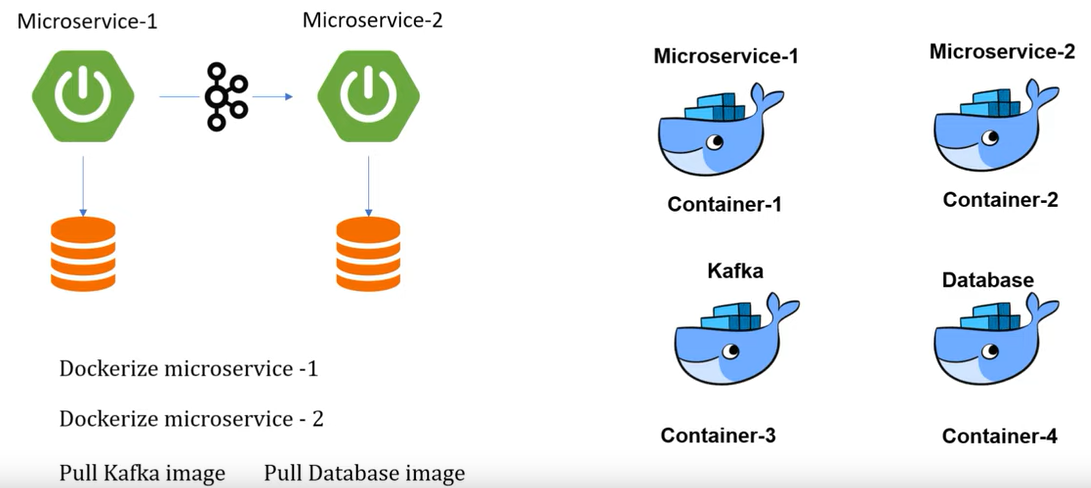
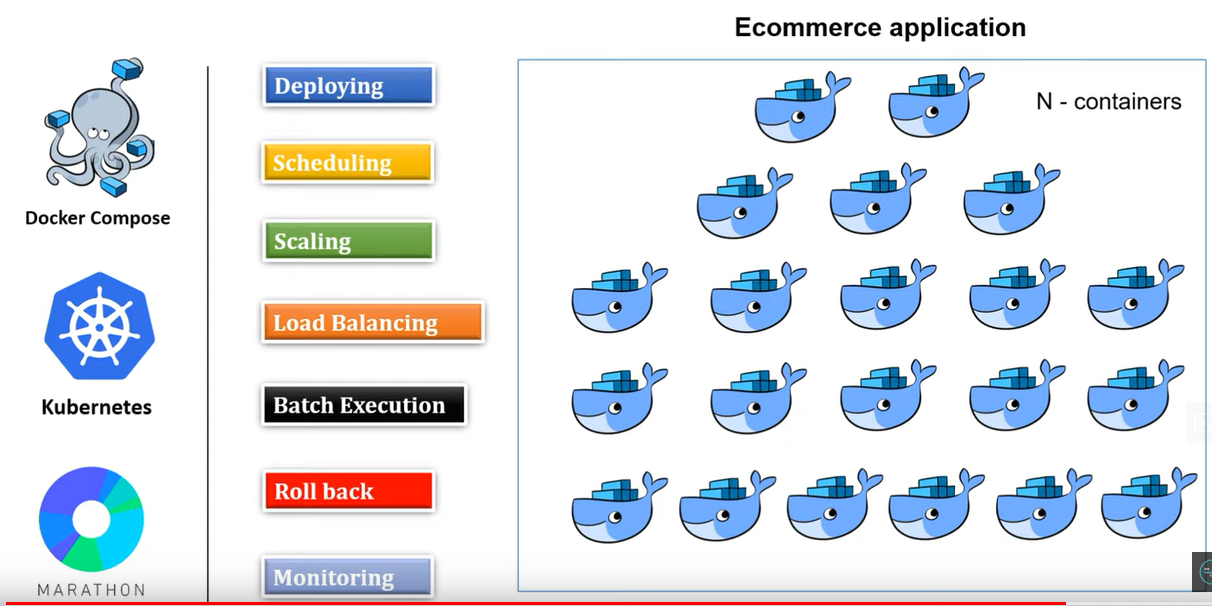
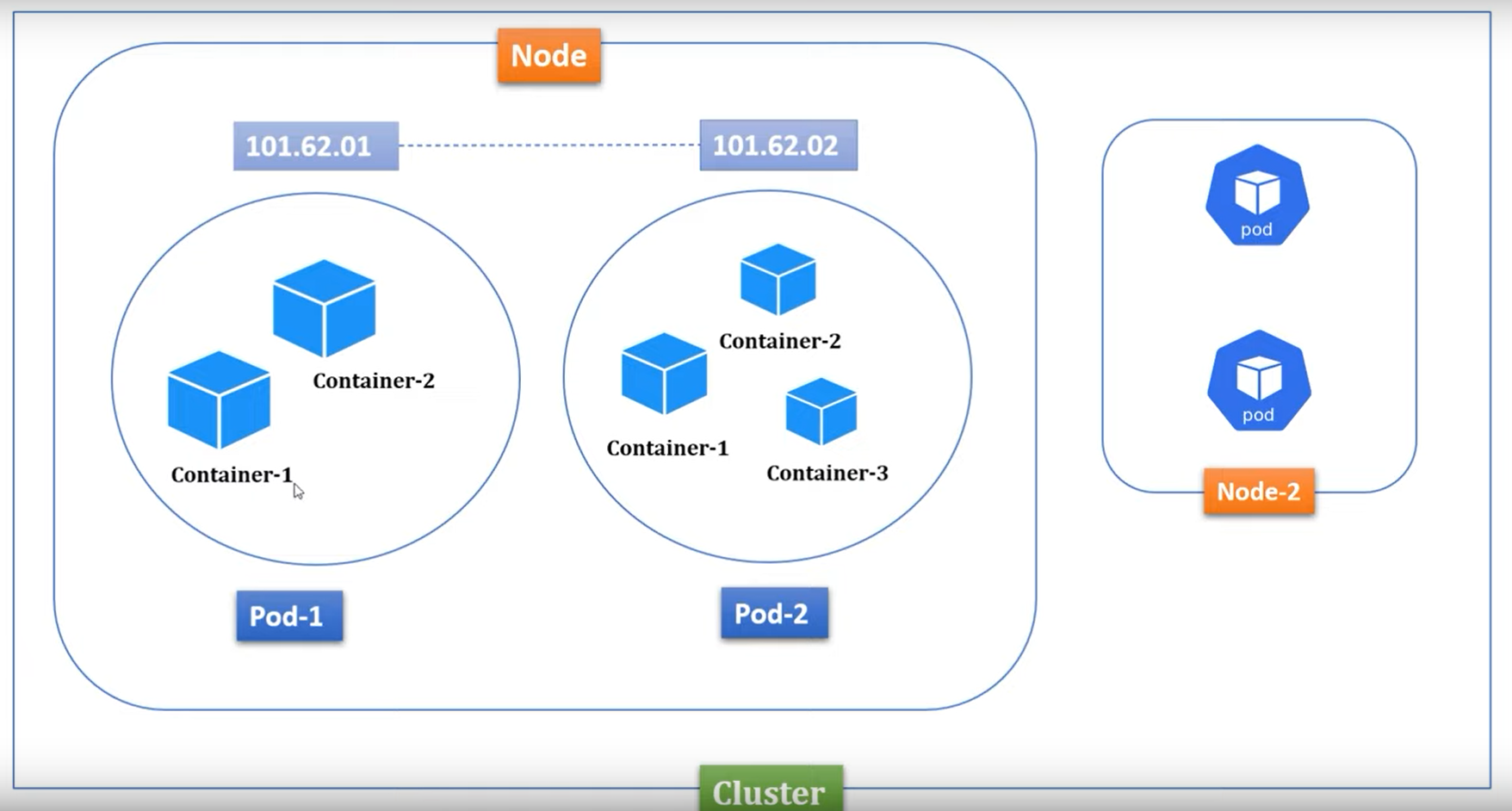
## What and Why Kubernetes?

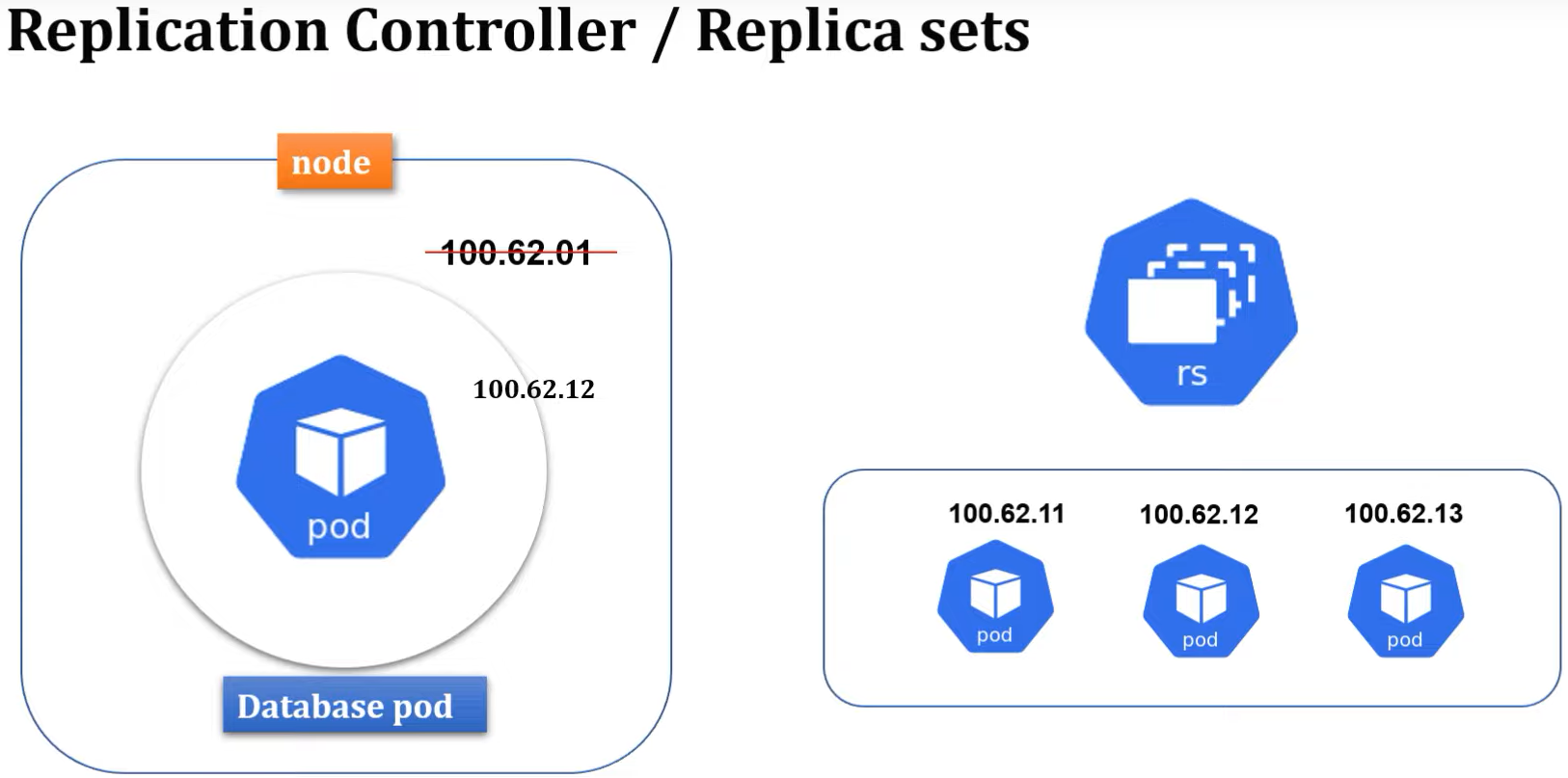
Kubernetes is container management tool

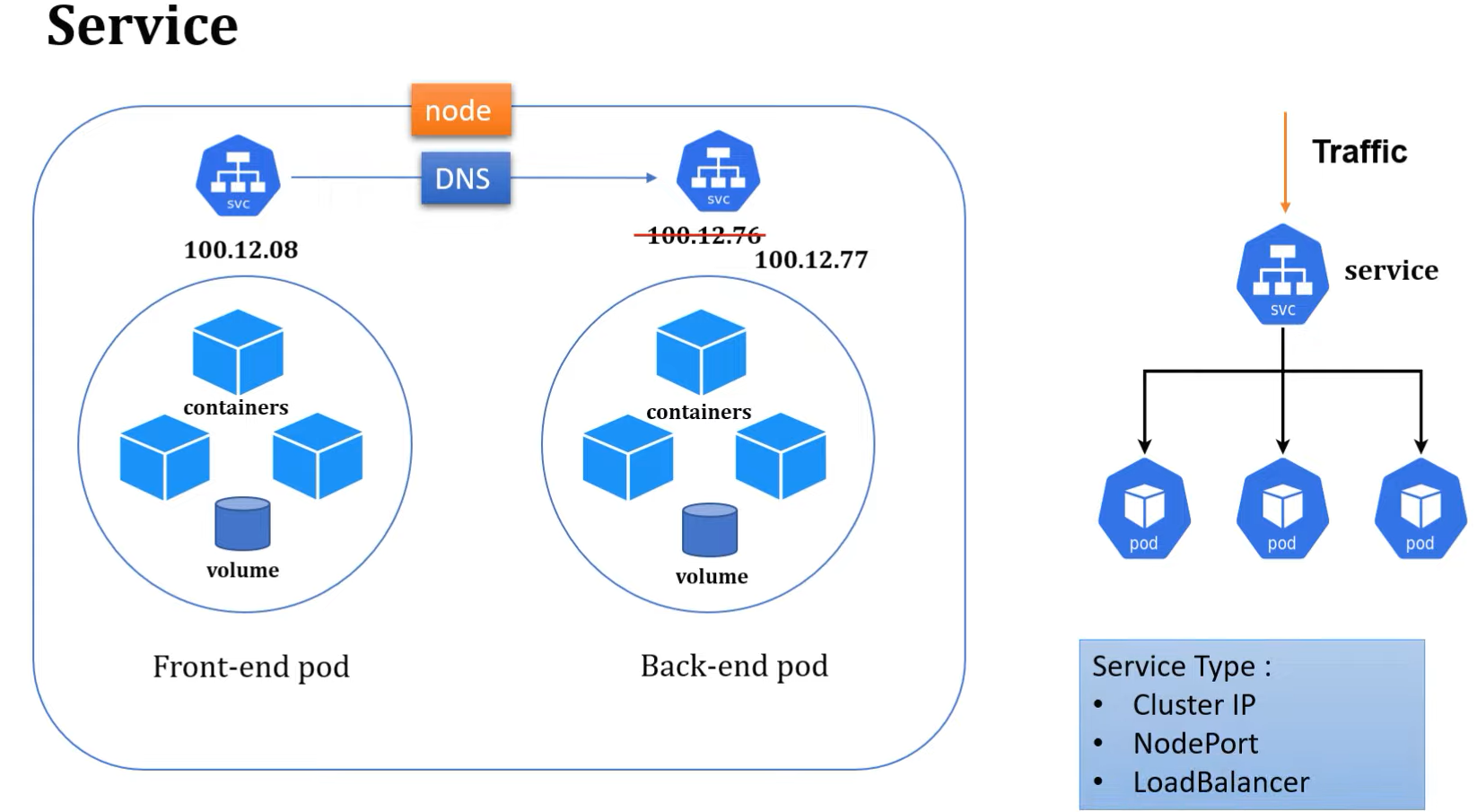


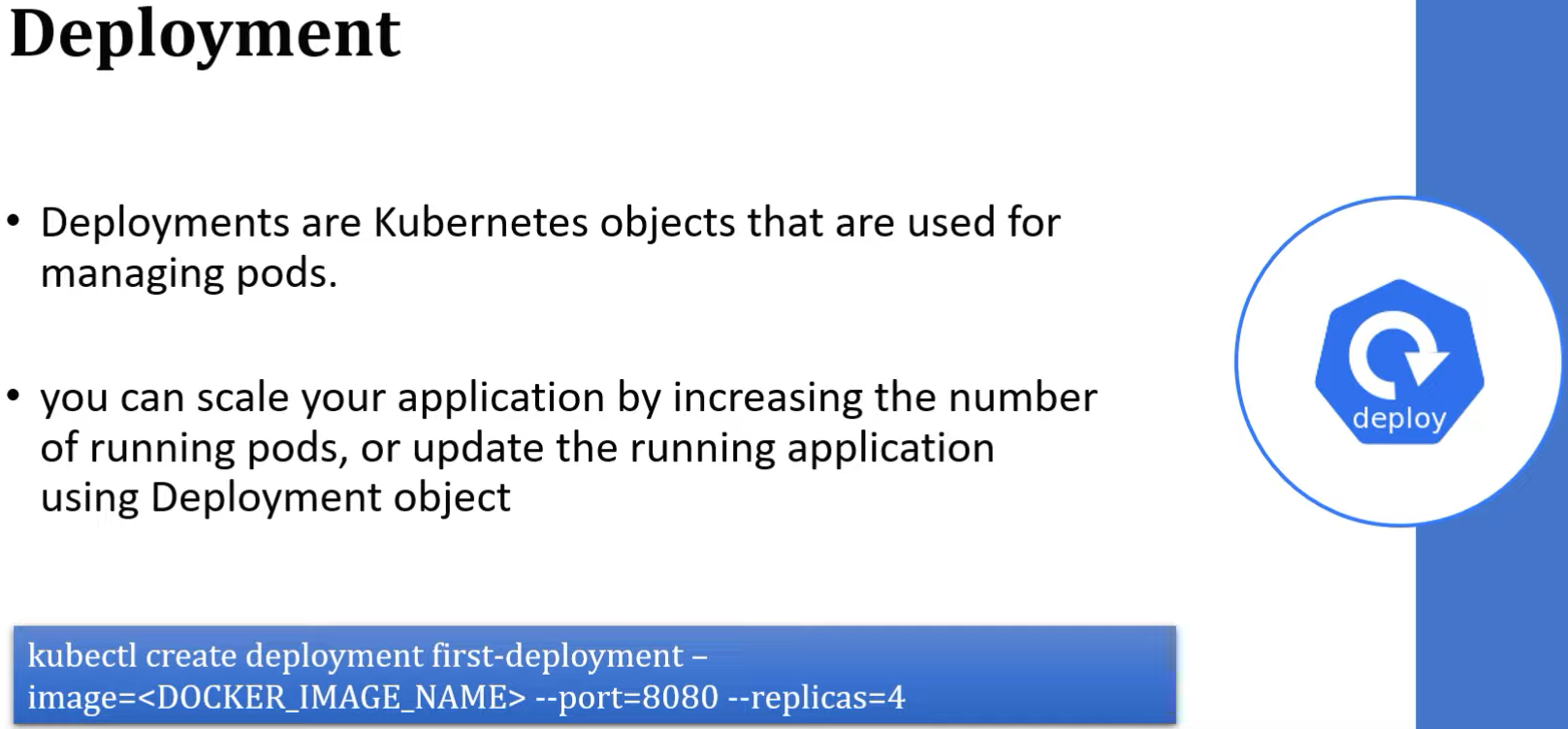


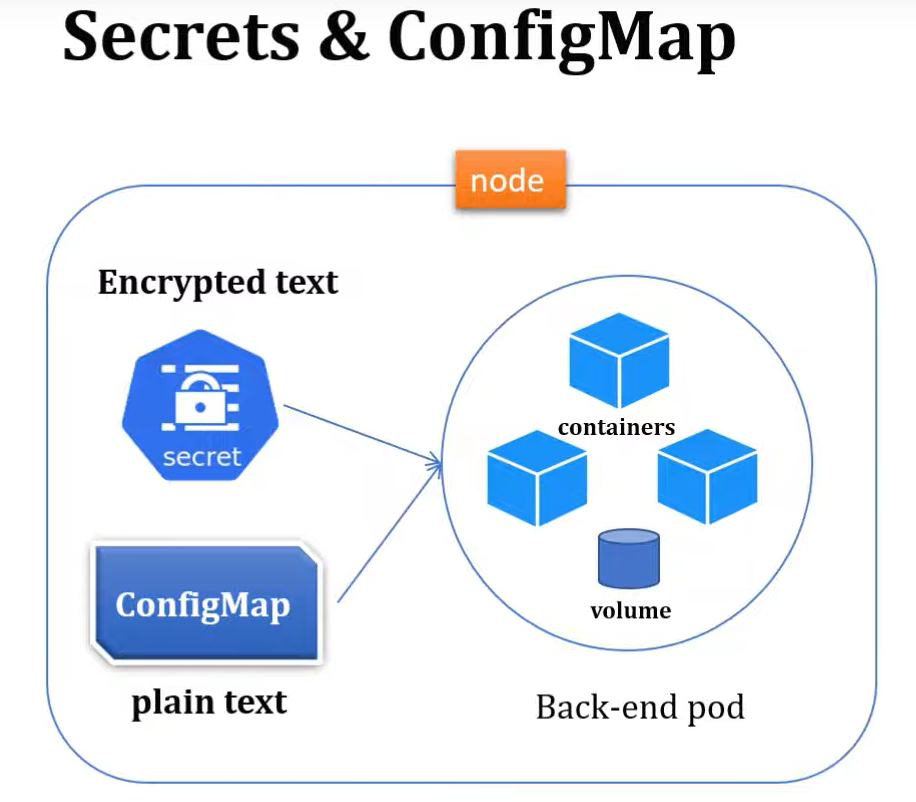
## Kubernetes components?

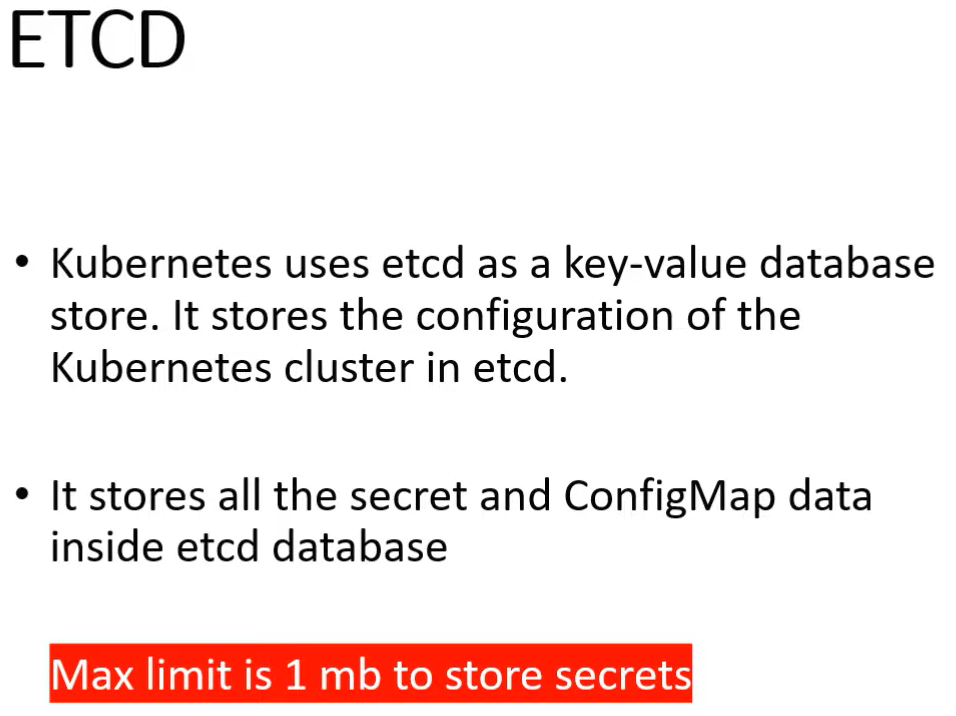


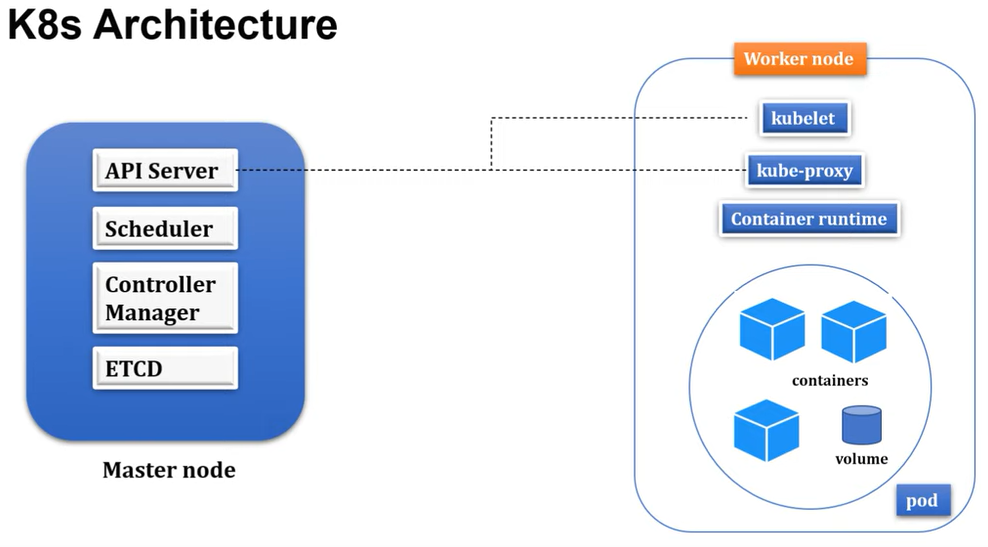








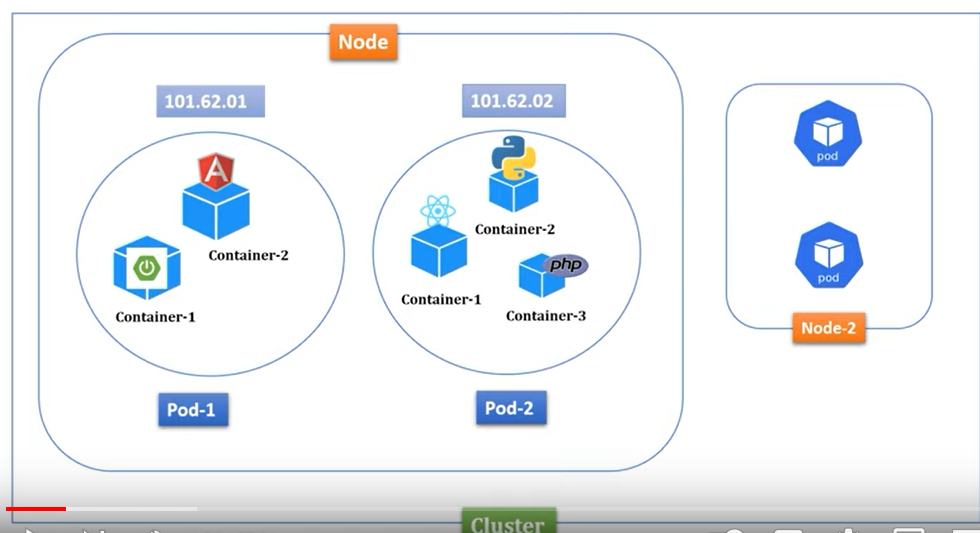




## Spring boot, Doker and Kubernetes

Resource: <https://www.youtube.com/watch?v=qof9A8k64rA&list=PLVz2XdJiJQxybsyOxK7WFtteH42ayn5i9>

1. Minimal Kubernetes structure



1. Install docker desktop and enabled Kubernetes
2. Follow this <https://www.youtube.com/watch?v=xhxmExC9N1U>
3. Check the version: *minikube version*
4. Start minikube: *minikube start --driver=doker*
5. Verify minikube well started: *minikube status*
6. Verify cluster: *kubectl cluster-info*
7. Verify node: *kubectl get node*
8. Allow kubernetes to read our repository: *eval $(minikube docker-env)*
9. Deploy first spring boot application to Kubernetes

Method 1: using command line

1. Create a spring boot application
2. Dockerize your spring boot application by adding a Dockerfile

|  |  |
| --- | --- |
| Build image | docker build -t *spring-boot-k8s:1.0* . |
| Display images | docker images |
| Deploy your images | kubectl create deployment spring-boot-k8s --image=spring-boot-k8s-demo:1.0 --port=8080 |
| Verify your deployment | kubectl get deployment |
| Describe your deployment | kubectl describe deployment *spring-boot-k8s* |
| Check the deployment status pods | kubectl get pods |
| Check the logs of the particular pods | kubectl logs *spring-boot-k8s-694d58c9b4-6gnq5* |
| Expose your deployment | kubectl expose deployment *spring-boot-k8s* –type=NodePort |
| Verify your service | kubectl get service |
| Get url | minikube service *spring-boot-k8s* --url |
| Get a dashboard. Copy the link and paste it on the browser | minikube dashboard |
| Delete all the component | kubectl delete service *spring-boot-k8s*  kubectl delete deployment *spring-boot-k8s* |
| Stop minikube | minikube stop |
| Delete minikube | minikube delete |

Resources:

<https://www.youtube.com/watch?v=xhxmExC9N1U>

Method 2: run and deploy using yaml file

1. Create a spring boot application
2. Dockerize your spring boot application by adding a Dockerfile
3. Build the image: docker build -t *spring-boot-k8s:1.0* .
4. Add the deployment file
5. Create deployment object: kubectl apply -f *k8s-deployment.yaml*
6. Verify deployment: kubectl get deployments
7. Delete deployment: kubectl delete deployment *spring-boot-k8s*
8. Verify pods: kubectl get pods
9. Logs a pods: kubectl logs *spring-boot-k8s-79844c64cd-5sr6k*
10. Addservice yaml
11. Create the service yaml: kubectl apply -f *k8s-service.yaml*
12. Verify service: kubectl get service
13. Delete deployment: kubectl delete service *spring-boot-k8s*
14. Verify the node IP : kubectl get node -o wide
15. Get the internal or minikube Ip: minikube ip
16. Get service url : minikube service *spring-boot-k8s* --url
17. Display the dashboard: minikube dashboard

Resources:

<https://www.youtube.com/watch?v=7o7e8OAAWyg>