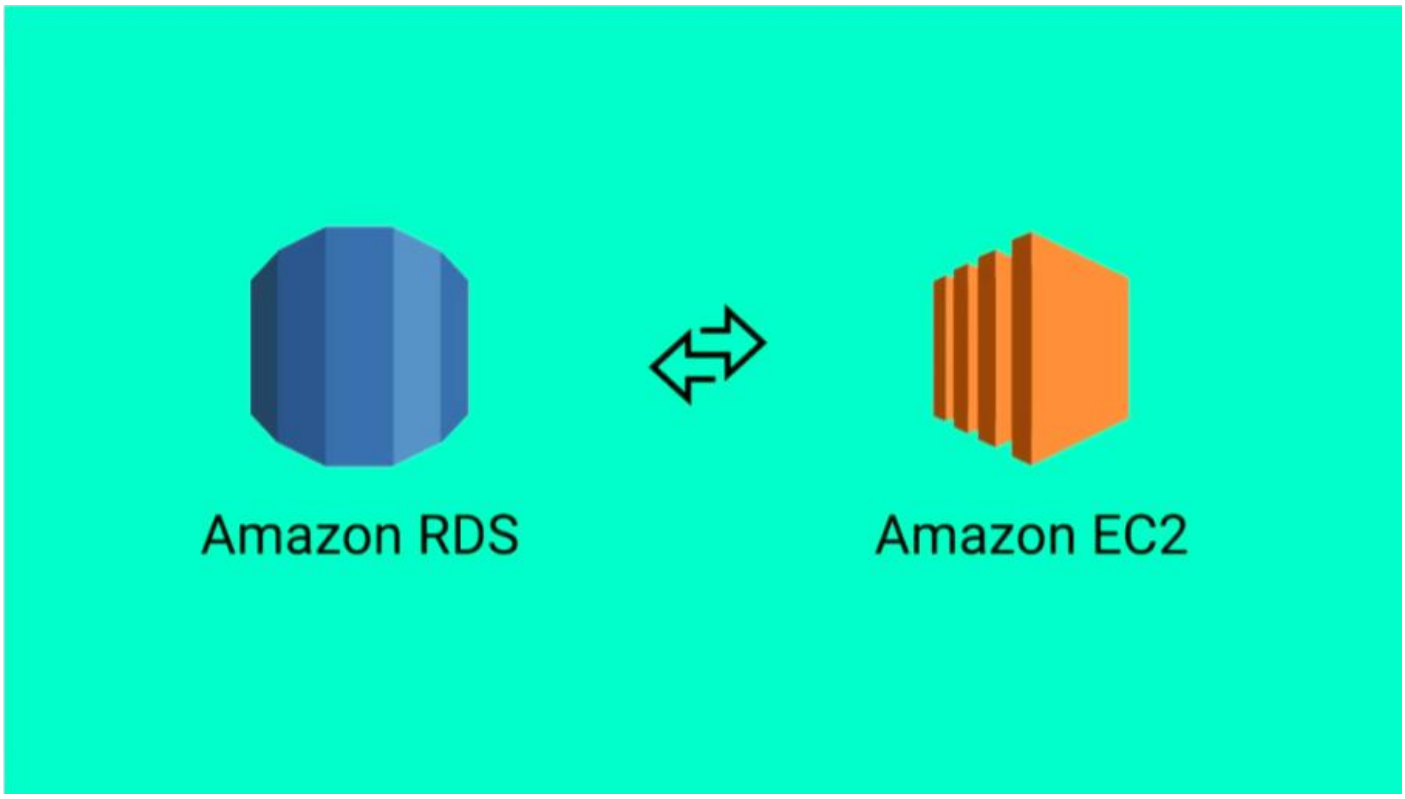
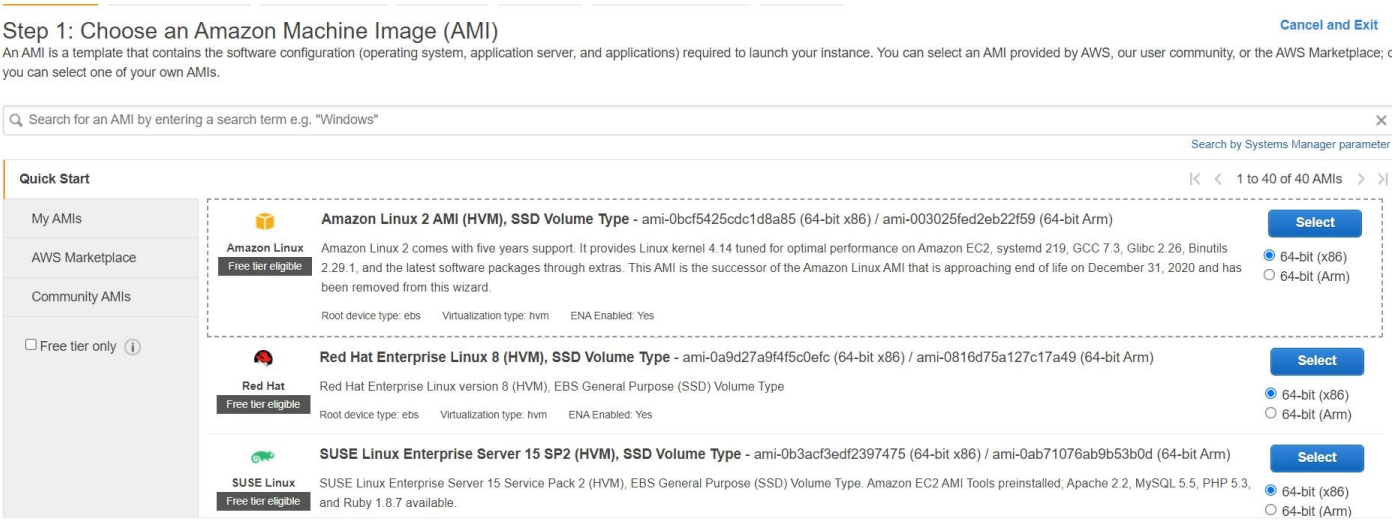
Task 18:



Integrating Amazon EC2 with Amazon RDS using Wordpress

1.Creating Ec2 Instance



1. Choose AMI | 2. Choose Instance Type | 3. Configure Instance | 4. Add Storage | 5. Add Tags | 6. Configure Security Group | 7. Review

Step 2: Choose an Instance Type

Amazon EC2 provides a wide selection of instance types optimized to fit different use cases. Instances are virtual servers that can run applications. They have varying combinations of CPU, memory, storage, and networking capacity, and give you the flexibility to choose the appropriate mix of resources for your applications. Learn more about instance types and how they can meet your computing needs.

Filter by: All instance families | Current generation | Show/Hide Columns

Currently selected: t2.micro (- ECUs, 1 vCPUs, 2.5 GHz, -, 1 GiB memory, EBS only)

	Family	Type	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance	IPv6 Support
☐	t2	t2.nano	1	0.5	EBS only	-	Low to Moderate	Yes
☑	t2	t2.micro (Free tier eligible)	1	1	EBS only	-	Low to Moderate	Yes
☐	t2	t2.small	1	2	EBS only	-	Low to Moderate	Yes
☐	t2	t2.medium	2	4	EBS only	-	Low to Moderate	Yes
☐	t2	t2.large	2	8	EBS only	-	Low to Moderate	Yes
☐	t2	t2.xlarge	4	16	EBS only	-	Moderate	Yes
☐	t2	t2.2xlarge	8	32	EBS only	-	Moderate	Yes
☐	t3	t3.nano	2	0.5	EBS only	Yes	Up to 5 Gigabit	Yes

Cancel | Previous | Review and Launch | Next: Configure Instance Details

1. Choose AMI | 2. Choose Instance Type | 3. Configure Instance | 4. Add Storage | 5. Add Tags | 6. Configure Security Group | 7. Review

Step 3: Configure Instance Details

Configure the instance to suit your requirements. You can launch multiple instances from the same AMI, request Spot instances to take advantage of the lower pricing, assign an access management role to the instance, and more.

Number of instances	1 Launch into Auto Scaling Group
Purchasing option	☐ Request Spot instances
Network	vpc-cf03e3a4 (default) Create new VPC
Subnet	No preference (default subnet in any Availability Zone) Create new subnet
Auto-assign Public IP	Use subnet setting (Enable)
Placement group	☐ Add instance to placement group
Capacity Reservation	Open
Domain join directory	No directory Create new directory
IAM role	None Create new IAM role
Shutdown behavior	Stop
Stop - Hibernate behavior	☐ Enable hibernation as an additional stop behavior
Enable termination protection	☐ Protect against accidental termination

Cancel | Previous | Review and Launch | Next: Add Storage

2.Creating RDS Instance

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

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.

Wordpressdb

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

☐ 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 @ (at sign).

Confirm password Info

••••••••

DB instance class

DB instance class Info

Choose a DB instance class that meets your processing power and memory requirements. The DB instance class options below are limited to those supported by the engine you selected above.

◉ Standard classes (includes m classes)

○ Memory optimized classes (includes r and x classes)

○ Burstable classes (includes t classes)

db.m6g.large ▼

2 vCPUs 8 GiB RAM Network: 4,750 Mbps

Include previous generation classes

Storage

Storage type Info

Provisioned IOPS (SSD) ▼

Allocated storage

100 GiB

Minimum: 100 GiB, Maximum: 65,536 GiB

Provisioned IOPS Info

3000 IOPS

Minimum: 1,000 IOPS, Maximum: 80,000 IOPS

Your actual IOPS might vary from the amount that you provisioned based on your database workload and instance type. Learn more

Storage autoscaling Info

Provides dynamic scaling support for your database's storage based on your application's needs.

- [x] Enable storage autoscaling
  Enabling this feature will allow the storage to increase once the specified threshold is exceeded.

Maximum storage threshold Info

Charges will apply when your database autoscales to the specified threshold

1000 GiB

Minimum: 101 GiB, Maximum: 65,536 GiB

Availability & durability

Multi-AZ deployment Info

- (•) Create a standby instance (recommended for production usage)
  Creates a standby in a different Availability Zone (AZ) to provide data redundancy, eliminate I/O freezes, and minimize latency spikes during system backups.
- ( ) Do not create a standby instance

Virtual private cloud (VPC) Info

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

Default VPC (vpc-cf03e3a4)

Only VPCs with a corresponding DB subnet group are listed.

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

Subnet group Info

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

default-vpc-cf03e3a4

Public access Info

- ( ) 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.
- (•) No
  RDS will not assign a public IP address to the database. Only Amazon EC2 instances and devices inside the VPC can connect to your database.

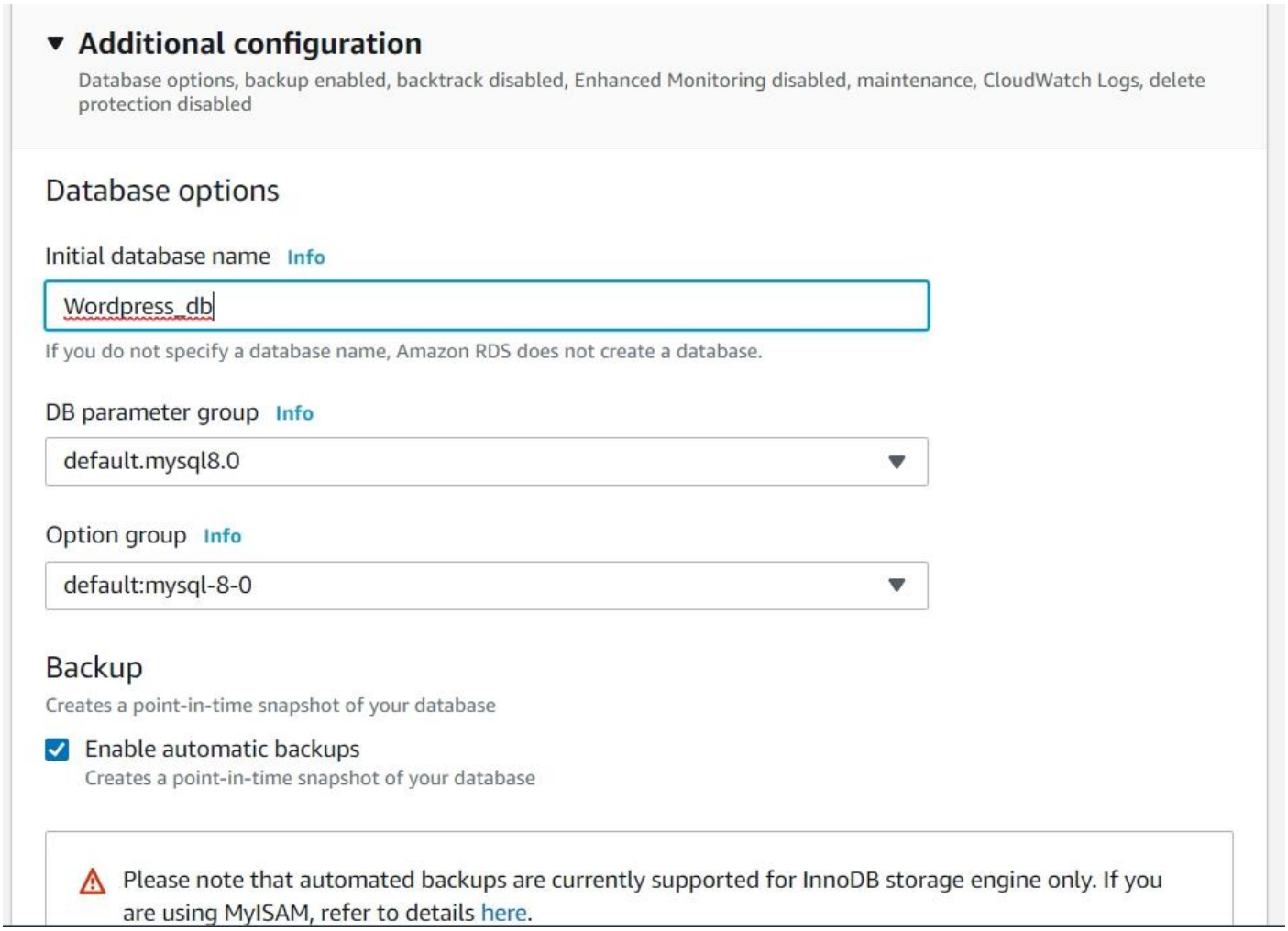
VPC security group

Choose a VPC security group to allow access to your database. Ensure that the security group rules allow the appropriate incoming traffic.

- (•) Choose existing
  Choose existing VPC security groups
- ( ) Create new
  Create new VPC security group

Existing VPC security groups

Choose VPC security groups



3.Installing Dependencies

```
[root@ip-172-31-32-248 ec2-user]# yum install php-mysqlnd -y
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
Resolving Dependencies
--> Running transaction check
---> Package php-mysqlnd.x86_64 0:5.4.16-46.amzn2.0.2 will be installed
--> Processing Dependency: php-pdo(x86-64) = 5.4.16-46.amzn2.0.2 for package: php-mysqlnd-5.4.16-46.amzn2.0.2.x86_64
--> Running transaction check
---> Package php-pdo.x86_64 0:5.4.16-46.amzn2.0.2 will be installed
--> Processing Dependency: php-common(x86-64) = 5.4.16-46.amzn2.0.2 for package: php-pdo-5.4.16-46.amzn2.0.2.x86_64
--> Running transaction check
---> Package php-common.x86_64 0:5.4.16-46.amzn2.0.2 will be installed
--> Processing Dependency: libzip.so.2()(64bit) for package: php-common-5.4.16-46.amzn2.0.2.x86_64
--> Running transaction check
---> Package libzip010-compat.x86_64 0:0.10.1-9.amzn2.0.5 will be installed
--> Finished Dependency Resolution

Dependencies Resolved
```

```
[root@ip-172-31-32-248 ec2-user]# yum install php-fpm -y
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
Resolving Dependencies
--> Running transaction check
---> Package php-fpm.x86_64 0:5.4.16-46.amzn2.0.2 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

================================================================================
 Package                Arch                   Version
================================================================================
Installing:
 php-fpm                x86_64                 5.4.16-46.amzn2.0.2

Transaction Summary
================================================================================
Install  1 Package

Total download size: 1.4 M
Installed size: 4.5 M
Downloading packages:
php-fpm-5.4.16-46.amzn2.0.2.x86_64.rpm
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
```

```
[root@ip-172-31-32-248 ec2-user]# yum install mariadb-server -y
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
Resolving Dependencies
--> Running transaction check
---> Package mariadb-server.x86_64 1:5.5.68-1.amzn2 will be installed
--> Processing Dependency: mariadb(x86-64) = 1:5.5.68-1.amzn2 for package: 1:mariadb-server
--> Processing Dependency: perl-DBI for package: 1:mariadb-server-5.5.68-1.amzn2.x86_64
--> Processing Dependency: perl-DBD-MySQL for package: 1:mariadb-server-5.5.68-1.amzn2.x86_
--> Processing Dependency: perl(Data::Dumper) for package: 1:mariadb-server-5.5.68-1.amzn2.
--> Processing Dependency: perl(DBI) for package: 1:mariadb-server-5.5.68-1.amzn2.x86_64
--> Running transaction check
---> Package mariadb.x86_64 1:5.5.68-1.amzn2 will be installed
---> Package perl-DBD-MySQL.x86_64 0:4.023-6.amzn2 will be installed
---> Package perl-DBI.x86_64 0:1.627-4.amzn2.0.2 will be installed
--> Processing Dependency: perl(RPC::PlServer) >= 0.2001 for package: perl-DBI-1.627-4.amzn
--> Processing Dependency: perl(RPC::PlClient) >= 0.2000 for package: perl-DBI-1.627-4.amzn
---> Package perl-Data-Dumper.x86_64 0:2.145-3.amzn2.0.2 will be installed
--> Running transaction check
---> Package perl-PlRPC.noarch 0:0.2020-14.amzn2 will be installed
--> Processing Dependency: perl(Net::Daemon) >= 0.13 for package: perl-PlRPC-0.2020-14.amzn
--> Processing Dependency: perl(Net::Daemon::Test) for package: perl-PlRPC-0.2020-14.amzn2.
```

```
[root@ip-172-31-32-248 ec2-user]# yum install httpd wget -y
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
amzn2-core
Package wget-1.14-18.amzn2.1.x86_64 already installed and latest version
Resolving Dependencies
--> Running transaction check
---> Package httpd.x86_64 0:2.4.46-1.amzn2 will be installed
--> Processing Dependency: httpd-tools = 2.4.46-1.amzn2 for package: httpd-2.4.46-1.amz
--> Processing Dependency: httpd-filesystem = 2.4.46-1.amzn2 for package: httpd-2.4.46-
--> Processing Dependency: system-logos-httpd for package: httpd-2.4.46-1.amzn2.x86_64
--> Processing Dependency: mod_http2 for package: httpd-2.4.46-1.amzn2.x86_64
--> Processing Dependency: httpd-filesystem for package: httpd-2.4.46-1.amzn2.x86_64
--> Processing Dependency: /etc/mime.types for package: httpd-2.4.46-1.amzn2.x86_64
--> Processing Dependency: libaprutil-1.so.0()(64bit) for package: httpd-2.4.46-1.amzn2
--> Processing Dependency: libapr-1.so.0()(64bit) for package: httpd-2.4.46-1.amzn2.x86
--> Running transaction check
---> Package apr.x86_64 0:1.6.3-5.amzn2.0.2 will be installed
---> Package apr-util.x86_64 0:1.6.1-5.amzn2.0.2 will be installed
--> Processing Dependency: apr-util-bdb(x86-64) = 1.6.1-5.amzn2.0.2 for package: apr-ut
---> Package generic-logos-httpd.noarch 0:18.0.0-4.amzn2 will be installed
---> Package httpd-filesystem.noarch 0:2.4.46-1.amzn2 will be installed
```

```
[root@ip-172-31-32-248 ec2-user]# amazon-linux-extras install php7.3
Installing php-pdo, php-fpm, php-mysqlnd, php-cli, php-json
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
Cleaning repos: amzn2-core amzn2extra-docker amzn2extra-php7.3
12 metadata files removed
4 sqlite files removed
0 metadata files removed
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
amzn2-core
amzn2extra-docker
amzn2extra-php7.3
(1/7): amzn2-core/2/x86_64/group_gz
(2/7): amzn2-core/2/x86_64/updateinfo
(3/7): amzn2extra-php7.3/2/x86_64/primary_db
(4/7): amzn2extra-docker/2/x86_64/updateinfo
(5/7): amzn2extra-docker/2/x86_64/primary_db
```

4.Download the wordpress file and unzip it and copy in /var/www/html

```
[root@ip-172-31-32-248 ec2-user]# cd /var/www/html
[root@ip-172-31-32-248 html]# ls
[root@ip-172-31-32-248 html]# wget https://wordpress.org/latest.tar.gz
--2021-04-25 13:50:22--  https://wordpress.org/latest.tar.gz
Resolving wordpress.org (wordpress.org)... 198.143.164.252
Connecting to wordpress.org (wordpress.org)|198.143.164.252|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 15750352 (15M) [application/octet-stream]
Saving to: 'latest.tar.gz'

100%[======================================================================================

2021-04-25 13:50:27 (3.60 MB/s) - 'latest.tar.gz' saved [15750352/15750352]
```

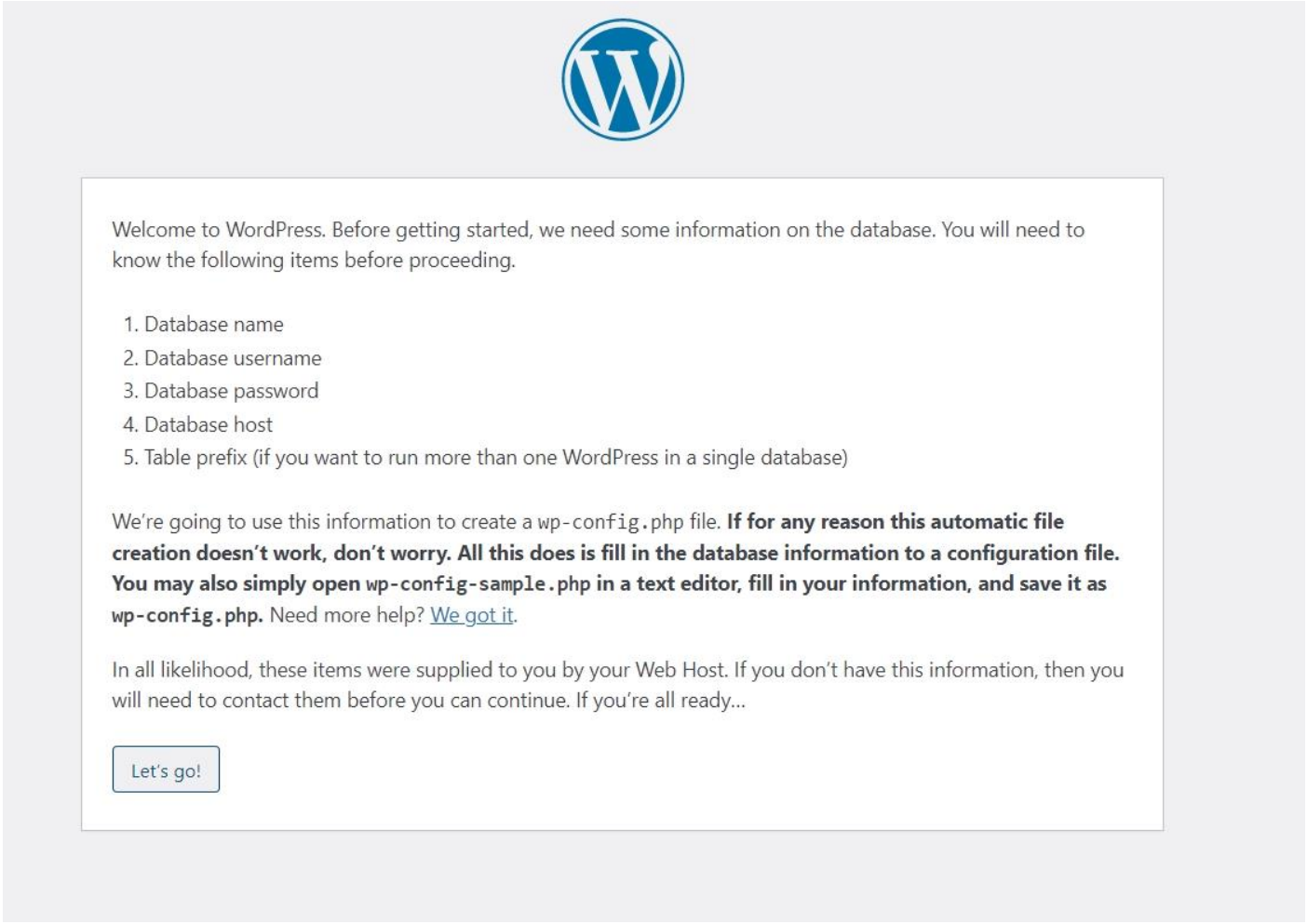
```
[root@ip-172-31-32-248 html]# ls
latest.tar.gz
[root@ip-172-31-32-248 html]# tar xf latest.tar.gz
[root@ip-172-31-32-248 html]# ls
latest.tar.gz  wordpress
[root@ip-172-31-32-248 html]#
```

5.Start Server

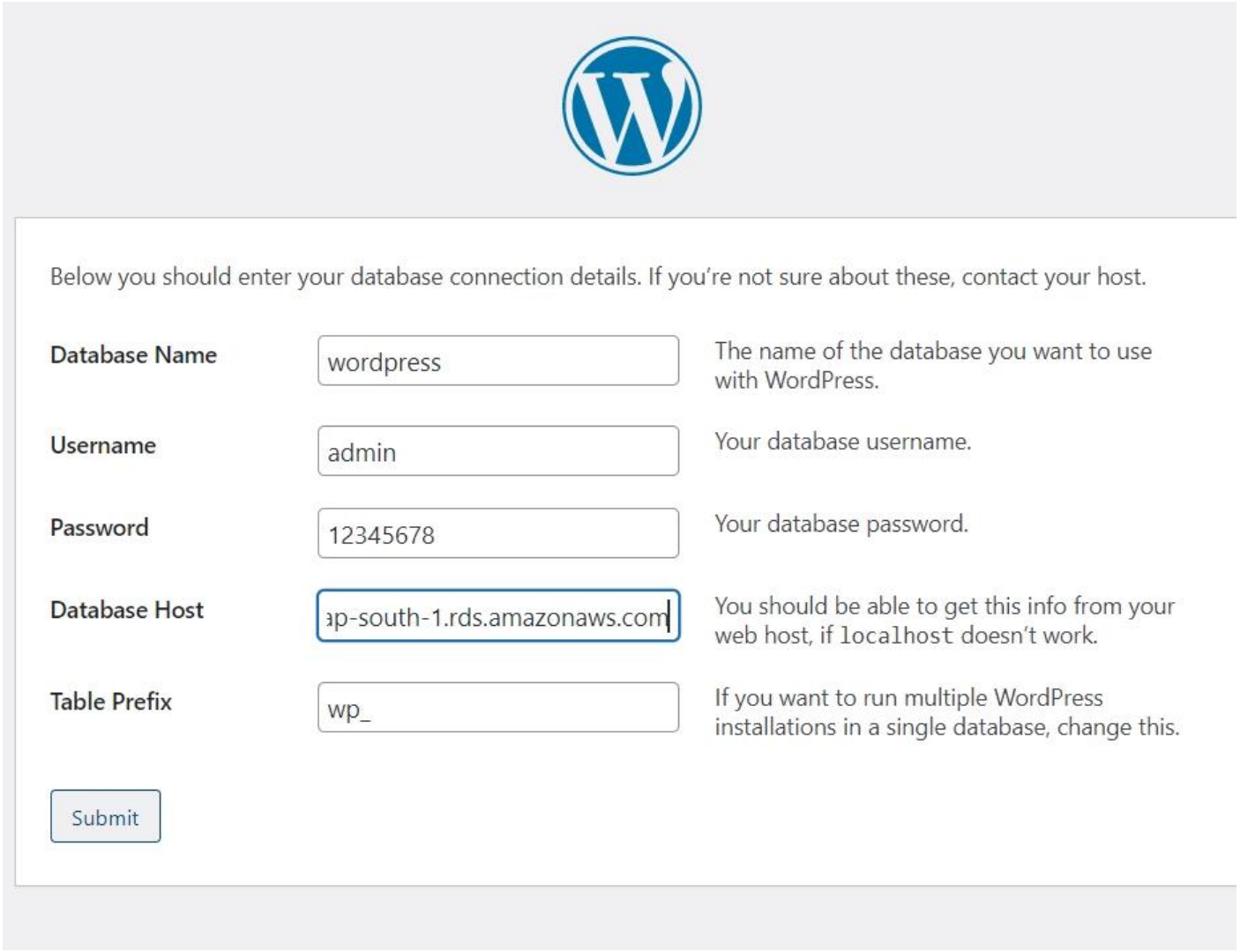
```
[root@ip-172-31-32-248 html]# cd wordpress
[root@ip-172-31-32-248 wordpress]# ls
index.php    wp-activate.php     wp-comments-post.php  wp-cron.php        wp-load.php   wp-settings.php   xmlrpc.php
license.txt  wp-admin            wp-config-sample.php  wp-includes        wp-login.php  wp-signup.php
readme.html  wp-blog-header.php  wp-content            wp-links-opml.php  wp-mail.php   wp-trackback.php
[root@ip-172-31-32-248 wordpress]# systemctl start httpd
[root@ip-172-31-32-248 wordpress]# systemctl status httpd
● httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; disabled; vendor preset: disabled)
  Drop-In: /usr/lib/systemd/system/httpd.service.d
           └─php-fpm.conf
   Active: active (running) since Sun 2021-04-25 13:52:28 UTC; 4s ago
     Docs: man:httpd.service(8)
 Main PID: 4168 (httpd)
   Status: "Processing requests..."
   CGroup: /system.slice/httpd.service
           ├─4168 /usr/sbin/httpd -DFOREGROUND
           ├─4174 /usr/sbin/httpd -DFOREGROUND
           ├─4175 /usr/sbin/httpd -DFOREGROUND
           ├─4176 /usr/sbin/httpd -DFOREGROUND
           ├─4177 /usr/sbin/httpd -DFOREGROUND
           └─4178 /usr/sbin/httpd -DFOREGROUND

Apr 25 13:52:28 ip-172-31-32-248.ap-south-1.compute.internal systemd[1]: Starting The Apache HTTP Server...
Apr 25 13:52:28 ip-172-31-32-248.ap-south-1.compute.internal systemd[1]: Started The Apache HTTP Server.
```

5.Start Wordpress by using URL : [ipaddress/wordpress/wp-admin/setup-config.php]



6.Wordpress Login



Wordpress Dashboard Ready