Evaluation and Implementation of Various Persistent Storage Options for CMSWEB Services in Kubernetes Infrastructure

Muhammad Imran[1], Valentin Kuznetsov[2], Panos Paparrigopoulos[1] Spyridon Trigazis[1], Andreas Pfeiffer[1]
[1]CERN, Geneva, Switzerland, [2]Cornell University, USA
Email: muhammad.imran@cern.ch

Introduction

- This poster summarizes the various storage options that we implemented for the CMSWEB cluster in k8s.
- All CMSWEB services require storage for logs, while some services also require storage for data.
- We also provide a feasibility analysis of various storage options and describe the pros/cons of each technique from the perspective of the CMSWEB cluster and its users.
- In the end, we provide the recommendations with respect to the service requirements.

1- EOS

- In the VM cluster, we use EOS for archiving logs.
- In the K8s cluster, we tested it by configuring EOS inside the pods of the CMSWEB services.
- Logs from the pods are sent to EOS using the rsync approach at predefined interval.
- Cons
  - Logs between rsync intervals are lost if the service container restarts

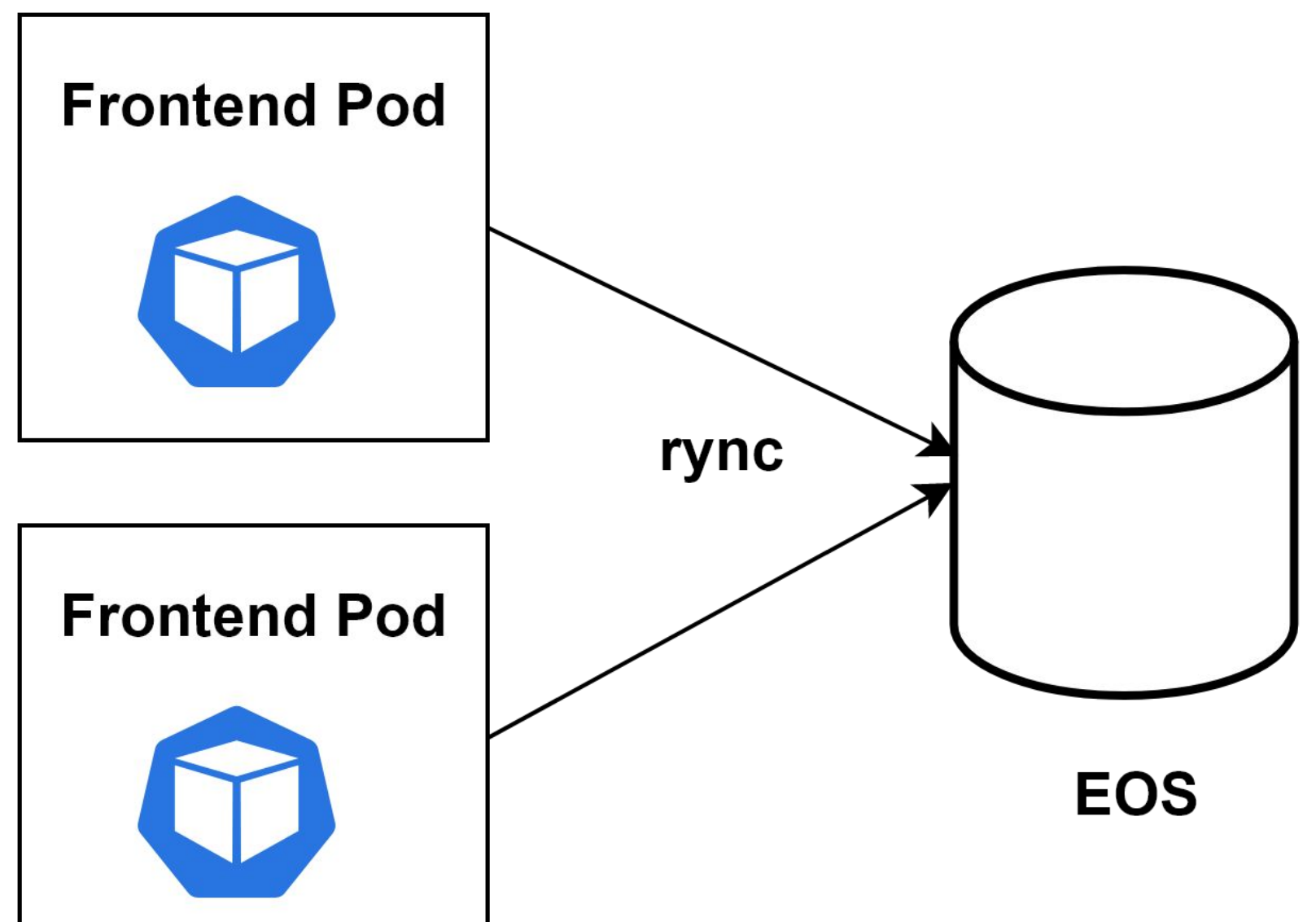

Fig. 1

2- Dedicated VM

- In the VM cluster, we used dedicated VM to provide few days logs to users.
- In the K8s cluster, we implemented the same technique using rsync approach to a dedicated VM.
- Logs from the pods are sent to the VM at predefined interval.
- Cons
  - Logs between intervals are lost if the service container restarts.

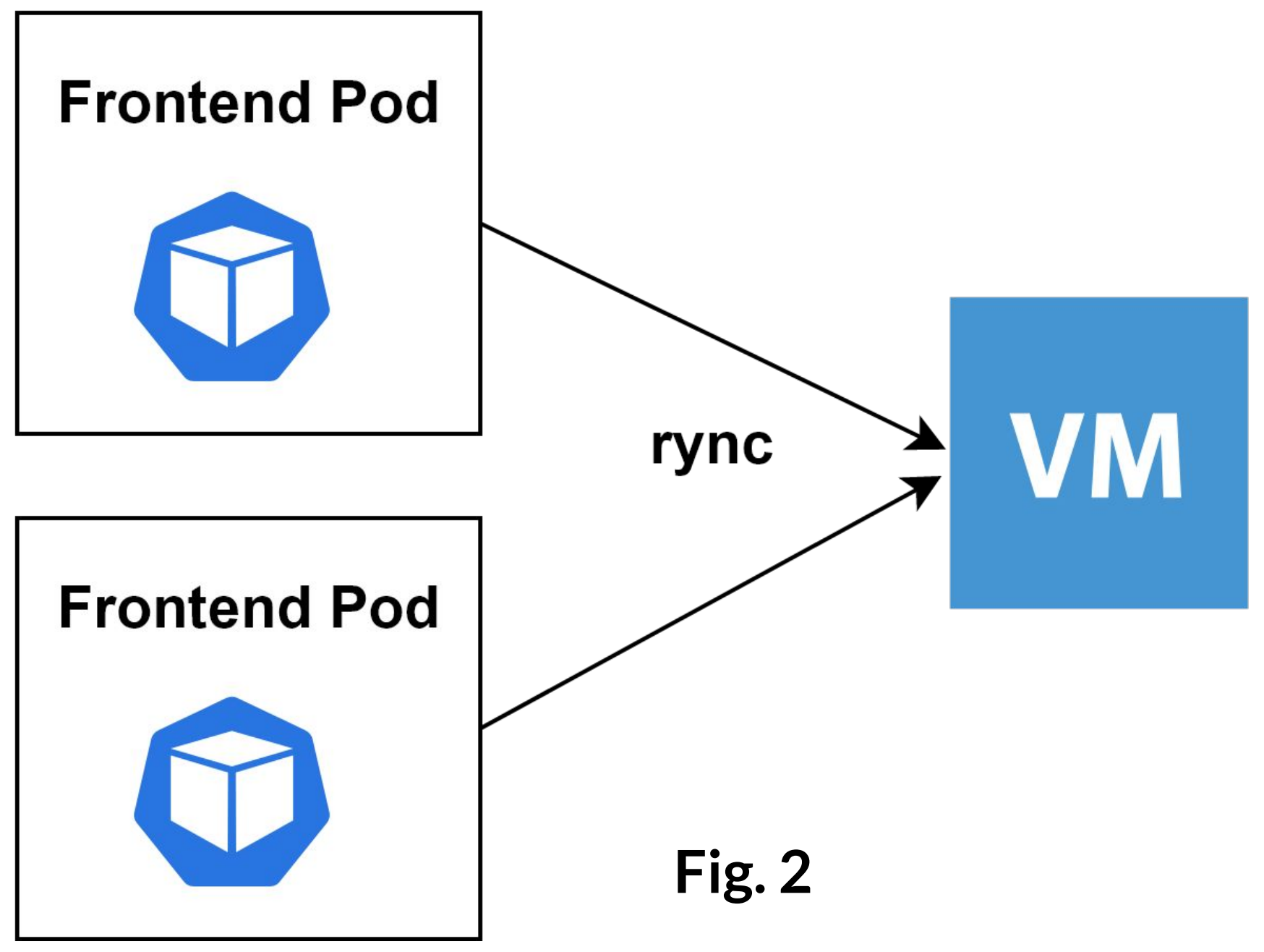

Fig. 2

3- CephFS

- CephFS shares can be mounted multiple times across various clusters and VMs and work very well with k8s.
- We can use it both for data and the logs. The data/logs are available to the users in real-time.
- Cons
  - Service availability depends upon CephFS shares.
  - Logs searching e.g. grep is very slow.

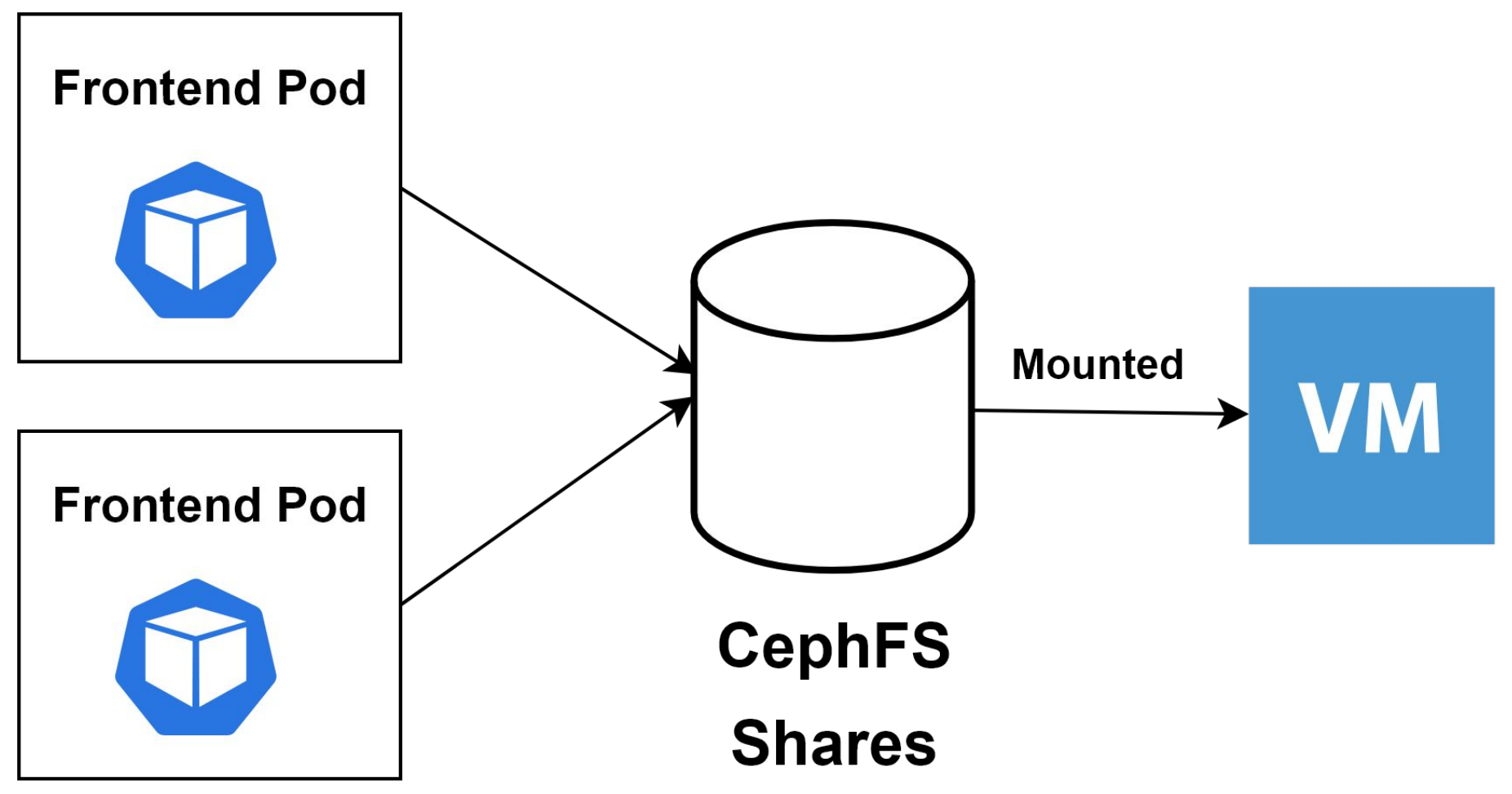

Fig. 3

4- Cinder

- Cinder is the block storage that runs the filesystem on top of it.
- It can only be attached to one instance at a time.
- We use this option where large amount of storage for the data is needed for singleton service.
- Cons
  - Cannot mount to any VM.
  - Accessible via services pod only.
  - Can only be attached to single instance.

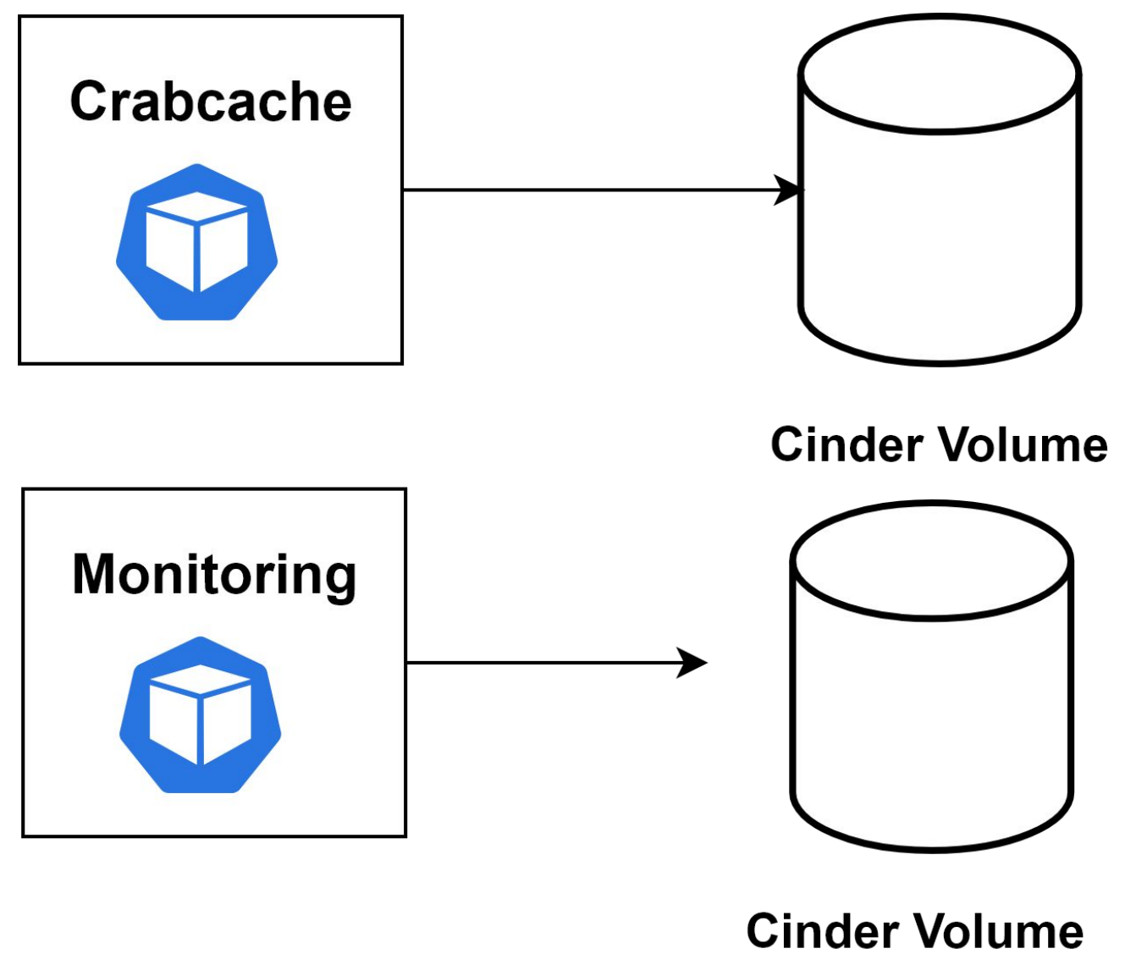

Fig. 4

5- S3 using Fluentd

- S3 is an object storage that offers a scalable storage service that can be used by applications compatible with the Amazon S3 protocol.
- It is used for the logs.
- Services pod run fluentd as a sidecar container that sends logs to S3 bucket at predefined interval.
- Cons
  - Logs between fluentd intervals are lost if the service container restarts
  - Fluentd requires additional resources and maintenance

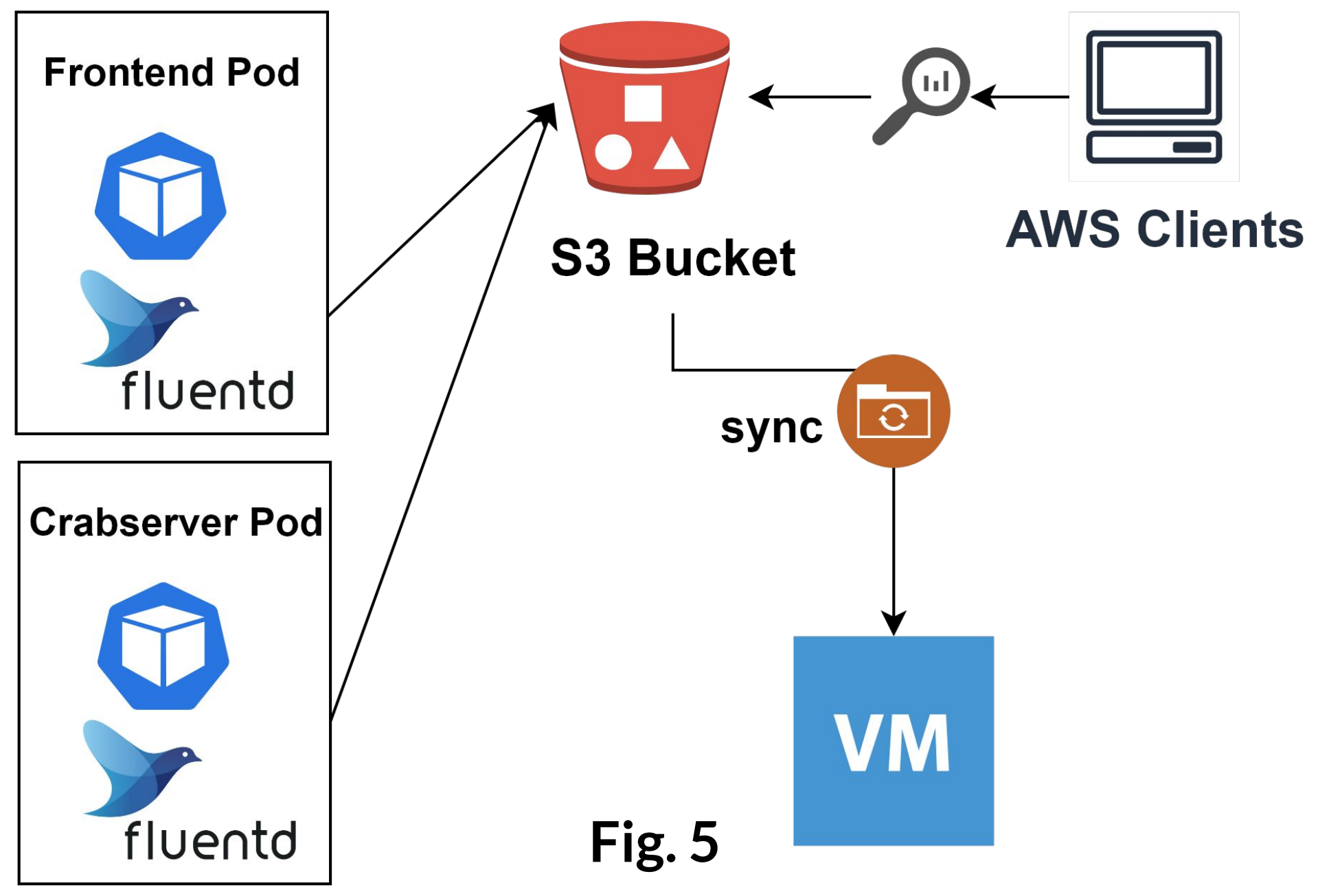

Fig. 5

6- S3 and Elasticsearch using Filebeat & Fluentd

- Services pod run filebeat as a sidecar container.
- Filebeat scraps logs to fluentd.
- Fluentd sends logs parallel to S3 bucket and elasticsearch at predefined interval.
- Logs are synced to VM from S3 bucket using AWS clients.
- Logs are not lost with pods restart.
- Cons
  - Extra metadata in the log entry i.e. podname
  - Complex workflow which requires proper setup and maintenance

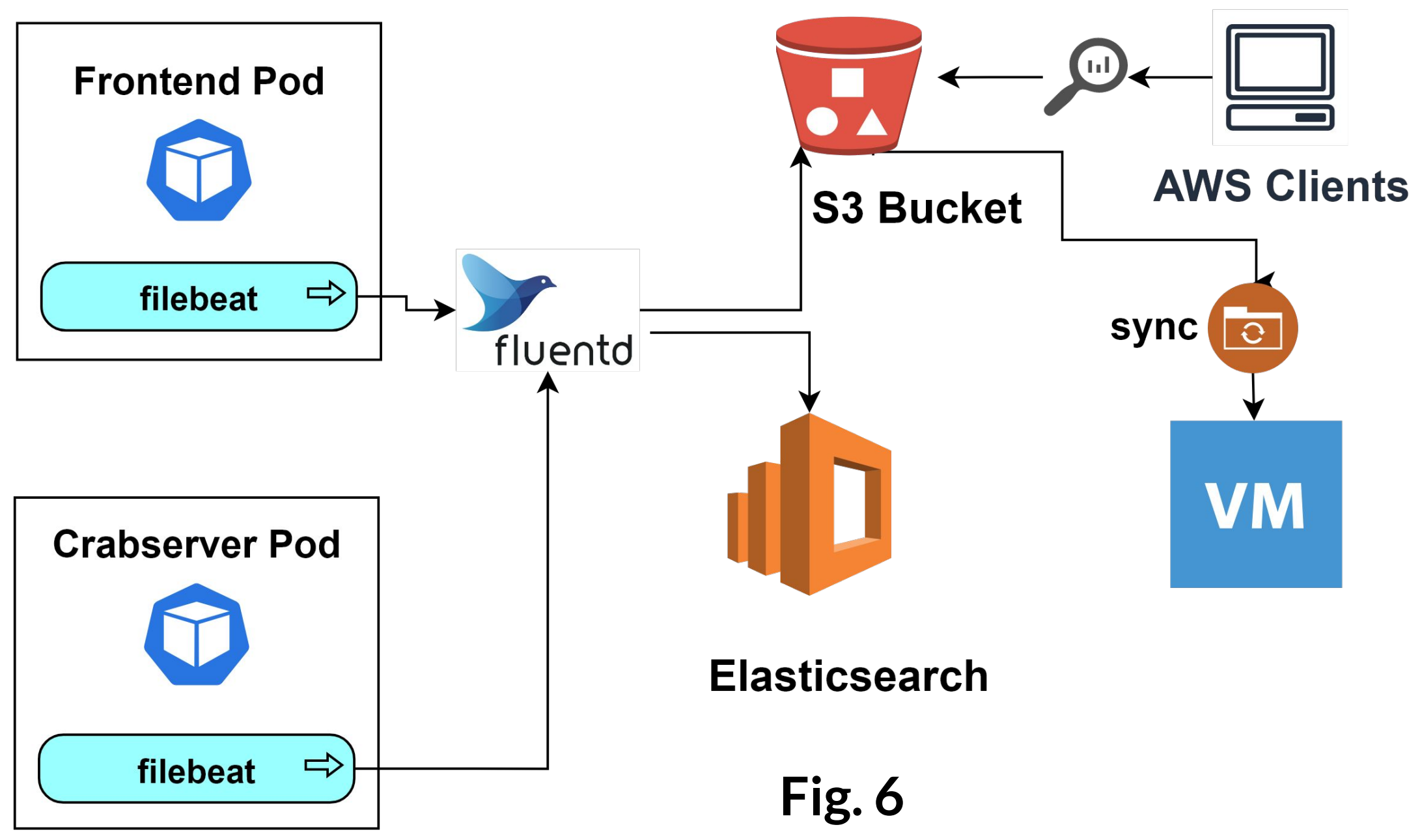

Fig. 6

Feasibility Analysis

Storage Options	Accessibility	Usage	Searching	Data Lost	Service Dependency
EOS	At interval	Logs	Easy	yes	No
Dedicated VM	At interval	Logs	Easy	yes	No
CephFS	Real time	Data / Logs	Slow grep	No	Yes
Cinder	Real time	Data	Within pod	No	Yes
S3 using Fluentd	At interval	Logs	Easy	yes	No
S3 and Elasticsearch using Filebeat and Fluentd	At interval	Logs	Easy	No	No

Conclusion

- We investigated the feasibility of various storage options for logs and data for the CMSWEB k8s cluster.
- A number of different storage options, from file-system-like (EOS) to object-store-like (S3) were studied and the results compared.
- Recommendations:
  - Use “S3 and Elasticsearch using Filebeat and Fluentd” for maintaining logs.
  - Use Cinder for large data volumes e.g. DB backend.
  - Use CephFS for service data storage e.g. cache