**鲲鹏展翅Native认证测试用例执行结果**

1. **测试用例执行结果**
   1. DevKit IDE插件安装测试
      1. IDE插件安装测试

IDE插件安装测试

|  |  |
| --- | --- |
| **用例名称** | IDE插件安装测试 |
| **用例编号** | 01 |
| **测试目的** | 确认用户已安装DevKit IDE插件并完成应用迁移工具、亲和分析工具、应用开发工具、编译调试工具、系统性能分析工具、Java性能分析工具、系统诊断工具安装 |
| **预置条件** | 完成Visual Studio Code安装  参考[插件安装指导](https://www.hikunpeng.com/document/detail/zh/kunpengdevps/userguide/Plugins_UserGuide/KunpengDevKitVS_0004.html)，完成鲲鹏DevKit插件安装。  参考[工具安装指导](https://www.hikunpeng.com/document/detail/zh/kunpengdevps/userguide/Plugins_UserGuide/KunpengDevKitVS_0010.html)，完成应用迁移工具、亲和分析工具、应用开发工具、编译调试工具、系统性能分析工具、Java性能分析工具、系统诊断工具安装。 |
| **测试步骤** | 1. 打开VSCODE主页面，左边插件栏存在鲲鹏DevKit      1. 点击插件按钮，跳转到登录页面，输入账号密码完成登录 2. 登录后点击右上角齿轮按钮，再点击左边列表中的“工具管理”，查看已安装的工具，查看应用迁移工具、亲和分析工具、应用开发工具、编译调试工具、系统性能分析工具、Java性能分析工具、系统诊断工具的安装状态，有预期结果A 3. 查看左侧代码目录，有预期结果B |
| **预期结果** | [A] 应用迁移工具、亲和分析工具、应用开发工具、编译调试工具、系统性能分析工具、Java性能分析工具、系统诊断工具为已安装状态  [B] 左侧代码目录显示产品代码仓名称 |
| **测试结果**  **（测试日志或截图）** | *请补充相关测试结果的截图* |
| **测试结论** | *通过/有条件通过/不通过* |
| **备注** | *若不通过或有条件通过，在此备注说明* |

* 1. Devkit开发流水门禁测试
     1. 门禁配置迁移扫描工具测试

门禁配置迁移扫描工具测试

|  |  |
| --- | --- |
| **用例名称** | 门禁配置迁移扫描工具测试 |
| **用例编号** | 02 |
| **测试目的** | 验证用户流水线已配置DevKit迁移扫描工具门禁 |
| **预置条件** | 流水线场景：   1. 参考[devkit-pipeline使用说明](https://www.hikunpeng.com/document/detail/zh/nativedevp/userguide/devkitpipepre/NativeDevelopment_0001.html)，完成流水线安装部署 2. 参考[创建流水线门禁检查任务](https://www.hikunpeng.com/document/detail/zh/nativedevp/userguide/pipeline/NativeDevelopment_0018.html)，完成DevKit迁移扫描工具（软件迁移评估、源码迁移）门禁配置   非流水线场景：  参考[DevKit用户指南](https://www.hikunpeng.com/document/detail/zh/kunpengdevps/userguide/cliuserguide/KunpengDevKitCli_0001.html)，完成DevKit命令行工具安装 |
| **测试步骤** | 流水线场景：   1. 通过流水线发布产品包 2. 收集迁移扫描报告（软件迁移评估报告、源码迁移报告）并一起打包成tar.gz压缩包 3. 参考[DevKit验收工具使用说明](https://gitee.com/openeuler/devkit-pipeline/blob/master/document/%E9%AA%8C%E6%94%B6%E6%8A%A5%E5%91%8A%E6%94%B6%E9%9B%86/devkit-pipeline%E7%9B%B8%E5%85%B3%E6%8A%A5%E5%91%8A%E9%AA%8C%E6%94%B6%E6%96%87%E6%A1%A3.md)，使用验收工具解析压缩包并生成验收报告，查看验收报告，有预期结果A   非流水线场景：  1. 参考[软件迁移评估说明](https://www.hikunpeng.com/document/detail/zh/kunpengdevps/userguide/cliuserguide/KunpengDevKitCli_0010.html)，完成软件迁移评估，并将生成的检查报告（.html文件）重命名为SoftwareMigrationAssessment.html  2. 参考[源码迁移说明](https://www.hikunpeng.com/document/detail/zh/kunpengdevps/userguide/cliuserguide/KunpengDevKitCli_0013.html)，完成源码迁移，并将生成的检查报告（.html文件）重命名为SourceCodeScanningReport.html  3. 收集步骤1~2的检查报告（SoftwareMigrationAssessment.html, SourceCodeScanningReport.html）并一起打包成tar.gz压缩包  4. 参考[DevKit验收工具使用说明](https://gitee.com/openeuler/devkit-pipeline/blob/master/document/%E9%AA%8C%E6%94%B6%E6%8A%A5%E5%91%8A%E6%94%B6%E9%9B%86/devkit-pipeline%E7%9B%B8%E5%85%B3%E6%8A%A5%E5%91%8A%E9%AA%8C%E6%94%B6%E6%96%87%E6%A1%A3.md)，使用验收工具解析压缩包并生成验收报告，查看验收报告，有预期结果A |
| **预期结果** | [A] 验收报告迁移扫描工具评估结果显示通过 |
| **测试结果**  **（测试日志或截图）** | *请补充相关测试结果的截图* |
| **测试结论** | *通过/有条件通过/不通过* |
| **备注** | *若不通过或有条件通过，在此备注说明* |

* + 1. 门禁配置亲和扫描工具测试

门禁配置亲和扫描工具测试

|  |  |
| --- | --- |
| **用例名称** | 门禁配置亲和扫描工具测试 |
| **用例编号** | 03 |
| **测试目的** | 验证用户流水线已配置DevKit亲和扫描工具门禁 |
| **预置条件** | 流水线场景：   1. 参考[devkit-pipeline使用说明](https://www.hikunpeng.com/document/detail/zh/nativedevp/userguide/devkitpipepre/NativeDevelopment_0001.html)，完成流水线安装部署 2. 参考[创建流水线门禁检查任务](https://www.hikunpeng.com/document/detail/zh/nativedevp/userguide/pipeline/NativeDevelopment_0018.html)，完成DevKit亲和扫描工具（64位运行模式检查、字节对齐检查、内存一致性检查、向量化检查、构建亲和检查）门禁配置   非流水线场景：  参考[DevKit用户指南](https://www.hikunpeng.com/document/detail/zh/kunpengdevps/userguide/cliuserguide/KunpengDevKitCli_0001.html)，完成DevKit命令行工具安装 |
| **测试步骤** | 流水线场景：   1. 通过流水线发布产品包 2. 收集迁移扫描报告（64位运行模式检查报告、字节对齐检查报告、内存一致性检查报告、向量化检查报告、构建亲和检查报告）并一起打包成tar.gz压缩包 3. 参考[DevKit验收工具使用说明](https://gitee.com/openeuler/devkit-pipeline/blob/master/document/%E9%AA%8C%E6%94%B6%E6%8A%A5%E5%91%8A%E6%94%B6%E9%9B%86/devkit-pipeline%E7%9B%B8%E5%85%B3%E6%8A%A5%E5%91%8A%E9%AA%8C%E6%94%B6%E6%96%87%E6%A1%A3.md)，使用验收工具解析压缩包并生成验收报告，查看验收报告，有预期结果A   非流水线场景：  1. 参考[64位运行模式检查说明](https://www.hikunpeng.com/document/detail/zh/kunpengdevps/userguide/cliuserguide/KunpengDevKitCli_0021.html)，完成64位运行模式检查，并将生成的检查报告（.html文件）重命名为64-bit-running-mode-check.html  2. 参考[字节对齐检查说明](https://www.hikunpeng.com/document/detail/zh/kunpengdevps/userguide/cliuserguide/KunpengDevKitCli_0024.html)，完成字节对齐检查，并将生成的检查报告（.html文件）重命名为byte-alignment-check.html  3. 参考[内存一致性检查说明](https://www.hikunpeng.com/document/detail/zh/kunpengdevps/userguide/cliuserguide/KunpengDevKitCli_0027.html)，完成内存一致性检查，并将生成的检查报告（.html文件）重命名为memory-consistency-check.html  4. 参考[向量化检查说明](https://www.hikunpeng.com/document/detail/zh/kunpengdevps/userguide/cliuserguide/KunpengDevKitCli_0030.html)，完成向量化检查，并将生成的检查报告（.html文件）重命名为vectorized-check.html  5. 参考[构建亲和检查说明](https://www.hikunpeng.com/document/detail/zh/kunpengdevps/userguide/cliuserguide/KunpengDevKitCli_0041.html)，完成构建亲和检查，并将生成的检查报告（.html文件）重命名为build-affinity-check.html  6. 收集步骤1~5的检查报告（64-bit-running-mode-check.html, byte-alignment-check.html, memory-consistency-check.html, vectorized-check.html, build-affinity-check.html）并一起打包成tar.gz压缩包  7. 参考[DevKit验收工具使用说明](https://gitee.com/openeuler/devkit-pipeline/blob/master/document/%E9%AA%8C%E6%94%B6%E6%8A%A5%E5%91%8A%E6%94%B6%E9%9B%86/devkit-pipeline%E7%9B%B8%E5%85%B3%E6%8A%A5%E5%91%8A%E9%AA%8C%E6%94%B6%E6%96%87%E6%A1%A3.md)，使用验收工具解析压缩包并生成验收报告，查看验收报告，有预期结果A |
| **预期结果** | [A] 验收报告亲和扫描工具评估结果显示通过 |
| **测试结果**  **（测试日志或截图）** | *请补充相关测试结果的截图* |
| **测试结论** | *通过/有条件通过/不通过* |
| **备注** | *若不通过或有条件通过，在此备注说明* |

* 1. 编译构建测试
     1. 构建内核测试

构建内核测试

|  |  |
| --- | --- |
| **用例名称** | 构建内核测试 |
| **用例编号** | 04 |
| **测试目的** | 验证构建执行机OS内核为openEuler |
| **预置条件** | 构建执行机OS内核为openEuler |
| **测试步骤** | 1. 登录构建执行机 2. 进入产品代码目录，执行命令：pwd，有预期结果A 3. 执行命令：cat /etc/os-release，有预期结果B |
| **预期结果** | [A] 回显中能看到产品名称或产品代码仓名称  [B] 回显中有openEuler或执行机OS为openEuler系OS |
| **测试结果**  **（测试日志或截图）** | *请补充相关测试结果的截图* |
| **测试结论** | *通过/有条件通过/不通过* |
| **备注** | *若不通过或有条件通过，在此备注说明* |

* + 1. 鲲鹏亲和编译器安装测试

鲲鹏亲和编译器（毕昇编译器）测试

|  |  |
| --- | --- |
| **用例名称** | 鲲鹏亲和编译器（毕昇编译器）测试 |
| **用例编号** | 05 |
| **测试目的** | 验证构建执行机已正确安装毕昇编译器 |
| **预置条件** | 构建执行机已成功安装毕昇编译器 |
| **测试步骤** | 1. 登录构建执行机 2. 进入产品代码目录，执行命令：pwd，有预期结果A 3. 执行命令：clang --version或clang++ --version或flang --version，有预期结果B |
| **预期结果** | [A] 回显中能看到产品名称或产品代码仓名称  [B] 回显中有BiSheng |
| **测试结果**  **（测试日志或截图）** | *请补充相关测试结果的截图* |
| **测试结论** | *通过/有条件通过/不通过* |
| **备注** | *若不通过或有条件通过，在此备注说明* |

鲲鹏亲和编译器（毕昇JDK）测试

|  |  |
| --- | --- |
| **用例名称** | 鲲鹏亲和编译器（毕昇JDK）测试 |
| **用例编号** | 06 |
| **测试目的** | 验证构建执行机已正确安装毕昇JDK |
| **预置条件** | 构建执行机已成功安装毕昇JDK |
| **测试步骤** | 1. 登录构建执行机 2. 进入产品代码目录，执行命令：pwd，有预期结果A 3. 执行命令：java -version，有预期结果B |
| **预期结果** | [A] 回显中能看到产品名称或产品代码仓名称  [B] 回显中有BiSheng |
| **测试结果**  **（测试日志或截图）** | *请补充相关测试结果的截图* |
| **测试结论** | *通过/有条件通过/不通过* |
| **备注** | *若不通过或有条件通过，在此备注说明* |

鲲鹏亲和编译器（GCC for openEuler）测试

|  |  |
| --- | --- |
| **用例名称** | 鲲鹏亲和编译器（GCC for openEuler）测试 |
| **用例编号** | 07 |
| **测试目的** | 验证构建执行机已正确安装GCC for openEuler |
| **预置条件** | 构建执行机已成功安装GCC for openEuler |
| **测试步骤** | 1. 登录构建执行机 2. 进入产品代码目录，执行命令：pwd，有预期结果A 3. 执行命令：gcc --version或g++ --version或gfortran --version，有预期结果B |
| **预期结果** | [A] 回显中能看到产品名称或产品代码仓名称  [B] 回显中有gcc for openEuler字段    补充说明：  若使用的是欧拉系OS默认的gcc编译器，回显中无gcc for openEuler字段时，可通过以下方式判断：  场景1：gcc版本为7.3.0版本时  1. 执行命令gcc -v，回显中有gcc version 7.3.0  2. 执行下列命令：  gcc -mtune=native -E -v - </dev/null 2>&1 | grep cc1 | grep tsv110  返回结果如下即说明集成的gcc for openEuler：    场景2：gcc版本非7.3.0版本时  使用gcc -v 回显中有gcc version x.x.1版本，即说明集成的gcc for openEuler，如 openEuler 22.03操作系统下显示 gcc version 10.3.1。 |
| **测试结果**  **（测试日志或截图）** | *请补充相关测试结果的截图* |
| **测试结论** | *通过/有条件通过/不通过* |
| **备注** | *若不通过或有条件通过，在此备注说明* |

* 1. DevKit调优工具测试
     1. DevKit性能分析工具测试

DevKit性能分析工具测试

|  |  |
| --- | --- |
| **用例名称** | DevKit性能分析工具测试 |
| **用例编号** | 08 |
| **测试目的** | 验证用户已使用DevKit性能分析工具 |
| **预置条件** | 参考[鲲鹏DevKit开发套件](https://www.hikunpeng.com/developer/devkit/download)，完成鲲鹏DevKit安装（VS Code插件模式、浏览器工作模式、命令行模式，任选其一）。  根据不同的工作模式，选择合适的安装方式，完成性能分析工具（系统性能分析工具、Java性能分析工具、系统诊断工具）安装。 |
| **测试步骤** | 1. 登录鲲鹏DevKit，输入账号密码完成登录 2. 使用DevKit调优工具任意功能（系统性能分析工具、Java性能分析工具、系统诊断工具），完成业务性能调优 |
| **预期结果** | NA |
| **测试结果**  **（测试日志或截图）** | *提供DevKit调优工具使用情况说明，如DevKit调优报告日志或DevKit调优工具截图* |
| **测试结论** | *通过/有条件通过/不通过* |
| **备注** | *若不通过或有条件通过，在此备注说明* |

* 1. 鲲鹏平台测试用例测试
     1. 鲲鹏平台测试用例测试

鲲鹏平台测试用例测试

|  |  |
| --- | --- |
| **用例名称** | 鲲鹏平台测试用例测试 |
| **用例编号** | 09 |
| **测试目的** | 用户产品包能通过鲲鹏平台测试工具测试 |
| **预置条件** | 流水线场景：  参考[devkit-pipeline使用说明](https://www.hikunpeng.com/document/detail/zh/nativedevp/userguide/devkitpipepre/NativeDevelopment_0001.html)，完成流水线安装部署并集成配置DevKit测试平台  非流水线场景：  参考[DevKit测试平台安装指导](https://www.hikunpeng.com/document/detail/zh/nativedevp/userguide/pipeline/NativeDevelopment_0013.html)，单独部署测试平台 |
| **测试步骤** | 流水线场景：  1. 通过流水线发布产品包  2. 收集DevKit测试平台测试报告并一起打包成tar.gz压缩包  3. 将压缩包上传至验证平台，查看验收报告，有预期结果A  非流水线场景：  1. 参考[DevKit测试平台使用指导](https://www.hikunpeng.com/document/detail/zh/nativedevp/userguide/pipeline/NativeDevelopment_0043.html)，完成测试平台测试并生成测试报告  2. 收集步骤1中测试平台测试报告并打包成tar.gz压缩包  3. 参考[DevKit验收工具使用说明](https://gitee.com/openeuler/devkit-pipeline/blob/master/document/%E9%AA%8C%E6%94%B6%E6%8A%A5%E5%91%8A%E6%94%B6%E9%9B%86/devkit-pipeline%E7%9B%B8%E5%85%B3%E6%8A%A5%E5%91%8A%E9%AA%8C%E6%94%B6%E6%96%87%E6%A1%A3.md)，使用验收工具解析压缩包并生成验收报告，查看验收报告，有预期结果A |
| **预期结果** | [A] 验收报告中的如下测试用例都为pass  Compatibility\_Application\_Start  Compatibility\_Application\_Stop  Compatibility\_Hardware\_Server  Compatibility\_Software\_Name |
| **测试结果**  **（测试日志或截图）** | *请补充相关测试结果的截图* |
| **测试结论** | *通过/有条件通过/不通过* |
| **备注** | *若不通过或有条件通过，在此备注说明* |

* 1. 产品商用发布测试
     1. 产品商用发布测试

产品商用发布测试

|  |  |
| --- | --- |
| **用例名称** | 产品商用发布测试 |
| **用例编号** | 10 |
| **测试目的** | 证明本次鲲鹏版本发布是首发，即本次只发布了鲲鹏版本，未发布其他处理器版本；或鲲鹏版本与其他处理器版本同步发布，即本次鲲鹏版本发布不晚于其他处理器版本。 |
| **预置条件** | 产品已发布鲲鹏商用版本 |
| **测试步骤** | 1. 若是外网发布，提供外网发布链接，打开链接后，进行截图，有预期结果A。  2. 若是内网发布，请进入发布页面，进行截图，有预期结果B。 |
| **预期结果** | [A] 截图能查看到产品商用版本名称、版本号和发布日期。  [B] 截图包含产品发布的路径，版本名称，版本号和发布日期。 |
| **测试结果**  **（测试日志或截图）** | *请补充相关测试结果的截图* |
| **测试结论** | *通过/有条件通过/不通过* |
| **备注** | *若不通过或有条件通过，在此备注说明* |

* 1. 流水线工具
     1. 流水线工具测试

流水线工具测试

|  |  |
| --- | --- |
| **用例名称** | 流水线工具测试 |
| **用例编号** | 11 |
| **测试目的** | 证明鲲鹏流水线，是基于Jenkins或gitlab等开源工具，或伙伴自研等自动化工具搭建。 |
| **预置条件** | X86平台流水线是基于Jenkins或gitlab或伙伴自研等自动化工具搭建 |
| **测试步骤** | 1. 若是X86平台流水线基于Jenkins搭建，结合鲲鹏原生开发工程，请搜集鲲鹏流水线工具，进行截图，有预期结果A。  2. 若是X86平台流水线基于gitlab搭建，结合鲲鹏原生开发工程，请搜集鲲鹏流水线工具，进行截图，有预期结果B。  3. 若是X86平台流水线基于Jenkins、gitlab之外的流水线工具（假定工具名称Tool\_A）搭建，结合鲲鹏原生开发工程，请搜集鲲鹏流水线工具，进行截图，有预期结果C。 |
| **预期结果** | [A] 截图能查看到鲲鹏流水线工具是基于Jenkins，以及Jenkins版本号。        [B] 截图能查看到鲲鹏流水线工具是基于gitlab，以及gitlab版本号。          [C] 截图能查看到鲲鹏流水线工具是基于Tool\_A，以及Tool\_A版本号。 |
| **测试结果**  **（测试日志或截图）** | *请补充相关测试结果的截图* |
| **测试结论** | *通过/有条件通过/不通过* |
| **备注** | *若不通过或有条件通过，在此备注说明* |

* 1. Boostkit调优测试

**执行结果只保留涉及场景，其他不涉及场景请删除**

* + 1. 大数据解决方案测试

大数据使能解决方案测试

|  |  |
| --- | --- |
| **用例名称** | 大数据使能解决方案测试 |
| **用例编号** | 12 |
| **测试目的** | 验证用户已使用大数据使能解决方案特性 |
| **预置条件** | 1、执行机已配置使用大数据使能解决方案特性  2、已下载BoostKit采集工具（[下载链接](https://gitee.com/openeuler/devkit-pipeline/releases/tag/v-1)），并按照[BoostKit验收工具使用说明](https://gitee.com/openeuler/devkit-pipeline/blob/master/tools/Boostkit_ISV_Tool.md)，修改采集工具配置文件，开启大数据相关特性使能信息收集功能并完成相关参数配置 |
| **测试步骤** | 1. 运行采集工具生成统一压缩包（压缩包包含客户id标识信息以及打包时间信息）  2. 压缩包上传至验证平台，查看验收报告，有预期结果A |
| **预期结果** | [A] 使能评估成功项包含“大数据【通过】” |
| **测试结果**  **（测试日志或截图）** | *请补充相关测试结果的截图* |
| **测试结论** | *通过/有条件通过/不通过* |
| **备注** | *若不通过或有条件通过，在此备注说明* |

* + 1. 分布式存储解决方案测试

分布式存储使能解决方案测试

|  |  |
| --- | --- |
| **用例名称** | 分布式存储使能解决方案测试 |
| **用例编号** | 13 |
| **测试目的** | 验证用户已使用分布式存储使能解决方案特性 |
| **预置条件** | 1、执行机已配置使用分布式存储使能解决方案特性  2、已下载BoostKit采集工具（[下载链接](https://gitee.com/openeuler/devkit-pipeline/releases/tag/v-1)），并按照[BoostKit验收工具使用说明](https://gitee.com/openeuler/devkit-pipeline/blob/master/tools/Boostkit_ISV_Tool.md)，修改采集工具配置文件，开启分布式存储相关特性使能信息收集功能并完成相关参数配置 |
| **测试步骤** | 1. 运行采集工具生成统一压缩包（压缩包包含客户id标识信息以及打包时间信息）  2. 压缩包上传至验证平台，查看验收报告，有预期结果A |
| **预期结果** | [A] 使能评估成功项包含“分布式存储【通过】” |
| **测试结果**  **（测试日志或截图）** | *请补充相关测试结果的截图* |
| **测试结论** | *通过/有条件通过/不通过* |
| **备注** | *若不通过或有条件通过，在此备注说明* |

* + 1. 数据库解决方案测试

数据库使能解决方案测试

|  |  |
| --- | --- |
| **用例名称** | 数据库使能解决方案测试 |
| **用例编号** | 14 |
| **测试目的** | 验证用户已使用数据库使能解决方案特性 |
| **预置条件** | 1、执行机已配置使用数据库使能解决方案特性  2、已下载BoostKit采集工具（[下载链接](https://gitee.com/openeuler/devkit-pipeline/releases/tag/v-1)），并按照[BoostKit验收工具使用说明](https://gitee.com/openeuler/devkit-pipeline/blob/master/tools/Boostkit_ISV_Tool.md)，修改采集工具配置文件，开启数据库相关特性使能信息收集功能并完成相关参数配置 |
| **测试步骤** | 1. 运行采集工具生成统一压缩包（压缩包包含客户id标识信息以及打包时间信息）  2. 压缩包上传至验证平台，查看验收报告，有预期结果A |
| **预期结果** | [A] 使能评估成功项包含“数据库【通过】” |
| **测试结果**  **（测试日志或截图）** | *请补充相关测试结果的截图* |
| **测试结论** | *通过/有条件通过/不通过* |
| **备注** | *若不通过或有条件通过，在此备注说明* |

* + 1. ARM原生解决方案测试

ARM原生使能解决方案测试

|  |  |
| --- | --- |
| **用例名称** | ARM原生使能解决方案测试 |
| **用例编号** | 15 |
| **测试目的** | 验证用户已使用ARM原生使能解决方案特性 |
| **预置条件** | 1、执行机已配置使用ARM原生使能解决方案特性  2、已下载BoostKit采集工具（[下载链接](https://gitee.com/openeuler/devkit-pipeline/releases/tag/v-1)），并按照[BoostKit验收工具使用说明](https://gitee.com/openeuler/devkit-pipeline/blob/master/tools/Boostkit_ISV_Tool.md)，修改采集工具配置文件，开启ARM原生相关特性使能信息收集功能并完成相关参数配置 |
| **测试步骤** | 1. 运行采集工具生成统一压缩包（压缩包包含客户id标识信息以及打包时间信息）  2. 压缩包上传至验证平台，查看验收报告，有预期结果A |
| **预期结果** | [A] 使能评估成功项包含“ARM原生【通过】” |
| **测试结果**  **（测试日志或截图）** | *请补充相关测试结果的截图* |
| **测试结论** | *通过/有条件通过/不通过* |
| **备注** | *若不通过或有条件通过，在此备注说明* |

* + 1. 加速库解决方案测试

加速库使能解决方案测试

|  |  |
| --- | --- |
| **用例名称** | 加速库使能解决方案测试 |
| **用例编号** | 16 |
| **测试目的** | 验证用户已使用加速库使能解决方案特性 |
| **预置条件** | 1、执行机已配置使用加速库使能解决方案特性  2、已下载BoostKit采集工具（[下载链接](https://gitee.com/openeuler/devkit-pipeline/releases/tag/v-1)），并按照[BoostKit验收工具使用说明](https://gitee.com/openeuler/devkit-pipeline/blob/master/tools/Boostkit_ISV_Tool.md)，修改采集工具配置文件，开启加速库相关特性使能信息收集功能并完成相关参数配置 |
| **测试步骤** | 1. 运行采集工具生成统一压缩包（压缩包包含客户id标识信息以及打包时间信息）  2. 压缩包上传至验证平台，查看验收报告，有预期结果A |
| **预期结果** | [A] 使能评估成功项包含“加速库【通过】” |
| **测试结果**  **（测试日志或截图）** | *请补充相关测试结果的截图* |
| **测试结论** | *通过/有条件通过/不通过* |
| **备注** | *若不通过或有条件通过，在此备注说明* |

* + 1. 虚拟化解决方案测试

虚拟化使能解决方案测试

|  |  |
| --- | --- |
| **用例名称** | 虚拟化使能解决方案测试 |
| **用例编号** | 17 |
| **测试目的** | 验证用户已使用虚拟化使能解决方案特性 |
| **预置条件** | 1、执行机已配置使用虚拟化使能解决方案特性  2、已下载BoostKit采集工具（[下载链接](https://gitee.com/openeuler/devkit-pipeline/releases/tag/v-1)），并按照[BoostKit验收工具使用说明](https://gitee.com/openeuler/devkit-pipeline/blob/master/tools/Boostkit_ISV_Tool.md)，修改采集工具配置文件，开启虚拟化相关特性使能信息收集功能并完成相关参数配置 |
| **测试步骤** | 1. 运行采集工具生成统一压缩包（压缩包包含客户id标识信息以及打包时间信息）  2. 压缩包上传至验证平台，查看验收报告，有预期结果A |
| **预期结果** | [A] 使能评估成功项包含“虚拟化【通过】” |
| **测试结果**  **（测试日志或截图）** | *请补充相关测试结果的截图* |
| **测试结论** | *通过/有条件通过/不通过* |
| **备注** | *若不通过或有条件通过，在此备注说明* |

* + 1. 机密计算解决方案测试

机密计算使能解决方案测试

|  |  |
| --- | --- |
| **用例名称** | 机密计算使能解决方案测试 |
| **用例编号** | 18 |
| **测试目的** | 验证用户已使用机密计算使能解决方案特性 |
| **预置条件** | 1、执行机已配置使用机密计算使能解决方案特性  2、已下载BoostKit采集工具（[下载链接](https://gitee.com/openeuler/devkit-pipeline/releases/tag/v-1)），并按照[BoostKit验收工具使用说明](https://gitee.com/openeuler/devkit-pipeline/blob/master/tools/Boostkit_ISV_Tool.md)，修改采集工具配置文件，开启机密计算相关特性使能信息收集功能并完成相关参数配置 |
| **测试步骤** | 1. 运行采集工具生成统一压缩包（压缩包包含客户id标识信息以及打包时间信息）  2. 压缩包上传至验证平台，查看验收报告，有预期结果A |
| **预期结果** | [A] 使能评估成功项包含“机密计算【通过】” |
| **测试结果**  **（测试日志或截图）** | *请补充相关测试结果的截图* |
| **测试结论** | *通过/有条件通过/不通过* |
| **备注** | *若不通过或有条件通过，在此备注说明* |

* + 1. HPC解决方案测试

HPC使能解决方案测试

|  |  |
| --- | --- |
| **用例名称** | HPC使能解决方案测试 |
| **用例编号** | 19 |
| **测试目的** | 验证用户已使用HPC使能解决方案特性 |
| **预置条件** | 1、执行机已配置使用HPC使能解决方案特性  2、已下载BoostKit采集工具（[下载链接](https://gitee.com/openeuler/devkit-pipeline/releases/tag/v-1)），并按照[BoostKit验收工具使用说明](https://gitee.com/openeuler/devkit-pipeline/blob/master/tools/Boostkit_ISV_Tool.md)，修改采集工具配置文件，开启HPC相关特性使能信息收集功能并完成相关参数配置 |
| **测试步骤** | 1. 运行采集工具生成统一压缩包（压缩包包含客户id标识信息以及打包时间信息）  2. 压缩包上传至验证平台，查看验收报告，有预期结果A |
| **预期结果** | [A] 使能评估成功项包含“HPC【通过】” |
| **测试结果**  **（测试日志或截图）** | *请补充相关测试结果的截图* |
| **测试结论** | *通过/有条件通过/不通过* |
| **备注** | *若不通过或有条件通过，在此备注说明* |

# 修订记录

| **文档版本** | **发布日期** | **修改说明** |
| --- | --- | --- |
| 1.0 | 2024-3-21 | 第1次正式发布 |