Full Guide: Introduction to Kubernetes Objects (Local Lab)

This guide adapts the IBM Practice Lab for a local **Arch Linux** environment using **Minikube** and **Docker**.

Prerequisites & Environment Setup

Before starting the lab tasks, ensure your local cluster is running:

1. **Start Docker Engine**:
   ```
   sudo systemctl start docker
   ```
2. **Start Minikube**:
   ```
   minikube start --driver=docker
   ```
3. **Verify Connection**:
   ```
   kubectl get nodes
   ```

Task 1: Create a Kubernetes Service

The goal is to deploy Nginx and make it accessible via a network port.

1. **Create the Deployment**:
   ```
   kubectl create deployment my-deployment1 --image=nginx
   ```
2. **Expose as a NodePort Service**:
   ```
   kubectl expose deployment my-deployment1 --port 80 --type=NodePort --name=my-service1
   ```
3. Access the Application:
   In a local environment, use Minikube to open the browser:
   ```
   minikube service my-service1
   ```

Task 2: Kubectl Management Commands

Practice inspecting and organizing your running pods.

1. **List Pods and Labels**:
   ```
   kubectl get pods --show-labels
   ```
2. Add a Label to your Pod:
   (Replace <pod-name> with the actual name from the command above)

```
kubectl label pods <pod-name> environment=deployment
```

3. **View Pod Details & Logs**:

   ```
   kubectl describe pod <pod-name>
   kubectl logs <pod-name>
   ```

Task 3: Deploy a StatefulSet

StatefulSets are used for applications that require a stable identity.

1. **Create the file statefulset.yaml**:

   ```
   apiVersion: apps/v1
   kind: StatefulSet
   metadata:
     name: my-statefulset
   spec:
     serviceName: "nginx"
     replicas: 3
     selector:
       matchLabels:
         app: nginx
     template:
       metadata:
         labels:
           app: nginx
       spec:
         containers:
         - name: nginx
           image: nginx
           ports:
           - containerPort: 80
             name: web
   ```

2. **Apply the Manifest**:

   ```
   kubectl apply -f statefulset.yaml
   ```

3. Verify Sticky Identity:
   Observe that pods are named my-statefulset-0, my-statefulset-1, etc.

   ```
   kubectl get pods
   ```

Task 4: Implement a DaemonSet

DaemonSets run a pod on every node in the cluster.

1. **Create the file daemonset.yaml**:
   ```
   apiVersion: apps/v1
   kind: DaemonSet
   metadata:
     name: my-daemonset
   spec:
     selector:
       matchLabels:
         name: my-daemonset
     template:
       metadata:
         labels:
           name: my-daemonset
       spec:
         containers:
         - name: my-daemonset
           image: nginx
   ```

2. **Apply the Manifest**:
   ```
   kubectl apply -f daemonset.yaml
   ```

3. **Verify Deployment**:
   ```
   kubectl get daemonsets
   ```

Cleanup (End of Session)

Always clean up your local machine to save RAM and battery.

1. **Delete Lab Resources**:
   ```
   kubectl delete -f daemonset.yaml
   kubectl delete -f statefulset.yaml
   kubectl delete service my-service1
   kubectl delete deployment my-deployment1
   ```

2. **Stop Services**:
   ```
   minikube stop
   sudo systemctl stop docker
   ```