

# DEVOPS

Best Practices per un prodotto migliore

Dario Pasquali

# AGENDA

## DAY 1 - MATTINA

- 1 Introduzione
- 2 Dev & Ops
- 3 Best Practices
- 4 Configuration Management
- 5 Infrastrucutre As a Code
- 6 Caso d'uso
- 7 Pratica



# INTRODUZIONE



# DEVOPS

## OBIETTIVI

- Automatizzare il processo di deployment (idealmente **ONE-CLICK-DEPLOY**) in modo da aumentare notevolmente la frequenza dei rilasci.
- Migliorare la qualità del prodotto sviluppato
- Aumentare la coesione e collaborazione all'interno del team, migliorando la qualità del lavoro.



# DEV & OPS

## FIGURE IN CONFLITTO

### DEVELOPERS



DEV

**CHI:** Developers, Testers, Product Owner, QAs

**SCOPO:** implementare le features richieste del cliente rapidamente

**VALUTAZIONE:** # features funzionanti / tempo

### OPERATIONS



OPS

**CHI:** System Engineer, System Admin, DB Admin, Security

**SCOPO:** rilasciare il prodotto e assicurarsi che il sistema funzioni in modo sicuro ed efficiente

**VALUTAZIONE:** stato del sistema, soddisfazione del cliente



# DEV & OPS

## FIGURE IN CONFLITTO



Creare un unico team **multifunzionale**, che si prenda cura del prodotto durante tutto il suo ciclo di vita.

Scopo e metodi di valutazione Unificati.

Il DevOps abbate le barriere tra le persone, Developer, Data Scientist, System Engineers, Security, Testers, QAs e Product Owner, collaborano per il bene del prodotto.





**«YOU BUILD IT, YOU RUN IT»**



# DEVOPS

## CULTURA + TOOLS

DevOps = 7 Best Practices + Tools di supporto unificati

**PRATICHE** = Cambiamento Culturale, verso la Trasparenza e la Collaborazione nel team.

**TOOLS** = Guida al cambiamento, unificando Strumenti e Responsabilità





# 7 BEST PRACTICES

- 1 Continuous Management
- 2 Infrastructure As a Code
- 3 Continuous Integration
- 4 Continuous Testing
- 5 Continuous Delivery
- 6 Continuous Deployment
- 7 Continuous Management



**CONTINUOUS LEARNING**



# 7 BEST PRACTICES

- 1 **CONTINUOUS MANAGEMENT**
- 2 **INFRASTRUCTURE AS A CODE**
- 3 Continuous Integration
- 4 Continuous Testing
- 5 Continuous Delivery
- 6 Continuous Deployment
- 7 Continuous Management



**CONTINUOUS LEARNING**



# CONFIGURATION MANAGEMENT



**«ESEGUIRE IL **PROVISIONING** DELLE  
ISTANZE DEL SISTEMA USANDO  
COMPONENTI SOFTWARE **CONVERGENTI**  
E **IDEMPOTENTI**, GESTIBILI CON GLI  
STESSI STRUMENTI DEL PRODOTTO  
SVILUPPATO»**



# CONFIGURATION MANAGMENT

## PROVISIONING VIA SCRIPT

- Difficoltà di Comprensione
- Difficoltà di Condivisione tra i membri del team
- Dipendenza dalle Abitudini del singolo
- Complessa automazione della configurazione
- Nessuna garanzia di Idempotenza e Convergenza



# CONFIGURATION MANAGMENT

## IDEMPOTENZA & CONVERGENZA

**IDEMPOTENZA** = L'applicazione multipla della stessa operazione, dopo la prima esecuzione, non ha effetti sul sistema target

**CONVERGENZA** = un'operazione è eseguita se e soltanto se è strettamente necessario.



# MODELLI DI CM

## IMPERATIVO

Configurazione come  
**SEQUENZA DI AZIONI** da  
eseguire per raggiungere lo stato  
desiderato.

Modello **MASTER-SLAVE**  
necessita di un agente installato sul  
target

**MASSIMO CONTROLLO** ma  
necessità di conoscere il processo



# MODELLI DI CM

## DICHIARATIVO

Configurazione come **STATO DA RAGGIUNGERE**, specificato in modo Descrittivo

Modello **MASTER-SLAVE**  
necessita di un agente installato sul target

**MASSIMA ASTRAZIONE** ma  
perdita del controllo nel dettaglio





# MODELLI DI CM

## IBRIDO

Configurazione come sequenza di azioni astratte in forma di **ROLES**

Modello **AGENT-LESS** basato su SSH e Python

**MASSIMA MODULARITA'** e customizzazione al livello di dettaglio desiderato



# MODELLI DI GESTIONE

## ARCHITETTURA DINAMICA – FRY MODEL

Istanza Vergine

Provisioning

Deploy



Mantenimento  
dello stato

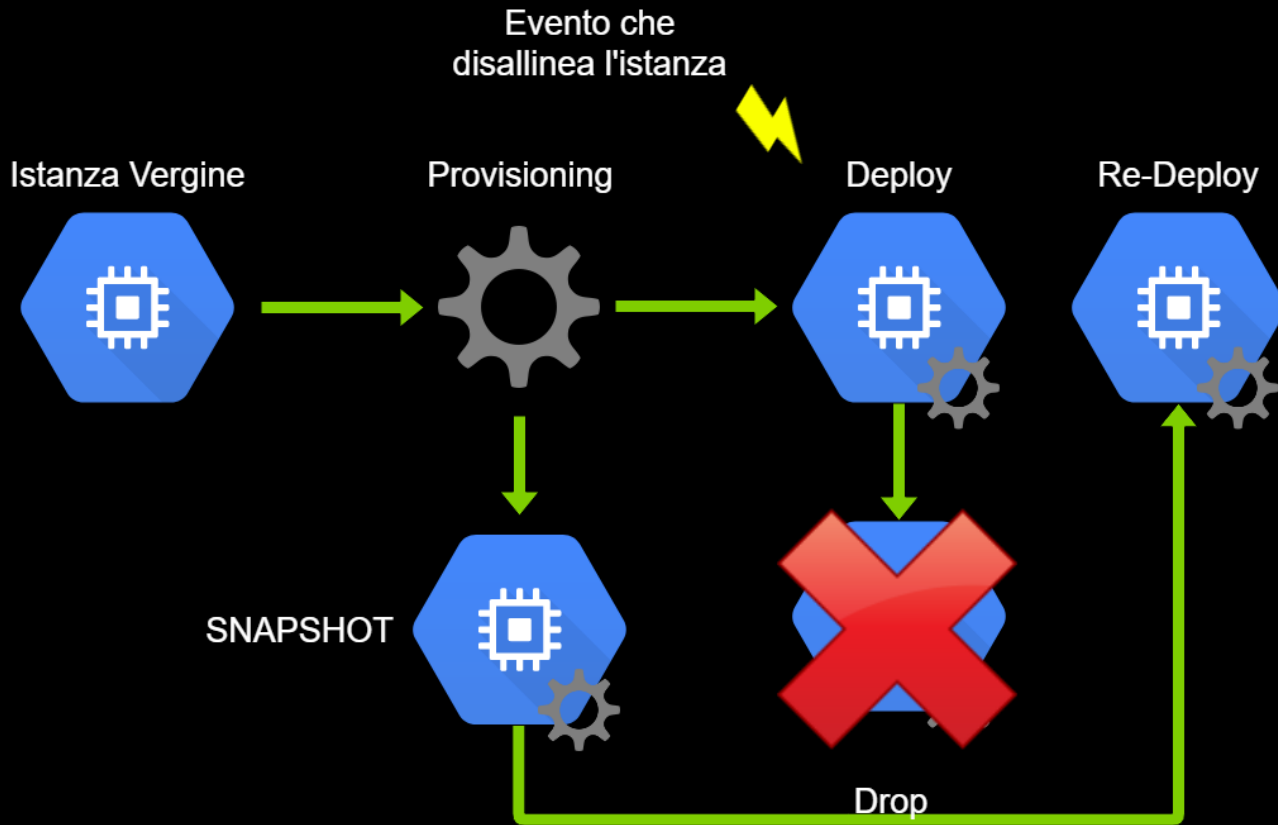


Evento che  
disallinea l'istanza



# MODELLI DI GESTIONE

## IMMUTABLE INFRASTRUCTURE – BAKED MODEL



# ANSIBLE

## CM IBRIDO

- Tool di CM ibrido, unisce i vantaggi degli approcci Imperativo e Dichiarativo
- Possibilità di specificare la configurazione come sequenza di azioni, o come astrazione di **ROLE** nel sistema
- Provisioning Agent – less basato su SSH



# ANSIBLE

## CM IBRIDO

**PLAYBOOK** = modulo auto – contenuto, scritto in YAML che descrive la configurazione da applicare ad un'istanza

**INVENTORY** = modulo auto – contenuto, scritto in YAML, che definisce la configurazione e l'istanza target

**ROLE** = Astrazione standalone di una configurazione del sistema



# ANSIBLE

## PLAYBOOK

### PLAYBOOK

Definisce il target del Provisioning e la sequenza di task da applicare.

Yum e get\_url sono moduli che racchiudono un comportamento

Target definiti tramite labels

```
- hosts: cluster-nodes
  become: yes
  become_user: root
  tasks:
    - name: Install JDK
      yum:
        name: java-1.8.0-openjdk
        state: present

    - name: Download SBT
      get_url:
        url: http://dl.bintray.com/sbt/rpm/sbt-0.13.12.rpm
        dest: /opt/sbt-0.13.12.rpm
        mode: 0440

    - name: Install SBT
      yum:
        name: /opt/sbt-0.13.12.rpm
        state: present
```



# ANSIBLE

## INVENTORY

### INVENTORY

Definisce le istanze target

Possibilità di:

- Creare gruppi di istanze identificate da label
- Usare regex nei nomi
- Configurare parametri specifici di istanza (porta, utente, chiave, ...)

```
[local]
```

```
127.0.0.1
```

```
[cluster-nodes]
```

```
gcp-cluster-node-[01:50]
```



# ANSIBLE

## ROLE

### ROLE

Componente **STANDALONE** che racchiude e definisce una responsabilità nel sistema.

Arricchito da tasks, variabili, tests, handlers, documentazione, metadati.

Condivisione e distribuzione su Ansible Galaxy

```
- hosts: cluster-nodes
  become: yes
  become_user: root
  roles:
    - sbt
```





# **INFRASTRUCTURE AS A CODE**



**«ORCHESTRATION  
DELL'INFRASTRUTTURA CON  
COMPONENTI ESEGUIBILI, IDEMPOTENTI E  
CONVERGENTI, GESTIBILI CON GLI  
STESSI STRUMENTI DEL PRODOTTO  
SVILUPPATO»**



# INFRASTRUCTURE AS A CODE

## IAC

### INFRASTRUTTURA

Modello eseguibile, Convergente, Idempotente e **REPLICABILE** in grado di essere compreso ed utilizzato da tutti i membri del team.

Nessuna dipendenza da un team di sistemisti o da complessi script bash.



# TERRAFORM

## ORCHESTRATION PROVIDER CLOUD

Orchestration di infrastrutture Cloud sui più comuni Cloud Providers ( GCP, AWS, Docker, OpenStack, ...)

Oltre a numerosi servizi standard (GitHub, Slack, DNS, ...)

Esecuzione **AGENT-LESS** via SSH, idempotente e convergente.



# TERRAFORM

## ORCHESTRATION PROVIDER CLOUD

Configurazione Dichiarativa, specifica lo  
**STATO DESIDERATO**.

Terraform si occupa di trovare il modo migliore per raggiungerlo.

Esecuzione idempotente basata sullo stato attuale e sulle azioni passate di Terraform.

### PLAN

Componente eseguibile che rappresenta lo snapshot di un'infrastruttura cloud. Replicabile a piacimento.

```
provider "google" {  
  credentials = "${file("terraform-admin.json")}"  
  project     = "endless-upgrade"  
  region     = "us-east1-b"  
}  
  
resource "google_compute_instance" "worker" {  
  project = "endless-upgrade"  
  zone    = "us-east1-b"  
  count   = "${var.server_count}"  
  name    = "gcp-cluster-node-${count.index}"  
  machine_type = "n1-standard-2"  
  tags     = ["jenkins", "monitoring", "http", "https"]  
    
  boot_disk {  
    initialize_params {  
      image = "centos-7-v20171213"  
      size  = "30"  
    }  
  }  
}  
  
output "worker" {  
  value = "${google_compute_instance.worker.self_link}"  
}
```



# TERRAFORM

## PROVISIONING

Terraform permette di eseguire il Provisioning con Tool esterni...

...come **ANSIBLE** !!

```
...  
  
provisioner "local-exec" {  
  command = "sleep 90;  
  ansible-playbook -i '${google_compute_instance.worker.name},'  
  --private-key=~/.ssh/ansible_rsa /opt/ansible/cluster-nodes.yml  
  -e 'ansible_ssh_user=dario_pasquali93' -e 'host_key_checking=False'"  
}  
}  
  
...
```



A person is paragliding over a mountain range. The paraglider is in the upper right quadrant, with a blue and white canopy. The background shows a vast landscape of mountains and valleys under a cloudy sky. The text is overlaid on the center of the image.

**CM + IAC**

**=**

**CREAZIONE E PROVISIONING  
AUTOMATICO BY-NEED**

Possibilità di Scalare il Sistema a piacimento, in modo  
Idempotente, Convergente e Replicabile



# CASO D'USO





# PRATICA



