/thoughtworks

Amit Dube

Infrastructure As Code (IaC)

About me

Lead Infrastructure Consultant @ Thoughtworks Romania

Around 13 years of experience in IT in India, Germany & Romania

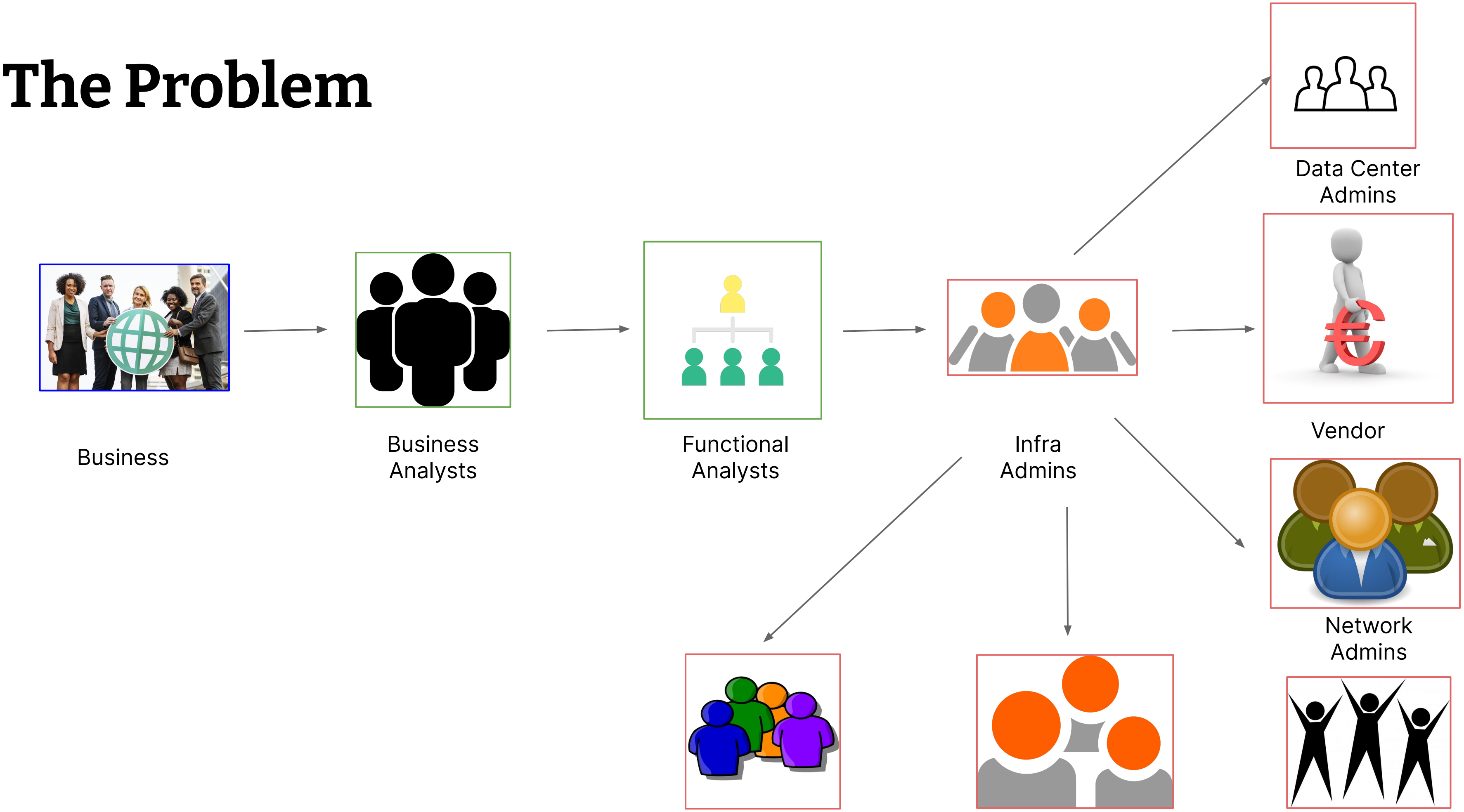
Certified Terraform Associate & writing testable infrastructure for more than 3 years

Around 3 years of experience on Google Cloud Platform (GCP)

linkedin.com/in/amit-dube-अमित-दुबे-40b6351a

https://medium.com/@amitkumardube

The Problem



Business

Business Analysts

Functional Analysts

Infra Admins

Data Center Admins

Vendor

Network Admins

Policy Admins

Security Admins

Configuration Admins

The Problem

1 Cost

Hiring an infra admin, network admin & security admin is costly.

2 Time

Provisioning new infra can take from weeks to months.

3 Issues at different levels

Many parties involved results in issues at different levels which are difficult to debug.

4 Infrastructure Consistency

Maintaining the infra consistency was a major challenge. Replicate the same infrastructure was almost impossible.

5 Configuration Drift

Configuration on servers might differ between environments.

6 Documentation

Every team required to produce some kind of documentation to later refer to provisioning and configuration.

What is infrastructure as code

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

– RedHat

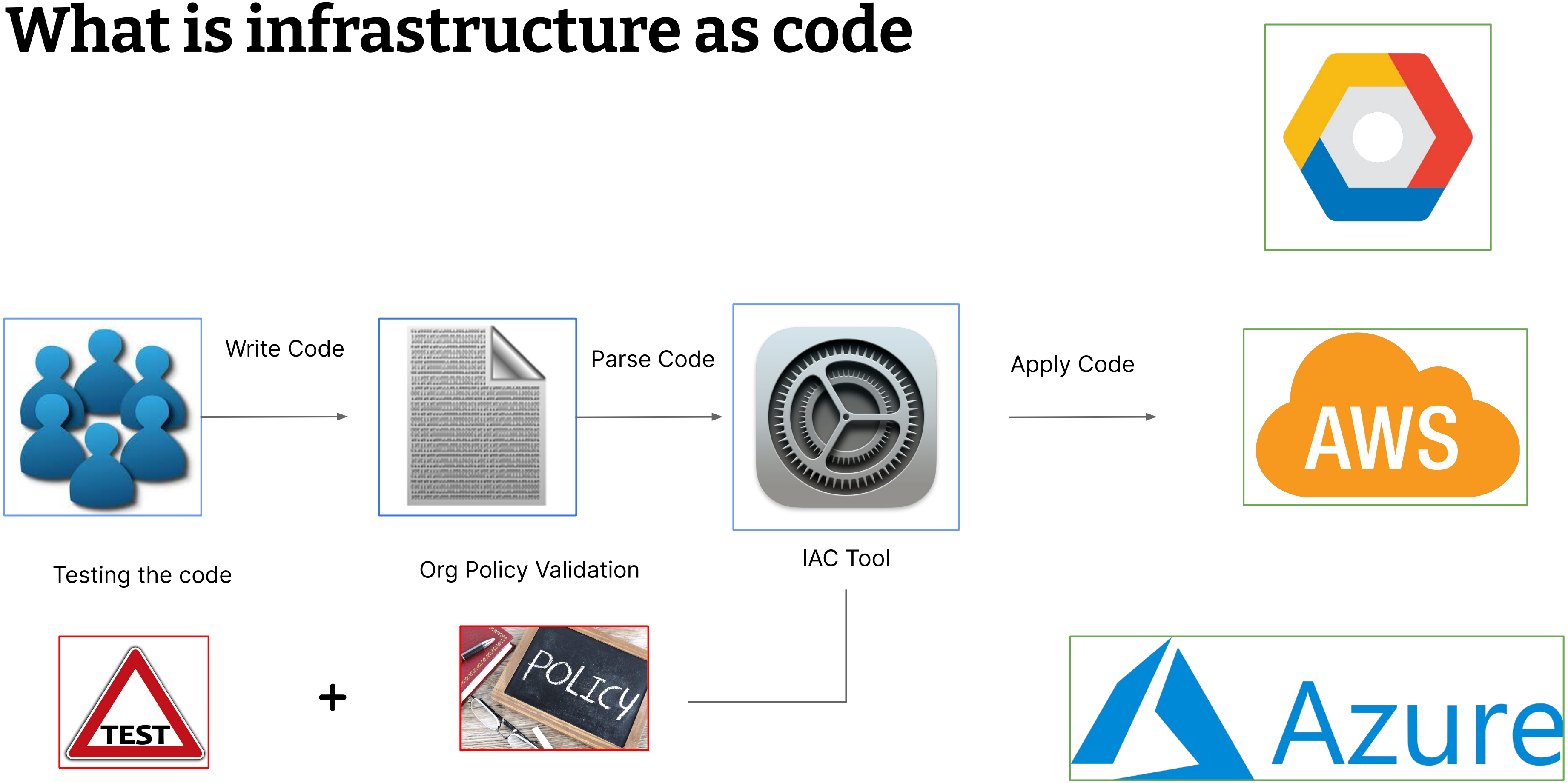
Infrastructure as code is the process of managing and provisioning computer data centers through machine-readable definition files, rather than physical hardware configuration or interactive configuration tools.

– Wikipedia

Infrastructure as Code is an approach to infrastructure automation based on practices from software development. It emphasises consistent, repeatable routines for provisioning and changing systems and their configuration. You make changes to code, then use automation to test and apply those changes to your systems.

– Kief Morris

What is infrastructure as code



Write Code

Parse Code

Apply Code

Testing the code

Org Policy Validation

IAC Tool

+

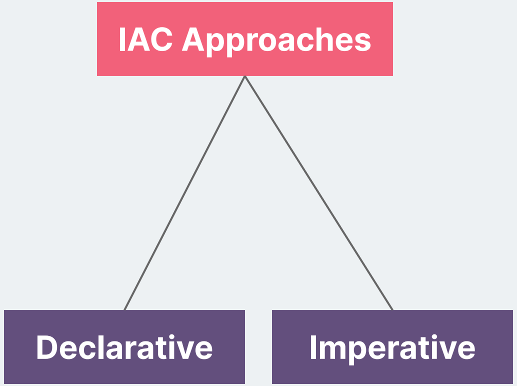
IAC Approaches

Declarative

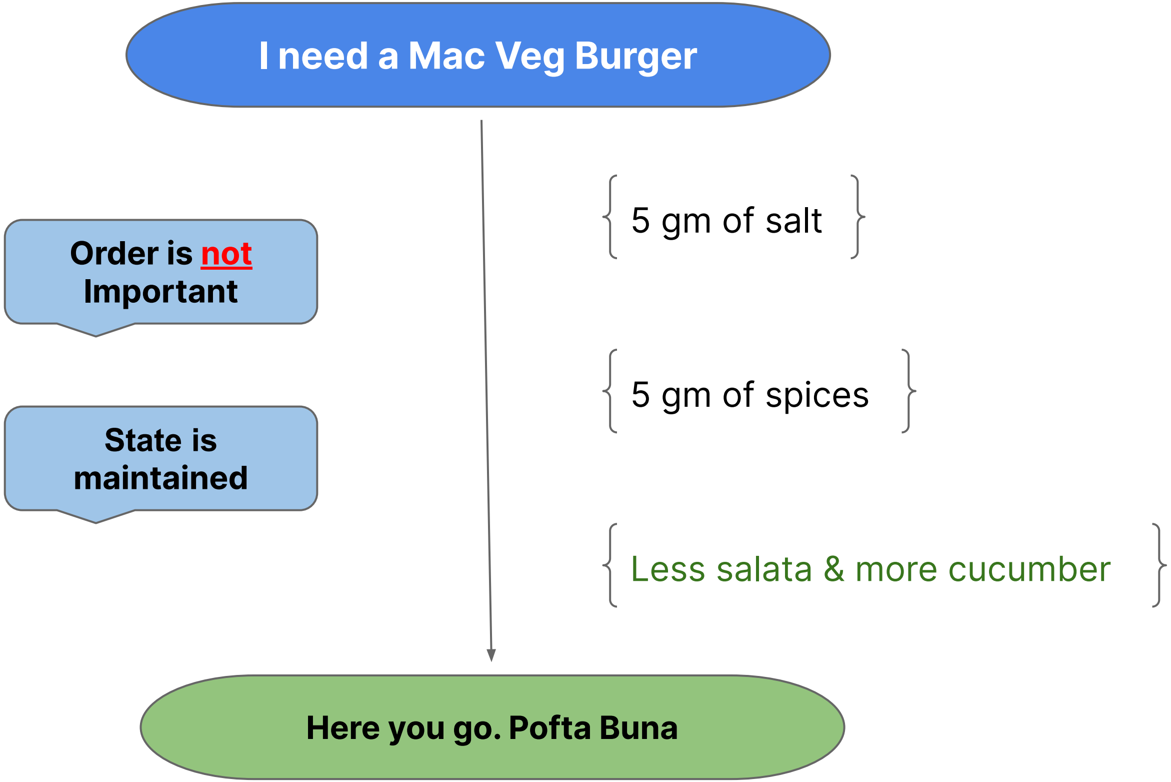
A **declarative approach** defines the desired state of the system, including what resources you need and any properties they should have, and an **IaC** tool will configure it for you.
A **declarative approach** also keeps a list of the current state of your system objects, which makes taking down the infrastructure simpler to manage.

Imperative

An **imperative approach** instead defines the specific commands needed to achieve the desired configuration, and those commands then need to be executed in the correct order.



Declarative IAC Approach



I need a Mac Veg Burger

Order is not Important

State is maintained

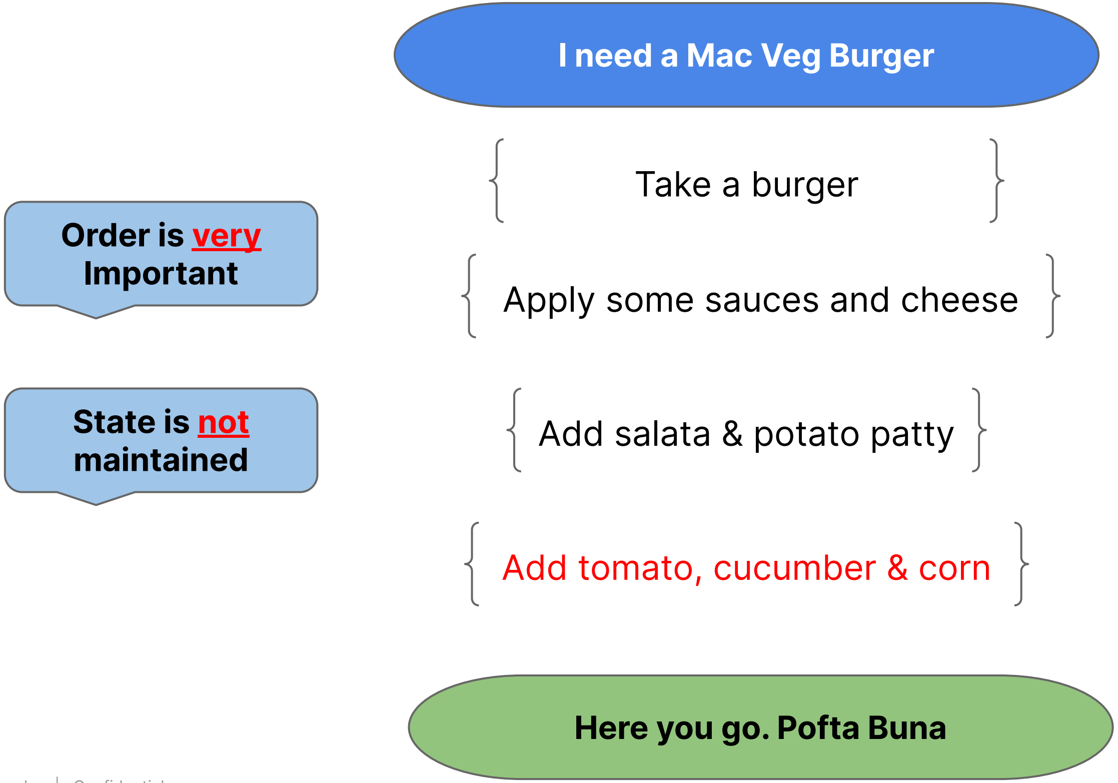
{ 5 gm of salt }

{ 5 gm of spices }

{ Less salata & more cucumber }

Here you go. Pofta Buna

Imperative IAC Approach



I need a Mac Veg Burger

Take a burger

Apply some sauces and cheese

Add salata & potato patty

Add tomato, cucumber & corn

Here you go. Pofta Buna

Order is very Important

State is not maintained

Benefits

- ❑ Using IT infrastructure as an enabler for rapid delivery of software
- ❑ Reducing the effort and risk of making changes to infrastructure
- ❑ Enabling users of infrastructure to get the resources they need, when they need it
- ❑ Providing common tooling across development, operations, and other stakeholders
- ❑ Creating systems that are reliable, secure, and cost-effective
- ❑ Make governance, security, and compliance controls visible
- ❑ Improving the speed to troubleshoot and resolve failures

Cost Reduction	**Increase in speed of deployments**
Reduce Errors	**Improve Infrastructure Consistency**
Eliminate Configuration Drift	**Documentation**
Automated Testing	**Automated Deployment**

Terraform

HashiCorp

Deployment Manager

Cloud Formation

aws

Resource Manager

Azure

IAC Tools

Terraform is one of the industry leading IAC tool. What makes terraform different from others is that it’s cloud agnostic.

Thank you!

/thoughtworks