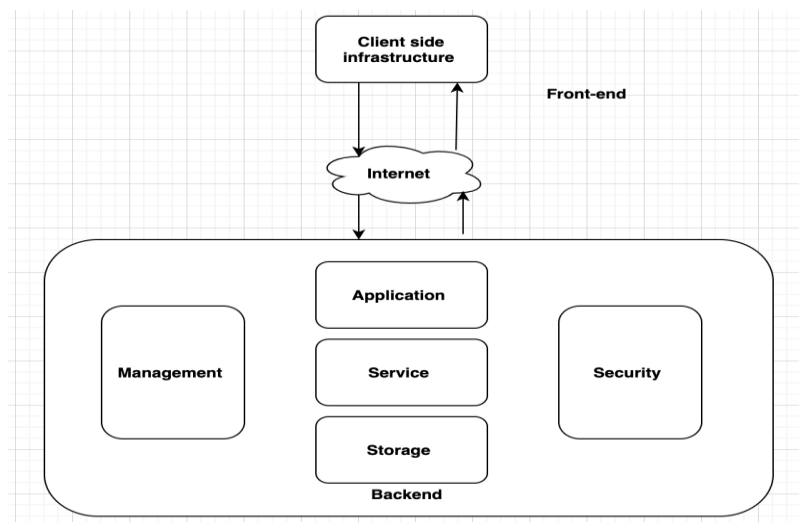


Table of Contents

- 1 Cloud architecture
 - 1.1 IBM cloud
 - 1.1.1 IBM kubernetes
 - 1.1.2 IBM PostgreSQL database
- 2 . Kubernetes System
 - 2.1 kubernetes architecture
 - 2.2 kubernetes features
 - 2.3 kubernetes Advantages
- 3 PostgreSQL
 - 3.1 Architecture of PostgreSQL
 - 3.2 Feature and benefits of PostgreSQL
- 4 Storage in Kubernetes
 - 4.1 Why do we focus on state full apps?
 - 4.2 Persistent volume
 - 4.3 Persistent volume Claim
 - 4.4 Storage class
- 5 Backup System
 - 5.1 Why backup?
 - 5.2 Different backup Strategies
- 6 Different methods of backup in Kubernetes
 - 6.1 Velero
 - 6.2 Crunchy Data operator
 - 6.3 Third Party Helm chart
 - 6.4 Alternative backup system
- 7 Conclusion

1. Cloud architecture:

Cloud is a fast growing technology with the computer system availability. This technology is mainly for storing and accessing data over the network. Cloud architecture is defined as the relationship between various components such as application, database and software capabilities which are necessary to build a cloud are connected to provide an online platform to run an application. The main goal of the cloud architecture is to provide the unique application with the user requirements. Based on the architecture the user can design their own customized application in the cloud. The cloud architecture is segregated into two segments front-end and back-end with network connectivity. Front-end is mainly for the users to access the required application. Backend consists of service, storage, server and various computers.



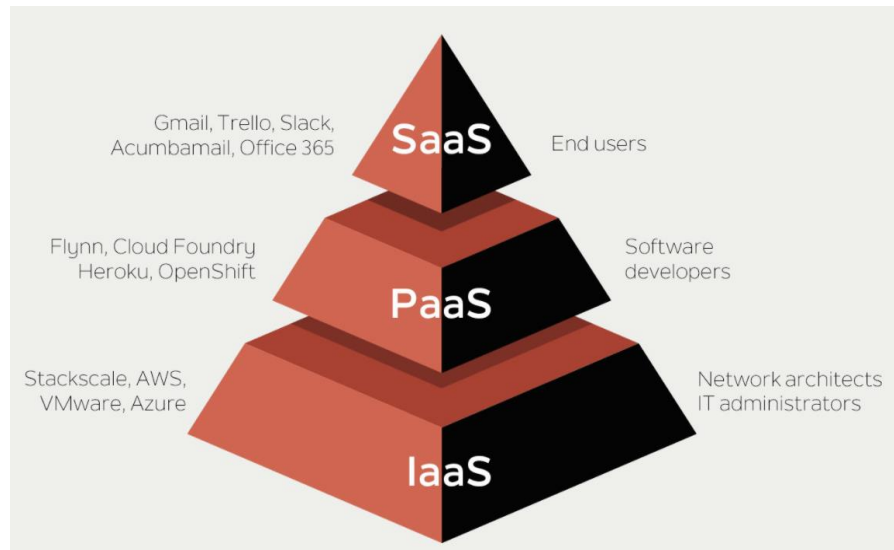
I. Cloud architecture

There are three main categories of the cloud services in cloud architecture are:

Software as a service (SaaS): Software as a service is a software distribution model which involves software hosting and maintained over the internet. With the SaaS the user can easily use the software through the web browser instead of installing and downloading on the user side

Platform as a service (PaaS): Platform as a service provides the cloud computing platform such as the third party provider which delivers the hardware and software to develop, run and manage the user application over the internet

Infrastructure as a service (IaaS): Infrastructure as a service provides infrastructure and hardware over the network which also helps the user to manage and plan the backup and recovery system. IaaS proposing pay per use method the customer can only pay for the resources that they are actually using.



II. cloud services in cloud architecture

1.1 IBM cloud

IBM cloud is appropriate for services like Platform as a service and Infrastructure as a service. Additionally, using IaaS a company can access by deploying a virtual IT resources for instance storage, Power, networking over the internet.

IBM cloud delivers more security and open public cloud for all organizations which also provide hybrid multi-cloud and private cloud. For security IBM cloud uses various cryptographic technologies. Provide reliability and continuous security across our client's complete environment. Moreover, IBM cloud platforms contain over 170 products and services with advanced data and AI, IoT and blockchain.

There are a lot of advantages in IBM cloud such as

- Innovate anywhere with the latest technologies from any vendor.
- Modernize once and deploy anywhere, without recoding.
- Improve ROI on existing clouds while maintaining full visibility and control.
- Optimize on the right cloud model and meet unique workload requirements.
- Speed innovation with advanced public cloud services from IBM or virtually any cloud.
- Realize a fast, secure journey with unmatched industry expertise.

In this document we have performed a **kubernetes** backup system for a **PostgreSQL database IBM Kubernetes:**

IBM Kubernetes managed intelligent scheduling, self-healing, horizontal scaling, service discovery and load balancing, automated rollouts and rollbacks, and secret and configuration management. Furthermore, IBM Kubernetes has advanced competencies over the cluster managements, security, design, and integrated tools for deployments. Additionally, IBM Watson and IBM Blockchain platform runs on Kubernetes with continuous and high availability master enable multi-zone clusters across 6 regions and 35 data centres. IBM Kubernetes also provides

management tools for deploying, scaling, automating, monitoring and many other options with high security. Moreover, IBM also have few set of Helm Charts in their internal IBM cloud Private Repo.

1.2 IBM PostgreSQL database:

PostgreSQL is an open source object relational database. IBM provides database as a service which is easy to set up, maintain and manage with high availability

IBM cloud database provides hosting, management, high availability, scalability, easy access, backup and restore, version upgrades, security, and many other advantages. IBM PostgreSQL database provides 9.X and 11 versions with latest features.

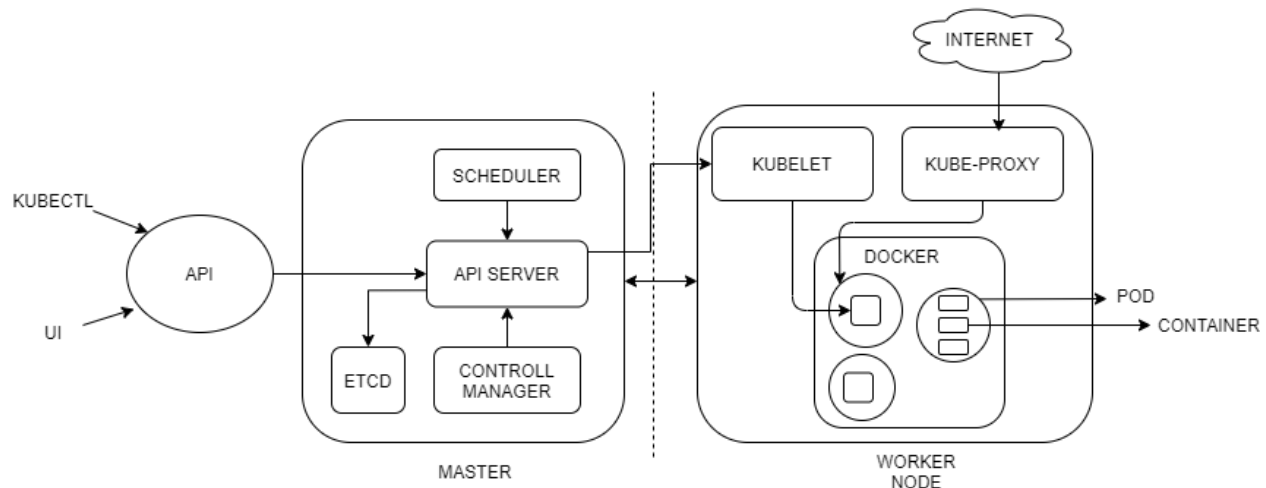
Key features:

- A database service built and accessed through a cloud platform
- Enables enterprise users to host databases without buying dedicated hardware
- Can be managed by the user or offered as a service and managed by a provider
- Can support relational databases (including MySQL and PostgreSQL) and NoSQL databases (including MongoDB and Apache CouchDB)
- Accessed through a web interface or vendor-provided API

2. Kubernetes System:

2.1 Kubernetes Architecture:

Kubernetes system environment contains Master and Worker nodes. In reality master node can be more than one for high availability purposes and the worker node will go up to hundred even thousand as well. Currently kubernetes supports up to 5000 worker nodes per cluster. Worker nodes are nothing but a virtual machine or a physical server running within the data center which runs the user workload by placing containers into pods to run a node. The master and the worker node form together as a cluster. Master node is responsible for managing the whole cluster which monitors the health check of the entire system. It stores the information of the members of the cluster and its configuration inside the master. Incase if the worker node fails it loads the worker node from another healthy worker node. Master node is responsible for the scheduling, provisioning, controlling and exposing API to the clients .Pod is a scheduling unit in the kubernetes each pod consists of one or more containers. Mostly the pod consists of one containers but there are scenarios where a pod consist of two or more containers. Those containers are dependent containers which set inside a single pod where one container can be helping another containers. We can deploy multiple container together with the help of pods, so the pod helps to wrap and manage the containers. Containers are run time environments for containerized applications. Mostly we run containerized applications inside the container. These containers fits inside the pod which runs micro-services and which are not ideal for running monolithic applications



III. Kubernetes Architecture

Master node is responsible for the entire system. It coordinates all the activity inside the cluster and communicates with the worker nodes to keep kubernetes and the user application processing. While installing Kubernetes there are four primary components of kubernetes master will get installed

The **API server** which is a gatekeeper for the complete cluster. API server validates and configures the API objects such as pod, services, replications, controller and deployments. It exposes various API for every operation. We can interact with the API server using the tool named as kubectl. The kubectl helps to perform the operations that are given from the command line.

Scheduler is responsible for physically scheduling the pods across multiple nodes depending upon the constraints that we mention in the configuration file. The scheduler schedules the pods accordingly. Moreover, it regulates the task worker nodes. It store the resource usage information for each worker node

Control Manager: There are four controllers behind the control manager. Node controller, replication controller, endpoint controller and service accountant token controllers these controllers are responsible for the health of the entire system. Controller observe the desire state of the objects it manager and notice their current state through the API server

Etd: Distributed key value lightweight database which is a central database to store the current cluster nodes at all points of time. It can be a part of kubernetes master or it may be configured externally

Worker node can be a virtual machine, a physical server where the containers are deployed. Every node in the kubernetes cluster must run a container run time such as Docker or Rocket. Worker node manage the connections, network, and assign the worker to the scheduler container

Kubelet and kube-proxy are responsible for the communication purpose. Kubelet is a primary node agent that runs on each worker node inside a cluster so the primary objective of the Kubelet is to look at the pod which is submitted to the API server on the kubernetes master and ensure that containers are described in the running pod and the health of the pod. Kube-proxy is a

critical element inside the kubernetes cluster it is responsible for maintaining the network for the total cluster configuration. Single worker node can have a single kube-proxy. It maintains the distributed network across all the nodes, across all the pods and across all containers and listens to the API server for each service endpoint creation and deletion.

Pods and Containers: Pod is a scheduling unit like VM ware. Each pod consists of one or more containers. Main advantage of Pod is we can deploy multiple dependent containers together. It act like a wrap around these containers. We used to interact these container via pod Container provides the run time environment for the applications which is designed to run micro-services. We can run containerized application processes inside these containers. This container is located inside the pod. Kubernetes supports Docker and Rocket based containers. Kubelet, kube-proxy, pod and container are the four major things inside the worker node.

2.2 Feature of Kubernetes:

- Self – monitoring - Kubernetes helps to monitor the worker node and pod health
- Horizontal Scaling – helps to monitor the entire application
- Automatic healing – incase if the pod dies it automatically recreate the pod
- Automatic Scheduling - help to schedule the tasks spontaneously
- Automatic Rollouts and rollbacks – it will roll back the changes of accident mistakes
- Provides extra services – kubernetes provided added services like storage, networking, security and many more.

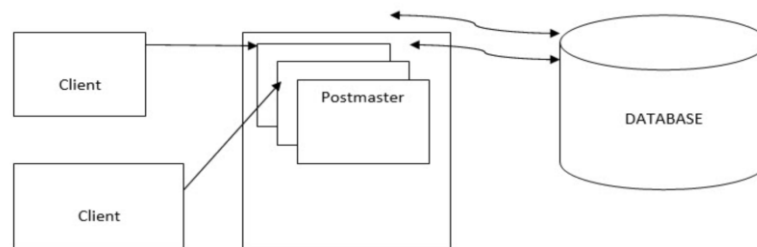
2.3 Advantage of Kubernetes:

Container management system provides a number of opportunities and benefits. The main advantage is cost efficiency. The system has the capability to run multiple containers in a shared operating system and network. This is one of the efficient way to use the resources. Moreover, it is easy to use and portable, with the help of the cloud environment the software or the container can run quickly and dependable also help for stable application. Continuous deployment and integration help application to be more reliable which also provide the roll back option after the deployment and integration. High performance containers are easily coupled and deployed for the application and easy to scale the components. Additionally, there are a lot of benefits available such as, managing, monitoring, security, storage and many more.

3. PostgreSQL:

PostgreSQL is the free and open source (RDBMS) relational database management system. PostgreSQL helps you to add different functionalities. It is the advanced version of SQL which provision special functions of SQL such as, subqueries, trigger, foreign key, user defined types and many more. PostgreSQL has the capability to perform difficult queries, data analyzation and data warehousing

3.1 PostgreSQL Architecture:



IV. PostgreSQL Architecture

Postmaster PostgreSQL multi-user database server. Postmaster act as an intermediate for the client to access the main database which helps to separate server process to handle the multi connection. The postmaster manages the entire communication process of the PostgreSQL. The postmaster maintain and manage the data of the PostgreSQL database as per the architecture of the PostgreSQL we can understand that multi users are trying to access the database where the postmaster perform the operation in an efficient way

3.2 Feature and benefit of PostgreSQL:

Main feature of the PostgreSQL relational database management system emphasizing flexibility and SQL compliance. Complex structural data can be easily managed and maintained using PostgreSQL . The main aim of these relational database management system is to support the user to build complex applications more efficient with various features. Additionally, it also helps to create, store, retrieve the users complex dataset which also supports image, video, audio storage and also supports graphical data with high recovery.

The PostgreSQL relational database management system has the combination of libraries, dependencies and combination of applications with large combined databases. The main benefits of the PostgreSQL is very flexible for the user which is designed portably which can be used in different operating system platforms. Additionally, it also has the high availability locking mechanism with the fault tolerance capacity.

4. Storage in Kubernetes:

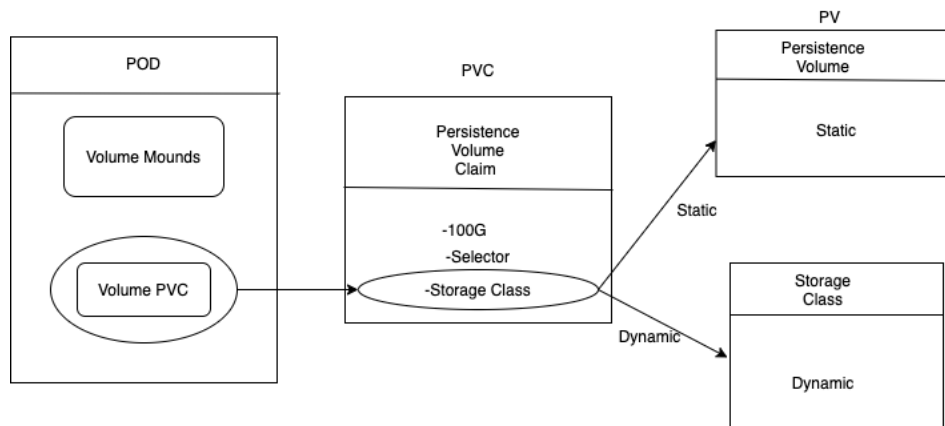
4.1 Why do we focus on Stateful apps?

Because stateless apps are built with YAML files, which can be backed up with git, have an automated deployment with CI/CD, etc. None of the data is saved inside these apps.

Configuration is made in a ConfigMap or Secrets.

Stateful apps on the other hand save data, mostly attached on volumes, and it is these volumes that contain all the information that apps need in order to run properly making it a priority to backup. Stateful applications are required because the data which is generated by the app is persisted, retained, backed up and accessible outside of the particular hosts that run the application. Most of the organization and the many businesses preferred the stateful applications which need minimum PV storage.

From the below diagram, it is clear that the Persistent Volume (PV) or the Storage Class (SC) configuration should be written in YAML file first. If the application needs a limited storage space then (PV) can be selected. On other hand, if the application is in the highly available environment and will need of auto scaling then Dynamic Storage class will comes into play, this SC will generally provide by the Cloud service provider. The Persistent Volume Claim (PVC) can request the resource required.



V. Setting Storage configuration in Kubernetes

4.2 Persistent volume (PV)

In case if we need to deploy the database inside the Kubernetes we need persistence storage. Persistent volume is the storage interface. Persistent volume sub system provides the API for the developer and administrators that abstract the details of storage. Persistent volume is a piece of pre-provision storage inside the Kubernetes cluster. Persistent volume is provision by an administrator. The data inside this volume can exist beyond the lifecycle of an individual pod that uses this Persistent volume

4.3 Persistent volume Claim (PVC)

Persistent volume Claim is a storage request by a developer. Persistent volume Claim is a request for storage which consumes PV resources. For instance, in Kubernetes every pod utilizes node resources such as CPU and memory similarly, Persistent volume Claim utilizes Persistent volume resources such as Developer request for storage of some capacity along with some access mode such as read, write or read only.

4.4 Storage class:

The storage class in Kubernetes stores all the data from the database or the user application. In case, if the pod restarts before setting up the Storage all the changes will be gone so for saving all the changes in the data we have to configure Storage which doesn't depend on the pod lifecycle for data persistence out of the pod. So with the storage we can replace the pod data where the previous pod left with up to date data. Each pod consists of different storages with high consistency. After configuring storage we need to create persistence volume then we can claim them using Persistent volume Claim. The Persistent volume is created for every application which consumes more time and space so to make this process more efficient Kubernetes has the storage class which creates and provisions PV dynamically when every PVC claims it this method can also be automated by YAML file by mentioning the own provisioner internally.

5. WHY backup is important:

Backup the documents and files is one of the mandatory things for all organizations. The saving of important documents and files will be an additional protection in the situation of data loss. It is impossible to predict the data loss which can happen in various ways like system crash, data corruption, hardware failure, man-made disaster, natural disaster, software failure etc, so backup and storing the important data helps to prevent the common data loss.

Backup helps to store the earlier copies of data helps to recover the business in future furthermore, backup helps to provide the security for the confidential data. The backups are happen in the regular basics there are various backups available such as file, folder backup, system backup, application backup, server backup etc.

Each and every organisation has a set of highly confidential data for their business purpose which need to be protected with sequential versions of data. Data versioning is saving the new file or recording the new file over time accordingly with a statement of version to avoid confusion for later use. The main advantage of backup are:

- quick accessibility
- security
- protection.

The accessibility becomes more comfortable in the data loss situation the user or the organization just needs the internet facility to gather the backup data which saves your money and time. The backup can also help to protect the confidential data by the process of encryption.

5.1 Different backup Strategies

There are three sectors of backup strategies available

- Full backup
- Incremental Backup
- Differential backup

Full back up is a commonly used method which is used to back up every copies of all the data in the disk or tape. The main benefit of this full backup strategies is after backup the user will have the entire copy of the data

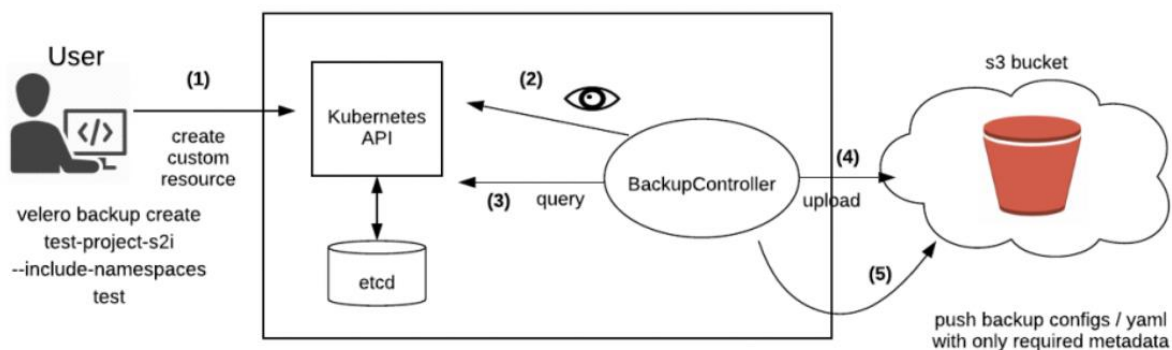
Incremental Backup changes the copy of all files it keeps on updating the data incrementally for an instance, in case there is a full back up on Monday the next day it will store the increment data likewise it will keep on store the data daily which keeps on incrementing. The main advantage of this restoration process can be easily done with the help of these backup strategies.

Differential Backup to overcome the disadvantage of the incremental backup. Where the differential backup can only store the data which made the changes. For instance, in the entire data there is a change in the particular folder so now the differential backup compares the old and new file then it will save the particular folder that changes are made. The advantage of this data is to save storage and time efficiently.

6. Different backup method in Kubernetes:

6.1 Velero:

Velero is one of the backup tools for the Kubernetes cluster. It backups the Kubernetes objects to object storage by compressing it. Additionally , it also takes a snapshot of the cluster's PV using the feature of cloud provider block storage then it will restore the cluster's object and PV where it left off.

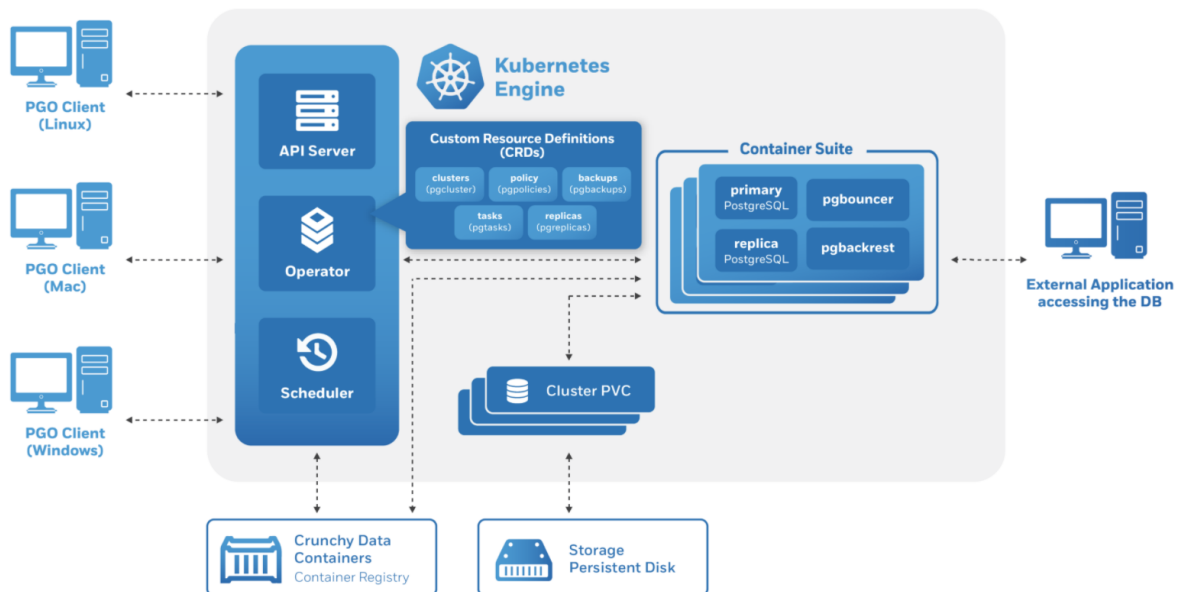


VI. Velero Architecture

Velero also includes controllers that process the custom resources to perform backups, restores, and all related operations. Velero is a disaster recovery tool which mainly snapshotting your application state, prior to performing system operations on your cluster (e.g. upgrades). The Velero has scheduled backup the first backup performed when it is created then it will perform backup in each interval then the restore method allows the user to restore all the object and PV from the backup which was created before.

6.2 Crunchy data operator for PostgreSQL:

The main aim of the Crunchy PostgreSQL Operator is to provide a means to instantly get your applications up and running on PostgreSQL for both progress and production environments with a set of resources. Crunchy data operator for PostgreSQL extends Kubernetes to provide high level abstraction for Creation and management of PostgreSQL cluster and this relay information about the status and health of PostgreSQL cluster



VII. Crunchy PostgreSQL Operator Architecture

There are three main function in the Crunchy PostgreSQL Operator for backup and restore option

- pgBackRest (crunchy-backrest-restore)
- pgdump (crunchy-pgdump)
- crunchy-pgrestore (restore)

pgBackRest is a high performance backup and restore function for PostgreSQL. The crunchy-backrest-restore container executes the pgBackRestfunction, allowing FULL and DELTA restore proficiency. The crunchy-pgdump container executes either a pg_dump or pg_dumpall database

backup against another PostgreSQL database. The restore image provides a means of performing a restore of a dump from pg_dump or pg_dumpall via psql or pg_restore to a PostgreSQL container database.

6.3 Third Party Helm chart:

IBM provides various plugins on the Helm catalogue for Kubernetes cluster. Those packages are supported in IBM cloud Kubernetes services such as,

- ibm-object-storage-plugin with various versions
- ibm-block-storage
- ibm-iks-cluster-autoscaler

We can add those IBM part Helm chart plugins in our Kubernetes system to perform the proper backup and storage system.

ibm-iks-cluster-autoscaler is used to scale up and scale down the worker nodes within the cluster by scaling the worker pools of the cluster. The cluster autoscaler calculates the number of worker nodes that your cluster needs to run its workload. Scaling the cluster up or down depends on many factors

IBM cloud block storage is a plugin for processing the storage with high performance iSCSI storage in that the user can add the apps using Kubernetes PV. The user needs to install IBM cloud block storage to create and launch the apps using Helm chart. Additionally, the chart includes the creation of a pre-defined block storage class.

6.4 Alternative Backup Method:

This is just an alternative backup method for the main design system. We have to make a custom Docker file we will have a base image file (say, Ubuntu) and inside the file the connection string and methods to the master pod.

A backup shell script will be written like connect directly to the database engine itself and ask to run a backup file that is bind mount and that path will be specified in the YAML files that you submitted to Kubernetes. Then in the Kubernetes we have to create the cron jobs to invoke that shell script (which is in that image that has been created from that Docker-file) and create a temporary pod and a Persistent Volume (PV) which will take all the back and stores in that PV. These PV can be pushed to the IBM Cloud Object Storage and maintain all the data, we could also have an option to back up all the resources, according to the namespaces, PV etc.

7. Conclusion:

To conclude, this detailed research document was completely based on the research done for designing a Kubernetes system that takes a backup for a PostgreSQL database within Kubernetes. I have come across various strategies and method to design a flawless Kubernetes back up system. Moreover, also explored and discussed on various backup method for a PostgreSQL DB in a Kubernetes.