



WORDPRESS



GALERA CLUSTER



Apache ZooKeeper™



K8S Storage Built-In Manage Storage Cluster for real- world

By Praparn Luengphoonlap
Email: praparn@opcellent.com

K8S Storage Built-In

OpCellent

Agenda

- Fantastic disk and where to find them
- Kubernetes for all (All-in-one solution)
- Case study: Push message processing
 - Solution design
 - Enhancement
 - Future improve
- Demo Case: Wordpress on scale
- Q&A

Who are we ? (Opcellent)

The image features the Opcellent logo at the top left, consisting of a blue stylized 'O' with three horizontal lines extending from its right side. To the right of the logo is the word "Opcellent" in a bold, black, sans-serif font. A thin horizontal line runs across the top of the page. Below the logo is a large, semi-transparent teal graphic element shaped like a comet's tail, pointing downwards. In the center of this teal shape is a smaller, solid teal circle containing a white stylized 'O'. Below this graphic, the text "Modern Server Technology Implementer" is displayed in a white, sans-serif font. At the bottom of the teal shape is a white rectangular button with the text "GET DETAILS" in a black, sans-serif font.

● ABOUT

Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s.

- [Domain ad. Setup SSOU Server](#)
- [MA SSOU Server](#)
- [Corporate Training](#)
- [Public Training](#)
- [SaaS-based Cloud](#)
- [Consult utilize SSOU Cloud u/w AWS like Google](#)
- [SaaS-based Hardware server SSOU](#)



● SERVICE

Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s.

 **docker**

Lorem Ipsum is simply dummy text of the printing and typesetting industry.

 **kubernetes**

Lorem Ipsum is simply dummy text of the printing and typesetting industry.

Lorem Ipsum is simply dummy text of the printing and typesetting industry.

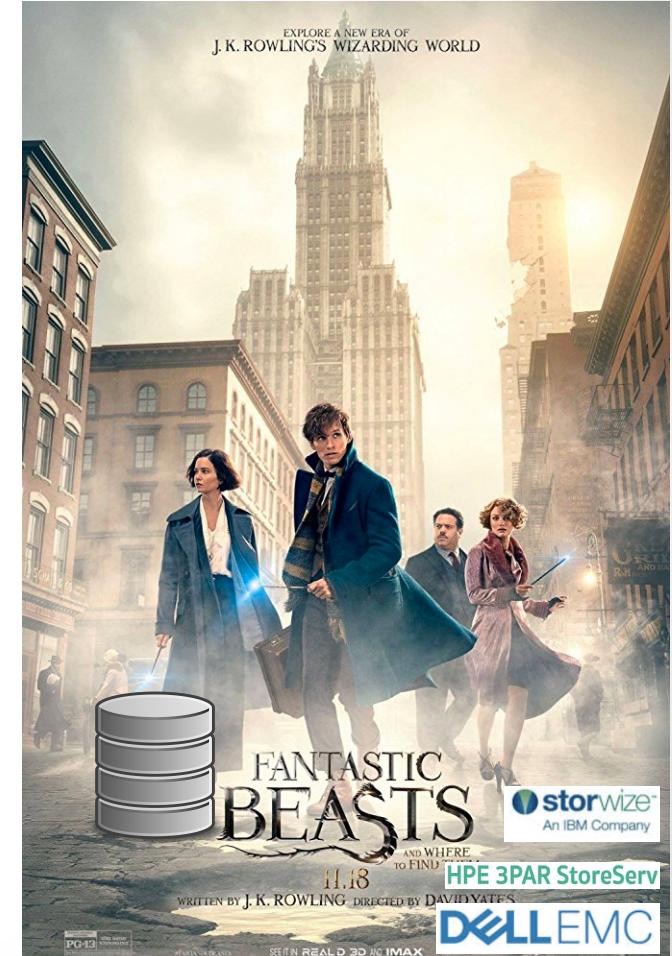


Present by: Praparn L. (eva10409@gmail.com)



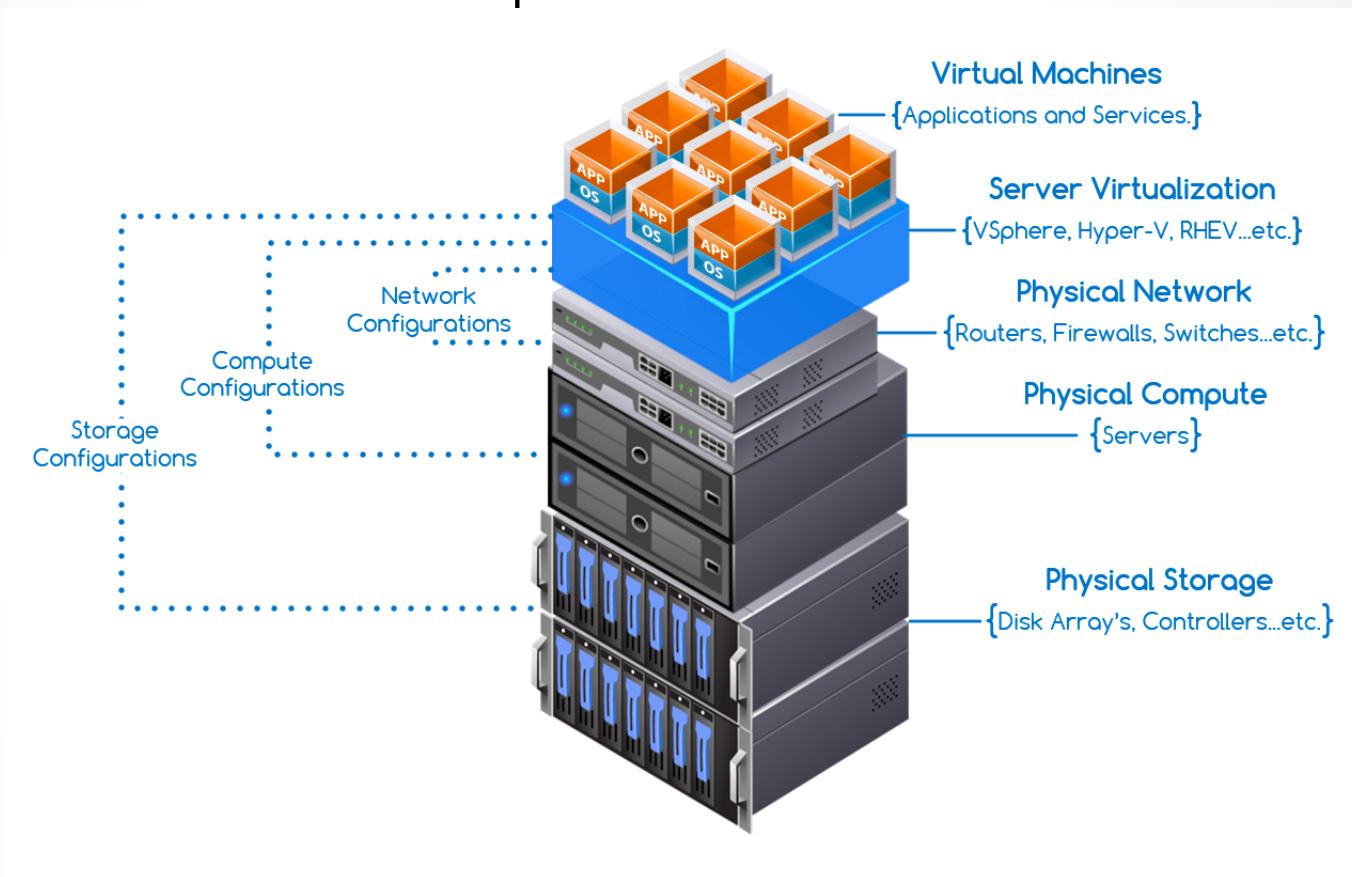
Fantastic disk and where to find them

- Find disk (Storage) for enterprise is like find monster
- Storage consist many in product
 - IBM V9000, V7000, V5000
 - HP 3PAR Storeserve
 - Dell EMC
 - NFS Storage
 - etc
- Each product need special skill
- Discussion ?
 - How many disk space ?
 - Is it need redundancy disk ?
 - What the characteristic of storage ? (ssd/sas)
 - Fiber disk ? How about network zoning ?
 - Ahhhh!!!



Fantastic disk and where to find them

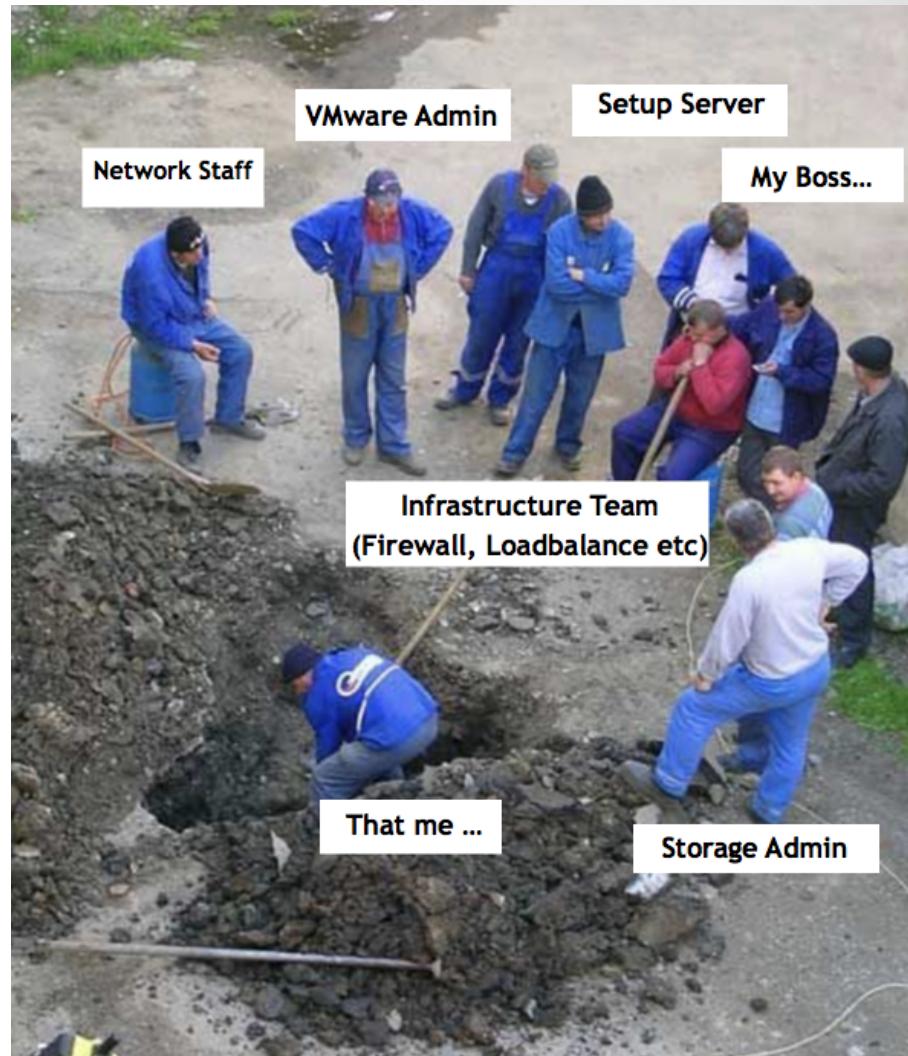
- How about other component ?



Fantastic disk and where to find them

Imaginary that ...

- How many staff need to join?
- How many skill be need ?
- How about complexity be here
- ...
- ...
- Is it necessary to like there ?



Kubernetes for All



K8S Storage Build-In

 Opcellent

Kubernetes for All

- Kubernetes is orchestrator for control container running in multiple server like cluster system
- Kubernetes will short name with “K8S”
- K8S will provide all component in single platform

VMWare / Physical Server	Kubernetes
Application/Database install	Pods (Set of container)
Server / Guest OS	No need
Networking	SDN (Calico, Flennel etc)
Load Balance	Service (native,Istio,kong,traefik etc)
Firewall	Service (native,Istio,kong,traefik etc)
Storage	All you need ☺

Kubernetes for All

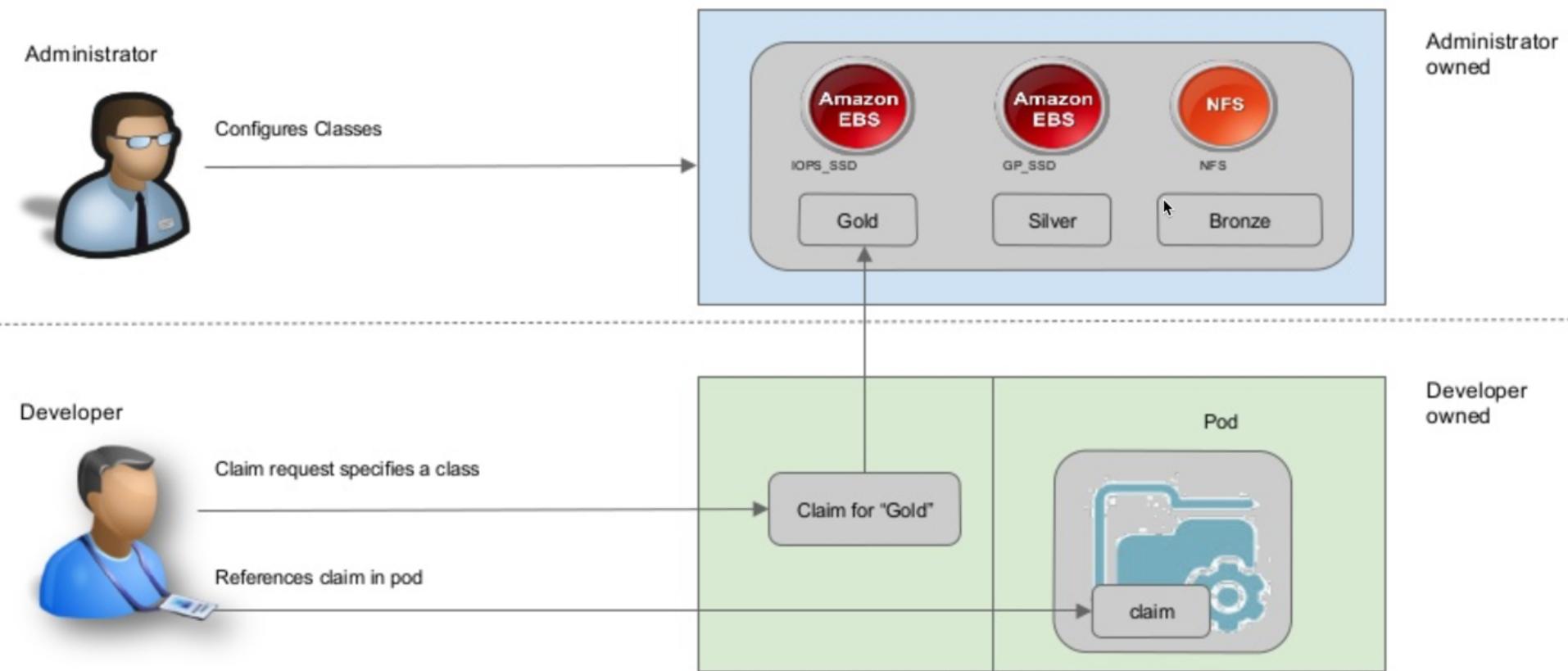
- Kubernetes support storage in many solution upon our request
 - Dynamic Provision by “StorageClass”
 - Create storage real-time when application request

Volume Plugin	Internal Provisioner	Config Example
AWElasticBlockStore	✓	AWS
AzureFile	✓	Azure File
AzureDisk	✓	Azure Disk
CephFS	-	-
Cinder	✓	OpenStack Cinder
FC	-	-
FlexVolume	-	-
Flocker	✓	-
GCEPersistentDisk	✓	GCE
Glusterfs	✓	Glusterfs
iSCSI	-	-
Quobyte	✓	Quobyte
NFS	-	-
RBD	✓	Ceph RBD
VsphereVolume	✓	vSphere
PortworxVolume	✓	Portworx Volume
ScaleIO	✓	ScaleIO
StorageOS	✓	StorageOS
Local	-	Local

Ref: <https://kubernetes.io/docs/concepts/storage/storage-classes/>

Kubernetes for All

- Dynamic Provision

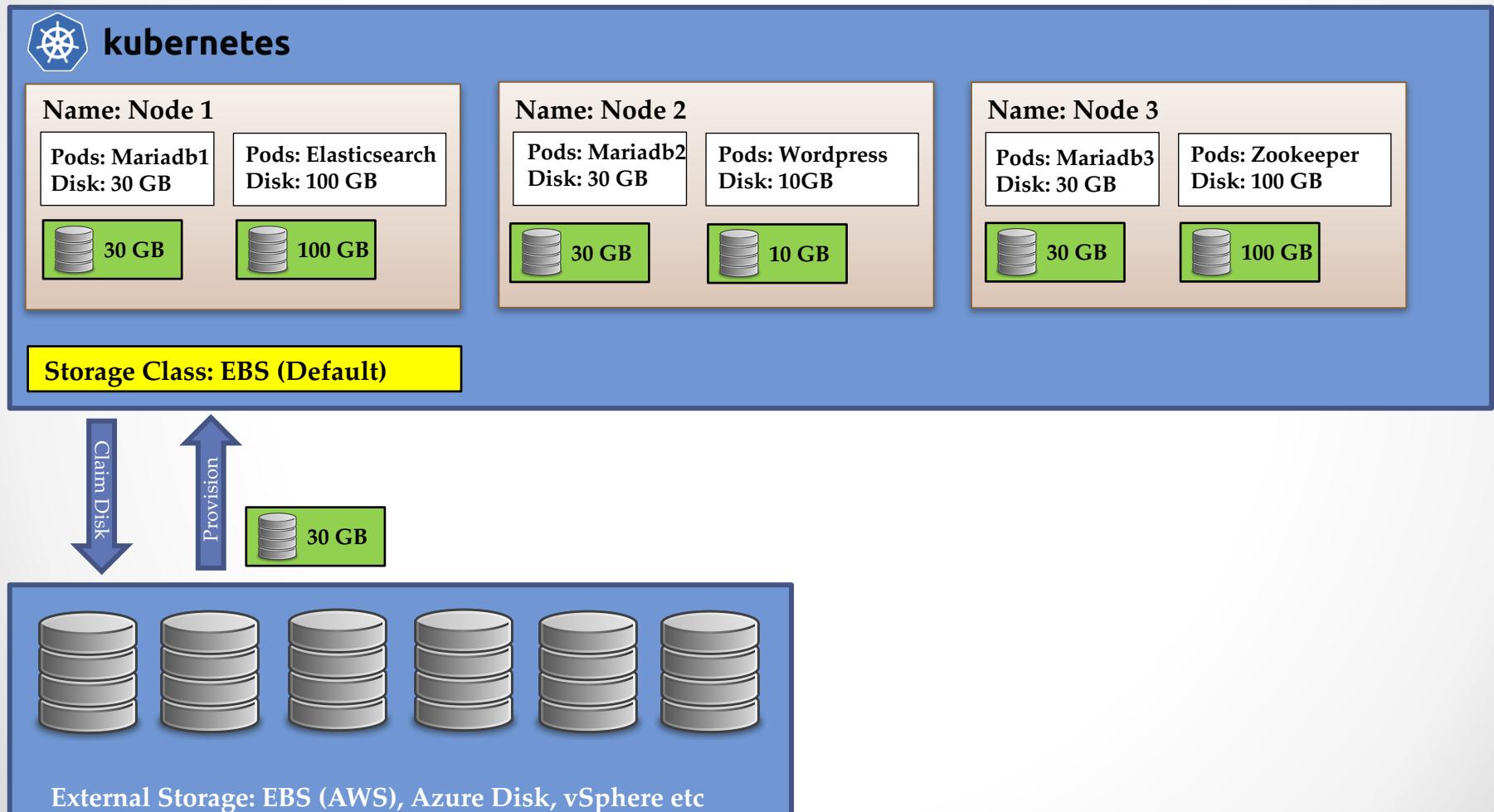


Ref: <https://kubernetes.io/docs/concepts/storage/storage-classes/>

K8S Storage Built-In

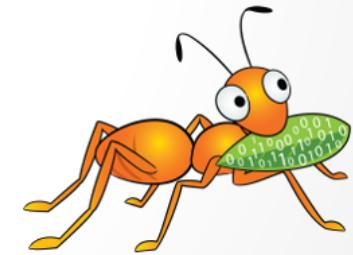
Kubernetes for All

- External Storage: Kubernetes will request storage from outside

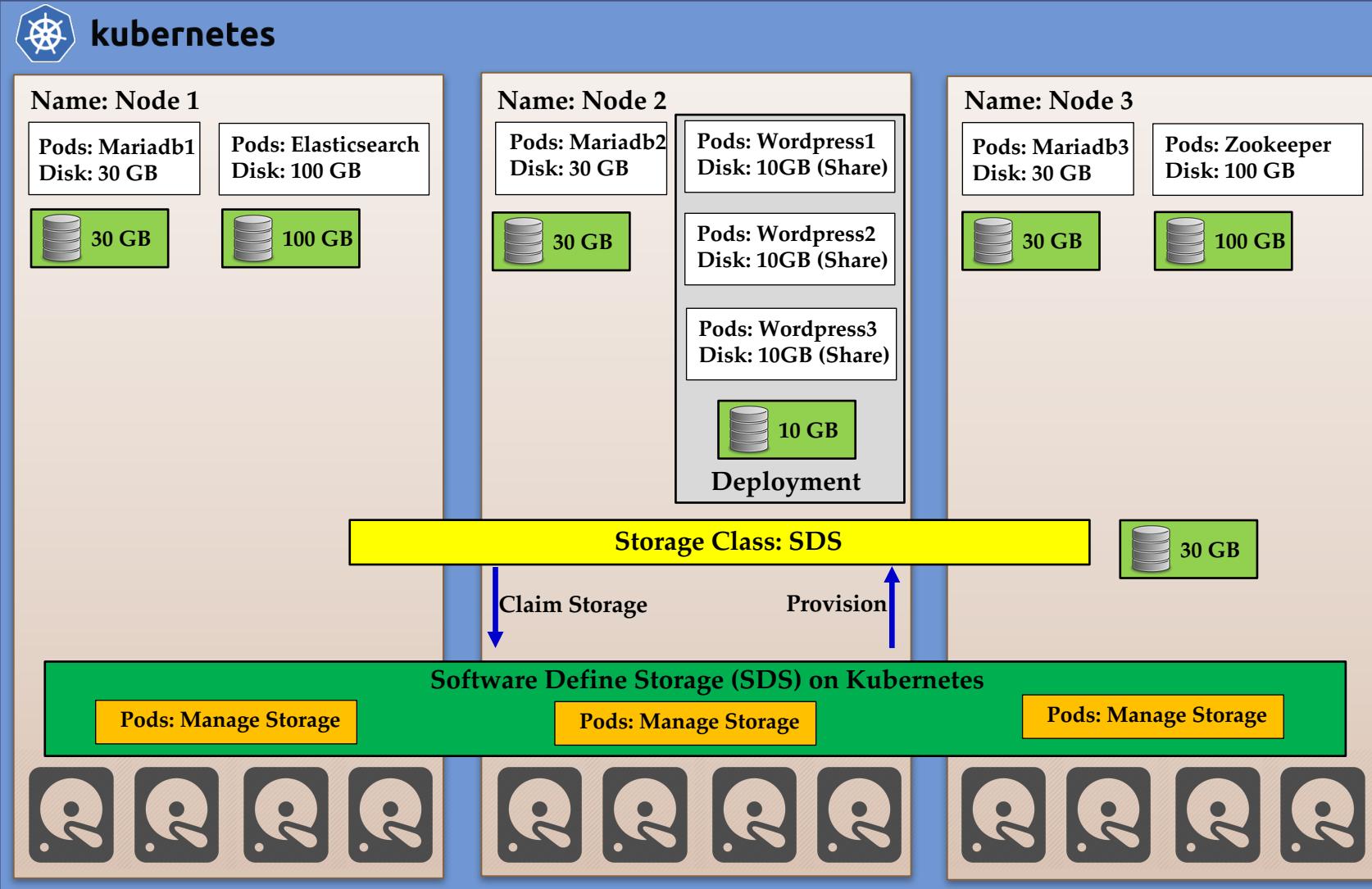


Kubernetes for All

- Storage Built-In: Kubernetes will manage disk with SDS (software define storage)
- Manage end-to-end with single platform
 - Manage raw disk to be storage system
 - Provision storage on application demand
 - Manage storage network to application (End-Point)
 - No overhead network outside kubernetes cluster
 - Total open-source solution



Kubernetes for All



Case Study: Push Message Processing



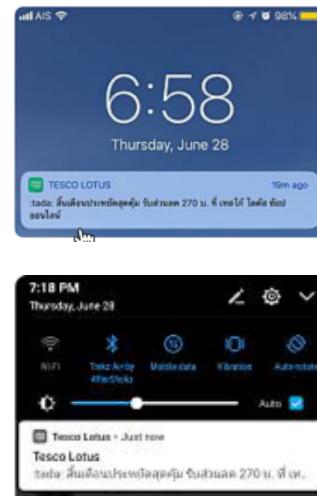
K8S Storage Build-In

 Opcellent

The logo consists of a stylized blue icon resembling a flame or a series of dots arranged in a circular pattern, followed by the word "Opcellent" in a bold, sans-serif font.

Push Messaging Processing

- Business Purpose
 - Operate push message notification to all mobile's customer
 - Message is about 50,000 – 1M message each round
 - Push notification need to keep logging for cross-check message
 - System need to handling all message



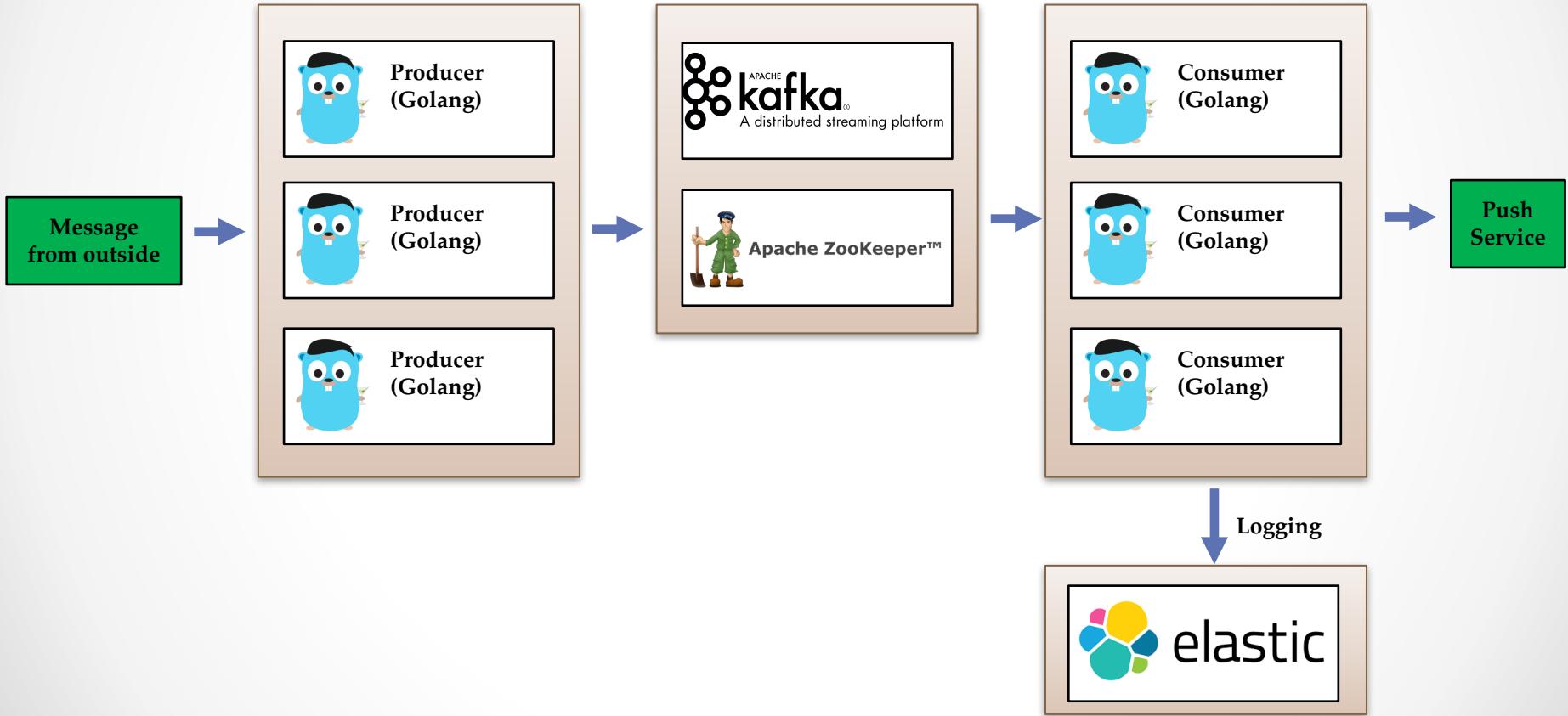
Push Messaging Processing

- Solution Design
 - We're select to handling message queue via "Apache Kafka" and keep data on zookeeper
 - Producer/Consumer was developed base on "golang"
 - Output will send to external push notification service
 - Logging will keep to elasticsearch
 - Solution was design base on "CNCF Landscape" for operate this system anywhere (On-prem/On-Cloud)



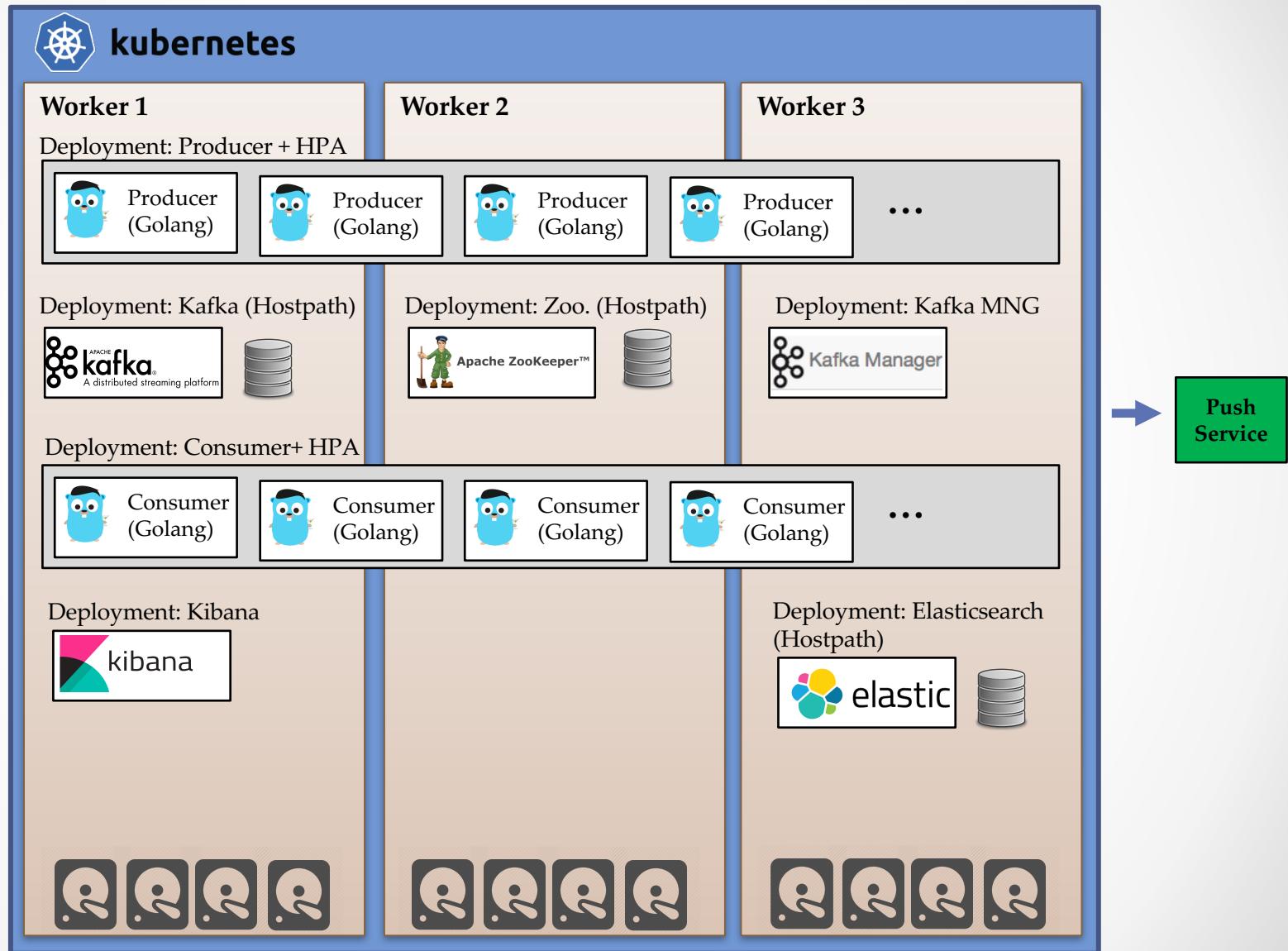
Push Messaging Processing

- Solution Design



Push Messaging Processing

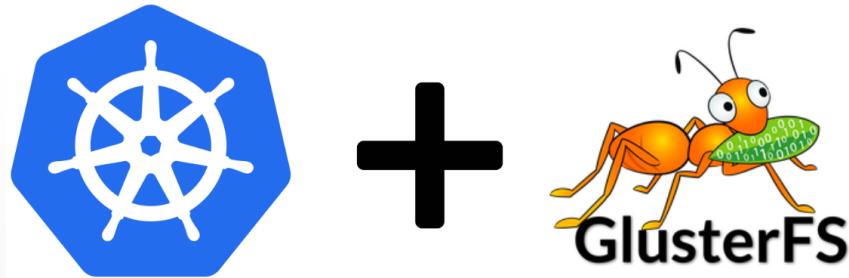
Version 1.0



K8S Storage Built-In

Push Messaging Processing

- Enhancement
 - Change “Hostpath” to “StorageClass” for dynamic provision storage.
 - For avoid dependency component. We will provide storage by kubernetes itself.
 - We choose “GlusterFS” to provide storage system from 3 worker node with standard
 - All GlusterFS need “heketi” for provide RESTful volume management interface



Push Messaging Processing

- Enhancement
 - <https://github.com/gluster/gluster-kubernetes>
 - GlusterFS official project “gluster-kubernetes”
 - Component
 - Kubernetes Farm (With RAW disk)
 - GlusterFS (Pods)
 - Heketi (Pods)
 - Control disk layout by define “topology.json”

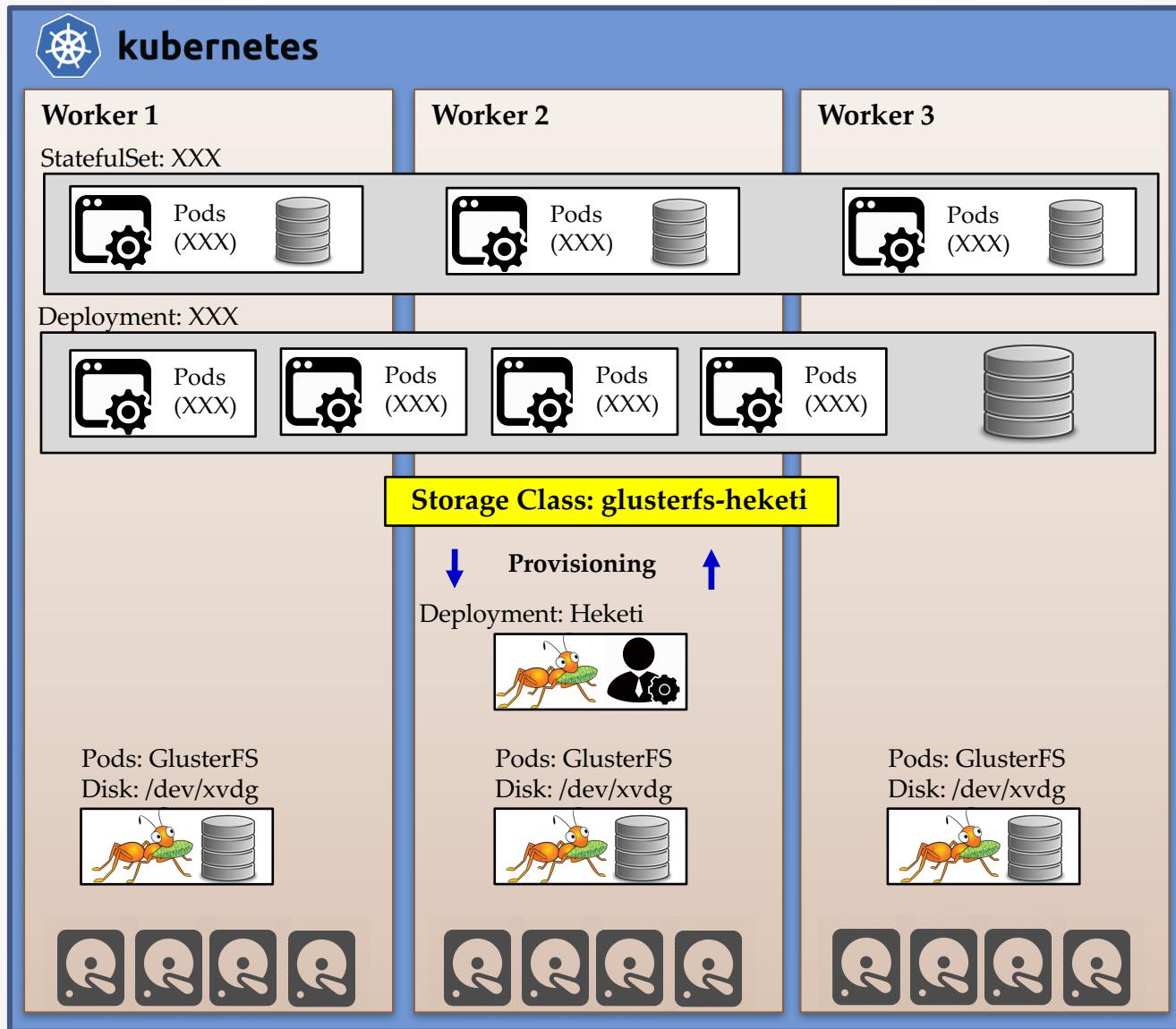
Push Messaging Processing

```
shortnote      topology.json  instructions

{
  "clusters": [
    {
      "nodes": [
        {
          "node": {
            "hostnames": {
              "manage": [
                "ip-10-21-1-164"
              ],
              "storage": [
                "10.21.1.164"
              ]
            },
            "zone": 1
          },
          "devices": [
            "/dev/xvdd"
          ]
        },
        {
          "node": {
            "hostnames": {
              "manage": [
                "ip-10-21-1-213"
              ],
              "storage": [
                "10.21.1.213"
              ]
            },
            "zone": 1
          },
          "devices": [
            "/dev/xvdd"
          ]
        }
      ]
    }
  ]
}

37   {
38     "node": [
39       {
40         "hostnames": {
41           "manage": [
42             "ip-10-21-1-97"
43           ],
44           "storage": [
45             "10.21.1.97"
46           ]
47         },
48         "zone": 1
49       },
50       {
51         "devices": [
52           "/dev/xvdd"
53         ]
54       }
55     ]
56   }
```

Push Messaging Processing



Push Messaging Processing

```
ubuntu@ip-10-21-1-158:~$ kubectl get nodes
NAME        STATUS    ROLES      AGE     VERSION
ip-10-21-1-130  Ready    <none>    13d    v1.9.2
ip-10-21-1-158  Ready    master     13d    v1.9.2
ip-10-21-1-210  Ready    <none>    13d    v1.9.2
ip-10-21-1-242  Ready    <none>    13d    v1.9.2
ubuntu@ip-10-21-1-158:~$ kubectl get pods --all-namespaces |grep gluster
default      glusterfs-44cp2           1/1       Running   0          13d
default      glusterfs-5hd2h           1/1       Running   0          13d
default      glusterfs-cpqkv          1/1       Running   1          13d
ubuntu@ip-10-21-1-158:~$ kubectl get pods --all-namespaces |grep heketi
default      deploy-heketi-7c4898d9cd-99s7t  1/1       Running   0          13d
ubuntu@ip-10-21-1-158:~$ kubectl get sc
NAME          PROVISIONER          AGE
glusterfs-heketi (default)  kubernetes.io/glusterfs  13d
ubuntu@ip-10-21-1-158:~$ 
```

```
ubuntu@ip-10-21-1-158:~/DevOpsThailand2018_Storage_K8S$ kubectl get pvc
No resources found.
ubuntu@ip-10-21-1-158:~/DevOpsThailand2018_Storage_K8S$ kubectl get pv
No resources found.
ubuntu@ip-10-21-1-158:~/DevOpsThailand2018_Storage_K8S$ kubectl create -f galera-mariadb.yaml
service "galera-mariadb" created
secret "mysql-password" created
statefulset "galera-mariadb" created
ubuntu@ip-10-21-1-158:~/DevOpsThailand2018_Storage_K8S$ kubectl get statefulset
NAME        DESIRED   CURRENT   AGE
galera-mariadb  3         1        18s
ubuntu@ip-10-21-1-158:~/DevOpsThailand2018_Storage_K8S$ kubectl get statefulset
NAME        DESIRED   CURRENT   AGE
galera-mariadb  3         1        1m
ubuntu@ip-10-21-1-158:~/DevOpsThailand2018_Storage_K8S$ kubectl get pvc
NAME          STATUS    VOLUME
mysql-datadir-galera-mariadb-0  Bound    pvc-770ca40c-9de4-11e8-9b3b-02a8867c3a3a  11G    RWO
ubuntu@ip-10-21-1-158:~/DevOpsThailand2018_Storage_K8S$ kubectl get pv
NAME          CAPACITY  ACCESS MODES  RECLAIM POLICY  STATUS    CLAIM
REASON AGE
pvc-770ca40c-9de4-11e8-9b3b-02a8867c3a3a  11G    RWO        Delete    Bound    default/mysql-datadir-galera-mariadb-0  glusterfs-heketi
i          2m
```

Push Messaging Processing

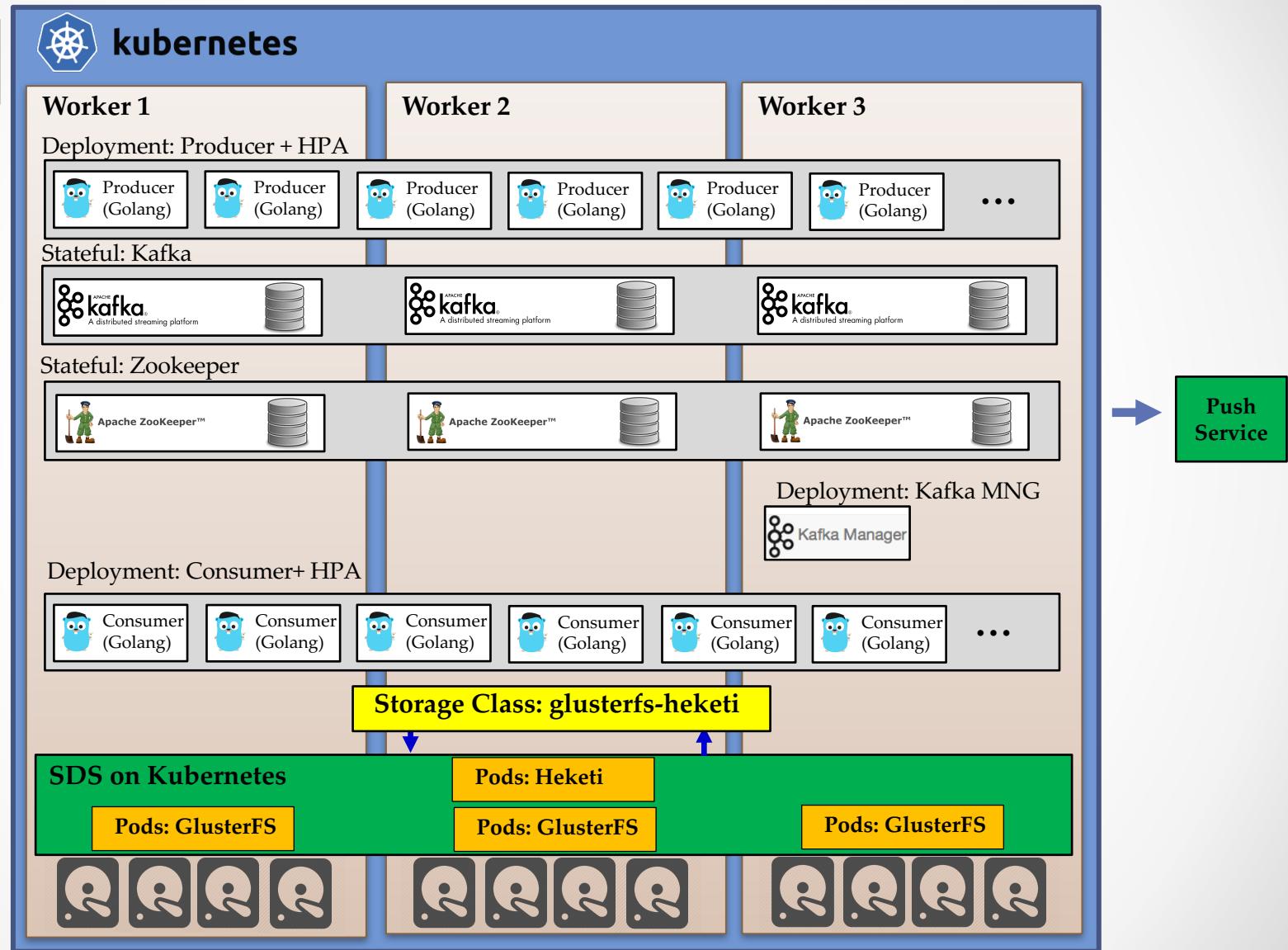
```
ubuntu@ip-10-21-1-158:~/DevOpsThailand2018_Storage_K8S$ kubectl get statefulset
NAME      DESIRED  CURRENT  AGE
galera-mariadb  3        2        3m
ubuntu@ip-10-21-1-158:~/DevOpsThailand2018_Storage_K8S$ kubectl get statefulset
NAME      DESIRED  CURRENT  AGE
galera-mariadb  3        2        4m
ubuntu@ip-10-21-1-158:~/DevOpsThailand2018_Storage_K8S$ kubectl get pvc
NAME          STATUS  VOLUME
mysql-datadir-galera-mariadb-0  Bound   pvc-770ca40c-9de4-11e8-9b3b-02a8867c3a3a  11G    RWO    glusterfs-heketi  4m
mysql-datadir-galera-mariadb-1  Bound   pvc-e6160659-9de4-11e8-9b3b-02a8867c3a3a  11G    RWO    glusterfs-heketi  1m
ubuntu@ip-10-21-1-158:~/DevOpsThailand2018_Storage_K8S$ kubectl get pv
NAME          CAPACITY  ACCESS MODES  RECLAIM POLICY  STATUS  CLAIM
REASON AGE
pvc-770ca40c-9de4-11e8-9b3b-02a8867c3a3a  11G    RWO        Delete  Bound  default/mysql-datadir-galera-mariadb-0  glusterfs-heketi
i        4m
pvc-e6160659-9de4-11e8-9b3b-02a8867c3a3a  11G    RWO        Delete  Bound  default/mysql-datadir-galera-mariadb-1  glusterfs-heketi
i        1m
ubuntu@ip-10-21-1-158:~/DevOpsThailand2018_Storage_K8S$ kubectl get statefulset
NAME      DESIRED  CURRENT  AGE
galera-mariadb  3        3        7m
ubuntu@ip-10-21-1-158:~/DevOpsThailand2018_Storage_K8S$ kubectl get pvc
NAME          STATUS  VOLUME
mysql-datadir-galera-mariadb-0  Bound   pvc-770ca40c-9de4-11e8-9b3b-02a8867c3a3a  11G    RWO    glusterfs-heketi  7m
mysql-datadir-galera-mariadb-1  Bound   pvc-e6160659-9de4-11e8-9b3b-02a8867c3a3a  11G    RWO    glusterfs-heketi  4m
mysql-datadir-galera-mariadb-2  Bound   pvc-54fc46b-9de5-11e8-9b3b-02a8867c3a3a  11G    RWO    glusterfs-heketi  55s
ubuntu@ip-10-21-1-158:~/DevOpsThailand2018_Storage_K8S$ kubectl get pv
NAME          CAPACITY  ACCESS MODES  RECLAIM POLICY  STATUS  CLAIM
REASON AGE
pvc-54fc46b-9de5-11e8-9b3b-02a8867c3a3a  11G    RWO        Delete  Bound  default/mysql-datadir-galera-mariadb-2  glusterfs-heketi
i        54s
pvc-770ca40c-9de4-11e8-9b3b-02a8867c3a3a  11G    RWO        Delete  Bound  default/mysql-datadir-galera-mariadb-0  glusterfs-heketi
i        7m
pvc-e6160659-9de4-11e8-9b3b-02a8867c3a3a  11G    RWO        Delete  Bound  default/mysql-datadir-galera-mariadb-1  glusterfs-heketi
i        4m
ubuntu@ip-10-21-1-158:~/DevOpsThailand2018_Storage_K8S$
```

Push Messaging Processing

- Enhancement
 - Component:
 - Kafka create cluster component:
 - Kafka (Stateful) with 3 Replicas
 - Zookeeper (Stateful) with 3 Replicas
 - Kafka MNG (Deployment) with 1 Replica
 - Producer (Deployment) with 10 Replicas + HPA
 - Consumer (Deployment) with 6 Replicas + HPA



Push Messaging Processing



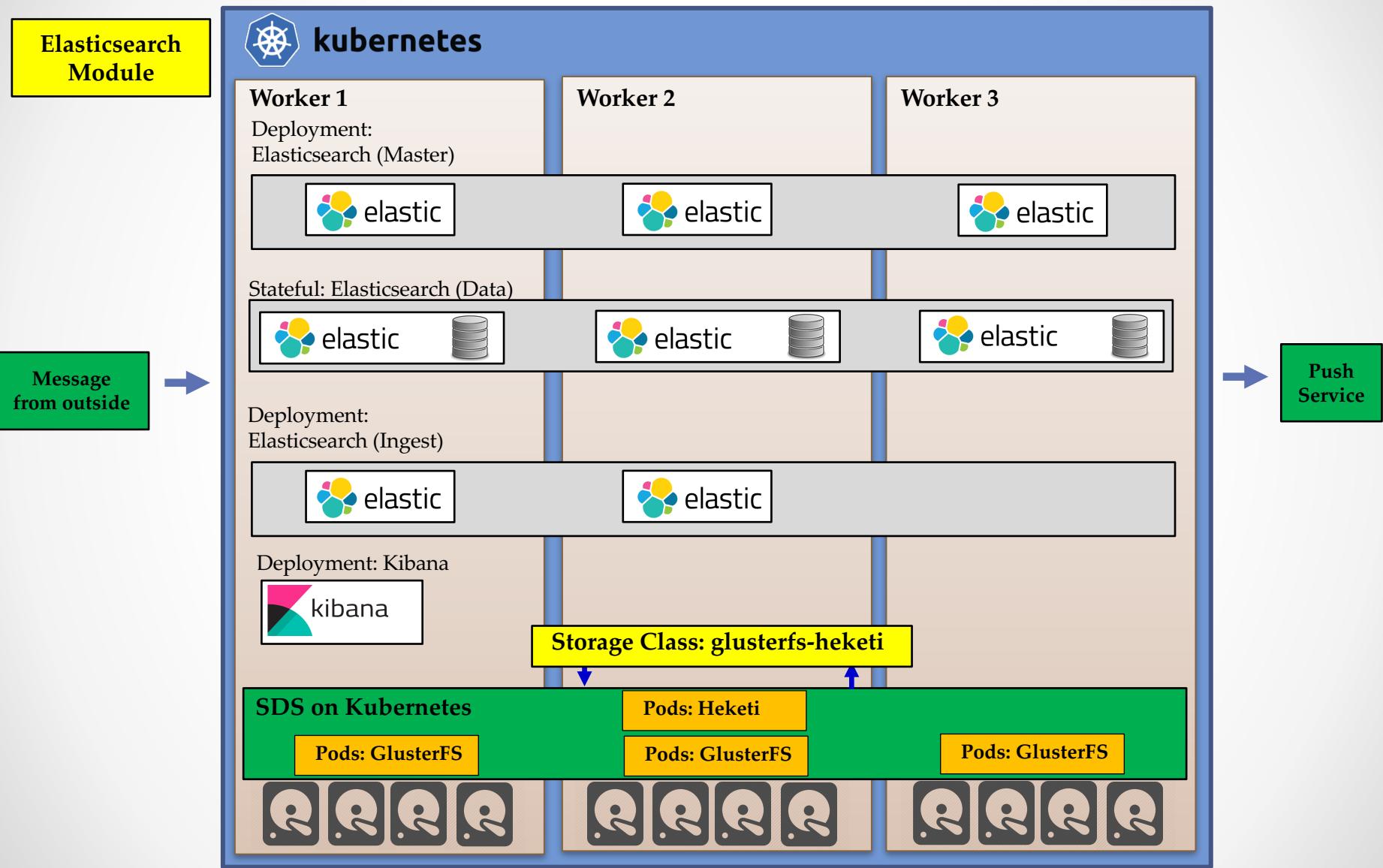
K8S Storage Built-In

Push Messaging Processing

- Enhancement
 - Component:
 - Elasticsearch create cluster and separate role as best practice:
 - Master Node (Deployment) with 3 Replicas
 - Data Node (Stateful) with 3 Replicas
 - Ingest Node (Deployment) with 2 Replicas
 - Kibana (Deployment) with 1 Repica
 - Ref:<https://www.elastic.co/guide/en/elasticsearch/reference/6.2/modules-node.html>



Push Messaging Processing



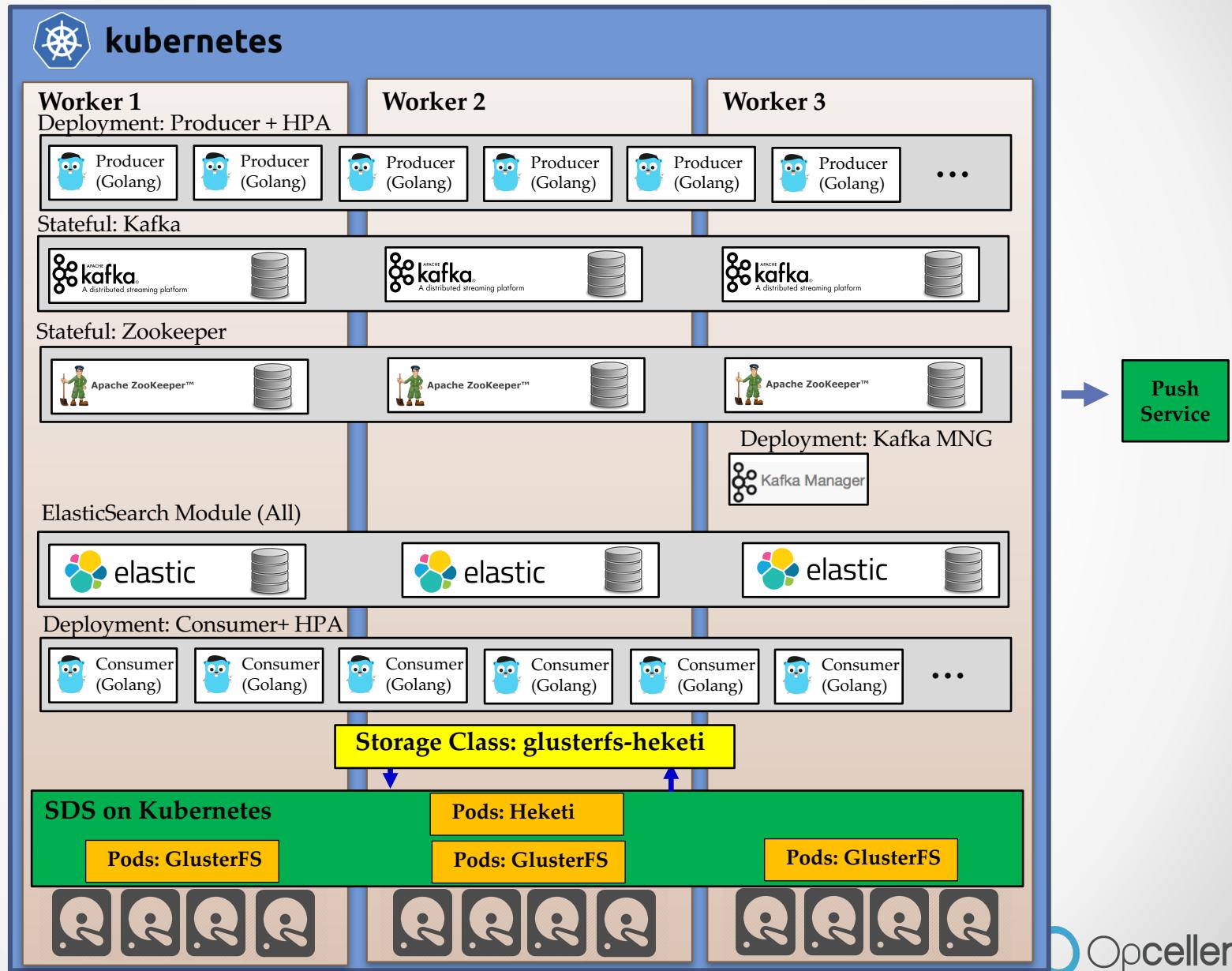
K8S Storage Built-In

Push Messaging Processing

- Benefit for enhancement
 - Cut storage dependency for each application by Storage Class
 - Remove all single point failure component
 - Compute
 - Network
 - Storage
 - Complete cluster for all component
 - Extend capability for support message queue

Push Messaging Processing

Version 2.0



Push Messaging Processing

- Future Improvement
 - Change storage provision to “Ceph RBD” (Block Storage)
 - Optional integrate with external storage on AzureDisk, AWS(EBS) etc
 - Enhance feature for online file system extension via kubernetes (“ExpandInUsePersistentVolumes”,Alpha feature on Version 1.11)
 - Replication storage from Production Site to DR Site

Demo Case: Wordpress on Scale



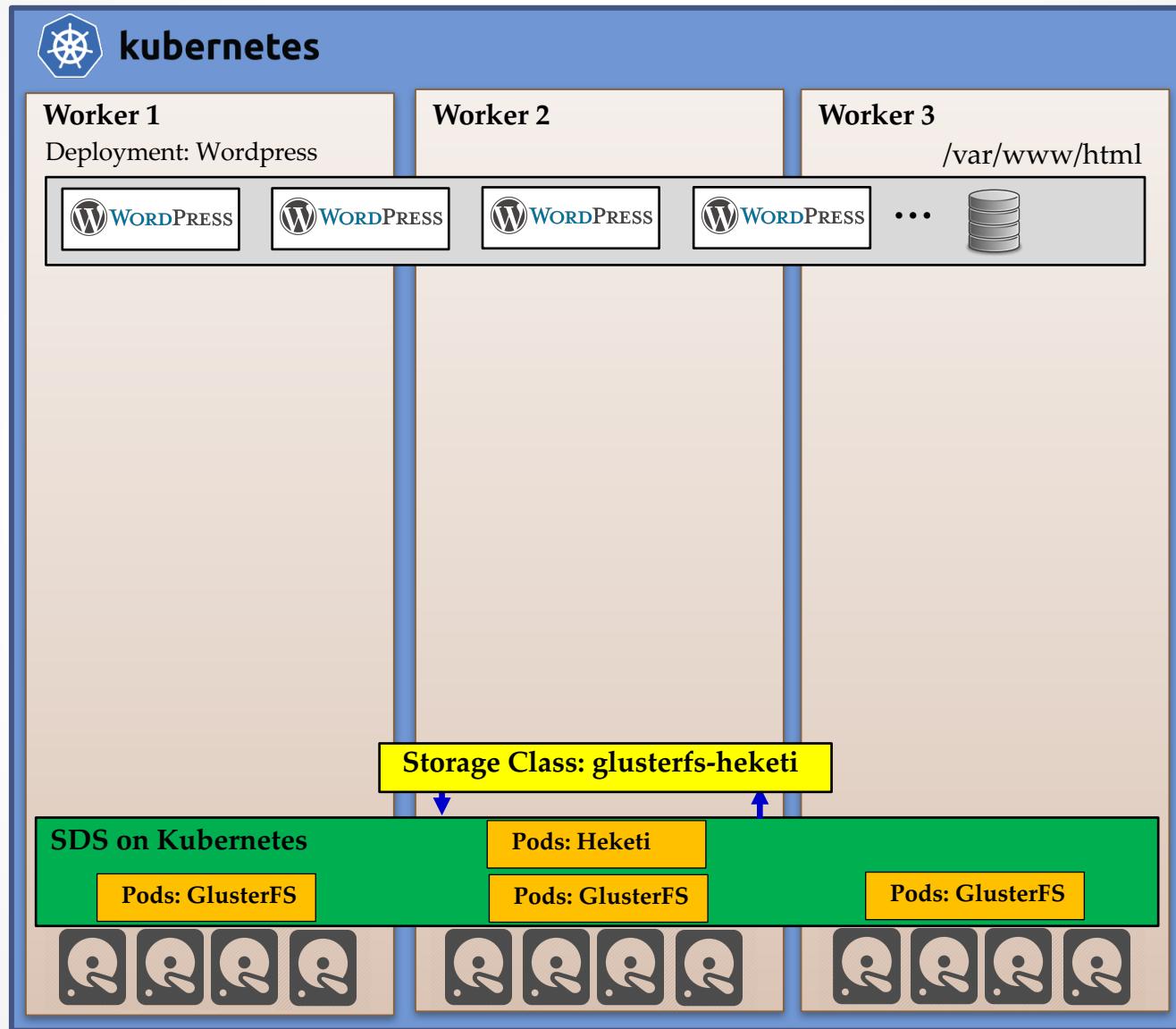
K8S Storage Build-In

 Opcellent

Wordpress OnScale

- Solution Design
 - Wordpress major container 2 components
 - Web Module: Apache + PHP (Last update also support NGINX)
 - Database Module: MariaDB / Mysql
 - How we scale it ?
 - Web Module: Will use deployment and share storage on path “/var/www/html” for all wordpress.
 - This will make all wordpress share session on same place and resolve problem of stateful at all

Push Messaging Processing



K8S Storage Built-In

Wordpress OnScale

- Solution Design
 - How we scale it ?
 - Database Module:
 - We select to use MariaDB with Galera Cluster
 - Provide Active/Active multi master technology
 - Provide 3 active database node (Read/Write)
 - Application read/write database normally
 - Keep state on “etcd” for all member of mariadb
 - wsrep_local_state_comment = “Synced”
 - wsrep_last_committed=<value>

Wordpress OnScale

The screenshot shows the MariaDB Documentation website with the following navigation bar:

- Products
- Services
- Resources
- About Us
- Contact
- Get Started (button)
- Download (button)

The breadcrumb navigation path is: Home > Resources > Knowledge Base > Library > MariaDB Documentation > High Availability & Performance Tuning > MariaDB Galera Cluster > What is MariaDB Galera Cluster?

What is MariaDB Galera Cluster?

On the left, there's a sidebar with links to various MariaDB components and utilities:

- Home
- Open Questions
- MariaDB
- MariaDB MaxScale
- MariaDB ColumnStore
- Connectors
- All Topics
- History
- Source
- Flag as Spam / Inappropriate
- Translate

Below the sidebar, there's a box containing creation and modification details:

Created
6 years, 4 months ago
Modified
6 days, 5 hours ago
Type
article
Status
active
License
CC BY-SA / Gnu FDL

Links for History and Comments are also present.

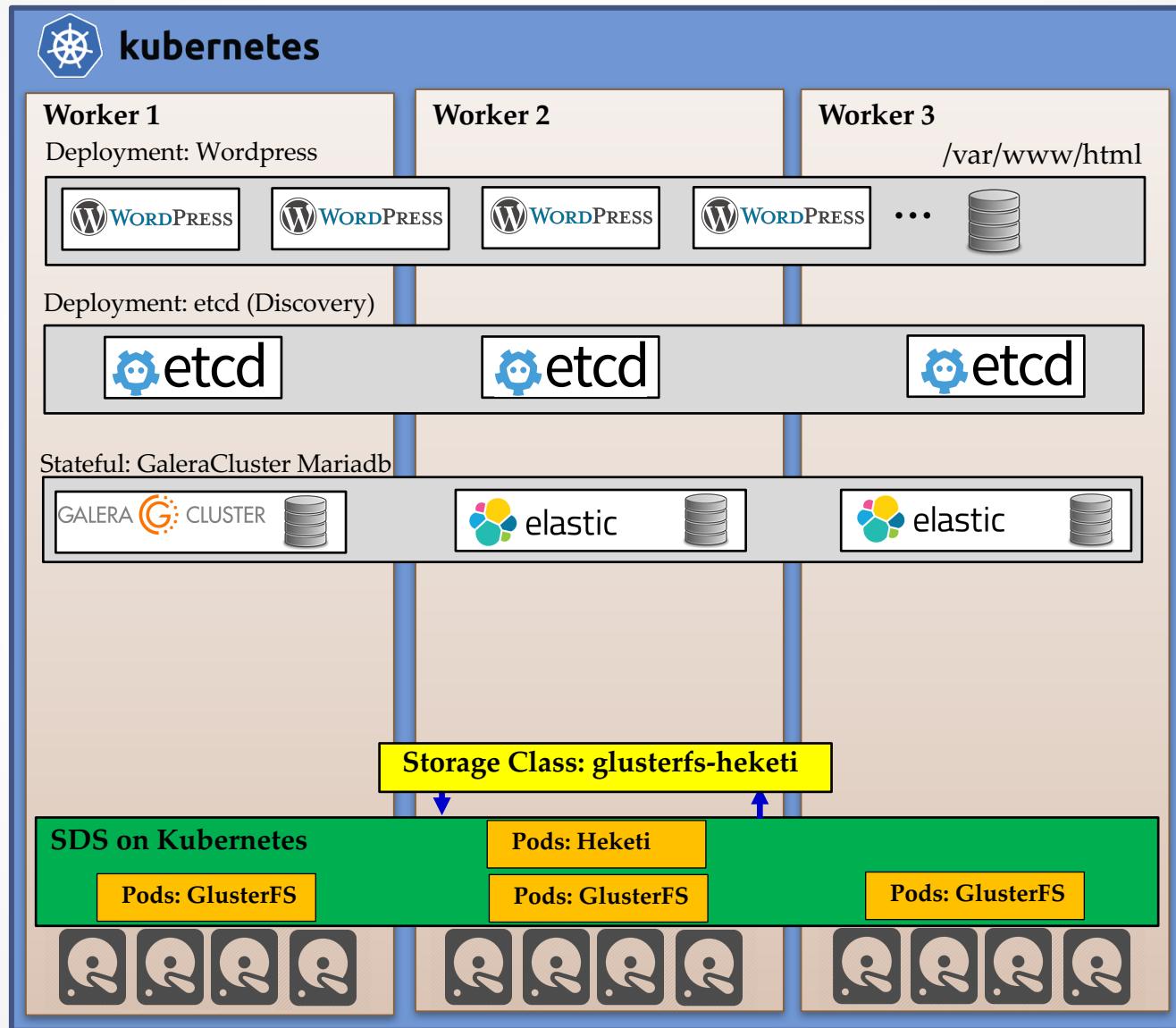
The main content area displays five boxes, each representing a different release of MariaDB Galera Cluster:

- The most recent release of MariaDB 10.3 is:
MariaDB 10.3.8 Stable (GA) [Download Now](#)
- The most recent release of MariaDB 10.2 is:
MariaDB 10.2.16 Stable (GA) [Download Now](#)
- The most recent release of MariaDB 10.1 is:
MariaDB 10.1.35 Stable (GA) [Download Now](#)
- The most recent MariaDB Galera Cluster 10.0 release is:
MariaDB Galera Cluster 10.0.36 [Download Now](#)
- The most recent MariaDB Galera Cluster 5.5 release is:
MariaDB Galera Cluster 5.5.61 [Download Now](#)

On the right side, there's a sidebar titled "What is MariaDB Galera Cluster?" with a list of related topics:

- ↑ MariaDB Galera Cluster
- ↑
- What is MariaDB Galera Cluster?**
- About Galera Replication
- Galera Use Cases
- Getting Started with MariaDB Galera Cluster
- MariaDB Galera Cluster - Known Limitations
- Galera Cluster Status Variables
- Galera Cluster System Variables
- Building the Galera wsrep Package on Ubuntu and Debian
- Building the Galera wsrep Package on Fedora
- Installing Galera from Source
- Galera Test Repositories
- wsrep_provider_options
- Galera Cluster Address
- Galera Load Balancer
- MariaDB Galera 10.0 Release Notes
- MariaDB Galera 10.0 Changelogs
- MariaDB Galera 5.5 Release Notes
- MariaDB Galera 5.5 Changelogs
- Tips on Converting to Galera
- Upgrading from MariaDB Galera Cluster 10.0 to MariaDB 10.1
- Upgrading from MariaDB Galera Cluster 5.5 to MariaDB Galera Cluster 10.0

Push Messaging Processing



K8S Storage Built-In

Wordpress OnScale

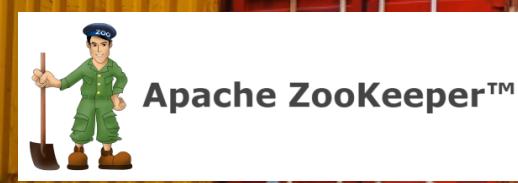
- Demo Session

```
ubuntu@ip-10-21-1-158:~/DevOpsThailand2018_Storage_K8S$ kubectl get pods -o wide
NAME           READY   STATUS    RESTARTS   AGE     IP          NODE
deploy-heketi-7c8989dcd-99s7t  1/1    Running   0          1h      192.168.250.1  ip-10-21-1-242
etcdb0         1/1    Running   0          47m    192.168.250.2  ip-10-21-1-242
etcdb1         1/1    Running   0          47m    192.168.99.130 ip-10-21-1-210
etcdb2         1/1    Running   0          47m    192.168.9.2    ip-10-21-1-130
galera-mariadb-0  1/1    Running   0          5m     192.168.99.144 ip-10-21-1-210
galera-mariadb-1  1/1    Running   0          43m    192.168.9.3    ip-10-21-1-130
galera-mariadb-2  1/1    Running   0          39m    192.168.250.3  ip-10-21-1-242
glusterfs-44cp2  1/1    Running   0          1h     10.21.1.242   ip-10-21-1-242
glusterfs-5hd2h  1/1    Running   0          1h     10.21.1.210   ip-10-21-1-210
glusterfs-cpqkv 1/1    Running   0          1h     10.21.1.130   ip-10-21-1-130
wordpress-7875b685c8-4fpp1  1/1    Running   1          22m    192.168.99.142 ip-10-21-1-210
wordpress-7875b685c8-4qp85  1/1    Running   1          22m    192.168.9.11   ip-10-21-1-130
wordpress-7875b685c8-522jw  1/1    Running   0          22m    192.168.250.10 ip-10-21-1-242
wordpress-7875b685c8-b9c7x  1/1    Running   2          22m    192.168.250.12 ip-10-21-1-242
wordpress-7875b685c8-bg9p9  1/1    Running   1          22m    192.168.250.11 ip-10-21-1-242
wordpress-7875b685c8-gd29v  1/1    Running   0          22m    192.168.9.10   ip-10-21-1-130
wordpress-7875b685c8-h9vdb  1/1    Running   2          22m    192.168.9.12   ip-10-21-1-130
wordpress-7875b685c8-nzkbk  1/1    Running   0          22m    192.168.99.143 ip-10-21-1-210
wordpress-7875b685c8-pb2st  1/1    Running   0          22m    192.168.99.140 ip-10-21-1-210
wordpress-7875b685c8-v5kdg  1/1    Running   2          22m    192.168.99.141 ip-10-21-1-210
ubuntu@ip-10-21-1-158:~/DevOpsThailand2018_Storage_K8S$ kubectl delete pods/galera-mariadb-0
pod "galera-mariadb-0" deleted
ubuntu@ip-10-21-1-158:~/DevOpsThailand2018_Storage_K8S$ 
```

```
ip-10-21-1-213      ip-10-21-1-213      Worker2      10.21.1.213  54.251.14.14  scblf_laminsta
ip-10-21-1-97      ip-10-21-1-97      Worker3      10.21.1.97   52.221.14.14  scblf_laminsta
=====
Device Disk on device: /dev/xvdf on Worker 1/2/3
1. Create StorageClass for Dynamic Provision:
kubectl get svc
cd ./DevOpsThailand2018_Storage_K8S
vi Storageclass.yaml ==> Add ip address of heketi server
kubectl create -f Storageclass.yaml
kubectl get sc
2. Create Etcdb Cluster for Discovery Service:
kubectl create -f etcdb-cluster.yaml
watch kubectl get pods -o wide
watch kubectl get svc
3. Create Galera Cluster MariaDB by command: (5 - 10 min)
kubectl create -f galera-mariadb.yaml
watch kubectl get pods -o wide
4. Create Wordpress for Access and Get by command: (5 - 10 min)
kubectl create -f wordpress.yaml
watch kubectl get pods -o wide
5. Check Service by Command: kubectl get svc and tried to access via browser==> http://<public ip address>:31000
```



WORDPRESS



By Praparn Luengphoonlap
Email: praparn@opcellent.com

K8S Storage Built-In

