Here's a comprehensive **collection and classification** of commands for **setting up, debugging, and managing** your **Kubernetes cluster, TensorFlow deployment, HPA, and metrics-server**.

🚀 TensorFlow Deployment & Management

Deploy TensorFlow Model

bash
```
kubectl apply -f tensorflow-training.yaml
```

Expose TensorFlow as a Service

bash
```
kubectl apply -f tensorflow-service.yaml
```

Setup Horizontal Pod Autoscaler (HPA)

bash
```
kubectl autoscale deployment tensorflow-training --min=2 --max=10 --cpu-percent=50
```

Check TensorFlow Pod Status

bash
```
kubectl get pods -n default | grep tensorflow-training
kubectl describe pod -l app=tensorflow-training
kubectl logs -l app=tensorflow-training --tail=50
```

Delete & Redeploy TensorFlow

bash
```
kubectl delete deployment tensorflow-training
kubectl apply -f tensorflow-training.yaml
kubectl rollout restart deployment tensorflow-training
```

Delete & Recreate HPA

bash
```
kubectl delete hpa tensorflow-training
kubectl autoscale deployment tensorflow-training --min=2 --max=10 --cpu-percent=50
```

📊 Metrics-Server Debugging

Verify Metrics Availability

bash
```
kubectl get --raw "/apis/metrics.k8s.io/v1beta1/nodes"
kubectl top nodes
```

kubectl top pods

Restart Metrics-Server

bash
kubectl rollout restart deployment metrics-server -n kube-system
kubectl logs -n kube-system deployment/metrics-server --tail=50

Check If Kubelet Is Exposing Metrics

bash
NODE_NAME=<your-node-name>
kubectl get --raw /api/v1/nodes/$NODE_NAME/proxy/metrics/resource

Restart Kubelet

bash
sudo systemctl restart kubelet

Check Metrics-Server APIService

bash
kubectl get apiservices | grep metrics

If it's failing, re-register it:

bash
kubectl delete apiservice v1beta1.metrics.k8s.io
kubectl apply -f https://github.com/kubernetes-sigs/metrics-server/releases/latest/download/components.yaml

🛠️ General Debugging Commands

Check Running Pods

bash
kubectl get pods -A
kubectl describe pod <pod-name>
kubectl logs <pod-name>

Check Services

bash
kubectl get svc -A
kubectl describe svc tensorflow-service

Check HPA Scaling Status

bash

```
kubectl get hpa tensorflow-training
kubectl describe hpa tensorflow-training
```

🔍 Additional Debugging Commands

Check Events for Issues

bash
```
kubectl get events --sort-by='.lastTimestamp'
```

This helps identify any errors related to pod scheduling, failures, or Kubernetes resource issues.

Check Cluster Component Health

bash
```
kubectl get componentstatuses
```

This ensures key components like the **API server, scheduler, and controller-manager** are running properly.

Troubleshoot Failed Pods

bash
```
kubectl describe pod <pod-name>
kubectl logs <pod-name>
kubectl logs -l app=tensorflow-training --tail=50
```

Check Kubelet Logs (Node-Specific Issues)

bash
```
journalctl -u kubelet --no-pager | tail -50
```

This helps detect whether Kubelet is failing to report pod metrics correctly.

📊 Advanced Metrics Debugging

Check APIService Registration for Metrics-Server

bash
```
kubectl get apiservices | grep metrics
kubectl describe apiservice v1beta1.metrics.k8s.io
```

Query Metrics Directly From Kubelet

bash
```
NODE_NAME=<your-node-name>
kubectl get --raw /api/v1/nodes/$NODE_NAME/proxy/metrics/resource
```

If pod metrics are missing, Kubelet isn't reporting them properly.

Force Metrics-Server to Refresh

bash
```
kubectl rollout restart deployment metrics-server -n kube-system
kubectl logs -n kube-system deployment/metrics-server --tail=50
```

🛠️ TensorFlow Deployment Enhancements

Scale Up TensorFlow Deployment Manually

bash
```
kubectl scale deployment tensorflow-training --replicas=5
```

This manually increases the number of TensorFlow pods.

Delete Everything Related to TensorFlow

bash
```
kubectl delete deployment tensorflow-training
kubectl delete service tensorflow-service
kubectl delete hpa tensorflow-training
kubectl delete configmap tensorflow-script
```

Then, **redeploy** everything using:

bash
```
kubectl apply -f tensorflow-deployment.yaml
kubectl apply -f tensorflow-service.yaml
kubectl create configmap tensorflow-script --from-file=train_and_delete.py
kubectl autoscale deployment tensorflow-training --min=1 --max=10 --cpu-percent=50
```

🛡️ Security & Resource Optimization

Check Resource Limits Across Namespace

bash
```
kubectl get resourcequotas -A
kubectl describe resourcequota <quota-name>
```

Useful if a pod is failing due to exceeding CPU/memory limits.

Check Network Policies (If Any)

bash
```
kubectl get networkpolicy -A
kubectl describe networkpolicy <policy-name>
```