JOINT INSTITUTE FOR NUCLEAR RESEARCH
Laboratory of Information Technologies

FINAL REPORT ON THE SUMMER STUDENT PROGRAM

Kubernetes cluster deployment and user documentation

Supervisor:
Oksana Ivanovna Strelstova

Student:
David Stromp, Slovakia
Stredna priemyselna skola elektrotechnicka, Kosice

Participation period:
July 11 – September 05

Dubna, 2017

Table of Contents

Abstract

In this report I would like to present my work I did during 2019 Summer school in JINR Dubna. My main task was to build a Kubernetes[1] cluster and create Role Based Access Control (RBAC)[2] policies for user authorization and authentication through TLS certificates[3], and to create instructions for users, as to how to apply for access to the kubernetes cluster. Apart from that I worked on some other smaller tasks, mostly updating configuration files of EOS and CVMFS, and also installing new virtual machines.

Introduction

During my stay in Dubna I came across and learned at least a bit from a lot of new technologies and administration tools: including oVirt management, some Linux administration commands as well as Slurm administration commands, but most importantly Kubernetes. Kubernetes is a cluster of physical or virtual computers created for easier and simpler management of docker-containers (docker-orchestration). My main tasks was to figure out and build RBAC policies for authorization with Authorization by TLS certificates for the Scalable Adaptive Large Structures Analysis (SALSA) project. The reason why containers are being used for SALSA project instead of virtual computers, is that containers are much more lightweight as they only need a kernel and nothing more, they are much easier to scale. Multiple instances of a single application can be launched at any second. To further ease the management of containers, Kubernetes wraps the containers into pods. These pods are much easier to work with than with sole containers. This can create a lot of containers at a time that they may become hard to manage. For these kind of purposes, docker-orchestration softwares were created, such as Kubernetes.

Managing authentication

My first task regarding the kubernetes cluster was, how to achieve authentication through TLS certification. The process has to be initiated by the user. He has to create a private key, a Certificate Signing Request (CSR) with certain information, and after that he has to encode his CSR with a base64 utility, prepare a short CSR-request.yaml configuration file and send it to the administrator. The administrator then has to approve or reject the CSR based on the information provided by the user. If approved, The administrator has to extract the certificate, also he has to extract the kubernetes cluster certificate-authority and send them to the user. Using the private key, his certificate and the certificate of the cluster, and some other information, the user has to configure a file called Kubeconfig[4]. The Kubecontrol (kubectl) utility[5] is used to input commands into the kubernetes cluster. The Kubeconfig file is the configuration file that kubecontrol uses, to locate the kubernetes cluster and to know as which user to logon to the cluster. After that the user should have access to the cluster, it no mistakes were made.

Managing authorization

My seconds task regarding Kubernetes cluster was to set up RBAC policies for authorization. We already had the authentication faze completed, so people were able to access the cluster, but because RBAC policies were not set, they were unable to launch a new pod or even list existing ones. To set up RBAC policies on the cluster, it was necessary to create 2 files on the cluster, 1 to define the role that will exist on the cluster, and the second to create Role Bindings , to bind the created role to a user. Eventually we managed to bind the role to a group and not a user. After that step the user was able to access resources on the cluster for which he has permissions as stated in the file describing the role and the access that role has. We also realized that instead assigning the same role to multiple users, having to change the role bindings all the time or create multiple of them, it would be much simpler to create groups and bind those roles to the groups. After a time I figured out that in order for users to be assigned to the groups already created, they have to include the name of the group during the creation of the CSR.

```
apiVersion: v1
clusters:
- cluster:
    certificate-authority-data: DATA+OMITTED
    server: https://10.0.36.79:6443
  name: kubernetes
contexts:
- context:
    cluster: kubernetes
    user: kubernetes-admin
  name: kubernetes-admin@kubernetes
current-context: kubernetes-admin@kubernetes
kind: Config
preferences: {}
users:
- name: kubernetes-admin
  user:
    client-certificate-data: REDACTED
    client-key-data: REDACTED
```

Figure 1: a Preview of a KubeConfig file. As we can see the certificates and private keys are hidden for security purposes.

Automation and problems with distribution

After finishing these 2 task we realized that, it should be much better if we automatized this process with bash scripts. In order for that to work, it was necessary to create 3 scripts. The first one for creation of the private key , CSR , and .yaml configuration file. The second one to approve or deny the certificates, and to extract the user certificate and cluster certificate. The third one to configure the kubeconfig file. Now the problem was the movement of those files. How to achieve automation with somehow a user sending the .yaml file and the administrator sending back the certificates. The first idea was to use SecureCopy (SCP) from inside the administration script. Launching the script would result in copying the CSR from the user and sending back the certificate. There was drawback when I realized that the root has no access into the home folders of the users. Nevertheless after discussions I was told that I should use a web mail for that. That proved to be much more difficult. I asked for help one of the administrators Yura Butenko to set me up a mail for kubernetes. After that was done I began configuring the /etc/mail.rc configuration files and other stuff. Now there was a huge setback, because we now had to make an automatic download of attachments from e-mail. That part is still unsolved. And I hope that I will be able to finish it in the coming days.

Acknowledgment

I would like to thank Oksana Ivanovna Streltsova for giving me the opportunity to participate in the JINR summer school program.

I would like to thank Yura Alexandrovich Butenko for helping me with when it was needed.

I would like to thank Martin Vala, who motivated me and inspired me to work on this project.

I would like to thank Elena Karpova for organizing my stay in Dubna.

Last but not least I would like to thank everyone who showed us support and made it possible to be here and learn something new.

1. Kubernetes cluster official page
[https://kubernetes.io/]

2. RBAC policies for kubernetes cluster
[https://kubernetes.io/docs/reference/access-authn-authz/rbac/]

3. managing TLS certificates with kubernetes cluster
[https://kubernetes.io/docs/tasks/tls/managing-tls-in-a-cluster/]

4. kubeconfig file for Kubernetes
[https://kubernetes.io/docs/concepts/configuration/organize-cluster-access-kubeconfig/]

5. Kubecontrol utility
[https://kubernetes.io/docs/reference/kubectl/overview/]