Jurnal Teknik dan Science
JTS Vol 1 No. 3 Oktober 2022 | P-ISSN: 2828-7002 E-ISSN: 2828-6871, Page 54-58

IMPLEMENTASI MULTI DATABASE SEKMA PADA INFRASTUKTUR BERBASIS MICROSERVICE

Andre Pratama Adiwijaya
Fakultas Ekonomi, andre64@staff.gunadarma.ac.id, Universitas Gunadarma

ABSTRACT

Currently the development of infrastructure is quite fast which must support business performance in a company. Infrastructure technology itself has several licenses such as monolid and microservice. The microservice scheme by using OKD we can produce multiservice applications where there are problems in the service this service will not interfere with other services. Due to the process within the microsoft itself stands alone, to integrate services one by one can use config files from the microservice side. In OKD itself there is already a tool to monitor each service such as monitoring logs, YAML configuration and others. In this research, the application was monitored actively and successfully up to the time. If there is a problem with the service, OKD will provide a warning on the dashboard.

Keywords: OKD, microservice, linux. centos 7, infrastructure

ABSTRAK

Saat ini perkembangan infrastrukur teknologi cukup pesat yang dimana harus mensupport kinerja bisnis pada suatu perusahaan. Infrastrukur teknologi sendiri memilik beberapa skema seperti monolid dan microservice. Skema microservice dengan menggunakan OKD kita dapat menghasilkan applikasi multiservice yang dimana apabila ada kondisi trouble di suatu service makan service tersebut tidak akan menggangu service yang lain. Dikarenakan prosesnya didalam microsercive sendiri bediri sendiri, untuk integrasi service satu dengan satu bisa menggunakan konfig file dari sisi microservicenya. Didalam OKD sendiri sudah ada tools untuk monitoring dari setiap servicennya seperti log monitoring, YAML config dan lain – lain. Didalam peneletian ini menghasilkan apps yang termonitor aktif dan terjaga uptimenya. Apabila ada ada kedala pada service OKD akan memberikan alert pada dashboardnya.

Kata Kunci: OKD, microservice, linux. centos 7, infrastuktur

1. PENDAHULUAN

Saat ini topologi infrasruktur pada perusahaan start-up masih didominasi dengan skema topologi monolide. Dimana topologi monolide mengakibatkan beban pada suatu server atau virtual machine cukup besar. Beban tersebut terdiri dari load CPU, memory, dan Hardisk. Akibat beban tersebut membuat kinerja aplikasi tidak optimal serta akan tumpang tindihnya satu aplikasi dan aplikasi pada 1 virtual machine. Adapun penelitian terkait pattern dari monolithic arsiterkur yang membuat seluruh transaksi data dari virtual machine tersebut menumpuk jadi [1]. Saat ini muncul teknologi baru yaitu docker yang membuat satu atau beberapa aplikasi bisa dimasukan kedalam satu container, yang dimana aplikasi tersebut masih dapat dilakukan perubahan walaupun sudah di bentuk kedalam satu container [2]. Dasar segi teknologi container juga muncul teknologi bari selain docker yaitu kubernates yang dipopulerkan oleh google. Kubernetes merupakan platform open-source yang digunakan untuk melakukan manajemen workloads aplikasi yang dikontainerisasi, serta menyediakan konfigurasi dan otomatisasi secara deklaratif. Kubernetes berada di dalam ekosistem yang besar dan berkembang cepat. Service, support, dan perkakas Kubernetes tersedia secara meluas.

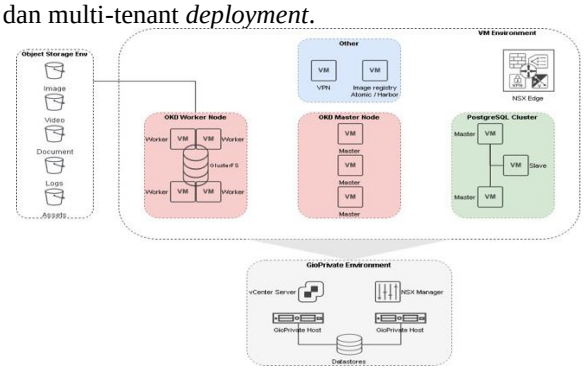
Google membuka Kubernetes sebagai proyek open source pada tahun 2014. Kubernetes dibangun berdasarkan pengalaman Google selama satu setengah dekade dalam menjalankan workloads bersamaan dengan kontribusi berupa ide-ide terbaik yang diberikan oleh komunitas [3]. Pada implementasi kurbenates yang masih hanya muncul dengan tampilan CLI atau tidak memilik interface yang membuat para devops agak

susah melakukan trace data apabila ada satu bagian dari applikasi yang terindkasi error. Pada saat ini muncul arsitketur Microservice yang membuat kinerja aplikasi terbagi sesusai dengan service didalamnya.

Arsitketur microservice adalah pendekatan untuk pengembangan aplikasi di mana aplikasi besar dibangun sebagai rangkaian layanan modular (mis., Modul / komponen yang digabungkan secara longgar). Setiap modul mendukung tujuan bisnis tertentu dan menggunakan antarmuka yang sederhana dan terdefinisi dengan baik untuk berkomunikasi dengan set layanan lainnya. [4]. Arsitektur microservice akan menjadi powerfull apabila didukung dari dengan aplikasi Openshift yang dimana aplikasi tersebut sudah memiliki antar muka yang cukup baik. Membuat devops atau pun user dapat melakukan pemantuaan terhadap service yang ada didalamnya. Didalam penelitian kita dapat memonitor dan deployment suatu service dengan waktu yang lebih effisien dan pembagaian beban terhadap server lebih proporsional.

2. METODOLOGI PENELITIAN

Perencaaan dalam penelitian ini menggunakan beberapa *tools* dan skema seperti Private VM (*virtual machine)*, HAProxy, PostgreSQL dan OKD (Redhat Openshift) keseluruhan skema tersebut dituangkan ke suatu topologi microservice pada Gambar 1. Virual Machine digunakan sebagai *host* (server) lalu HAProxy digunakan sebagai fasilitas untuk mengatur *supply* jaringan terhadap semua *service* yang ada topologi *microservice* tesebut. OKD digunakan pendistribusian *Kubernetes* yang dioptimisasi untuk pengembangan aplikasi secara terus menerus dan multi-tenant *deployment*.



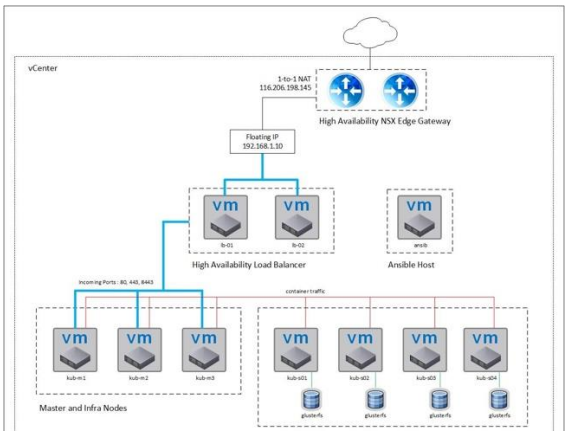
Gambar 1. Diagram Topologi Microservice

Penelitian ini memaksimalkan fungsi OKD yang dimana sebagai controller untuk seluruh pekerjaan yang ada didalam microservice.

3. HASIL DAN PEMBAHASAN

3.1. Pembuatan Skema OKD.

Pada penelitian ini skema OKD dibuat menjadi OKD master node dan OKD worker node yang dimana seluruh skema dibawah OKD tersebut diatur oleh dua load blancer yang mengatur load dari keseluruhan microservice dan satu virtual machine sebagai Ansible host yang dimana berfungsi sebagai instruksi instalasi aplikasi dibawahnya. Pada gambar 2 skema dari OKD diagaram.

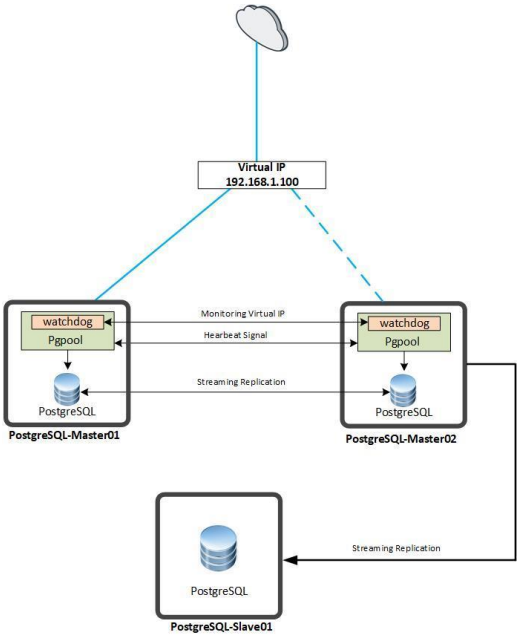


Gambar 2. Diagram OKD

Pembagian pada Gambar 2 menunjukan bahwa pada penelitian ini menggunakan tiga virtual machine untuk master dan empat virtual machine untuk *compute engine*. Yang dimana ke empat machine diatas memiliki log aggregator masing – masing.

3.2. Database Skema

Didalam penelitian ini database yang digunakan adalah postgre SQL yang dimana skema database dibagi menjadi dua skema seperti pada Gambar 3. Fungsi skema database tersebut yaitu dua sebagai *master* dan satu sebagai *slave*



Gambar 3. Diagram Database

3.3. Virtual Machine yang digunakan

Total virtual machine yang digunakan adalah 79 unit dengan pembagaian sebagai *Database*, *Kube – Pool dan Management & Network* pada Table 1 dijelaskan detail penggunakan seluruh virtual machine.

Kebutuhan	Virtual Machine	OS	vCPU	RAM	Basic Storage (GB)	mount
Databases	postgresql-master01	Centos 7	8	16	80	/
					200	/var/lib/pgsql
	postgresql-master02	Centos 7	8	16	80	/
					200	/var/lib/pgsql
	postgresql-slave01	Centos 7	8	16	80	/
					200	/var/lib/pgsql
Kube-Pool	kube-lb01	Centos 7	2	4	60	/
	kube-lb02	Centos 7	2	4	60	/
	kube-master01	Centos 7	4	18	60	/
	kube-master02	Centos 7	4	18	60	/
	kube-master03	Centos 7	4	18	60	/
	kube-node01	Centos 7	8	24	60	/
					500	/dev/sdb
	kube-node02	Centos 7	8	24	60	/
					500	/dev/sdb
	kube-node03	Centos 7	8	24	60	/
					500	/dev/sdb
	kube-node04	Centos 7	8	24	60	/
					500	/dev/sdb
	nfs-es	Centos 7	2	2	50	/
					500	/mnt/nfs
Management & Network	ansible-host	Centos 7	1	2	60	/
	nsx-edge-0-0	Centos 7	1	0,5	1	/
	nsx-edge-0-1	Centos 7	1	0,5	1	/
	vpn-server	Centos 7	2	4	60	/
TOTAL			79	215	4052	

Table 1. Berisikan Spesifikasi Virtual Machine yang diguanakan

Instalasi OKD Multi Master dan Multi Node menggunakan ansible

Berikut code instalasi OKD untuk penelitian ini menggunakan ansible :

[OSEv3:children]

- masters
- nodes
- etcd
- lb
- glusterfs

[masters]

- kub-m1.apps-example.com
- kub-m2.apps-example.com
- kub-m3.apps-example.com

[etcd]

- kub-m1.apps-example.com
- kub-m2.apps-example.com
- kub-m3.apps-example.com

[lb]

- kub-lb1.apps-example.com
- [nodes]
- kub-m1. apps-example.com openshift_schedulable=true openshift_node_group_name='node-config-compute'config-master-infra'
- kub-m2. apps-example.com openshift_node_group_name='node-config-compute'config-master-infra'
- kub-m3. apps-example.com openshift_node_group_name='node-config compute'config-master-infra'
- kub-s01. apps-example.com openshift_schedulable=true openshift_node_group_name='node-config-compute'
- kub-s02. apps-example.com openshift_schedulable=true openshift_node_group_name='node-config-compute'
- kub-s03. apps-example.com openshift_schedulable=true openshift_node_group_name='node-config-compute'
- kub-s04. apps-example.com openshift_schedulable=true openshift_node_group_name='node-config-compute'

[glusterfs]

- kub-s01.apps-example.com glusterfs_devices='["/dev/sdb"]'
- kub-s02.apps-example.com glusterfs_devices='["/dev/sdb"]'
- kub-s03.apps-example.com glusterfs_devices='["/dev/sdb"]'
- kub-s04.apps-example.com glusterfs_devices='["/dev/sdb"]'

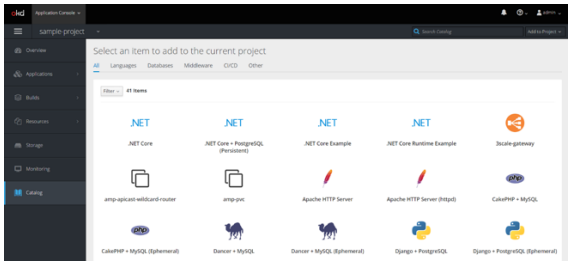
[OSEv3:vars]

- ansible_ssh_user=root
- enable_excluders=False
- enable_docker_excluder=False
- ansible_service_broker_install=False
- os_firewall_use_firewalld=True
- containerized=True
- os_sdn_network_plugin_name='redhat/openshift-ovs-multitenant' openshift_disable_check=disk_availability,docker_storage,memory_availability,doc ker_image_availability
- deployment_type=origin openshift_deployment_type=origin
- template_service_broker_selector={"region":"infra"} openshift_metrics_image_version="v3.11" openshift_logging_image_version="v3.11" openshift_logging_elasticsearch_proxy_image_version="v1.0.0" openshift_logging_es_nodeselector={"node-role.kubernetes.io/infra":"true"} logging_elasticsearch_rollout_override=false
- osm_use_cockpit=true
- openshift_metrics_install_metrics=True openshift_logging_install_logging=True
- openshift_master_identity_providers=[{'name': 'htpasswd_auth', 'login': 'true', 'challenge': 'true', 'kind': 'HTPasswdPasswordIdentityProvider'}] openshift_master_htpasswd_file='/etc/origin/master/htpasswd'
- openshift_public_hostname=kub-console.apps-example.com openshift_master_default_subdomain=api.apps-example.com
- openshift_storage_glusterfs_namespace=app-storage openshift_storage_glusterfs_storageclass=true openshift_storage_glusterfs_storageclass_default=false

```
openshift_storage_glusterfs_block_deploy=true
openshift_storage_glusterfs_block_host_vol_size=100
openshift_storage_glusterfs_block_storageclass=true
openshift_storage_glusterfs_block_storageclass_default=false
```

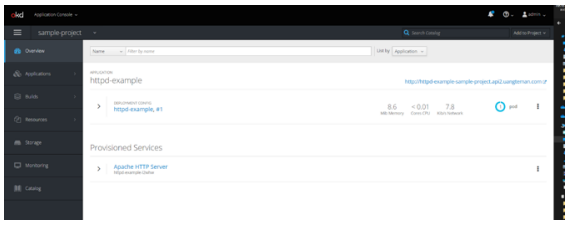
3.4. Hasil dari OKD yang sudah diimplementsasi pada microservice

Hasil dari OKD yang sudah terinstall pada penelitian terlihat pada Gambar 4 yang dimana tampilan sudah memilik tatap muka / *interface* dengan bentuk *web base*. User atau devops dapat langsung mengakses url OKD tersebut.



Gambar 4 Tampilan OKD

Pada tampilan di Gambar 4 terlihat user dapat membuat project dengan beberapa platform yang sudah disediakan oleh OKD. Cukup dengan klik projectnya container akan langsung terbentuk.



Gambar 5 Contoh project

4. KESIMPULAN DAN SARAN

Hasil perancangan dan implementasi OKD pada microsevice sudah berhasil dapat mengurangi dipendensi satu virtual machine yang tertumpuk dengan beberapa service atau aplikasi. Pada penelitian seluruh service dapat dipantau lewat Dashboard OKD dan apabila ada perubahan akan langsung dapat diganti pada config PODnya. Sedangkan apabila ada kegagalan pada servicenya atau aplikasi kita dapat kill/ restart service tersebut dan akan kembali ke versi awal dari service tersebut. Didalam dashboard OKD juga dapat memalukan pembuatan platform baru yang sudah disediakan seperti PHP,Go dan Rubby On Rails.

Saran dari penelitian user dapat memilih teknologi opensource untuk membangun suatu microservice berbasis container dan mempermudah saat eksekusi. Dalam skema microservice semua config dan diubah dengan lebih mudah dikarena sudah dapat di eksekusi langsung pada tampilan OKD tidak perlu membuka CLI (tampilan terminal.

DAFTAR PUSTAKA

[1]. Kong, “Pattern : Monotolithic Architecture” , 2019. [Online] Available: https://microservices.io/patterns/monolithic.html [Accessed: 29 April 2020].

[2]. Gilang Pambudi, “Development Environment dengan Docker Compose” , 2017. Available:https://medium.com/skyshidigital/development-environment-dengan-docker-compose-4f567eb4b538 [Accessed: 29 April 2020].

[3]. Kubernates Community “Apa itu Kubernetes? “October 10, 2019 [Online]. Available:. https://kubernetes.io/id/docs/concepts/overview/what-is-kubernetes/[Accessed: 29 April, 2020].

[4]. Siraj ul Haq “ Introduction to Monolithic Architecture and MicroServices Architecture” May 2, 2018 [Online]. Available:. https://medium.com/koderlabs/introduction-to-monolithic-architecture-and-microservices-architecture-b211a5955c63 [Accessed: 29 April, 2020].