**Deploying Fullstack Web Application Over AWS Cloud**

**Requirements:**

* **Need to have a valid AWS account**

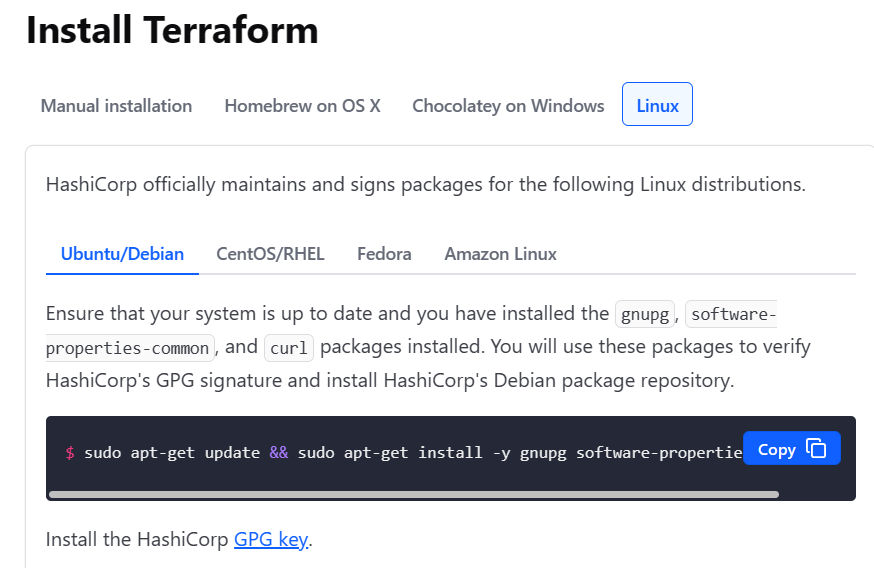
**Technology used:**

* **RDS Mysql**
* **Wordpress**
* **Terraform**
* **Apache Tomcat**
* **PHP**
* **AWS ALB**

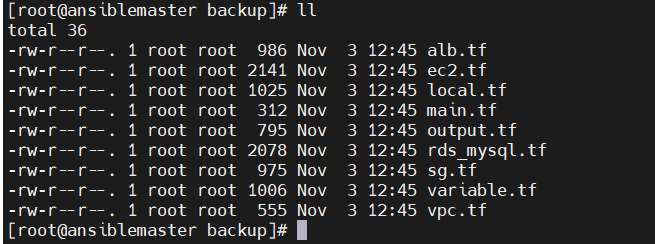
1. **Clone the Repository on your local Server:**

***git clone*** [*https://github.com/amithtg-199/Fullstack-Web-Application-Over-AWS-Cloud.git*](https://github.com/amithtg-199/Fullstack-Web-Application-Over-AWS-Cloud.git)

1. **Use Terraform to create standard 3 Tier network with standard Security practices.**
2. **Install** [Terraform](https://developer.hashicorp.com/terraform/tutorials/aws-get-started/install-cli) **on local server as per the provided link.**

****

1. **From the Cloned project navigate into terraform\_State\_files folder where you can see below files.**

****

**Before Starting with deployment need to setup the environment and generate ssh-keygen for ssh.**

1. **On the local server install install aws cli, run below commands:**

***sudo apt-get update***

***sudo apt install awscli***

1. **Then Input the AWS IAM user AccesssKey ID and Secret Access Key, run below command enter the required data prompted.**

***aws configure***

**AWS Access Key ID [None]: <AWS\_IAM\_USER\_ACCESS\_KEY\_ID>**

**AWS Secret Access Key ID [None]: <AWS\_IAM\_USER\_SEC\_ACCESS\_KEY>**

1. **Now Create ssh-keygen for ssh authentication for EC2 machines (Remember set the name of the file as given below only)**

**ssh-keygen -t rsa -b 4096**

**Enter file in which to save the key (/home/ubuntu/.ssh/id\_rsa): ec2\_rsa**

**Note**: do not pass any passphrase

Copy the Private Key ec2\_rsa to ~/.ssh/ path and keep the ec2\_rsa.pub under the terraform\_state\_files along with the other state files.

1. Now initialize the Terraform and import the providers required

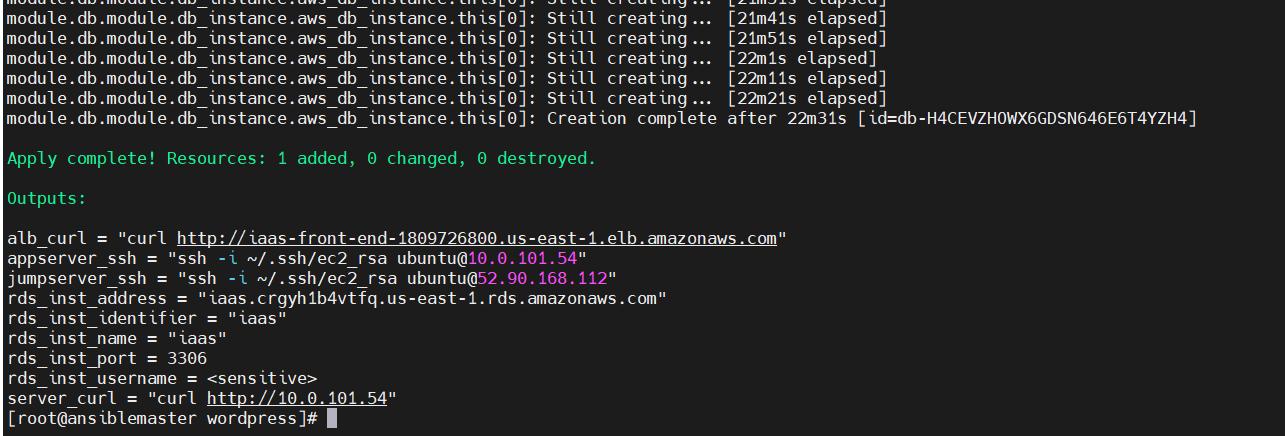
*terraform init*

*terraform plan*

If no error in above 2 commands now run below to create the Infra:

*terraform apply*

Once completed it show something like this:



Now the Infra is setup next we need to setup Application.

1. Loging into Jump Server and then accessing application server

On local terminal run as below:

*ssh -i ~/.ssh/ec2\_rsa ubuntu@<Jump\_server\_Public\_IP>*

Copy the content of ec2\_rsa content and create the same inside the Jump\_server and provide 400 permission.

*vi ec2\_rsa*

##Enter the Required data##

*:wq!*

*chmod 400 ec2\_rsa*

*ssh -i “ec2\_rsa” ubuntu@<Application\_server\_private\_ip>*

1. Configure WordPress Application.
2. All the required dependencies are already added in the EC2 TF state file, verify them once running below command:

*sudo apt install apache2 \*

*ghostscript \*

*libapache2-mod-php \*

*php \*

*php-bcmath \*

*php-curl \*

*php-imagick \*

*php-intl \*

*php-json \*

*php-mbstring \*

*php-mysql \*

*php-xml \*

*php-zip*

1. Install Wordpress:

*sudo mkdir -p /srv/www*

*sudo chown www-data: /srv/www*

*curl https://wordpress.org/latest.tar.gz | sudo -u www-data tar zx -C /srv/www*

1. Configure Apache for wordpress

*sudo touch /etc/apache2/sites-available/wordpress.conf*

*sudo vi /etc/apache2/sites-available/wordpress.conf*

Add below entry:

*<VirtualHost \*:80>*

*DocumentRoot /srv/www/wordpress*

*<Directory /srv/www/wordpress>*

*Options FollowSymLinks*

*AllowOverride Limit Options FileInfo*

*DirectoryIndex index.php*

*Require all granted*

*</Directory>*

*<Directory /srv/www/wordpress/wp-content>*

*Options FollowSymLinks*

*Require all granted*

*</Directory>*

*</VirtualHost>*

Enable the site:

*sudo a2ensite wordpress*

Enable the URL rewriting:

*sudo a2enmod rewrite*

Disable the default “It Works” site with:

*sudo a2dissite 000-default*

Reload apache2 to apply all changes:

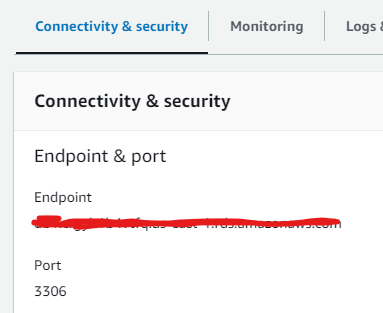
*sudo service apache2 reload*

1. Configure the RDS Mysql DB

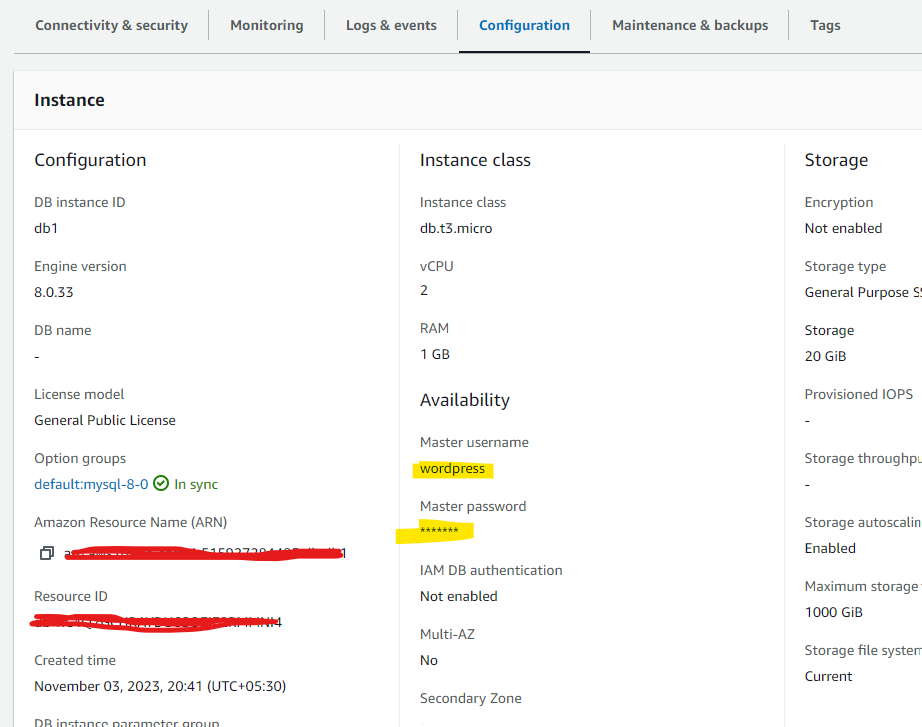
* Login to Application Server and access the Mysql DB:

*mysql -h <RDS\_ENDPOINT> -u <DB\_MASTER\_USER\_NAME> -p*

Obtain the 2 variable from AWS RDS UI



Can Obtain the Password set from the **Secret Manager**



Once Logged in Create Database and User:

CREATE DATABASE wordpress;

CREATE USER 'app '@ '% ' IDENTIFIED BY '<Your\_password> '

GRANT SELECT,INSERT,UPDATE,DELETE,CREATE,DROP,ALTER

ON wordpress.\* TO 'app '@ '% ' ;

FLUSH PRIVILEGES;

quit;

1. Configure WordPress to connect to the database

*sudo -u www-data cp /srv/www/wordpress/wp-config-sample.php /srv/www/wordpress/wp-config.php*

Configure the Connecting values:

*sudo -u www-data sed -i 's/database\_name\_here/wordpress/' /srv/www/wordpress/wp-config.php*

*sudo -u www-data sed -i 's/username\_here/app/' /srv/www/wordpress/wp-config.php*

*sudo -u www-data sed -i 's/password\_here/<your-password>/' /srv/www/wordpress/wp-config.php*

*sudo -u www-data nano /srv/www/wordpress/wp-config.php*

Replace the below lines with the values provided in <https://api.wordpress.org/secret-key/1.1/salt/>

define( 'AUTH\_KEY', 'put your unique phrase here' );

define( 'SECURE\_AUTH\_KEY', 'put your unique phrase here' );

define( 'LOGGED\_IN\_KEY', 'put your unique phrase here' );

define( 'NONCE\_KEY', 'put your unique phrase here' );

define( 'AUTH\_SALT', 'put your unique phrase here' );

define( 'SECURE\_AUTH\_SALT', 'put your unique phrase here' );

define( 'LOGGED\_IN\_SALT', 'put your unique phrase here' );

define( 'NONCE\_SALT', 'put your unique phrase here' );

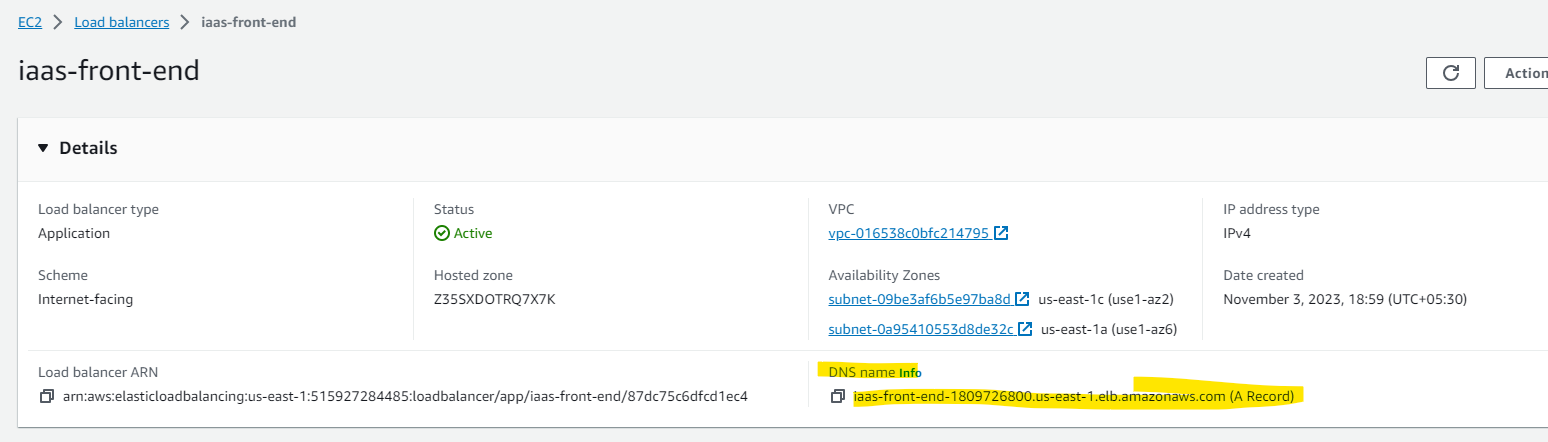
Go to the line and enter ctrl+k will delete a line each time you press the sequence.

Once all are replaced, change the localhost in DB\_HOST to RDS Endpoint.



At last enter *Ctrl+x* followed by *y* and then *enter*

1. Access the Wordpress URL from AWS ALB DNS



1. Final Application:

