

Terraform - Building Blocks of the Modern Cloud

Aliaksandra Kharushka

Cloud Engineer @ BestSecret



Join us on Slack

<http://shorturl.at/uzGKZ>

Agenda

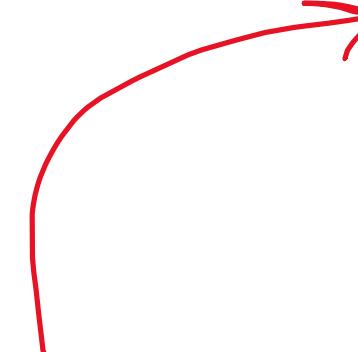
- 1. Introduction**
- 2. Definition of the Cloud**
- 3. IaaS, PaaS and SaaS**
- 4. Terraform as the Infrastructure tool**

Introduction

analogue era



"on-prem" era



cloud-native era

pictures from:

<https://intheblack.cpaaustralia.com.au/accounting/long-road-australian-female-accountants>

<https://lego.com>

<https://trends.rbc.ru/trends/education/62a8ff769a7947f537d0be91>

Definition of the Cloud – analogue era



- Data stored on: papers, tapes, punched cards
- Where: in offices, archives, closets, basements etc.
- Manual management and processing
- Data can be: burn, damaged by water, eaten by mice, stolen, become unreadable with time etc. etc.

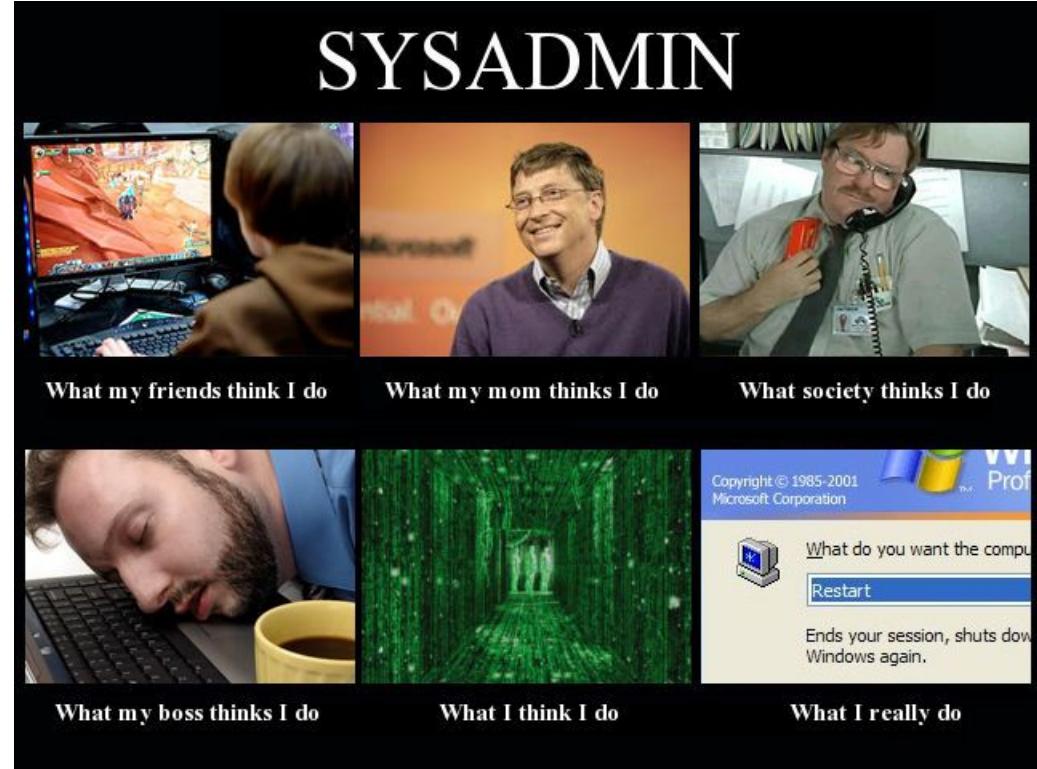
pictures from:

<https://www.factmag.com/2015/11/30/download-30-gigs-80s-cassette-underground/>
<https://www.irk.ru/news/articles/20121012/accountant/>

Definition of the Cloud – “on-prem” era



- Data stored on local servers
- Where: in the basements, in special rooms, etc.



- Have to hire system administrators to maintain the servers and computers
- Take care of the servers, networking, etc.
- Update hardware, software, fulfil security standards, backup policies ...
- Reliable, safe, expensive or unreliable, cheap

Definition of the Cloud – cloud-native era



Google Cloud



- Data is stored on remote (cloud) servers and is accessed via Internet
- You don't care of servers, data centers, hardware – everything is managed by the cloud provider

- Don't need to hire as many sys-admins, don't need to buy/setup/maintain your own servers
- Several service models:
 - **Infrastructure-as-a-Service - IaaS**
 - **Platform-as-a-Service - PaaS**
 - **Software-as-a-Service - SaaS**

Cloud service models

IaaS

- Cloud provider gives resources (infrastructure)
- You set up virtual networking, firewalls
- You build up your infrastructure as you like yourself
- Just like LEGO
- You build your **Cloud Platform** yourself
- Who works on that level: **cloud engineers**

PaaS

- You get a ready **Cloud Platform**
- You don't care about infrastructure
- You care about the **applications** you deploy to the platform
- They're automatically **scaled**, you can connect „**backing services**”
- Who works on that level: application developers

SaaS

- You get a ready-to-use product
- Typically a cloud-based application for any kind of businesses
- Who works on that level: HR specialists, accountants, doctors, teachers... etc.

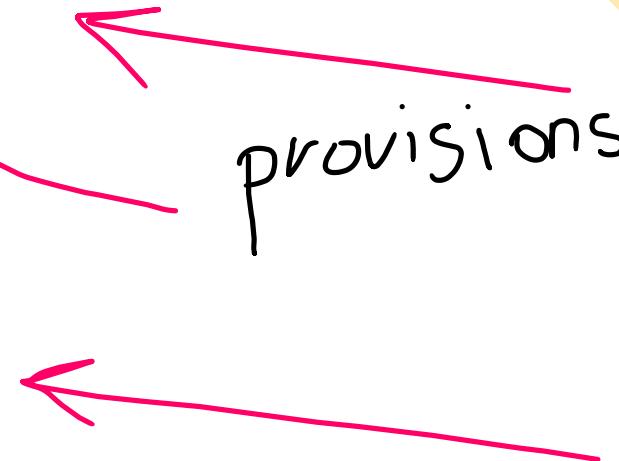
Cloud service models - IaaS

YOU



CLOUD
PROVIDER

provisions



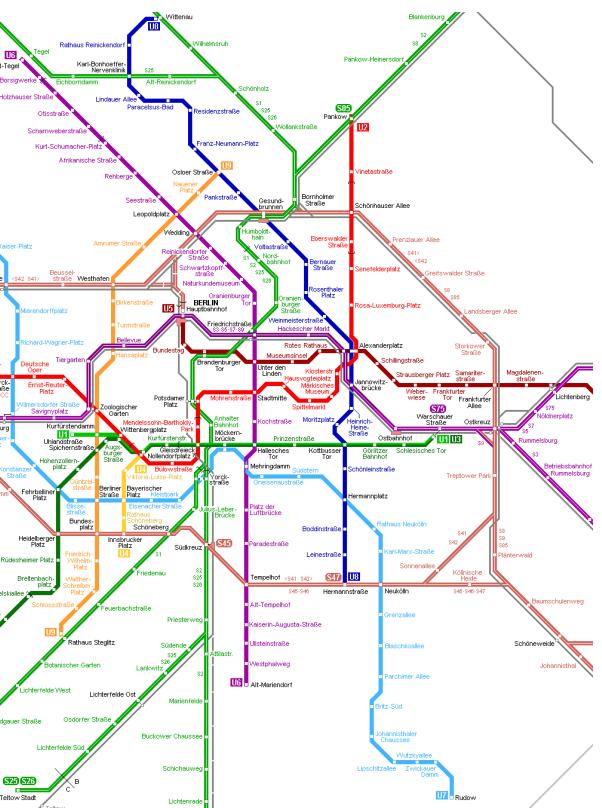
pictures from:
<https://coachwest.com/finding-a-driver-for-your-coach-bus/>
<https://citydog.io/post/odin-den-s-kontrolerami/>

Cloud service models - PaaS

YOU

BERLIN
U-Bahn
S-Bahn
Stadtbus (east-west) routes (S1, S2, S25, S26)
North-south routes (S3, S5, S7, S75, S9)
Ring line routes (341, 342, 545, 546, 547, 58, 59)
RE RB
Buses (yellow-green) — bus currently without passenger service
B/C fare zone border

2021 © U-BahnNet (R. Schwandt)



CLOUD PROVIDER



pictures from:

<https://coachwest.com/finding-a-driver-for-your-coach-bus/>

<https://citydog.io/post/odin-den-s-kontrolerami/>

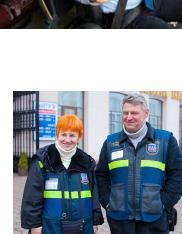
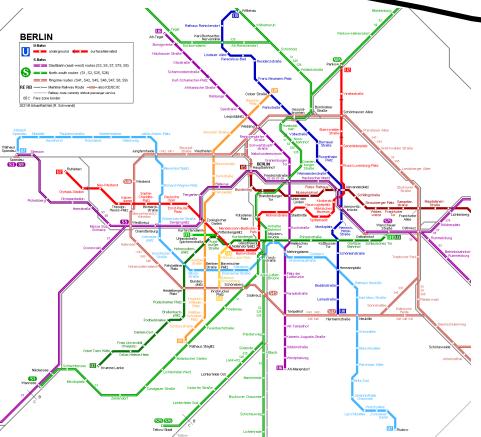
GLO
WAD PROJECT

Cloud service models - SaaS

YOU

Input:

- 10.000 daily passengers
- 10 sq. km city
- ...
- ...



CLOUD PROVIDER



GLO
MAP
PROJECT

pictures from:

<https://coachwest.com/finding-a-driver-for-your-coach-bus/>

<https://citydog.io/post/odin-den-s-kontrolerami/>

Terraform declares infra you request on IaaS level

main.tf

state

reality



Terraform

main.tf

```
terraform {  
    required_providers {  
        aws = {  
            source  = "hashicorp/aws"  
            version = "~> 4.16"  
        }  
    }  
  
    required_version = ">= 1.2.0"  
}  
  
provider "aws" {  
    region = "us-west-2"  
}  
  
resource "aws_instance" "app_server" {  
    ami          = "ami-830c94e3"  
    instance_type = "t2.micro"  
  
    tags = {  
        Name = "ExampleAppServerInstance"  
    }  
}
```

Copy 

```
0 references  
variable "location" {  
    description = "The location/region where we are creating the resource"  
    default     = ""  
}  
  
0 references  
variable "tags" {  
    description = "The tags to associate the resource we are creating"  
    type        = "map"  
    default     = {}  
}
```

```
module "org_policies" {  
    for_each      = try(var.org_policies_object, {})  
    source        = "terraform-google-modules/org-policy/google"  
    version       = "~> 5.1.0"  
    organization_id = var.org_id  
    policy_for   = "organization"  
    policy_type  = each.value.policy_type  
    enforce      = each.value.enforce  
    constraint   = each.key  
    allow        = each.value.allow  
}
```

Thanks for your attention!
Q & A



Join us on Slack
<http://shorturl.at/uzGKZ>