

Kubernetes jako jednotná platforma pro provoz aplikací



Ondřej Šika

ondrej@sika.io

@ondrejsika

Kontejnery v praxi, 22. 2. 2022

@ondrejsika ondrej@sika.io sika.io /in/ondrejsika



Ondřej Šika

Jsem DevOps lektor, architekt a konzultant z Prahy.

Navrhnou a implementuji Vám na míru DevOps architekturu od verzování v Gitu po provoz v Cloudu.

Dělám populární školení, kde své znalosti předávám tak, abyste si mohli vše udělat sami a bez zbytečných přešlapů a slepých cest.

@ondrejsika ondrej@sika.io sika.io /in/ondrejsika



Cíl

Jednoduchý způsob, jak provozovat aplikace



Současný stav (před Kubernetes)

- Každý team (nebo firma) to dělá po svém
- Aplikace jsou nasazované na virtuální stroje (manuálně, případně pomocí Ansible, ...)
- Nutný manuální capacity planning
- Různé servery pro různé aplikace (Java, PHP, ...)



Docker (bez Kubernetes)

Docker, stejně jako virtuální servery, odděluje jednotlivé aplikace. Dělá to pomocí kontejneru v rámci jednoho serveru (ať už fyzického, nebo virtuálního).

Nový přístup, který Docker přinesl je to, že na serveru, kde chceme aplikace v Docker kontejnerech provozovat, nepotřebujeme instalovat žádné závislosti. Vše je součástí Docker image.



Kubernetes

@ondrejsika ondrej@sika.io sika.io /in/ondrejsika



Co je Kubernetes?

Kubernetes is a portable, extensible, open-source platform for managing containerized workloads and services, that facilitates both declarative configuration and automation. –kubernetes.io

<https://kubernetes.io/docs/concepts/overview/what-is-kubernetes/>



Proč používat Kubernetes?

- Unifikace prostředí pro provoz aplikací
- De-facto standard provozu software
- Ovládání pomocí YAML souboru
- Deployment (deklarativní) požadovaného stavu
- Přístup ke clusteru místo k jednotlivým serverům
- Automatizace manuálních tasků
- Autoscaling
- Opensource, pod CNCF
- Velký ekosystém kolem Kubernetes
- Jednoduchý provoz v cloudu i on-premise



Unifikace prostředí

Kubernetes se ovládá pomocí YAML souboru a API. Díky tomu, že Kubernetes nás svazuje v tom, jak můžeme věci nasazovat, musíme všechny aplikace nasazovat podobným stylem a poskládané ze stejných komponent.

Už to není tak, že každý projekt byl nasazen na server jiným způsobem - vše se dělá stejně, což má za následek velmi jednoduchý onboarding do firmy nebo do projektu.



De-facto standard pro provoz aplikací

Kubernetes je dnes již velmi rozšířené - od startupu po enterprise.

Stalo se vlastně takovým standardem, jako provozovat kontejnerizované aplikace a aplikace vůbec (provoz bez kontejneru nedává smysl).



Ovládání pomocí YAML souboru

Kubernetes se ovládá primárně pomocí deklarativních YAML souboru.

Tyto soubory můžeme jednoduše verzovat v Gitu a spolupracovat nad nimi tak, jak jsme zvyklí z vývoje software - pull requesty, CI, ...



Deployment požadovaného stavu

Do Kubernetes nasazujeme vždy to, jak má vypadat požadovaný stav. Nezajímá nás, zda aplikace už běží (a deployment ji nezmění), neběží a budeme ji instalovat nebo pouze aktualizovat.

Kubernetes se podívá, co je potřeba nasadit nebo aktualizovat a to provede.

Tento přístup není plně deklarativní, ale jsou tooly, které nám s tím mohou pomoci.



Přístup ke clusteru jako celku

Pokud jsme používali Docker, měli jsme pořád servery, kde jsme pouštěli konkrétní aplikace. Pokud jsme chtěli něco pustit, museli jsme se rozhodnout, na jakém serveru to pustíme. Kontrolu, že server má dostatek CPU, RAM, ... jsme dělali manuálně.

Kubernetes za nás toto řeší. Chceme-li nasadit aplikaci s nějakými požadavky, Kubernetes najde vhodný node (server v clusteru) a na něj aplikaci nasadí.



Automatizace manuálních tasků

Kubernetes se nám stará o rutinní tasky, jako restart aplikací, pokud detekuje problém, který vyžaduje nebo vyřeší restart.

Aplikaci nám přesune na běžící node, pokud ten, kde aplikace běžela, spadl nebo je nedostupný.

Kubernetes nás zbaví zbytečných rutijních zásahů, které nemusí dělat člověk.



Autoscaling

Kubernetes má podporu automatickeho škálování, pokud na naší aplikaci směřuje větší traffic. Kubernetes dokáže spustit více instancí této aplikace, aby požadavky zvládla.

Pokud load opadne, aplikace se sama seškáluje směrem dolů.



Opensource, pod CNCF

Kubernetes vzniklo v Google, ale dnes jej zastřešuje CNCF - Cloud Native Computing Foundation, která je dceřiná organizace Linux Foundation.

Díky tomu, že je Kubernetes open source a vyvíjeno různými velkými hráči (Google, AWS, Microsoft, ...), tak si můžeme být jistí, že se dále bude rozvíjet, bude zdarma a nebudeme svázáni konkrétním poskytovatelem.



Velký ekosystém kolem Kubernetes

Díky popularitě Kubernetes se kolem něj vytvořila hromada nástrojů a pluginů, které řeší monitoring, správu logů, ...

Kubernetes jednoduše podporuje rozšiřitelnost pomocí pluginů, takže jich vznikla hromada pro storage, provoz aplikací jako Elastic, Postgres, ...



Jednoduchý provoz v cloudu i on-premise

Všechny velké cloudy (AWS, GCP, Azure) poskytují Kubernetes. I většina menších cloudů (jako DigitalOcean), nabízí managed Kubernetes.

O Kubernetes se stará poskytovatel cloudu, vy je jen používáte.

Pokud si chcete provozovat Kubernetes na on-premise, můžete například využít RKE (distribuce Kubernetes od Rancheru) nebo si je nainstalovat na přímo sami.



Provoz aplikací v Kubernetes

@ondrejsika ondrej@sika.io sika.io /in/ondrejsika



Provoz aplikací v Kubernetes

Do Kubernetes můžeme nasadit jakoukoli aplikaci. V Kubernetes jsme schopni provozovat i legacy aplikace, starší než samotné Kubernetes.

Pokud chceme aplikaci zmigrovat do Kubernetes, případně psát novou - už Kubernetes native, je vhodné zaměřit se na pár drobností, které nám provoz velmi ulehčí.



Stateless Aplikace

Ideálně chceme, aby aplikace provozované v Kubernetes byly stateless, to znamená, že nemají žádný lokálně uložený stav.

Pokud se jedná o webové servery, tak mají session uložené v DB, místo na file systemu.

Kubernetes dokáže provozovat i stateful aplikace - všechny databáze jsou stateful aplikace, ale pokud se snažíme psát naši aplikaci, snažíme se jí psát jako stateless.



Škálování

Stateless aplikace se výrazně lépe škálují.

Aplikace chceme škálovat tak, že zvedneme počet instancí. Díky bezstavosti aplikací můžeme pustit pouze další instance a Kubernetes bude samo distribuovat traffic na všechny instance a pomůže nám zvládnout vyšší load.

Kubernetes může škálovat automaticky - viz autoscaling.



Nepoužívat souborový systém

Pokud pracujeme v jakémkoliv distribuovaném prostředí, narážíme na to, že potřebujeme mít souborový systém přes více nodů (na kterých provozujeme naši aplikaci).

Kubernetes podporuje volumes, které jsou přístupné z více instancí aplikace, není to však ideální řešení.

Nejlepší způsob jak pracovat se soubory v Kubernetes je používat nějakou objektovou storage jako AWS S3 nebo Minio (S3 compatible API).



Probes

Kubernetes umí testovat aplikaci, zda funguje. Pokud ne, má několik způsobů jak problém vyřešit.

Kubernetes testuje aplikace pomocí probes, nejčastěji jsou to speciální HTTP end pointy. Podle toho, zda tyto proby odpovídají a odpovídají korektně, Kubernetes pozná stav v jakém se konkrétní instance nachází.

Kubernetes na základě failu těchto checků, může aplikaci restartovat nebo jen na nějaký čas vyřadit z loadbalancingu (pokud to stačí k vyřešení problému).



Logy

Docker a Kubernetes pracuje s logy jako se streamem eventů.

Aplikace by měla logovat na standardní výstup - ne do souborů, který potom Docker a Kubernetes sbírají a ukládají interně.

Logy pak typicky sbírá například ještě FileBeat a posílá je do ElasticSearchu pro dlouhodobé ukládání a zpracování.

Pokud naše aplikace umí logovat ve formátu JSON, ulehčí nám to parsování v Elasticu a zjednoduší následnou práci s logy.



Metriky

Pokud chceme vědět, co se v aplikaci děje, je dobré exponovat metriky pro Prometheus.

Prometheus není elementární součástí Kubernetes, ale je velmi používanou součástí Kubernetes stacku.

Prometheus metriky nám umožňují vidět, co se v aplikaci děje, kolik requestů selhalo nebo kolik paměti aplikace aktuálně alokuje.

Tyto informace se nám hodí pro monitoring, který Prometheus také řeší.



Závěr

@ondrejsika ondrej@sika.io sika.io /in/ondrejsika



Kubernetes

Pro mě je dnes Kubernetes jediná přijatelná možnost, jak provozovat aplikace a to jak v cloudu i on-premise.

Díky Kubernetes je celé prostředí unifikované a já se můžu věnovat více vývoji a nemusím řešit day-to-day provozní záležitosti.



Provoz aplikací v Kubernetes

- Stateless aplikace
- Nepoužívat `file.open` - místo toho využít S3
- Používat probes - ať Kubernetes ví, zda aplikace žije
- Logy - do `STDOUT` a `STDERR` místo log souborů, ideálně v JSONu
- Prometheus metriky pro monitoring a debugging



Díky za pozornost

@ondrejsika ondrej@sika.io sika.io /in/ondrejsika



Otázky?

@ondrejsika ondrej@sika.io sika.io /in/ondrejsika



Email

ondrej@sika.io

Twitter

@ondrejsika

LinkedIn

/in/ondrejsika

@ondrejsika ondrej@sika.io sika.io /in/ondrejsika

