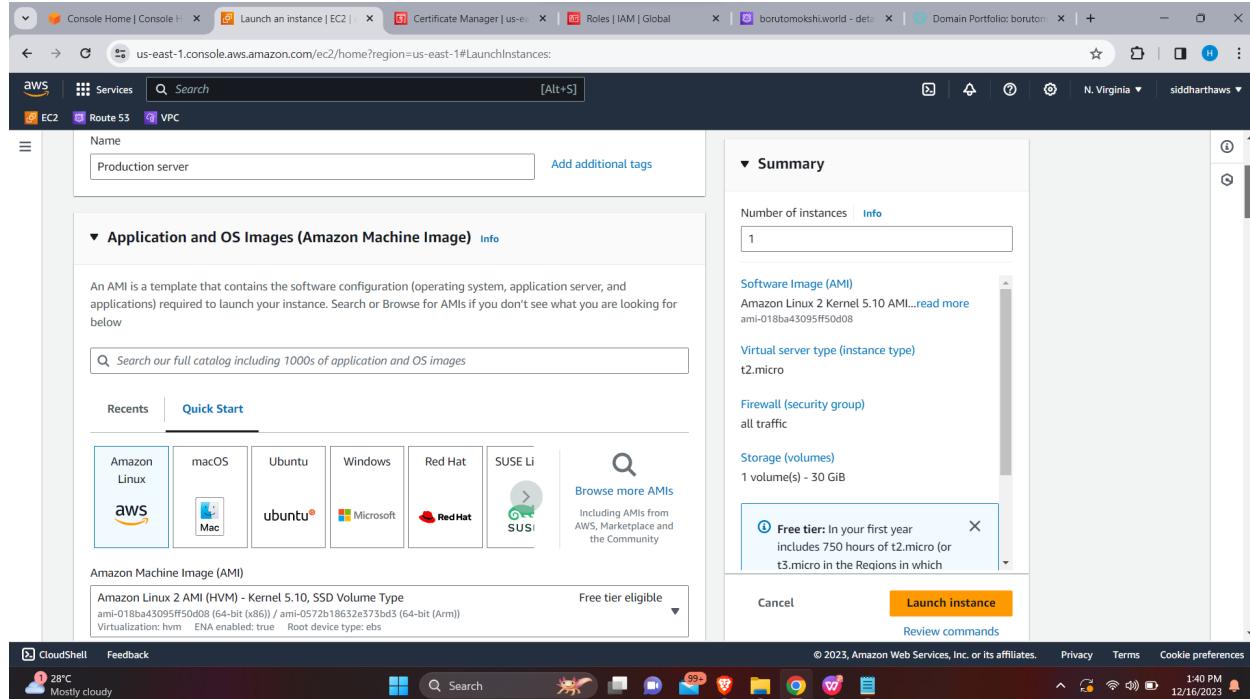
The screenshot shows the AWS Route 53 service. The left sidebar has sections like 'Hosted zones', 'IP-based routing', 'Traffic flow', 'Domains', 'Resolver', 'VPCs', and 'Query logging'. Under 'Hosted zones', 'borutomokshi.world - details' is selected. The main area shows 'Hosted zone details' with a table of 'Records (3)'. The table has columns for 'Record ...', 'Type', 'Routing policy', 'Alias', and 'Value/Route traffic to'. The records are: 'borutomo...' (NS, Simple, No, ns-41.awsdns-05.com), 'borutomo...' (SOA, Simple, No, ns-41.awsdns-05.com), and '_05a2381...' (CNAME, Simple, No, '_0e1a3e92b4df0dc72e4231...'). A message on the right says '0 records selected' and 'Select a record to see its details'.

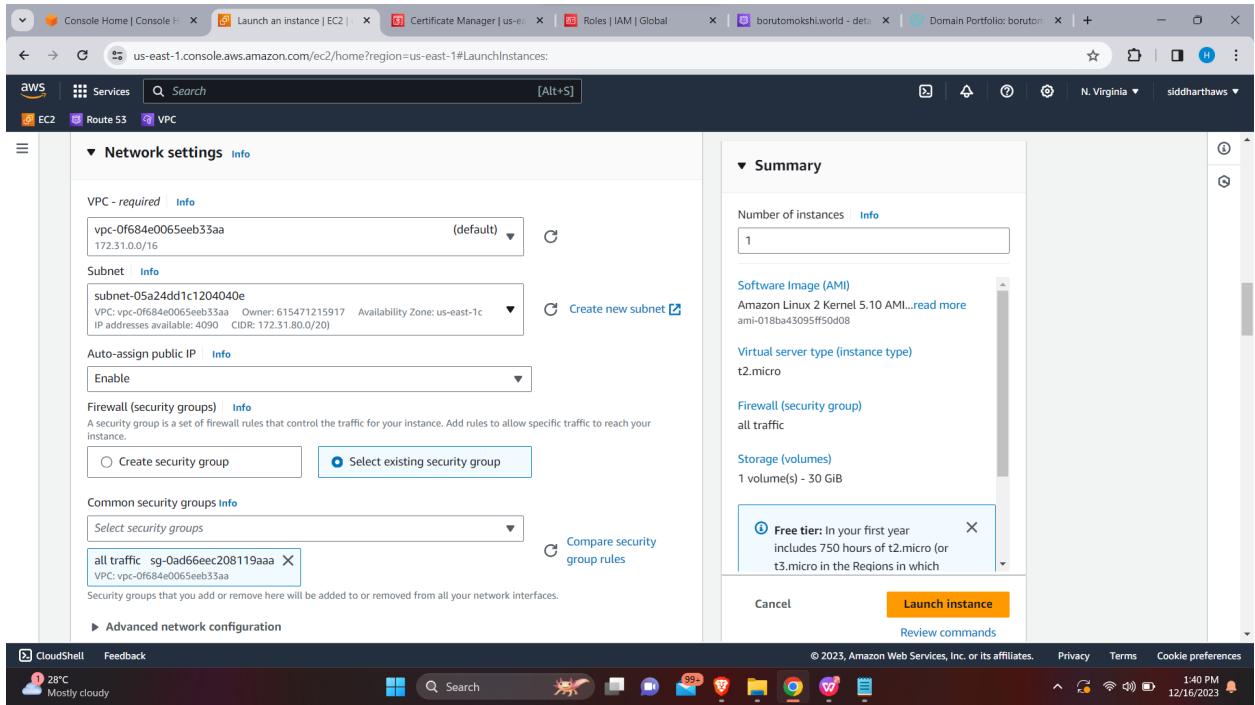
Step 4: create two instances for production and recovery with IAM role enabled.

Go to EC2 service – launch instance – name : **production** – in AMIs : amazon Linux 2 – instance Type : t2.micro – key pair : .ppk – Security group : All Traffic – configure storage : 30 GiB – in additional details : attach IAM role created – in user data : add below commands

```
#!/bin/bash
yum install httpd php-mysql -y
amazon-linux-extras install -y php7.3
cd /var/www/html
echo "healthy" > healthy.html
wget https://wordpress.org/latest.tar.gz
tar -xzf latest.tar.gz
cp -r wordpress/* /var/www/html/
rm -rf wordpress
rm -rf latest.tar.gz
chmod -R 755 wp-content
chown -R apache:apache wp-content
wget https://s3.amazonaws.com/bucketforwordpresslab-donotdelete/.htaccess.txt
mv htaccess.txt .htaccess
service httpd start
```

Then launch an instance.





Instance for production server launched successfully.

Go to EC2 service – launch instance – name : **recovery** – in AMIs : amazon Linux 2 – instance Type : t2.micro – key pair : .ppk – Security group: All Traffic – configure storage : 30 GiB – in additional details : attach IAM role created – in user data : add below commands

```
#!/bin/bash
yum install httpd php-mysql -y
amazon-linux-extras install -y php7.3
cd /var/www/html
echo "healthy" > healthy.html
wget https://wordpress.org/latest.tar.gz
tar -xzf latest.tar.gz
cp -r wordpress/* /var/www/html/
rm -rf wordpress
rm -rf latest.tar.gz
chmod -R 755 wp-content
chown -R apache:apache wp-content
wget https://s3.amazonaws.com/bucketforwordpresslab-donotdelete/htaccess.txt
mv htaccess.txt .htaccess
service httpd start
```

Then launch an instance.

The screenshot shows two instances of the AWS Management Console interface. Both instances are navigating to the 'Launch Instances' page for the 'us-east-1' region.

Top Instance (Left):

- Summary:** Number of instances: 1
- Software Image (AMI):** Amazon Linux 2 Kernel 5.10 AMI... (ami-018ba43095ff50d08)
- Virtual server type (instance type):** t2.micro
- Firewall (security group):** all traffic
- Storage (volumes):** 1 volume(s) - 30 GiB
- Free tier message:** In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which you launch)
- Buttons:** Cancel, Launch instance, Review commands

Bottom Instance (Right):

- Network settings:**
 - VPC - required:** (default)
 - Subnet:** subnet-05a24dd1c1204040e (VPC: vpc-0f684e0065eeb33aa, Owner: 615471215917, Availability Zone: us-east-1c, IP addresses available: 4089, CIDR: 172.31.80.0/20)
 - Auto-assign public IP:** Enable
 - Firewall (security groups):** Create security group, Select existing security group (selected)
 - Common security groups:** Select security groups (all traffic, sg-0ad66ec208119aaa)
 - Advanced network configuration:**
- Summary:** Number of instances: 1
- Software Image (AMI):** Amazon Linux 2 Kernel 5.10 AMI... (ami-018ba43095ff50d08)
- Virtual server type (instance type):** t2.micro
- Firewall (security group):** all traffic
- Storage (volumes):** 1 volume(s) - 30 GiB
- Free tier message:** In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which you launch)
- Buttons:** Cancel, Launch instance, Review commands

Common UI Elements:

- CloudShell Feedback
- CloudWatch Metrics icon
- Search bar
- Notification icons (99, 99+)
- File, Copy, Paste, Delete, Refresh, Minimize, Maximize, Close buttons
- Footer: © 2023, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences
- System tray: 145 PM, 12/16/2023, Weather (28°C), Battery, Network, Volume

Instance for recovery server has been successfully launched.

Step 5 : create a subnet group & database for production server.

Creating subnet group for the database

The screenshot shows the AWS RDS console in the N. Virginia region. The user has just successfully deleted a DB instance named 'hemanth'. They are now creating a new DB subnet group named 'Project-subnet'.

Subnet group details:

- Name:** Project-subnet
- Description:** Project-subnet
- VPC:** Choose a VPC identifier that corresponds to the subnets you want to use for your DB subnet group. You won't be able to choose a different VPC identifier after your subnet group has been created.
vpc-0f684e0065eeb33aa

Add subnets:

- Availability Zones:** Choose the Availability Zones that include the subnets you want to add.
Choose an availability zone
us-east-1a X us-east-1c X
- Subnets:** Choose the subnets that you want to add. The list includes the subnets in the selected Availability Zones.
Select subnets
subnet-05a24dd1c1204040e (172.31.80.0/20) X
subnet-0c58d4e5698c8ad12 (172.31.2.0/24) X
- Subnets selected (2):**

Availability zone	Subnet ID	CIDR block
us-east-1c	subnet-05a24dd1c1204040e	172.31.80.0/20
us-east-1a	subnet-0c58d4e5698c8ad12	172.31.2.0/24

In RDS service – create DB subnet – create database for production

Choose a database creation method

- 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](#)

- Aurora (MySQL Compatible)
- Aurora (PostgreSQL Compatible)
- MySQL
- MariaDB
- PostgreSQL
- Oracle

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 64 TiB.
- Supports General Purpose, Memory Optimized, and Burstable Performance instance classes.
- Supports automated backup and point-in-time recovery.
- Supports up to 15 Read Replicas per instance, within a single Region or 5 read replicas cross-region.

Configure DB settings:

DB instance identifier [Info](#)

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

proddb

The DB instance identifier is case-insensitive, but is stored as all lowercase (as in "mydbinstance"). Constraints: 1 to 60 alphanumeric characters or hyphens. 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.

hemanth

1 to 16 alphanumeric characters. The first character must be a letter.

Manage master credentials in AWS Secrets Manager

Manage master user credentials in Secrets Manager. RDS can generate a password for you and manage it throughout its lifecycle.

If you manage the master user credentials in Secrets Manager, some RDS features aren't supported. [Learn more](#)

Auto generate a password

Amazon RDS can generate a password for you, or you can specify your own password.

Master password [Info](#)

Constraints: At least 8 printable ASCII characters. Can't contain any of the following: / (slash), '(single quote), "(double quote) and @

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 64 TiB.
- Supports General Purpose, Memory Optimized, and Burstable Performance instance classes.
- Supports automated backup and point-in-time recovery.
- Supports up to 15 Read Replicas per instance, within a single Region or 5 read replicas cross-region.

In instance configuration, DB instance class : burstable class – db.t2.micro.

In storage : disable auto scaling.

In connectivity : connect EC2 compute resource – choose : created **production** instance.

Connectivity Info



Compute resource

Choose whether to set up a connection to a compute resource for this database. Setting up a connection will automatically change connectivity settings so that the compute resource can connect to this database.

Don't connect to an EC2 compute resource

Don't set up a connection to a compute resource for this database. You can manually set up a connection to a compute resource later.

Connect to an EC2 compute resource

Set up a connection to an EC2 compute resource for this database.

EC2 instance Info

Choose the EC2 instance to add as the compute resource for this database. A VPC security group is added to this EC2 instance. A VPC security group is also added to the database with an inbound rule that allows the EC2 instance to access the database.

i-059020469e913a811

Production server



DB subnet choose created subnet

DB subnet group Info

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

Choose existing

Choose existing DB subnet group

Automatic setup

RDS creates a new subnet group for you or reuses an existing subnet group

Existing DB subnet groups

project-subnet

2 Subnets, 2 Availability Zones



VPC security group : choose All Traffic

VPC security group (firewall) Info

Choose one or more VPC security groups to allow access to your database. Make sure that the security group rules allow the appropriate incoming traffic.

Choose existing

Choose existing VPC security groups

Create new

Create new VPC security group

Additional VPC security group

Choose one or more options



all traffic

In additional configuration – set initial database name : proddb – in backup : unmark enable

automated backup – in maintenance : unmark enable auto minor version upgrade.
Click Create database.

The screenshot shows the AWS RDS console interface for creating a new database. On the left, there's a sidebar with 'Database options' and dropdown menus for 'Initial database name' (set to 'proddb'), 'DB parameter group' (set to 'default.mysql5.7'), and 'Option group' (set to 'default:mysql-5.7'). Below these are sections for 'Backup' (with an unchecked 'Enable automated backups' checkbox) and 'Log exports' (listing 'Audit log', 'Error log', 'General log', and 'Slow query log'). At the bottom, there's an 'IAM role' section and a 'Create database' button. A tooltip for 'MySQL' on the right side lists its features, including support for up to 64 TiB, various instance classes, automated backups, and up to 15 read replicas per instance.

Database has been successfully created for the production server.

Step 6: Creating Database for Recovery server.

In RDS service – create DB subnet – create database for recovery.

Engine option : MySQL-5.7.37

Template : free tier

Configure DB settings:

The screenshot shows the AWS RDS console for creating a new database instance. The instance identifier is set to "recoverydb". Under "Credentials Settings", the master username is "hemanth". A note states that managing master user credentials in AWS Secrets Manager is not supported. There are options for auto-generating a password or specifying a master password. The MySQL sidebar on the right details its popularity and various features.

In instance configuration, DB instance class: burstable class – db.t2.micro.

In storage : disable auto scaling.

In connectivity : connect EC2 compute resource – choose : created **recovery** instance.

Connectivity Info



Compute resource

Choose whether to set up a connection to a compute resource for this database. Setting up a connection will automatically change connectivity settings so that the compute resource can connect to this database.

Don't connect to an EC2 compute resource

Don't set up a connection to a compute resource for this database. You can manually set up a connection to a compute resource later.

Connect to an EC2 compute resource

Set up a connection to an EC2 compute resource for this database.

EC2 instance Info

Choose the EC2 instance to add as the compute resource for this database. A VPC security group is added to this EC2 instance. A VPC security group is also added to the database with an inbound rule that allows the EC2 instance to access the database.

i-0c2383fe0bfee9fe6

Recovery server



DB subnet choose created subnet

DB subnet group [Info](#)

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

Choose existing

Choose existing DB subnet group

Automatic setup

RDS creates a new subnet group for you or reuses an existing subnet group

Existing DB subnet groups

project-subnet

2 Subnets, 2 Availability Zones

VPC security group : choose All Traffic

VPC security group (firewall) [Info](#)

Choose one or more VPC security groups to allow access to your database. Make sure that the security group rules allow the appropriate incoming traffic.

Choose existing

Choose existing VPC security groups

Create new

Create new VPC security group

Additional VPC security group

Choose one or more options

all traffic 

In additional configuration – set initial database name: recoverydb – in backup: unmark enable automated backup – in maintenance: unmark enable auto minor version upgrade.

Click Create database.

The screenshot shows the AWS RDS console interface. On the left, there's a sidebar with 'Services' (selected), EC2, Route 53, and VPC. The main area is titled 'Database options' and contains fields for 'Initial database name' (set to 'recoverydb'), 'DB parameter group' (set to 'default.mysql5.7'), and 'Option group' (set to 'default:mysql-5.7'). Below these are sections for 'Backup' (with an unchecked checkbox for 'Enable automated backups') and 'Log exports' (with checkboxes for 'Audit log', 'Error log', 'General log', and 'Slow query log', all of which are unchecked). A 'IAM role' section is also present. To the right, a sidebar titled 'MySQL' provides a brief overview of the service, mentioning it's the most popular open source database and highlighting features like support for up to 64 TiB, various performance classes, automated backups, and up to 15 Read Replicas per instance. At the bottom of the screen, the Windows taskbar shows icons for CloudShell, Feedback, SHT13 / NSK Salai Construction, and a weather widget indicating 29°C and mostly cloudy.

Database for the recovery server has been successfully created.

Step 7: configure initial database credentials for production server.

Copy public IPv4 address of production server & hit in browser.

The screenshot shows a web browser window with the URL '54.163.131.184/wp-admin/setup-config.php'. The page features the classic blue WordPress logo at the top. Below it, a box contains the text: 'Welcome to WordPress. Before getting started, you will need to know the following items.' followed by a numbered list: 1. Database name, 2. Database username, 3. Database password, 4. Database host, and 5. Table prefix (for running multiple sites in one database). A note below states: 'This information is being used to create a wp-config.php file. If for any reason this automatic file creation does not work, do not worry. All this does is fill in the database information to a configuration file. You may also simply open wp-config-sample.php in a text editor, fill in your information, and save it as wp-config.php. Need more help? Read the support article on wp-config.php.' At the bottom of the form, there's a button labeled 'Let's go!' and the browser's standard navigation and search bars at the bottom.

Configure initial database for production Database.

1. Database name
2. Database username
3. Database password
4. Database <endpoint>:3306

Click let's go

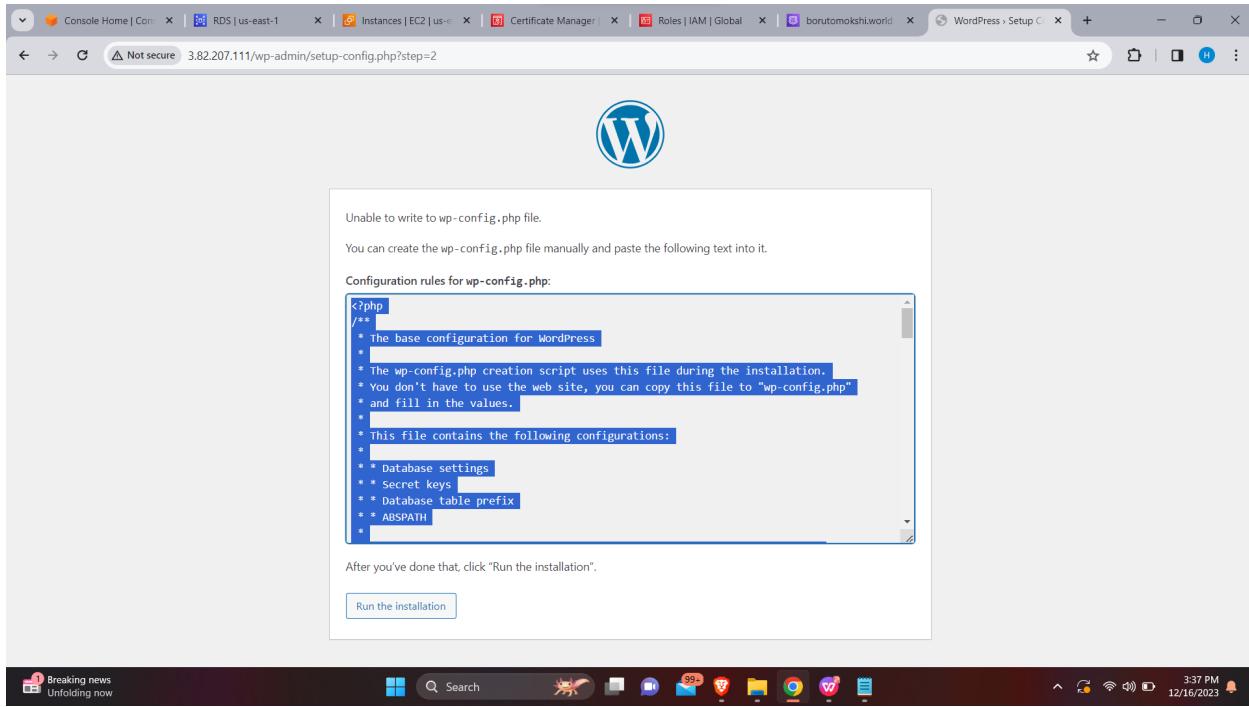
The screenshot shows a web browser window with multiple tabs open, including AWS services like RDS and IAM. The active tab is 'WordPress > Setup C'. The URL in the address bar is '3.82.207.111/wp-admin/setup-config.php?step=1'. The page content is a form for entering database connection details:

Below you should enter your database connection details. If you are not sure about these, contact your host.

Database Name	proddb	<small>The name of the database you want to use with WordPress.</small>
Username	hemanth	<small>Your database username.</small>
Password	*****	<small>Your database password.</small>
Database Host	proddb.c3htdkw1zjja.us-east-1.rds.amazonaws.com	<small>You should be able to get this info from your web host, if localhost does not work.</small>
Table Prefix	wp_	<small>If you want to run multiple WordPress installations in a single database, change this.</small>

Submit

Here, in WordPress Application it can't able to write wp-config.php file. so, we need write manually.



Copy the content of **wp-config.php** from WordPress and paste in the path /var/www/html directory.

Open terminal in **production server**

```
[root@ip-172-31-81-108 ~]# cd /var/www/html/  
[root@ip-172-31-81-108 html]# vi wp-config.php
```

```

root@ip-172-31-81-108:/var/www/html

define( 'AUTH_KEY', 'cVc *twjJgw2nL4*taw10dprojnub_11FF(-8D-0TENMMHlcayb2LEvzW),(fr,' );
define( 'SECURE_AUTH_KEY', '11+NjW-~dnnd1 W-waQORC)xiA_0/(Y02SwW+mdXN'-m3rB(f7x)XEE(b)cT' );
define( 'LOGGED_IN_KEY', '5Dvk1a,<fzH>x2q-u771t),20HFk(wuA5M6ix#8TK2(p..cuUFERmekv-32E!B!' );
define( 'NONCE_KEY', 't(UVMyi1i'N'DIM1S 0s)r9hkg7WXYDE2 [yv77k) [442 ovjbP9 SThvv' );
define( 'AUTH_SALT', 'Y:Cda-i6VM49'sj0)U'SB9<nChCxs1Rz)yxw(mk+vUNte{|| (oph#UGIMy(' );
define( 'SECURE_AUTH_SALT', 'Fm-|SWLh=ch9(x5LoF1gfb(SXFCf1s L69)WGH'VEEEEn Vnz[K07()!`R' );
define( 'LOGGED_IN_SALT', '+r<L+F8Y49q-aOM5Be:u/SyYCO5JAfpB nw''4fOD1*Tr<zAAP)KQ48-eElIV' );
define( 'NONCE_SALT', 'su:v5.x*x'jgm"7zg6)2R1i4+1fUA3g|[BSGNaX/H9UW[!r|c8cB+)SRRRE)#+A' );
/*$#*/ */

/*
* WordPress database table prefix.
*
* you can have multiple installations in one database if you give each
* a unique prefix. Only numbers, letters, and underscores please!
*/
$table_prefix = 'wp_';

/*
* For developers: WordPress debugging mode.
*
* Change this to true to enable the display of notices during development.
* It is strongly recommended that plugin and theme developers use WP_DEBUG
* in their development environments.
*
* For information on other constants that can be used for debugging,
* visit the documentation.
*
* @link https://wordpress.org/documentation/article/debugging-in-wordpress/
*/
define( 'WP_DEBUG', false );

/* Add any custom values between this line and the "stop editing" line. */

/* That's all, stop editing! Happy publishing. */

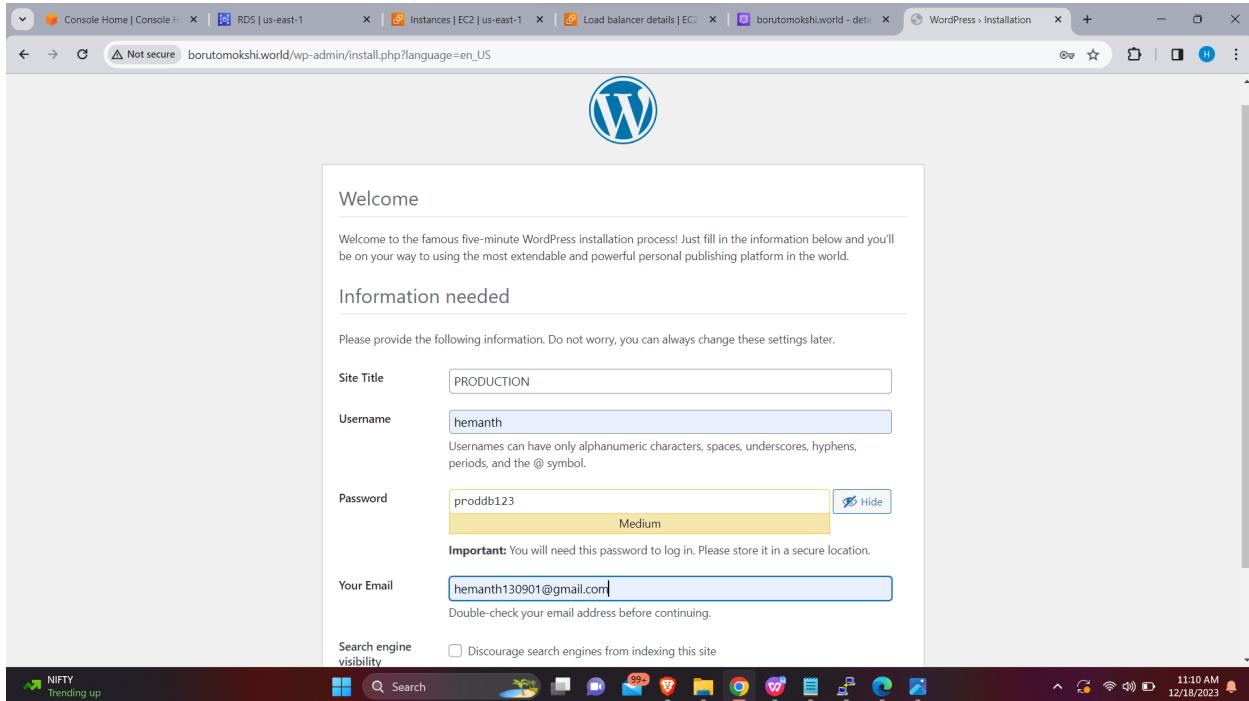
/** Absolute path to the WordPress directory. */
if ( ! defined( 'ABSPATH' ) ) {
    define( 'ABSPATH', __DIR__ . '/' );
}

/** Sets up WordPress vars and included files. */
require_once ABSPATH . 'wp-settings.php';

```

Now, click **Run the Installation.**

Step 8: configure database information details.



The screenshot shows the WordPress dashboard with a dark theme. The main header says "Welcome to WordPress!" with a link to "Learn more about the 6.4.2 version". Below the header, there are three cards: "Author rich content with blocks and patterns", "Customize your entire site with block themes", and "Switch up your site's look & feel with Styles". A notification bar at the bottom left says "PHP Update Recommended" with a message about an outdated PHP version (7.3.33). On the right, a "Quick Draft" window is open with a title field. The taskbar at the bottom shows various application icons.

Welcome to WordPress!

[Learn more about the 6.4.2 version](#)

Author rich content with blocks and patterns

Customize your entire site with block themes

Switch up your site's look & feel with Styles

PHP Update Recommended

Your site is running on an outdated version of PHP (7.3.33), which does not receive security updates. It should be updated.

What is PHP and how does it affect my site?

Quick Draft

Title

Post updated. View Post

Post Block

Summary

Visibility Public

Publish December 18, 2023 5:40 am UTC+0

Template Single Posts

URL borutomokshi.world/index.php/2023/12/18/hello-world/

Stick to the top of the blog

AUTHOR hemanth

[Switch to draft](#) [Move to trash](#)

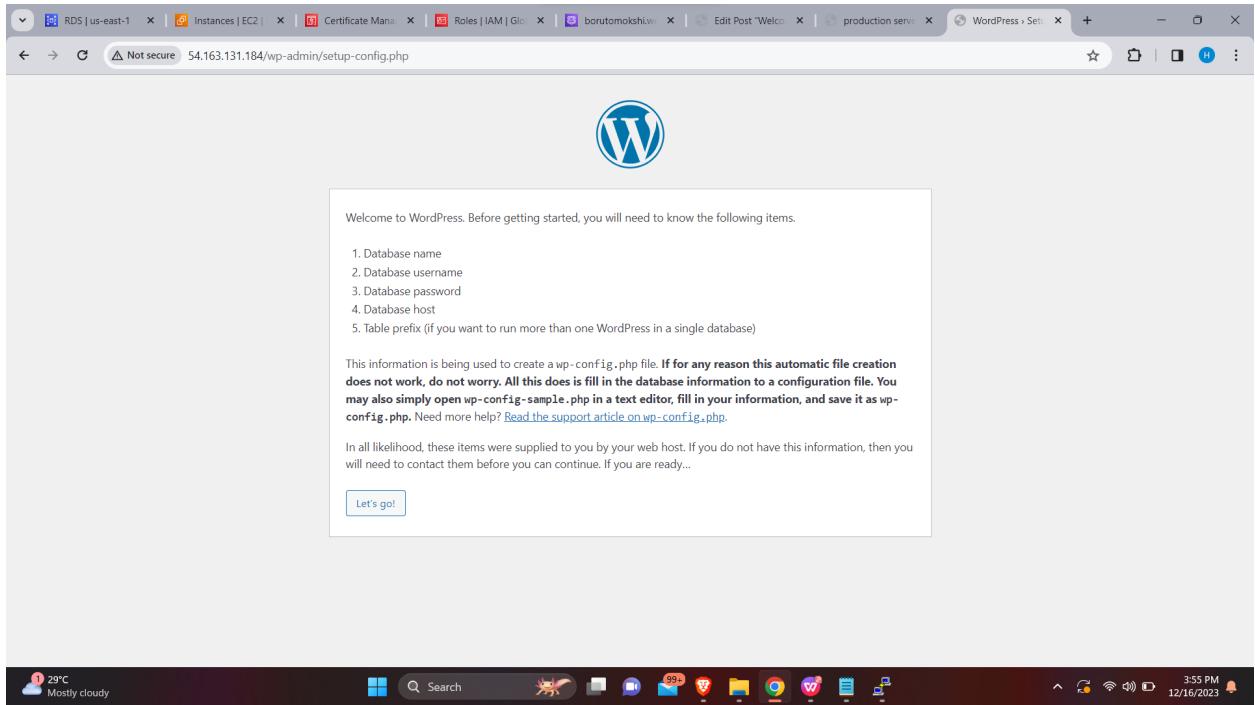
Categories

Tags

Featured image

Post updated. View Post

Step 9 : configure initial database credentials for recovery server.
Copy public IPv4 address of recovery server & hit in browser.

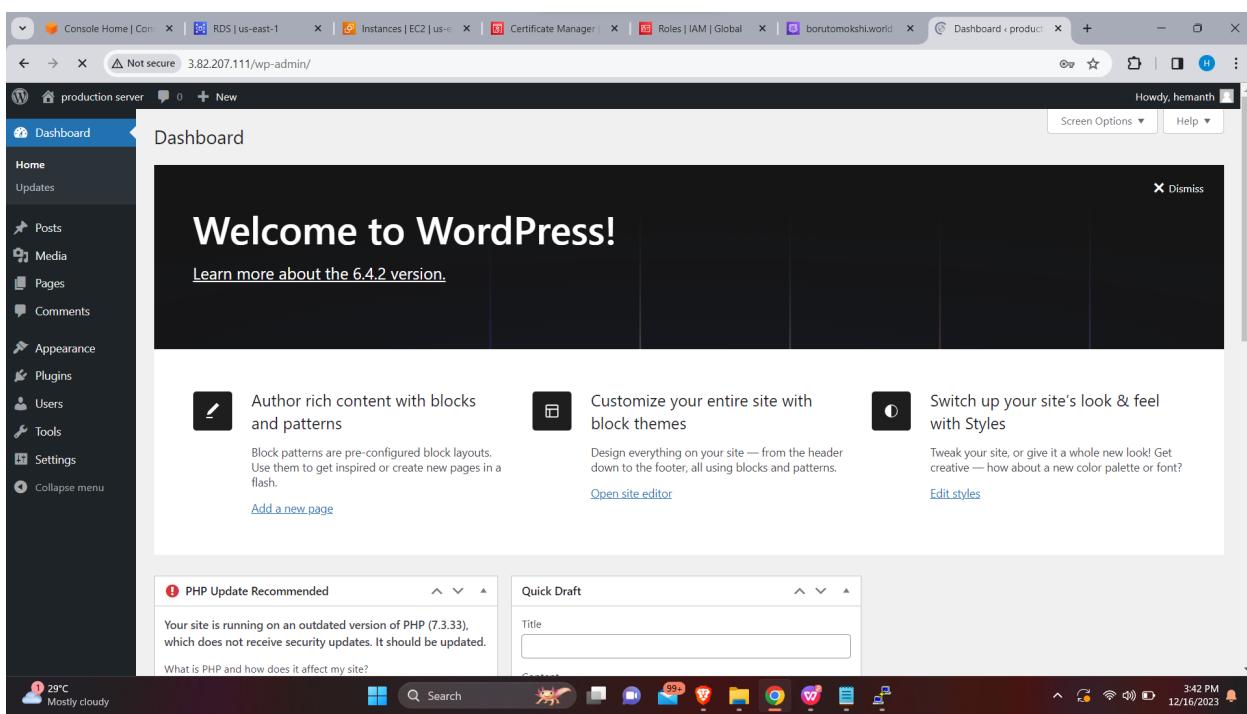
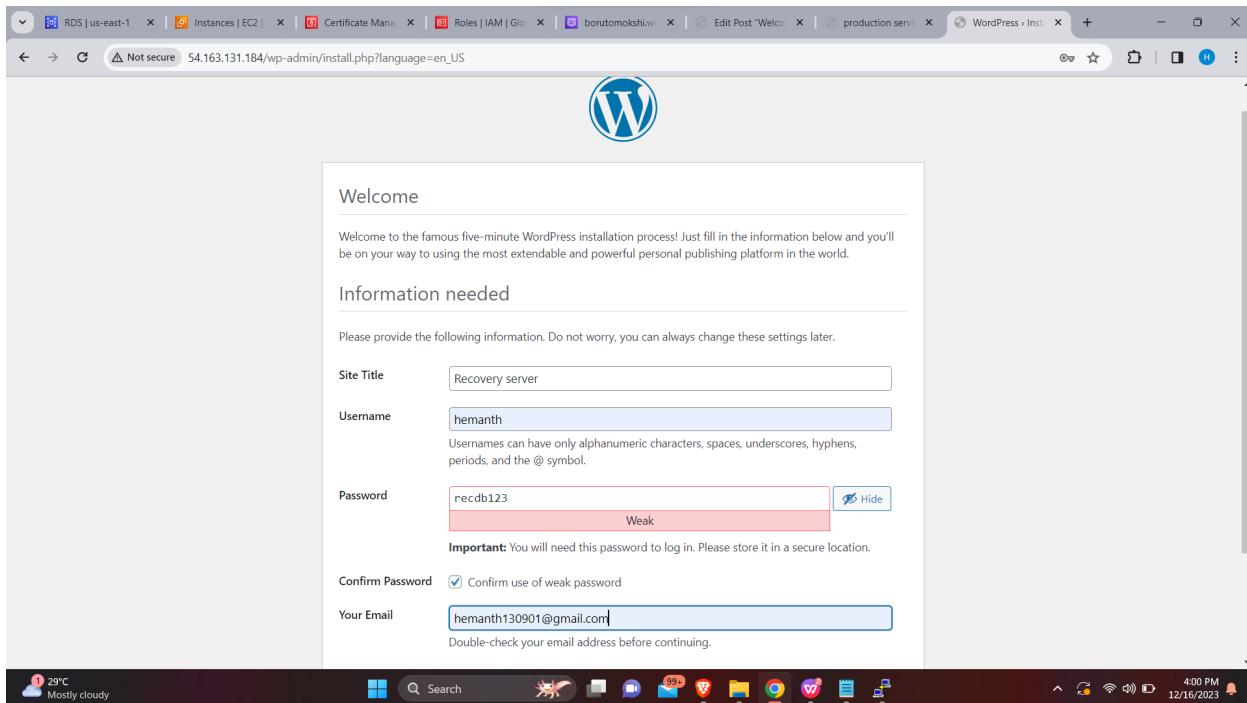


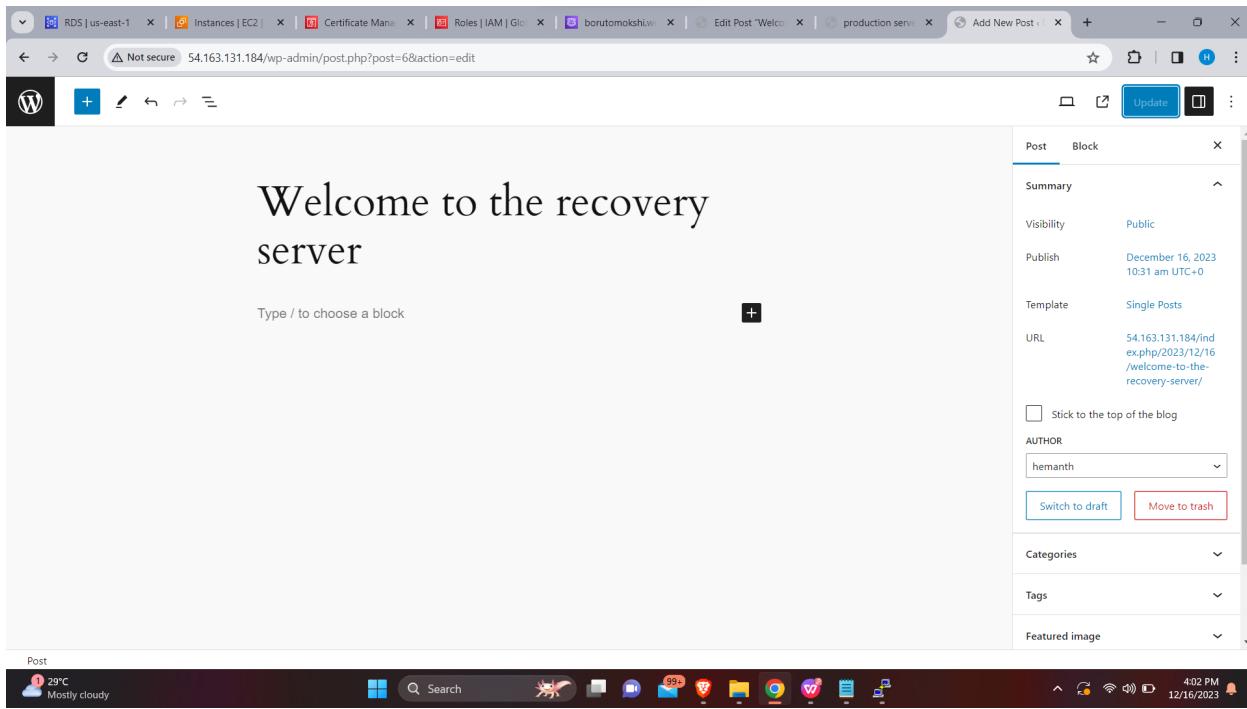
Configure initial database for production Database.

Write the wp-config.php file in /var/www/html directory and run the installation.

```
root@ip-172-31-85-62:/var/www/html
└─ login as: ec2-user
└─ Authenticating with public key "nvlink"
      ,#
      ~\###_
      ~~\###\ Amazon Linux 2
      ~~ \|#| AL2 End of Life is 2025-06-30.
      ~~ \|/ V~,'-->
      ~~~ / A newer version of Amazon Linux is available!
      ~~-. / Amazon Linux 2023, GA and supported until 2028-03-15.
      _/m/, / https://aws.amazon.com/linux/amazon-linux-2023/
[ec2-user@ip-172-31-85-62 ~]$ sudo -i
[root@ip-172-31-85-62 ~]# cd /var/www/html/
[root@ip-172-31-85-62 html]# vi wp-config.php
```

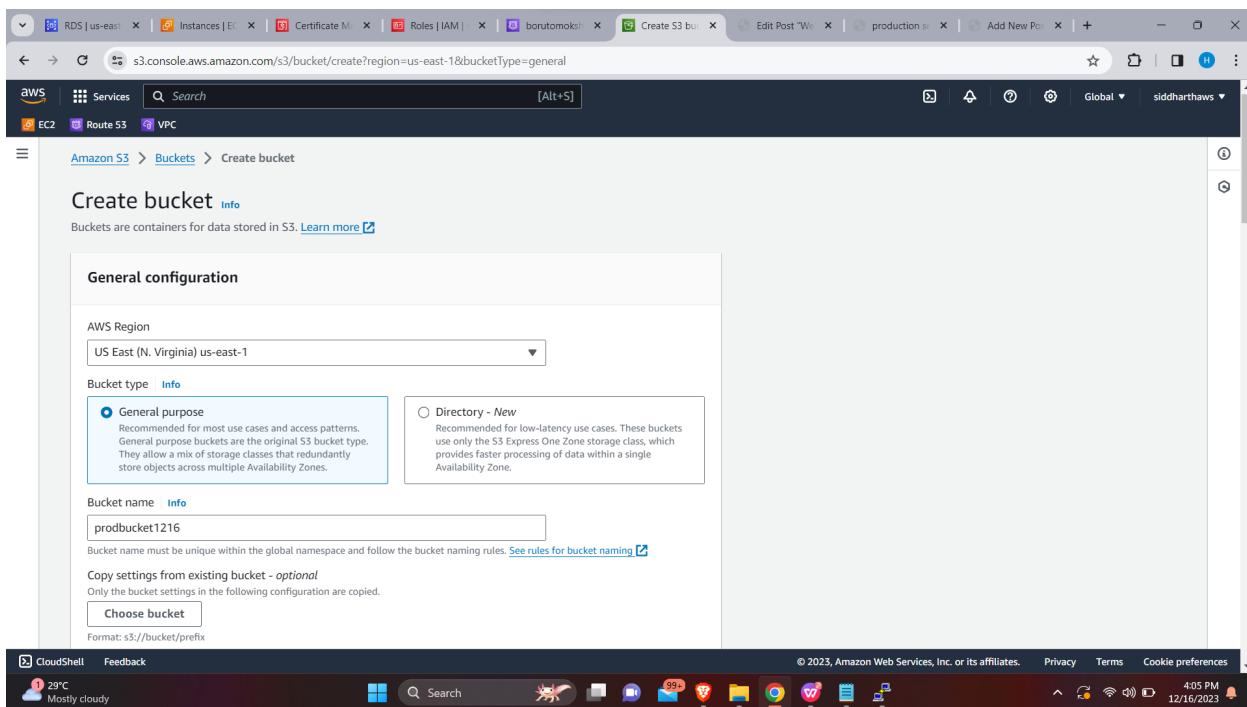
Step 10: configure database information details.





Step 11: create two S3 buckets to add WordPress content

In S3 bucket – create bucket – set unique bucket name – in object ownership: ACLs enabled – remove mark from block all public access and acknowledge it – in bucket versioning enabled – click create bucket.



Go to the details of created bucket – go to the permission – in Access Control List.

Access control list (ACL)

Grantee	Objects	Bucket ACL
Bucket owner (your AWS account)	<input checked="" type="checkbox"/> List <input checked="" type="checkbox"/> Write	<input checked="" type="checkbox"/> Read <input checked="" type="checkbox"/> Write
Canonical ID: 7659393bb581594e62ac afb31feb7285f2c02ba709972 47649e396014db809		
Everyone (public access) Group: http://acs.amazonaws.com/groups/global/AllUsers	<input checked="" type="checkbox"/> List <input type="checkbox"/> Write	<input checked="" type="checkbox"/> Read <input type="checkbox"/> Write
Authenticated users group (anyone with an AWS account) Group: http://acs.amazonaws.com/groups/global/AuthenticatedUsers	<input type="checkbox"/> List <input type="checkbox"/> Write	<input type="checkbox"/> Read <input type="checkbox"/> Write

Here, bucket is publicly accessible.

prodbucket1216 Info Publicly accessible

Permissions overview

Access
Public

Two buckets created successfully.

General purpose buckets (2)

Name	AWS Region	Access	Creation date
prodbucket1216	US East (N. Virginia) us-east-1	Public	December 16, 2023, 16:05:59 (UTC+05:30)
recbucket1216	US East (N. Virginia) us-east-1	Public	December 16, 2023, 16:10:06 (UTC+05:30)

Step 12 : assign cron job for production server.

```
#crontab -e
```

Add cronjob.

```
root@ip-172-31-81-108:~#
*/2 * * * * aws s3 sync --delete /var/www/html/wp-content/uploads s3://prodbucket1216
*/2 * * * * aws s3 sync --delete /var/www/html/ s3://recbucket1216

~
```

Step 13 : assign cron job for recovery server.

```
#crontab -e
```

Add cronjob.

```
root@ip-172-31-85-62:~#
*/2 * * * * aws s3 sync --delete s3://prodbucket1216 /var/www/html/wp-content/uploads
*/2 * * * * aws s3 sync --delete s3://recbucket1216 /var/www/html/
~
```

Step 14: create load balancer for production server.

Create load balancer – choose classic load balancer – set LB name: production-lb.

Create Classic Load Balancer info

The Classic Load Balancer distributes incoming application traffic across multiple EC2 instance targets in multiple Availability Zones. This increases the fault tolerance of your applications. Elastic Load Balancing detects unhealthy instances and routes traffic only to healthy instances.

▶ How Classic Load Balancers work

Basic configuration

Load balancer name
Name must be unique within your AWS account and can't be changed after the load balancer is created.

A maximum of 32 alphanumeric characters including hyphens are allowed, but the name must not begin or end with a hyphen.

Network mapping info

The load balancer routes traffic to targets in the selected subnets, and in accordance with your network settings.

VPC Info
Select the virtual private cloud (VPC) for your targets or you can create a new VPC. Only VPCs with an internet gateway are available for selection. The selected VPC cannot be changed after the load balancer is created. When selecting a VPC for your load balancer, ensure each subnet has a CIDR block with at least a /27 bitmask and at least 8 free IP addresses. [Learn more](#)

-
vpc-0f684e0065eeb33aa
IPv4: 172.31.0.0/16

Mappings
Select at least one Availability Zone and one subnet for each zone. We recommend selecting at least two Availability Zones. The load balancer will route traffic only to targets in the selected Availability Zones. Availability Zones that are not supported by the load balancer or the VPC are not available for selection.

us-east-1a (use1-az6)
Subnet
 default-pub

IPv4 address
Assigned by AWS

us-east-1c (use1-az2)
Subnet
 IPv4 address
Assigned by AWS

CloudShell Feedback © 2023, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences 29°C Mostly cloudy 4:24 PM 12/16/2023

Add a health check ping path.
In instances, add production server.

The screenshot shows the 'Create CLB Wizard' step for a new load balancer. In the 'Health checks' section, a ping target is defined with 'Ping protocol' set to 'HTTP' and 'Ping port' set to '80'. The 'Ping path' is '/healthy.html'. Below this, there's a link to 'Advanced health check settings'. In the 'Instances (1)' section, a single EC2 instance is registered as a target. The instance ID is 'i-059020469e913a811', it has the name 'Production server', and its state is 'Running'. It is associated with the security group 'ec2-rds-1, ec2-rds-3, all traffic'. At the bottom of the page, the AWS navigation bar includes CloudShell, Feedback, and various service icons.

Step 15: create load balancer for recovery server.

Create load balancer – choose classic load balancer – set LB name : recovery-lb.

The screenshot shows the 'Create Classic Load Balancer' wizard. In the 'Basic configuration' section, the 'Load balancer name' is set to 'recovery-lb'. The 'Scheme' dropdown is set to 'Internet-facing', which is described as routing requests from clients over the internet to targets. There is also an option for 'Internal' which routes requests from clients to targets using private IP addresses. The AWS navigation bar at the bottom includes CloudShell, Feedback, and various service icons.

RDS | us-east-1 Instances | Create classic Certificate Manager | Certificate M... | S3 buckets | Edit Post "We... | Add New Post | +

us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#CreateCLBWizard:

aws Services Search [Alt+S]

EC2 Route 53 VPC

Network mapping Info

The load balancer routes traffic to targets in the selected subnets, and in accordance with your network settings.

VPC Info

Select the virtual private cloud (VPC) for your targets or you can [create a new VPC](#). Only VPCs with an internet gateway are available for selection. The selected VPC cannot be changed after the load balancer is created. When selecting a VPC for your load balancer, ensure each subnet has a CIDR block with at least a /27 bitmask and at least 8 free IP addresses. [Learn more](#)

-
vpc-0f684e0065eeb33aa
IPv4: 172.31.0.0/16

Mappings

Select at least one Availability Zone and one subnet for each zone. We recommend selecting at least two Availability Zones. The load balancer will route traffic only to targets in the selected Availability Zones. Availability Zones that are not supported by the load balancer or the VPC are not available for selection.

us-east-1a (use1-az6)

Subnet: subnet-0c58d4e5698c8ad12 default-pub ▾

IPv4 address
Assigned by AWS

us-east-1c (use1-az2)

Subnet: subnet-05a24dd1c1204040e ▾

IPv4 address
Assigned by AWS

CloudShell Feedback © 2023, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

29°C Mostly cloudy

Add a health check ping path.
In instances, add recovery server.

The screenshot shows the AWS EC2 Load Balancers console. On the left, there's a navigation sidebar with options like EC2 Dashboard, EC2 Global View, Events, Console-to-Code, Instances, and more. The main area is titled "Load balancers (2)" and contains a table with two rows. The first row is for "recovery-lb" and the second for "production-lb". The columns include Name, DNS name, State, VPC ID, Availability Zones, and Type. Both load balancers are listed as classic type and have two availability zones.

Load balancers for both production & recovery servers have been created successfully.

Step 16 : create record in route 53.

In route 53 service - create record : give a record name - select a record type - enable alias - in route traffic to : select alias to application and classic load balancer - select the region - choose the created production-lb - in routing policy : select simple routing policy.

The screenshot shows the AWS Route 53 "Create record" page. The URL in the browser is "us-east-1.console.aws.amazon.com/route53/v2/hostedzones?region=us-east-1#CreateRecordSet/Z09274171OJN4XFRANYNP". The page has a "Quick create record" form. In the "Record name" field, "subdomain" is entered, and the "Record type" is set to "A – Routes traffic to an IPv4 address and some AWS resources". The "Alias" option is selected, and the "Route traffic to" dropdown is set to "Alias to Application and Classic Load Balancer". The "US East (N. Virginia)" region is chosen. The target for the alias is "dualstack.production-lb-925634824.us-east-1.elb.amazonaws.com". The "Routing policy" is set to "Simple routing". At the bottom, there are "Evaluate target health" and "Yes" checkboxes.

The screenshot shows the AWS Route 53 service console. On the left, a sidebar navigation includes 'Route 53', 'Dashboard', 'Hosted zones' (selected), 'Health checks', 'IP-based routing', 'Traffic flow', 'Domains', 'Resolver', and 'Query logging'. The main area displays the 'Hosted zones' page for 'borutomokshi.world'. Under 'Records', there are four entries:

- borutomo... A Simple**: Alias Yes, Value/dualstack.production-lb-925...
- borutomo... NS Simple**: Alias No, Value/ns-41.awsdns-05.com, ns-697.awsdns-23.net, ns-1229.awsdns-25.org, ns-1922.awsdns-48.co.uk.
- ns-41.awsdns-05.com**
- ns-697.awsdns-23.net**
- ns-1229.awsdns-25.org**
- ns-1922.awsdns-48.co.uk**

A modal window titled 'Create record' is open, showing fields for 'Record name' (recovery), 'Record type' (A), 'Alias' (selected), 'Route traffic to' (Alias to Application and Classic Load Balancer), 'Region' (US East (N. Virginia)), 'Value' (dualstack.recovery-lb-1778841024.us-east-1.elb.amazonaws.com), and 'Routing policy' (Simple). The 'Create record' button is highlighted in orange.

Step 17: create another record on route 53.

In route 53 service - create record : give a record name(recovery) - select a record type - enable alias - in route traffic to : select alias to application and classic load balancer - select the region - choose the created recovery-lb - in routing policy : select simple routing policy.

The screenshot shows the 'Create record' wizard for the 'Create record' step. The 'Quick create record' section is displayed, with the following configuration:

- Record name**: recovery
- Record type**: A – Routes traffic to an IPv4 address and some AWS resources
- Alias**: Selected
- Route traffic to**: Alias to Application and Classic Load Balancer
- Region**: US East (N. Virginia)
- Value**: dualstack.recovery-lb-1778841024.us-east-1.elb.amazonaws.com
- Routing policy**: Simple

The 'Evaluate target health' checkbox is checked. The 'Switch to wizard' link is visible at the top right of the wizard interface.

Step 18 : add certificate in Load balancer for production and recovery.

- In production load balancer - click actions - select manage listeners - edit - in listener protocol: choose HTTPS - in SSL certificate - edit - choose certificate form ACM - save.

- In recovery load balancer - click actions - select manage listeners - edit - in listener protocol: choose HTTPS - in SSL certificate - edit - choose certificate from ACM - save.

Manage listeners

Load balancer details: recovery-lb

Listeners (1)

Listener protocol	Port	Instance protocol	Instance port	Security policy	Default SSL/TLS certificate	Cookie stickiness
HTTPS	443	HTTP	80	ELBSecurityPolicy-2016-08	ACM:borutomokshi.world	Disabled

Add listener

Server-side tasks and status

Certificate has been added for both load balancers.

Step 19: verify whether both servers are synced and https certificate is issued.

1. Hit as <https://borutomokshi.world>

PRODUCTION

Production & recovery sync

Études is a pioneering firm that seamlessly merges creativity and functionality to redefine architectural excellence.

About us

2. Hit as <https://recovery.borutomokshi.world>

← → G recovery.borutomokshi.world Incognito : 

PRODUCTION

Production & recovery sync

Études is a pioneering firm that seamlessly merges creativity and functionality to redefine architectural excellence.

[About us](#)

