**Offline Revision clean up**

<http://blog.magmalabs.io/2017/07/31/reducing-size-aem-instance.html>

By default, AEM uses Tar files to store the nodes and binaries of the application. One characteristic of using Tar files for storage is that the changes are never overwritten even if we are only updating data, which continually increases disk space usage.

AEM uses a mechanism called Revision Cleanup to remove unused data from the repository and reclaim disk space.

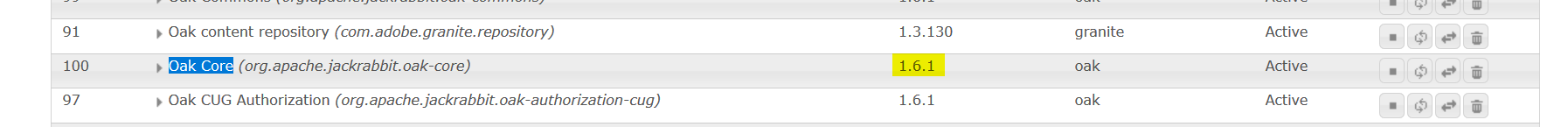
**Performing an Offline Revision Cleanup**

Adobe provides a tool called Oak-run that performs an offline cleanup revision. The Offline term means that the AEM instance should be stopped for the cleanup process to work.

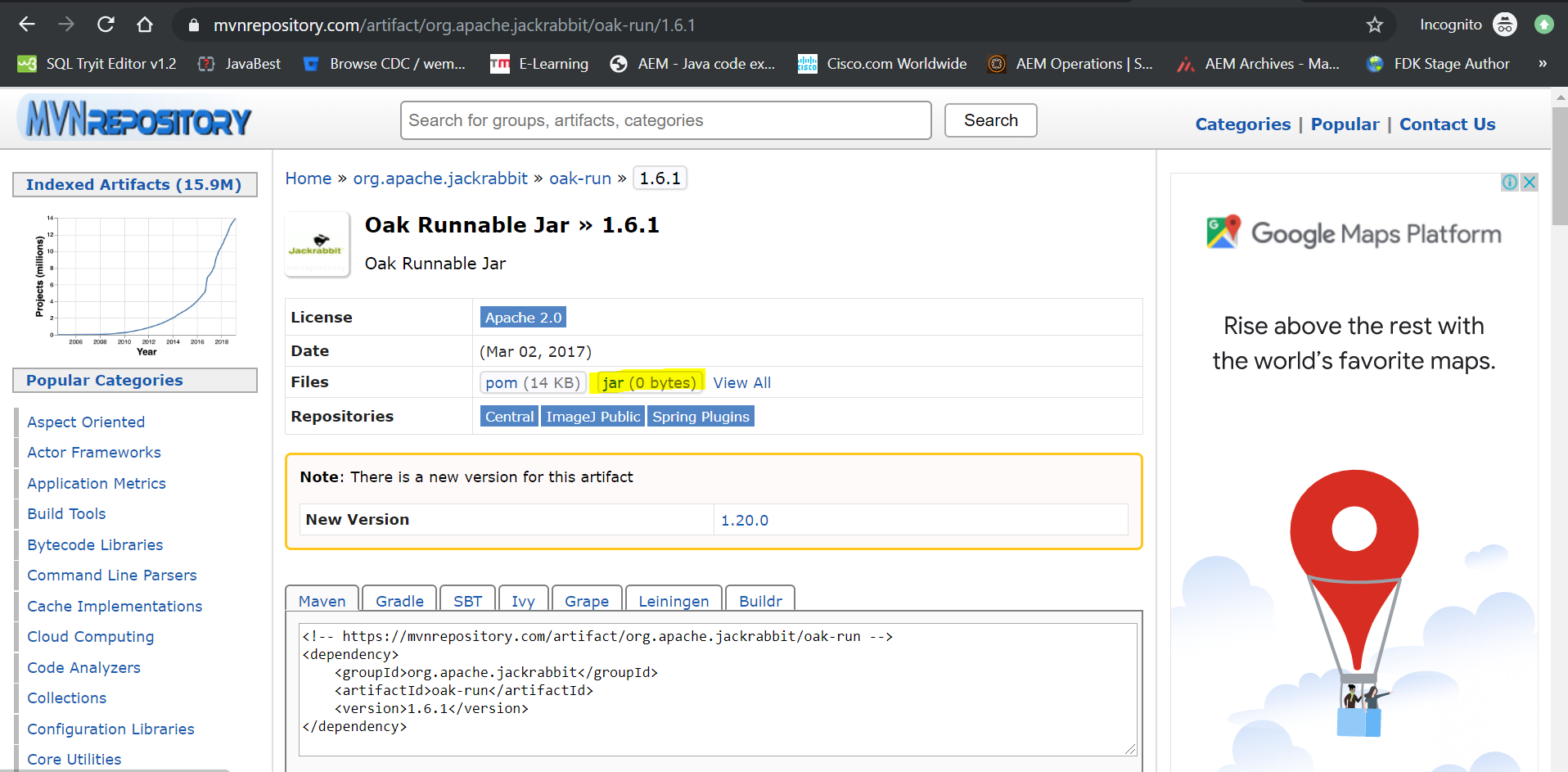
Steps:

1. **Downloading the correct Oak-run version**

* Shut down the AEM instance
* Go to /system/console/bundles (e.g. http://localhost:4502/system/console/bundles) and look for “Oak Core” to check the Oak version.

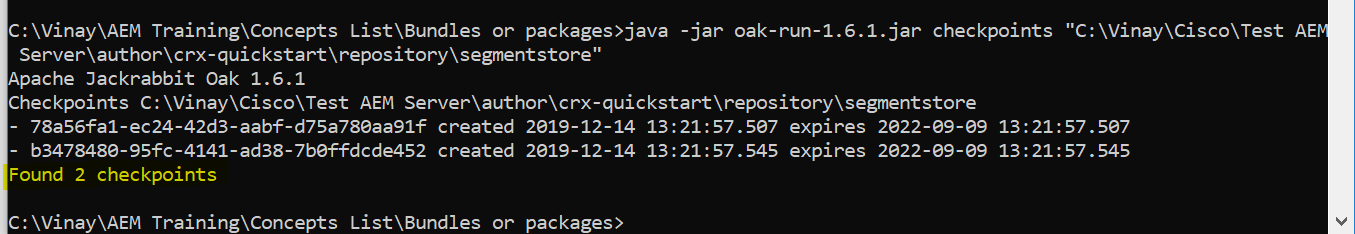


* Go to http://mvnrepository.com/artifact/org.apache.jackrabbit/oak-run/ and look for the Oak-run version that matches the “Oak Core” version (e.g. https://repo1.maven.org/maven2/org/apache/jackrabbit/oak-run/1.2.2/ if the Oak Core version is 1.2.2)



1. **Deleting unreferenced checkpoints (Optional)**

* Optionally, before running the cleanup process, you can use the tool for deleting unreferenced checkpoints
* Go to the folder where the Oak-run jar present and execute the below command in cmd.
* Run: java -jar oak-run.jar checkpoints install-folder/crx-quickstart/repository/segmentstore to find the checkpoints



* Once the checkpoints are found, we should be deleting them.

Then, to delete them run java -jar oak-run.jar checkpoints install-folder/crx-quickstart/repository/segmentstore rm-unreferenced

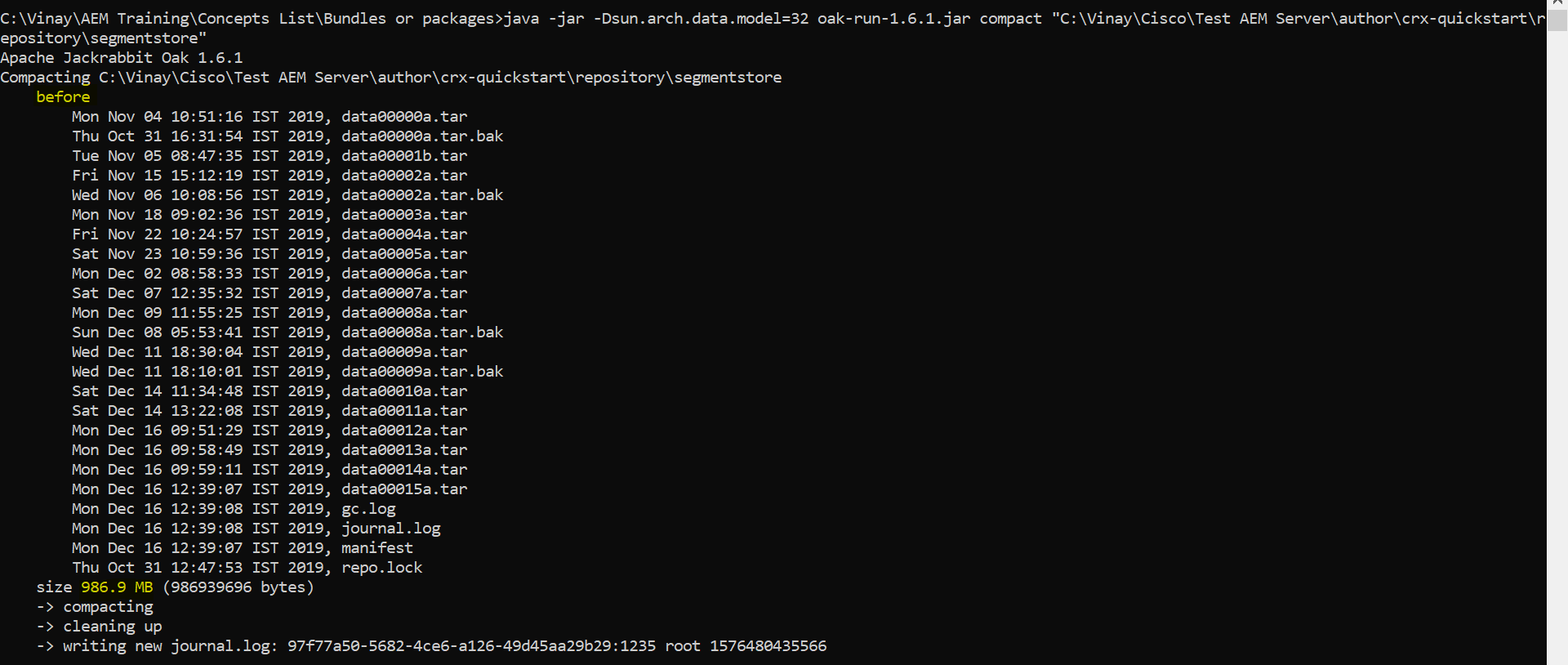


1. **Running the cleanup process**

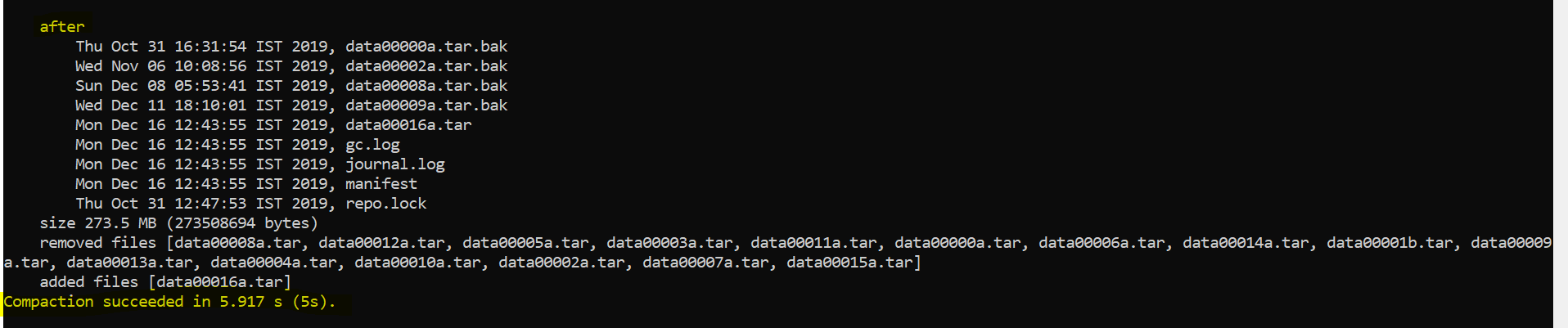
* Run the following command to start the cleanup process.

java -jar -Dsun.arch.data.model=32 oak-run-1.6.1.jar compact "C:\Vinay\\*\*\*\Test AEM Server\author\crx-quickstart\repository\segmentstore"

* While process is in progress, we see the log like shown below.



* Once the process is completed, we see the following message in console.



Finally we would see a difference after compaction.

**Before:** 986.9 MB

**After:** 273.5MB