**Terraform and AWS**

**Day: 01**

**What are the Challenges with Traditional IT?**

* Slow deployment
* Expensive
* Manual Process
* Human Errors

**Types of Infrastructure as Code:**

* Terraform
* Cloud Formalion
* Heat
* Ansible
* Salt Stack
* Chef, Puppet and others

**Configuration Management Tools:**

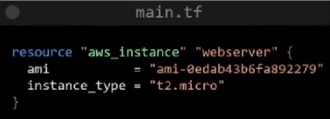
* Ansible, Chef, Puppet are configuration management tools."
* They are primarily designed to install and manage software on existing servers.
* They maintain a standard structure of code.
* Supports version control.
* Idempotent: Meaning you can run the code multiple times and every time you run, I will only make changes that are necessary to bring the environment into a defined state. It will leave as it is if it is already in place.
* Configuration Management tools can do some degree of infrastructure provisioning, but the focus here is that some tools are going to be better fit for certain types of"
* tasks.

**Infrastructure Provisioning Tools:**

* Terraform, CloudFormation are the Infrastructure Provisioning Tools which basically means they can provision the servers and infrastructure by themselves."
* Can deploy VPC, Servers Databases etc.

**But. Why Terraform?**

* As discussed, iiis a popular laC (Infrastructure Provisioning Tools).
* IIis a free and open-source tool developed by Hashicorp.
* Simple configuration language and faster learning curve.



* We can deploy Infrastructure in multiple platforms (not only AWS).
* It is possible because terraform something called as Providers which we will see in future.
* Easy integration with configuration management tools like Ansible.