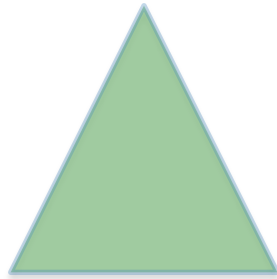


Cliente Telco: miglior time to market con Microservizi e DevOps su Openshift

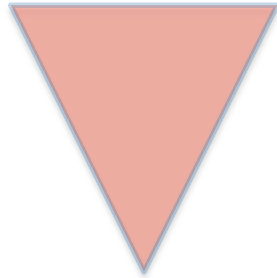
*Alessandro Giangreco, Team Manager, Sirti*  
*Eugenio Marzo, DevOps Engineer, Sourcesense*

# Obiettivi



Time to market

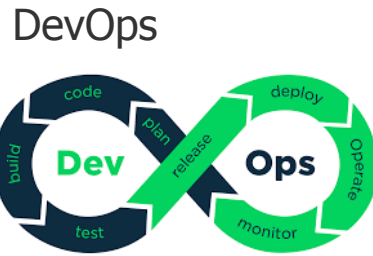
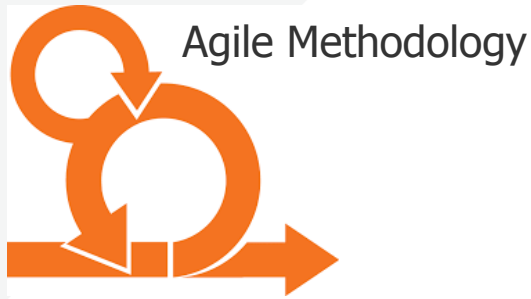
Faults



Aumentare la velocità nell'attivare nuovi servizi e **gestire nuovi apparati e tecnologie per stare al passo con le evoluzioni di rete.**

Partendo dall'esigenza di business fino al rilascio in produzione. Migliore time to market.

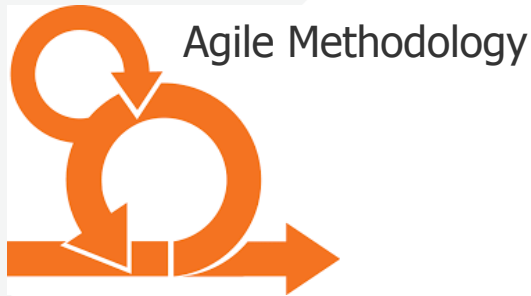
Diminuire gli errori di attivazione dei nuovi servizi.



Microservices

## Aree principali d'intervento:

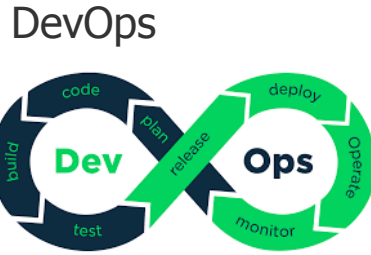
- Processi aziendali;
- Automazione, al fine di massimizzare la produttività e ridurre le inefficienze (individuazione di ciò che è definito e ripetibile);
- Applicazione di metodologie e tecnologie innovative a supporto del cambiamento;
- Utilizzo di architetture moderne in ambito enterprise;
- Sviluppo delle Soft Skills: velocità, elasticità e flessibilità.



Agile Methodology

## Perché DevOps e microservizi

Aumentare la velocità di deployment e la velocità con cui si è in grado di soddisfare le richieste di mercato, diminuendo costantemente il Time-to-Market.



DevOps

Come

- Diverso approccio culturale e conseguente revisione dei processi aziendali;
- Utilizzo di metodologie, strumenti, e tecnologie a supporto dell'innovazione;
- Architetture moderne in ambito enterprise, che consentono di focalizzarsi su logiche di business.

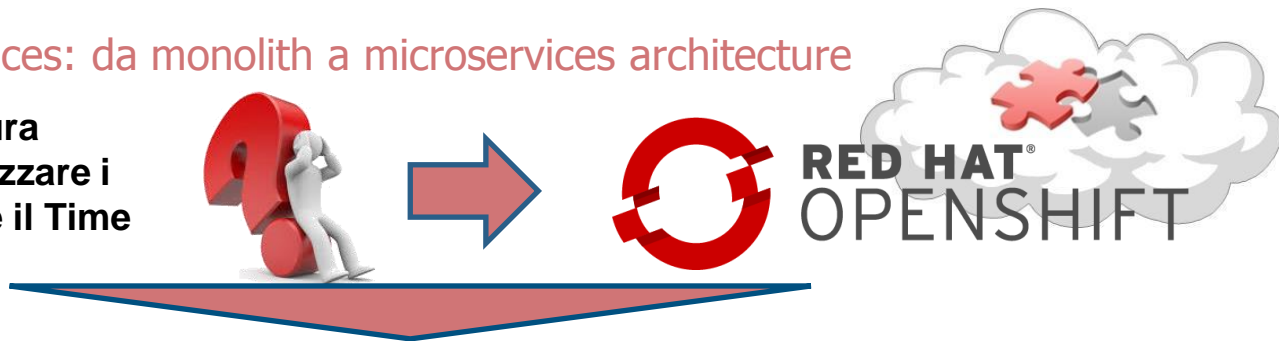


Microservices

# I progetti in TELCO Assurance e Network/Service Creation

Microservices: da monolith a microservices architecture

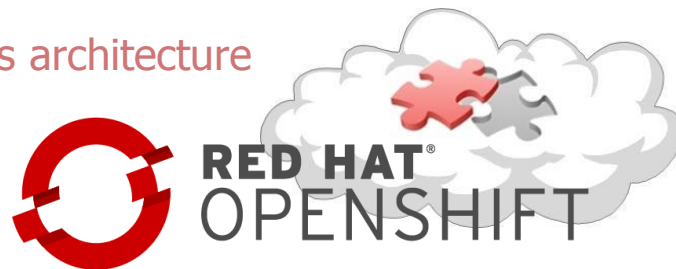
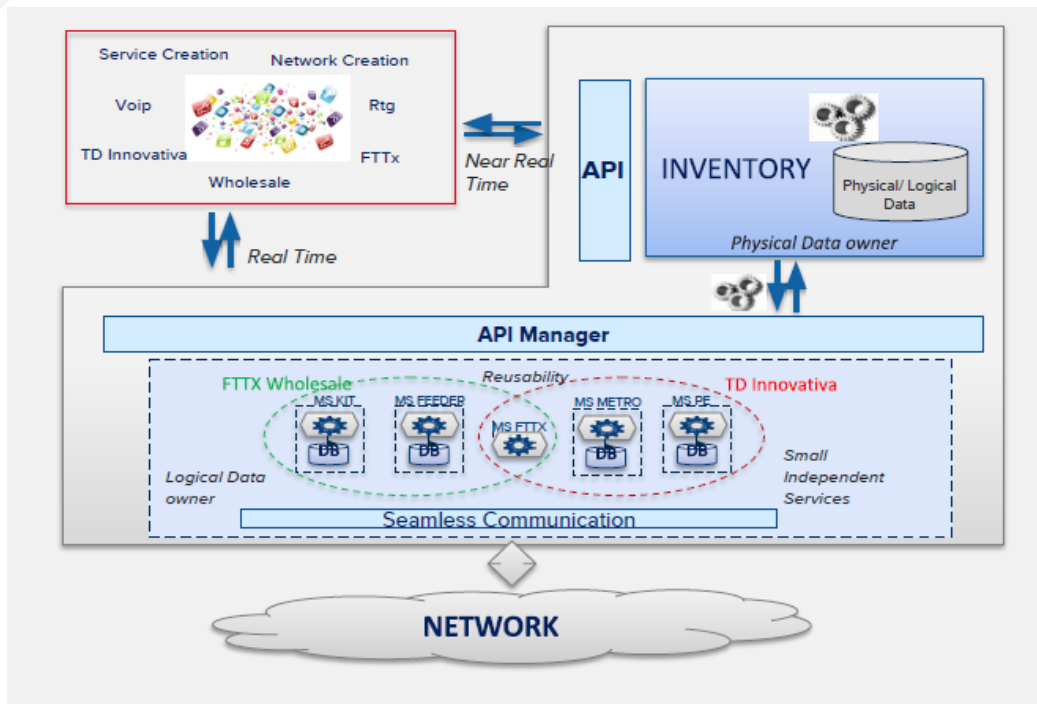
Come realizzare un' architettura software, in grado di massimizzare i vantaggi del DevOps e ridurre il Time to Market?



- 4 L'architettura a microservizi ha abilitato la capacità di poter modificare in modo puntuale il software, riducendo la diffusione degli impatti, a vantaggio della Continuous Integration, Continuous Delivery e Continuous Deployment, semplificando la revisione del processo interno di delivery (ancora in atto);
- 4 La PaaS è risultata essere a supporto della governance, deployment e orchestrazione dei microservizi. I dev team si sono concentrati sull'implementazione delle funzionalità di business, demandando alla piattaforma gli aspetti di scalabilità, monitoring, logging, deployment e riducendo la quantità di codice custom (script, ecc.).

# I progetti in TELCO Service Creation use case

Microservices: da monolith a microservices architecture

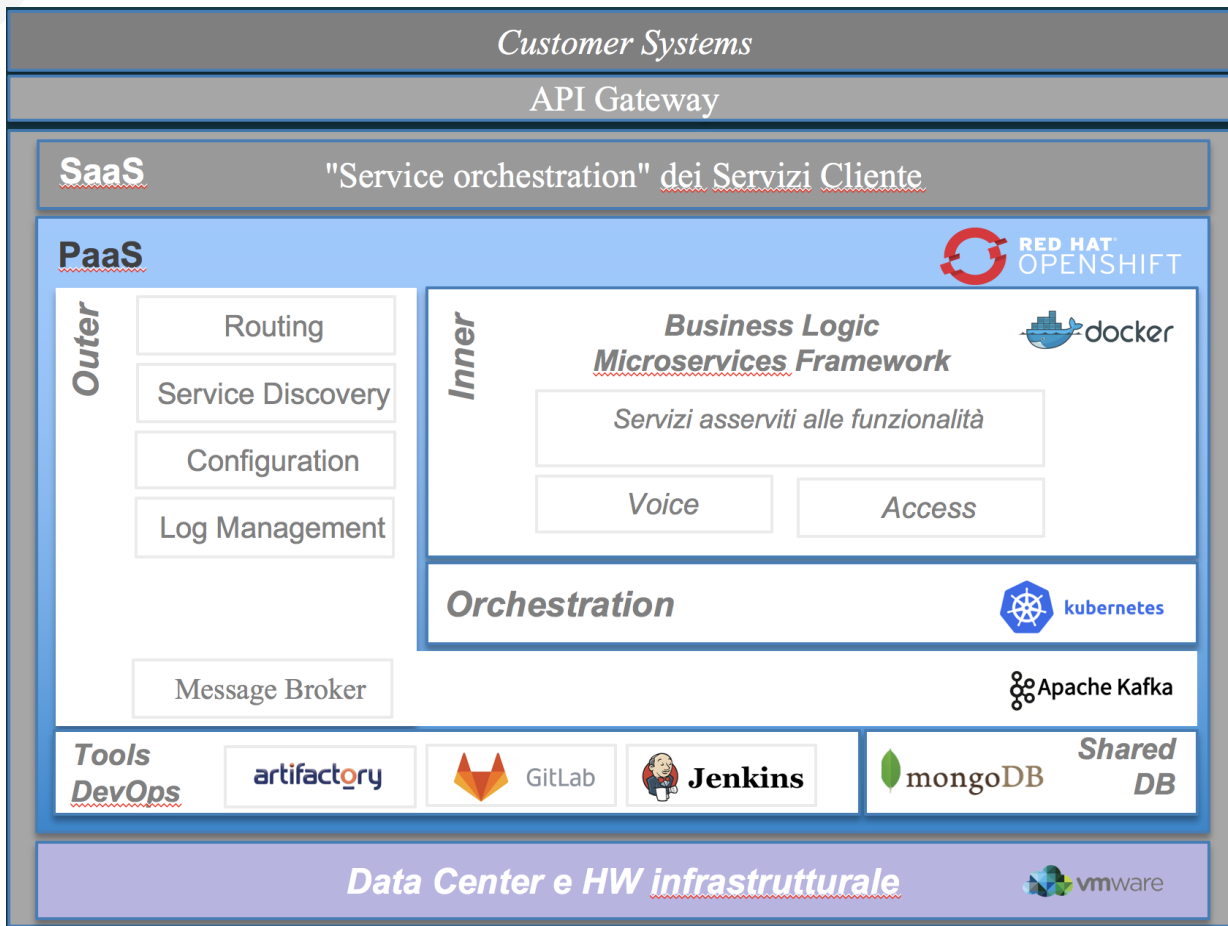


## KPI Activation

- ~118K apparati gestiti
- ~10,314 Mln servizi attivati
- ~18K servizi attivati per Day (IP, FTTC, FTTH, VLAN IP/Fibra)

## Test di carico e scaling

- In situazione di sovraccarico è stato effettuato lo scaling via PaaS dei microservizi. I tempi medi di gestione dell'OL si sono ridotti considerevolmente.



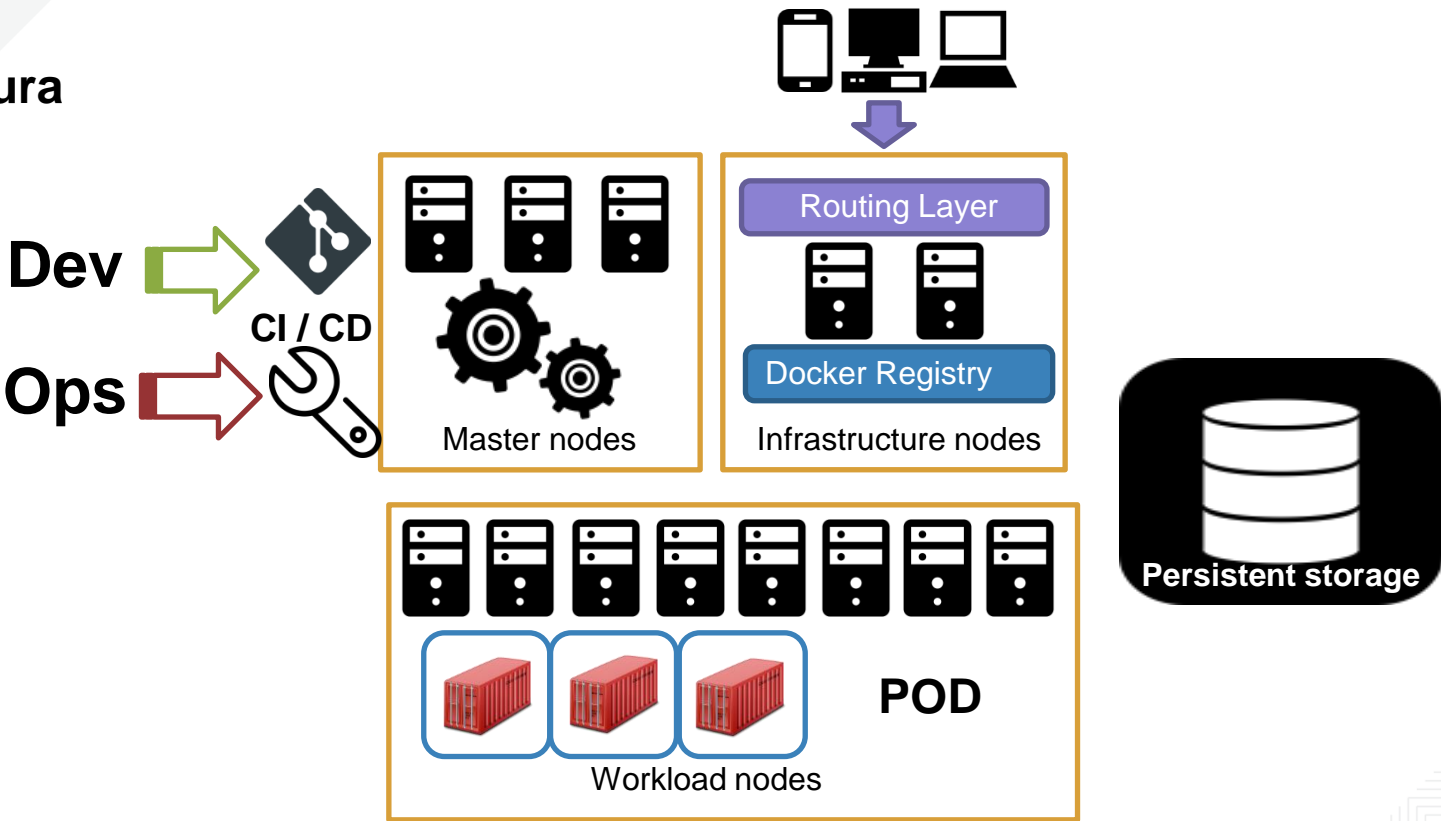
# Installazione Openshift Container Platform



- 3 nodi master
- 9 nodi workload
- 2 nodi infra (router)



# L'architettura

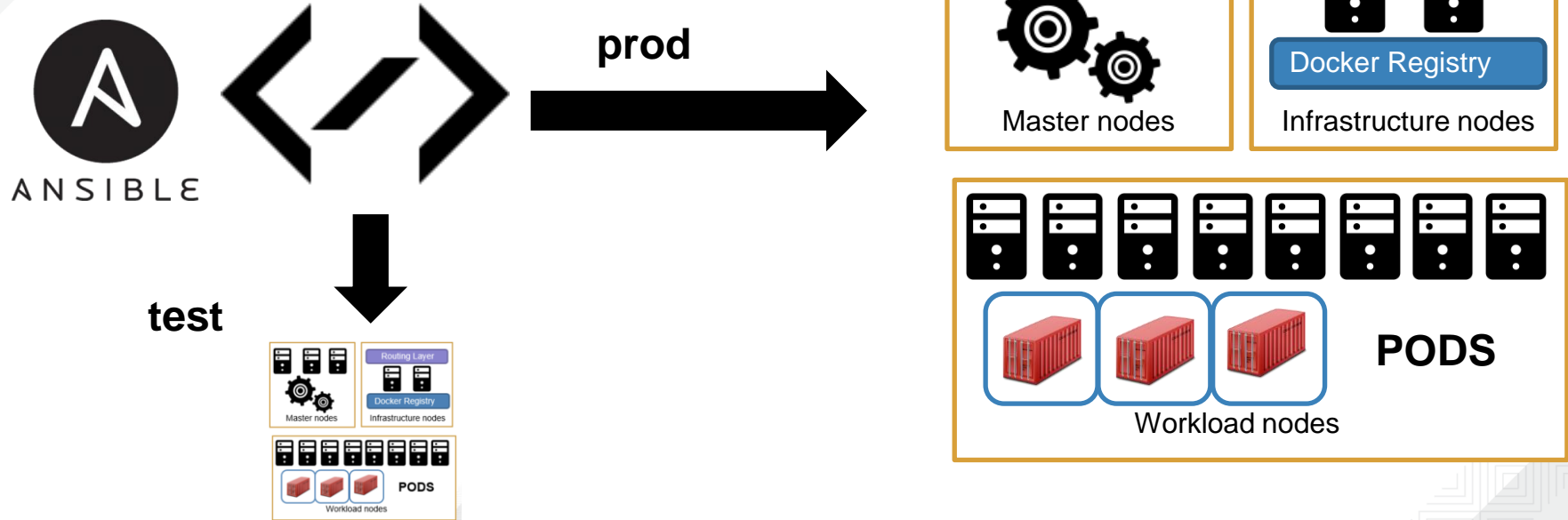


# Ansible per l'installazione

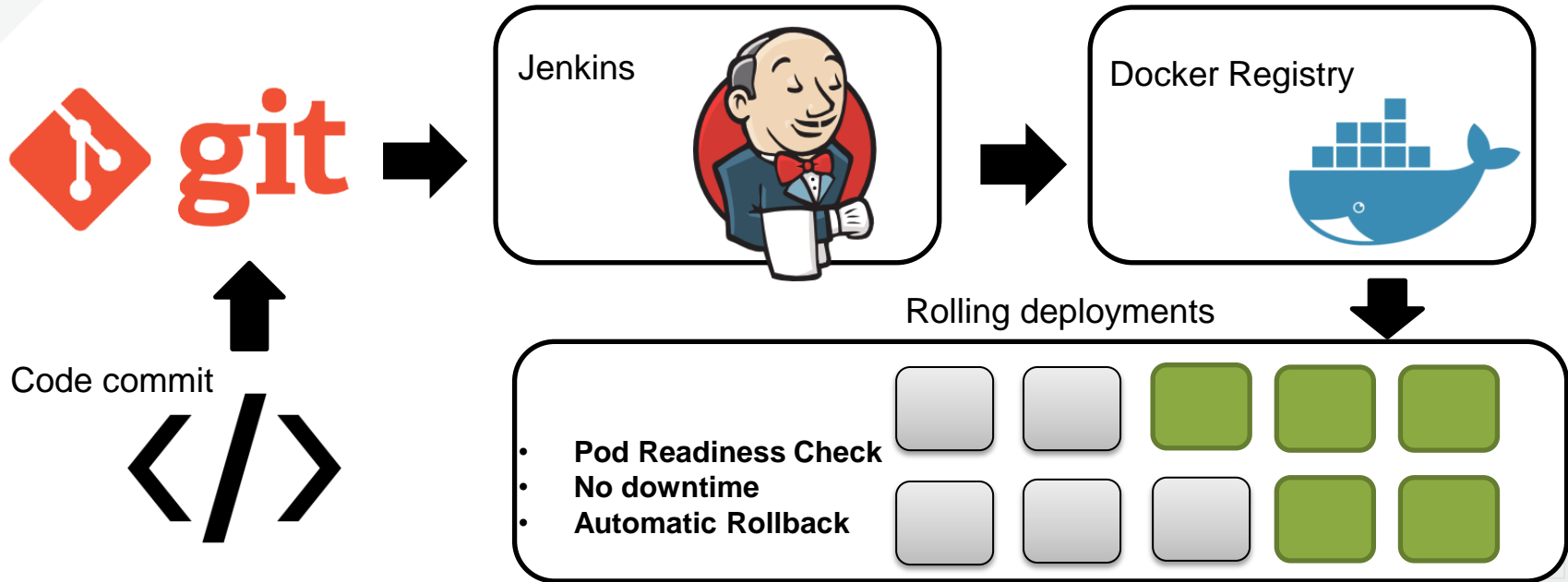


- Controllo e fix rapidi dei prerequisiti;
- Tempi brevi per installazione Openshift;
- Overview caratteristiche di ogni nodo.

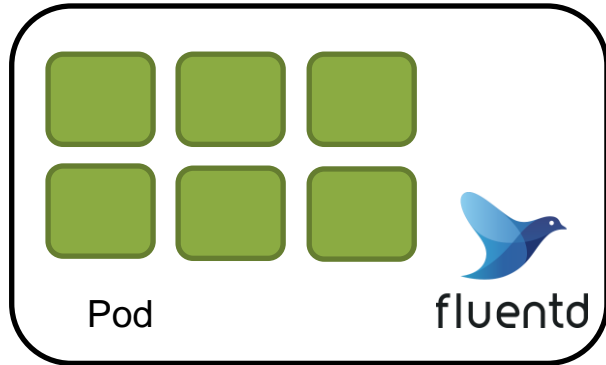
# Infrastructure as code



# Deploy delle immagini Docker

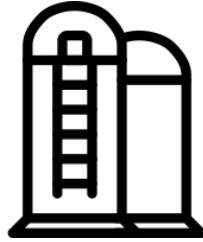


# Collezionamento dei log (stack EFK)



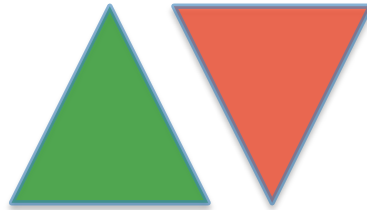
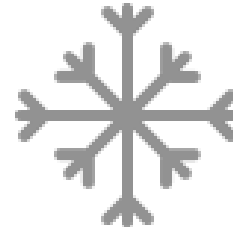
- Playbook Ansible per un'agevole installazione;
- Log centralizzati;
- Multi-tenant.

# Risultati



No *silos*:  
dev e operation lavorano  
a stretto contatto

No *snowflakes*:  
nessun ambiente è unico  
ed irreplicabile



Time to market: **x10**  
Faults: from 6% to **1%**

# Grazie!



# RED HAT OPEN SOURCE DAY

Europe, Middle East & Africa

**SOURCESENSE**



#redhatosd