**Kubernetes Basics Tutorial Documentation**

**Introduction**

This document is based on the official Kubernetes tutorials and provides a complete step-by-step guide from application deployment to application updates. It covers core operations such as setting up a local Kubernetes environment with Minikube, deploying applications, exploring services, exposing services, scaling applications, and updating applications.

**1. Preparation: Install Minikube**

**1.1 Install Minikube**

1. Download and install Minikube according to your operating system:  
    Windows: Use Chocolatey or download the installer directly
2. macOS: Use Homebrew
3. Linux: Download the installation script using curl
4. Installation command example (Linux):  
    curl -LO https://storage.googleapis.com/minikube/releases/latest/minikube-linux-amd64  
   sudo install minikube-linux-amd64 /usr/local/bin/minikube
5. Verify the installation:  
    minikube version

*Figure 1: Minikube version verification output*

**1.2 Start the Minikube Cluster**

|  |
| --- |
| bash minikube start |

This command will:  
 Download and start a lightweight Kubernetes clusterConfigure the kubectl command-line tool to communicate with the cluster

*Figure 2: Minikube startup process*

**1.3 Install kubectl**

kubectl is the Kubernetes command-line tool used to interact with the cluster:  
 Download and install kubectlVerify the installation:  
 kubectl version

*Figure 3: kubectl version information*

**2. Deploy App**

**2.1 Create a Deployment**

A Deployment manages the creation and updates of Pods:  
 kubectl create deployment hello-node --image=registry.k8s.io/e2e-test-images/agnhost:2.39 -- /agnhost netexec --http-port=8080

* hello-node is the name of the Deployment
* --image specifies the container image to use
* The subsequent command specifies the program to run after the container starts

*Figure 4: Create Deployment output*

**2.2 View the Deployment**

|  |
| --- |
| bash kubectl get deployments |

The output will display:  
 Deployment nameDesired number of replicasCurrent number of ready replicasNumber of available replicasAge

*Figure 5: Deployment status information*

**3. Explore App**

**3.1 View Pods**

A Pod is the smallest deployment unit in Kubernetes:  
 kubectl get pods

*Figure 6: Pod status information*

**3.2 View Cluster Events**

|  |
| --- |
| bash kubectl get events |

This command displays events that have occurred in the cluster, which is helpful for troubleshooting.

**3.3 View kubectl Configuration**

|  |
| --- |
| bash kubectl config view |

*Figure 7: kubectl configuration details*

**4. Expose App**

**4.1 Create a Service**

A Service exposes Pods to the outside world or other services in the cluster:  
 kubectl expose deployment hello-node --type=LoadBalancer --port=8080

* --type=LoadBalancer creates a load balancer to distribute traffic
* --port=8080 specifies the port to expose

*Figure 8: Service creation output*

**4.2 View the Service**

|  |
| --- |
| bash kubectl get services |

For Minikube, use the following command to open the service in a browser:  
 minikube service hello-node

*Figure 9: Exposed service access page*

**5. Scale App**

**5.1 Scale the Deployment**

Adjust the number of Pod replicas to scale the application:  
 kubectl scale deployment hello-node --replicas=3

This command increases the number of Pods to 3 for load balancing and high availability.

**5.2 Verify Scaling**

|  |
| --- |
| bash kubectl get deployments kubectl get pods |

*Figure 10: Scaled Pod status*

**6. Update App**

**6.1 Update the Container Image**

Update the application by changing the container image:  
 kubectl set image deployment hello-node hello-node=registry.k8s.io/e2e-test-images/agnhost:2.40

This command updates the image from version 2.39 to 2.40.

**6.2 Monitor the Update**

|  |
| --- |
| bash kubectl rollout status deployment hello-node |

Verify the update is complete and check the new Pods:  
 kubectl get pods