1. AWS Terraform:- Here wordpress acts as front end and it should talk with backend mysqldb (two tier architecture)

Text

Description automatically generated

Step1:-we will be having vpc inside that vpc we need to establish subnets,security group, routing table etc..

Here we are creating security groups that will be attached to nic of web server wordpress by allowing traffic in all directions based on subnet ranges.

Text

Description automatically generated

Step2:- we have to create a 2 subnets and attach the security group id to that subnet of wordpress webserver inside it to give security.

Text

Description automatically generated

Step3:- from a wordpress vm it gives a call to script.sh file which is used to install all the packages and other installation required to host the wordpress

Graphical user interface, text, application, email

Description automatically generated

Step4:- Here is my wordpress instance created when we run the above tf file

2.)Azure terraform:- 1.)install docker and git using startup script. create a storage account and container finally push .tfstate on container

Text

Description automatically generated

Step1:- create a subscription\_id in provider which says in which format data should be transmitted and deployed , creation of subnet under common resource group name by attaching subnet range and create public IP through which data should be tarfficed

Text

Description automatically generated

Step2:- create a NSG by attaching security rules and other inbound and outbound connections

And create a network interface which enables the VM to communicate with internet on cloud and on-premises by configuring.

Text

Description automatically generated

Step3:-we have to configure the security group with network interface and create a storage account with a proper location

Text

Description automatically generated

Step4:- finally create Vm by configuring the disk and source image which is ubuntuserver as offering and call the data or all other depencies to be installed

Graphical user interface, text, application

Description automatically generated

This is used to show the installation of git and docker when ever vm is configured.

Graphical user interface, table

Description automatically generated

Storage account with appropriate name

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, application

Description automatically generated

Vm created with appropriate name and docker and git installed

Graphical user interface, text, application, email

Description automatically generated