Highly available cross-region deployments with Kubernetes

Bastian Hofmann
Container orchestration platform
Deploy, run and scale your services in isolated containers
No vendor lock in
Standardized APIs
Runs on
Your laptop
Bare metal
Cloud Providers
And if you don't want to install and maintain Kubernetes yourself
Managed Kubernetes
You are viewing 33 cards with a total of 180 stars, market cap of $4.01T and funding of $1.19B.

### Platform - Certified Kubernetes - Hosted (33)

<table>
<thead>
<tr>
<th>Logo</th>
<th>Name</th>
<th>Status</th>
<th>Market Cap</th>
<th>Funding</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alibaba Cloud</td>
<td>Alibaba Cloud</td>
<td>Certified</td>
<td>$45B</td>
<td>$40M</td>
</tr>
<tr>
<td>Amazon EKS</td>
<td>Amazon EKS</td>
<td>Certified</td>
<td>$50B</td>
<td>$11M</td>
</tr>
<tr>
<td>Azure Container Service</td>
<td>Azure Container Service</td>
<td>Certified</td>
<td>$8B</td>
<td>$1B</td>
</tr>
<tr>
<td>Azure Kubernetes Service (AKS)</td>
<td>Azure Kubernetes Service (AKS)</td>
<td>Certified</td>
<td>$300M</td>
<td>$4B</td>
</tr>
<tr>
<td>Baidu Cloud Container</td>
<td>Baidu Cloud Container</td>
<td>Certified</td>
<td>$32B</td>
<td>$3M</td>
</tr>
<tr>
<td>BoCloud</td>
<td>BoCloud</td>
<td>Certified</td>
<td>$13B</td>
<td>$1B</td>
</tr>
<tr>
<td>Catalyst Kubernetes Services</td>
<td>Catalyst Kubernetes Services</td>
<td>Certified</td>
<td>$2B</td>
<td>$2B</td>
</tr>
<tr>
<td>Cisco Container Platforms</td>
<td>Cisco Container Platforms</td>
<td>Certified</td>
<td>$16B</td>
<td>$1B</td>
</tr>
<tr>
<td>eBaoCloud</td>
<td>eBaoCloud</td>
<td>Certified</td>
<td>$300M</td>
<td>$1M</td>
</tr>
<tr>
<td>eKong Technology</td>
<td>eKong Technology</td>
<td>Certified</td>
<td>$700M</td>
<td>$1M</td>
</tr>
<tr>
<td>ELASTX</td>
<td>ELASTX</td>
<td>Certified</td>
<td>$1B</td>
<td>$1B</td>
</tr>
<tr>
<td>Google Kubernetes Engine</td>
<td>Google Kubernetes Engine</td>
<td>Certified</td>
<td>$1B</td>
<td>$100M</td>
</tr>
<tr>
<td>Haproxy Cloud Container Platform</td>
<td>Haproxy Cloud Container Platform</td>
<td>Certified</td>
<td>$50M</td>
<td>$10M</td>
</tr>
<tr>
<td>IBM Cloud Kubernetes Service</td>
<td>IBM Cloud Kubernetes Service</td>
<td>Certified</td>
<td>$12B</td>
<td>$1B</td>
</tr>
<tr>
<td>Intellect SECC</td>
<td>Intellect SECC</td>
<td>Certified</td>
<td>$3B</td>
<td>$1B</td>
</tr>
<tr>
<td>Nirmata Managed Kubernetes</td>
<td>Nirmata Managed Kubernetes</td>
<td>Certified</td>
<td>$4B</td>
<td>$1B</td>
</tr>
<tr>
<td>Nutanix</td>
<td>Nutanix</td>
<td>Certified</td>
<td>$3B</td>
<td>$1B</td>
</tr>
<tr>
<td>Oracle</td>
<td>Oracle</td>
<td>Certified</td>
<td>$5B</td>
<td>$1B</td>
</tr>
<tr>
<td>OVH</td>
<td>OVH</td>
<td>Certified</td>
<td>$2B</td>
<td>$1M</td>
</tr>
<tr>
<td>RackSpace</td>
<td>RackSpace</td>
<td>Certified</td>
<td>$1B</td>
<td>$1M</td>
</tr>
<tr>
<td>Samsung SDS</td>
<td>Samsung SDS</td>
<td>Certified</td>
<td>$5B</td>
<td>$1M</td>
</tr>
<tr>
<td>SAP Certified</td>
<td>SAP Certified</td>
<td>Certified</td>
<td>$10B</td>
<td>$1M</td>
</tr>
</tbody>
</table>

Crunchbase data is used under license from Crunchbase to CNCF. For more information, please see the license info.
Self-healing built in
But what happens when a complete datacenter is not available
Multi-Region Kubernetes Setups
Availability
Scalability
More Points-of-Presence
Reduce dependencies on one single cloud provider
Because of the standardised API across providers, Kubernetes can help.
Features
Scheduling by Node Labels
failure-domain.beta.kubernetes.io/
region=dbl1

failure-domain.beta.kubernetes.io/
zone=dbl11
Regions vs availability zone
failure-domain.beta.kubernetes.io/
region=dbl1

failure-domain.beta.kubernetes.io/
zone=dbl11
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx
spec:
template:
spec:
  containers:
  - image: nginx
    name: nginx
nodeSelector:
  failure-domain.beta.kubernetes.io/region: dbl
Affinities
spec:
  affinity:
    podAntiAffinity:
      requiredDuringSchedulingIgnoredDuringExecution:
        - topologyKey: "failure-domain.beta.kubernetes.io/region"
          labelSelector:
            matchLabels:
              app: nginx
Service discovery with built in DNS
apiVersion: v1
kind: Service
metadata:
  name: nginx
spec:
  type: ClusterIP
  ports:
    - port: 80
      targetPort: 80
  selector:
    app: nginx
External LoadBalancers
apiVersion: v1
kind: Service
metadata:
  name: nginx
spec:
  type: LoadBalancer
  ports:
    - port: 80
      targetPort: 80
  selector:
    app: nginx
external-dns
StorageClasses
Some storage providers support dynamic volume provisioning
apiVersion: storage.k8s.io/v1
type: StorageClass
metadata:
    name: standard
provisioner: kubernetes.io/gce-pd
parameters:
    type: pd-standard
volumeBindingMode: WaitForFirstConsumer
allowedTopologies:
- matchLabelExpressions:
  - key: failure-domain.beta.kubernetes.io/zone
    values:
    - us-central1-a
    - us-central1-b
Challenges
Especially over different regions
External load balancing and traffic management
Internal load balancing
Connectivity between private networks
Kubernetes DNS across multiple clusters
Latencies
Synchronising Deployments across multiple clusters
Storage
Data replication
Split-brain problem
Possible setups
One cluster across AZs in one region
One cluster across regions
Multiple clusters connected via VPN
One cluster setup
One Kubernetes cluster across multiple availability zones
Easiest option
Low latencies
You can use one network
All pods and services can talk with each other
One cluster => one internal DNS
Service Discovery and internal load balancing works
External LoadBalancers work
You can mount storage across AZ
One cluster state => One deployment
But what if the whole region is down?
One Kubernetes cluster across multiple regions and VPN connection between networks
Latencies get higher by distance
You need a VPN to connect the networks
WireGuard as a VPN
https://github.com/squat/kilo
Kilo gets all Nodes from Kubernetes
Discovers regions and zones by Kubernetes Node labels
Configures WireGuard
Routing tables
All pods and services can talk with each other
One cluster => one internal DNS
Service Discovery and internal load balancing works
You need a solution for external load balancing
You have to replicate storage yourself
Not every Storage provider supports dynamic volume provisioning
One cluster state => One deployment
Multi cluster setup
Connecting multiple clusters with a VPN
WireGuard as a VPN
https://github.com/squat/kilo
All pods and services can talk with each other
Separate clusters => separate internal DNS
For service discovery configure each internal DNS to resolve to other clusters
cluster.region2:53 {
    forward . 10.10.11.10
}
.::53 {
    kubernetes cluster.local cluster.region1 in-addr.arpa ip6.arpa {
        pods insecure
        upstream
        fallthrough in-addr.arpa ip6.arpa
    }
    forward . /etc/resolv.conf
    loop
    loadbalance
}
cluster.region1:53 {
  forward . 10.10.10.10
}
.
53 {
  kubernetes cluster.local cluster.region2 in-addr.arpa
  ip6.arpa {
    pods insecure
    upstream
    fallthrough in-addr.arpa ip6.arpa
  }
  forward . /etc/resolv.conf
  loop
  loadbalance
}

Internal load balancing works
You have to replicate storage yourself
Every cluster has their own StorageClass that works on all nodes
Separate clusters have separate state
Management of deployments across clusters
Kubefed

https://github.com/kubernetes-sigs/kubefed
FederatedNamespaces,
FederatedDeployments,
FederatedConfigMaps,
FederatedServices,
...
Cluster aware controller that manages resources in all connected clusters
Cluster 1

- Kubernetes Master Components
- Kubefed Controller
- Nginx Service
- Nginx

Cluster 2

- Kubernetes Master Components
- Kubefed Controller
- Nginx Service
- Nginx
Demo
Solution for external traffic
Multi cluster external-dns
service.namespace.domain.svc.example.com
service.namespace.domain.svc.region1.example.com
service.namespace.domain.svc.region2.example.com
Demo
More options
Multiple clusters connected via Service Mesh (Istio)
Conclusion
Kubernetes makes it easier to create multi region setups
There are still challenges you have to overcome
Often just physics
Federation Tooling is just getting started
Test it
30 days
For free

Visit us at our booth
Rate today’s session

Cyberconflict: A new era of war, sabotage, and fear

David Sanger

Cyberconflict: A new era of constant sabotage, misinformation, and fear, in which everyone is a target, and you’re often after the weakest targets. A growing conflict among states. From everything from infrastructure to weapons to defense and deal, cyber is now the weapon of choice for democracies, dictator, and terrorists.

David Sanger explains how the rise of cyberweapons has transformed geopolitics like nothing since the invention of the atomic bomb. Moving from the White House Situation Room to the deserts of Chinese, Russia, North Korea, and Iran, he tells the stories of Silicon Valley, New York, and Chicago. He reveals the secrets of when the United States hacked Iran when it began using cyberweapons against Russian nuclear plants and North Korea’s missile launches. But now we’ve reached in a conflict we’re uncertain how to control, as our adversaries exploit vulnerabilities in our interconnected nation and we struggle to figure out how to deter these complex, short-war attacks.

David Sanger
The New York Times

O’Reilly Events App
https://github.com/bashofmann/kubernetes-multicluster-demos
Connecting multiple clusters with a Service Mesh Gateway
No VPN necessary
Fairly easy to set up
Pods from different clusters communicate over public IPs
Traffic encrypted and authenticated with mutual TLS
Communication is only possible through Istio proxies
Easy service discovery
Flexible, location aware traffic management
Telemetry and tracing included