Configuration Management tools - Lab Assignment 2

This week we will focus a lot on creating flows what will bring software into a production environment. We have in previous lab set up a kubernetes cluster that we will leverage in this assignment. So if you haven't finished the previous lab, then go back and do so.

Written material should be available by the deadline and to get practical parts approved will be done by showing me what you have done, either during workshop or schedule an online session with me by sending an email to max@scaleoutsystems.com

Task 1 : Describe the following

- Continuous Integration vs. Continuous Delivery
- Jenkins vs. Gitlab
- SDN (Software Defined Networks)
- OpenFlow
- Control Plane vs. Data Plane
- Microservices
    - Api Gateway
    - What a Service is.
    - Advantages of Microservices
    - Limitations/Pitfalls of Microservices
    - Service Discovery
- Inter-Service Communication
    - REST
    - RPI / RPC
    - gRPC
    - Message Queues (alt. Service Bus)
    - AMQP
    - RabbitMQ & Kafka
- Storage
    - Openstack Cinder vs. GlusterFs
    - Openstack Swift vs. S3 Storage
    - Block storage vs. File Storage vs. Object Storage
- Kubernetes
    - RBAC
    - Persistent Volumes (PV) vs. Persistent Volume Claims (PVC)
    - Storage Classes
    - Ingress Controller

Task 2: Making some upgrades.
Subtask 1:

Your cluster is probably missing some key component for us to really utilize it in production just yet. And by now you might have an idea of what that might be. Our cluster needs a way to provision persistent volume claims against our infrastructure. Due to some infrastructure limitations we will be setting up our dynamic provisioning with NFS. Begin by setting up an instance to serve NFS. Then configure the provisioner as explained by the following links.

- https://github.com/yhl17cmt/coursematerial/blob/master/Chapter2/Resourses/NFS.md
- https://github.com/helm/charts/tree/master/stable/nfs-client-provisioner
- Hints
    - https://github.com/yhl17cmt/coursematerial/blob/master/Chapter2/Resourses/nfs-setup-example.sh
    - https://github.com/yhl17cmt/coursematerial/blob/master/Chapter2/Resourses/test-claim.yaml

Subtask 2:

Some web protocols needs an ingress controller to redirect traffic to respective services. Configure an ingress controller so that traffic can flow between 2 or more services of your choice (you may use the services we configured in previous assignment). Some guidance to setting up an ingress controller with Traefik can be found in the repo, but you are welcome to set it up with nginx as well.

- Hints
    - https://github.com/yhl17cmt/coursematerial/blob/master/Chapter2/Resourses/haproxy.cfg
    - https://github.com/yhl17cmt/coursematerial/blob/master/Chapter2/Resourses/traefik-ds.yaml
    - https://github.com/yhl17cmt/coursematerial/blob/master/Chapter2/Resourses/traefik-examplesetup.sh

Subtask 3: Answer the following question; How does an Api-gateway pattern relate to an ingress controller?

Task 3: Setting up Continuous Integration / Continuous Deployment. In this task we will use the kubernetes cluster we have set up in the previous lab to deploy applications to production. Two of the most popular solutions to do this is Jenkins or Gitlab CI, both very capable solutions for our purposes. You may choose to implement your pipeline in either, but we will focus on Gitlab.

Jenkins Approach. If you choose to go the jenkins approach instead on gitlab, then I recommend provisioning the blue ocean distribution of jenkins where you can edit your pipeline visually (more information: https://jenkins.io/projects/blueocean/, https://hub.docker.com/r/jenkinsci/blueocean)

Gitlab approach. You may choose if you want to leverage Gitlab.com or provision Gitlab CE yourself.

- Add your cluster to your project : https://gitlab.com/help/user/project/clusters/index.md#adding-an-existing-kubernetes-cluster
- Hint: You might have to make a service account for gitlab for it to be able to work.

Make sure your pipeline contains at least the following stages: Build, Test, Deploy

Task 4: Setup Minio object storage with helm charts.

Task 5: Write a short essay min. 1 pages (11 pt, Arial or Times New Roman, reference system of choice) about **Message Queues** including, but not limited to:

- A compareasand of rabbitmq and ZMQ (ZeroMq)
- amqp