**Getting a Heap Dump**

The first thing we need to do is to **get the running pod's name**. We can take it from the *kubectl get pods* command from the previous chapter. In our case, it's *prime-number-finder-pod*.

The next step is to **use that name to get into our running pod**. We'll use the Kubernetes *exec* command to do that:

$ kubectl exec -it prime-number-finder-pod bashCopy

Now, we need to **get the process id of our running JVM app**. We can use the *jps* command which is built into the JDK.

The next step is to **create the heap dump**. Once more, we'll use a built-in JDK tool:

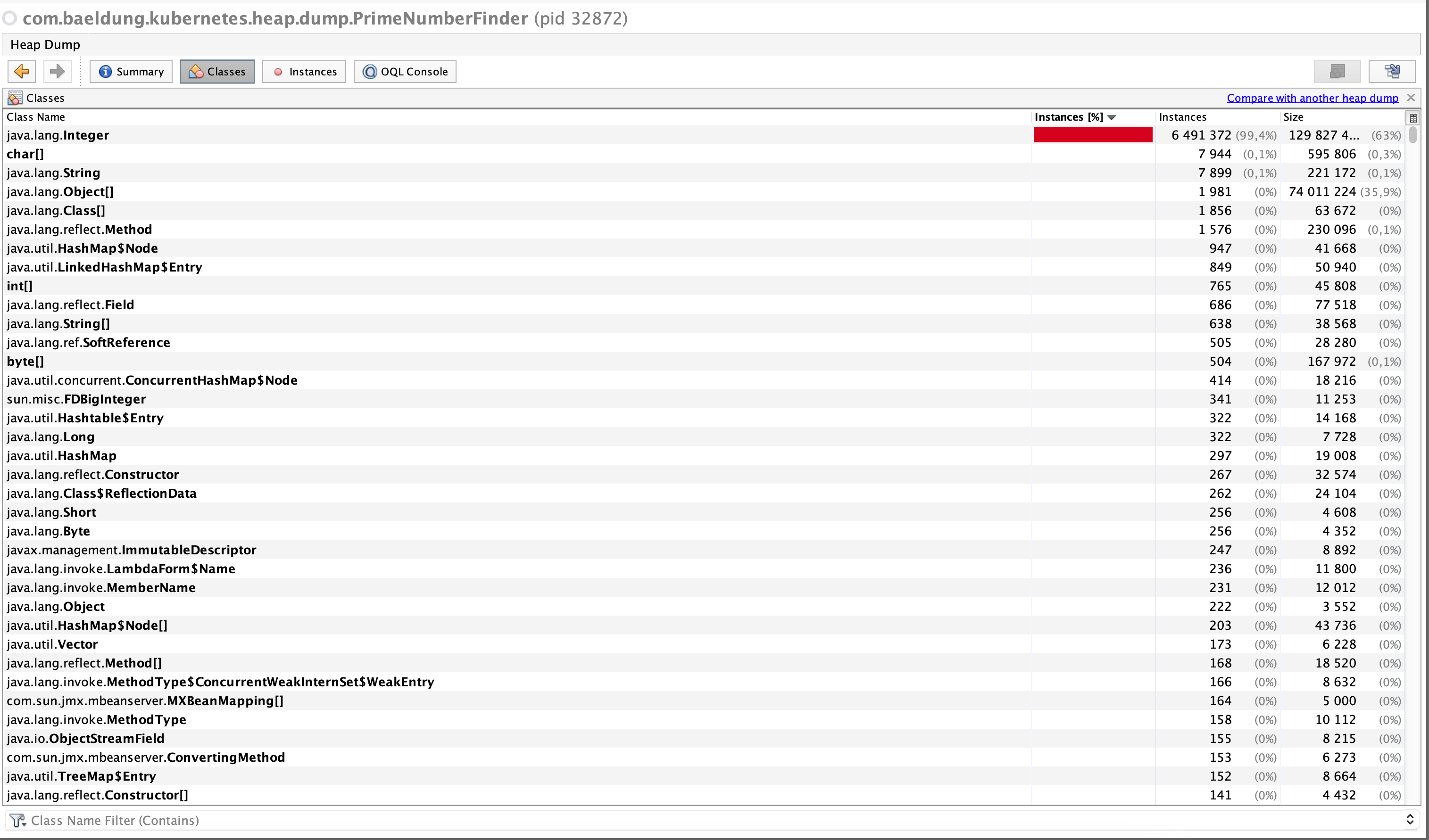
$ jmap -dump:live,format=b,file=prime\_number\_heap\_dump.bin <process\_id>Copy

The last thing we need to do is to **copy the newly created heap dump from the pod to our local machine**:

$ kubectl cp prime-number-finder-pod:prime\_number\_heap\_dump.bin <our local destination directory>Copy

Now, we can use [any memory analysis tool](https://www.baeldung.com/java-analyze-thread-dumps), such as JvisualVM provided with JDK, or 3rd-party applications, such as JProfiler or JStack Review, to analyze the heap dump

This is what a 10-minute prime number app's heap dump analysis in JvisualVM looks like:

[](https://www.baeldung.com/wp-content/uploads/2023/03/prime-number-heap-dump-1.png)

# **How to get a heap or thread dump from JVM inside a pod in Kubernetes?**

When you run Java programs on a desktop or a server, you can quickly get heap or thread dump using JDK tools like jmap, jstack Or other tools and commands.

## 1- Connect to the POD

First, you need to connect to the pod which is running your Java application. For this purpose, you can get the shell to that pod using this command:

kubectl exec -it POD\_NAME -- /bin/bash

and then you can get the heap or thread dump by using the jmap or jstack .

You can directly run the jmap or jstack command using the kubectl command:

kubectl exec POD\_NAME -c CONTAINER\_NAME -- bash -c "COMMAND TO GET HEAP OR THREAD DUMP"

## 2- Get the heap and thread dump

There are several commands to get the heap or thread dump. To get the heap dump, you can use this command:

jmap -dump:file=/tmp/HEAP\_DUMP\_FILENAME.jmap JAVA\_PROCESS\_ID

To get the thread dump, you can use this command:

jstack JAVA\_PROCESS\_ID > /tmp/THREAD\_DUMP\_FILENAME.tdump

You have other options to get heap and thread dump. You can read more about these options in [this](https://www.baeldung.com/java-heap-dump-capture) and [this](https://www.baeldung.com/java-thread-dump) article.

## 3- Copy dump files to your machine

In the last step, you need to copy the dump files to your machine to analyze them using other tools like [VisualVM](https://visualvm.github.io/" \t "_blank). To do that, you can use the kubectl copy command:

kubectl cp POD\_NAME:/tmp/DUMP\_FILENAME.tdump /PATH\_IN\_YOUR\_MACHINE/DUMP\_FILENAME.tdump -c CONTAINER\_NAME