**🔧 Problem Simulation on Production Environments**

After completing the Dynatrace integration on your Linux machine and Kubernetes cluster, the next step is to simulate various performance issues in a production environment. We will use the **BuggyApp** application to induce and simulate different performance problems.

**🚨 Performance Problems to Simulate:**

1. **⚡ CPU Issues**
2. **💥 OutOfMemoryError**
3. **🧠 Memory Leak**
4. **🔒 Blocked Threads**
5. **💾 I/O Issues**

Buggy App is a simple java application that simulates different performance problems like Memory Leak, OutOfMemoryError, CPU spike, thread leak, StackOverflowError, deadlock, unresponsiveness.

**🛠️ Steps to Simulate Problems**

1. **Clone the Repository:**
   * First, clone the **BuggyApp** repository to your local machine using the following command

git clone <https://github.com/Sumanth17-git/buggyApp.git>

**🔧 Simulate CPU Spike:**

To simulate a **CPU spike**, invoke the **buggyApp.jar** with the following command:

java -jar buggyApp.jar PROBLEM\_CPU

**💥 Simulate OutOfMemoryError (Java Heap Space):**

**To generate a Java.lang.OutOfMemoryError: Java heap space, use the following command:**

java -Xmx512m -jar buggyApp.jar PROBLEM\_OOM

**🧠 Simulate Memory Leak:**

To simulate a **memory leak** without causing an OutOfMemoryError, use the following command:

java -jar buggyApp.jar PROBLEM\_MEMORY

**🔄 Simulate Thread Leak:**

**To simulate a thread leak (where the app slowly spawns new threads), run the following command:**

java -jar buggyApp.jar PROBLEM\_THREADLEAK

When you launch the Buggy App with these arguments eventually  ‘java.lang.OutOfMemoryError: Unable to create new native thread’ will be generated.

**🔒 Simulate Deadlock:**

To simulate a **deadlock** in the application, use this command:

 java -jar buggyApp.jar PROBLEM\_DEADLOCK

**🛑 Simulate Blocked Threads:**

To put multiple threads in a **BLOCKED** state, invoke the app with the following command:

 java -jar buggyApp.jar PROBLEM\_BLOCKED

**💾 Simulate Heavy I/O Activity:**

To simulate **heavy I/O** activity, run the following command:

 java -jar buggyApp.jar PROBLEM\_IO

When you launch buggyApp with this argument, application will create 5 unique files by name: fileIO-1.txt, fileIO-2.txt, fileIO-3.txt, fileIO-4.txt, fileIO-5.txt. Then 20 lines of contents will be written in to this file and read back again and again infinietly. It will cause heavy I/O activity on the host.