Investigate issues with the a deployment in Kubernetes

You can follow these steps:

1. **Check Deployment Status**:

   ```
   kubectl get deployments
   ```

   This command will list all deployments, including their status.

2. **Check Pod Status**:

   ```
   kubectl get pods -l app=<name>
   ```

   This command will list the pods associated with the `<name>` deployment.

3. **Check Logs**:

   ```
   kubectl logs -f -c <name> <pod-name>
   ```

   Replace `<pod-name>` with the name of the pod you identified in the previous step. The `-f` flag will stream the logs continuously.

4. **Check Events**:

   ```
   kubectl get events --sort-by=.metadata.creationTimestamp
   ```

   This command will show recent events in your cluster, which can help you understand what might be causing the issues.

5. **Check Resource Utilization**:

   ```
   kubectl top pod <pod-name>
   ```

   This command will show the CPU and memory usage of the pod, which can help you identify if there are any resource constraints.

6. **Check Deployment History**:

   ```
   kubectl rollout history deployment masterdata
   ```

   This command will show the deployment history, which can help you understand if there have been any recent changes or rollbacks.

7. **Check Configuration**:

   ```
   kubectl describe deployment masterdata
   ```

   This command will provide detailed information about the deployment configuration, including any recent changes.

8. **Check Persistent Volumes (if applicable)**:

   ```
   kubectl get pvc -l app=<name>
   ```

   This command will list any Persistent Volume Claims (PVCs) associated with the `<name>` deployment.

9. **Check Cluster Resources**:

   ```
   kubectl get nodes -o wide
   ```

   This command will show the status of your cluster nodes, including their resource utilization.

By following these steps, you should be able to gather enough information to understand what might be causing the crashes and issues in your `<name>` deployment.