WIR HABEN KUBERNETES ZUHAUSE KUBERNETES ON PREMISES

Über die Herausforderungen, die es ohne große Cloud-Provider zu meistern gilt.

WHOAMI

- Emma Heinle (sie/ihr)
- Im DevOps-Team bei makandra - wir haben draußen einen Stand.

KUBERNETES

Heute kein "was ist Kubernetes"-Vortrag! Aber kurz:

- Kubernetes ist ein System zum Verwalten eines Clusters.
- Betrieben werden dort hauptsächlich Container.
- Das ist alles per APIs automatisch konfigurier- und steuerbar, nicht mit der Maus.
- Ziele sind Ausfallsicherheit und Skalierung.
- Cluster bestehen meistens aus mehreren Nodes.

EINSTIEG

Unabhängigkeit von großen Cloud-Providern

- kein managed Service
- keine abstrahierte Infrastruktur
- alles selbst gebaut

WARUM MACHT MAN DAS SELBER?

- Verstehen statt Konsumieren
- Kontrolle über die ganze Kette
- Unabhängigkeit von der Cloud
- Realistische Spielwiese

HEUTE KEIN CODE

- keine langen YAML-Dateien
- Wir sprechen über Konzepte und Komponenten.
- Welche Bauteile brauchen wir - und warum?

WAS BRAUCHT MAN?

- Nicht unbedingt neue Hardware, aber zuhause ist Hardware cool.
- Im professionellen Einsatz sind VMs nochmal viel praktischer.

WAS BRAUCHE ICH DENN ALLES?

- Server bereitstellen
- Netzwerk bauen
- Netzwerktraffic von außen in den Cluster bekommen
- Storage bereitstellen
- Authentifizierung und Autorisierung
- Betrieb gewährleisten

NODES BEREITSTELLEN

IN DER CLOUD

- Neue Maschinen sind in Sekundenschnelle online.
- Nicht mehr nötige Maschinen verschwinden in Sekundenschnelle.
- Maschinen können auf der ganzen Welt verteilt werden.

ZUHAUSE

- Neue Maschinen brauchen viel Vorlaufzeit.
- Nicht unbegrenzt Platz und Strom für Hardware.
- Ich habe nur ein Zuhause

NETZWERK BAUEN

IN DER CLOUD

- Größtenteils virtualisiert und per API verwaltbar
- Eigenes VPC abgetrennt vom ganzen Rest, eigene Subnets, eigenes Routing
- Internet-Zugang stellt der Cloud-Provider, inklusive Redundanz und mit viel Bandbreite

ZUHAUSE

- Für gewöhnlich weniger professionelle Setups
- Kein von zuhause wichtigen Dingen isoliertes Setup
- Nur ein Internet-Zugang, keine Redundanz, begrenzte und geteilte Bandbreite

STORAGE

Im Cluster-Betrieb inklusive Shared Storage!

IN DER CLOUD

- Schnell und einfach bereitzustellen - EBS für Block Storage (schnell, RWO) und EFS (RWX)

ZUHAUSE

- Lokale Disks
- Disks müssen über das Netzwerk ansprechbar sein
- Redundanz muss selbst bereitgestellt werden

BACKUP

Wer Storage hat, braucht auch Backup!

IN DER CLOUD

- Leicht konfigurierbar
- Automatisch in geschützten Locations

ZUHAUSE

- Noch mehr Hardware notwendig
- Oft nur eine Location möglich

IDENTITY

Zentrale Verwaltung von Accounts und Rechten

IN DER CLOUD

- Identitätsverwaltung beliebig komplex
- Ich kann bestehende Accounts anbinden

ZUHAUSE

- Es gibt Software.
- Interoperabilität mit bestehenden Systemen nicht immer so gut
- Muss selbst installiert und implementiert werden.

ZUGRIFF FÜR ENDUSER

Irgendwie muss der Traffic von Nutzer*innen an die Anwendungen kommen, die im Cluster laufen.

IN DER CLOUD

- Cloud-Spezfische Load Balancer wie ALB oder NLB kümmern sich einfach.
- Sie haben eine oder mehrere IPs, einen DNS-Eintrag dafür.
- Diese Load Balancer verteilen das dann in den Cluster.

ZUHAUSE

- Mehrere Komponenten notwendig - z.B. eine virtuelle IP und Proxyserver, die Anfragen in den Cluster verteilen.

TLS-ZERTIFIKATE

Für `https://` - unerlässlich!

IN DER CLOUD

- Leicht bereitzustellen
- Lifecycle Management geht automatisch.

ZUHAUSE

- Zertifikate müssen selbst bereitgestellt und dem Proxyserver gegeben werden.
- Let's Encrypt funktioniert gut - muss aber authentifiziert werden. Das braucht auch Konfiguration.
- Lifecycle Management muss selbst automatisiert werden.

DNS

- Anwendungen brauchen ihre Zugriffs-URLs.
- Verifikation von Zertifikaten geht oft am besten über DNS.

IN DER CLOUD

- Automatische Dienste bei den Cloud-Anbietern

ZUHAUSE

- Via API verwaltbarer DNS-Anbieter muss bestellt werden

OBSERVABILITY

Monitoring, Logfiles zentralisiert und Metriken über Ressourcen und Fehler

CLOUD

- Cloudprovider haben Angebote dafür.
- Sehr detailliert konfigurierbar
- Meist sehr sehr teuer

ZUHAUSE

- Alles selber bauen
- Auch sehr detailliert konfigurierbar
- Braucht Hardware-Kapazitäten

PRAXIS

- Die großen Probleme sind nun grob umrissen.
- Wir brauchen Lösungen!

HARDWARE OHNE CLOUD

BEI EMMA ZUHAUSE: TURING PI

- Turing Pi mit 4x Raspberry Pi Compute Module 4
- Je 8 GB RAM, Quad Core ARM CPU
- Lokaler Speicher jeweils eine 32GB micro-SD

ALTERNATIVE

- Gebrauchte Office PCs mit amd64-CPU
- Ordentliche Server in einem ordentlichen Rechenzentrum

SOFTWARE OHNE CLOUD

Alles selbst zu installieren.

- Ubuntu Linux LTS
- Kubernetes-Distribution K3s
- 1 Control Plane Node, 3 Worker Nodes.

NETZWERK OHNE CLOUD

Server erreichen sich untereinander, Container auch.

- Der Turing Pi hat einen Switch eingebaut, Netzwerk ist relativ einfach.
- Jede Node hat eine IP aus dem Bereich meiner Fritz!Box.
- Die Nodes spannen untereinander ein internes Cluster-Netzwerk auf.
- Es gibt die Wahl zwischen verschiedenen Netzwerk-Plugins

NETZWERK OHNE CLOUD, MIT CILIUM

Cilium ist ein modernes Netzwerk-Plugin auf Basis von eBPF.

- Das interne Cluster-Netzwerk funktioniert dann damit.
- Cilium kann sogar virtuelle IPs bereitstellen!
- Cilium hat praktische Features zur Überwachung von Netzwerktraffic.
- Cilium kann `NetworkPolicies`, also interne Firewall-artige Regeln.

STORAGE OHNE CLOUD, MIT NFS

Am einfachsten ist NFS.

- NFS erlaubt Zugriff mehrerer Container gleichzeitig auf ein Verzeichnis.
- Ein schon vorhandenes NAS kann einfach genutzt werden.
- Im Cluster gibt es den NFS Subdir Provisioner, der für jedes "Laufwerk" ein Verzeichnis im NFS-Share anlegt und in den Container mounted.

STORAGE OHNE CLOUD, MIT LONGHORN

Longhorn macht ISCSI oder auch NFS.

- Longhorn macht Storage im Cluster selbst.
- Lokal an den jeweiligen VMs verfügbarer Speicher kann verwendet werden.
- Longhorn repliziert die Daten ggf. zwischen zwei oder mehr Nodes für Redundanz.
- Longhorn kümmert sich um die Bereitstellung der "Laufwerke" an die Nodes, auf denen der Container läuft.
- Bereitstellung kann über ISCSI direkt oder auch über NFS, für mehrere beteiligte Nodes sein.
- Backup, Snapshots, Webgui, Redundanz, Metriken, etc. kommen als gratis-Feature mit.

BACKUP OHNE CLOUD, MIT BORG

Erst Daten sammeln, dann wegsichern.

- Das NAS hat die Daten bereits zentral gesammelt.
- Longhorn's Backup-Feature kann die Daten auch in einen Object Storage oder an ein NFS schicken zum Sammeln.
- Ich nutze borgmatic um die Backups dann außer Haus zu spiegeln.

IDENTITÄT OHNE CLOUD, MIT AUTHENTIK

Authentik ist Nutzerinnenverwaltung, Gruppenverwaltung, kann OIDC und SAML und viel mehr.

- Ein zentrales Verzeichnis von Usern und Gruppen
- Eignet sich als Single-Sign-On - Lösung
- Verschiedene Anmeldungsregeln können konfiguriert werden.
- Bei Kubernetes selbst und bei Anwendungen ist die Anmeldung dann via Authentik möglich.
- OIDC oder SAML für Anwendungen, die das können
- Vorgeschalteter Proxy für Anwendungen, die das nicht können
- Das Verzeichnis lässt sich ggf. auch aus externen Quellen wie Active Directory, Google uvm. speisen.

ZUGRIFF FÜR ENDUSER OHNE CLOUD

Virtuelle IPs und ein Reverse Proxy

- Die Nodes im Cluster einigen sich auf eine virtuelle IP-Adresse und wählen untereinander einen Leader.
- Auf dem nimmt ein Reverse Proxy die `https`-Anfragen entgegen und verteilt sie an die Container mit den Anwendungen, die zur URL passen.
- Der Reverse Proxy hat eine Sonderstellung - er hat eine IP-Adresse von der Fritz!Box *und* kann die Services im Cluster erreichen.

ZUGRIFF FÜR ENDUSER OHNE CLOUD, MIT CILIUM

Cilium stellt virtuelle IPs mit Failover-Mechanismus *und* Reverse Proxy.

- Cilium kennt die Teilnehmenden Nodes im Cluster.
- Virtuelle IP ist immer an einer erreichbaren Node.
- Dort kann über die moderne GatewayAPI `https` traffic entegen genommen und verteilt werden.

TLS-ZERTIFIKATE

cert-manager hat das Problem schon lange gelöst.

- Let's Encrypt-Zertifikate können einfach bestellt werden.
- Cert Manager kümmert sich um Verlängerung, Einbindung in den Proxy-Server, etc.
- Verifikation der Domain geht über `http01`-Challenge oder `dns01`-Challenge intern.
- Zertifikate sind normale Kubernetes-Ressourcen und im Cluster verwaltbar.

DNS OHNE CLOUD, MIT CERT MANAGER UND EXTERNAL DNS

Manuell geht immer, aber automatisch ist besser.

- Einen DNS-Provider mit unterstützter API vorausgesetzt, kann Software in Kubernetes auch DNS verwalten.
- Cert Manager hat Plugins für verschiedene Provider, um bei denen `dns01`-Challenges zu lösen
- External DNS kann DNS-Einträge generell verwalten.

OBSERVABILITY OHNE CLOUD, MIT PROMETHEUS, LOKI UND GRAFANA

- Prometheus speichert Metriken.
- Grafana Loki speichert Logs.
- Grafana Alloy sammelt Metriken und Logs ein.
- Grafana zeigt Dashboards mit Metriken und Logs an.
- Alertmanager schickt Notifications.

OBSERVABILITY OHNE CLOUD, MIT DEM PROMETHEUS OPERATOR

Einfacheres Deployment mit dem Prometheus Operator

- Statt alle Anwendungen einzeln zu deployen und zu konfigurieren, macht das der Operator.
- Ich definiere Monitoring-Ziele als Kubernetes-Ressourcen, der Operator erkennt sie und setzt sie um.

WOFÜR DAS GANZE?

- Setup auf eigener Infrastruktur möglich
- Auf dem Turing Pi, auf dem Laptop in VMs oder im Rechenzentrum meiner Wahl
- Unabhängig von großen Cloudanbietern
- Daten-Souverenität bleibt erhalten - alles kann lokal bleiben - in Deutschland oder in meinem Schlafzimmer.

EINORDNUNG

- Mehr Kontrolle
- Mehr Aufwand
- Mehr Verständnis
- Mehr Vergnügen

DANKE

- Fragen?
- Trefft mich am Stand von makandra für Diskussionen