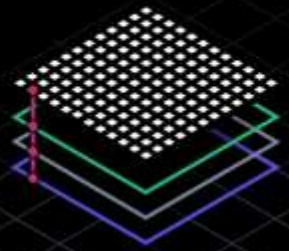
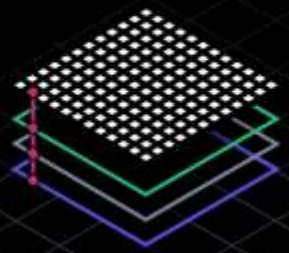
Seshagiri Sriram

Infrastructure as code (IAC)



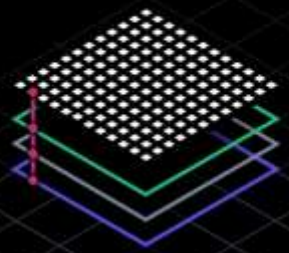
Who is this for?



Who would be interested?

- SysAdmins
- Operations Engineers
- Release Engineers
- Site Reliability Engineers
- DevOps Engineers
- Infrastructure Developers
- Full Stack Developers
- Engineering Managers
- IT Professors
- IT Students

What is infrastructure as code (IaC)?



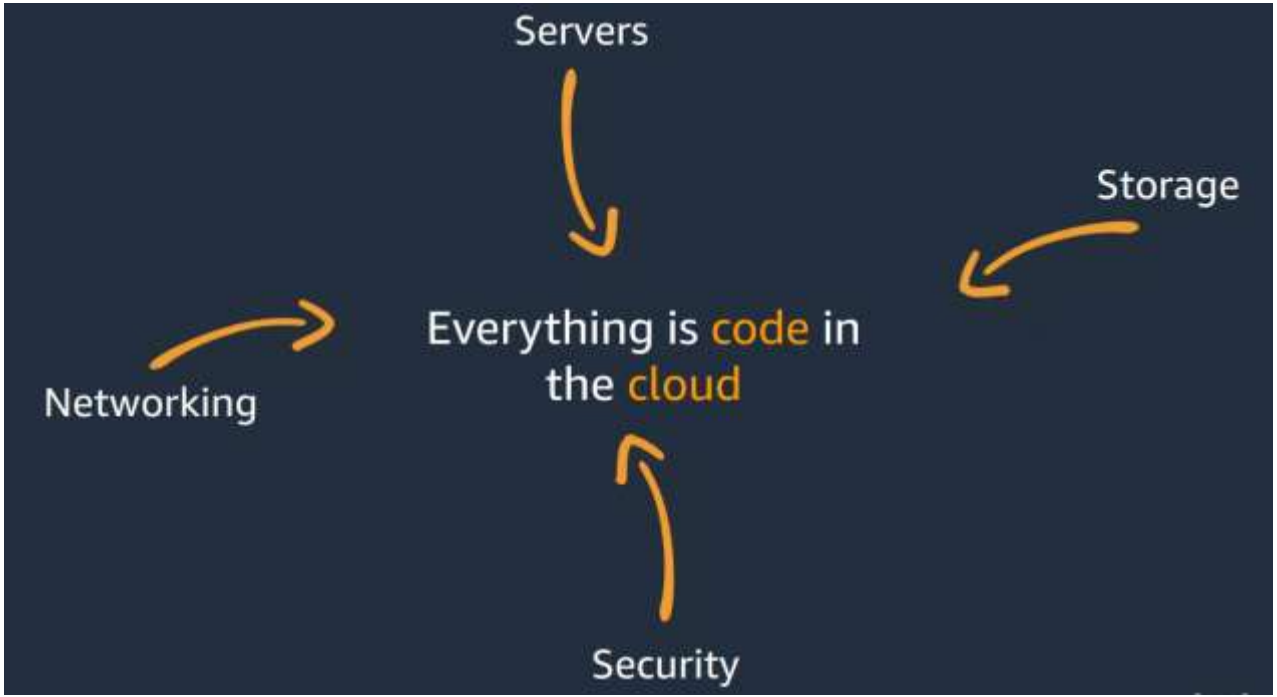
What is infrastructure as code (IaC)?

The idea behind infrastructure as code (IaC) is that you write and execute code to:

- Define
- Deploy
- Update

your infrastructure

What is infrastructure as code (IaC)?



Why IAC?

Background

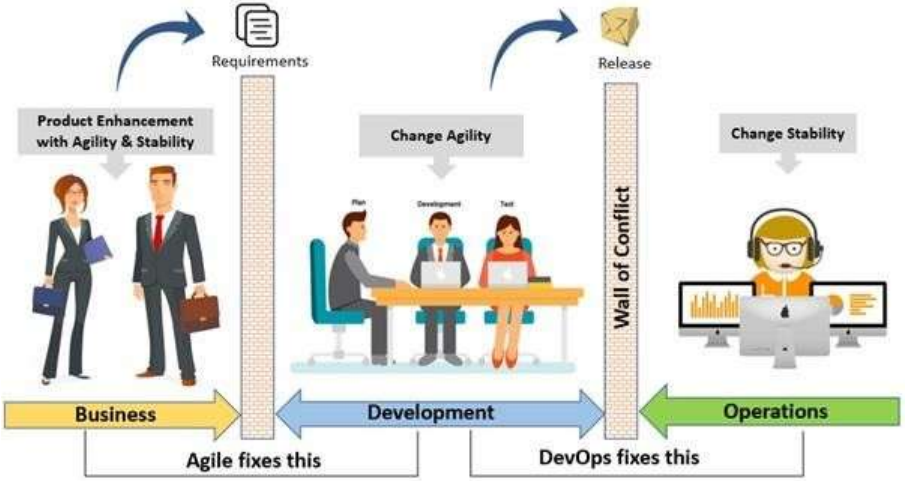
Business

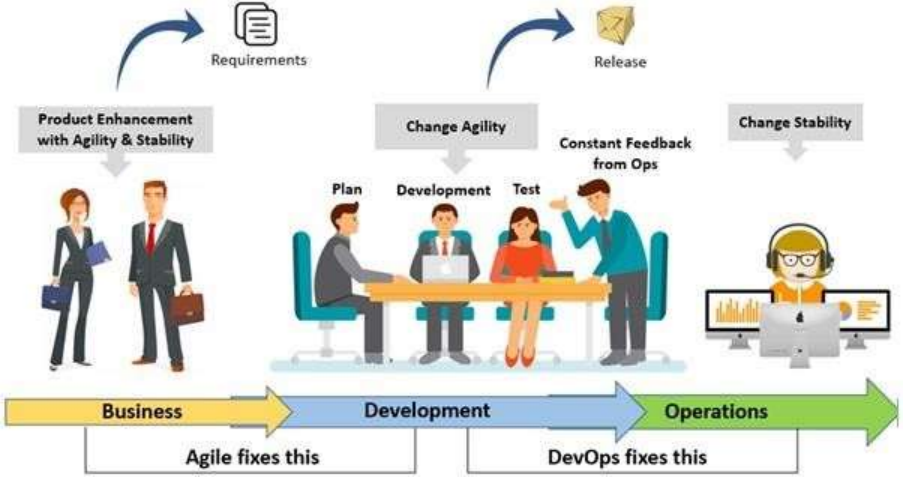
1. Ask Network team for Resources
2. Ask Server team for Resources
3. Ask Storage team for Resources
4. Procure required hardware
5. Finally build/deploy App and test

Development teams

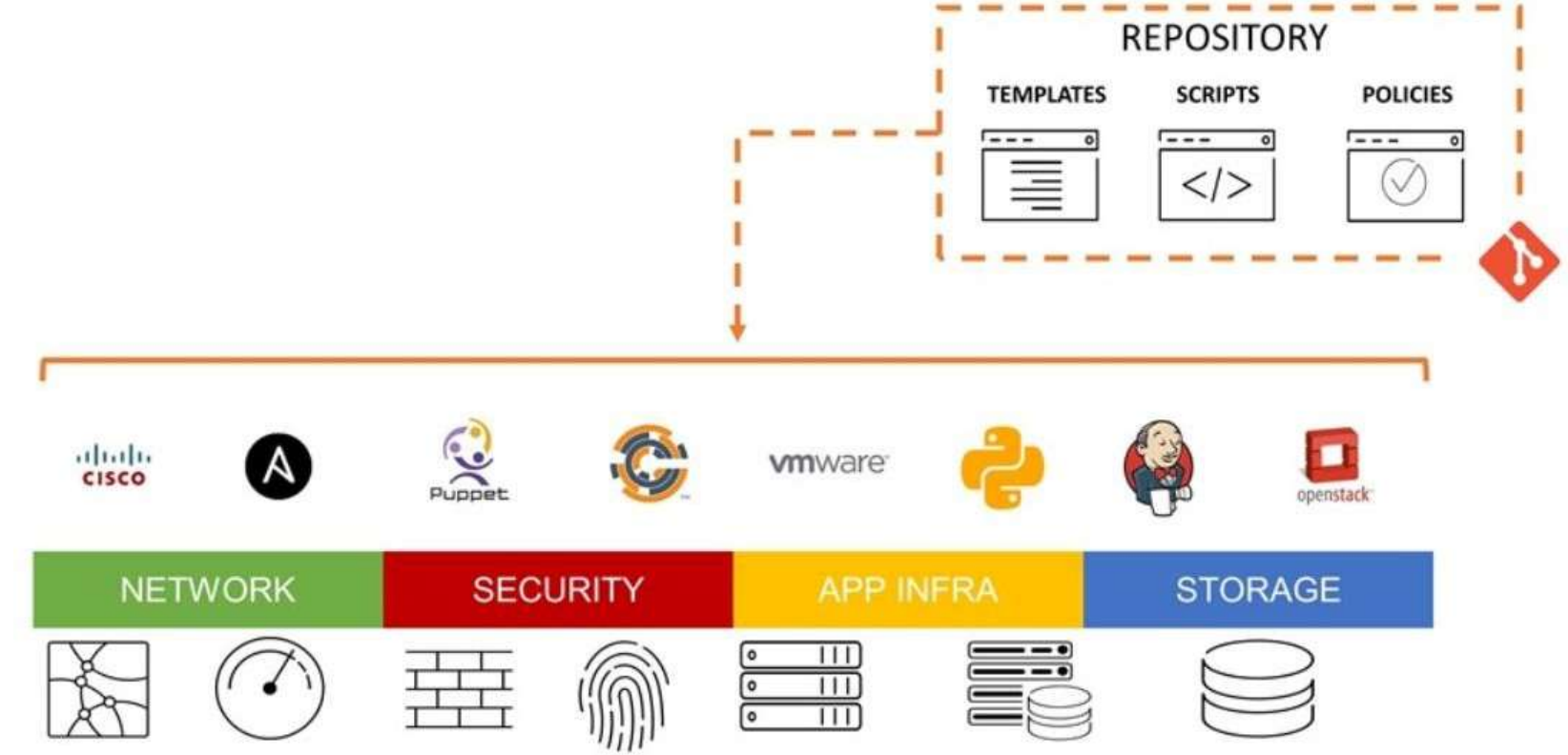
Operations/ IT

Why IAC?


Requirements
Release
Product Enhancement with Agility & Stability
Change Agility
Change Stability
Wall of Conflict
Business
Development
Operations
Agile fixes this
DevOps fixes this


Requirements
Release
Product Enhancement with Agility & Stability
Change Agility
Constant Feedback from Ops
Change Stability
Plan
Development
Test
Business
Development
Operations
Agile fixes this
DevOps fixes this

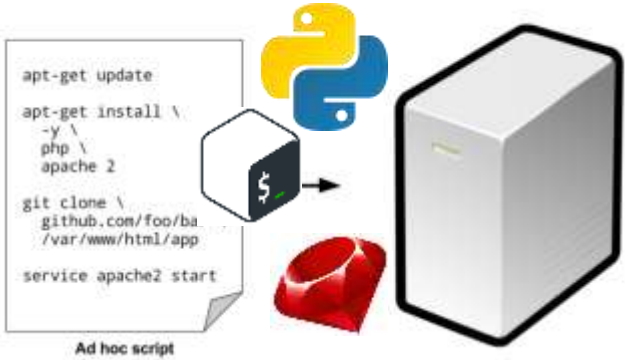
What is infrastructure as code (IaC)?



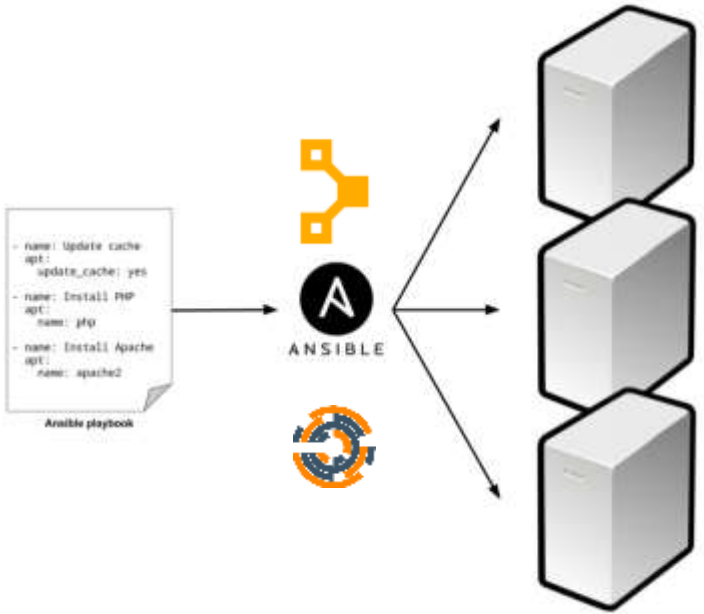
IAC Core Principles

- System are easily reproduced
- System are disposable
- Systems are consistent
- Processes are repeatable
- Design is always changing - dynamic

Categories of IaC Tools

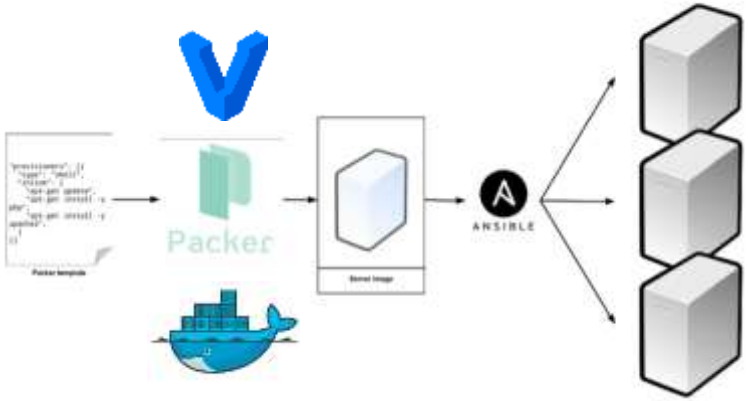


Ad Hoc Scripts

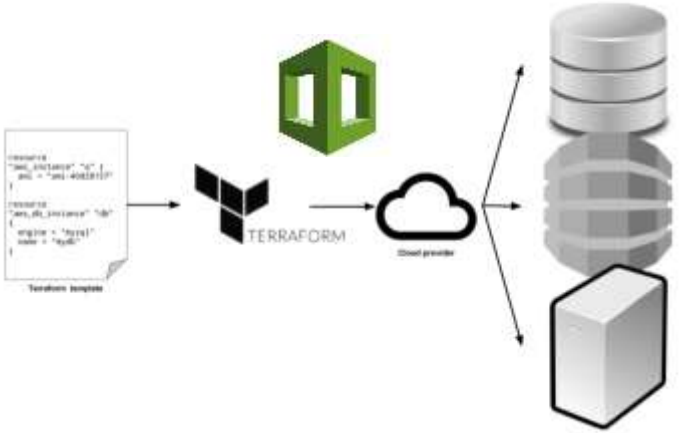


Configuration Management Tools

Categories of IaC Tools

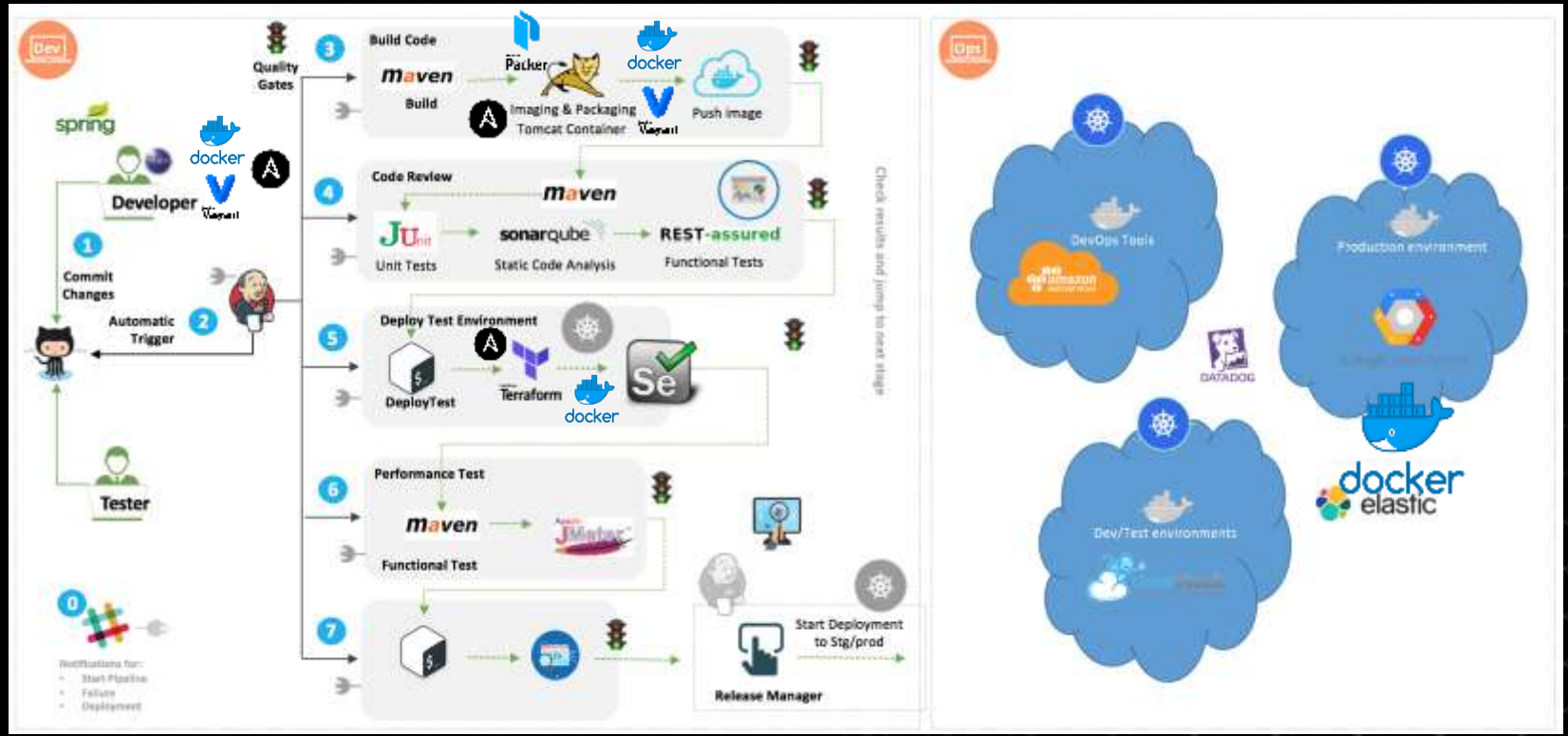


Server Templating Tools



Provisioning Tools

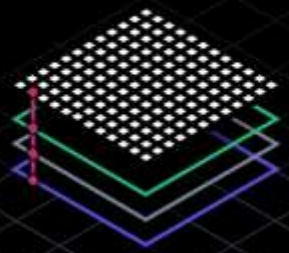
CI/CD Pipeline



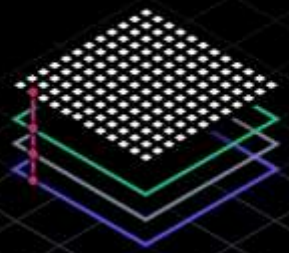
Benefits of infrastructure as code (IaC)

- Self-service
- Speed and safety
- Documentation
- Version control
- Validation
- Reuse
- Happiness

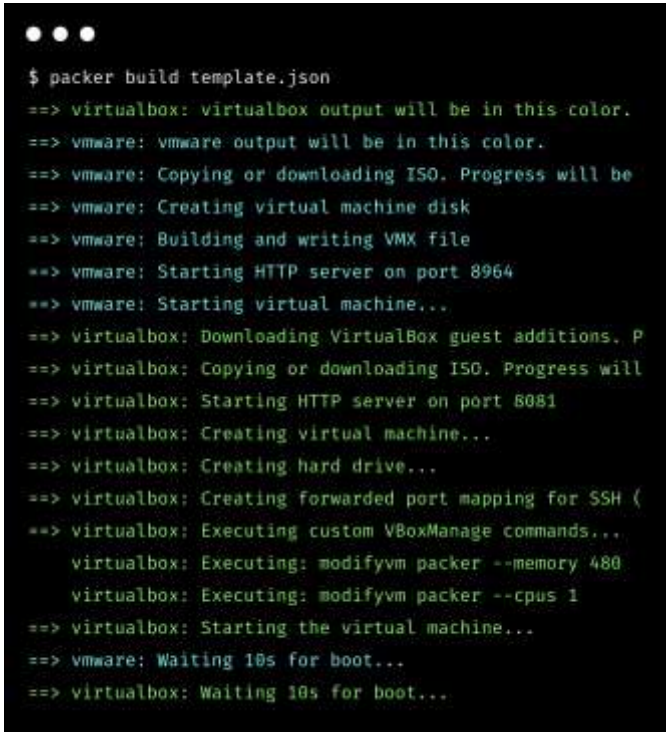
Tools and Use Cases



Packer



HashiCorp Packer

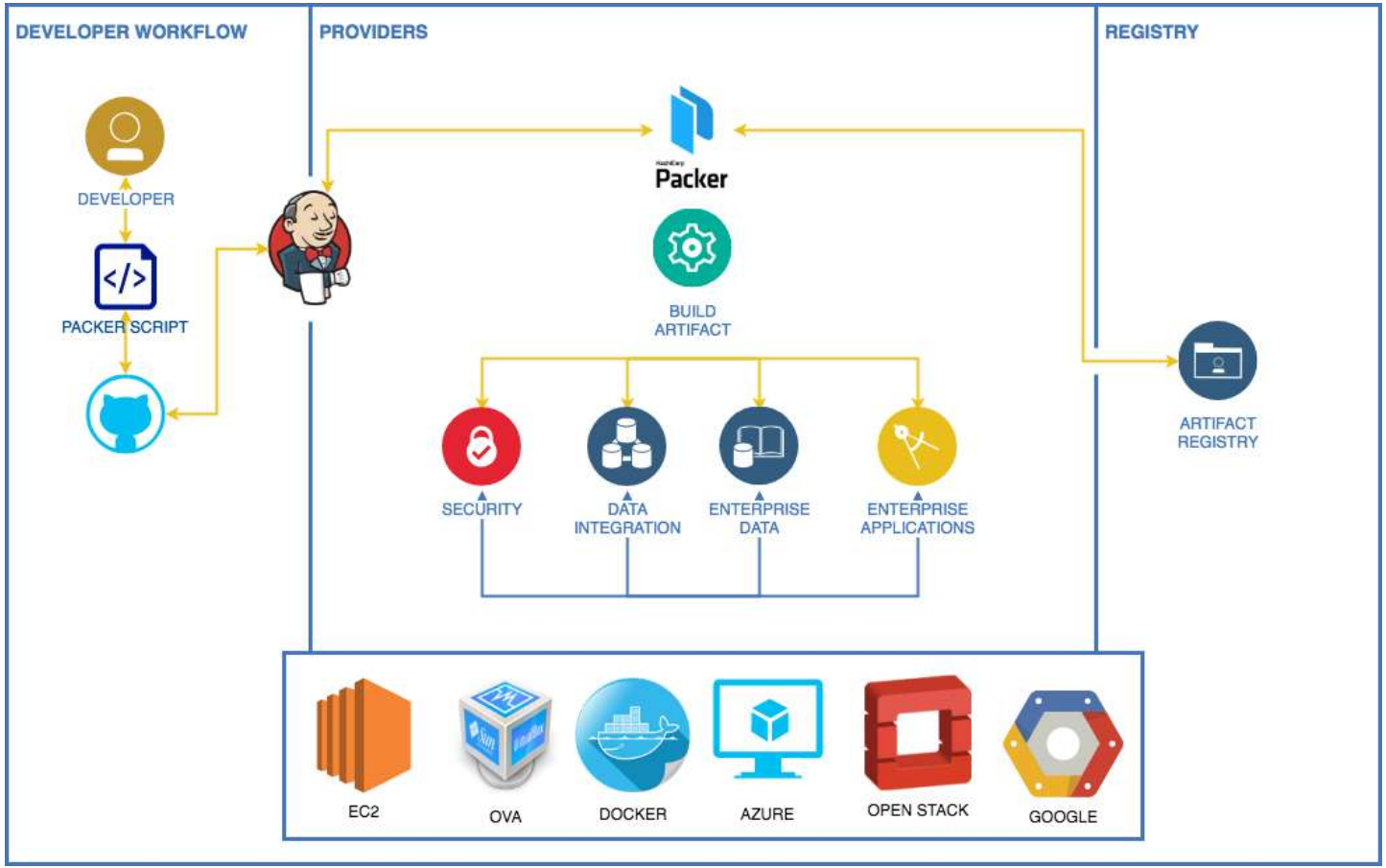


Is easy to use and automates the creation of any type of machine image.

It embraces modern configuration management by encouraging you to use automated scripts to install and configure the software

Packer brings machine images into the modern age, unlocking untapped potential and opening new opportunities.

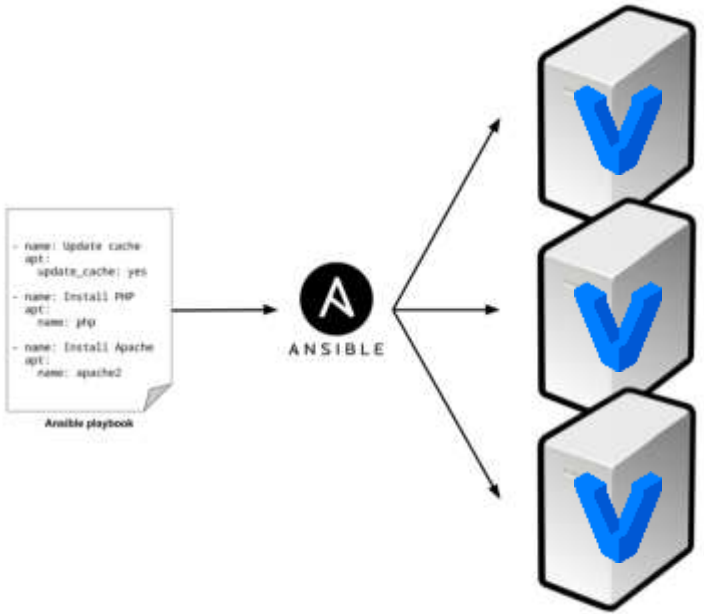

Terminal
"builders": [
{
"type": "amazon-ebs",
...
}
],
"provisioners": [
{
"type": "puppet-masterless",
...
}
],
"post-processors": [
{
"type": "atlas",
"artifact": "hashicorp/logstream",
"artifact_type": "amazon.ami"
}
]


DEVELOPER WORKFLOW
PROVIDERS
REGISTRY
DEVELOPER
PACKER SCRIPT
Packer
BUILD ARTIFACT
ARTIFACT REGISTRY
SECURITY
DATA INTEGRATION
ENTERPRISE DATA
ENTERPRISE APPLICATIONS
EC2
OVA
DOCKER
AZURE
OPEN STACK
GOOGLE

Ansible and Vagrant

Vagrant: Is a tool for building and managing virtual machines environments in a single workflow.

Ansible: Is a tool that automates provisioning, configuration management and application deployment. Works over SSH, Power Shell or remote APIs.

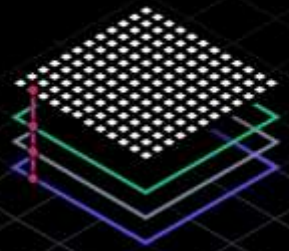


Infrastructure paradigms:

Mutable vs Immutable

Infrastructure paradigms:

Mutable vs Immutable



Mutable Infrastructure

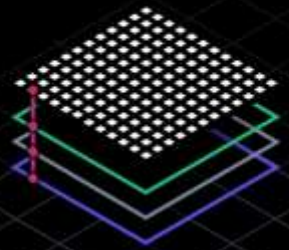
mutable infrastructure Servers are continually updated and modified in place

Immutable Infrastructure

Immutable infrastructure Servers are never modified after they're deployed

Load balancer with docker

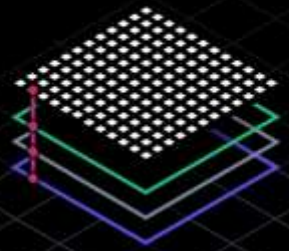
Source code: https://gitlab.com/slackmart/webserver-example



A Quick Comparison

	Source	Cloud	Type	Infrastructure	Language	Agent	Master	Community	Maturity
Chef	Open	All	Config Mgmt	Mutable	Procedural	Yes	Yes	Large	High
Puppet	Open	All	Config Mgmt	Mutable	Declarative	Yes	Yes	Large	High
Ansible	Open	All	Config Mgmt	Mutable	Procedural	No	No	Huge	Medium
SaltStack	Open	All	Config Mgmt	Mutable	Declarative	Yes	Yes	Large	Medium
CloudFormation	Closed	AWS	Provisioning	Immutable	Declarative	No	No	Small	Medium
Heat	Open	All	Provisioning	Immutable	Declarative	No	No	Small	Low
Terraform	Open	All	Provisioning	Immutable	Declarative	No	No	Huge	Low

Terraform



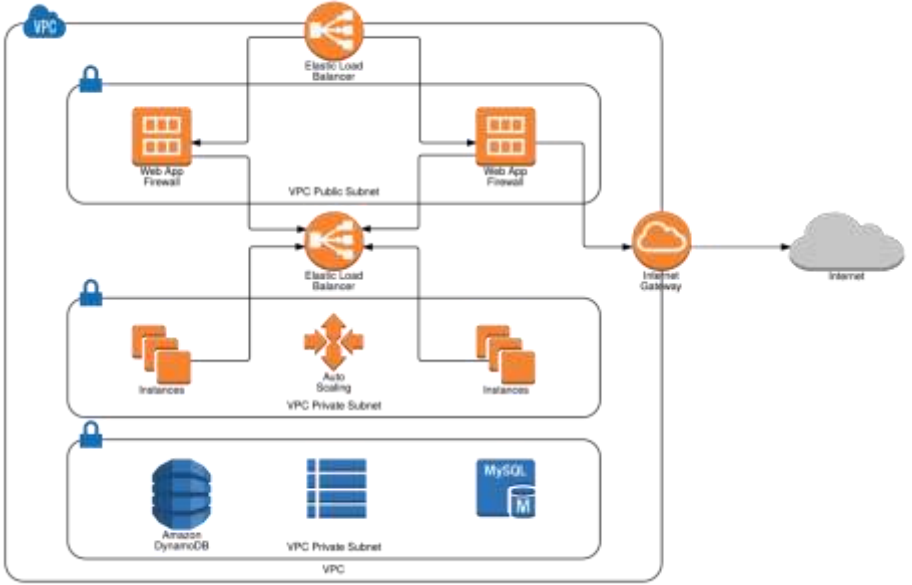
What is Terraform

Terraform is an open source infrastructure as code tool. Terraform was developed by HashiCorp company who is based in San Francisco, CA. We can also say that Terraform is a tool for building, changing, and versioning infrastructure safely and efficiently.

Provisioning challenges

- Day 1: Where we haven't actually started running anything. How do we go from running nothing to run something.

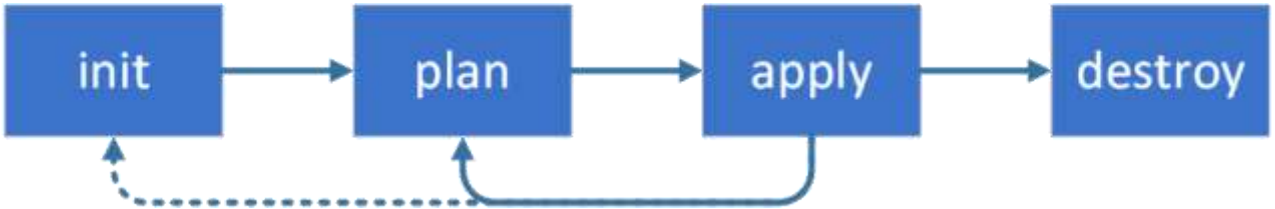
- Day 2: When we already have an infrastructure and we are trying to evolve it, changing it over time or adding and removing new services.



Terraform principles

Terraform has these three principal simple steps:

- Init
- Plan
- Apply



Terraform sample code

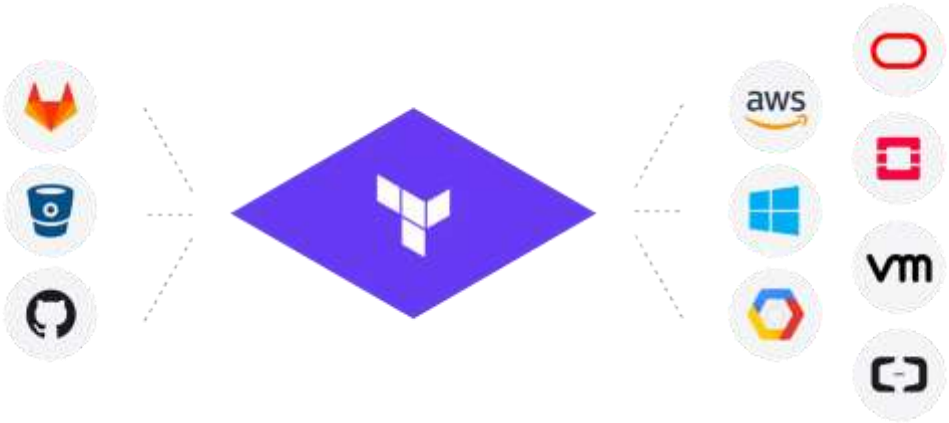
```
1 references
resource "outscale_keypair" "a_key_pair" {
  key_name   = "terraform-key-pair-name"
}

1 references
resource "outscale_firewall_rules_set" "web" {
  group_name = "terraform_acceptance_test_example"
  group_description = "Used in the terraform presentation"
}

0 references
resource "outscale_vm" "basic" {
  image_id = "ami-8a6a0120"
  instance_type = "t2.micro"
  security_group = ["${outscale_firewall_rules_set.web.id}"]
  key_name = "${outscale_keypair.a_key_pair.key_name}"
}
```

Multi-cloud

Terraform can be used to manage multi-cloud It can combine multiple provides in a single workflow which is a very nice feature.
Terraform provides one consistent workflow for developers and operators to provision resources on any infrastructure provider. One workflow to learn increases user productivity, and also reduces organizational risk as that becomes one workflow to secure, one workflow to audit, and one workflow to govern.

Terraform providers

- Right now, there are more than 100 providers and those individually can manage over a thousand resources. Providers are responsible to provide API interaction. You can find the list of providers on the terraform website.



Thank You