**Microservice Log Controlling Commands**

**Authentication & Login Commands**

**1. Basic vSphere Login**

kubectl vsphere login --server=x.x.x.x --insecure-skip-tls-verify -u ibdev01@vsphere.local

**Purpose:** Authenticates to the vSphere environment using kubectl vsphere plugin

* --server=x.x.x.x: Connects to vSphere server at IP x.x.x.x
* --insecure-skip-tls-verify: Bypasses SSL certificate validation (for development environments)
* -u ibdev01@vsphere.local: Login username for vSphere domain

**2. Tanzu Kubernetes Cluster Login**

kubectl vsphere.exe login --insecure-skip-tls-verify --server=x.x.x.x -u ibuat@vsphere.local --tanzu-kubernetes-cluster-namespace=ib-dev --tanzu-kubernetes-cluster-name=dev-ib-cluster

**Purpose:** Authenticates and connects to a specific Tanzu Kubernetes cluster

* --server=x.x.x.x: Different vSphere server endpoint
* -u ibuat@vsphere.local: UAT environment user account
* --tanzu-kubernetes-cluster-namespace=ib-dev: Target namespace for the cluster
* --tanzu-kubernetes-cluster-name=dev-ib-cluster: Specific cluster name to connect to

**Pod Management Commands**

**3. List Pods in Namespace**

kubectl get pods -n ib-dev

**Purpose:** Displays all running pods in the 'ib-dev' namespace

* Shows pod names, status, restarts, and age
* Essential for identifying which pods are available for debugging

**4. Access Pod Shell**

kubectl exec -it content-556b5d8b7-wtr26 -n ib-dev -- /bin/bash

**Purpose:** Opens an interactive terminal session inside a specific pod

* -it: Interactive terminal with TTY allocation
* content-556b5d8b7-wtr26: Specific pod name
* -n ib-dev: Target namespace
* -- /bin/bash: Execute bash shell inside the container

**Log Analysis Commands**

**5. Navigate to Application Logs**

root@content-556b5d8b7-wtr26:/# cd logs

root@content-556b5d8b7-wtr26:/logs# cd content

**Purpose:** Navigate to the application's log directory structure within the pod

* Standard directory traversal to reach log files location

**6. Filter Log Warnings**

root@content-556b5d8b7-wtr26:/logs/content# cat today-payment-b5c48877d-w86z2.log | grep WARN

**Purpose:** Search for WARNING level messages in payment service logs

* cat: Display file contents
* | grep WARN: Filter output to show only lines containing "WARN"
* Useful for identifying potential issues or error conditions

**7. Decompress Log Files**

gzip -d log-payment-65b8d9b49f-xmdjp.2025-05-15\_16-19.0.1og.gz

**Purpose:** Extract compressed log files for analysis

* -d: Decompress the gzipped log file
* Enables access to archived log data from specific time periods

**Remote Log Monitoring Commands**

**8. Stream Application Logs (Incorrect Syntax)**

# Incorrect command (has typo: kubect1 instead of kubectl)

C:\Users\it232059> kubect1 logs --all-containers -l app=portal -n ib-dev -f --max-log-requests=20

**9. Stream Application Logs to File (Corrected)**

C:\Users\it232059> kubectl logs --all-containers -l app=portal -n ib-dev -f --max-log-requests=20 >C:\Users\it232059\Desktop\LOGS\1.txt

**Purpose:** Stream real-time logs from portal application to a local file

* --all-containers: Include logs from all containers in matching pods
* -l app=portal: Select pods with label "app=portal"
* -n ib-dev: Target namespace
* -f: Follow/stream logs in real-time
* --max-log-requests=20: Limit concurrent log streaming requests
* >C:\Users\it232059\Desktop\LOGS\1.txt: Redirect output to local file

**Typical Workflow Summary**

This command sequence represents a common troubleshooting workflow:

1. **Authentication:** Login to vSphere/Tanzu environment
2. **Discovery:** List available pods to identify targets
3. **Access:** Enter pod for direct log analysis
4. **Investigation:** Navigate log directories and search for issues
5. **Archival:** Extract compressed historical logs when needed
6. **Monitoring:** Stream live logs to local files for analysis

**Environment Details**

* **Development Environment:** ib-dev namespace
* **Application Components:** Content service, Payment service, Portal application
* **Infrastructure:** VMware vSphere with Tanzu Kubernetes clusters
* **Log Management:** Combination of in-pod analysis and remote streaming