PROJECT DETAILS

Project Title

Deployment of WordPress Environment using CMAT

Problem statement

You are a DevOps engineer at SL Ltd. Your company is working mostly on WordPress projects. A lot of development hours are lost to perform WordPress setup with all dependencies like PHP, MySQL, etc. The Company wants to automate it with the help of a configuration management tool so that they can follow a standard installation procedure for WordPress and its components whenever a new requirement or client comes in. The below mentioned components should be included:

- PHP
- Nginx/Apache Web Server
- MySQL
- WordPress

Tool required

- Terraform (installed on a controller machine)
- Ansible controller (installed on a controller machine)
- AWS account with security credentials
- For SSH I have created in terraform command code put ssh-key with instance that will create in future
- PHP
- MYSQL
- WORDPRESS
- APACHE2/NGINX

Need configuration on instance after created

- Launch an EC2 instance using Terraform (linkup ssh-key pair)
- Terraform host will connect created ec2 instance via ssh-key-pair.
- Run playbook and install required tools on ec2 instance.
- Setup WordPress on created ec2 instance.
- Install php, MySQL, apache2/nginx and Python in the instance.
- Validate that the packages are installed with WordPress running.

Procedure

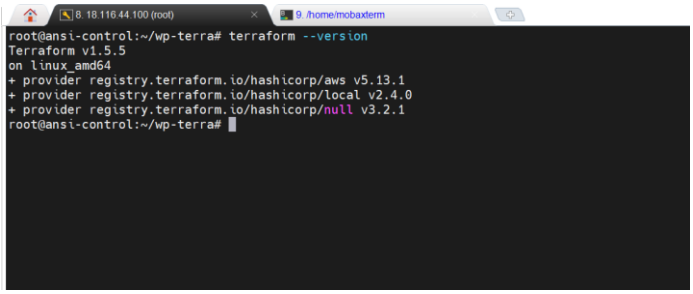
Step 1. Configure the controller node

Installation steps:

- Procedure to install terraform in controller:
  - First need to visit https://developer.hashicorp.com/terraform/downloads?product_intent=terraform
  - Follow the below command to install terraform in linux os at controller node

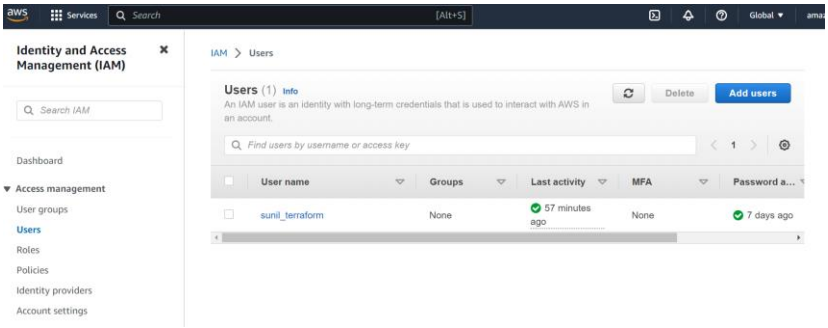
```
wget -O- https://apt.releases.hashicorp.com/gpg | sudo gpg
--dearmor -o /usr/share/keyrings/hashicorp-archive-
keyring.gpg
echo "deb [signed-by=/usr/share/keyrings/hashicorp-archive-
keyring.gpg] https://apt.releases.hashicorp.com
$(lsb_release -cs) main" | sudo tee
/etc/apt/sources.list.d/hashicorp.list
sudo apt update && sudo apt install terraform
```

  - To validate:
    - Terraform version
    - Which terraform

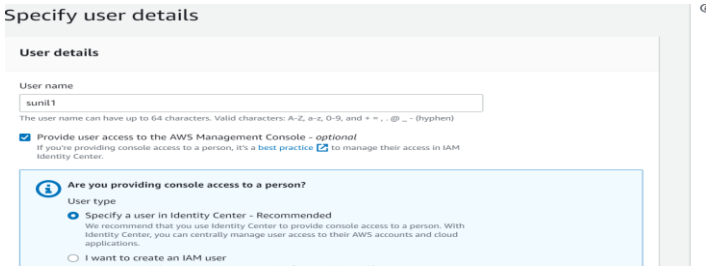


Step 2: Setup AWS "access key" && "secret key" (same followed with below setup)

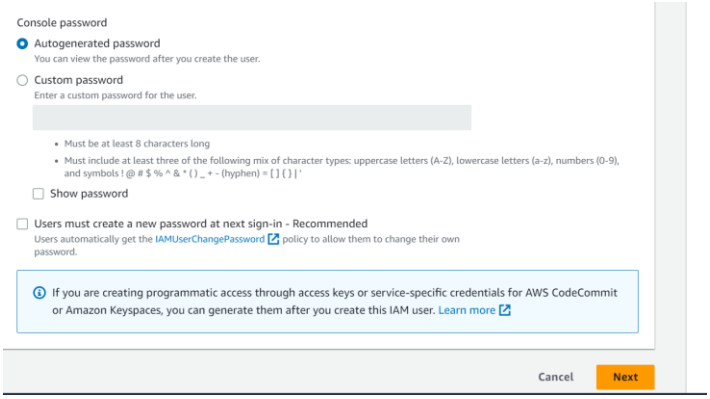
- Procedure to setup access key and secret key:
  - Visit AWS account > go to IAM service > user section > create user



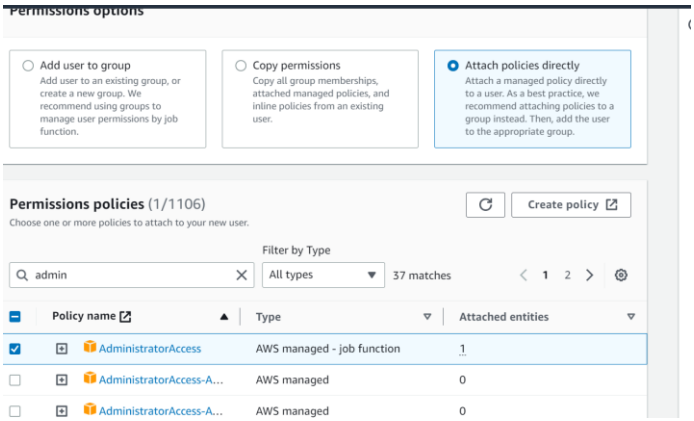
- Click “Add users” in case it not created in my case it is already created



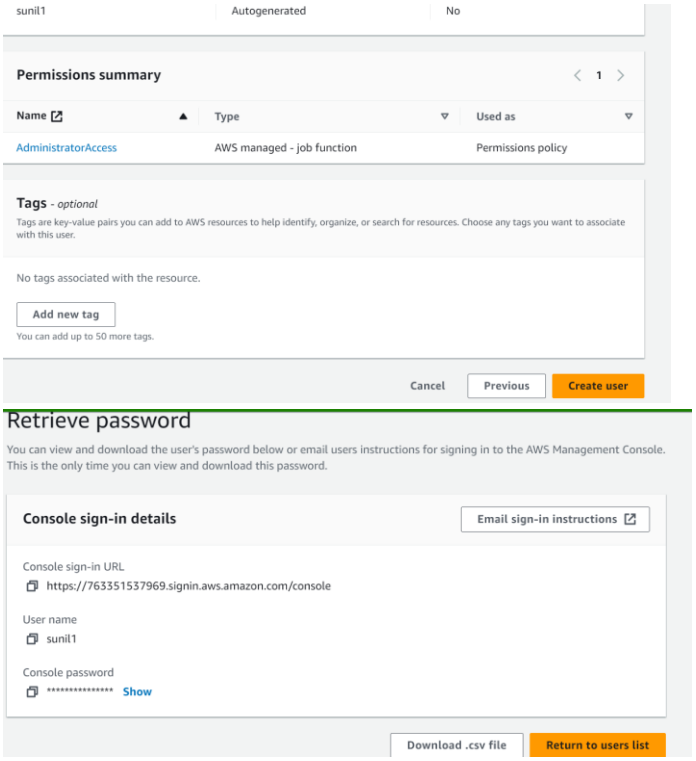
- Click “ I want to create an IAM user” and then then follow as below



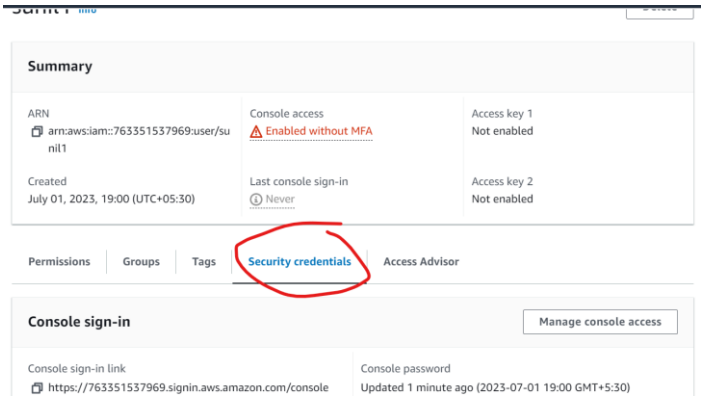
- Click “next” && attach policy as “ AdministratorAccess”



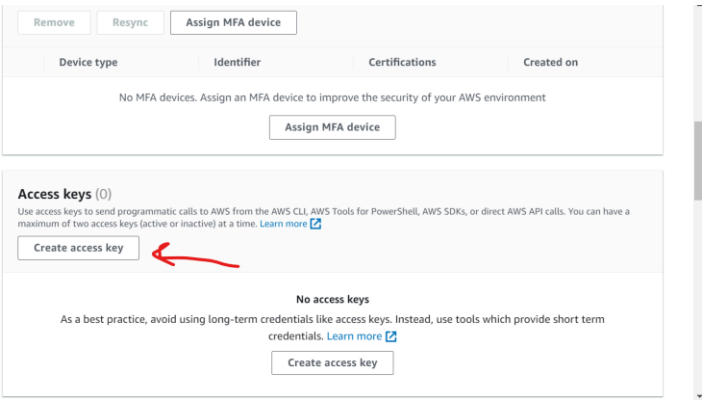
- Now create user



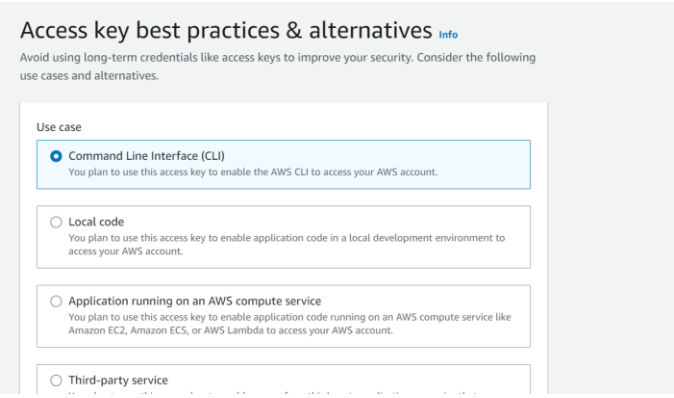
- Now create access and secret key for above user



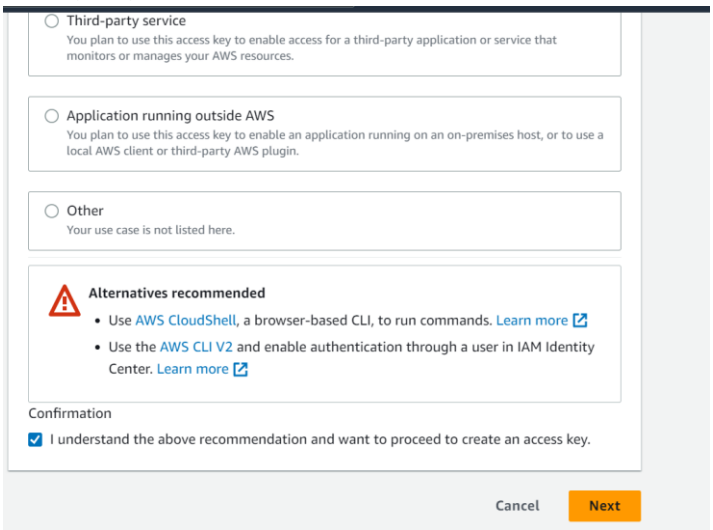
- Click create access key



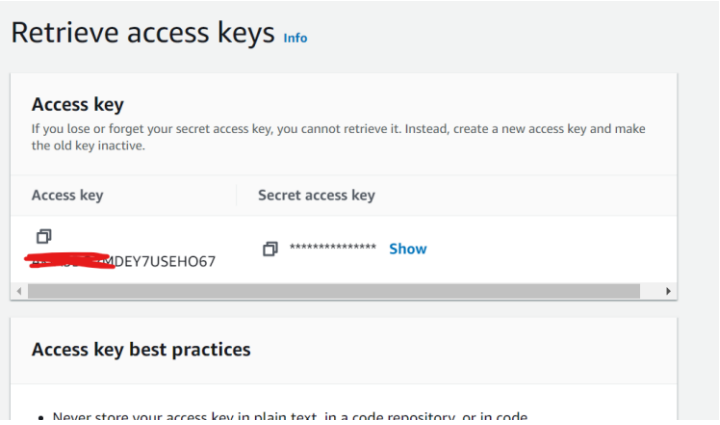
- Select for command line



- Check policy and click next



- Access and secret key generated and need to save for future used



Step3 Setup wordpress via terraform and ansible.

- **Procedure to follow as mentioned below:**
  - **First need to create directory as "wp-terra" in my case**
  - **Use command "mkdir wp-terra" && "cd wp-terra"**

```
root@ansi-control:~/wp-terra#
root@ansi-control:~/wp-terra#
root@ansi-control:~/wp-terra#
root@ansi-control:~/wp-terra#
```

- **I have used roles for install dependent tools like (php, apache, mysql, wordpress)**
- **So in project directory I have created roles, inside roles I have installed roles as below**

```
root@ansi-control:~/wp-terra/roles# ll
total 24
drwxr-xr-x  6 root root 4096 Aug 20 06:00 ./
drwxr-xr-x  6 root root 4096 Aug 20 08:25 ../
drwxr-xr-x 10 root root 4096 Aug 20 05:38 mysql/
drwxr-xr-x 10 root root 4096 Aug 20 05:38 php/
drwxr-xr-x 10 root root 4096 Aug 20 05:38 server/
drwxr-xr-x 10 root root 4096 Aug 20 05:38 wordpress/
root@ansi-control:~/wp-terra/roles#
```

- **On project directory, I have to create terraform files and ansible-playbook to execute different tools by roles on ec2 instance.**

```
root@ansi-control:~/wp-terra# ll
total 64
drwxr-xr-x  6 root root  4096 Aug 20 08:25 ./
drwx------ 14 root root  4096 Aug 20 09:38 ../
drwxr-xr-x  8 root root  4096 Aug 20 08:35 .git/
drwxr-xr-x  3 root root  4096 Aug 20 06:44 .terraform/
-rw-r--r--  1 root root  3468 Aug 20 06:44 .terraform.lock.hcl
-rw-r--r--  1 root root  1668 Aug 20 08:00 main.tf
-rw-r--r--  1 root root   102 Aug 20 08:09 playbook.yml
-rw-r--r--  1 root root   174 Aug 20 08:01 provider.tf
drwxr-xr-x  6 root root  4096 Aug 20 06:00 roles/
-rw-r--r--  1 root root   593 Aug 20 06:46 sg.tf
drwxr-xr-x  2 root root  4096 Aug 20 06:37 templates/
-rw-r--r--  1 root root   181 Aug 20 08:21 terraform.tfstate
-rw-r--r--  1 root root 10886 Aug 20 08:20 terraform.tfstate.backup
-rw-r--r--  1 root root   232 Aug 20 08:01 variable.tf
root@ansi-control:~/wp-terra#
```

- **Vi provider.tf >> press "i" insert for write then write code as mention below**

```
provider "aws" {
  profile = "skadm"
  region  = "us-east-1"
}

resource "aws_key_pair" "key-tfnew" {
  key_name   = "key-tfnew"
  public_key = file("~/.ssh/id_rsa.pub")

}
```

- Then press ":wq!" to save

- Variable.tf files look like below

```
variable "aws_ami_id" {
  # ami for ubuntu
  default = "ami-0261755bbcb8c4a84"
  ## "ami-0261755bbcb8c4a84"
}
variable "ssh_key_pair" {
  default = "~/.ssh/id_rsa"
}

variable "ssh_key_pair_pub" {
  default = "~/.ssh/id_rsa.pub"
}

```

- Security_group.tf file shown below:

```
resource "aws_security_group" "skv-terra" {
  name        = "skv-tf3"
  description = "Allow http https and ssh traffic via terraform"

  ingress {
    from_port   = 80
    to_port     = 80
    protocol    = "tcp"
    cidr_blocks = ["0.0.0.0/0"]
  }
  ingress {
    from_port   = 443
    to_port     = 443
    protocol    = "tcp"
    cidr_blocks = ["0.0.0.0/0"]
  }
  ingress {
    from_port   = 22
    to_port     = 22
    protocol    = "tcp"
    cidr_blocks = ["0.0.0.0/0"]
  }
  egress {
    from_port   = 0
    to_port     = 0
    protocol    = "-1"
    cidr_blocks = ["0.0.0.0/0"]
  }
}

```

- **Hosts file**

```
hosts
[wordpress]
${public_ipaddr}

[wordpress:vars]
ansible_ssh_user = ubuntu
ansible_ssh_private_key_file = ${key_path}
```

- **Main.tf file look like below:**

```
# Main Terraform configuration file

# Define the local variables for the root module
locals {
  ami_id = "ami-0261755bbcb8c4a84"
  ssh_user = "ubuntu"
  key_name = "wpserver"
  private_key_path = "${path.module}/wpserver.pem"
}


# This creates the EC2 instance and makes an initial SSH connection.
resource "aws_instance" "wpserver" {
  ami = var.aws_ami_id
  instance_type = "t2.micro"
  associate_public_ip_address = true
  vpc_security_group_ids = [aws_security_group.skv-terra.id]
  key_name = "key-tfnew"

  tags = {
    Name = "WordPress Server"
  }

  connection {
    type = "ssh"
    host = self.public_ip
    user = local.ssh_user
    private_key = file(var.ssh_key_pair)
    timeout = "4m"
  }

  provisioner "remote-exec" {
    inline = [
      "touch /home/ubuntu/demo-file-from-terraform.txt"
    ]
```

```
  }
}

# Creating a local hosts file for Ansible to use
resource "local_file" "hosts" {
  content = templatefile("${path.module}/templates/hosts",
    {
      public_ipaddr = aws_instance.wpserver.public_ip
      key_path = var.ssh_key_pair
      ansible_user = local.ssh_user
    }
  )
  filename = "${path.module}/hosts"
}

# We will use a null resource to execute the playbook with a local-exec provisioner.

resource "null_resource" "run_playbook" {
  depends_on = [

    # Running of the playbook depends on the successfull creation of the EC2
    # instance and the local inventory file.

    aws_instance.wpserver,
    local_file.hosts,
  ]

  provisioner "local-exec" {
    command = "ANSIBLE_HOST_KEY_CHECKING=False ansible-playbook -u ubuntu -i hosts --private-key ${var.ssh_key_pair} -e 'pub_key=${var.ssh_key_pair_pub}' playbook.yml"
  }
}

```

- **Now I have setup all prerequisites and installed terraform and ansible in controller To run ansible-playbook install wordpress.**
- **Now I have initiated terraform "terraform init".**

```
root@ansi-control:~/wp-terra# terraform init

Initializing the backend...

Initializing provider plugins...
- Reusing previous version of hashicorp/aws from the dependency lock file
- Reusing previous version of hashicorp/local from the dependency lock file
- Reusing previous version of hashicorp/null from the dependency lock file
- Using previously-installed hashicorp/aws v5.13.1
- Using previously-installed hashicorp/local v2.4.0
- Using previously-installed hashicorp/null v3.2.1

Terraform has been successfully initialized!

You may now begin working with Terraform. Try running "terraform plan" to see
any changes that are required for your infrastructure. All Terraform commands
should now work.

If you ever set or change modules or backend configuration for Terraform,
rerun this command to reinitialize your working directory. If you forget, other
commands will detect it and remind you to do so if necessary.
root@ansi-control:~/wp-terra#
```

- **Run the "Terraform plan"**

```
If you ever set or change modules or backend configuration for Terraform,
rerun this command to reinitialize your working directory. If you forget, other
commands will detect it and remind you to do so if necessary.
root@ansi-control:~/wp-terra#
root@ansi-control:~/wp-terra# terraform plan

Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the
following symbols:
  + create

Terraform will perform the following actions:

  # aws_instance.wpserver will be created
  + resource "aws_instance" "wpserver" {
      + ami                                  = "ami-0261755bbcb8c4a84"
      + arn                                  = (known after apply)
      + associate_public_ip_address          = true
      + availability_zone                    = (known after apply)
      + cpu_core_count                       = (known after apply)
      + cpu_threads_per_core                 = (known after apply)
      + disable_api_stop                     = (known after apply)
      + disable_api_termination              = (known after apply)
      + ebs_optimized                        = (known after apply)
      + get_password_data                    = false
      + host_id                              = (known after apply)
      + host_resource_group_arn              = (known after apply)
      + iam_instance_profile                 = (known after apply)
      + id                                   = (known after apply)
      + instance_initiated_shutdown_behavior = (known after apply)
      + instance_lifecycle                   = (known after apply)
      + instance_state                       = (known after apply)
      + instance_type                        = "t2.micro"
```

- **Create ssh-key on controller (prerequisites assuming completed earlier)**

  - Run command " ssh-keygen -t rsa"
  - Copy public id from id_rsa.pub file
  - And paste in code as mentioned below

```
To check for new updates run: sudo apt update

Last login: Sun Jun 25 07:29:26 2023 from 54.253.93.135
ubuntu@ip-172-31-8-112:~$ sudo -i
root@ip-172-31-8-112:~# exit
logout
ubuntu@ip-172-31-8-112:~$ exit
logout
Connection to 3.25.176.67 closed.
root@terraform:~# ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):
/root/.ssh/id_rsa already exists.
Overwrite (y/n)? n
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_rsa
Your public key has been saved in /root/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:oD1NqmInYsmqWwOdtqGIXosCb8lV7aLHz3bOAFlVZE root@terraform
The key's randomart image is:
+---[RSA 3072]----+
|          ..oo   |
|      o o   E    |
|     o +         |
|.. o . o.        |
|.* . . .+S       |
|o   . o++. o .   |
|+o. . .o.oB+ o   |
|=+o. oo.*+o. .   |
|=*o . o+.o+. .   |
+----[SHA256]-----+
root@terraform:~#
support MobaXterm by subscribing to the professional edition here: https://mobaxterm.mobatek.net
```

- **Now final the apply the terraform and create ec2 instance.**

```
8. 18.116.44.100 (root)

Note: You didn't use the -out option to save this plan, so Terraform can't guarantee to take exactl
"terraform apply" now.
root@ansi-control:~/wp-terra#
root@ansi-control:~/wp-terra#
root@ansi-control:~/wp-terra#
root@ansi-control:~/wp-terra# terraform apply

Terraform used the selected providers to generate the following execution plan. Resource actions a
following symbols:
  + create

Terraform will perform the following actions:

  # aws_instance.wpserver will be created
  + resource "aws_instance" "wpserver" {
      + ami                                  = "ami-0261755bbcb8c4a84"
      + arn                                  = (known after apply)
      + associate_public_ip_address          = true
      + availability_zone                    = (known after apply)
      + cpu_core_count                       = (known after apply)
      + cpu_threads_per_core                 = (known after apply)
      + disable_api_stop                     = (known after apply)
      + disable_api_termination              = (known after apply)
      + ebs_optimized                        = (known after apply)
      + get_password_data                    = false
      + host_id                              = (known after apply)
      + host_resource_group_arn              = (known after apply)
      + iam_instance_profile                 = (known after apply)
      + id                                   = (known after apply)
      + instance_initiated_shutdown_behavior = (known after apply)
```

```
8. 18.116.44.100 (root)

      + owner_id              = (known after apply)
      + revoke_rules_on_delete = false
      + tags_all              = (known after apply)
      + vpc_id                = (known after apply)
    }

  # local_file.hosts will be created
  + resource "local_file" "hosts" {
      + content             = (known after apply)
      + content_base64sha256 = (known after apply)
      + content_base64sha512 = (known after apply)
      + content_md5         = (known after apply)
      + content_sha1        = (known after apply)
      + content_sha256      = (known after apply)
      + content_sha512      = (known after apply)
      + directory_permission = "0777"
      + file_permission     = "0777"
      + filename            = "./hosts"
      + id                  = (known after apply)
    }

  # null_resource.run_playbook will be created
  + resource "null_resource" "run_playbook" {
      + id = (known after apply)
    }

Plan: 5 to add, 0 to change, 0 to destroy.

Do you want to perform these actions?
  Terraform will perform the actions described above.
  Only 'yes' will be accepted to approve.

  Enter a value:
```

- Put “yes” to proceed creation instance and run playbook to install wordpress setup.

```
8. 18.116.44.100 (root)

  + resource "local_file" "hosts" {
      + content             = (known after apply)
      + content_base64sha256 = (known after apply)
      + content_base64sha512 = (known after apply)
      + content_md5         = (known after apply)
      + content_sha1        = (known after apply)
      + content_sha256      = (known after apply)
      + content_sha512      = (known after apply)
      + directory_permission = "0777"
      + file_permission     = "0777"
      + filename            = "./hosts"
      + id                  = (known after apply)
    }

  # null_resource.run_playbook will be created
  + resource "null_resource" "run_playbook" {
      + id = (known after apply)
    }

Plan: 5 to add, 0 to change, 0 to destroy.

Do you want to perform these actions?
  Terraform will perform the actions described above.
  Only 'yes' will be accepted to approve.

  Enter a value: yes

aws_security_group.skv-terra: Creating...
aws_key_pair.key-tfnew: Creating...
aws_key_pair.key-tfnew: Creation complete after 0s [id=key-tfnew]
aws_security_group.skv-terra: Creation complete after 2s [id=sg-050d3345cdc295d20]
aws_instance.wpserver: Creating...

support MobaXterm by subscribing to the professional edition here: https://mobaxterm.mobatek.net
```

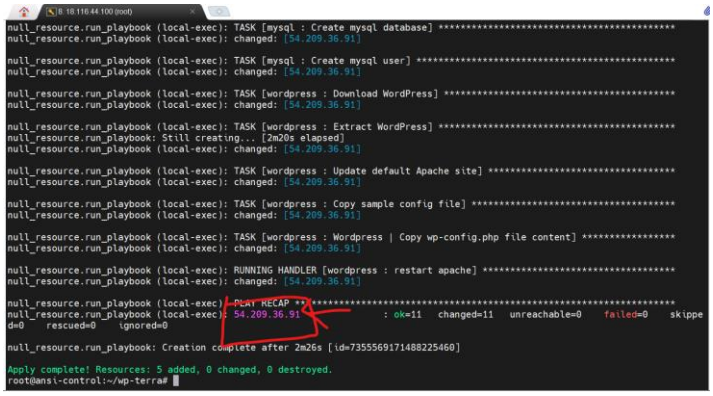
```
aws_instance.wpserver (remote-exec):   User: ubuntu
aws_instance.wpserver (remote-exec):   Password: false
aws_instance.wpserver (remote-exec):   Private key: true
aws_instance.wpserver (remote-exec):   Certificate: false
aws_instance.wpserver (remote-exec):   SSH Agent: false
aws_instance.wpserver (remote-exec):   Checking Host Key: false
aws_instance.wpserver (remote-exec):   Target Platform: unix
aws_instance.wpserver (remote-exec): Connecting to remote host via SSH...
aws_instance.wpserver (remote-exec):   Host: 54.209.36.91
aws_instance.wpserver (remote-exec):   User: ubuntu
aws_instance.wpserver (remote-exec):   Password: false
aws_instance.wpserver (remote-exec):   Private key: true
aws_instance.wpserver (remote-exec):   Certificate: false
aws_instance.wpserver (remote-exec):   SSH Agent: false
aws_instance.wpserver (remote-exec):   Checking Host Key: false
aws_instance.wpserver (remote-exec):   Target Platform: unix
aws_instance.wpserver (remote-exec): Connected!
aws_instance.wpserver: Still creating... [1m0s elapsed]
aws_instance.wpserver: Creation complete after 1m1s [id=i-03788c15fc6bdf74a]
local_file.hosts: Creating...
local_file.hosts: Creation complete after 0s [id=61ca35d6e89fe159af709247a87f6ab2c875522a]
null_resource.run_playbook: Creating...
null_resource.run_playbook: Provisioning with 'local-exec'...
null_resource.run_playbook (local-exec): Executing: ["/bin/sh" "-c" "ANSIBLE_HOST_KEY_CHECKING=False ansible-playbook -u ubunt
u -i hosts --private-key ~/.ssh/id_rsa -e 'pub_key=~/.ssh/id_rsa.pub' playbook.yml"]

null_resource.run_playbook (local-exec): PLAY [wordpress] ***********************************************************

null_resource.run_playbook (local-exec): TASK [server : Update apt cache] ******************************************
null_resource.run_playbook: Still creating... [10s elapsed]
null_resource.run_playbook (local-exec): changed: [54.209.36.91]

null_resource.run_playbook (local-exec): TASK [server : Install required software] ************************************
```

```
local_file.hosts: Creating...
local_file.hosts: Creation complete after 0s [id=61ca35d6e89fe159af709247a87f6ab2c875522a]
null_resource.run_playbook: Creating...
null_resource.run_playbook: Provisioning with 'local-exec'...
null_resource.run_playbook (local-exec): Executing: ["/bin/sh" "-c" "ANSIBLE_HOST_KEY_CHECKING=False ansible-playbook -u ubunt
u -i hosts --private-key ~/.ssh/id_rsa -e 'pub_key=~/.ssh/id_rsa.pub' playbook.yml"]

null_resource.run_playbook (local-exec): PLAY [wordpress] ***********************************************************

null_resource.run_playbook (local-exec): TASK [server : Update apt cache] ******************************************
null_resource.run_playbook: Still creating... [10s elapsed]
null_resource.run_playbook (local-exec): changed: [54.209.36.91]

null_resource.run_playbook (local-exec): TASK [server : Install required software] ************************************
null_resource.run_playbook: Still creating... [20s elapsed]
null_resource.run_playbook: Still creating... [30s elapsed]
null_resource.run_playbook: Still creating... [40s elapsed]
null_resource.run_playbook: Still creating... [50s elapsed]
null_resource.run_playbook: Still creating... [1m0s elapsed]
null_resource.run_playbook: Still creating... [1m10s elapsed]
null_resource.run_playbook: Still creating... [1m20s elapsed]
null_resource.run_playbook: Still creating... [1m30s elapsed]
null_resource.run_playbook: Still creating... [1m40s elapsed]
```

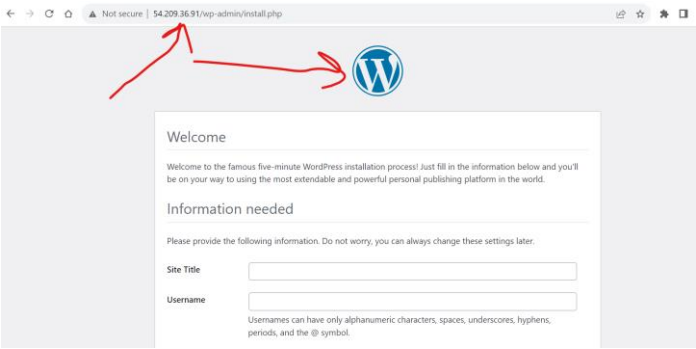
```
null_resource.run_playbook: Still creating... [1m50s elapsed]
null_resource.run_playbook: Still creating... [2m0s elapsed]
null_resource.run_playbook (local-exec): changed: [54.209.36.91]

null_resource.run_playbook (local-exec): TASK [php : Install php extensions] ****************************************
null_resource.run_playbook: Still creating... [2m10s elapsed]
null_resource.run_playbook (local-exec): changed: [54.209.36.91]

null_resource.run_playbook (local-exec): TASK [mysql : Create mysql database] ***************************************
null_resource.run_playbook (local-exec): changed: [54.209.36.91]

null_resource.run_playbook (local-exec): TASK [mysql : Create mysql user] ******************************************
null_resource.run_playbook (local-exec): changed: [54.209.36.91]

null_resource.run_playbook (local-exec): TASK [wordpress : Download WordPress] *************************************
null_resource.run_playbook (local-exec): changed: [54.209.36.91]

null_resource.run_playbook (local-exec): TASK [wordpress : Extract WordPress] ***************************************
null_resource.run_playbook: Still creating... [2m20s elapsed]
null_resource.run_playbook (local-exec): changed: [54.209.36.91]

null_resource.run_playbook (local-exec): TASK [wordpress : Update default Apache site] *******************************
null_resource.run_playbook (local-exec): changed: [54.209.36.91]

null_resource.run_playbook (local-exec): TASK [wordpress : Copy sample config file] ***********************************
null_resource.run_playbook (local-exec): changed: [54.209.36.91]

null_resource.run_playbook (local-exec): TASK [wordpress : Wordpress | Copy wp-config.php file content] *****************
null_resource.run_playbook (local-exec): changed: [54.209.36.91]
```

```
null_resource.run_playbook (local-exec): TASK [mysql : Create mysql database] ***************************************
null_resource.run_playbook (local-exec): changed: [54.209.36.91]

null_resource.run_playbook (local-exec): TASK [mysql : Create mysql user] ******************************************
null_resource.run_playbook (local-exec): changed: [54.209.36.91]

null_resource.run_playbook (local-exec): TASK [wordpress : Download WordPress] *************************************
null_resource.run_playbook (local-exec): changed: [54.209.36.91]

null_resource.run_playbook (local-exec): TASK [wordpress : Extract WordPress] ***************************************
null_resource.run_playbook: Still creating... [2m20s elapsed]
null_resource.run_playbook (local-exec): changed: [54.209.36.91]

null_resource.run_playbook (local-exec): TASK [wordpress : Update default Apache site] *******************************
null_resource.run_playbook (local-exec): changed: [54.209.36.91]

null_resource.run_playbook (local-exec): TASK [wordpress : Copy sample config file] ***********************************
null_resource.run_playbook (local-exec): changed: [54.209.36.91]

null_resource.run_playbook (local-exec): TASK [wordpress : Wordpress | Copy wp-config.php file content] *****************
null_resource.run_playbook (local-exec): changed: [54.209.36.91]

null_resource.run_playbook (local-exec): RUNNING HANDLER [wordpress : restart apache] *******************************
null_resource.run_playbook (local-exec): changed: [54.209.36.91]

null_resource.run_playbook (local-exec): PLAY RECAP *****************************************************************
null_resource.run_playbook (local-exec): 54.209.36.91              : ok=11   changed=11   unreachable=0    failed=0    skippe
d=0    rescued=0    ignored=0

null_resource.run_playbook: Creation complete after 2m26s [id=7355569171488225460]

Apply complete! Resources: 5 added, 0 changed, 0 destroyed.
root@ansi-control:~/wp-terra#
```

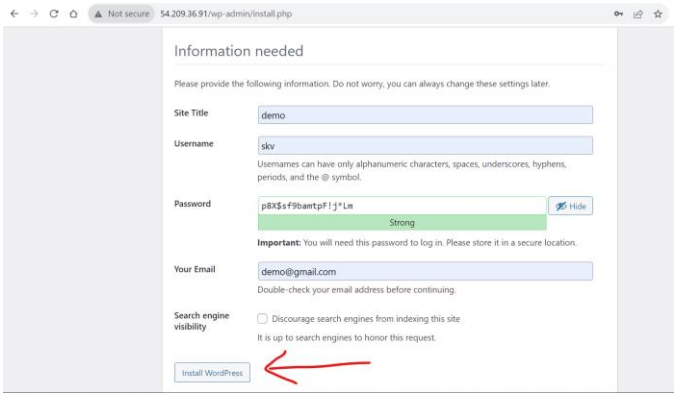
- **Now terraform has created ec2 instance with below public ip**
  - **Public ip**



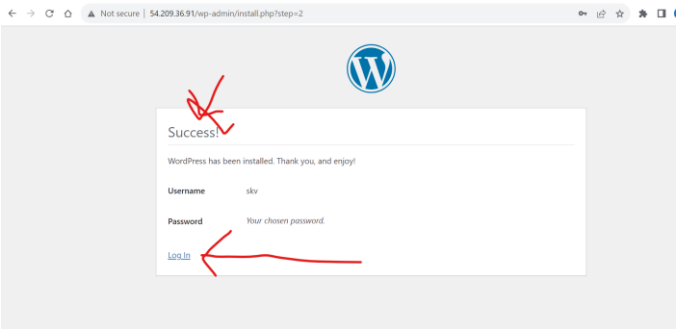
- **Now take this public ip and put in web browser and see the magic**
- **Wow!!!!... wordpress is installed on newly created instance as below:**



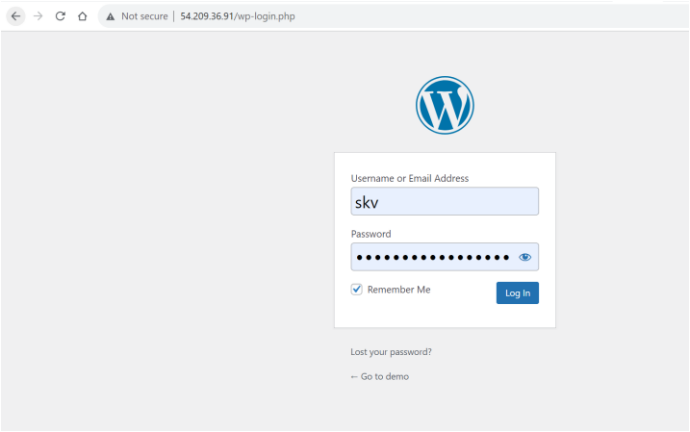
- **Now setup the wordpress with enter few required details. And then click install wordpress.**



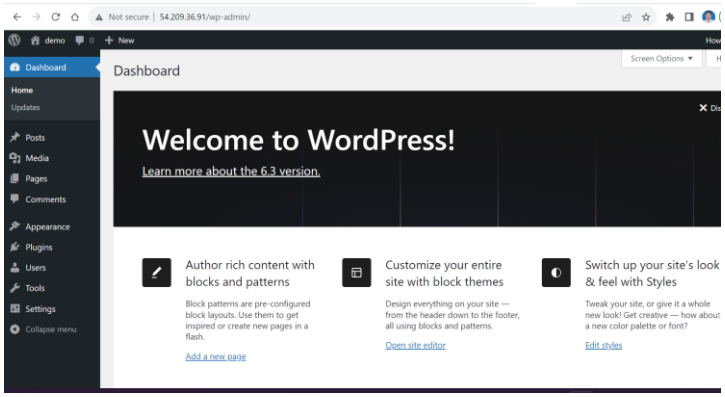
- **To check if data base is accessible the put above details and look below snip.**



- **Same Followed below:**



- **Now I have setup wordpress and it looks like below:**



Documents for project at github:

- Documents for wordpress:
  My github project link "https://github.com/vermasunilk792/cmat-proj2-sl.git"