



Illustrata

**Guida di Kubernetes
per piccoli naviganti**

Brought to you by...



Scritto da: Matt Butcher

Illustrato da: Bailey Beougher

Ideato da: Karen Chu

L'illustrazione di Goldie è basata su Go Gopher ideato da Renee French

Tradotto da Annalisa Gennaro ed editato da Christian Boragine di SparkFabrik



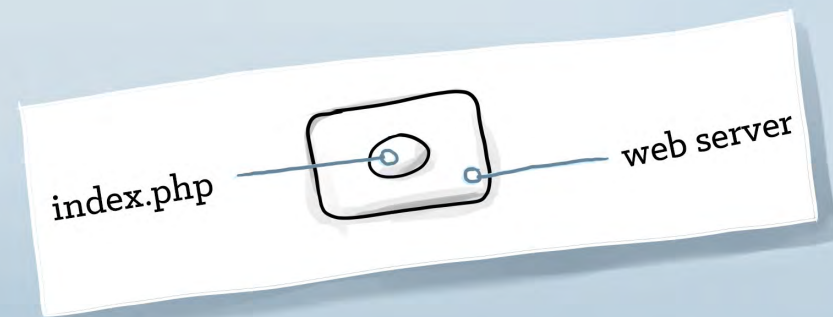
Dedicato a tutti quei genitori che cercano di spiegare ai figli il proprio lavoro di sviluppatori.



C'era una volta un'app che si chiamava Phippy. Era un'app semplice, di una pagina soltanto, scritta in PHP. Viveva su un hosting che condivideva con altre app paurose che lei non conosceva e che non voleva conoscere. Avrebbe desiderato tanto avere uno spazio tutto per sé, lei e un web server che potesse chiamare casa.

L'ambiente

- L'app (index.php)
- La struttura che la supporta
 - > web server
 - > vari parti di OS



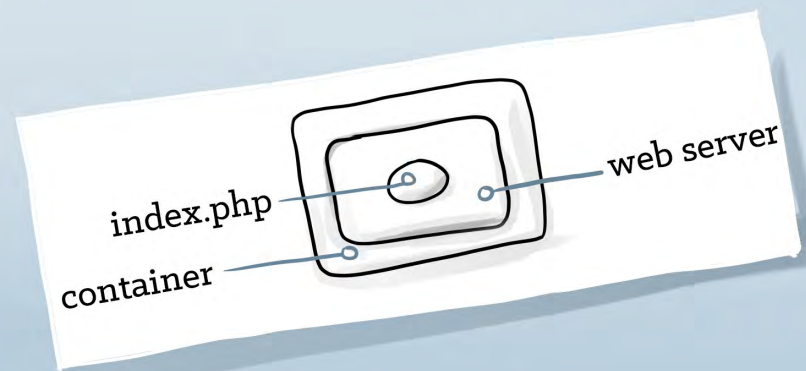
Un'app funziona perché può girare in un ambiente adatto. Per un'app PHP quell'ambiente può includere un web server, un filesystem leggibile e un motore PHP.



Un giorno Phippy incontrò una balena gentile che le suggerì che sarebbe stata molto più felice a vivere in un container. Phippy ci si trasferì immediatamente. Ma per quanto il container fosse tanto carino, era come vivere in un salotto fluttuante nel bel mezzo dell'oceano.

Containers

- Hanno bisogno di essere gestiti
- Il networking non è facile
- I Container devono essere schedulati, distribuiti e bilanciati
- I dati persistenti devono stare da qualche parte



Un container fornisce un contesto (uno spazio) isolato nel quale un'app gira indisturbata. Ma tanti containers diversi hanno bisogno di essere gestiti e connessi al mondo esterno. Filesystem condivisi, la rete, la schedulazione, il bilanciamento e la distribuzione sono tutte sfide per nulla facili da cogliere.



La balena alzò le spalle: "Beh, scusami tanto" e scomparì tra le onde dell'oceano. Ma prima ancora che Phippy potesse iniziare a disperarsi, all'orizzonte apparve una nave gigantesca condotta da un curioso Capitano. La nave in realtà era costruita da decine di zattere tutte legate insieme, ma dall'esterno pareva proprio un'unica barca.

"Salve piccola app! Il mio nome è Capitano Kube", disse il vecchio e saggio capitano.

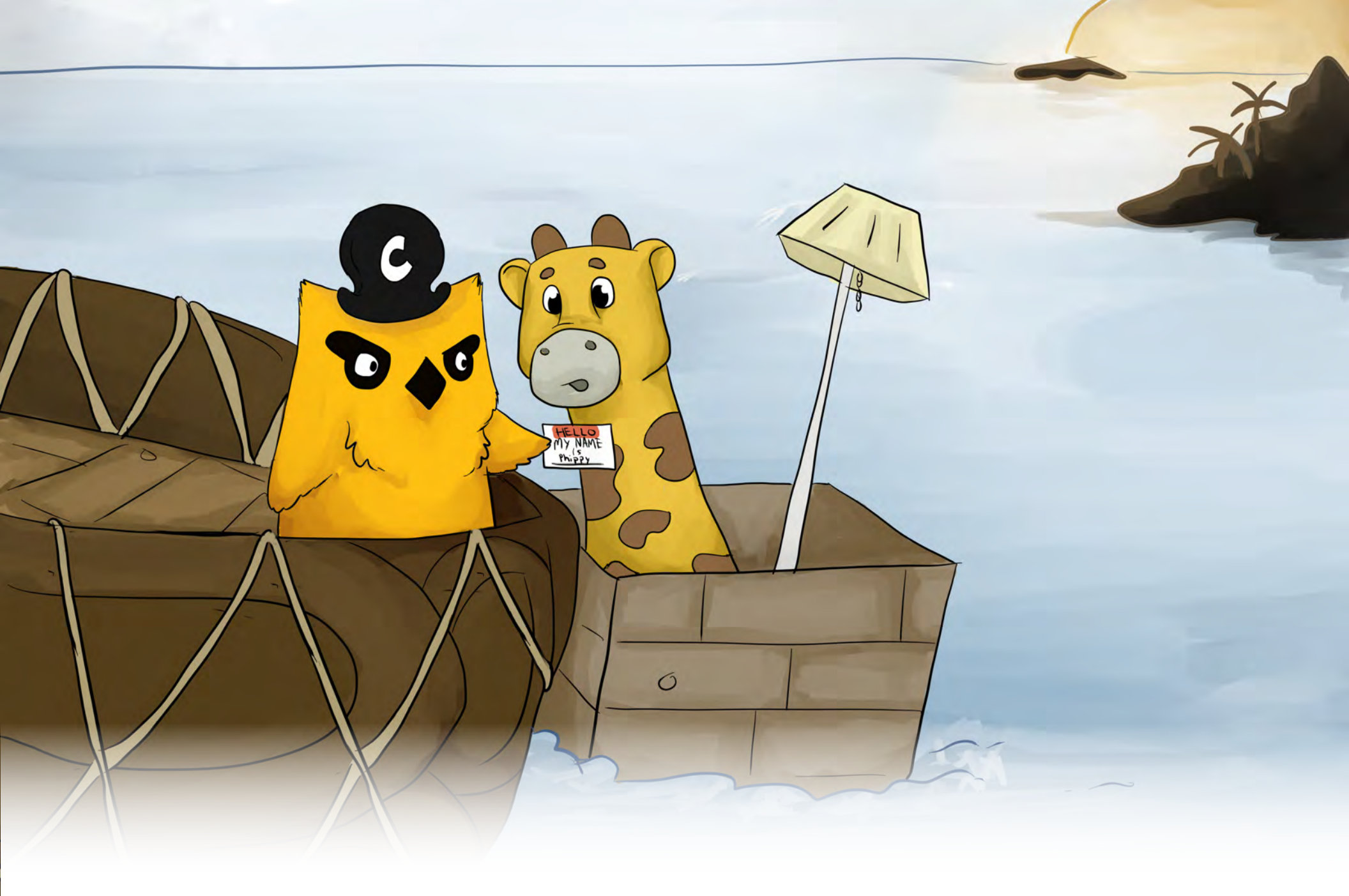
Kubernetes

- Phi-Beta-Kappa (ΦΒΚ) sta per Φιλοσοφία Βίου Κυβερνήτης (Philosophia Biou Kybernētēs), in altre parole: L'amore per il Sapere è la Guida nella nostra Vita.



kubernetes

Kubernetes è un termine di origine greca che indica il timone o il capitano di una nave e dal quale derivano parole quali cibernetica e governo. E infatti, il progetto Kubernetes ha come obiettivo la creazione di una piattaforma solida per gestire in produzione migliaia di container.



Sono Phippy", disse la piccola app.

"Piacere di conoscerti", rispose il Capitano appuntandole una targhetta con il suo nome al petto.

Kubernetes usa etichette (labels)

- Le ricerche (query) si basano su queste etichette



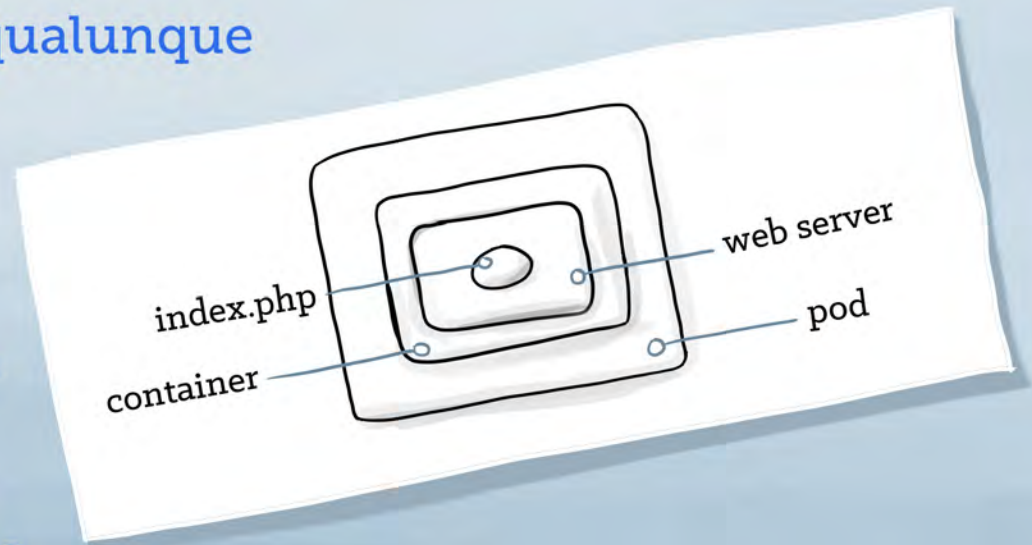
Kubernetes usa etichette come targhette identificative aperte (a tempo indeterminato) con le quali noi possiamo indicare ruoli, livello di stabilità o altri importanti attributi.



Il Capitano Kube invitò la piccola app a portare il suo container all'interno di una delle zattere (pod) a bordo della nave. E Phippy ne fu felice! Si sentiva finalmente a casa.

Pods

- Un pod può contenere un qualunque numero di container, ma di solito si limita a due
- Facciamo finta che uno di questi container non esista
- Un pod è collegato al resto dell'ambiente attraverso una rete che sta sopra un'altra rete (overlay network)



Un pod rappresenta un'unità di software eseguibile. In genere, in un pod gira un unico container, ma nel caso in cui dei container fossero strettamente legati, allora questi potrebbero girare serenamente all'interno dello stesso pod (chi vorrebbe mai separarli??). Kubernetes si preoccupa di connettere il pod alla rete e al resto dell'ambiente.

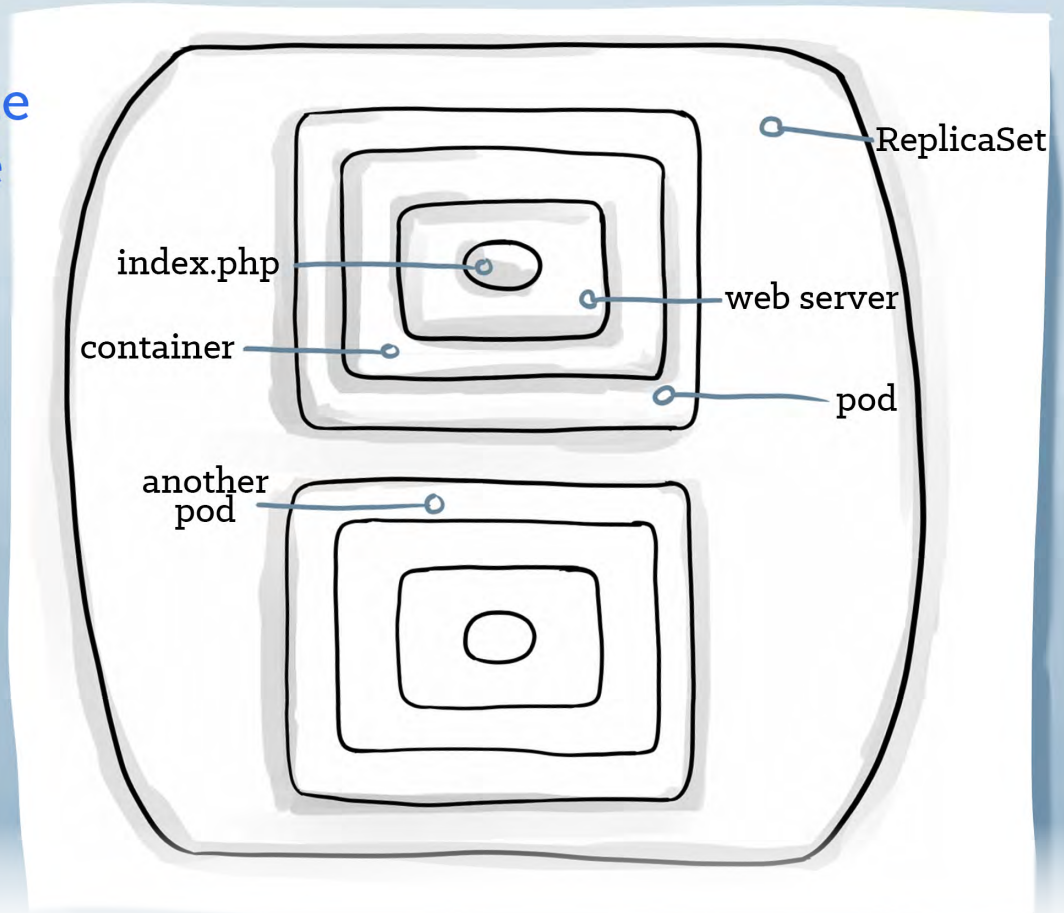


Phippy coltivava interessi fuori dal comune - le piacevano tanto la genetica e le pecore, tanto che chiese al Capitano: "Cosa accadrebbe se volessi clonarmi? Più volte, su richiesta?"

"Non ci sarebbero problemi!", rispose il Capitano che le presentò ReplicaSets.

ReplicaSets

- Mette a disposizione un template per pod per crearne un numero infinito di copie
- Fornisce le logiche per scalare il pod in su e in giù
- Può essere usato per mettere a terra i deploy



ReplicaSets è una funzione fantastica che fornisce un metodo per gestire un numero arbitrario di pod e che contiene un template che può essere replicato potenzialmente all'infinito. Attraverso ReplicaSets Kubernetes può supervisionare l'intero ciclo di vita di un pod, inclusa la sua scalabilità, gestire deploy e monitorare il tutto.



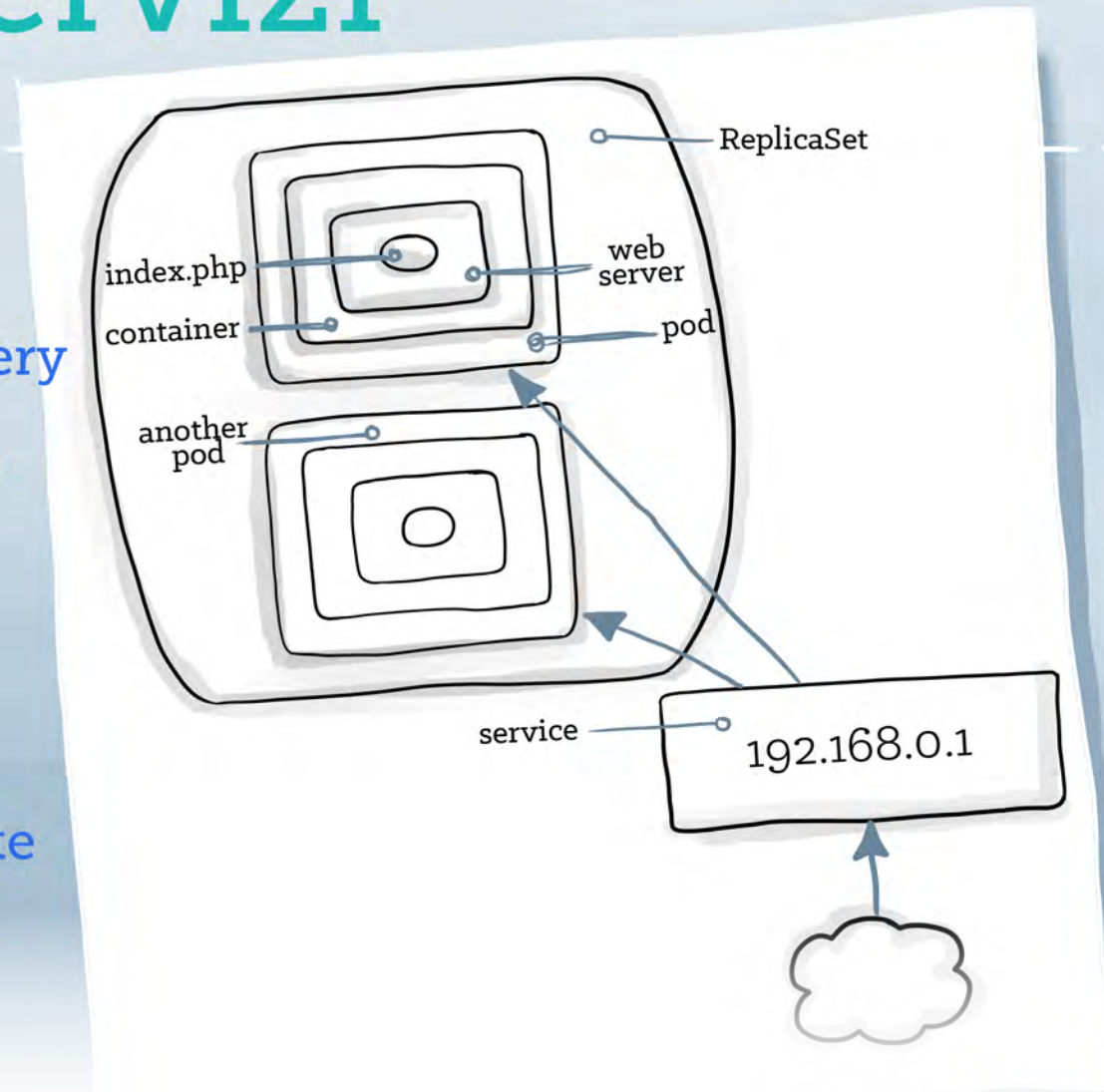
La piccola app trascorse molti giorni e molte notti felici nel pod con i suoi amici cloni. Ma, diciamoci la verità, stare solo in compagnia di se stessi, anche se con tante copie identiche, non è proprio il massimo.

Il Capitano Kube sorrise benevolo " So io che fare".

Appena pronunciò queste parole, apparve un tunnel tra il controllore della replicazione di Phippy e il resto della nave. Ridendo di cuore, il Capitano Kube disse: "Anche se i tuoi cloni dovessero andare e venire, questo tunnel rimarrà sempre aperto per te, così che tu possa scoprire altri pod e loro possano scoprire te!"

Servizi

- Devono essere persistenti
- Permettono la fase di discovery
- Permettono il bilanciamento (load balancing)
- Forniscono indirizzi di servizio stabili
- Permettono di trovare pod in base al selettore di etichette



Un servizio spiega al resto dell'ambiente Kubernetes (inclusi i pod e ReplicaSets) quali servizi ogni applicazione fornisca. Mentre i pod vanno e vengono (!), gli indirizzi IP dei servizi e le porte rimangono invariati. È così che altre applicazioni possono cercare e trovare i servizi esistenti attraverso il servizio discovery di Kubernetes.



Phippy iniziò così ad esplorare il resto della nave e non passò molto prima che incontrasse Goldie e che diventassero migliori amici. Un giorno Goldie fece qualcosa di straordinario: fece un regalo a Phippy. Quando lo vide, la piccola app si lasciò sfuggire la più triste delle lacrime.

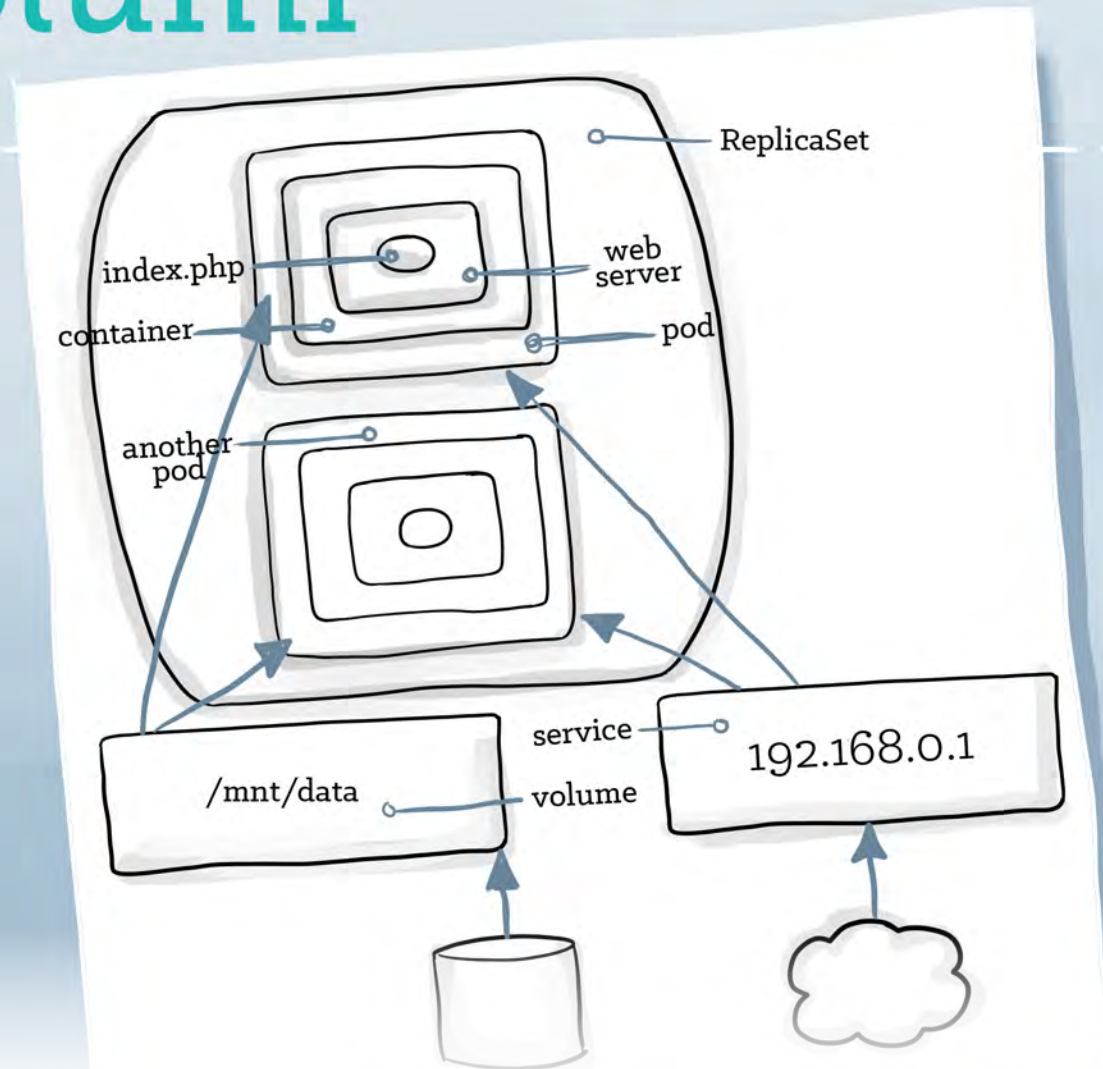
"Ma perché sei così triste?", chiese Goldie.

"Il tuo regalo è bellissimo, ma non so dove riporlo!", tirò su col naso Phippy.

Ma Goldie sapeva bene cosa fare "Perché non lo metti in un volume?"

Volumi

- I provider dispongono di spazi di archiviazione permanenti o effimeri (quali, tra gli altri: Cloud Block Storage, Ceph, Gluster ...)
- I pod possono montare volumi come filesystems



Un volume è uno spazio al quale i containers possono accedere per archiviare informazioni. Il volume appare come parte del filesystem locale. I volumi possono essere supportati da un'archiviazione locale, Ceph, Gluster, storage in cloud a blocchi o altri backend per lo storage.

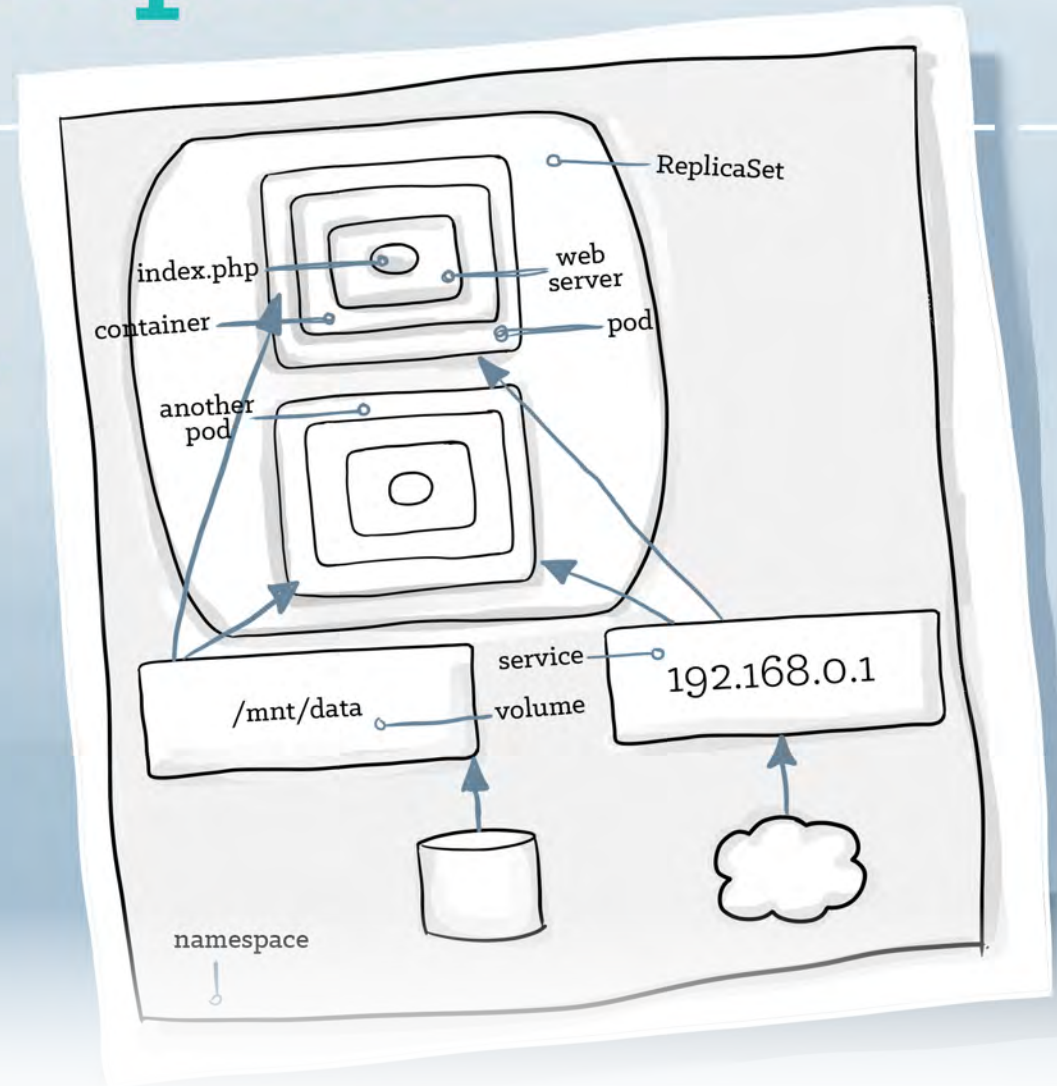


Phippy amava la vita a bordo della nave del Capitano Kube e la compagnia dei suoi nuovi amici (ogni copia del pod di Goldie era egualmente adorabile!). Ma ripensando ai tempi in cui si trovava ospitata su quel provider spaventoso, iniziò a sentire che forse avrebbe voluto di nuovo un po' di quella privacy, di tanto in tanto.

"Sembra tu abbia bisogno di un namespace", affermò il Capitano Kube.

Namespace

- Raggruppamento e segmentazione di pod, ReplicaSets, volumi e secrets



Un namespace è un meccanismo di raggruppamento all'interno di Kubernetes. Servizi, pod, ReplicaSets e volumi possono cooperare agilmente all'interno di un namespace e, d'altra parte, i namespace garantiscono un certo livello di isolamento dalle altre parti del cluster.



La vita a bordo della nave del Capitano Kube era davvero fantastica. Phippy solcò i mari e visse avventure grandiose insieme ai suoi nuovi amici, ma sopra ogni cosa si sentì finalmente a casa.

E fu così che Phippy visse per sempre felice e contenta.

