# PIB: Una soluzione alla complessità di installazione dei cluster moderni

Alberto Damo Alessandro Pirolo

August 30, 2024

# Indice

# 1 Preambolo

Il mondo di oggi è estremamente interconnesso e complesso. Per questa ragione, ogni applicativo pensato per il grande pubblico deve garantire l'affidabilità del servizio in ogni momento. Per risolvere questi problemi, vengono spesso utilizzati sistemi distribuiti in rete, detti anche cluster, che permettono di garantire la scalabilità e l'affidabilità del servizio. Tuttavia, questi sistemi sono estremamente complessi e difficili da installare. Basti pensare alla procedura di configurazione di un cluster Kubernetes o Ceph, che è delicata e richiede molto tempo per essere completata, a meno che non si utilizzino strumenti esterni per facilitare il lavoro, che non sempre sono affidabili. Tale complessità risiede, tra i tanti fattori, nell'architettura scelta per modellare il sistema, solitamente di tipo Orchestrator. Questo modello offre numerosi vantaggi, ma a un costo: i nodi che compongono il cluster non sono omogenei. Questo dettaglio, per quanto sembri insignificante, è ciò che rende complessa la procedura di configurazione, aggiornamento e mantenimento del cluster. In questo paper descriveremo come abbiamo creato un cluster che garantisce le richieste di affidabilità necessarie, permettendo inoltre una veloce e semplice installazione/configurazione.

## 2 Composizione di un cluster

Prima di poter analizzare i nostri obiettivi, come li abbiamo raggiunti e perché siano state fatte determinate scelte, è necessario descrivere gli elementi che compongono un cluster e le loro caratteristiche:

- **Nodi**: Sono i computer che si occupano della computazione necessaria per mantenere il servizio attivo. Possono essere fisici (server dedicati) o virtuali (macchine virtuali, container).
- **Gestore dei nodi**: Questo componente si occupa dell'aggiunta, rimozione e sostituzione dei nodi nel cluster. Può essere un hypervisor, nel caso i nodi siano virtuali, o un operatore, nel caso i nodi siano server reali. È probabile che ci siano più gestori, soprattutto se i nodi sono geograficamente distanti.
- **Gestore di rete**: Questo componente gestisce le connessioni interne ed esterne al cluster. In particolare, si occupa dell'assegnazione degli indirizzi di rete ai nodi (è possibile che ogni nodo abbia più interfacce di rete e quindi più indirizzi IP).
- **Sistema operativo**: Sebbene appartenga al nodo, il sistema operativo deve essere progettato su misura per il cluster, poiché anche questo componente è responsabile per la corretta operatività dell'intera infrastruttura. In particolare, dovrà:
  - Ottenere gli indirizzi di rete (è possibile siano più di uno per ogni nodo).
  - Eseguire una diagnostica interna sullo stato della macchina.
  - Eseguire eventuali programmi o daemon responsabili delle funzionalità del cluster e degli applicativi.

## 3 Obiettivi

Come già accennato, il nostro obiettivo è realizzare un cluster che sia allo stesso tempo affidabile e semplice da configurare/installare.

In particolare, il sistema dovrà:

- Permettere la scalabilità del servizio aumentando o diminuendo il numero di nodi senza interrompere i servizi forniti.
- Garantire la disponibilità del servizio anche in caso di guasti sia a livello fisico (server, rete), sia a livello software (crash del sistema operativo o dell'applicativo stesso).
- Integrare tool per l'interazione dell'amministratore con il cluster.
- Essere privo di single point of failure.
- Avere tutti i nodi del cluster uguali tra di loro a livello strutturale.