**Kubernetes**

Basic objectives:   
1. Get to know Kubernetes basics.

2. Explore more deeply such features as automated rollouts and rollbacks, automatic bin packing, self - healing. load balancing, horizontal pod scaling and helm.

3. Find a project which we could manipulate and change to exercise these features and get hands-on experience with Kubernetes.

Resources used:

1. Kubernetes.io
2. Educative.io
3. Udemy.com
4. Github.com

Tool used to manipulate Kubernetes cluster:

1. Kubectl - command line interface for running commands against kubernetes cluster.
2. Minikube – single – node kubernetes cluster on PC.

Most common used Kubectl commands:

1. **Context and configuration:**   
   kubectl config view  
   kubectl config get-contexts  
   kubectl config current-context  
   kubectl config set-context
2. **Creating objects:**  
   kubectl apply –f  
   kubectl create deployment
3. **Viewing and finding resources:**  
   kubectl get services  
   kubectl get deployment  
   kubectl get pods  
   kubectl get configmap  
   kubectl get nodes   
   kubectl get events
4. **Updating resources:**  
   kubectl set  
   kubectl rollout history  
   kubectl rollout undo  
   kubectl rollout status  
   kubectl replace
5. **Deleteing resources:**  
   kubectl delete  
   kubectl delete pods,services
6. **Interacting with running pods:**  
   kubectl logs my-pod  
   kubectl run  
   kubectl attach
7. **Interacting with nodes and clusters:**  
   kubectl top node  
   kubectl cluster-info

Features explored in-depth:

1. Automated rollouts and rollbacks – Explored by creating a number of pods and continuously printing their current version. As soon as we roll an update, we can see in real time how the version of these pods start changing to the new version.
2. Secrets – Created .yml file with a new secret. We can see how the information provided as a secret gets encoded and we can decode it back. The secret is deployed in a pod and can be used inside it. Sensitive information can be safely stored this way.
3. Self-healing – automatic feature of Kubernetes. Explored by creating a big number of pods and making one of them crash, we can observe that Kubernetes quickly creates another one to replace it.
4. Load balancing
5. Horizontal pod scaling
6. Helm