Kubernetes

Basics

Secrets

1. Kubectl: Kubernetes command-line tool

- `kubectl get pods`: List all pods in the current namespace
- `kubectl get runtimeclasses`: List all the classes of the cluster
- `kubectl get nodes`: List all nodes in the cluster
- `kubectl get secrets` : List all the secrets
- `kubectl get services`: List all services in the current namespace
- `kubectl get deployments`: List all deployments in the current namespace

2. Namespaces: Kubernetes supports multiple virtual clusters backed by the same physical cluster via namespaces

- `kubectl create namespace <name>`: Create a new namespace
- `kubectl get namespaces`: List all namespaces
- `kubectl delete pod <name>`: Delete the pod
- `kubectl delete namespace [namespace]`: Delete namespace
- `kubectl config set-context --current --namespace=<namespace>`: Set the default namespace for the current context

3. Pods: The smallest deployable units in Kubernetes, representing one or more containers

- `kubectl describe pod <pod_name>`: Get detailed information about a pod

- `kubectl logs <pod_name>`: View logs for a specific pod
- `kubectl apply -f python_image.yml`: launch pod usign python_image.yml
- `kubectl exec -it <pod_name> -- <command>`: Execute a command inside a running pod

Deployments

1. Deployment: A higher-level object that manages the desired state for pods

- `kubectl create deployment <name> --image=<image_name>`: Create a new deployment
- `kubectl get deployments`: List all deployments
- `kubectl scale deployment <name> --replicas=<count>`: Scale a deployment to the specified number of replicas

2. Updates and Rollbacks

- `kubectl set image deployment/<name> <container_name>=<new_image>`: Update the container image of a deployment
- `kubectl rollout status deployment/<name>`: Check the status of a rollout
- `kubectl rollout history deployment/<name>`: View the history of a deployment rollout
- `kubectl rollout undo deployment/<name>`: Rollback to the previous deployment version

Services

1. Service: Exposes an application running on a set of pods as a network service

- `kubectl expose deployment <name> --port=<port> --target-port=<target_port> --type=<type>`: Create a new service
- `kubectl get services`: List all services
- `kubectl delete service <name>`: Delete a service

2. Ingress: Exposes HTTP and HTTPS routes from outside the cluster to services within the cluster

- (Assuming you have an Ingress Controller installed): `kubectl apply -f <ingress_yaml_file>`: Create an ingress resource
- `kubectl get ingress`: List all ingress resources

Configurations

1. ConfigMaps: Store configuration data separately from the container image

- `kubectl create configmap <configmap-name> --from-file=<path-to-file>`: Create a ConfigMap from a file
- `kubectl get configmaps`: List all ConfigMaps

2. Secrets: Similar to ConfigMaps but designed for sensitive information

- `kubectl create secret generic <name> --from-literal=<key>=<value>`: Create a Secret
- `kubectl get secrets`: List all Secrets

Load Kubeconfig into Kubectl

```
# 1. Check Prerequisites
uname -a                    # Check Linux kernel
systemctl status docker     # Check Docker status
echo $PATH                  # Verify path

# 2. Install kubectl
curl -LO "https://dl.k8s.io/release/$(curl -L -s https://dl.k
sudo install -o root -g root -m 0755 kubectl /usr/local/bin/k

# Create .kube directory if it doesn't exist
mkdir -p ~/.kube
```

```
# Copy config from Windows Desktop
cp /mnt/c/Users/YourUsername/Desktop/kubeconfig ~/.kube/confi

# Set proper permissions
chmod 600 ~/.kube/config

# Add to .bashrc for persistence
echo 'export KUBECONFIG=~/.kube/config' >> ~/.bashrc

# Reload .bashrc
source ~/.bashrc

# 3. Check Cluster/Config
kubectl cluster-info
kubectl config view
kubectl get nodes
kubectl get namespaces

# 4. Basic Kubernetes Commands
kubectl get pods                      # List pods
kubectl get services                  # List services
kubectl create -f file.yaml           # Create resource
kubectl apply -f file.yaml            # Apply changes
kubectl delete -f file.yaml           # Delete resource
kubectl describe pod podname          # Pod details
kubectl logs podname                  # Pod logs
kubectl exec -it podname -- /bin/sh   # Shell into pod
kubectl get deployment                # List deployments
kubectl rollout status deployment     # Deployment status
kubectl get events                    # View cluster events
kubectl top nodes                     # Node resource usage
kubectl api-resources                 # List API resources
```