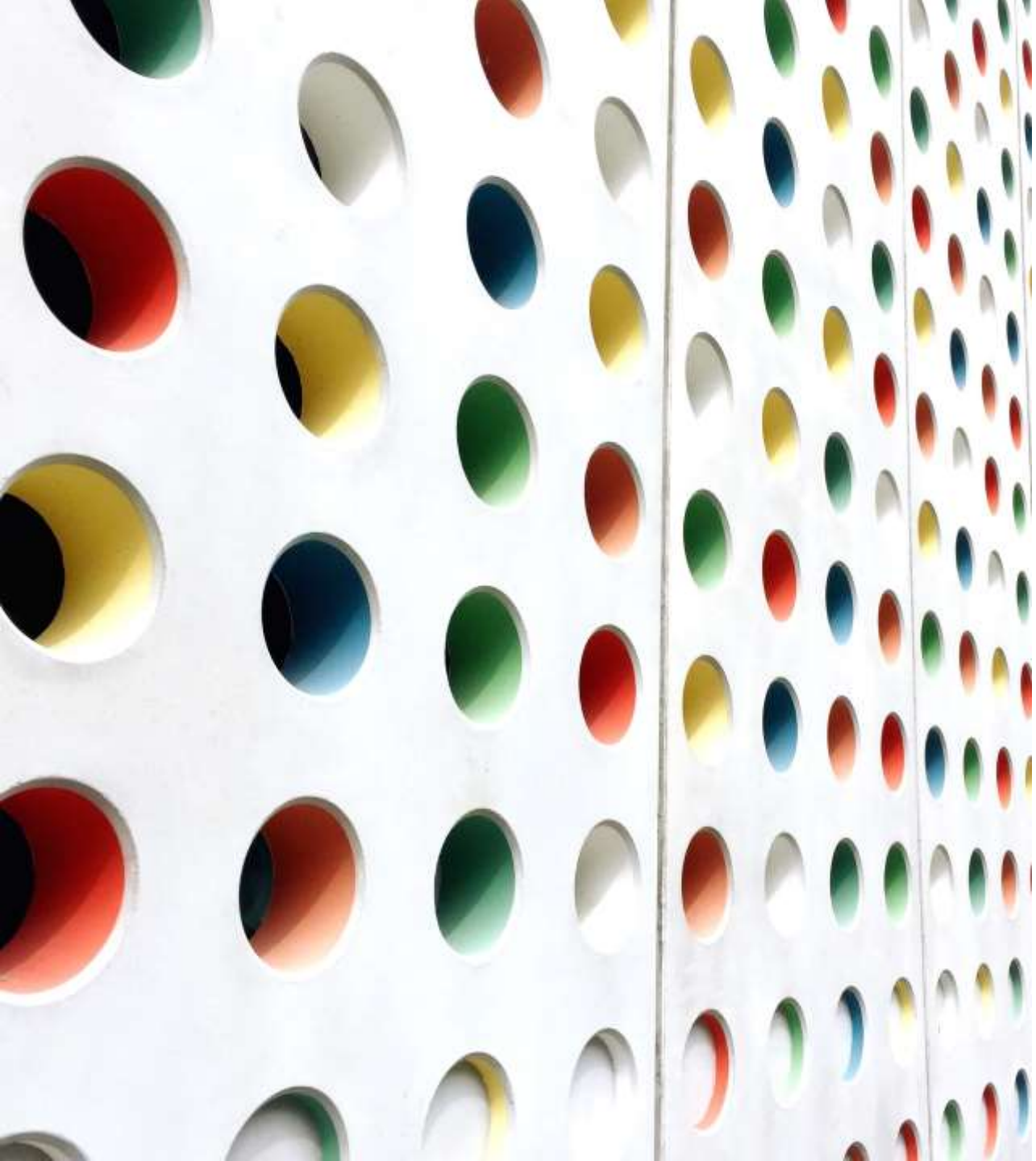


PPT-1
DEPLOY THREE-TIER ARCHITURE IN AWS USING TERRAFORM

RANJITH VENUGOPAL

What is Terraform?

TERRAFORM IS AN OPEN-SOURCE INFRASTRUCTURE-AS-CODE SOFTWARE TOOL CREATED BY HASHCORP. IT IS AN INFRASTRUCTURE AS CODE TOOL THAT LETS YOU BUILD, CHANGE, AND VERSION CLOUD AND ON-PREM RESOURCES SAFELY AND EFFICIENTLY.

INFRASTRUCTURE AS CODE

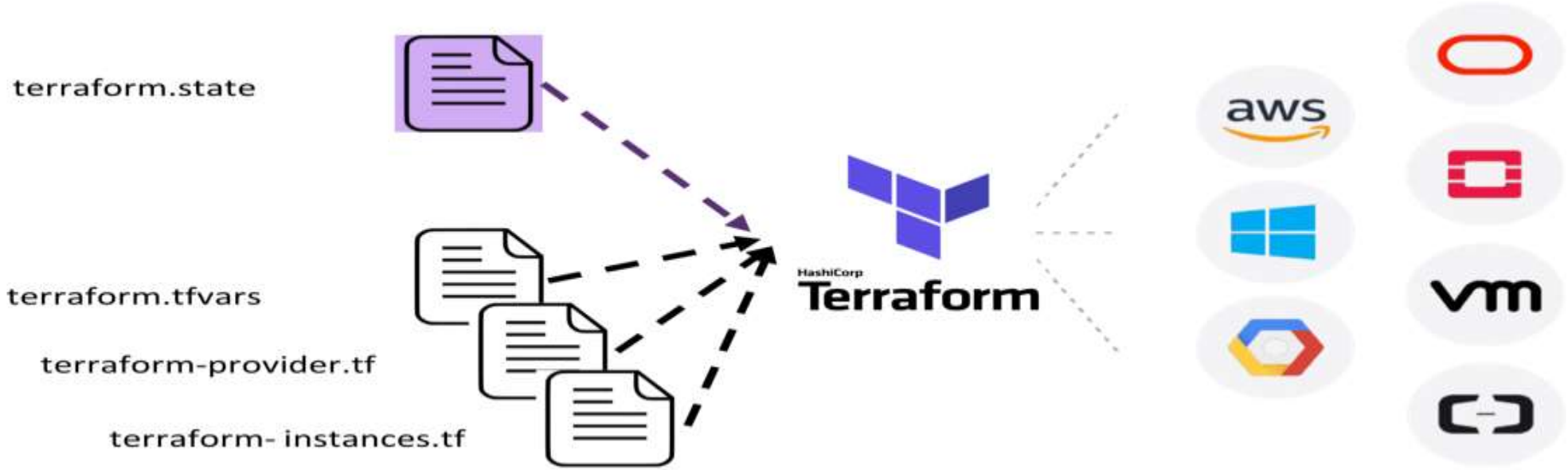
Terraform is an Infrastructure as Code (IaC) tool that allows engineers to define their software infrastructure in code. While the idea of “code” may not be novel to engineers; the ability to provision infrastructure this way is a powerful abstraction that enables managing large distributed systems at scale.

SOME BASIC COMMANDS IN TERRAFORM :-

- terraform init :- To initialize Terraform.
- terraform plan :- Make corrections to the configuration as necessary.
- terraform apply :- Wait until Terraform displays the "Apply complete!" message.
- terraform validate :- To check whether your configuration is valid, enter the following command.
- terraform fmt :- To reformat your Terraform configuration in the standard style, enter the following command.
- terraform destroy :- Remove resources previously applied with your Terraform configuration by running the following command and entering yes at the prompt

Prerequisites :-

1. Go to Aws console login to root user.
2. Create an IAM user account and log into IAM user login.
3. Create an EC2 Instance and connect to the terminal.
4. Create a provider file inside the directory (vi provider.tf)
5. Create access key and secret key in provider file or using configure command.



LIST OF FILES TO CREATE INFRASTRUCTURE :-

1. Create a file for Vpc (vi vpc.tf)
2. Create a file for Subnet (vi subnet.tf) [Create both public and private subnets]
3. Create a file for Internet gate way (vi igw.tf)
4. Create a file for Route table (vi route_table_public.tf) [Route table association and also create and attach public subnet.]
5. Create a file for EC2 instances (vi ec2.tf)
6. Create a file for Security group for the front end tier (vi web_sg.tf)
7. Create a file for Security group for the Database tier (vi database_sg.tf)
8. Create a file for Application load balancer (vi alb.tf)
9. Create a file for Rds instance (vi rds.tf)
10. Create a file for Outputs (vi outputs.tf)
11. Create a file for User data (vi data.sh)

OUTPUT :-

Hello World from ip-10-0-1-188.ec2.internal

CONCLUSION :-

In conclusion, Terraform and AWS are powerful tools for streamlining your infrastructure. With Terraform, you can manage your infrastructure as code, and with AWS, you can easily provision and manage your infrastructure as code, and with AWS, you can easily provision and manage your infrastructure. Together, Terraform and AWS provide a flexible and scalable solution for building and managing your applications

- Ranjith Venugopal