

Deploying a Kubernetes Cluster with Kubernetes-Operation (kops) on AWS Cloud: Experiments and Lessons Learned



Sindhu G, Pavithra D R

Abstract: In today's world, new virtualization technologies have become popular like container-based virtualization technologies due to its nature like lightweight, scalability, and flexibility. Kubernetes is the most popular open-source platform for container-based cluster management systems that provide mechanisms for deploying, maintaining, and scaling containerized applications across a cluster of hosts. KOPS is a Kubernetes Operations for attempting at simplifying complex installation and maintenance processes of the Kubernetes cluster. Kops helps in deploying highly available Kubernetes cluster, make installation secure, easy, and automatable applications on the AWS cloud. The rapid increase of cloud-native applications has taken cloud computing to the next level, but there are several issues in the deployment of automatable applications, and also complexity increases due to this. Nevertheless, there are still issues to be solved, such as cluster monitoring and cluster deployment automation. The goal of this paper is to deploy a containerized application onto a Kubernetes cluster using KOPS on cloud providers like AWS. This paper discusses how the Kubernetes cluster is deployed on the cloud service provider like AWS cloud using Kops which is the best tool for deploying Kubernetes cluster on AWS and also the Kubernetes services available today on cloud providers like AWS cloud and examines what they offer in comparison with other cloud providers like Google(GKE), Azure(AKE), and also the security issues in AWS cloud provider.

Keywords: Kubernetes, KOPS(Kubernetes operation), AWS cloud, Terraform.

I. INTRODUCTION

The technology is emerging in every field like cloud computing, container-based applications, for example, Kubernetes[1][2], helps in obtaining the computational resources as per the application need or requirements. To satisfy the different requirements or performance in each of these fields like deployment requirement of the application, cloud resources[3]such as virtual machines(VMs), containers can be selected. Thus containers have better performance than virtual machines, like higher utilization of computing resources, simpler deployment of the application. This helps in reducing the management overhead of containers, in turn, helps in the rapid start of new containers. Therefore the container-based virtualization technology is been used in the cloud computing environment.

Kubernetes is one of the most popular open-source container management tools by Google. It is a vendor-agnostic tool. Kubernetes provides a platform for automating application deployment, scaling, flexibility, and operations of containerized applications across clusters.

Kubernetes is one of the most widely used container management tools for containerization applications for the management of a cluster of hosts in a system and controls all the lifecycle operations of the cluster. Kubernetes is not a platform for the containerization, it is a solution for multi-container management. As Kubernetes is a tool for a container management system, it manages Docker containers, and also manages the automated deployment of application and scaling of containers. Kubernetes also provides automatic restarting of failed containers and also rescheduling the containers when their host dies. This increases the availability of the applications. It makes the life of a developer easy by scaling, management of containerized application, and automating the deployment.

KOPS(Kubernetes operation) [4]is a good attempt for simplifying some complex installation and maintenance processes. Kops helps you to create a highly available cluster, for destroying the cluster, upgrade the cluster, and maintain the Kubernetes cluster from the command line interface. AWS(Amazon Web services)[5][10] is now officially supported by cloud services. Kops is a Kubernetes project for managing Kubernetes clusters. Kops is the best tool for deploying Kubernetes clusters to AWS(Amazon web services). Kops has commands for creating Kubernetes clusters, upgrading the cluster setting, validating the cluster, and also applying any changes to the clusters. Kops uses a high-level configuration so that it knows how to apply the infrastructure changes to the existing Kubernetes clusters. It also supports clusters' operational tasks like scaling the nodes up or horizontally scaling the clusters. Kops is used to automating the application deployment[6][8] of Kubernetes on AWS.

Key features of kops(Kubernetes operation) are: it is used for automating the application deployment of clusters to the existing virtual private clouds(VPC) or to create a new VPC. It also supports public topologies and private topologies. It also used to provisions single clusters or master clusters. It also used to the configurable machine for SSH access to individual cluster nodes. It also supports heterogeneous clusters by creating multiple instance groups.

Even though Kubernetes automates container processes, creating the Kubernetes cluster is a huge task. With Kops(Kubernetes operation) setting up a cluster is easy that it works without much difficulty.

Revised Manuscript Received on June 21, 2020.

* Correspondence Author

Sindhu G, MTech*, Networking and Internet Engineering, Electronics and Communication Engineering, JSS Science and Technological University, Mysuru, Karnataka, India. E-mail: sindhuguruswamy04@gmail.com

Pavithra D R, assistant professor, Electronics and Communication Engineering, JSS Science and Technological University, Mysuru, Karnataka, India. E-mail: pavithra@sjce.ac.in

© The Authors. Published by Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP). This is an [open access](http://creativecommons.org/licenses/by-nc-nd/4.0/) article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>)

Even though kops make to create the Kubernetes cluster, some best practices need to ensure to create a cluster in an optimal Kubernetes cluster.

Terraform [9] is a Kubernetes provider's tool for building the Kubernetes cluster, changing the infrastructure efficiently for the Kubernetes cluster by managing the existing service providers. The configuration files describe the components required for terraform that need to run a single application over a cluster. Terraform gives raise to the execution of the plan file that is describing what actions it will take to reach the destination state and executes the plan files to build the infrastructure described. If the configuration changes Terraform can examine what all the changes occurred when it runs a single application over a cluster and create execution plan files that can be applied. The Kubernetes provider requires a cluster to be set up and running before it is been used.

In this paper, a detailed description of how to deploy a Kubernetes cluster in a cloud service like AWS (Amazon web service), and also to run the automating deployment application on a cluster.

The Kubernetes provider Terraform is used to execute the configuration plan files and run it and also look for the cluster up and running. This paper also discusses the disadvantage of using cloud services like AWS and also Kubernetes services available in today's world and also to determine the comparison of AWS with other cloud services.

II. DEPLOYING KUBERNETES CLUSTER ON AWS CLOUD

Kubernetes can run its cluster on the AWS cloud[10] provider. Deploying an application on the Kubernetes cluster requires more effort in a cloud provider which is a private cloud than does for a public cloud. The steps on how to run the Kubernetes cluster on AWS are discussed below. Kops stands for Kubernetes operation. Using kops it is easy to get production-grade, setting up Kubernetes cluster and running. Kops let us create the Kubernetes cluster, deploy the automating application, destroy the Kubernetes cluster and upgrade the cluster, and supported on AWS.

Kops provide high available Kubernetes masters. It can generate Terraform and also support command line auto-completion, API configuration through YAML file and support for kops up and running, ability to add containers and files to nodes via cluster manifest.

Kubeadm is used to deploy the Kubernetes cluster on existing infrastructure. Kubeadm cannot provide the infrastructure which is one of the differences between kubeadm and kops. kubeadm acts as not the only installer and building block. kubeadm setup a feasible cluster on your infrastructure. The advantage of using kubeadm can be used anywhere even in Raspberry pi to set up a cluster.

The difference between kubeadm and kops in the deployment of the cluster is that kubeadm is the middle of the stack and it is used to run on each node of the cluster, and also it basically creates and then listens to Kubernetes API. On the other hand, kops that is responsible for the creation of Kubernetes cluster in the entire lifecycle from the infrastructure provided to upgrade the cluster to deleting the cluster and kops know about everything like nodes,

master, load balancers[7], cloud providers, monitoring, networking, logging of the cluster, etc.

A. Kubernetes on AWS using KOPS

KOPS(Kubernetes operation) maintains the Kubernetes cluster on computing instances and by organizing the containers on the instances to run on the cluster based on the available cluster and resource requirements for each container. Containers running on the compute instance are grouped to form logical groups called pods and also to scale up one or more containers running on the cluster together as a pod. Kubernetes decides when and where to run the pods in the cluster, manages the traffic routing in the network, and also to scale pods based on the utilization that is defined. Kubernetes automatically run the pods on the cluster based on the available computing resource requirements and automatically restarts pods if the computing instance that is running fail. Each pod is given by its IP address and single DNS(domain name server) name, which the Kubernetes cluster uses to connect the user's vices and external traffic.

AWS cloud service provider makes it easy to run the Kubernetes cluster and also to manage Kubernetes infrastructure with Amazon EC2 instance and can obtain self-service automatically provisioned to deploy the Kubernetes cluster and maintains the Kubernetes cluster control plan with AWS EKS that is a managed Kubernetes service. Either of the way this can get integrations to AWS cloud services like VPC,(virtual private cloud), IAM(identity and access management), and service discovery as well as the security, scalability, and high availability of AWS cloud service provider.

Configuration of AWS cloud as follows:

A. IAM user permission:

Kubernetes cluster can be created having the IAM user permissions like:

- AmazonEC2 full Access
- AmazonRoute53 full Access
- AmazonS3 full Access
- IAM full Access
- AmazonVPC full Access

B.Create AmzonS3 bucket for the Kubernetes :

Kops stores the information regarding the configurations of the Kubernetes cluster in-state store. For example like the number of nodes, compute EC2 Instance type of node, version of the Kubernetes cluster. The state is stored during the initial creation of the Kubernetes cluster. Any changes to the cluster are also stored in the state as well.

C. Create an amazon s3 bucket

Create an S3 bucket:

```
aws s3api create-bucket --bucket <bucket name>  
DNS(domain name server) configuration
```

In the KOPS domain and subdomain, it is required to create the Kubernetes cluster.

This domain allows the master node which discovers all the servers where the master node is discovered by worker nodes. It allows kubectl to interact directly with the master.

This domain is registered with AWS by creating a Route53 hosted zone. Specifying the name server(NS) records from the created hosted zone as NS record with the registered domain.

D. Create the Kubernetes cluster using kops

The Kops command-line interface(CLI) creates the highly available Kubernetes cluster, with multiple master nodes across multiple available hosted zones. Worker nodes spread across multiple hosted zones, as shown in Fig 1 phase 1. CLI runs the kops command and creates a cluster. Fig 2 shows the output in the AWS cloud the Kubernetes cluster formed by the master node and the two nodes instances formed on the AWS cloud. Fig 3 shows the hosted zones with the API being created after the Kubernetes cluster formed on the AWS cloud, this API is used to establish communication with the nodes being created. The output will be a folder that contains all the terraform files in it. Convert the code in terraform and store the output code. The following are the tasks for the creation of the Kubernetes cluster.

- Provisioning the compute EC2 instances on AWS
- Setup AWS computing resources like networks, Auto scaling groups, IAM users, and security groups.
- Deploying the Kubernetes cluster.

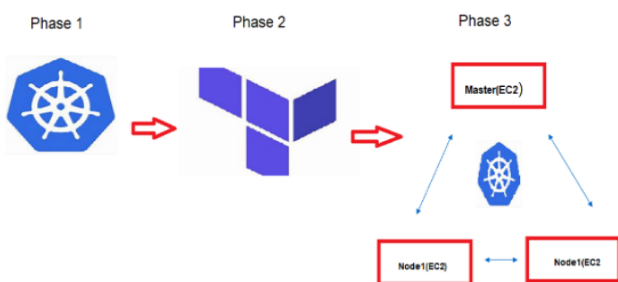


Fig 1: Creating a Kubernetes cluster using kops on AWS

Fig 2 shows the cluster formation on the AWS cloud with three nodes one master node and the other two nodes it shows the running instances of Kubernetes nodes on AWS cloud.

Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS (IPv4)
master-ec2-	i-426a2b64708c0d01f	t2.micro	eu-central-1a	running	2/2 checks	None	ec2-18-184-149-178
nodes.kops-	i-098ba9564da7cda01	t2.micro	eu-central-1a	running	2/2 checks	None	ec2-18-197-167-218
nodes.kops-	i-06340c70198937c	t2.micro	eu-central-1a	running	2/2 checks	None	ec2-18-184-161-43

Fig 2: Figure shows the Kubernetes cluster formation on the AWS cloud

Name	Type	Value	Evaluate Target Health	Health Check ID	TTL	Region	Weight
ns-345.amazonaws.com	NS	ns-345.amazonaws.com	-	-	-	-	-
ns-678.amazonaws.com	NS	ns-678.amazonaws.com	-	-	-	-	-
ns-901.amazonaws.com	NS	ns-901.amazonaws.com	-	-	-	-	-
ns-234.amazonaws.com	NS	ns-234.amazonaws.com	-	-	-	-	-
ns-567.amazonaws.com	NS	ns-567.amazonaws.com	-	-	-	-	-
ns-890.amazonaws.com	NS	ns-890.amazonaws.com	-	-	-	-	-
ns-123.amazonaws.com	NS	ns-123.amazonaws.com	-	-	-	-	-
ns-456.amazonaws.com	NS	ns-456.amazonaws.com	-	-	-	-	-
ns-789.amazonaws.com	NS	ns-789.amazonaws.com	-	-	-	-	-
ns-012.amazonaws.com	NS	ns-012.amazonaws.com	-	-	-	-	-
ns-345.amazonaws.com	NS	ns-345.amazonaws.com	-	-	-	-	-
ns-678.amazonaws.com	NS	ns-678.amazonaws.com	-	-	-	-	-
ns-901.amazonaws.com	NS	ns-901.amazonaws.com	-	-	-	-	-
ns-234.amazonaws.com	NS	ns-234.amazonaws.com	-	-	-	-	-
ns-567.amazonaws.com	NS	ns-567.amazonaws.com	-	-	-	-	-
ns-890.amazonaws.com	NS	ns-890.amazonaws.com	-	-	-	-	-
ns-123.amazonaws.com	NS	ns-123.amazonaws.com	-	-	-	-	-
ns-456.amazonaws.com	NS	ns-456.amazonaws.com	-	-	-	-	-
ns-789.amazonaws.com	NS	ns-789.amazonaws.com	-	-	-	-	-
ns-012.amazonaws.com	NS	ns-012.amazonaws.com	-	-	-	-	-
ns-345.amazonaws.com	NS	ns-345.amazonaws.com	-	-	-	-	-
ns-678.amazonaws.com	NS	ns-678.amazonaws.com	-	-	-	-	-
ns-901.amazonaws.com	NS	ns-901.amazonaws.com	-	-	-	-	-
ns-234.amazonaws.com	NS	ns-234.amazonaws.com	-	-	-	-	-
ns-567.amazonaws.com	NS	ns-567.amazonaws.com	-	-	-	-	-
ns-890.amazonaws.com	NS	ns-890.amazonaws.com	-	-	-	-	-
ns-123.amazonaws.com	NS	ns-123.amazonaws.com	-	-	-	-	-
ns-456.amazonaws.com	NS	ns-456.amazonaws.com	-	-	-	-	-
ns-789.amazonaws.com	NS	ns-789.amazonaws.com	-	-	-	-	-
ns-012.amazonaws.com	NS	ns-012.amazonaws.com	-	-	-	-	-
ns-345.amazonaws.com	NS	ns-345.amazonaws.com	-	-	-	-	-
ns-678.amazonaws.com	NS	ns-678.amazonaws.com	-	-	-	-	-
ns-901.amazonaws.com	NS	ns-901.amazonaws.com	-	-	-	-	-
ns-234.amazonaws.com	NS	ns-234.amazonaws.com	-	-	-	-	-
ns-567.amazonaws.com	NS	ns-567.amazonaws.com	-	-	-	-	-
ns-890.amazonaws.com	NS	ns-890.amazonaws.com	-	-	-	-	-
ns-123.amazonaws.com	NS	ns-123.amazonaws.com	-	-	-	-	-
ns-456.amazonaws.com	NS	ns-456.amazonaws.com	-	-	-	-	-
ns-789.amazonaws.com	NS	ns-789.amazonaws.com	-	-	-	-	-
ns-012.amazonaws.com	NS	ns-012.amazonaws.com	-	-	-	-	-
ns-345.amazonaws.com	NS	ns-345.amazonaws.com	-	-	-	-	-
ns-678.amazonaws.com	NS	ns-678.amazonaws.com	-	-	-	-	-
ns-901.amazonaws.com	NS	ns-901.amazonaws.com	-	-	-	-	-
ns-234.amazonaws.com	NS	ns-234.amazonaws.com	-	-	-	-	-
ns-567.amazonaws.com	NS	ns-567.amazonaws.com	-	-	-	-	-
ns-890.amazonaws.com	NS	ns-890.amazonaws.com	-	-	-	-	-
ns-123.amazonaws.com	NS	ns-123.amazonaws.com	-	-	-	-	-
ns-456.amazonaws.com	NS	ns-456.amazonaws.com	-	-	-	-	-
ns-789.amazonaws.com	NS	ns-789.amazonaws.com	-	-	-	-	-
ns-012.amazonaws.com	NS	ns-012.amazonaws.com	-	-	-	-	-
ns-345.amazonaws.com	NS	ns-345.amazonaws.com	-	-	-	-	-
ns-678.amazonaws.com	NS	ns-678.amazonaws.com	-	-	-	-	-
ns-901.amazonaws.com	NS	ns-901.amazonaws.com	-	-	-	-	-
ns-234.amazonaws.com	NS	ns-234.amazonaws.com	-	-	-	-	-
ns-567.amazonaws.com	NS	ns-567.amazonaws.com	-	-	-	-	-
ns-890.amazonaws.com	NS	ns-890.amazonaws.com	-	-	-	-	-
ns-123.amazonaws.com	NS	ns-123.amazonaws.com	-	-	-	-	-
ns-456.amazonaws.com	NS	ns-456.amazonaws.com	-	-	-	-	-
ns-789.amazonaws.com	NS	ns-789.amazonaws.com	-	-	-	-	-
ns-012.amazonaws.com	NS	ns-012.amazonaws.com	-	-	-	-	-
ns-345.amazonaws.com	NS	ns-345.amazonaws.com	-	-	-	-	-
ns-678.amazonaws.com	NS	ns-678.amazonaws.com	-	-	-	-	-
ns-901.amazonaws.com	NS	ns-901.amazonaws.com	-	-	-	-	-
ns-234.amazonaws.com	NS	ns-234.amazonaws.com	-	-	-	-	-
ns-567.amazonaws.com	NS	ns-567.amazonaws.com	-	-	-	-	-
ns-890.amazonaws.com	NS	ns-890.amazonaws.com	-	-	-	-	-
ns-123.amazonaws.com	NS	ns-123.amazonaws.com	-	-	-	-	-
ns-456.amazonaws.com	NS	ns-456.amazonaws.com	-	-	-	-	-
ns-789.amazonaws.com	NS	ns-789.amazonaws.com	-	-	-	-	-
ns-012.amazonaws.com	NS	ns-012.amazonaws.com	-	-	-	-	-
ns-345.amazonaws.com	NS	ns-345.amazonaws.com	-	-	-	-	-
ns-678.amazonaws.com	NS	ns-678.amazonaws.com	-	-	-	-	-
ns-901.amazonaws.com	NS	ns-901.amazonaws.com	-	-	-	-	-
ns-234.amazonaws.com	NS	ns-234.amazonaws.com	-	-	-	-	-
ns-567.amazonaws.com	NS	ns-567.amazonaws.com	-	-	-	-	-
ns-890.amazonaws.com	NS	ns-890.amazonaws.com	-	-	-	-	-
ns-123.amazonaws.com	NS	ns-123.amazonaws.com	-	-	-	-	-
ns-456.amazonaws.com	NS	ns-456.amazonaws.com	-	-	-	-	-
ns-789.amazonaws.com	NS	ns-789.amazonaws.com	-	-	-	-	-
ns-012.amazonaws.com	NS	ns-012.amazonaws.com	-	-	-	-	-
ns-345.amazonaws.com	NS	ns-345.amazonaws.com	-	-	-	-	-
ns-678.amazonaws.com	NS	ns-678.amazonaws.com	-	-	-	-	-
ns-901.amazonaws.com	NS	ns-901.amazonaws.com	-	-	-	-	-
ns-234.amazonaws.com	NS	ns-234.amazonaws.com	-	-	-	-	-
ns-567.amazonaws.com	NS	ns-567.amazonaws.com	-	-	-	-	-
ns-890.amazonaws.com	NS	ns-890.amazonaws.com	-	-	-	-	-
ns-123.amazonaws.com	NS	ns-123.amazonaws.com	-	-	-	-	-
ns-456.amazonaws.com	NS	ns-456.amazonaws.com	-	-	-	-	-
ns-789.amazonaws.com	NS	ns-789.amazonaws.com	-	-	-	-	-
ns-012.amazonaws.com	NS	ns-012.amazonaws.com	-	-	-	-	-
ns-345.amazonaws.com	NS	ns-345.amazonaws.com	-	-	-	-	-
ns-678.amazonaws.com	NS	ns-678.amazonaws.com	-	-	-	-	-
ns-901.amazonaws.com	NS	ns-901.amazonaws.com	-	-	-	-	-
ns-234.amazonaws.com	NS	ns-234.amazonaws.com	-	-	-	-	-
ns-567.amazonaws.com	NS	ns-567.amazonaws.com	-	-	-	-	-
ns-890.amazonaws.com	NS	ns-890.amazonaws.com	-	-	-	-	-
ns-123.amazonaws.com	NS	ns-123.amazonaws.com	-	-	-	-	-
ns-456.amazonaws.com	NS	ns-456.amazonaws.com	-	-	-	-	-
ns-789.amazonaws.com	NS	ns-789.amazonaws.com	-	-	-	-	-
ns-012.amazonaws.com	NS	ns-012.amazonaws.com	-	-	-	-	-
ns-345.amazonaws.com	NS	ns-345.amazonaws.com	-	-	-	-	-
ns-678.amazonaws.com	NS	ns-678.amazonaws.com	-	-	-	-	-
ns-901.amazonaws.com	NS	ns-901.amazonaws.com	-	-	-	-	-
ns-234.amazonaws.com	NS	ns-234.amazonaws.com	-	-	-	-	-
ns-567.amazonaws.com	NS	ns-567.amazonaws.com	-	-	-	-	-
ns-890.amazonaws.com	NS	ns-890.amazonaws.com	-	-	-	-	-
ns-123.amazonaws.com	NS	ns-123.amazonaws.com	-	-	-	-	-
ns-456.amazonaws.com	NS	ns-456.amazonaws.com	-	-	-	-	-
ns-789.amazonaws.com	NS	ns-789.amazonaws.com	-	-	-	-	-
ns-012.amazonaws.com	NS	ns-012.amazonaws.com	-	-	-	-	-
ns-345.amazonaws.com	NS	ns-345.amazonaws.com	-	-	-	-	-
ns-678.amazonaws.com	NS	ns-678.amazonaws.com	-	-	-	-	-
ns-901.amazonaws.com	NS	ns-901.amazonaws.com	-	-	-	-	-
ns-234.amazonaws.com	NS	ns-234.amazonaws.com	-	-	-	-	-
ns-567.amazonaws.com	NS	ns-567.amazonaws.com	-	-	-	-	-
ns-890.amazonaws.com	NS	ns-890.amazonaws.com	-	-	-	-	-
ns-123.amazonaws.com	NS	ns-123.amazonaws.com	-	-	-	-	-
ns-456.amazonaws.com	NS	ns-456.amazonaws.com	-	-	-	-	-
ns-789.amazonaws.com	NS	ns-789.amazonaws.com	-	-	-	-	-
ns-012.amazonaws.com	NS	ns-012.amazonaws.com	-	-	-	-	-
ns-345.amazonaws.com	NS	ns-345.amazonaws.com	-	-	-	-	-
ns-678.amazonaws.com	NS	ns-678.amazonaws.com	-	-	-	-	-
ns-901.amazonaws.com	NS	ns-901.amazonaws.com	-	-	-	-	-
ns-234.amazonaws.com	NS	ns-234.amazonaws.com	-	-	-	-	-
ns-567.amazonaws.com	NS	ns-567.amazonaws.com	-	-	-	-	-
ns-890.amazonaws.com	NS	ns-890.amazonaws.com	-	-	-	-	-
ns-123.amazonaws.com	NS	ns-123.amazonaws.com	-	-	-	-	-
ns-456.amazonaws.com	NS	ns-456.amazonaws.com	-	-	-	-	-
ns-789.amazonaws.com	NS	ns-789.amazonaws.com	-	-	-	-	-
ns-012.amazonaws.com	NS	ns-012.amazonaws.com	-	-	-	-	-
ns-345.amazonaws.com	NS	ns-345.amazonaws.com	-	-	-	-	-
ns-678.amazonaws.com	NS	ns-678.amazonaws.com	-	-	-	-	-
ns-901.amazonaws.com	NS	ns-901.amazonaws.com	-	-	-	-	-
ns-234.amazonaws.com	NS	ns-234.amazonaws.com	-	-	-	-	-
ns-567.amazonaws.com	NS	ns-567.amazonaws.com	-	-	-	-	-
ns-890.amazonaws.com	NS	ns-890.amazonaws.com	-	-	-	-	-
ns-123.amazonaws.com	NS	ns-123.amazonaws.com	-	-	-	-	-
ns-456.amazonaws.com	NS	ns-456.amazonaws.com	-	-	-	-	-
ns-789.amazonaws.com	NS	ns-789.amazonaws.com	-	-	-	-	-
ns-012.amazonaws.com	NS	ns-012.amazonaws.com	-	-	-	-	-
ns-345.amazonaws.com	NS	ns-345.amazonaws.com	-	-	-	-	-
ns-678.amazonaws.com	NS	ns-678.amazonaws.com	-	-	-	-	-
ns-901.amazonaws.com	NS	ns-901.amazonaws.com	-	-	-	-	-
ns-234.amazonaws.com	NS	ns-234.amazonaws.com	-	-	-	-	-
ns-567.amazonaws.com	NS	ns-567.amazonaws.com	-	-	-	-	-
ns-890.amazonaws.com	NS	ns-890.amazonaws.com	-	-	-	-	-
ns-123.amazonaws.com	NS	ns-123.amazonaws.com	-	-	-	-	-
ns-456.amazonaws.com	NS	ns-456.amazonaws.com	-	-	-	-	-
ns-789.amazonaws.com	NS	ns-789.amazonaws.com	-	-	-	-	-
ns-012.amazonaws.com	NS	ns-012.amazonaws.com	-	-	-	-	-
ns-345.amazonaws.com	NS	ns-345.amazonaws.com	-	-	-	-	-
ns-678.amazonaws.com	NS	ns-678.amazonaws.com	-	-	-	-	-
ns-901.amazonaws.com	NS	ns-901.amazonaws.com	-	-	-	-	-
ns-234.amazonaws.com	NS	ns-234.amazonaws.com	-	-	-	-	-
ns-567.amazonaws.com	NS	ns-567.amazonaws.com	-	-	-	-	-
ns-890.amazonaws.com	NS	ns-890.amazonaws.com	-	-	-	-	-
ns-123.amazonaws.com	NS	ns-123.amazonaws.com	-	-	-		

B. Application deployment on Kubernetes cluster

The Kubernetes deployment is a simple way to create a deployment of any application by using a single command and also it can create a service deployment for the application created in the Kubernetes cluster as shown in Fig 4. The service deployment created can be exposed to the service to the outside world by using the external IP address available for each of the instances created using the service IP and then finally the application deployed in the web browser. Use service as a gateway to communicate with the application and use the port number for communication. The service is special node port Kubernetes exposes some random port and the application can be accessed through the browser by looking at the EC2 instances and check the public IP address of the physical servers where the Kubernetes cluster is up and running.

Perform the following operations to deploy an application into Kubernetes cluster

- The application can be deployed into the Kubernetes cluster by creating a simple YAML file.
- Scale the application to create a highly available application in the Kubernetes cluster.

Deployment is a method of converting the images created into the containers and then allocate those images to pods in the Kubernetes cluster. This helps in setting up the application cluster that includes the deployment of service, pods created, replication controller, and replica set. The Kubernetes cluster can be set up when the applications created are deployed on the pod and the pods can communicate with each other. In this setup of the Kubernetes cluster, there is a load balancer set up on one of the applications deployed on Kubernetes cluster diverting traffic to a set of pods, and then these pods communicate to the backend pods. The communication between pods can be done via the service object built inside the Kubernetes cluster. The application that is created can be deployed and push the code to the cloud service provider like AWS. Then the Kubernetes cluster can ensure that the applications are running all the time to serve our users.

IV. KUBERNETES OPERATIONS CHALLENGES

Kops allows automating the deployment of applications that are highly available Kubernetes clusters on the AWS cloud. It has the features such as automating the deployment of applications which are idempotency, terraform configuration files generation making the kops command easy for the developers.

Most of the developers set up the Kubernetes cluster by using minikube and then move to set up the cluster using Kops. One of the key features of kops is that for the deploying of small production-grade using kops is the best tool. It also includes a larger set of networks that are user flexible for choosing a network stack that is used for use cases. Some of the challenges for the standing up production-grade Kubernetes cluster are:

A. Complexity:

For the production-grade deployment of the applications, all the enterprises have a different need for computing

resources like networking systems, and storage systems. Consider networking as an example. Kubernetes cluster can be deployed on AWS cloud with different networking configuration as follows:

Running all Kubernetes cluster nodes on a public VPC

Kubernetes service has access to the public internet by running all the cluster nodes on a private subnet.

Direct internet access from AWS cloud is prevented by running a cluster on the AWS cloud VPC and then connect to the data center on AWS VPC.

B. Deployment Speed

The complexities in the Kubernetes operation take more time and effort for setting up and running the Kubernetes cluster. The team should know the Kops configuration, AWS cloud architecture, and Kubernetes cluster management is necessary. All these will show down the organization if the team whose primary focus is on the application development and setting up the Kubernetes cluster.

C. Supportability

Kubernetes cluster is created using the Kops tool which is the best tool for setting up the Kubernetes cluster. The operations and maintenance required for the Kubernetes cluster setup require more effort and also a separate team. There are many tools available in open source Kubernetes cluster setup and reliability but knowing all the tools well and understanding them and choosing the right tool can be tedious.

V. COMPARISON OF KUBERNETES SERVICES ACROSS DIFFERENT CLOUD PROVIDERS(AWS, GKE, AZURE)

Kubernetes is adaptable, extensible, and modular that it is deployed on-prem, in most of the cloud providers, and also in the third-party data center.

A. Google GKE(Google Kubernetes Engine)

Google is more influence on Kubernetes that is used as the container management and orchestration solution of Google and Kubernetes came from Google.

Much of the testing of new features and capability of Kubernetes happens on Google cloud(GKE). The GKE cloud need not pay for the Kubernetes control plane Google itself pays for the worker nodes. GKE provides a fully automatic upgrade for the cluster. GKE support kubectl CLI command-line utility. For resource monitoring, manual steps are required using a stack driver. It also provides autoscaling of nodes. Google provides high availability of clusters.

B. Microsoft Azure AKS(Azure Kubernetes Service)

AKS also manages Kubernetes clusters free of cost it is also similar to GKE. There are strong authentication and authorization, monitoring, and Azure storage system with Active Directory, and also a built-in collection of the repository to store container images, and also Graphic processing unit(GPU) enabled nodes.



In ACI there is no need for node provisioning in the cluster, which creates a burden when dealing with highly variable load. It also uses the kubectl command-line utility. An automatic upgrade is done only on demand, master, and nodes. Azure monitor resources for containers and applications on Insight. High availability of clusters is supported on Azure.

C. Amazon AWS EKS(Elastic Kubernetes Service)

Amazon already has its own ECS(Elastic Container Service), but the customers it lately supports Kubernetes service. Nowadays many organizations are deploying their Kubernetes cluster on EC2 using Kops. AWS now provides official interaction with EKS today with IAM for identity management, AWS load balancers networking, and various storage options. For managing the Kubernetes control plane EKS needs to pay. AWS providers can automatically upgrade the demand of the customers by manually updating the nodes and also using the command line interface. It also supports kubectl CLI command-line utility and supports the auto-scaling of nodes.

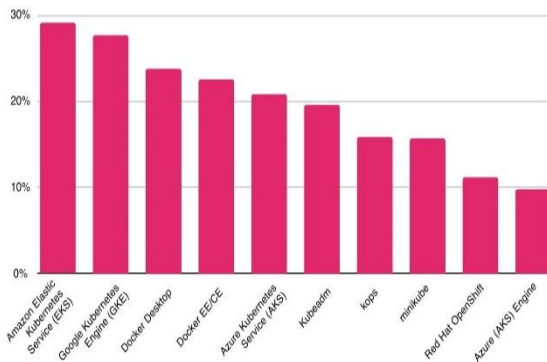


Fig 5. Comparison of Kubernetes services across different cloud providers(AWS, GKE, Azure) source: CNCF Survey 2019[12]

Kubernetes is used as a tool for container orchestration systems today. The biggest question lies is that where should the Kubernetes cluster run and deploy the application on these different cloud service providers. The answer is not yet clear but it can be done by migrating the system to Kubernetes to the cloud service providers on which the cluster is running the Kubernetes else to stick with the same service on which the Kubernetes is running. Fig 5 is taken from source CNCF Survey 2019[12] shows the chart breakdown of the Kubernetes service used by different cloud service providers and it shows how Kubernetes is widely used on service providers EKS followed by GKE.

VI. SECURING KUBERNETES CLUSTER ON AWS

Securing the application is the most essential part of the application deployment framework. Nowadays all the companies are adopting microservice architecture because of its benefits. However, security is one of the main concerns. Kubernetes is growing as the actual container orchestration management tool. But still, the Kubernetes secrets are not as secure as they are. Kubernetes can be easily decoded and also has less place to store sensitive credentials. If the Kubernetes cluster that is deployed on the

AWS cloud service provider is running here are some things that need to be considered:

A. AWS IAM roles

IAM(Identity and Access Management) is used to manage access and identity of the user to the AWS cloud service provider within the organization. These IAM roles can be used to set permissions to the user to access the AWS cloud. These IAM roles allow users to change the password frequently and are provided with temporary credentials for a certain role. Based on the requirements IAM roles are assigned to each of the nodes in the Kubernetes cluster. The roles are applied to the Kubernetes node, where each node is composed of several pods. If the IAM role is applied to the node all the pods associated with that node will inherit the permissions and privileges of that node. Then all the pods associated with that node will get access to the services which they shouldn't be allowed to have access permission. This is not effective from a security point of view and dangerous if there are third part pods in the cluster.

B. Least privilege principle for Kubernetes on AWS

According to this principle, every pod in a Kubernetes cluster does not have unnecessary permissions that can lead to attacks. Pods can have access only to perform a particular task. Pods that have access to the key can pose the risk of developers accidentally logging these keys to avoid secrets, leaving them exposed.

C.IAM controller on AWS for Kubernetes

AWS IAM roles are being distributed by the Kubernetes controller via the Kubernetes' secret resources, as these resources are mounted on the pods as files. Then the SDK in AWS is aware of the file being mounted and points to the file using the shared credential variable(AWS_SHARED_CREDENTIALS_FILE).

D.EKS approach

The EKS approach for the IAM roles as made pods in AWS request to EC2 metadata API to retrieve temporary credentials. AWS changes the AWS identity APIs to make Kubernetes pods to recognize.

E. Balancing security and agility

The security and agility are the two security issues that the organization need to constantly innovate things that treat security issues as an afterthought. In addition to this service mesh to Kubernetes cluster may seem to add its complexity by making important business logic more visible. The service also provides more visibility for microservices interactions to the operation team.

With the complexity in the cloud-based deployments, security is not considered in the Kubernetes cluster. Kubernetes is an agentless API based command-line tool for the automating deployment of applications, creating od clusters, nodes, and pods.



VII. CONCLUSION

This paper presents the deployment of the Kubernetes cluster on the AWS cloud using Kops which is the best tool for deployment of Kubernetes cluster on AWS. The evaluation of deployment of the application on cloud shows that certain circumstances in the AWS cloud need to overcome like security issues, networking issues, and also the IAM issues in the AWS cloud. Further, as future goals, the aim is to enhance the functionality of the services and offer it to the Kubernetes service of its own and also overcome the networking problems and security issues, improve the cluster resource in a Kubernetes cluster.

REFERENCES

1. "Kubernetes," Kubernetes. [Online]. Available: <https://kubernetes.io/>. [Acc.: 24-Jan-18].
2. "Kubernetes Documentation," Kubernetes. [Online]. Available: <https://kubernetes.io/docs/home/>. [Acc.: 23-Jan-18].
3. C. Chang, S. Yang, E. Yeh, P. Lin, and J. Jeng, "A Kubernetes-based monitoring platform for dynamic cloud resource provisioning," in 2017 IEEE Global Communications Conference, GLOBECOM 2017, Singapore, December 4-8, 2017, pp. 1-6, 2017
4. Installing Kubernetes with kops [Online] <https://kubernetes.io/docs/setup/production-environment/tools/kops/>
5. Kubernetes and AWS [online] <https://aws.amazon.com/kubernetes/>
6. S. Brunner, M. Blochlinger, G. Toffetti, J. Spillner, and T. M. Bohnert, Experimental Evaluation of the Cloud-Native Application Design, in Proceedings - 2015 IEEE/ACM 8th International Conference on Utility and Cloud Computing, UCC 2015, 2015, pp. 488493.
7. K. Takahashi, K. Aida, T. Tanjo, and J. Sun, "A portable load balancer for Kubernetes cluster," in Proceedings of the International Conference on High-Performance Computing in Asia-Pacific Region, HPC Asia 2018, Chiyoda, Tokyo, Japan, January 28-31, 2018, pp. 222-231, 2018.
8. J. Xu, P. Chen, L. Yang, F. Meng, and P. Wang, LogDC: Problem Diagnosis for Declaratively-Deployed Cloud Applications with Log, in Proceedings - 14th IEEE International Conference on EBusiness Engineering, ICEBE 2017 - Including 13th Workshop on Service-Oriented Applications, Integration, and Collaboration, SOAIC 2017, pp. 282287.
9. Terraform [Online] <https://www.terraform.io/intro/index.html>
10. AWScloud [Online] <https://docs.aws.amazon.com/eks/latest/userguide/eks-ug.pdf>
11. NGINX Inc. 2017. NGINX Ingress Controller. Retrieved July 14, 2017, from <https://github.com/nginxinc/kubernetes-ingress>
12. Comparison of Kubernetes services across different cloud providers (AWS, GKE, Azure) [online] <https://blog.alcide.io/kubernetes-as-a-service-eks-vs.-aks-vs.-gke>

AUTHORS PROFILE



Sindhu G has received her Bachelor of Engineering in Computer Science and Engineering from GSSS Institute of Engineering and Technology for Women, Mysuru, affiliated to Visvesvaraya Technological University, Belagavi, Karnataka in the year 2018 and now pursuing her M.Tech in Networking and Internet Engineering, Dept Electronics and Communication from Sri Jayachamarajendra College of Engineering Mysuru, affiliated to Visvesvaraya Technological University, Belagavi, Karnataka. Her interest includes the Internet of things, DevOps, Docker, and Kubernetes. Email: sindhuguruswamy04@gmail.com

Pavithra D R has received her Bachelor of Engineering in Electronics and Communication from Sri Jayachamarajendra College of Engineering, Mysuru, affiliated to Visvesvaraya Technological University, Belagavi, Karnataka in the year 2004, and M.Tech in the area of Computer Networks and Engineering from National Institute of Engineering, Mysuru affiliated to Visvesvaraya Technological University, Belagavi, Karnataka in the year 2010 and now pursuing her Ph.D. in the area of digital image signal processing from Visvesvaraya Technological University, Belagavi. Her area of interest includes Signal Processing.

