Use your preferred methods to retrieve the contents of `/var/lib/icp/helmrepo` from the master node and store the data outside of the cluster.

Backing up the Elasticsearch cluster

As the Elasticsearch, Logstash and Kibana (ELK) stack provides users with the ability to retrieve historical log data it is an important component to backup so the same data can be restored in the event of cluster failure. The current logging release does not provide the capability to backup the logging data without suspending the logging service (only partially for installations with multiple management node).

General considerations

By default, IBM Cloud Private configures the ELK stack to retain log data using the logstash index for one day, and in this configuration, it is worth considering whether or not the platform log data is actually meaningful enough to keep. In most cases, the platform log data is transient, especially if the default curation is kept and logs are removed after 24 hours.

Backing up log data is only really valuable if the default 24 hours is extended to a longer duration and where the platform ELK stack is the primary source of application log data. The Logging and Monitoring Chapter provides information on the various use cases for platform and application logging and the alternative approaches to consider, such as deploying a dedicated ELK for applications.

Backing up the logging filesystem

This method requires downtime of management nodes to be able to accurately backup the logging data. This is because Elasticsearch is constantly reading/writing data to the filesystem at `/var/lib/icp/logging` on the management nodes especially when the cluster is active. Attempting to copy the logging data from the filesystem whilst the Elasticsearch cluster is online has a high chance of creating a faulty backup and some shards will be corrupted upon restoration, resulting in data loss.

The recommended way to backup the logging file system is to ensure Elasticsearch is not running on the management node by stopping all containers running on it. Use the method described in “Stopping an IBM Cloud Private node” on page 78 to stop kubelet and docker, then the `/var/lib/icp/logging/elk-data/nodes/0/indices` directory can be safely copied. During this time, no log data generated by the platform or application will be persisted.

In environments with more than one management node, multiple Elasticsearch data pods are deployed (one on each management node), so it is possible to keep the logging services running by repeating the above approach on each management node one by one. During this time, Elasticsearch will persist the data only to the available data pods, which means that there will be an imbalance of spread of replicas per shard on each management node.

Elasticsearch will attempt to correct this as data pods are brought back online so higher CPU utilization and disk I/O is normal in this situation. More information about how replica shards are stored and promoted when data pods are taken offline can be found here:

https://www.elastic.co/guide/en/elasticsearch/guide/current/replica-shards.html

Backing up the monitoring data

The monitoring stack comprises of the Alert Manager, Prometheus and Grafana. By default, IBM Cloud Private does not deploy these components with a PersistentVolume (PV) or PersistentVolumeClaim (PVC), so any data stored by these components will be deleted if the container gets restarted. The platform does, however, store any user-configured AlertRules and MonitoringDashboard resources within Kubernetes itself, so these will persist during container restarts.