



Terraform

What is Infrastructure as a code?

Understanding the Basics

There are two ways in which you can create and manage your infrastructure:

- Manually approach.
- Through Automation

Eg: we have to create a 10 vm in Azure.

Manual Process Implementation

1. Need to login in Portal
2. Create a vm (same process for 10 vm)

Automation Process

1. We are going to write a code that can be called multiple times as needed.

What is Infrastructure as Code (IaC)?



Definition: Infrastructure as Code (IaC) is the managing and provisioning of infrastructure through code instead of through manual processes

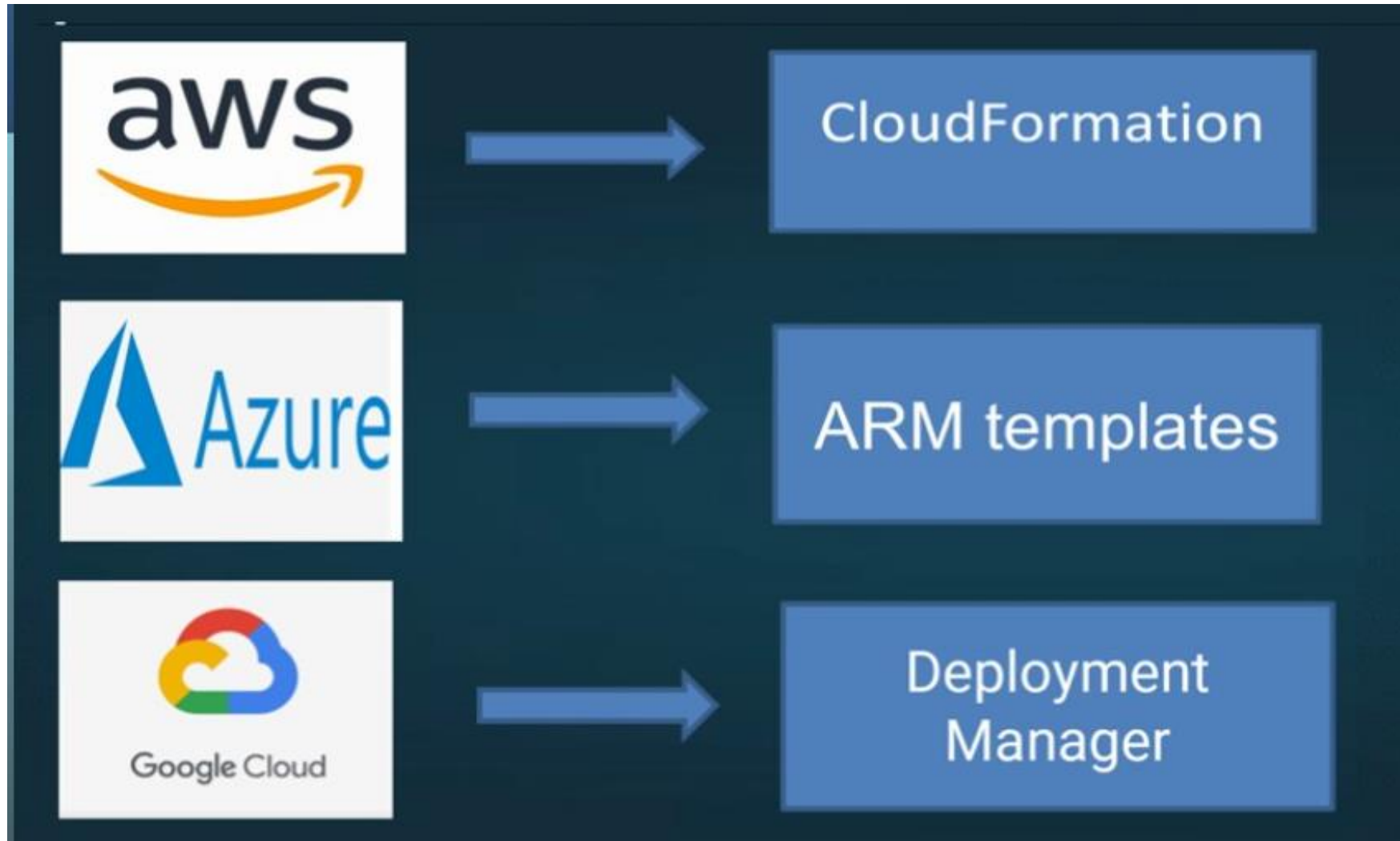
.



Benefits:

- Speed of Infrastructure Management.
- Low Risk of Human Errors.
- Version Control.
- Easy collaboration between Teams.

IAC Tools Provided By Cloud



Why should you choose Terraform?



What is Terraform?

Terraform is an open-source infrastructure as code tool developed by HashiCorp. It is used to define and provision the complete infrastructure using an easy-to-learn declarative language.

Terraform official documentation link: **<https://developer.hashicorp.com/terraform/docs>**

It is an infrastructure provisioning tool where you can store your cloud infrastructure setup as codes.

Prerequisites to learn Terraform:

- Basic understanding of Cloud Platform
- No Programming skill is required.

Benefits of Terraform



SUPPORTS MULTIPLE
PLATFORMS, HAS HUNDREDS
OF PROVIDERS.



SIMPLE CONFIGURATION
LANGUAGE AND FASTER
LEARNING CURVE.



EASY INTEGRATION WITH
CONFIGURATION
MANAGEMENT TOOLS LIKE
ANSIBLE.



EASILY EXTENSIBLE WITH THE
HELP OF PLUGINS.



FREE !!!

Join us in our Adventure



<https://www.linkedin.com/in/akash-kumar-480b3858/>



https://www.instagram.com/akash_sinha08/

Categories of Infrastructure As Code

Categories of Infrastructure As Code



Terraform, CloudFormation, Bicep, Deployment manager are the **infrastructure orchestration tools** which basically means they can provision the servers and infrastructure by themselves.



Ansible, Chef and Puppet are **configuration management tools** which means they are primarily designed to install and manage software on existing servers.

Ansible vs Terraform

Ansible

Configuration Management Tool

Use Case: Deploy and Configure the applications that run on top of the infrastructure.

Ansible is well suited for configuring servers with the right software and updates on already provisioned cloud.

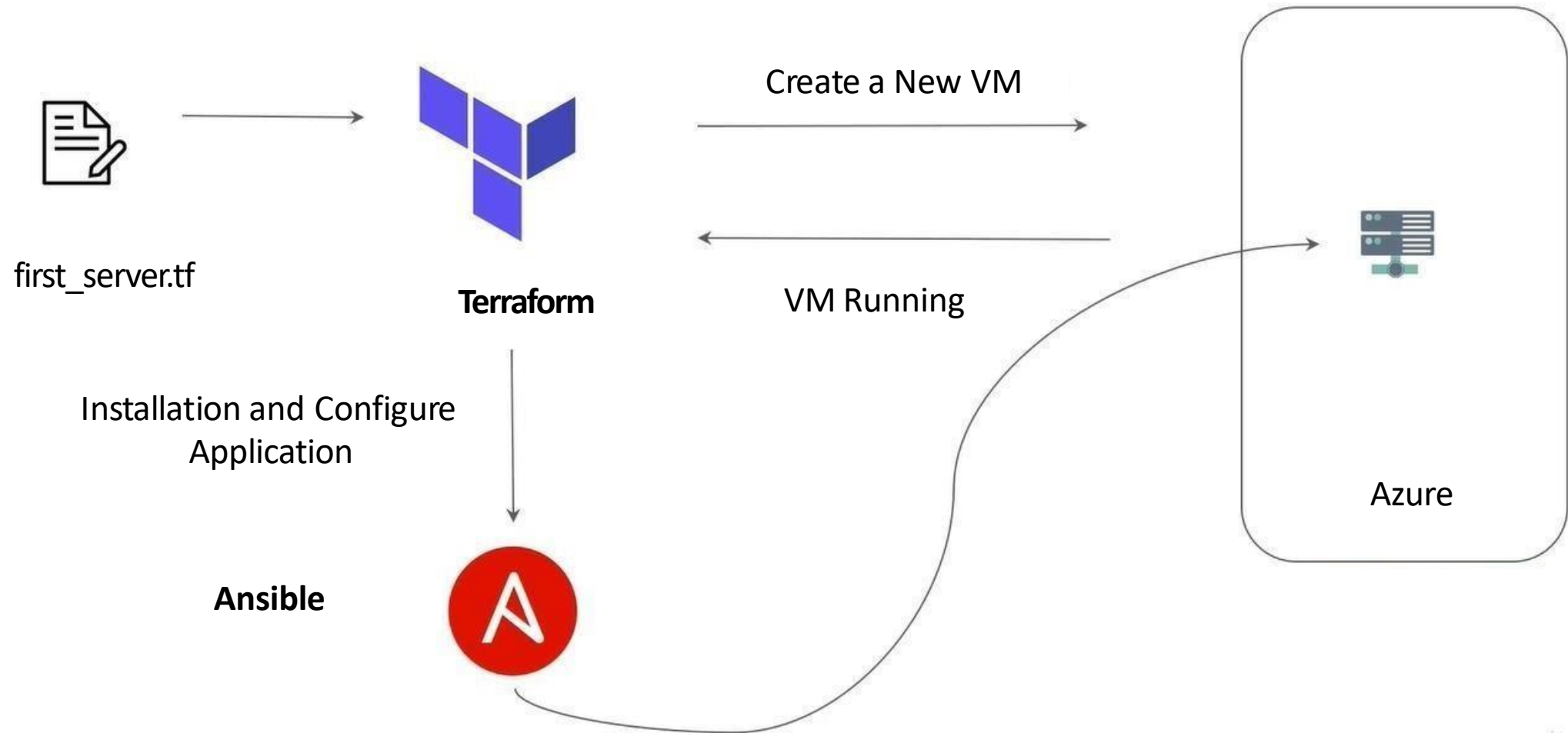
Terraform

Orchestration Tool

Use Case: Terraform is focused on infrastructure provisioning.

Terraform is well suited for orchestration cloud services and setup cloud infrastructure from scratch.

IAC & Configuration Management = Friends



Join us in our Adventure



<https://www.linkedin.com/in/akash-kumar-480b3858/>



https://www.instagram.com/akash_sinha08/

Setting up Environment

Tools Required



Visual studio Installation ---> Write our code



Terraform cli Installation ---> Terraform Cli for our Terraform Commands



Azure CLI Installation ---> Authentication to the Azure Portal

Join us in our Adventure



<https://www.linkedin.com/in/akash-kumar-480b3858/>



https://www.instagram.com/akash_sinha08/