ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 8, Issue 5, May 2021

Automated Preserving of the Required Storage Volumes in Cloud Infrastructure

[1] Achala N Gowda, [2] Dr. B.M Sagar

[1][2] Department of Information Science and Engineering, RV College of Engineering®, Bengaluru, Karnataka, India
Email: [1] achalangowda.sit19@rvce.edu.in ,[2] sagarbm@rvce.edu.in

***Abstract*----In Cloud computing, Kubernetes and Docker containers will provide an infrastructure for running the code in the cloud, but managing storage in the cloud infrastructure is a different problem from managing the compute instances. Storage for the containers will be provided in the form of volumes attached to the pods, but the problem is the loss of files when a container or the pod crashes. So, there is a need of using the persistent volumes and the services for managing the persistent volume. The persistent volume manager is a special module of Kubernetes API which creates and manages the persistent volume based on the developer's requirement. Which has been written and detailed in this paper.**

***Keywords*--- Cloud computing, Docker containers, Kubernetes, Persistent volume**

I. INTRODUCTION

In Cloud computing, Kubernetes helps running an application on a cluster (in "the cloud") and allows the developer to focus on the functionality of their Application while ignoring the details of servers and networking, and a cluster is a collection of servers (nodes) managed by Kubernetes, the smallest deployable unit of computing in Kubernetes that you can create and manage is pods. Kubernetes put pods on the node and the Docker runs container inside the pod. Containers are small environments where the application code lives and runs, when a container is started as part of a Kubernetes pod the data created inside such a container will be stored in the volumes.

Containers define the volume mounts which were mapped to the volumes and if the volumes are having confidential information it will map to the secrets and for stateful services, that rely on file persistence will map to the Persistent Volume Claim (PVC) and PVC will refer to persistent volume. So, Kubernetes supports volumes and persistent volumes.

A persistent volume is a part of a cloud cluster where the long-span data will be stored, as volumes are attached to pods, and if the pod got crashed then the volume attached to it will also going to be lost so the data that need to be stored beyond the lifespan of the pods are stored in the persistent volume. Users can request the persistent volume by Persistent Volume Claim.

There are types of persistent volumes for example Network File Storage (NFS), Internet Small Computer Systems Interface (iSCSI), and GlusterFS where each type uses different storage classes where NFS uses File storage service (FSS) storage class which can be either shared or not shared between the pods, iSCSI uses Block-based storage and GlusterFS is open-source software that provides a storage node as a shared file system in the cloud as it is a shared file system it is not scalable when 100's of a file system are required and performance of volumes will get degraded so now it is not been used as a persistent volume.

II. RELATED WORK

In cloud computing for balancing the load on nodes, the workload will be distributed among the different nodes. The researcher of [1] gave a clear-cut idea of how the docker swarm and Kubernetes help in providing an instance to develop applications by balancing the load and how these services help a developer to concentrate on developing the application instead of focusing on networking and load distribution among the nodes of the cloud. Also, in [3], the author gives an in-depth knowledge about Kubernetes as a microservice orchestration, container management tool, and how Kubernetes provides flexibility and scalability for deploying enterprise applications and services in the cloud by taking an example of AWS cloud.

Providing infrastructure for running the code in compute instance is one part where the other important part is storage, Kubernetes will manage the storage in the form of volumes which attaches to the pod where the code runs, [4] gives an in-depth knowledge about the types of storage Kubernetes provide, how the Kubernetes storage works and how the volume management is done in Kubernetes. The researchers of [2] proposed a new storage engine called "Kubestorage" which optimizes the speed of tracking the small files and also optimizes the utilization of storage space by not using metadata to track a file as

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 8, Issue 5, May 2021

metadata itself take a lot of space to compare to the traditional file storage system.

There are different storage classes used by the volumes, some of the most used are Network file storage and Block Volume storage, Gets the complete knowledge of these two storage classes in [11] and [12] respectively. Different storage classes are used based on the requirements of the storage if the single resource to be shared between different pods, then recommended to use file storage service, and if the sharing of the volume is not required and bandwidth should be less than the block volume storage is used as block volumes present inside the network of the cloud and not external storage as file storage service.

The researcher of [6] has reviewed the microservice called Repaver which checks the health of the node and reschedules the node automatically if it is unhealthy and the algorithms of adding the new node and checking the health of the node are explained and the developer of [14] launched rook storage operator that has self-healing capacity.

III. KUBERNETES ARCHITECTURE AND PERSISTENT VOLUME MANAGER

a. Architecture of Kubernetes

Kubernetes is a compartment the executive's innovation created in Google lab to oversee containerized applications in various sort of situations, for example, physical, virtual, and cloud framework. It is an open-source framework that helps in making and overseeing containerization of use.

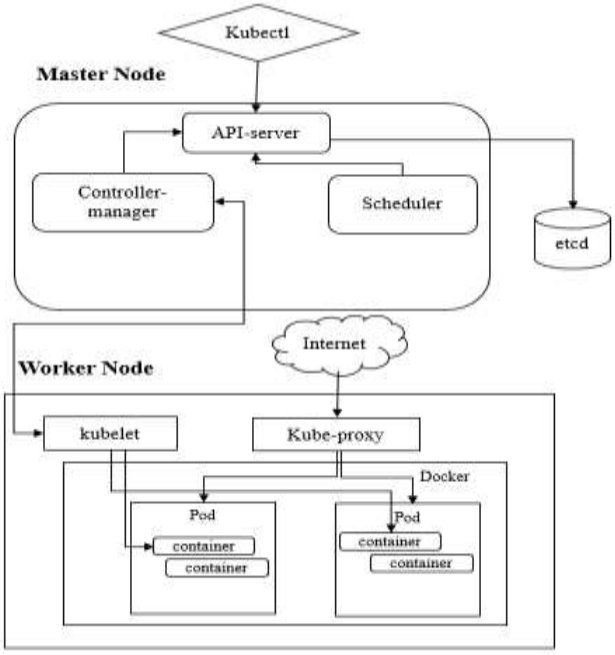

Fig 1: Kubernetes Architecture

On the principles of client-server architecture, Kubernetes is implemented. It is also possible to configure multi-master setups (for high availability), but by default, there is a single master server that acts as both the contact point and the control node. The master/server includes multiple modules, including an API-server, ideal storage, a cloud controller manager, a Kube-Scheduler, a Kube controller manager, and a Kubernetes services DNS server. The node components are made up of kubelet and Kube-proxies on top of the Docker.

Kubernetes architecture involves two different components. I.e., Master & worker components. Each of which is illustrated below.

b. Master Node Component

Master Node involves components such as etcd which is a distributed key-value storage. It can only be accessed from the API server to provide better security. Using watchers, etcd allows all configuration changes to be notified. Notifications are API requests for triggering information changes in a node storage location on each cluster node etcd.

API-Server is the central manager that handles all REST change requests as a front-end to the cluster. Kube Scheduler helps to organize pods for specific resource-based nodes, such as when a group of containers co-located within which our application processes are running.

It also reads the operational requirements of the service and is planned for the best-fit node. For example, if an application requires a gigabyte of physical memory and two CPU cores, for this application instance the pods will be programmed such that the application instance at a node will match these resources at least. The Cloud-Controller Manager is mainly responsible for managing controller processes that are dependent on the underlying cloud service provider. For example, when a controller wants to check and verify whether a node has been terminated or routes set up, the cloud controller manager will handle volumes or load balancers in cloud provider infrastructure.

c. Worker Node Component

Components of worker node involve Kubelet. Kubelet is the key service in every node in the Kubernetes area. It regulates, primarily through the Kube API-Server. It manages the pod specifications to ensure that newly created or modified pods and their containers are running, and health is in the desired state (validated by readiness and liveliness probes) and it ensures they do not run into an unhealthy state.

Kube-proxy is a proxy service that manages user host subnetting and sub-namespace and exposes network services to external networks on all worker nodes. It is also

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 8, Issue 5, May 2021

responsible for requesting the routing of a packet or container across the different isolated networks in a cluster.

d. Persistent Volume manager

The persistent Volume manager module aims to manage the creation and deletion of persistent volume and manages the storage service that stores the persistent volumes. The PV-manager and its storage services will be deployed as services running on the microservice platform. Fig 2 shows the high-level design of the persistent volume manager service.

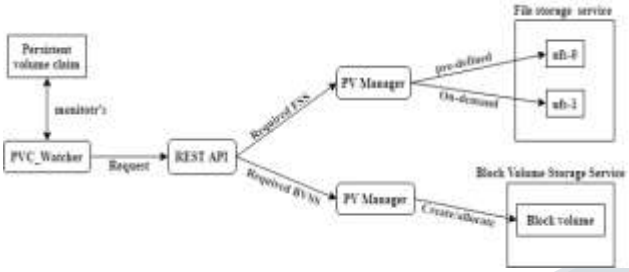

Fig 2: Design of the persistent volume manager service

PVC_watcher monitors the Persistent Volume Claim and if a new Persistent Volume Claim is created by a developer which needs a matching volume, then PVC_watcher sends a POST request to PV-manager REST API for creating the persistent volume as specified in Persistent Volume Claim then REST API informs Persistent Volume Manager to create a new persistent volume. Similarly for Persistent Volume Claim deletion, PVC_watcher requests to delete the persistent volume. There are mainly two types of persistent volume iSCSI and NFS which use different storage classes, one is Block volume storage service and the other is File storage service respectively. Block volume storage is used for high performance as the block volume will be attached to the particular availability domain, so low bandwidth is enough leads to high performance but some of the drawbacks are the need to take backup regularly. File storage service is also known as a highly scalable storage service. FSS storage class has two storage services namely nfs-0 and nfs-1 where nfs-0 supports NFS file system pre-allocated and nfs-1 supports NFS file system on-demand where file system created based on demand.

e. Persistent Volume Claim:

A persistent volume claim is a request by a user for a persistent volume and it is a link between persistent volume and a pod. Kubernetes search a suitable volume based on the persistent volume claim. For a stateful set, each replica will be a pod and each pod will have its persistent volume claim and its persistent volume.

The fields of the persistent volume claim are:

- name: the Kubernetes name of the claim
- namespace: optional namespace of the claim
- StorageClassName: must be a supported storage class
- accessModes: may be shared or exclusive
- storage: size in GB

f. Allocating File storage service as persistent volume

The core technology that implements the file storage service is NFS (v3) and the idea is that a user will create one or more NFS shares per Available Domain and these shares can be mounted by instances. An NFS share is implemented through three cloud abstractions: a file system, a mount target, and an export path. A file system is created in an AD and it can be associated with one or more mount targets (IP addresses). When a file system is associated with a mount target an export path can be specified. The combination of the mount target and the export path is used by clients to access files on the file system. Fig 3 is a Conceptual Diagram of file storage service.

Containers access FSS file systems as network-attached file systems by using an NFSv3 driver which is needed to be able to mount/unmount NFS PVs from pods.

NFS PV Provisioner will allocate and deallocate NFS PVs and cloud resources (ie. file systems, targets, export paths). There are two types of storage services in FSS storage class, nfs-0, and nfs-1. nfs-0 supports NFS file system pre-allocated and nfs-1 supports NFS file system on-demand where file system created based on demand.

For nfs-0, Project specifies NFS PV details in the form of resources in the project's entitlements (e.g., Targets, ADs, sizes) is mentioned by which the file systems, targets, the export path will be created then PV-manager will create the associated NFS PVs. NFS PV Provisioner will allocate the NFS PVs and pod.

For nfs-1, Project specifies storage class quota in project's entitlements NFS PV provisioner will set up the storage class and provision PVs

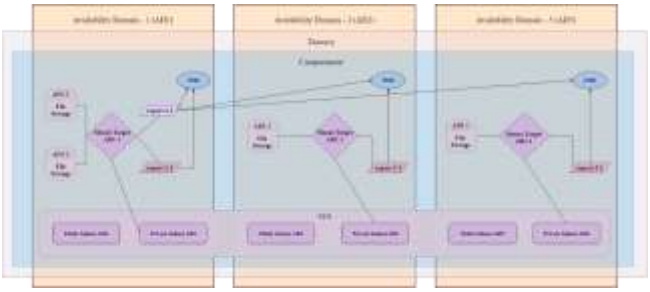

Fig 3: Conceptual Diagram of file storage service

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 8, Issue 5, May 2021

g. Allocating Block volume storage as Persistent volume

Block volume storage is high-performance storage as it is independent of instance shape and volume size and as located inside the cloud network hence uses less bandwidth. All volumes are automatically replicated to protect against data loss. Multiple copies of data are stored redundantly across storage servers with built-in repair mechanisms. Block and boot volumes are designed to provide 99.99 percent uptime.

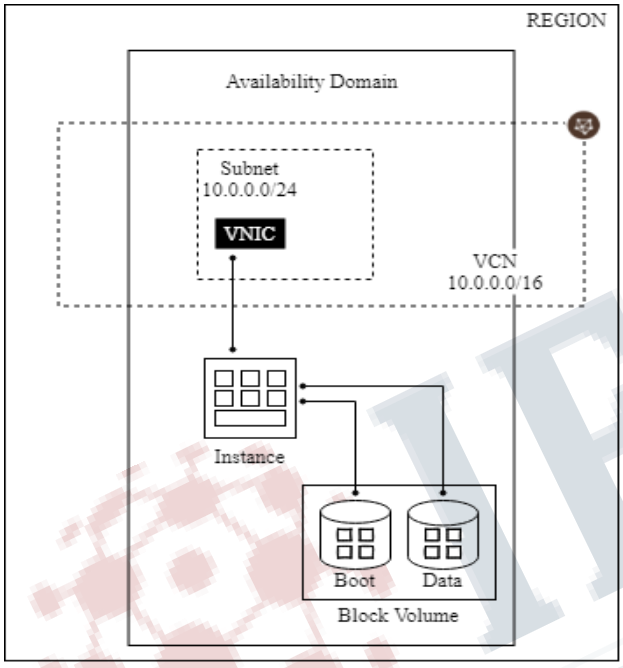

Fig 4: Conceptual Diagram of block volume storage service

Containers would access block volume storage by using a flex volume driver. Flex volume driver provides a robust method for implementing persistent volumes. It is a requirement that pods that use Block Volume Persistent Volume Claims are scheduled on nodes in the same Availability Domain. So, another field is added in PVC which is used to determine the Availability Domain in which the Block Volume will be created. Flex volume driver takes this information and controls the ADs where pods are scheduled so that pods are scheduled to the AD where the persistent volumes are hosted.

The volume provisioner can be used to create and delete persistent volumes (PVs). To create a PV, a persistent volume claim should be defined that uses a storage class linked to the volume provisioner. The provisioner will look for PVC creation requests and will in turn create a matching PV. When the PVC is deleted, the provisioner will delete the bound PV.

Block volumes have a very limited capacity for sharing and are best attached to a single host. While in principle pods running on the same host could share access to a persistent volume mounted on the host, implementing such a model reliably could be impractical. Block volume can only be used if having block volumes quota entitlement.

IV. CONCLUSION

The persistent volume manager module simplifies the task of managing the Persistent volume for pods of a node in cloud infrastructure and it is a service that manages the allocation of the persistent volume for the particular pods and limits the people access to the persistent data to maintains the security of persistent volume data and it also gives the option of dynamic allocation of the persistent volume to the pods if the persistent volume is not pre-allocated in the Persistent Volume Claim. It satisfies the objective of the service by managing the Persistent volume and watching if the Persistent Volume Claim and not allow to delete the Persistent Volume Claim until the pods using those persistent volumes get deleted.

REFERENCES

[1] Nikhil Marathe, Ankita Gandhi and Jaimeel M Shah "Docker Swarm and Kubernetes in Cloud Computing Environment" 2019 ICOEI Third International Conference on Trends in Electronics and Informatics, IEEE Xplore Part Number: CFP19J32-ART; ISBN: 978-1-5386-9439-8

[2] Fuxin Liu, Jingwei Li, Yihong Wang and Lin Li "Kubestorage: A Cloud Native Storage Engine for Massive Small Files" 2019 6th International Conference on Behavioral, Economic and Socio-Cultural Computing.

[3] "The NetApp Guide to Kubernetes" NetApp, Inc. 2020

[4] https://cloud.netapp.com/blog/cvo-blgkubernetes-storage-an-in-depth-look

[5] TME for Data Essentials - DevOps "Persistent Storage for Containerized Applications on Kubernetes with PowerMax SAN Storage" Dell Inc. October 2019

[6] Pramod K and Dr. Ramakanth Kumar P "Automatic Repaving of Unhealthy Nodes in Cloud Infrastructure" 2020 International Journal of Advanced Science and Technology Vol. 29, No. 04, pp. 7046 – 7054

[7] Sandor Acs, Mark Gergely, Peter Kacsuk and Miklos Kozlovszky "Block Level Storage Support for Open Source IaaS Clouds" 2013 21st Euromicro International Conference on Parallel, Distributed, and Network-Based Processing

[8] R. Arokia Paul Rajan, S. Shanmugapriyaa "Evolution

of Cloud Storage as Cloud Computing Infrastructure Service" 2012 IOSR Journal of Computer Engineering (IOSRJCE) ISSN: 2278-0661 Volume 1, PP 38-45

[9] Jiyi WU, Lindi PING, Xiaoping GE, Ya Wang, Jianqing FU "Cloud Storage as the Infrastructure of Cloud Computing" 2010 International Conference on Intelligent Computing and Cognitive Informatics

[10] Xiaoming Gao, Mike Lowe, Yu Ma, and Marlon Pierce "Supporting Cloud Computing with the Virtual Block Store System" 2009 Fifth IEEE International Conference on e-Science

[11] https://docs.oracle.com/enus/iaas/Content/File/Concepts/filestorageoverview.htm

[12] https://docs.oracle.com/enus/iaas/Content/Block/Concepts/overview.htm

[13] https://docs.oracle.com/enus/iaas/Content/ContEng/Tasks/contengcreatingpersistentvolumeclaim.htm

[14] Louis Baumann, Stefan Benz, Leonardo Militano, Thomas Michael Bohnert "Monitoring Resilience in a Rook-managed Containerized Cloud Storage System" 2019 IEEE conference