1. mongodb-deployment.yaml

This file defines the Deployment for your MongoDB database. It tells Kubernetes how to run your MongoDB container and ensures that a MongoDB Pod is always available.

```
# mongodb-deployment.yaml
apiVersion: apps/v1 # Specifies the API version for Deployment objects
kind: Deployment    # Defines this as a Deployment kind of resource
metadata:
  name: mongodb-deployment # A unique name for your Deployment
  labels:
    app: mongodb # Labels are key-value pairs used to organize and select resources
spec:
  replicas: 1 # We start with 1 replica for a single MongoDB instance.
              # For production, consider StatefulSets for high availability of databases.
  selector:
    matchLabels:
      app: mongodb # This links the Deployment to the Pods it manages
  template: # This defines the Pods that this Deployment will create
    metadata:
      labels:
        app: mongodb # Labels for the Pods, matching the selector above
    spec:
      containers:
      - name: mongodb # Name of the container within the Pod
        image: mongo:7.0 # The Docker image to use
        ports:
        - containerPort: 27017 # The port your MongoDB container listens on
        env: # Environment variables for the container
        - name: MONGO_INITDB_DATABASE
          value: linkshare_db
        volumeMounts: # How volumes are mounted inside the container
        - name: mongodb-storage # Refers to the volume defined at the Pod level
          mountPath: /data/db # The path inside the container where the volume will be
mounted
      volumes: # Defines the volumes available to this Pod
      - name: mongodb-storage # Name of the volume
        persistentVolumeClaim: # This connects to a PersistentVolumeClaim for durable
storage
          claimName: mongodb-pvc # We'll define this 'mongodb-pvc' in the next file!
```

2. mongodb-service.yaml

This file defines the Service for your MongoDB database. It provides a stable internal DNS name (mongodb-service) and IP address within the Kubernetes cluster, allowing other services (like your backend) to reliably connect to MongoDB, even if the underlying Pod changes.

```
# mongodb-service.yaml
apiVersion: v1     # API version for Service objects
kind: Service      # Defines this as a Service kind of resource
metadata:
  name: mongodb-service # A unique name for your Service.
                        # This is what your backend will use to connect (e.g.,
mongodb://mongodb-service:27017).
  labels:
    app: mongodb # Labels for the Service, often matching the app it exposes
spec:
  selector:
    app: mongodb # This is CRUCIAL! It tells the Service to target Pods with the label
'app: mongodb'.
  ports:
    - protocol: TCP
      port: 27017       # The port on which this Service will be accessible inside the
cluster
      targetPort: 27017 # The port on the Pod (container) that the Service will forward
traffic to
  clusterIP: None # This creates a 'headless' service, useful for stateful applications
like databases.
                  # It means DNS will resolve directly to the Pod's IP(s) instead of a single
Service IP.
```

3. mongodb-pvc.yaml

This file defines the PersistentVolumeClaim (PVC) for your MongoDB data. It requests a specific amount of storage that Kubernetes will provision (if dynamic provisioning is enabled) and attach to your MongoDB Pod, ensuring your data persists even if the Pod is deleted or recreated.

```
# mongodb-pvc.yaml
apiVersion: v1
kind: PersistentVolumeClaim # We are defining a request for storage
metadata:
  name: mongodb-pvc         # The name of our PVC, referenced by the MongoDB
Deployment
  labels:
    app: mongodb
spec:
  accessModes:
    - ReadWriteOnce # This means the volume can be mounted as read-write by a
single Node.
  resources:
    requests:
      storage: 1Gi   # We are requesting 1 Gigabyte of storage. Adjust this size as
needed!
  # storageClassName: standard # (Optional) Uncomment and specify if your cluster
has a
                           # specific StorageClass you want to use (e.g., 'standard', 'gp2',
'azurefile').
                           # If omitted, the default StorageClass will be used.
```