普天同签
Signatory Chain

软件使用手册

产品名称：区块链功能测试手册

版本：V1.0

杭州玺湖科技有限公司

目录

1. 公司简介

杭州玺湖科技有限公司全球首创基于区块链去中心化或多中心化多根共识共治的数字安全框架和底层技术（DeSe: Decentralized Security），是 60 年以来全球数字安全领域的一场框架性革命，也是区块链和实体经济，特别是区块链和数字安全领域结合的独特场景和接入点：除了区块链用于数据信任背书，还进一步将区块链思维内植入安全管理的内在逻辑。公司独创了一个崭新的蓝海，在该领域全球没有竞争对手，可以主导该领域的行业标准和话语权。目前所有其他现存的安全管理框架都是基于中心化等级式单根治理。我们的核心技术克服了当代中心化等级式数字系统安全管理的框架性漏洞和系统性风险，用区块链思维，搭建去中心化，多根共治扁平的数字系统安全管理框架和标准。公司所研发的颠覆性技术是一个普遍的底层方案，可应用在数字化的所有层级和行业：硬件，操作系统，数据库，中间件，应用层等，以及所有行业和场景，包括办公，农业，医疗，教育，食品安全，国防等。具有一个 2 万亿的左右的安全升级市场，10 万亿左右的区块链落地实体经济的市场。公司的核心技术可以通过柔性拥抱的方式，对现有的安全系统无感无缝升级，不需要修改现有的系统，极大地降低了安全升级成本。

2. 产品的名称，目的，和版本

2.1 产品名称

区块链功能测试手册

2.2 产品的目标

本手册旨在帮助用户顺利完成区块链国家标准（GB/T 42752-2023）所要求的功能测试，确保区块链系统符合相关规范和标准。

2.3 产品的版本

版本号：V1.0

3. 产品的主要功能模块和流程

3.1 检测环境准备

在开始测试之前，必须准备好相关的检测环境，确保所有硬件和软件资源符合测试要求。包括区块链网络的搭建、节点配置、链码部署等准备工作。

3.2 测试案例

编写并执行一系列测试用例，验证区块链系统是否符合《区块链和分布式记账技术 参考架构》中的要求。这些用例涵盖了系统的各项功能，包括但不限于网络连接、数据存储、共识机制等关键模块。

4. 每个功能模块的使用方法

4.1 检测环境准备

检 测 环 境			
序号	硬件配置		软件配置
	设备类型	配置/性能参数	软件名称及版本号
数据库/应用服务器（台）			
1	Peer 节点	CPU3.0 GHz/内存 2G/磁盘 40G	Fabric1.4.8
2	Peer 节点	CPU3.0 GHz/内存 2G/磁盘 40G	Fabric1.4.8
3	Peer 节点	CPU3.0 GHz/内存 2G/磁盘 40G	Fabric1.4.8
4	Peer 节点	CPU3.0 GHz/内存 2G/磁盘 40G	Fabric1.4.8
5	Peer 节点	CPU3.0 GHz/内存 2G/磁盘 40G	Fabric1.4.8
6	Peer 节点	CPU3.0 GHz/内存 2G/磁盘 40G	Fabric1.4.8

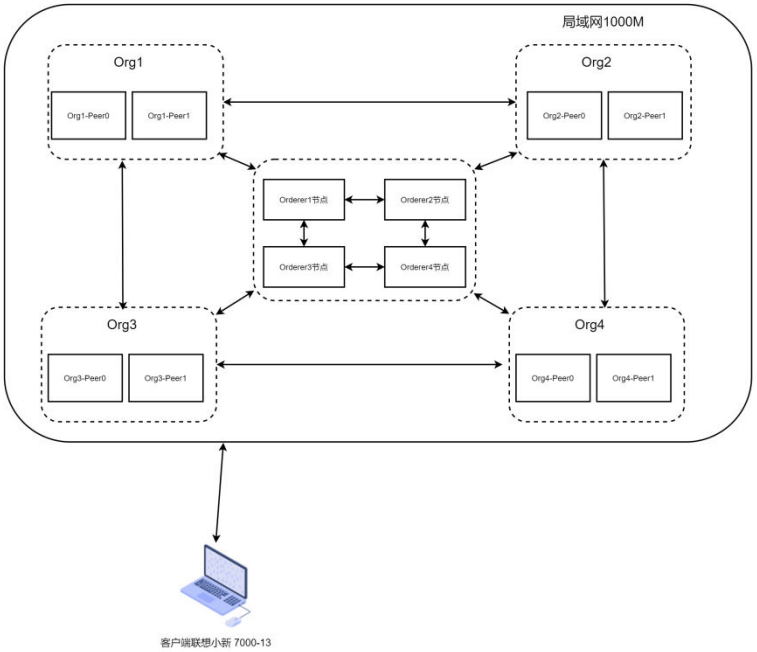
7	Peer 节点	CPU3.0 GHz/内存 2G/磁盘 40G	Fabric1.4.8
8	Peer 节点	CPU3.0 GHz/内存 2G/磁盘 40G	Fabric1.4.8
9	Orderer 节点	CPU3.0 GHz/内存 2G/磁盘 40G	Fabric1.4.8
10	Orderer 节点	CPU3.0 GHz/内存 2G/磁盘 40G	Fabric1.4.8
11	Orderer 节点	CPU3.0 GHz/内存 2G/磁盘 40G	Fabric1.4.8
12	Orderer 节点	CPU3.0 GHz/内存 2G/磁盘 40G	Fabric1.4.8
客户端			
1	Windows	CPU1.8 GHz/内存 8G	Windows10
测试工具			

1	名称	版本号	生产商	运行方式
	——	——	——	——

网络环境			
1	局域网	1000M 带宽	Linux
其他辅助设备			
——	——	——	——

【注】：功能测试无需测试工具，测试环境由委托方提供。

拓扑结构图



网络环境

192.168.101.103 orderer1.signatorychain.com
192.168.101.105 orderer2.signatorychain.com
192.168.101.124 orderer3.signatorychain.com
192.168.101.136 orderer4.signatorychain.com

192.168.101.103 peer0.org1.signatorychain.com
192.168.101.103 peer1.org1.signatorychain.com

192.168.101.105 peer0.org2.signatorychain.com
192.168.101.105 peer1.org2.signatorychain.com

192.168.101.124 peer0.org3.signatorychain.com
192.168.101.124 peer1.org3.signatorychain.com

192.168.101.136 peer0.org4.signatorychain.com
192.168.101.136 peer1.org4.signatorychain.com

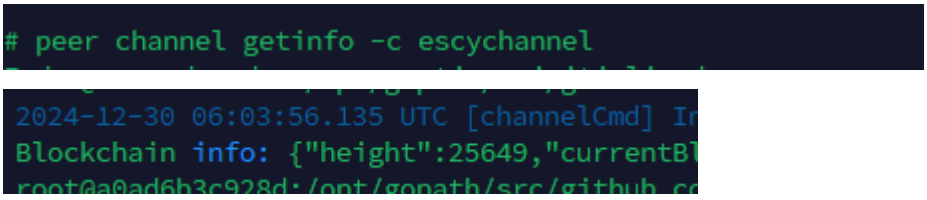
4.2 测试案例

1　区块链功能测试用例

1.1 BC-T101：用户功能测试

1.1.1　命令行交互功能

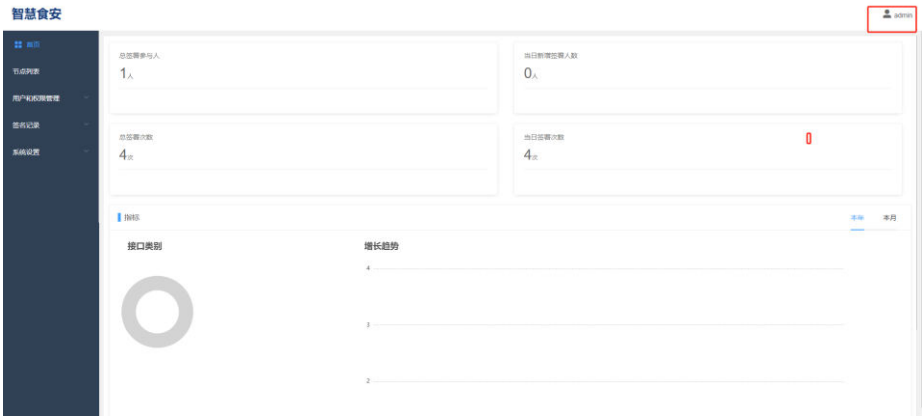
用例编号	BC-T101-C1	**用例名称**	命令行交互功能
用例类型	必选		
测试目标	检验用户功能的命令行交互功能。		
前置条件	1）通过授权帐号已成功开启系统用户层。		
测试过程	**步骤描述（参考）**	**输入数据**	

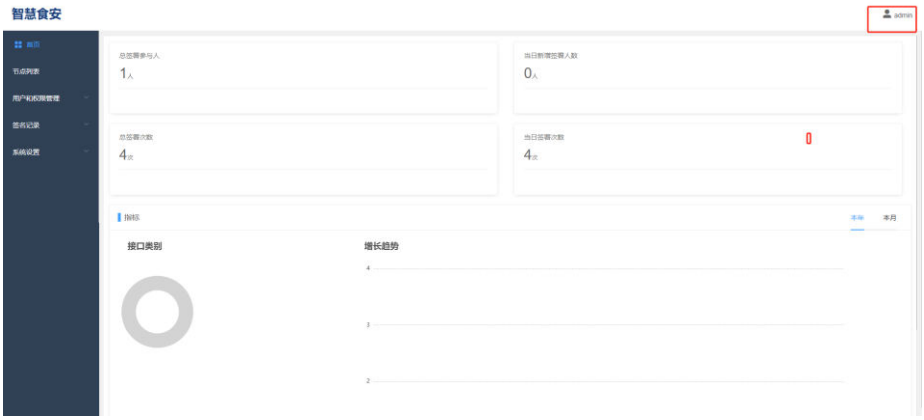
	1) 进入用户界面； 2) 以送测产品的用户功能列表为准， 测试命令行指令可执行性；	
预期结果	命令行指令功能交互完全	
测试结果	通过	
备注	利用命令行查询当前区块高度并返回结果 	
测试人		**测试时间** yyyy.mm.dd

1.1.2 图形交互功能

用例编号	BC-T101-C2	**用例名称**	图形交互功能
用例类型	可选		
测试目标	检验用户功能中的应用程序交互功能。		
前置条件	1) 通过授权帐号已成功开启系统用户层。		
测试过程	**步骤描述（参考）**	**输入数据**	
	1) 进入用户界面； 2) 以送测产品的用户功能列表为准， 测试图形交互功能；		
预期结果	图形交互功能完全		
测试结果	通过/此选项不支持		
备注	管理员用户登录界面		

<table>
<tr><td></td><td> 

登录成功界面



</td></tr>
<tr><td>测试人</td><td></td><td>测试时间</td><td></td></tr>
</table>

1.1.3 应用程序接口交互功能

用例编号	BC-T101-C3	**用例名称**	应用程序接口交互功能
用例类型	可选		
测试目标	以送测产品的用户功能列表为准，检验应用程序接口交互可执行性		
前置条件	1） 通过授权帐号已成功开启系统用户层。		
测试过程	**步骤描述（参考）**	**输入数据**	
	1） 进入用户界面； 2） 测试应用程序接口交互功能（可选）		
预期结果	应用程序接口交互功能完全		
测试结果	通过/此选项不支持		
备注	接口主要实现用区块链为被追溯的对象(数据，设备等)提供可信化升级:保真，确权，保密和溯源		

	目录 1. 安全升级策略及接口对接说明 ..... 3 1.1 无缝无感升级 ..... 3 1.2 切割和黑箱 ..... 3 1.3 跳转和回跳 ..... 3 1.4 私有化部署 ..... 4 1.5 接口概述 ..... 4 1.6 接口环境说明 ..... 4 1.7 接口请求方式 ..... 5 1.8 接口请求公共参数 ..... 5 1.9 接口请求响应示例 ..... 6 1.10 接口状态码说明 ..... 7 1.11 接口调用注意事项 ..... 9 1.12 接口更新说明 ..... 9 1.13 多根共治升级的颗粒度 ..... 9 2.框架接口 ..... 10 2.1 区块链保真接口 ..... 10 2.2 区块链原产地认证接口 ..... 13 2.3 区块链流转节点脚印的记录和认证接口 ..... 17 2.4 区块链确权接口 ..... 20 2.5 区块链安全证书 ..... 22 2.6 区块链原产地认证验证接口 ..... 25 3 字典值域列表 ..... 28 3.1 代码字典说明 28

测试人		**测试时间**	

1.1.4 事务提交

用例编号	BC-T101-C4	**用例名称**	事务提交功能
用例类型	必选		
测试目标	检测是否支持发起事务（支付）操作		
前置条件	通过授权帐号已成功开启系统用户功能。		
测试过程	**步骤描述（参考）**	**输入数据**	
	进入用户界面； 发起事务（支付）操作。	无	
预期结果	测试事务返回（账户余额是否一致）正确		

测试结果	通过		
备注	root@a0ad6b3c928d:/opt/gopath/src/github.com/hyperledger/fabric/peer# peer chaincode invoke -o ${ORDERER_ADDRESS} -C ${CHANNEL_NAME} -n ${CC_NAME} -c '{"Args":["publishAsset","waste_oil_id-22122221","User1","Wasteoil","100","desc"]}' --tls --cafile ${ORDERER_CA} --peerAddresses peer0.org1.signatorychain.com:7051 --tlsRootCertFiles ${P0_O1_CRT} --peerAddresses peer0.org2.signatorychain.com:7051 --tlsRootCertFiles ${P0_O2_CRT} --peerAddresses peer0.org3.signatorychain.com:7051 --tlsRootCertFiles ${P0_O3_CRT} --peerAddresses peer0.org4.signatorychain.com:7051 --tlsRootCertFiles ${P0_O4_CRT} 通过命令行数据上链 2024-12-30 06:15:07.356 UTC [chaincodeCmd] chaincodeInvokeOrQuery -> INFO 001 Chaincode invoke successful. result: status:200 payload:"{\"transaction_hash\":\"seconds:1735539307 nanos:315167727 \",\"transaction_id\":\"25d49e67ade28d83e6daf0d144a5082978ebb6269fad133454e9325ff528f952\"}" 数据成功上链		
测试人		**测试时间**	

1.2 BC-T102：业务功能测试

1.2.1　区块链服务选择

用例编号	BC-T102-C1	**用例名称**	区块链服务选择
用例类型	可选		
测试目标	比照业务功能列表，能选择正确的区块链业务功能。		
前置条件	1）通过授权帐号已成功开启系统。		
测试过程	**步骤描述（参考）**	**输入数据**	
	1）进入业务界面； 2）以送测产品的用户功能列表为准，随机选择区块链业务功能；	无	
预期结果	业务功能选择后正确执行		
测试结果	通过/此选项不支持		
备注			

测试人		**测试时间**	

1.2.2 区块链服务订购

用例编号	BC-T102-C2	**用例名称**	区块链服务订购功能
用例类型	可选		
测试目标	检验是否能订购区块链服务。		
前置条件	1）通过授权帐号已成功开启系统。		
测试过程	**步骤描述（参考）**	**输入数据**	
	1）进入业务界面； 2）订购区块链系统功能列表，并执行区块链功能列表的内容；	无	
预期结果	业务功能交互完全		
测试结果	通过/此选项不支持		
备注			
测试人		**测试时间**	

1.2.3 使用区块链账务

用例编号	BC-T102-C3	**用例名称**	使用区块链账务功能
用例类型	可选		
测试目标	比照业务功能列表，是否能使用区块链账务，根据实际业务运行。		
前置条件	1）通过授权帐号已成功开启系统。		
测试过程	**步骤描述（参考）**	**输入数据**	

	1) 进入业务界面； 2) 向区块链账本中写入数据； 3) 通过不同节点查询账务信息。	无
预期结果	1) 两个以上节点成功查询到账务信息； 2) 节点间的账务信息一致	
测试结果	通过/此选项不支持	
备注		
测试人		**测试时间**

1.2.4 财务管理

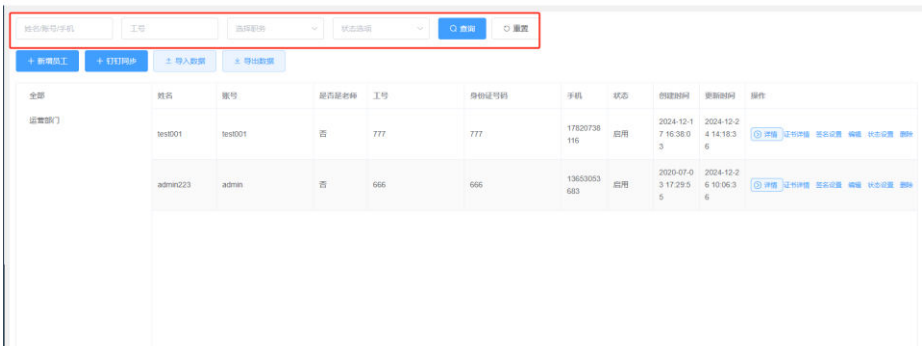
用例编号	BC-T102-C4	**用例名称**	区块链财务管理功能
用例类型	可选		
测试目标	比照业务功能列表，是否能使用区块链财务管理功能，根据实际业务运行。		
前置条件	1) 通过授权帐号已成功开启系统。		
测试过程	**步骤描述（参考）**		**输入数据**
	1) 进入业务界面； 2) 开启业务功能列表，向节点写入不同的数据； 3) 检测不同功能节点读取或写入数据是否符合预期		无
预期结果	节点数据读取和写入正确；		
测试结果	通过/此选项不支持		

备注			
测试人		**测试时间**	

1.3 BC-T103：管理功能

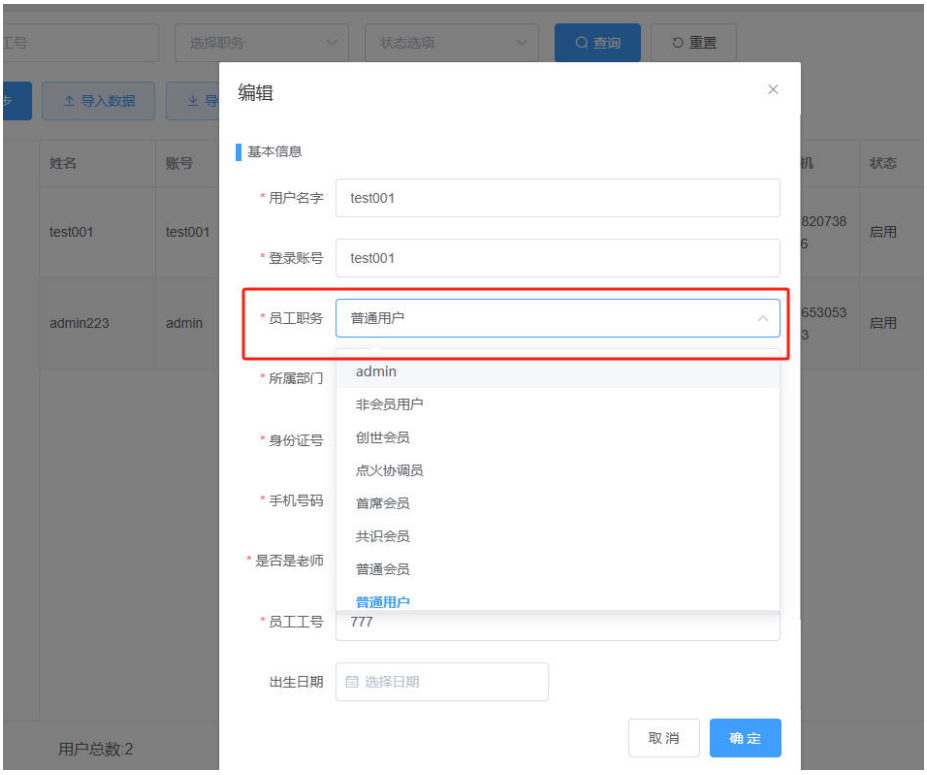
1.3.1　身份管理

用例编号	BC-T103-C1	**用例名称**	身份管理
用例类型	必选		
测试目标	检测是否支持在系统中对成员身份增加和查询		
前置条件	1）通过授权帐号已成功开启系统		
测试过程	**步骤描述**	**输入数据**	
	1）进入系统； 2）在其中对成员身份的增加和查询；	无	
预期结果	身份管理操作正确执行		
测试结果	通过		
备注	系统支持对成员身份的增加  系统支持对成员身份的查询		

	
测试人	**测试时间**

1.3.2　权限管理

用例编号	BC-T103-C2	**用例名称**	权限管理
用例类型	必选		
测试目标	检测是否支持对成员的权限进行设置，做不同级别操作		
前置条件	1) 通过授权帐号已成功开启系统		
测试过程	**步骤描述**	**输入数据**	
	1) 进入系统 2) 对成员的权限进行设置，做不同级别操作；	无	
预期结果	低级权限成员不能做高级操作；		
测试结果	通过		
备注	系统支持对成员进行权限设置		



管理员身份支持查询节点列表，用户和权限管理，系统设置

	 普通用户仅支持查询签名记录

			
测试人		**测试时间**	

1.3.3 数据保密

用例编号	BC-T103-C3	**用例名称**	数据保密
用例类型	必选		
测试目标	直接查看数据库中的用户数据，检测是否支持数据保密		
前置条件	1) 通过授权帐号已成功开启系统		
测试过程	**步骤描述**	**输入数据**	
	1) 进入系统； 2) 直接查看数据库中的用户数据；	无	
预期结果	用户数据不使用明文		
测试结果	通过		
备注	该平台支持管理员查询用户明文数据		

	 普通用户只能查询加密后的数据 
测试人	**测试时间**

1.3.4 可审计功能

用例编号	BC-T103-C4	**用例名称**	可审计功能
用例类型	可选		
测试目标	检测是否支持配置审计角色，并以审计角色核查区块链数据		
前置条件	1) 通过授权帐号已成功开启系统		
测试过程	**步骤描述**	**输入数据**	

	1) 进入系统； 2) 配置审计角色，并以审计角色审查区块链数据。	无
预期结果	具备审计角色标识、审计权限设置及审计结果可反馈为书面报告	
测试结果	通过/此选项不支持	
备注		
测试人		**测试时间**

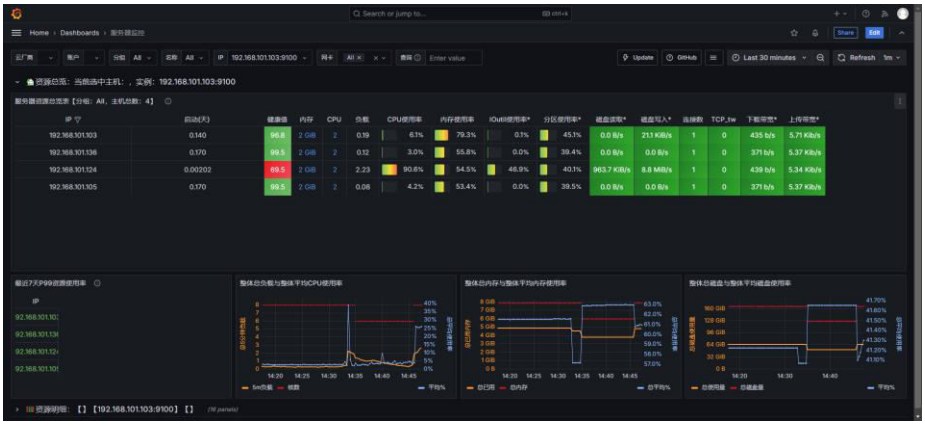
1.3.5 故障监测

用例编号	BC-T103-C5	**用例名称**	故障监测
用例类型	必选		
测试目标	查看故障监测仪表盘，制造故障（断开网络或者节点），查看仪表盘是否响应		
前置条件	通过授权帐号已成功开启系统监控管理功能。		
测试过程	**步骤描述（参考）**		**输入数据**
	1) 进入监控管理功能界面； 2) 查看故障监测仪表盘，制造故障（断开网络或者节点），查看仪表盘响应； 3) 查看系统监控图表。		无
预期结果	有对系统性能的监控及系统故障解决方案（文档或触发机制）		
测试结果	通过		

备注	 当共识节点断连时监控界面会显示		
测试人		**测试时间**	

1.3.6 网络运行状态监控

用例编号	BC-T103-C6	**用例名称**	网络运行状态监控
用例类型	可选		
测试目标	检测是否支持查看系统监控图表		
前置条件	1) 通过授权帐号已成功开启系统监控管理功能		
测试过程	**步骤描述（参考）**	**输入数据**	
	1) 进入系统监控图表 2) 具有系统的网络监控数据可视化图表（比如流量、节点数目、延迟等）	无	
预期结果	有系统的网络监控数据可视化图表（比如流量、节点数目、延迟等）		
测试结果	通过/此选项不支持		
备注	系统支持查看监控图表		

	
测试人	**测试时间**

1.3.7 预定义事件功能

用例编号	BC-T103-C7	**用例名称**	预定义事件管理功能
用例类型	必选		
测试目标	检测是否支持问题和信息安全事件管理		
前置条件	1）系统具有事件管理功能		
测试过程	**步骤描述（参考）**		**输入数据**
	1）进入事件管理功能界面； 2）有预定义事件功能的列表； 3）通过新增事件验证自定义事件。		无
预期结果	有预定义事件功能的列表。		
测试结果	通过		
备注	内存监控异常，支持以邮件的方式推送		

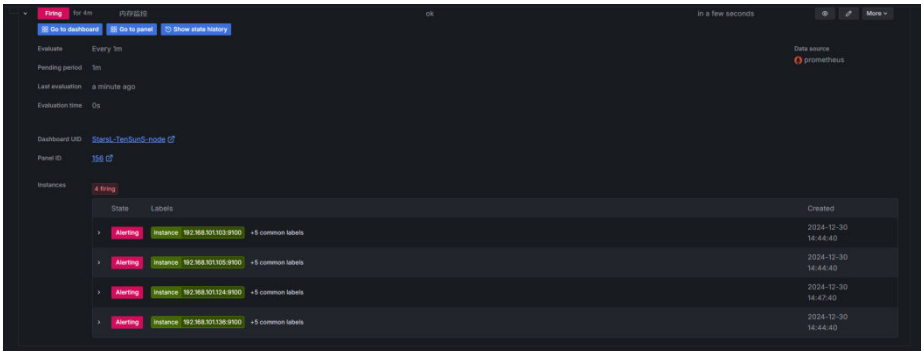
	
测试人	**测试时间**

1.3.8 自定义事件功能

用例编号	BC-T103-C8	**用例名称**	自定义事件管理功能
用例类型	可选		
测试目标	以区块链服务合作方定义事件为准，检测是否支持自定义事件功能		
前置条件	1）系统具有事件管理功能		
测试过程	**步骤描述（参考）**	**输入数据**	
	1）进入事件管理功能界面； 2）有自定义事件功能的列表； 3）通过新增事件验证自定义事件。	无	
预期结果	1）有自定义事件功能的列表； 2）通过新增事件验证自定义事件。		
测试结果	通过/此选项不支持		
备注			

测试人		**测试时间**	

1.3.9 问题管理

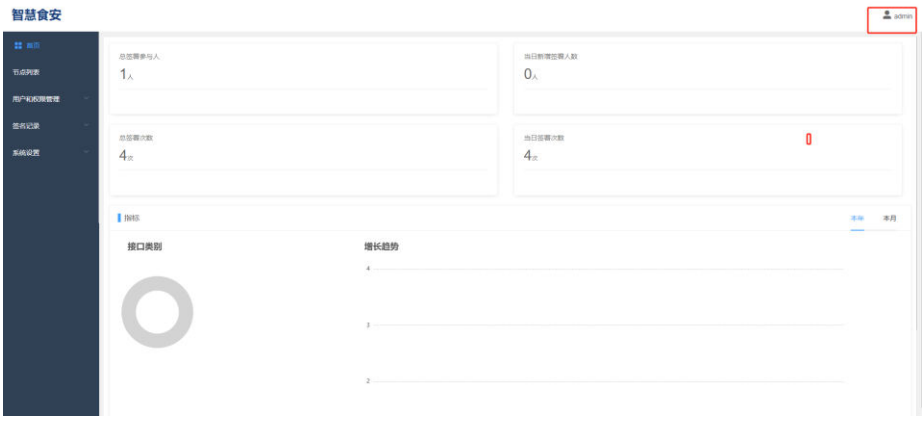
用例编号	BC-T103-C9	**用例名称**	问题管理
用例类型	必选		
测试目标	检验是否具有网络问题跟踪及报告。		
前置条件	已存在正在运行的系统		
测试过程	**步骤描述（参考）**	**输入数据**	
	1) 进入网络问题管理界面； 2) 测试系统对出现问题的响应。	无	
预期结果	系统提供客户服务响应的机制或功能，可查看相应报告		
测试结果	通过		
备注	系统支持查看内存问题的跟踪与报告 		
测试人		**测试时间**	

1.3.10 安全管理

用例编号	BC-T103-C10	**用例名称**	安全管理
用例类型	必选		

<table>
<tr><td>测试目标</td><td colspan="2">检验是否有账号安全功能。</td></tr>
<tr><td>前置条件</td><td colspan="2">1) 系统已经启动运行；</td></tr>
<tr><td rowspan="2">测试过程</td><td>步骤描述（参考）</td><td>输入数据</td></tr>
<tr><td>1) 进入系统；
2) 查看文档和账户存储情况</td><td>无</td></tr>
<tr><td>预期结果</td><td colspan="2">1) 有账号具备安全机制并应用在系统中；
2) 账号信息被加密保护。</td></tr>
<tr><td>测试结果</td><td colspan="2">通过</td></tr>
<tr><td>备注</td><td colspan="2">该平台需要输入正确的账号密码才可以登录
 

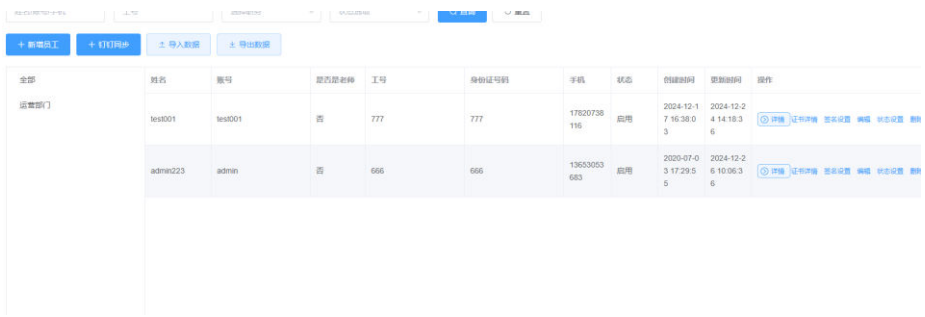
登录成功界面</td></tr>
</table>

	
测试人	**测试时间**

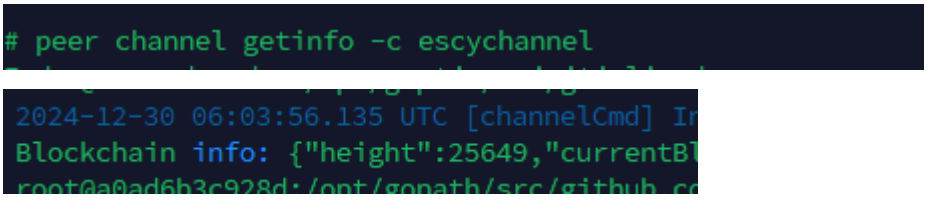
1.4 BC-T104：接入管理功能

1.4.1　账户信息查询

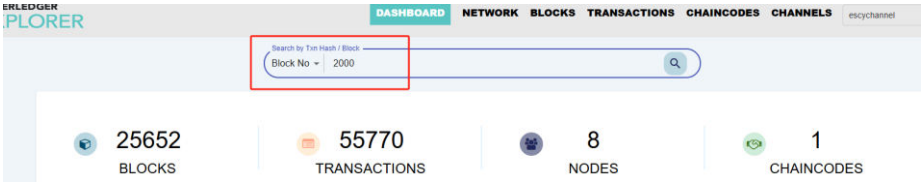
用例编号	BC-T104-C1	**用例名称**	账户信息查询
用例类型	必选		
测试目标	检测是否支持通过查询工具或查询接口查询指定账户的状态及基本信息		
前置条件	1)　已有系统；		
测试过程	**步骤描述（参考）**	**输入数据**	
	1)　进入系统； 2)　通过查询工具或查询接口查询指定帐户的状态及基本信息。	无	
预期结果	1)　测试过程能够正常完成，不出现报警，错误等异常； 2)　观察返回的帐户状态和基本信息。		
测试结果	通过		
备注	系统支持前端查询用户状态及基本信息		

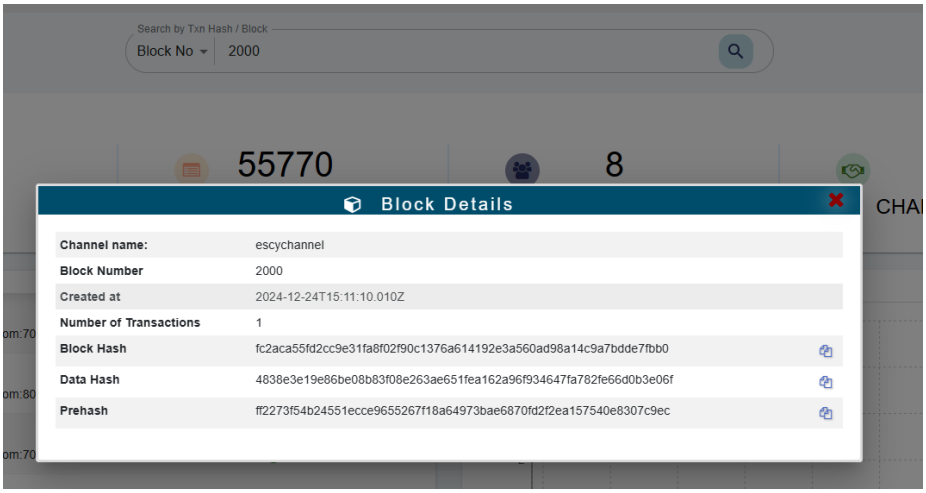
	
测试人	**测试时间**

1.4.2 区块总高度查询服务

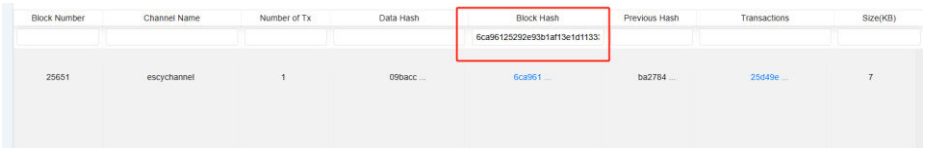
用例编号	BC-T104-C2	**用例名称**	区块总高度查询服务
用例类型	必选		
测试目标	检测是否支持通过查询工具或查询接口查询区块总高度信息		
前置条件	1) 已经进入系统；		
测试过程	**步骤描述（参考）**	**输入数据**	
	通过查询工具或查询接口查询区块总高度信息；	无	
预期结果	能够查到正确的区块总高度；		
测试结果	通过		
备注	利用命令行查询当前区块高度并返回结果 		
测试人		**测试时间**	

1.4.3 指定高度区块信息查询

<table>
<tr><td>用例编号</td><td>BC-T104-C3</td><td>用例名称</td><td>指定高度区块查询服务</td></tr>
<tr><td>用例类型</td><td colspan="3">必选</td></tr>
<tr><td>测试目标</td><td colspan="3">检测是否支持通过查询工具或查询接口查询指定高度区块的信息</td></tr>
<tr><td>前置条件</td><td colspan="3">1) 已经进入系统；</td></tr>
<tr><td rowspan="2">测试过程</td><td colspan="2">步骤描述（参考）</td><td>输入数据</td></tr>
<tr><td colspan="2">1) 通过查询工具或查询接口查询指定高度区块的信息；</td><td>无</td></tr>
<tr><td>预期结果</td><td colspan="3">能查到指定的区块数据信息</td></tr>
<tr><td>测试结果</td><td colspan="3">通过</td></tr>
<tr><td>备注</td><td colspan="3">查询第 2000 区块的信息
 

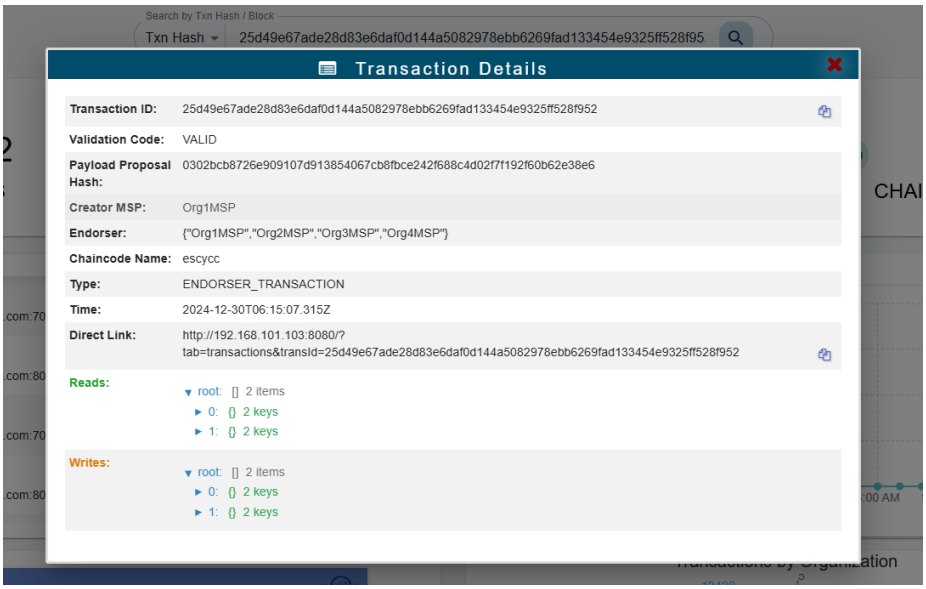
返回区块结果
 
</td></tr>
<tr><td>测试人</td><td></td><td>测试时间</td><td></td></tr>
</table>

1.4.4 区块标识查询服务

用例编号	BC-T104-C4	**用例名称**	区块标识查询服务
用例类型	必选		
测试目标	检测是否支持通过查询工具或查询接口查询区块标识（Hash）/梅克尔树信息		
前置条件	1) 已经进入系统；		
测试过程	**步骤描述（参考）**	**输入数据**	
	通过查询工具查询指定标识信息。	无	
预期结果	能查到正确的标识数据		
测试结果	通过		
备注	系统支持查询指定标识信息 		
测试人		**测试时间**	

1.4.5 事务查询服务

用例编号	BC-T104-C5	**用例名称**	事务查询服务
用例类型	必选		
测试目标	检测是否支持通过查询工具查询指定标识的事务的信息		
前置条件	已经进入系统；		
测试过程	**步骤描述（参考）**	**输入数据**	
	通过查询工具查询指定事务信息。	无	
预期结果	能查到正确的事务数据		

测试结果	通过		
备注	系统支持查询指定事务信息 		
测试人		**测试时间**	

1.4.6　事务操作处理

用例编号	BC-T104-C6	**用例名称**	事务操作处理
用例类型	必选		
测试目标	检验是否可以进行特定事务操作请求提交功能。		
前置条件	1)　已进入系统；		
测试过程	**步骤描述（参考）**		**输入数据**
	1)　进入事务操作界面； 2)　通过服务接口提交一个系统可支持的事务请求，如转账事务。		无
预期结果	通过服务接口提交一个系统可支持的事务请求，并执行成功		
测试结果	通过		

备注	root@a0ad6b3c928d:/opt/gopath/src/github.com/hyperledger/fabric/peer# peer chaincode invoke -o ${ORDERER_ADDRESS} -C ${CHANNEL_NAME} -n ${CC_NAME} -c '{"Args":["publishAsset","waste_oil_id-22122221","User1","Wasteoil","100","desc"]}' --tls --cafile ${ORDERER_CA} --peerAddresses peer0.org1.signatorychain.com:7051 --tlsRootCertFiles ${P0_O1_CRT} --peerAddresses peer0.org2.signatorychain.com:7051 --tlsRootCertFiles ${P0_O2_CRT} --peerAddresses peer0.org3.signatorychain.com:7051 --tlsRootCertFiles ${P0_O3_CRT} --peerAddresses peer0.org4.signatorychain.com:7051 --tlsRootCertFiles ${P0_O4_CRT} 通过命令行数据上链 2024-12-30 06:15:07.356 UTC [chaincodeCmd] chaincodeInvokeOrQuery -> INFO 001 Chaincode invoke successful. result: status:200 payload:"{\"transaction_hash\":\"seconds:1735539307 nanos:315167727 \",\"transaction_id\":\"25d49e67ade28d83e6daf0d144a5082978ebb6269fad133454e9325ff528f952\"}" 数据成功上链		
测试人		**测试时间**	

1.4.7　接口调用频度管理

用例编号	BC-T104-C7	**用例名称**	接口调用频度管理
用例类型	可选		
测试目标	检测是否支持通过设置界面或工具设置接口调用频度		
前置条件	1)　已进入系统；		
测试过程	**步骤描述（参考）**	**输入数据**	
	1)　进入接口服务能力管理界面； 2)　通过设置界面或工具设置接口调用频度；	无	
预期结果	1)　操作不出现报警，错误等异常； 2)　能够正确配置接口调用频率。		
测试结果	通过/此选项不支持		
备注			
测试人		**测试时间**	

1.4.8　接口查询缓存管理

用例编号	BC-T104-C8	**用例名称**	接口查询缓存管理
用例类型	可选		
测试目标	检测是否支持通过设置界面或工具设置账本查询缓存数量		
前置条件	1)　已进入系统；		
测试过程	**步骤描述（参考）**	**输入数据**	
	1)　进入接口服务能力管理界面； 2)　通过设置界面或工具设置账本查询缓存数量。	无	
预期结果	1)　操作不出现报警，错误等异常； 2)　能够正确配置缓存数量。		
测试结果	通过/此选项不支持		
备注			
测试人	张栋、毛超逸	**测试时间**	2022.5.27

1.4.9　较低等级权限接口访问

用例编号	BC-T104-C9	**用例名称**	较低等级权限接口访问
用例类型	可选		
测试目标	检验是否可以使用接口访问权限管理。		
前置条件	1)　已经进入系统；		
测试过程	**步骤描述（参考）**	**输入数据**	

	1) 进入接口访问权限管理界面； 2) 分别设置特定用户的接口为较低等级权限，在每个等级权限时，调用低/高等级权限的几种接口。		无
预期结果	1) 测试过程能够正常完成，不出现报警，错误等异常； 2) 用户在较低等级权限时，只能调用对应等级以及更低等级的接口，不能调用更高等级的接口。		
测试结果	通过/此选项不支持		
备注			
测试人		**测试时间**	

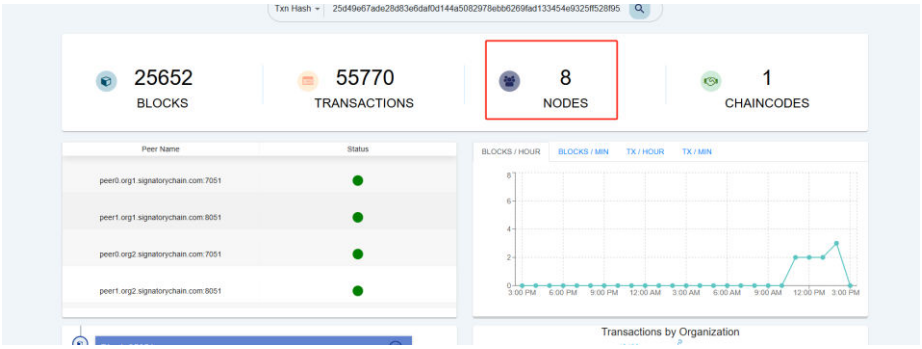
1.4.10 较高等级权限接口访问

用例编号	BC-T104-C10	**用例名称**	较高等级权限接口访问
用例类型	可选		
测试目标	检验是否可以使用接口访问权限管理。		
前置条件	2) 已经进入系统；		
测试过程	**步骤描述（参考）**		**输入数据**
	1) 进入接口访问权限管理界面； 2) 分别设置特定用户的接口为较高等级权限，在每个等级权限时，调用低/高等级权限的几种接口。		无
预期结果	1) 测试过程能够正常完成，不出现报警，错误等异常；		

	2) 用户具有高等级权限时，可以调用低/高等级权限的接口。		
测试结果	通过/此选项不支持		
备注			
测试人		**测试时间**	

1.5 BC-T105：节点管理功能

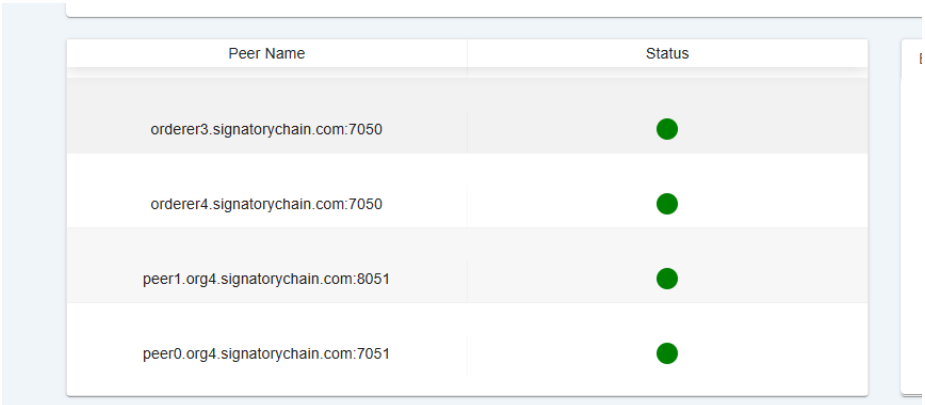
1.5.1 节点服务器信息查询

用例编号	BC-T105-C1	**用例名称**	节点服务器信息查询
用例类型	必选		
测试目标	检测是否支持通过查询工具查询节点的状态		
前置条件	1) 已进入系统		
测试过程	**步骤描述（参考）**	**输入数据**	
	1) 进入节点服务器信息查询界面； 2) 通过查询工具查询节点的状态。	无	
预期结果	1) 测试过程能够正常完成，不出现报警，错误等异常； 2) 正常返回节点状态数据		
测试结果	通过		
备注	系统支持查询节点状态 		

测试人		**测试时间**	

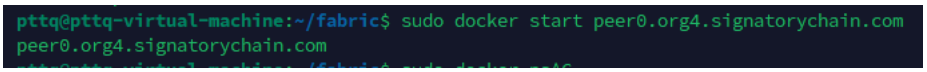
1.5.2 节点启动功能

用例编号	BC-T105-C2	**用例名称**	节点启动功能
用例类型	必选		
测试目标	检测是否支持启动节点进程		
前置条件	1) 已经进入系统		
测试过程	**步骤描述（参考）**	**输入数据**	
	1) 进入节点服务启动控制界面； 2) 启动节点进程；	无	
预期结果	1) 上述修改虚拟机信息的过程不出现报警，错误等异常； 2) 节点进程存在，且工作正常；		
测试结果	通过		

备注	输入节点开启命令行
	`pttq@pttq-virtual-machine:~/fabric$ sudo docker start peer0.org4.signatorychain.com`
	`peer0.org4.signatorychain.com`
	查询节点状态
	 		

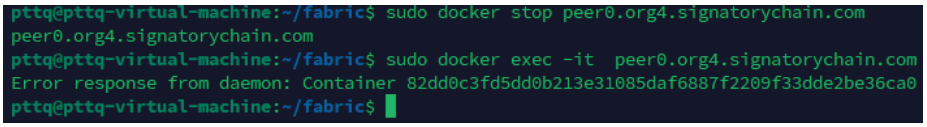
测试人		**测试时间**	

1.5.3 节点服务启动功能

<table>
<tr><td>用例编号</td><td>BC-T105-C3</td><td>用例名称</td><td>节点服务启动功能</td></tr>
<tr><td>用例类型</td><td colspan="3">必选</td></tr>
<tr><td>测试目标</td><td colspan="3">检测是否支持启动共识节点的共识服务</td></tr>
<tr><td>前置条件</td><td colspan="3">1) 已经进入系统</td></tr>
<tr><td rowspan="2">测试过程</td><td colspan="2">步骤描述（参考）</td><td>输入数据</td></tr>
<tr><td colspan="2">1) 进入节点服务启动控制界面；
2) 启动共识节点的共识服务；</td><td>无</td></tr>
<tr><td>预期结果</td><td colspan="3">1) 上述修改虚拟机信息的过程不出现报警，错误等异常；
2) 共识节点正常参与共识；</td></tr>
<tr><td>测试结果</td><td colspan="3">通过</td></tr>
<tr><td>备注</td><td colspan="3">启动共识节点服务
 
查询共识节点事务
  </td></tr>
<tr><td>测试人</td><td></td><td>测试时间</td><td></td></tr>
</table>

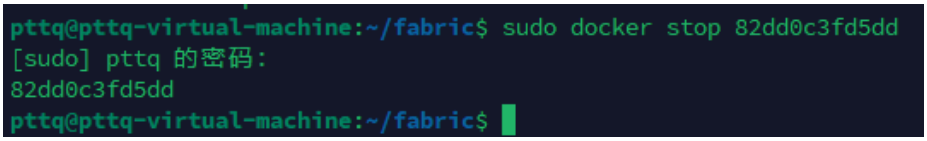
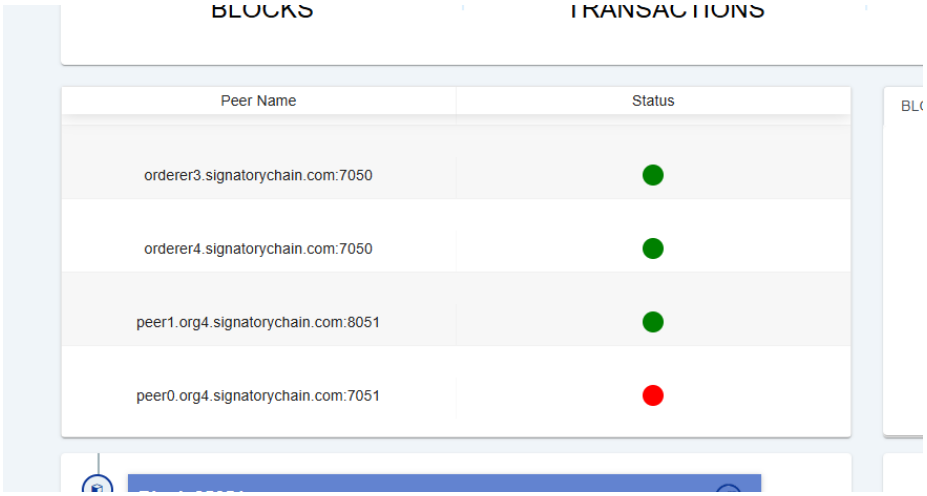
1.5.4 节点服务关闭功能

用例编号	BC-T105-C4	**用例名称**	节点服务关闭功能

用例类型	必选		
测试目标	检测是否支持关闭共识节点的共识服务		
前置条件	1) 已经进入系统		
测试过程	**步骤描述（参考）**	**输入数据**	
	1) 进入节点服务启动控制界面； 2) 关闭共识节点的共识服务；	无	
预期结果	1) 上述修改虚拟机信息的过程不出现报警，错误等异常； 2) 共识节点正常关闭；		
测试结果	通过		
备注	关闭共识节点服务 		
测试人		**测试时间**	

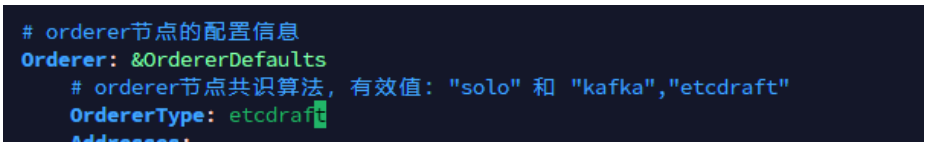
1.5.5 节点关闭功能

用例编号	BC-T105-C5	**用例名称**	节点关闭功能
用例类型	必选		
测试目标	检测是否支持关闭节点进程		
前置条件	1) 已经进入系统		
测试过程	**步骤描述（参考）**	**输入数据**	
	1) 进入节点服务启动控制界面； 2) 关闭节点进程。	无	

预期结果	1) 上述修改虚拟机信息的过程不出现报警，错误等异常； 2) 进程被关闭
测试结果	通过
备注	输入关闭节点命令行  查询节点状态 
测试人		**测试时间**	

1.5.6 节点参与共识算法配置

用例编号	BC-T105-C6	**用例名称**	节点参与共识算法配置
用例类型	必选		
测试目标	检测配置节点（共识节点和接入节点）是否参与共识算法		
前置条件	1) 已经进入系统；		
测试过程	**步骤描述（参考）**	**输入数据**	
	1) 进入节点服务配置界面	无	

	2) 查看配置节点(共识节点和接入节点)是否参与共识算法	
预期结果	1) 测试过程能够正常完成，不出现报警，错误等异常； 2) 共识算法能正确配置	
测试结果	通过	
备注	节点共识算法配置  	
测试人		**测试时间**

1.5.7 节点连接数量配置

用例编号	BC-T105-C7	**用例名称**	节点连接数量配置
用例类型	可选		
测试目标	检测配置节点能接受的连接数量		
前置条件	1) 已经进入系统；		
测试过程	**步骤描述（参考）**		**输入数据**
	1) 进入节点服务配置界面 2) 查看配置节点能接受的连接数量		无
预期结果	1) 测试过程能够正常完成，不出现报警，错误等异常； 2) 节点能接受的连接数量能正确配置		
测试结果	通过/此选项不支持		
备注			

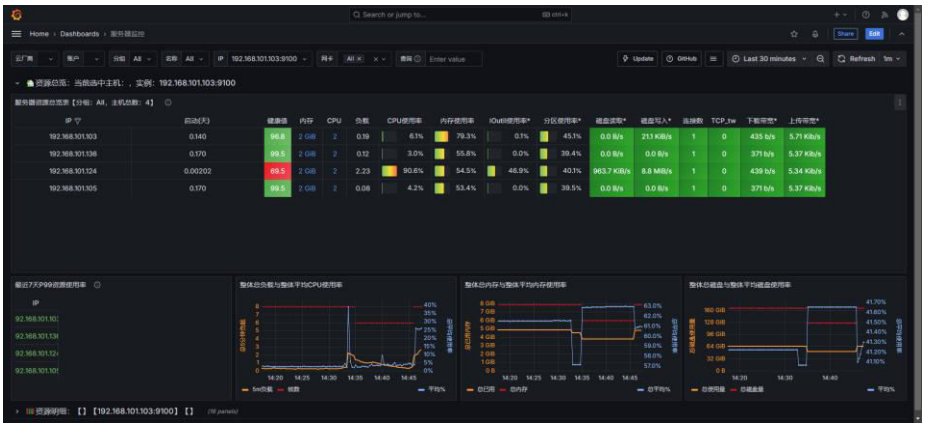
测试人		**测试时间**	

1.5.8　节点对外提供接入服务配置

用例编号	BC-T105-C8	**用例名称**	节点对外提供接入服务配置
用例类型	可选		
测试目标	检测配置节点是否对外提供接入服务		
前置条件	1)　已经进入系统；		
测试过程	**步骤描述（参考）**		**输入数据**
	1)　进入节点服务配置界面 2)　查看配置节点是否对外提供接入服务		无
预期结果	1)　测试过程能够正常完成，不出现报警，错误等异常； 2)　对外提供接入服务能正确配置。		
测试结果	通过/此选项不支持		
备注			
测试人		**测试时间**	

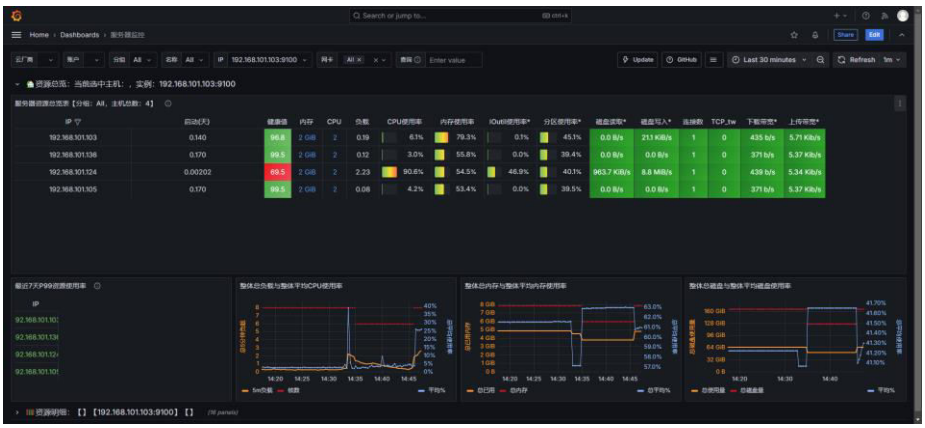
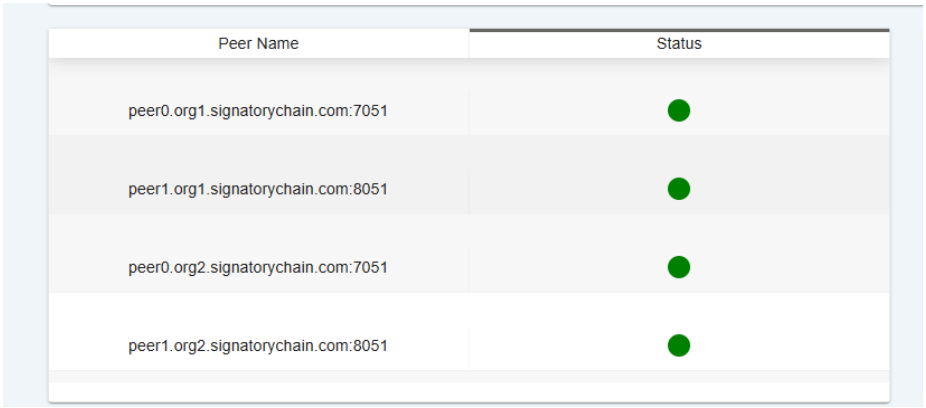
1.5.9　节点连通状况监控服务

用例编号	BC-T105-C9	**用例名称**	节点连通状况监控服务
用例类型	必选		
测试目标	检测是否可以查询节点连通状况监控服务		
前置条件	1)　已经进入系统；		
测试过程	**步骤描述（参考）**		**输入数据**

	1) 进入查询节点状态信息查询 2) 监控节点的连通性，从网络中断开节点，再重新将节点加入网络	无
预期结果	1) 测试过程能够正常完成，不出现报警，错误等异常； 2) 节点连通性在监控平台上的表现符合实际连通情况。	
测试结果	通过	
备注	支持监控界面查询节点连通状态 	
测试人		**测试时间**

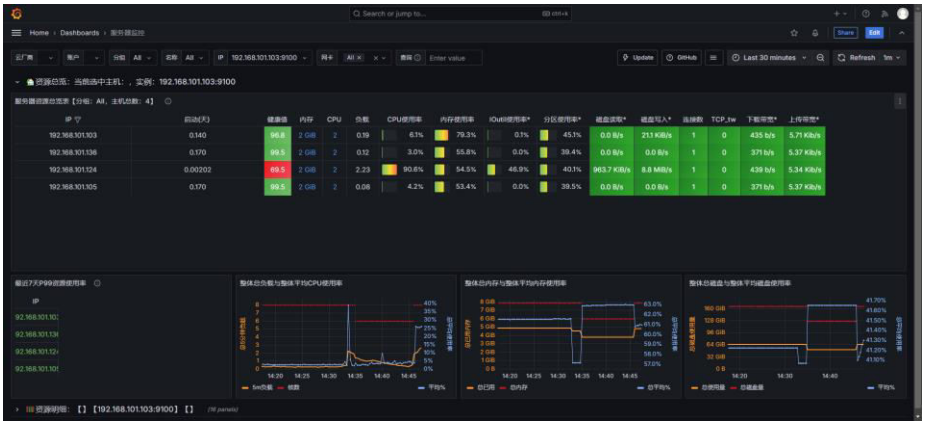
1.5.10 节点连接数量监控服务

用例编号	BC-T105-C10	**用例名称**	节点连接数量监控服务
用例类型	必选		
测试目标	检测是否可以查询节点连接数量监控服务		
前置条件	1) 已经进入系统；		
测试过程	**步骤描述（参考）**		**输入数据**
	1) 进入查询节点状态信息查询		无

	2) 监控节点的连接数量，将不同数量的其他节点连接到被监控节点上	
预期结果	1) 测试过程能够正常完成，不出现报警，错误等异常； 2) 节点连接数在监控平台上的表现符合实际情况。	
测试结果	通过	
备注	支持多监控界面节点状态保持一致   	
测试人		**测试时间**

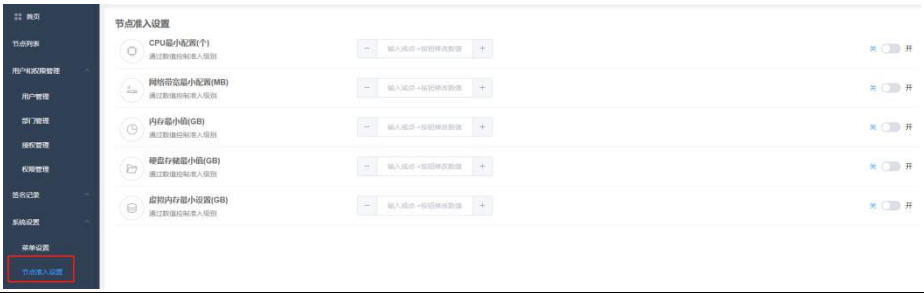
1.5.11 节点带宽监控服务

用例编号	BC-T105-C11	**用例名称**	节点带宽监控服务
用例类型	必选		

测试目标	是否支持检测监控节点带宽		
前置条件	1) 已经进入系统；		
测试过程	**步骤描述（参考）**	**输入数据**	
	1) 进入查询节点状态信息查询 2) 监控节点带宽	无	
预期结果	1) 测试过程能够正常完成，不出现报警，错误等异常； 2) 监控平台上的节点数据流量占用带宽。		
测试结果	通过		
备注	系统支持监控节点带宽 		
测试人		**测试时间**	

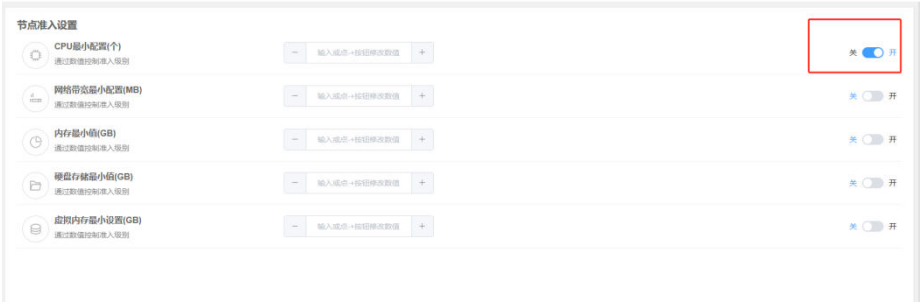
1.5.12 节点准入配置

用例编号	BC-T105-C12	**用例名称**	节点准入配置
用例类型	必选		
测试目标	检测是否可以配置准入配置		
前置条件	1) 已经进入系统		
测试过程	**步骤描述（参考）**	**输入数据**	

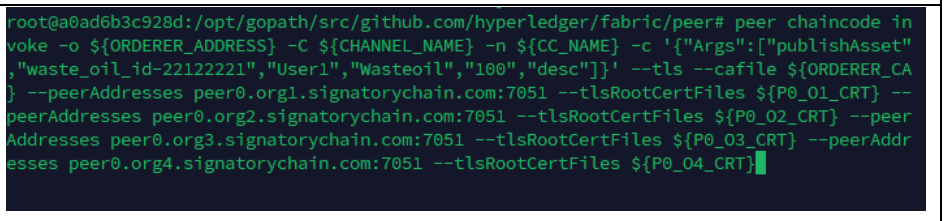
	1) 进入节点授权管理功能界面 2) 在配置平台上配置准入配置	无
预期结果	配置平台上允许配置准入配置，且被测试节点接受连接的情况符合配置结果	
测试结果	通过	
备注	系统支持在平台上配置准入配置 	
测试人		**测试时间**

1.5.13 节点准出配置

用例编号	BC-T105-C13	**用例名称**	节点准出配置
用例类型	必选		
测试目标	检测是否可以配置准出配置		
前置条件	1) 已经进入系统		
测试过程	**步骤描述（参考）**		**输入数据**
	1) 进入节点授权管理功能界面 2) 在配置平台上配置准出配置		无
预期结果	配置平台上允许配置准出配置，且被测试节点连接其他节点的情况符合配置结果		
测试结果	通过		

备注	系统支持节点配置启用和关闭 		
测试人		**测试时间**	

1.5.14 被测试节点事务处理

用例编号	BC-T105-C14	**用例名称**	被测试节点事务处理
用例类型	必选		
测试目标	检测是否支持在被测试节点上提交事务		
前置条件	1) 已经进入系统		
测试过程	**步骤描述（参考）**	**输入数据**	
	1) 进入节点事务处理功能界面 2) 在被测试节点上提交事务	无	
预期结果	事务在链上被接受处理		
测试结果	通过		
备注	 通过命令行 8 个 peer 同时数据上链 		

	数据成功上链		
测试人		**测试时间**	

1.5.15 被测试节点以外节点的事务记录

用例编号	BC-T105-C15	**用例名称**	被测试节点以外节点的事务记录
用例类型	必选		
测试目标	检测是否支持在被测试节点之外节点提交事务		
前置条件	1) 已经进入系统		
测试过程	**步骤描述（参考）**	**输入数据**	
	1) 进入节点事务处理功能界面 2) 在被测试节点之外的其他节点上提交事务	无	
预期结果	事务在被测试节点上被记录		
测试结果	通过		

备注	在节点 2 3 4 提交事务
	`root@a0ad6b3c928d:/opt/gopath/src/github.com/hyperledger/fabric/peer# peer chaincode in`
	`> -c '{"Args":["publishAsset","waste_oil_id22","User2","Wasteoil","50","desc"]}' \`
	`> --tls --cafile ${ORDERER_CA} \`
	`> --peerAddresses peer0.org2.signatorychain.com:7051 --tlsRootCertFiles ${P0_O2_CRT} \`
	`> --peerAddresses peer0.org3.signatorychain.com:7051 --tlsRootCertFiles ${P0_O3_CRT} \`
	`> --peerAddresses peer0.org4.signatorychain.com:7051 --tlsRootCertFiles ${P0_O4_CRT}`
	查询事务提交成功
	`root@a0ad6b3c928d:/opt/gopath/src/github.com/hyperledger/fabric/peer# peer chaincode qu`
	`ery -C escychannel -n escycc -c '{"Args":["GetAsset","waste_oil_id22"]}'`
	`{"description":"desc","id":"waste_oil_id22","owner":"User2","quantity":50,"type":"Waste`
	`oil"}`		

测试人		**测试时间**	

1.5.16 账本允许查询授权配置

用例编号	BC-T105-C16	**用例名称**	账本允许查询授权配置
用例类型	可选		
测试目标	检测是否可以使用账本查询授权配置		
前置条件	1) 已经进入系统		
测试过程	**步骤描述（参考）**		**输入数据**
	1) 进入账本查询授权配置功能界面 2) 配置节点为允许查询账本，并通过本节点使用账本查询		无
预期结果	可以查询到账本信息		
测试结果	通过/此选项不支持		
备注			
测试人		**测试时间**	

1.5.17 账本禁止查询配置

用例编号	BC-T105-C17	**用例名称**	账本禁止查询配置
用例类型	可选		
测试目标	检测是否可以使用账本禁止查询配置		
前置条件	1) 已经进入系统		
测试过程	**步骤描述（参考）**		**输入数据**

	1) 进入账本查询授权配置功能界面 2) 配置为不允许查询账本，通过本节点使用账本查询	无
预期结果	不可以查询到账本信息	
测试结果	通过/此选项不支持	
备注		
测试人		**测试时间**

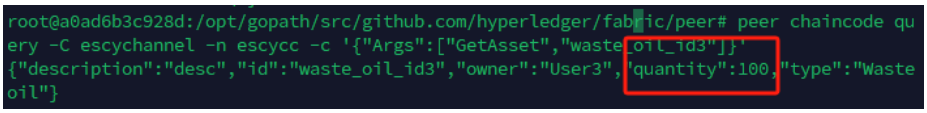
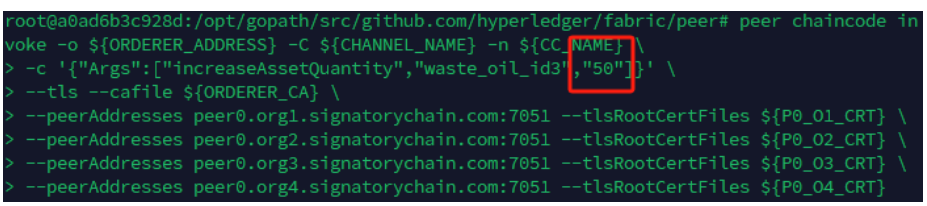
1.6 BC-T106：账本应用

1.6.1 链上内容发布功能

用例编号	BC-T106-C1	**用例名称**	链上内容发布功能
用例类型	必选		
测试目标	检测是否支持发布新的链上内容		
前置条件	1) 已经进入系统		
测试过程	**步骤描述（参考）**		**输入数据**
	1) 进入链上内容发行功能界面 2) 发布新的链上资产类型		无
预期结果	指定链上资产存在		
测试结果	通过		
备注	发布一个新的废油资产 root@a0ad6b3c928d:/opt/gopath/src/github.com/hyperledger/fabric/peer# peer chaincode in > -c '{"Args":["publishAsset","waste_oil_id22","User2","Wasteoil","50","desc"]}' \ > --tls --cafile ${ORDERER_CA} \ > --peerAddresses peer0.org2.signatorychain.com:7051 --tlsRootCertFiles ${P0_O2_CRT} \ > --peerAddresses peer0.org3.signatorychain.com:7051 --tlsRootCertFiles ${P0_O3_CRT} \ > --peerAddresses peer0.org4.signatorychain.com:7051 --tlsRootCertFiles ${P0_O4_CRT}		

	查询费油事务提交成功  		
测试人		**测试时间**	

1.6.2　链上内容增加功能

用例编号	BC-T106-C2	**用例名称**	链上内容增加功能
用例类型	必选		
测试目标	检测是否支持增加已有链上内容数量		
前置条件	1) 已经进入系统		
测试过程	**步骤描述（参考）**		**输入数据**
	1) 进入链上内容发行功能界面 2) 增加已有链上资产的数量		无
预期结果	链上资产数量是否符合增加后预期结果		
测试结果	通过		
备注	查询增加前废油数量 100  增加 50 废油数量  增加成功查询增加后结果为 150		

	root@a0ad6b3c928d:/opt/gopath/src/github.com/hyperledger/fabric/peer# peer chaincode query -C escychannel -n escycc -c '{"Args":["GetAsset","waste_oil_id3"]}' {"description":"desc","id":"waste_oil_id3","owner":"User3","quantity":150,"type":"Wasteoil"}
测试人	**测试时间**

1.6.3 链上内容撤销功能

用例编号	BC-T106-C3	**用例名称**	链上内容撤销功能
用例类型	可选		
测试目标	检测是否支持撤销已有链上内容		
前置条件	1) 已经进入系统		
测试过程	**步骤描述（参考）**	**输入数据**	
	1) 进入链上内容发行功能界面 2) 撤销已有链上资产	无	
预期结果	资产被撤消后应为不存在或状态为“已撤销”		
测试结果	通过/此选项不支持		
备注	1. 提交 waste_oil_idGaaa1 废油资产 root@0404bf2fd7e4:/opt/gopath/src/github.com/hyperledger/fabric/peer# peer chaincode invoke -o ${ORDERER_ADDRESS} -C ${CHANNEL_NAME} -n ${CC_NAME} \ > -c '{"Args":["publishAsset","waste_oil_idGaaa1","G1","Wasteoil","50","desc"]}' \ > --tls --cafile ${ORDERER_CA} \ > --peerAddresses peer0.org1.signatorychain.com:7051 --tlsRootCertFiles ${P0_O1_CRT} \ > --peerAddresses peer0.org2.signatorychain.com:7051 --tlsRootCertFiles ${P0_O2_CRT} \ > --peerAddresses peer0.org3.signatorychain.com:7051 --tlsRootCertFiles ${P0_O3_CRT} 2025-01-02 09:14:49.601 UTC [chaincodeCmd] chaincodeInvokeOrQuery -> INFO 001 Chaincode invoke successful. result: status:200 payload:"{\"transaction_hash\":\"seconds:1735809289 nanos:579541846 \",\"transaction_id\":\"bab540ccf39d7d9c68754789f8c680ebb46d838c26a33933025a45b9ffb5790e\"}" 2. 撤销之前查询 waste_oil_idGaaa1 废油资产 root@0404bf2fd7e4:/opt/gopath/src/github.com/hyperledger/fabric/peer# peer chaincode query -C ${CHANNEL_NAME} -n ${CC_NAME} \ > -c '{"Args":["GetAsset","waste_oil_idGaaa1"]}' {"description":"desc","id":"waste_oil_idGaaa1","owner":"G1","quantity":50,"type":"Wasteoil"} 3. 撤销 waste_oil_idGaaa1 废油资产		

	root@0404bf2fd7e4:/opt/gopath/src/github.com/hyperledger/fabric/peer# peer chaincode invoke -o ${ORDERER_ADDRESS} -C ${CHANNEL_NAME} -n ${CC_NAME} \ > -c '{"Args":["RevokeAsset","waste_oil_idGaaa1"]}' \ > --tls --cafile ${ORDERER_CA} \ > --peerAddresses peer0.org1.signatorychain.com:7051 --tlsRootCertFiles ${P0_O1_CRT} \ > --peerAddresses peer0.org2.signatorychain.com:7051 --tlsRootCertFiles ${P0_O2_CRT} \ > --peerAddresses peer0.org3.signatorychain.com:7051 --tlsRootCertFiles ${P0_O3_CRT} 2025-01-02 09:14:59.028 UTC [chaincodeCmd] chaincodeInvokeOrQuery -> INFO 001 Chaincode invoke successful. result: status:200 payload:"Asset waste_oil_idGaaa1 has been revoked" 4. 查询撤销的 waste_oil_idGaaa1 废油资产 root@0404bf2fd7e4:/opt/gopath/src/github.com/hyperledger/fabric/peer# peer chaincode query -C ${CHANNEL_NAME} -n ${CC_NAME} \ > -c '{"Args":["GetAsset","waste_oil_idGaaa1"]}' Error: endorsement failure during query. response: status:500 message:"Asset with ID waste_oil_idGaaa1 not found"

测试人		**测试时间**	

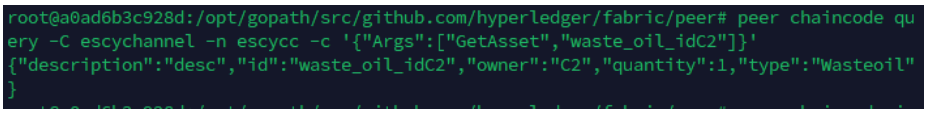
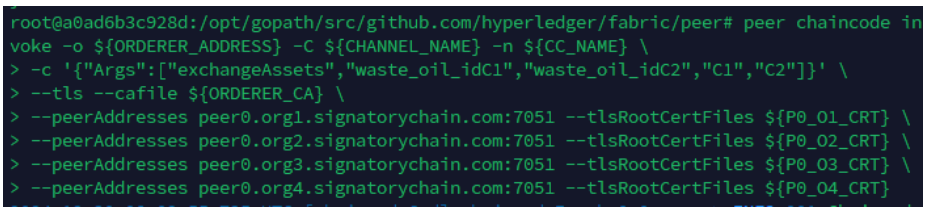
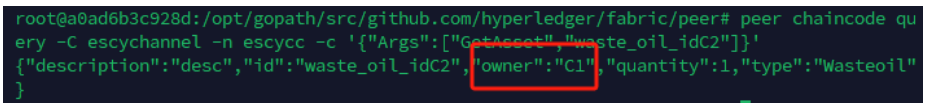
1.6.4 链上内容分配功能

用例编号	BC-T106-C4	**用例名称**	链上内容分配功能
用例类型	必选		
测试目标	检测是否支持给指定用户分配一定的数量的链上内容		
前置条件	1) 已经进入系统		
测试过程	**步骤描述（参考）**	**输入数据**	
	1) 进入链上内容发行功能界面 2) 给指定用户分配一定的数量的链上资产	无	
预期结果	用户名下的资产的数量符合分配的结果预期		
测试结果	通过		
备注	查询废油 waste_oil_idA1 root@a0ad6b3c928d:/opt/gopath/src/github.com/hyperledger/fabric/peer# peer chaincode query -C escychannel -n escycc -c '{"Args":["GetAsset","waste_oil_idA1"]}' {"description":"desc","id":"waste_oil_idA1","owner":"A1","quantity":0,"type":"Wasteoil"} 查询废油 waste_oil_idA2		

	root@a0ad6b3c928d:/opt/gopath/src/github.com/hyperledger/fabric/peer# peer chaincode query -C escychannel -n escycc -c '{"Args":["GetAsset","waste_oil_idA2"]}' {"description":"desc","id":"waste_oil_idA2","owner":"A2","quantity":10,"type":"Wasteoil"} 分配 waste_oil_idA1 给 waste_oil_idA2 root@a0ad6b3c928d:/opt/gopath/src/github.com/hyperledger/fabric/peer# peer chaincode invoke -o ${ORDERER_ADDRESS} -C ${CHANNEL_NAME} -n ${CC_NAME} \ > -c '{"Args":["allocateAsset","waste_oil_idA1","A1","A2"]}' \ > --tls --cafile ${ORDERER_CA} \ > --peerAddresses peer0.org1.signatorychain.com:7051 --tlsRootCertFiles ${P0_O1_CRT} \ > --peerAddresses peer0.org2.signatorychain.com:7051 --tlsRootCertFiles ${P0_O2_CRT} \ > --peerAddresses peer0.org3.signatorychain.com:7051 --tlsRootCertFiles ${P0_O3_CRT} \ > --peerAddresses peer0.org4.signatorychain.com:7051 --tlsRootCertFiles ${P0_O4_CRT} 废油 waste_oil_idA1 数据分配给了 A2 root@a0ad6b3c928d:/opt/gopath/src/github.com/hyperledger/fabric/peer# peer chaincode query -C escychannel -n escycc -c '{"Args":["GetAsset","waste_oil_idA1"]}' {"description":"desc","id":"waste_oil_idA1","owner":"A2","quantity":0,"type":"Wasteoil"}
测试人	**测试时间**

1.6.5 链上内容交换功能

用例编号	BC-T106-C5	**用例名称**	链上内容交换功能
用例类型	必选		
测试目标	检测是否支持在不同用户之间交换链上内容		
前置条件	1) 已经进入系统		
测试过程	**步骤描述（参考）**	**输入数据**	
	1) 进入链上内容发行功能界面 2) 在不同用户之间交换链上资产	无	
预期结果	用户名下的资产的数量符合交换的结果预期		
测试结果	通过		
备注	查询 waste_oil_idC1 废油资产 root@a0ad6b3c928d:/opt/gopath/src/github.com/hyperledger/fabric/peer# peer chaincode query -C escychannel -n escycc -c '{"Args":["GetAsset","waste_oil_idC1"]}' {"description":"desc","id":"waste_oil_idC1","owner":"C1","quantity":30,"type":"Wasteoil"}		

	查询 waste_oil_idC2 废油资产  waste_oil_idC1 废油资产和 waste_oil_idC2 资产交换  查询交换后 waste_oil_idC1 的废油资产  查询交换后 waste_oil_idC2 的废油资产 

测试人		**测试时间**	

1.6.6　共识前特定标识的资产的逻辑验证

用例编号	BC-T106-C6	**用例名称**	共识前特定标识的资产的逻辑验证
用例类型	必选		
测试目标	检测是否可以使用共识前特定标识的资产的逻辑验证		
前置条件	1) 已经进入系统		
测试过程	**步骤描述（参考）**		**输入数据**
	1) 进入链上内容发行功能界面 2) 发起具有特定标识的资产在不同用户		无

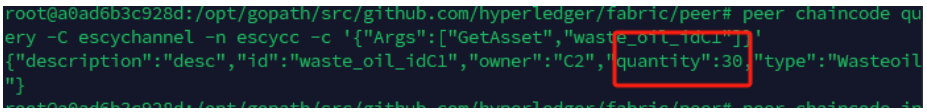
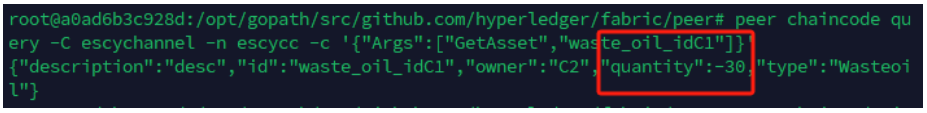
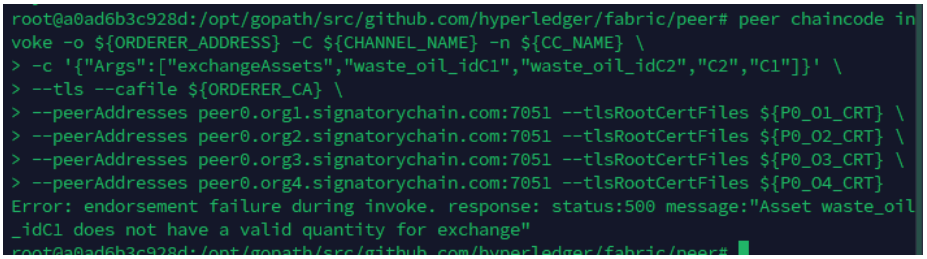
	之间的转移的事务，并经过共识确认	
预期结果	资产归属关系符合预期	
测试结果	通过	

备注	发起特定标识的废油资产在不同用户之间转移
	`root@a0ad6b3c928d:/opt/gopath/src/github.com/hyperledger/fabric/peer# peer chaincode invoke -o ${ORDERER_ADDRESS} -C ${CHANNEL_NAME} -n ${CC_NAME} \`
	`> -c '{"Args":["exchangeAssets","waste_oil_idC1","waste_oil_idC2","C1","C2"]}' \`
	`> --tls --cafile ${ORDERER_CA} \`
	`> --peerAddresses peer0.org1.signatorychain.com:7051 --tlsRootCertFiles ${P0_O1_CRT} \`
	`> --peerAddresses peer0.org2.signatorychain.com:7051 --tlsRootCertFiles ${P0_O2_CRT} \`
	`> --peerAddresses peer0.org3.signatorychain.com:7051 --tlsRootCertFiles ${P0_O3_CRT} \`
	`> --peerAddresses peer0.org4.signatorychain.com:7051 --tlsRootCertFiles ${P0_O4_CRT}`
	交易失败 提示该废油资产不属于该用户
	`Error: endorsement failure during invoke. response: status:500 message:"Asset waste_oil_idC1 is not owned by user C1"`	

测试人		**测试时间**

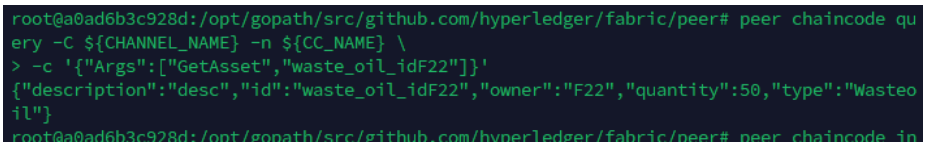
1.6.7 共识前资产数额逻辑验证

用例编号	BC-T106-C7	**用例名称**	共识前资产数额逻辑验证
用例类型	必选		
测试目标	检测是否可以使用共识前资产数额逻辑验证		
前置条件	1) 已经进入系统		
测试过程	**步骤描述（参考）**	**输入数据**	
	1) 进入链上内容发行功能界面 2) 发起一定数额的资产在不同用户之间的转移的事务，但未经过共识确认	无	
预期结果	共识前各自名下的资产数量符合预期		
测试结果	通过		

备注	查询当前废油资产为 30  模拟别的交易把当前废油数量支出 60  再次查询废油资产为-30  再次发起交换 提示 waste_oil_idC1 余额不足 
测试人		**测试时间**	

1.6.8 共识后的结果验算

用例编号	BC-T106-C8	**用例名称**	共识后的结果验算
用例类型	必选		
测试目标	检测是否可以使用共识后的结果验算		
前置条件	1) 已经进入系统		
测试过程	**步骤描述（参考）**		**输入数据**

	1) 进入链上内容发行功能界面 2) 发起一定数额的资产在不同用户之间的转移的事务，并经过共识确认	无
预期结果	共识后各自名下的资产数量符合预期	
测试结果	通过	
备注	转移前查询 waste_oil_idF11 废油资产为 50  转移前查询 waste_oil_idF22 废油资产为 50  waste_oil_idF11 废油资产向 waste_oil_idF22 废油资产转移 50  转移后查询 waste_oil_idF11 废油资产为 0  转移后查询 waste_oil_idF22 废油资产为 100 	

测试人		**测试时间**	

1.6.9　可对多签名权限控制设置

用例编号	BC-T106-C9	**用例名称**	可对多签名权限控制设置
用例类型	可选		
测试目标	设置事务是否需要多签名和所需签名列表（配置）		
前置条件	1) 已经进入系统		
测试过程	**步骤描述（参考）**	**输入数据**	
	1) 进入签名权限控制设置界面 2) 设置事务是否需要多签名和所需签名列表	无	
预期结果	正确的设置需要多签名和所需签名列表		
测试结果	通过/此选项不支持		
备注			
测试人		**测试时间**	

1.6.10 可对特定事务处理进行多签名

用例编号	BC-T106-C10	**用例名称**	可对特定事务进行多签名处理
用例类型	可选		
测试目标	检测是否可以对特定事务进行多签名处理		
前置条件	1) 已经进入系统		
测试过程	**步骤描述（参考）**	**输入数据**	

	1) 进入签名权限控制设置界面 2) 发起一个需要多签名的特定事务，为事务赋值足够的数量的签名	无
预期结果	获得足够签名的事务能执行成功	
测试结果	通过/此选项不支持	
备注		
测试人		**测试时间**

1.6.11 可对多签名事务处理进行验证

用例编号	BC-T106-C11	**用例名称**	可对多签名事务处理进行验证
用例类型	可选		
测试目标	系统需要经过多签名验证才能顺利执行事务请求		
前置条件	1) 已经进入系统		
测试过程	**步骤描述（参考）**	**输入数据**	
	1) 进入签名权限控制设置界面 2) 发起一个需要多签名的特定事务，为事务赋值数量不足的签名	无	
预期结果	事务未获得足够签名不能执行成功		
测试结果	通过/此选项不支持		
备注			
测试人		**测试时间**	

1.6.12 执行合约逻辑

用例编号	BC-T106-C12	**用例名称**	执行合约逻辑
用例类型	可选		
测试目标	检测是否可基于智能合约功能组件执行合约逻辑		
前置条件	1) 已经进入系统		
测试过程	**步骤描述（参考）**	**输入数据**	
	1) 进入智能合约功能界面 2) 查看文档定义该系统是/否支持指定智能合约，并发起一个事务，调用特定合约接口。	无	
预期结果	如文档定义该系统支持智能合约，且特定合约接口存在，则事务调用应成功。 如文档定义该系统不支持智能合约，则事务无法调用任何智能合约的接口		
测试结果	通过/此选项不支持		
备注			
测试人		**测试时间**	

1.7 BC-T107：共识机制

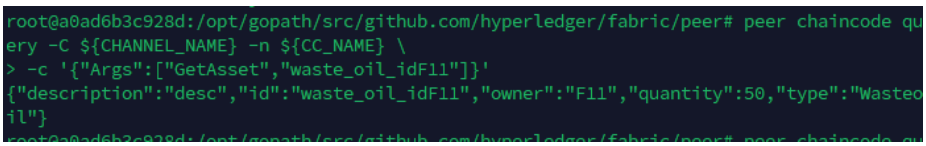
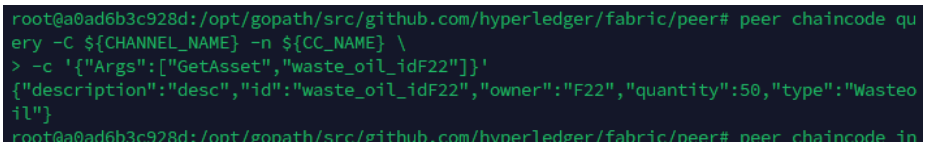
1.7.1 多节点共识确认

用例编号	BC-T107-C1	**用例名称**	多节点共识确认
用例类型	必选		
测试目标	检测是否支持多个节点参与共识和确认		
前置条件	1) 已经进入系统		

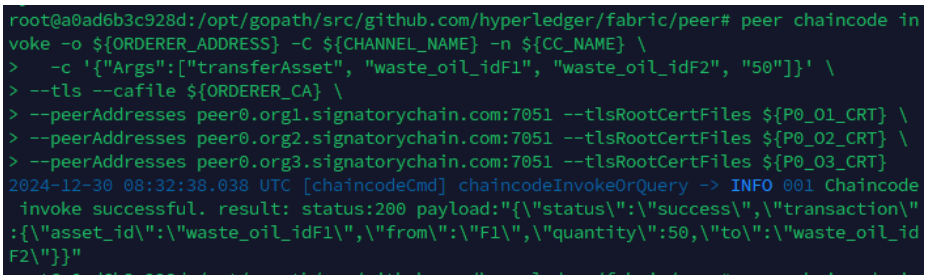
测试过程	**步骤描述（参考）**	**输入数据**
	1) 进入智能合约功能界面 2) 发起事务，根据文档共识方法要求，查看事务是否被记录到区块，同时此区块是否包含相应节点的签名	无
预期结果	满足记录到区块和包含相应节点签名	
测试结果	通过	
备注	通过区块查询交易信息，查询签名信息返回结果 	
测试人		**测试时间**

1.7.2 独立节点提交正确事务逻辑验证

用例编号	BC-T107-C2	**用例名称**	独立节点提交正确事务逻辑验证
用例类型	必选		
测试目标	提交正确事务到区块链系统，检测是否能通过		
前置条件	1) 已经进入系统		
测试过程	**步骤描述（参考）**		**输入数据**
	1) 进入事务功能界面 2) 提交正确事务到区块链系统		无

预期结果	正确事务被记录到区块		
测试结果	通过		
备注	转移前查询 waste_oil_idF11 废油资产为 50  转移前查询 waste_oil_idF22 废油资产为 50  waste_oil_idF11 废油资产向 waste_oil_idF22 废油资产转移 50  转移后查询 waste_oil_idF11 废油资产为 0  转移后查询 waste_oil_idF22 废油资产为 100 		
测试人		**测试时间**	

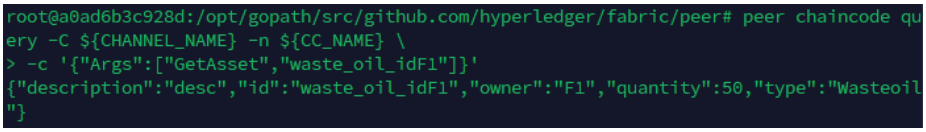
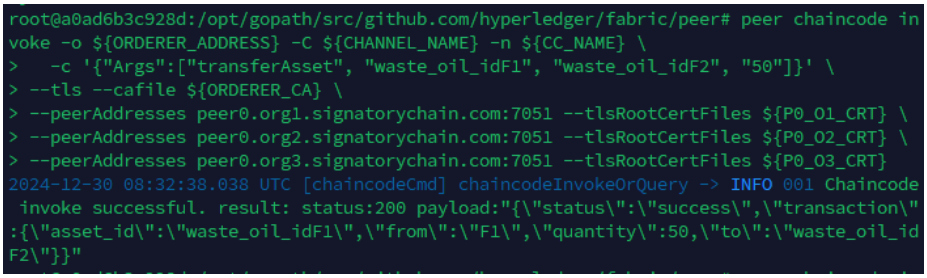
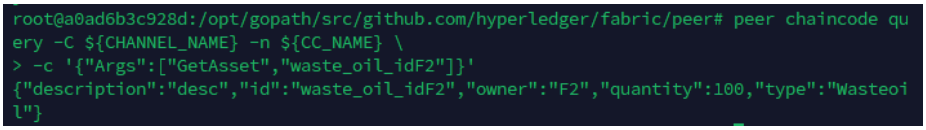
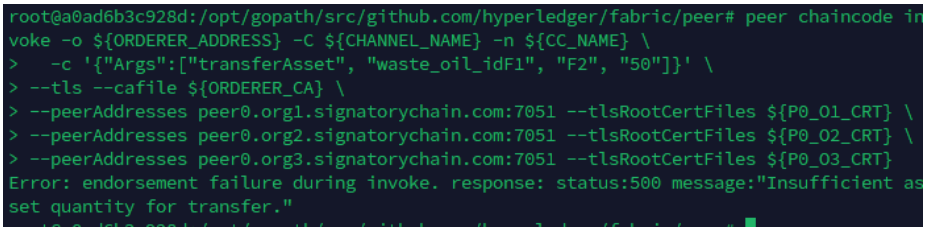
1.7.3 独立节点提交错误事务逻辑验证

<table>
<tr><td>用例编号</td><td>BC-T107-C3</td><td>用例名称</td><td>独立节点提交错误事务逻辑验证</td></tr>
<tr><td>用例类型</td><td colspan="3">必选</td></tr>
<tr><td>测试目标</td><td colspan="3">提交错误事务到区块链系统，检测是否能通过</td></tr>
<tr><td>前置条件</td><td colspan="3">1) 已经进入系统</td></tr>
<tr><td rowspan="2">测试过程</td><td colspan="2">步骤描述（参考）</td><td>输入数据</td></tr>
<tr><td colspan="2">1) 进入事务功能界面
2) 提交错误事务到区块链系统</td><td>无</td></tr>
<tr><td>预期结果</td><td colspan="3">错误事务有错误提示</td></tr>
<tr><td>测试结果</td><td colspan="3">通过</td></tr>
<tr><td>备注</td><td colspan="3">交易前查询废油资产 waste_oil_idF1
 
交易前查询废油资产 waste_oil_idF2
 
发起 waste_oil_idF1 废油资产给 waste_oil_idF2 废油资产转移 50 的交易，
提示交易成功
 
交易后查询废油资产 waste_oil_idF1</td></tr>
</table>

	root@a0ad6b3c928d:/opt/gopath/src/github.com/hyperledger/fabric/peer# peer chaincode query -C ${CHANNEL_NAME} -n ${CC_NAME} \ > -c '{"Args":["GetAsset","waste_oil_idF1"]}' {"description":"desc","id":"waste_oil_idF1","owner":"F1","quantity":0,"type":"Wasteoil"} 交易后查询废油资产 waste_oil_idF2 root@a0ad6b3c928d:/opt/gopath/src/github.com/hyperledger/fabric/peer# peer chaincode query -C ${CHANNEL_NAME} -n ${CC_NAME} \ > -c '{"Args":["GetAsset","waste_oil_idF2"]}' {"description":"desc","id":"waste_oil_idF2","owner":"F2","quantity":100,"type":"Wasteoil"} 再次发起发起 waste_oil_idF1 废油资产给 waste_oil_idF2 废油资产转移 50 的交易 提示交易失败 废油资产余额不足 root@a0ad6b3c928d:/opt/gopath/src/github.com/hyperledger/fabric/peer# peer chaincode invoke -o ${ORDERER_ADDRESS} -C ${CHANNEL_NAME} -n ${CC_NAME} \ > -c '{"Args":["transferAsset", "waste_oil_idF1", "F2", "50"]}' \ > --tls --cafile ${ORDERER_CA} \ > --peerAddresses peer0.org1.signatorychain.com:7051 --tlsRootCertFiles ${P0_O1_CRT} \ > --peerAddresses peer0.org2.signatorychain.com:7051 --tlsRootCertFiles ${P0_O2_CRT} \ > --peerAddresses peer0.org3.signatorychain.com:7051 --tlsRootCertFiles ${P0_O3_CRT} Error: endorsement failure during invoke. response: status:500 message:"Insufficient asset quantity for transfer."
测试人	**测试时间**

1.7.4 独立节点记录信息需通过共识

用例编号	BC-T107-C4	**用例名称**	独立节点记录信息需通过共识
用例类型	必选		
测试目标	防止独立节点未经共识进行信息记录或修改		
前置条件	1) 已经进入系统		
测试过程	**步骤描述（参考）**	**输入数据**	
	1) 进入正确事务功能界面 2) 提交“双花”事物	无	
预期结果	只有一笔事物被记录到区块		
测试结果	通过		

备注	交易前查询废油资产 waste_oil_idF1  交易前查询废油资产 waste_oil_idF2  发起 waste_oil_idF1 废油资产给 waste_oil_idF2 废油资产转移 50 的交易， 提示交易成功  交易后查询废油资产 waste_oil_idF1  交易后查询废油资产 waste_oil_idF2  再次发起发起 waste_oil_idF1 废油资产给 waste_oil_idF2 废油资产转移 50 的交易 提示交易失败 废油资产余额不足 
测试人		**测试时间**	

1.7.5 共识机制容错性

<table>
<tr><td>用例编号</td><td>BC-T107-C5</td><td>用例名称</td><td>物理故障导致的非恶意错误容错性</td></tr>
<tr><td>用例类型</td><td colspan="3">必选</td></tr>
<tr><td>测试目标</td><td colspan="3">检测系统是否具有一定的容错性，处理如物理或网络故障的非恶意错误</td></tr>
<tr><td>前置条件</td><td colspan="3">1) 已经进入系统</td></tr>
<tr><td rowspan="2">测试过程</td><td colspan="2">步骤描述（参考）</td><td>输入数据</td></tr>
<tr><td colspan="2">1) 进入正确事务功能界面
2) 少量节点断开网络，查看网络状态是否正常</td><td>无</td></tr>
<tr><td>预期结果</td><td colspan="3">产生区块</td></tr>
<tr><td>测试结果</td><td colspan="3">通过</td></tr>
<tr><td>备注</td><td colspan="3">网络共识机制 etcdraft
当前节点数量为 4 个
停止第 4 个节点命令行
 
依然可以提交事务上链</td></tr>
</table>

<table>
<tr><td></td><td>
<pre>root@a0ad6b3c928d:/opt/gopath/src/github.com/hyperledger/fabric/peer# peer chaincode in
> -c '{"Args":["publishAsset","waste_oil_idG1","G1","Wasteoil","50","desc"]}' \
> --tls --cafile ${ORDERER_CA} \
> --peerAddresses peer0.org1.signatorychain.com:7051 --tlsRootCertFiles ${P0_O1_CRT} \
> --peerAddresses peer0.org2.signatorychain.com:7051 --tlsRootCertFiles ${P0_O2_CRT} \
> --peerAddresses peer0.org3.signatorychain.com:7051 --tlsRootCertFiles ${P0_O3_CRT}
2024-12-30 08:39:53.103 UTC [chaincodeCmd] chaincodeInvokeOrQuery -> INFO 001 Chaincode
87224f704c842d95b6fa3b73c56f7937eb06453686e4\"}"</pre>
验证事务成功提交到区块
<pre>root@a0ad6b3c928d:/opt/gopath/src/github.com/hyperledger/fabric/peer# peer chaincode qu
ery -C ${CHANNEL_NAME} -n ${CC_NAME} \
> -c '{"Args":["GetAsset","waste_oil_idG1"]}'
{"description":"desc","id":"waste_oil_idG1","owner":"G1","quantity":50,"type":"Wasteoil
"}</pre>
停止第 3 个节点命令行
<pre>pttq@pttq-virtual-machine:~/fabric$ sudo sh stop.sh
[sudo] pttq 的密码:
[+] Running 7/7
 ✓ Container orderer3.signatorychain.com   Removed
 ✓ Container peer0.org3.signatorychain.com Removed
 ✓ Container peer1.org3.signatorychain.com Removed
 ✓ Container ca3.signatorychain.com   Removed
 ✓ Container couchdb0   Removed
 ✓ Container couchdb1   Removed
 ✓ Network fabric_formal   Removed</pre>
再次提交事务上链提示失败
<pre>root@a0ad6b3c928d:/opt/gopath/src/github.com/hyperledger/fabric/peer# peer chaincode in
> -c '{"Args":["publishAsset","waste_oil_idG2","G2","Wasteoil","50","desc"]}' \
> --tls --cafile ${ORDERER_CA} \
> --peerAddresses peer0.org1.signatorychain.com:7051 --tlsRootCertFiles ${P0_O1_CRT} \
> --peerAddresses peer0.org2.signatorychain.com:7051 --tlsRootCertFiles ${P0_O2_CRT}
Error: error sending transaction for invoke: got unexpected status: SERVICE_UNAVAILABLE
ction_id\":\"b9c67d4fb49bae61b03a70120ed522b59c5136cb858f404b950f095d716e962e\"}" > pay
2m\032k\n\016waste_oil_idG2\032Y{\"id\":\"waste_oil_idG2\",\"owner\":\"G2\",\"type\":\"
"transaction_hash\":\"seconds:1735548138 nanos:344054944 \",\"transaction_id\":\"b9c67d
 CERTIFICATE-----\nMIICPTCCAeSgAwIBAgIQbLdLjsem2UmJiehywN5msjAKBggqhkjOPQQDAjCBgTEL\nMA
Z25hdG9yeWNoYWluLmNvbTAeFw0yNDEyMTMwMTE1MDBa\nFw0zNDEyMTEwMTE1MDBaMHExCzAJBgNVBAYTAlVTM
9AgEG\nCCqGSM49AwEHA0IABEcdsubpeVeCebWeDxhdfV/NbTue8FoumRkaCu56Ar7Lrb9F\nTaGx4ScC0ZSHeI
A0cAMEQCIDKqZyD5Ui3jgLnSDIo5s6FZ\nhVGnc2cN0WjlBNr5gi3kAiBvjEZkgK7FnfNCYNy4H5//RS+eh5FmJ
h]\002 \014\204fw\031;x=\342\t\340Q\314\325\037gA\2154\361Wz\022\350\247.\230\213\232w`</pre>
允许一个共识节点断连
</td></tr>
<tr><td>测试人</td><td></td><td>测试时间</td><td></td></tr>
</table>

1.8 BC-T108：账本记录

1.8.1 持久化存储账本记录

用例编号	BC-T108-C1	**用例名称**	持久化存储账本记录

<table>
<tr><td>用例类型</td><td colspan="2">必选</td></tr>
<tr><td>测试目标</td><td colspan="2">检测是否支持持久化存储账本记录</td></tr>
<tr><td>前置条件</td><td colspan="2">1) 已经进入系统</td></tr>
<tr><td rowspan="2">测试过程</td><td>步骤描述（参考）</td><td>输入数据</td></tr>
<tr><td>1) 查询技术文档看是否符合要求，包括数据库种类、数据库指标（安全性、兼容性、可扩展性）账本存储格式和区块格式规范
2) 在一个节点构造事务请求，在其他任意节点查询区块链数据看是否符合要求。通过查询数据库相应记录，查看是否符合要求</td><td>无</td></tr>
<tr><td>预期结果</td><td colspan="2">1) 账本上存在记录
2) 数据库存在记录</td></tr>
<tr><td>测试结果</td><td colspan="2">通过</td></tr>
<tr><td>备注</td><td colspan="2"> 
</td></tr>
</table>

4. 扩容云硬盘分区和文件系统

扩容分区的时候可以选择在已有的分区上继续扩大，也可以选择新增一个分区，请根据实际情况选择。

示例说明:数据盘"/dev/vdb"原有容量 100GiB,扩大至 150GiB,将新增的 50GB 增加至已有分区"/dev/vdb1.

1. 检查当前系统是否已安装 growpart 扩容工具。
2. 如果回显为工具使用介绍，则表示已安装，无需重复安装.
3. 如果没有以上回显信息，请执行以下安装命令:**yum install cloud-utils-growpart**

```
Loaded plugins: fastestmirror

Determining fastest mirrors

epel/x86_64/metalink

| 8.0 kB  00:00:00

...

Package cloud-utils-growpart-0.29-2.el7.noarch already installed and

latest version

Nothing to do
```

4.
5. 表示安装成功
6. 查看数据盘"/dev/vdb"的分区信息:**lsblk**

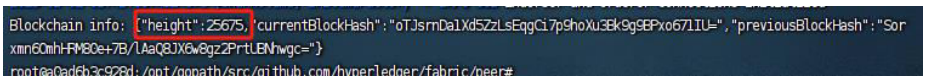
```
[root@ecs-centos76 ~]# lsblk
NAME   MAJ:MIN RM SIZE RO TYPE MOUNTPOINT
vda    253:0    0  40G  0 disk
├vda1 253:1    0  40G  0 part /
vdb    253:16   0  150G 0 disk
├vdb1 253:17   0  100G 0 part /mnt/sdc
```

7.
8. 表示数据盘"/dev/vdb"有 150GiB 的容量，分区"/dev/vdb1"已有 100GiB，扩容的 50GiB 容量还未划分磁盘分区
9. 如果磁盘未分区，需要直接扩容文件系统，请执行 6
10. 将扩容新增的容量增加至分区"/dev/vdb1"。
11. **growpart /dev/vdb 1**
12. 扩容分区"/dev/vdb1"文件系统大小

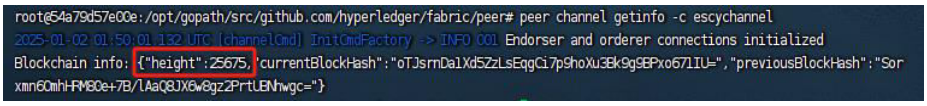
13. **查看数据盘"/dev/vdb"对应分区"/dev/vdb1"的文件系统类型。**
14. **parted /dev/vdb**
15. **P**
16. 查看完成后，输入"q"，按"Enter"，退出 parted 模式
17. 根据回显可知分区"/dev/vdb1"的文件系统类型为 ext4，执行以下命令进行扩容
18. **resize2fs /dev/vdb1**
19. 如果出现报错"open: No such file or directory while opening /dev/vdb1"，则表示您指定的磁盘分区不正确，请使用 parted 命令查看磁盘分区。
20. 如果文件系统为 xfs，请执行以下命令（/mnt/sdc 为/dev/vdb1 的挂载点，您需要根据实际情况修改）。
21. **sudo xfs_growfs /mnt/sdc**
22. 查看扩容后分区的容量
23. **lsblk**
24. 表示数据盘"/dev/vdb"的容量大小为 150GiB，该数据盘下的分区"/dev/vdb1"大小也为 150GiB，说明扩容的 50GB 容量已生效

测试人		测试时间	

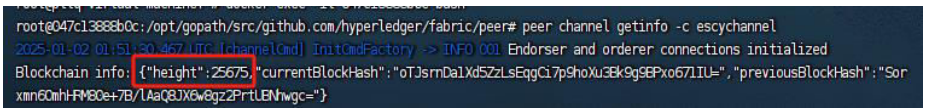
1.8.2 支持多节点拥有完整区块记录

<table>
<tr><td>用例编号</td><td>BC-T108-C2</td><td>用例名称</td><td>支持多节点拥有完整区块记录</td></tr>
<tr><td>用例类型</td><td colspan="3">必选</td></tr>
<tr><td>测试目标</td><td colspan="3">检测是否支持多个节点查询同一区块记录，多个节点显示的区块记录一致</td></tr>
<tr><td>前置条件</td><td colspan="3">1) 已经进入系统</td></tr>
<tr><td rowspan="2">测试过程</td><td colspan="2">步骤描述（参考）</td><td>输入数据</td></tr>
<tr><td colspan="2">1) 查询技术文档看是否符合要求，包括数据库种类、数据库指标（安全性、兼容性、可扩展性）账本存储格式和区块格式规范
2)在一个节点构造事务请求，在其他 N (N 大于总节点数量一半）个任意节点查询区块链数据都能查看一致的区块数据</td><td>无</td></tr>
<tr><td>预期结果</td><td colspan="3">账本上存在记录</td></tr>
<tr><td>测试结果</td><td colspan="3">通过</td></tr>
<tr><td>备注</td><td colspan="3">利用 peer0.org1 查询节点高度为 25675
 

利用 peer0.org2 查询节点高度为 25675
 

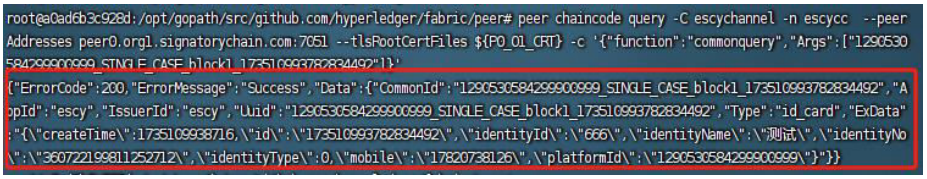
利用 peer0.org3 查询节点高度为 25675
 

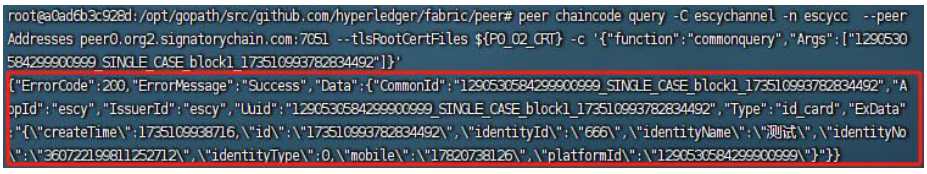
利用 peer0.org4 查询节点高度为 25675</td></tr>
</table>

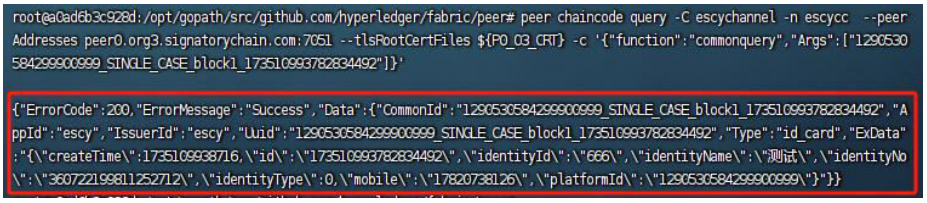
	 通过共识节点查询最新区块高度预期一致，证明系统支持多节点拥有区块完整高度记录
测试人	**测试时间**

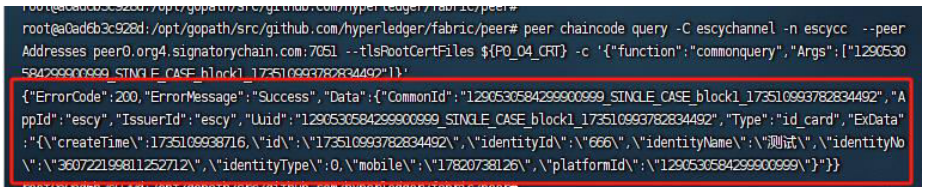
1.8.3 支持多节点拥有完整的数据记录

用例编号	BC-T108-C3	**用例名称**	支持多节点拥有完整的数据记录
用例类型	必选		
测试目标	检测是否支持多个节点查询完整的账本记录		
前置条件	1) 已经进入系统		
测试过程	**步骤描述（参考）**	**输入数据**	
	1) 查询技术文档看是否符合要求，包括数据库种类、数据库指标（安全性、兼容性、可扩展性）账本存储格式和区块格式规范 2) 在一个节点构造事务请求，交易双方和任意节点都能查询链上账本记录	无	
预期结果	数据库存在记录		
测试结果	通过		
备注	利用 peer0.org1 查询交易 1290530584299900999_SINGLE_CASE_block1_1735109937828344 92 的数据		

<table>
<tr>
<td></td>
<td>


利用peer0.org2查询交易1290530584299900999_SINGLE_CASE_block1_173510993782834492的数据


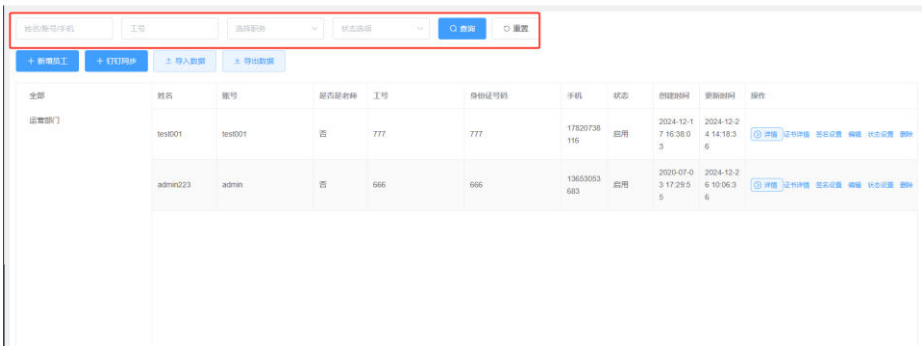
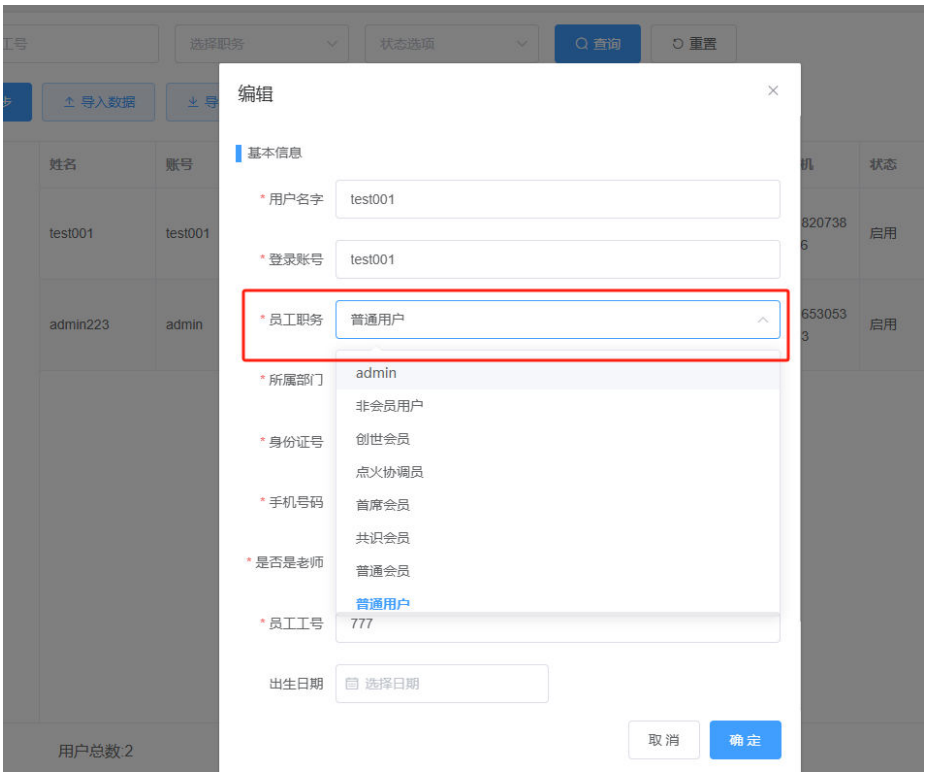
利用peer0.org3查询交易1290530584299900999_SINGLE_CASE_block1_173510993782834492的数据


利用peer0.org4查询交易1290530584299900999_SINGLE_CASE_block1_173510993782834492的数据


通过共识节点查询交易1290530584299900999_SINGLE_CASE_block1_173510993782834492的数据预期一致，证明系统支持任意节点都能查询区块记录
</td>
</tr>
<tr>
<td>测试人</td>
<td>张栋、毛超逸</td>
<td>测试时间</td>
<td>2022.5.26</td>
</tr>
</table>

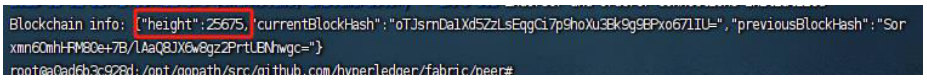
1.8.4 自定义账本权限

<table>
<tr><td>用例编号</td><td>BC-T108-C4</td><td>用例名称</td><td>自定义账本权限</td></tr>
<tr><td>用例类型</td><td colspan="3">必选</td></tr>
<tr><td>测试目标</td><td colspan="3">检测是否支持向获得授权者提供真实的数据记录</td></tr>
<tr><td>前置条件</td><td colspan="3">1) 已经进入系统</td></tr>
<tr><td rowspan="2">测试过程</td><td colspan="2">步骤描述（参考）</td><td>输入数据</td></tr>
<tr><td colspan="2">1) 查询技术文档看是否支持在权限管理中设置账本权限控制
2) 自定义账本的权限，在任意节点构造事务请求，在授权节点/非授权节点查询区块链数据看是否符合要求。调整账本权限级别，将授权节点和非授权节点角色互换，查询区块链数据看是否符合要求</td><td>无</td></tr>
<tr><td>预期结果</td><td colspan="3">仅授权节点可查询账本记录</td></tr>
<tr><td>测试结果</td><td colspan="3">通过</td></tr>
<tr><td>备注</td><td colspan="3">系 统 支 持 对 特 定 节 点 功 能 分 配
 
管理员身份能对系统节点进行授权分配</td></tr>
</table>

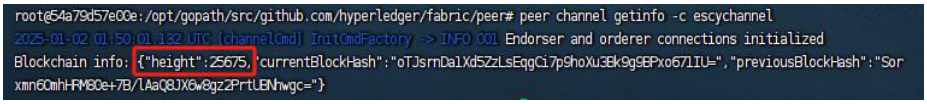
	 普通用户身份能对已分配的节点进行查看 
测试人	**测试时间**

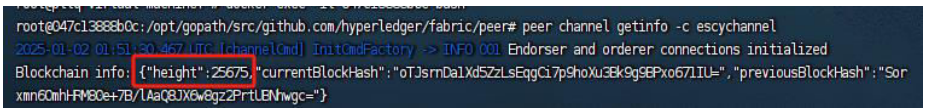
1.8.5 各节点数据一致性

用例编号	BC-T108-C5	**用例名称**	各节点数据一致性
用例类型	必选		
测试目标	检测是否能确保有相同账本记录的各节点数据一致性		
前置条件	1) 已经进入系统		

<table>
<tr><td rowspan="2">测试过程</td><td>步骤描述（参考）</td><td colspan="2">输入数据</td></tr>
<tr><td>1) 查询技术文档最终一致性的保障机制
2) 向随机 N 个节点（n>总数的一半）查询区块链最新状态数据和历史数据看是否符合要求</td><td colspan="2">无</td></tr>
<tr><td>预期结果</td><td colspan="3">账本上查询到的历史数据和最新状态数据一致</td></tr>
<tr><td>测试结果</td><td colspan="3">通过</td></tr>
<tr><td>备注</td><td colspan="3">利用 peer0.org1 查询节点高度为 25675
 

利用 peer0.org2 查询节点高度为 25675
 

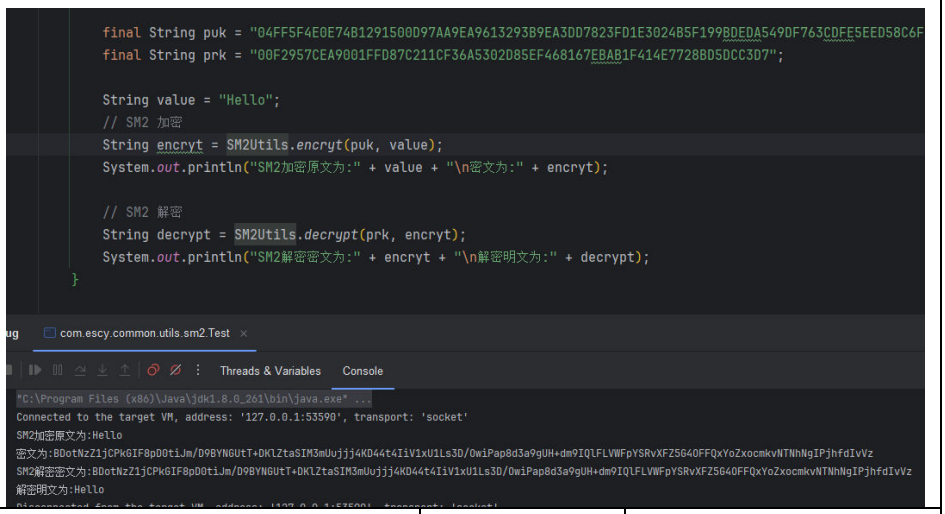
利用 peer0.org3 查询节点高度为 25675
 

利用 peer0.org4 查询节点高度为 25675
 

系统验证各节点数据一致性成功</td></tr>
<tr><td>测试人</td><td></td><td>测试时间</td><td></td></tr>
</table>

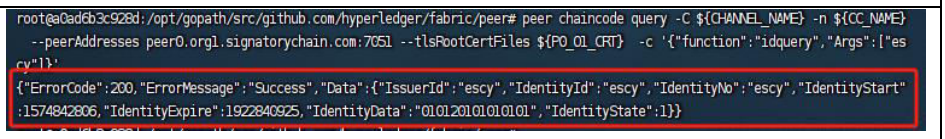
1.9 BC-T109：加密

1.9.1 支持国际主流加密算法和我国商密

<table>
<tr><td>用例编号</td><td>BC-T109-C1</td><td>用例名称</td><td>查询文档验证支持国际主流加密算法和我国商密算法</td></tr>
<tr><td>用例类型</td><td colspan="3">可选</td></tr>
<tr><td>测试目标</td><td colspan="3">查询文档是否支持国际主流加密算法如 AES256 等对称加密算法和 RSA、ECC 等非对称加密算法等加密算法，我国商密算法如 SM4、SM7 等对称加密算法和 SM2、SM9 等非对称加密算法</td></tr>
<tr><td>前置条件</td><td colspan="3">1) 已经进入系统</td></tr>
<tr><td rowspan="2">测试过程</td><td colspan="2">步骤描述（参考）</td><td>输入数据</td></tr>
<tr><td colspan="2">1) 查询设计文档看是否符合要求</td><td>无</td></tr>
<tr><td>预期结果</td><td colspan="3">文档包含对该特性的描述</td></tr>
<tr><td>测试结果</td><td colspan="3">通过/此选项不支持</td></tr>
<tr><td>备注</td><td colspan="3">支持我国商密算法 SM2 非对称加密算法
 
</td></tr>
<tr><td>测试人</td><td></td><td>测试时间</td><td></td></tr>
</table>

1.9.2 持有正确密钥的访问者能解密和访问数据

用例编号	BC-T109-C2	**用例名称**	持有正确密钥的访问者能解密

			和访问数据
用例类型	必选		
测试目标	检测是否对用户数据的访问采用权限控制，持有正确密钥的访问者才能解密和访问数据		
前置条件	1) 已经进入系统		
测试过程	**步骤描述（参考）**	**输入数据**	
	1) 查询设计文档看是否符合要求 2) 用正确的密钥访问区块链系统获取受保护的用户数据	无	
预期结果	正确的密钥能够访问区块链系统获取受保护的用户数据		
测试结果	通过		
备注	 用正确的密钥能够访问区块链系统获取受保护的用户数据		
测试人		**测试时间**	

1.9.3 持有错误密钥访问者不能解密和访问数据

用例编号	BC-T109-C3	**用例名称**	持有错误密钥访问者不能解密和访问数据
用例类型	必选		
测试目标	检测是否对用户数据的访问采用权限控制，持有错误密钥的访问者不能解密和访问数据		

前置条件	1) 已经进入系统		
测试过程	**步骤描述（参考）**		**输入数据**
	1) 查询设计文档看是否符合要求 2) 用错误的密钥访问区块链系统获取受保护的用户数据		无
预期结果	错误的密钥不能够访问区块链系统获取受保护的用户数据		
测试结果	通过		
备注	root@a0ad6b3c928d:/opt/gopath/src/github.com/hyperledger/fabric/peer# peer chaincode query -C ${CHANNEL_NAME} -n ${CC_NAME} --peerAddresses peer0.org1.signatorychain.com:7051 --tlsRootCertFiles ${PO_O2_CRT} -c '{"function":"idquery","Args":["escy"]}' Error: error getting endorser client for query: endorser client failed to connect to peer0.org1.signatorychain.com:7051: failed to create new connection: context deadline exceeded 使用错误的密钥不能够访问区块链系统获取受保护的用户数据		
测试人		**测试时间**	

1.9.4 具备抵御破解的能力

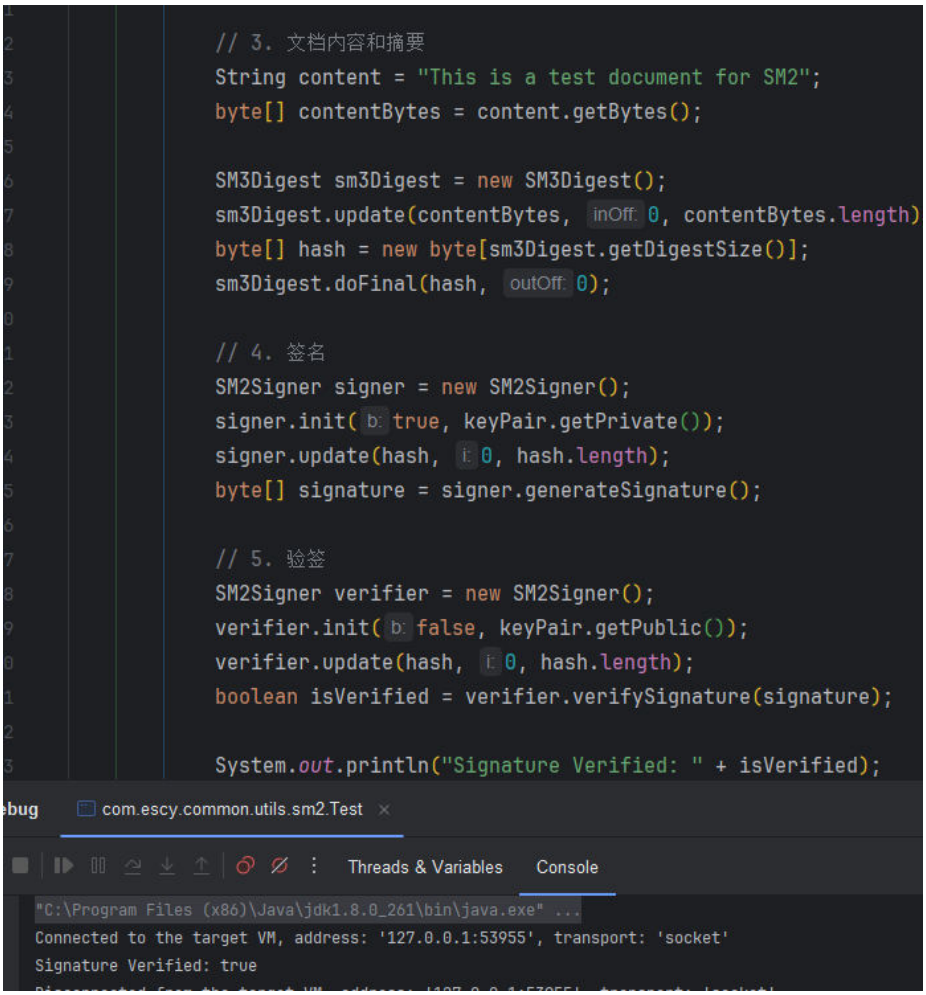
用例编号	BC-T109-C4	**用例名称**	具备抵御破解的能力
用例类型	可选		
测试目标	检测是否采用例如零知识证明、环签名和同态加密等隐私保护算法		
前置条件	1) 已经进入系统		
测试过程	**步骤描述（参考）**		**输入数据**
	1) 查询设计文档看是否符合要求 2) 构造文档声称的包含零知识证明、环签名和同态加密等机制的事务请求，确认事务并正确处理		无

预期结果	事务能被正确处理		
测试结果	通过/此选项不支持		
备注			
测试人		**测试时间**	

1.10 BC-T110：摘要

1.10.1 支持国际主流或我国商密摘要算法

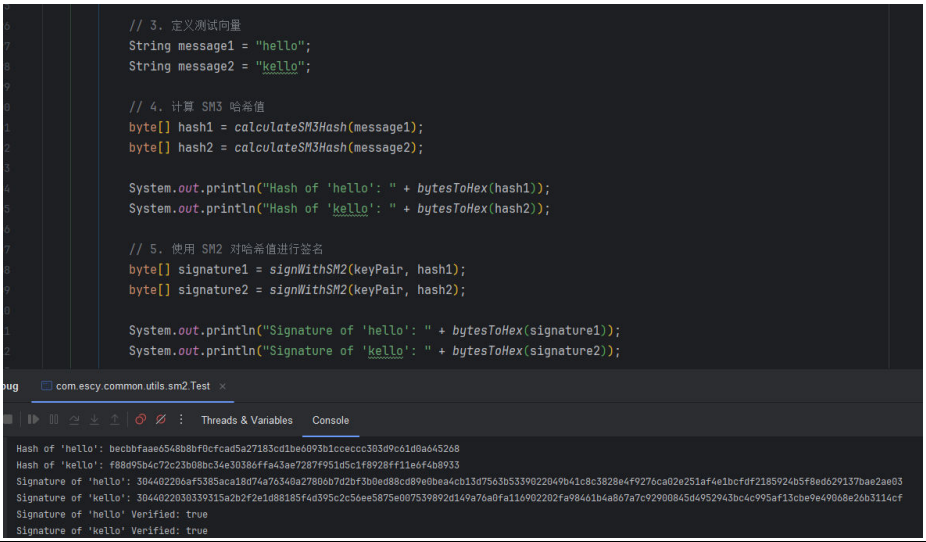
用例编号	BC-T110-C1	**用例名称**	支持国际主流或我国商密摘要算法
用例类型	可选		
测试目标	检测是否区块链系统是否支持国际主流或我国商密摘要算法		
前置条件	1) 已经进入系统		
测试过程	**步骤描述（参考）**	**输入数据**	
	1) 查询文档是否支持使用国际主流或我国商密摘要算法，对测试向量进行哈希运算，与测试向量进行对比，检测是否一致	无	
预期结果	结果一致		
测试结果	通过/此选项不支持		
备注	支持使用我国商密摘要算法进行对比，检测一致		

<table>
<tr><td></td><td colspan="3"> 
</td></tr>
<tr><td>测试人</td><td></td><td>测试时间</td><td></td></tr>
</table>

1.10.2 摘要算法应具备抵御破解的能力

<table>
<tr><td>用例编号</td><td>BC-T110-C2</td><td>用例名称</td><td>摘要算法应具备抵御破解的能力</td></tr>
<tr><td>用例类型</td><td colspan="3">可选</td></tr>
<tr><td>测试目标</td><td colspan="3">检测区块链系统的摘要算法输入值变动后产生不同的运算结果</td></tr>
<tr><td>前置条件</td><td colspan="3">1) 已经进入系统</td></tr>
<tr><td rowspan="2">测试过程</td><td colspan="2">步骤描述（参考）</td><td>输入数据</td></tr>
<tr><td colspan="2">1) 对相差非常小测试向量使用哈希运算</td><td>无</td></tr>
</table>

	如 hello 和 kello	
预期结果	产生显著不同的摘要	
测试结果	通过/此选项不支持	

备注	区块系统使用哈希运算 hello 和 kello,返回不同的结果
	 
	// 3. 定义测试向量
	String message1 = "hello";
	String message2 = "kello";
	
	// 4. 计算 SM3 哈希值
	byte[] hash1 = calculateSM3Hash(message1);
	byte[] hash2 = calculateSM3Hash(message2);
	
	System.out.println("Hash of 'hello': " + bytesToHex(hash1));
	System.out.println("Hash of 'kello': " + bytesToHex(hash2));
	
	// 5. 使用 SM2 对哈希值进行签名
	byte[] signature1 = signWithSM2(keyPair, hash1);
	byte[] signature2 = signWithSM2(keyPair, hash2);
	
	System.out.println("Signature of 'hello': " + bytesToHex(signature1));
	System.out.println("Signature of 'kello': " + bytesToHex(signature2));
	com.escy.common.utils.sm2.Test
	Threads & Variables Console
	Hash of 'hello': becbbfaae6548b8bf0cfcad5a27183cd1be6093b1cceccc303d9c61d0a645268
	Hash of 'kello': f88d95b4c72c23b08bc34e30386ffa43ae7287f951d5c1f8928ff11e6f4b8933
	Signature of 'hello': 304402206af5385aca18d74a76340a27806b7d2bf3b0ed88cd89e0bea4cb13d7563b533902204 9b41c8c3828e4f9276ca02e251af4e1bcfdf2185924b5f8ed629137bae2ae03
	Signature of 'kello': 3044022030339315a2b2f2e1d88185f4d395c2c56ee5875e007539892d149a76a0fa116902202fa98461b4a867a7c92900845d4952943bc4c995af13cbe9e49068e26b3114cf
	Signature of 'hello' Verified: true
	Signature of 'kello' Verified: true	

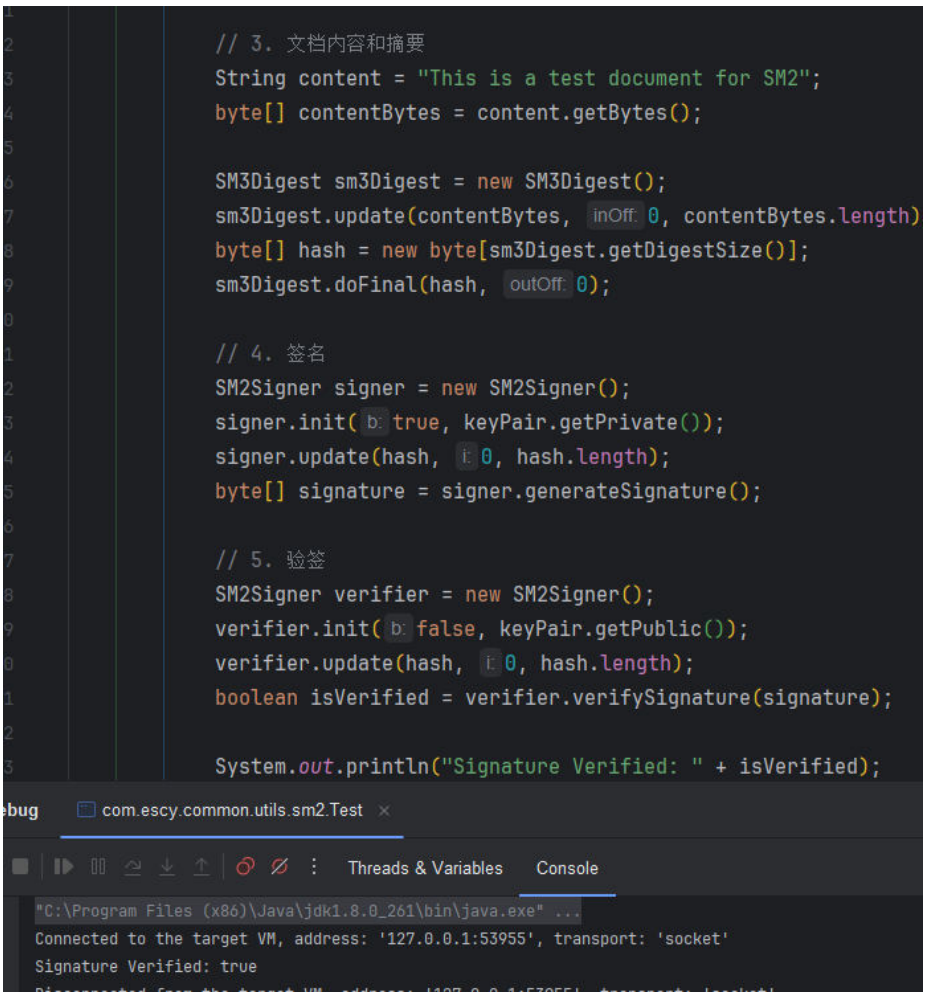
测试人		**测试时间**

1.11 BC-T111：数字签名

1.11.1 支持国际主流和我国商密数字签名算法

用例编号	BC-T111-C1	**用例名称**	支持国际主流和我国商密数字签名算法
用例类型	可选		
测试目标	查询系统是否支持国际主流和我国商密数字签名算法		
前置条件	1) 已经进入系统		
测试过程	**步骤描述（参考）**	**输入数据**	
	1) 查询设计文档看是否符合要求	无	

预期结果	文档包含对该特性的描述		
测试结果	通过/此选项不支持		

备注	系统支持我国商密 SM2 数字签名算法
	 
	// 3. 文档内容和摘要
	String content = "This is a test document for SM2";
	byte[] contentBytes = content.getBytes();
	
	SM3Digest sm3Digest = new SM3Digest();
	sm3Digest.update(contentBytes, inOff: 0, contentBytes.length);
	byte[] hash = new byte[sm3Digest.getDigestSize()];
	sm3Digest.doFinal(hash, outOff: 0);
	
	// 4. 签名
	SM2Signer signer = new SM2Signer();
	signer.init(b: true, keyPair.getPrivate());
	signer.update(hash, i: 0, hash.length);
	byte[] signature = signer.generateSignature();
	
	// 5. 验签
	SM2Signer verifier = new SM2Signer();
	verifier.init(b: false, keyPair.getPublic());
	verifier.update(hash, i: 0, hash.length);
	boolean isVerified = verifier.verifySignature(signature);
	
	System.out.println("Signature Verified: " + isVerified);
	ebug com.escy.common.utils.sm2.Test
	Threads & Variables Console
	"C:\Program Files (x86)\Java\jdk1.8.0_261\bin\java.exe" ...
	Connected to the target VM, address: '127.0.0.1:53955', transport: 'socket'
	Signature Verified: true		

测试人		**测试时间**	

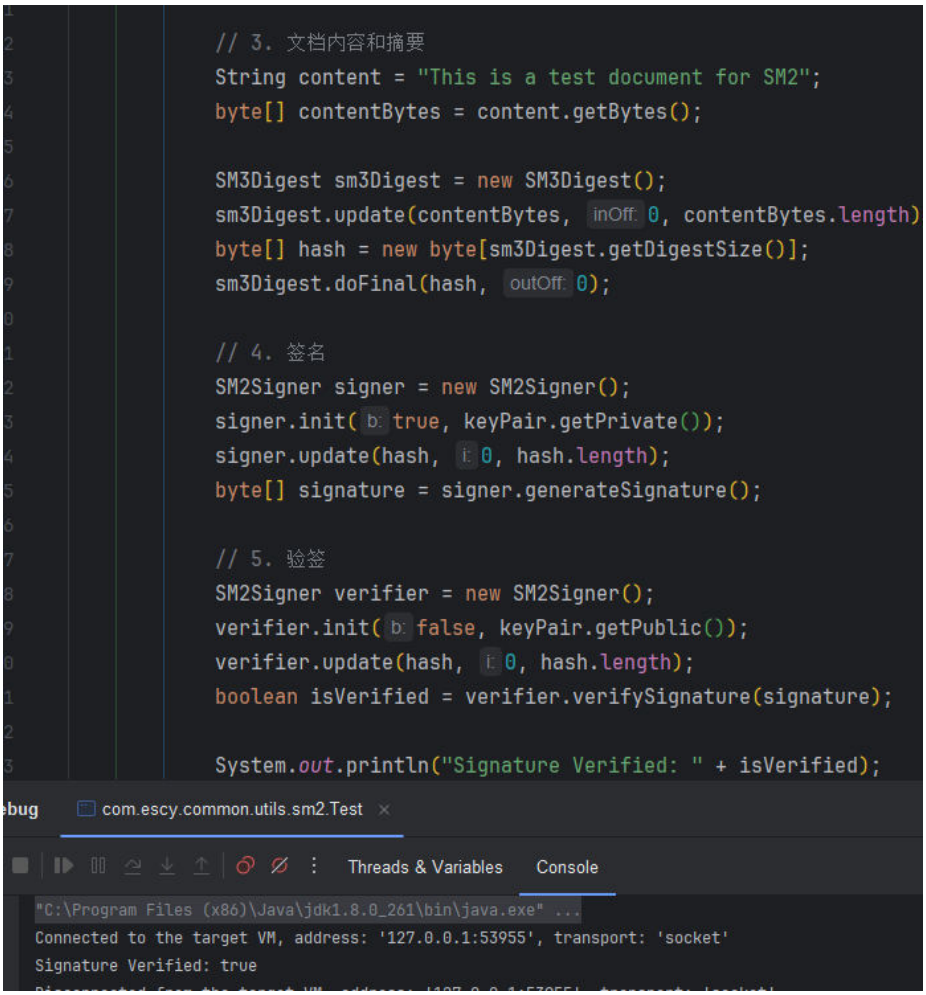
1.11.2 对区块链上数据进行签名

用例编号	BC-T111-C2	**用例名称**	对区块链上数据进行签名
用例类型	可选		
测试目标	使用正确的密钥，检测能否对数据签名		
前置条件	1) 已经进入系统		

测试过程	**步骤描述（参考）**		**输入数据**
	1) 持有正确的密钥对数据使用签名		无
预期结果	能够产生签名		
测试结果	通过/此选项不支持		
备注			
测试人		**测试时间**	

1.11.3 数字签名的验签

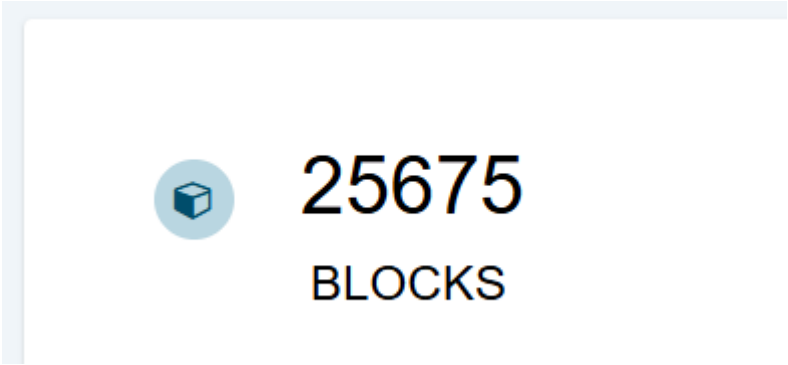
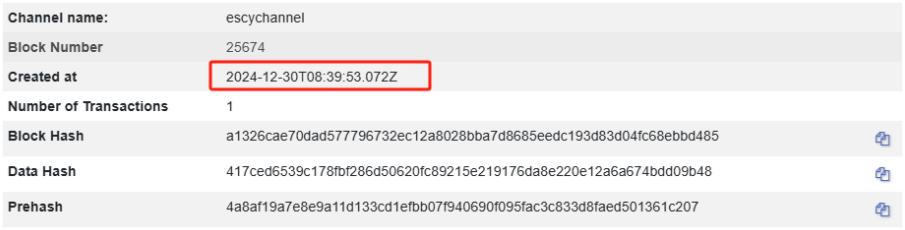
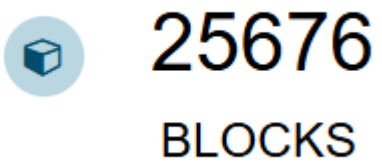
用例编号	BC-T111-C3	**用例名称**	数字签名的验签
用例类型	可选		
测试目标	检测是否能对区块链上的数字签名进行验签		
前置条件	1) 已经进入系统		
测试过程	**步骤描述（参考）**		**输入数据**
	1) 持有正确的公钥对签名可以验签		无
预期结果	能够通过验签		
测试结果	通过/此选项不支持		
备注	系统支持对区块链上的数字签名进行验签		

	
测试人	**测试时间**

1.12 BC-T112：时序服务

1.12.1 统一账本记录

用例编号	BC-T112-C1	**用例名称**	统一账本记录
用例类型	必选		
测试目标	检测是否支持统一账本记录时序		
前置条件	1) 已经进入系统		
测试过程	**步骤描述（参考）**	**输入数据**	
	1) 查询设计文档是否符合要求	无	

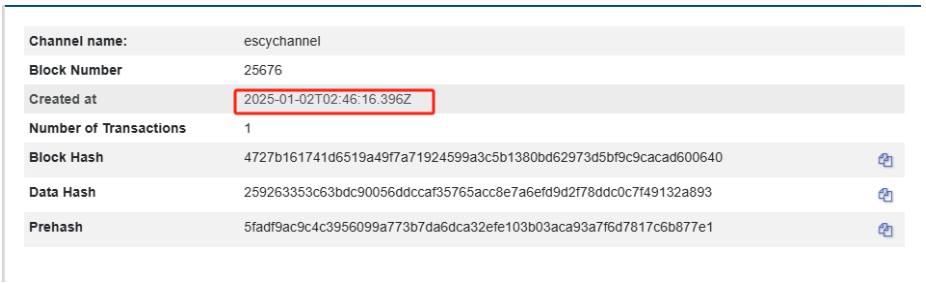
	2）查询区块链数据是否符合相关要求	
预期结果	查看区块链数据的区块时间戳是顺序增加的，对于非顺序增加的记录说明原因，该原因应在设计文档中体现	
测试结果	通过	
备注	数据上链前区块的高度为 25675   进行一条数据上链操作  数据上链后区块的高度为 25676 

	 随着区块高度增加，时间戳顺序增加

测试人		**测试时间**	

1.12.2 时序容错性

用例编号	BC-T112-C2	**用例名称**	时序容错性
用例类型	必选		
测试目标	检测是否具备时序容错性		
前置条件	1) 已经进入系统		
测试过程	**步骤描述（参考）**		**输入数据**
	1) 查询设计文档是否符合要求 2) 构造错误时序的事务请求，测试其处理结果。		无
预期结果	修改本地时间，构建先于区块链当前时间戳的数据写入请求，系统成功把区块写入区块链		
测试结果	通过		
备注	修改本地时间为 2024 年 1 月 2 号 		

	提交一条数据上链成功  查看上链后的区块数据时间戳并不随本地时间变化，还是自然时间 
测试人		**测试时间**	

1.12.3 第三方时序服务

用例编号	BC-T112-C3	**用例名称**	第三方时序服务
用例类型	可选		
测试目标	检测是否支持第三方时序服务		
前置条件	1) 已经进入系统		
测试过程	**步骤描述（参考）**		**输入数据**
	1) 查询设计文档是否符合要求 2) 测试是否支持第三方时序服务		无
预期结果	连接第三方时序服务，发起请求，请求的处理结果（新增区块的时间戳）与第三方时序服务发起时间一致		
测试结果	通过/此选项不支持		
备注	系统支持第三方时序服务		

	root@pttq-virtual-machine:~# sudo systemctl status ntp ● ntp.service - Network Time Service Loaded: loaded (/lib/systemd/system/ntp.service; enabled; vendor preset: enabled) Active: active (running) since Thu 2025-01-02 16:50:29 CST; 1min 54s ago Docs: man:ntpd(8) Main PID: 95283 (ntpd) Tasks: 2 (limit: 2446) CGroup: /system.slice/ntp.service └─95283 /usr/sbin/ntpd -p /var/run/ntpd.pid -g -u 123:128

测试人		**测试时间**	

1.13 BC-T113：智能合约

1.13.1 提供编程语言支持

用例编号	BC-T113-C1	**用例名称**	提供编程语言支持
用例类型	可选		
测试目标	检测是否提供编程语言支持		
前置条件	1) 已经进入系统		
测试过程	**步骤描述（参考）**		**输入数据**
	1) 查询设计文档是否支持使用一到多种编程语言开发智能合约，并在系统验证		无
预期结果	支持使用一到多种编程语言开发智能合约		
测试结果	通过/此选项不支持		
备注	系统支持 go，java 等多种语言开发智能合约		

	
测试人	**测试时间**

1.13.2 提供配套的集成开发环境

用例编号	BC-T113-C2	**用例名称**	提供配套的集成开发环境
用例类型	可选		
测试目标	检测是否提供配套的集成开发环境		
前置条件	1) 已经进入系统		
测试过程	**步骤描述（参考）**	**输入数据**	
	1) 查看开发文档，判定是否提供了集成开发环境，并进行验证	无	
预期结果	支持符合开发文档描述的集成开发环境		

测试结果	通过/此选项不支持		
备注			
测试人		**测试时间**	

1.13.3 合约内容静态和动态检查

用例编号	BC-T113-C3	**用例名称**	合约内容静态和动态检查
用例类型	可选		
测试目标	检测是否提供支持采用系统指定的静态和动态检查工具检查合约内容		
前置条件	1) 已经进入系统		
测试过程	**步骤描述（参考）**	**输入数据**	
	1) 进入智能合约编辑工具 2) 采用系统指定的静态和动态检查工具检查合约内容	无	
预期结果	有工具可对智能合约使用静态和动态检查，且输出结果符合预期		
测试结果	通过/此选项不支持		
备注			
测试人		**测试时间**	

1.13.4 支持运行载体

用例编号	BC-T113-C4	**用例名称**	支持运行载体
用例类型	可选		
测试目标	检测是否提供运行载体支持，如虚拟机等		

前置条件	1) 已经进入系统		
测试过程	**步骤描述（参考）**		**输入数据**
	1) 根据系统文档里关于运行载体支持章节，把智能合约提交到系统中运行		无
预期结果	智能合约可以在系统中正确执行，说明系统提供了运行载体支持		
测试结果	通过/此选项不支持		
备注			
测试人		**测试时间**	

1.13.5 外部数据源和智能合约交互

用例编号	BC-T113-C5	**用例名称**	外部数据源和智能合约交互
用例类型	可选		
测试目标	检测对于与区块链系统外部数据使用交互的智能合约，外部数据源的影响范围应仅限于智能合约范围内，不应影响区块链系统的整体运行		
前置条件	1) 已经进入系统		
测试过程	**步骤描述（参考）**		**输入数据**
	1) 修改外部数据源的配置（修改存储值）和数据，运行和外部数据源相关的智能合约		无
预期结果	外部数据源的影响范围仅限于智能合约范围内，系统无异常		
测试结果	通过/此选项不支持		
备注			

测试人		**测试时间**	

1.13.6 合约防篡改

用例编号	BC-T113-C6	**用例名称**	合约防篡改
用例类型	可选		
测试目标	检测是否能防止对合约内容使用篡改		
前置条件	1) 已经进入系统		
测试过程	**步骤描述（参考）**		**输入数据**
	1) 直接在存储里修改智能合约的内容。然后执行该智能合约		无
预期结果	合约内容应有摘要，共识等机制保证不能被篡改，篡改后的合约无法被校验通过，且运行不正确		
测试结果	通过/此选项不支持		
备注			
测试人		**测试时间**	

1.13.7 多方共识下的合约升级

用例编号	BC-T113-C7	**用例名称**	多方共识下的合约升级
用例类型	可选		
测试目标	检测是否支持多方共识下的合约内容升级		
前置条件	1) 已经进入系统		
测试过程	**步骤描述（参考）**		**输入数据**

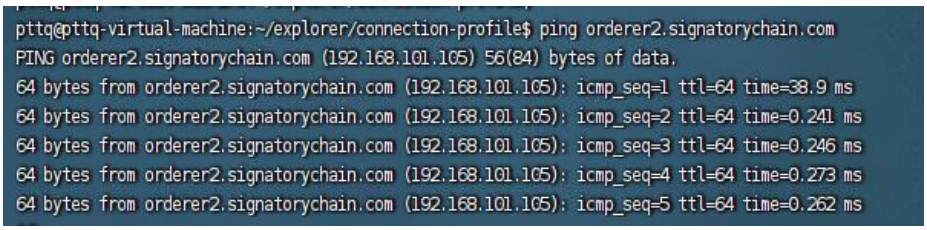
	1) 升级某个特定合约	无
预期结果	确认其升级过程被多方共识后才写入存储，并有多方共识的记录	
测试结果	通过/此选项不支持	
备注		
测试人		**测试时间**

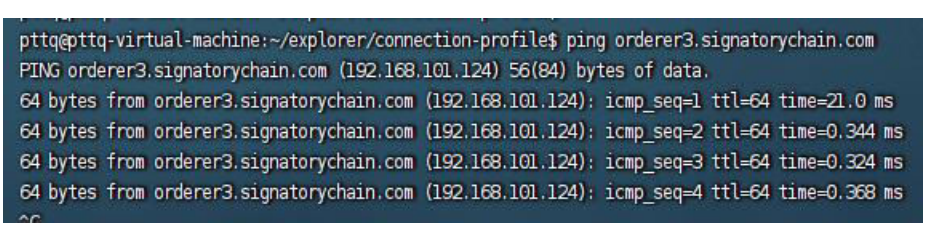
1.13.8 账本中写入合约内容

用例编号	BC-T113-C8	**用例名称**	账本中写入合约内容
用例类型	可选		
测试目标	提交包含特定内容的事务，检测能否调用或部署合约		
前置条件	1) 已经进入系统		
测试过程	**步骤描述（参考）**		**输入数据**
	1) 提交包含特定内容的事务，调用或部署合约		无
预期结果	如事务为部署合约，则存储里包含合约本身的代码等内容。如事务为调用合约写入账本存储的接口功能，则账本存储里应有合约处理后生成并存储的内容		
测试结果	通过/此选项不支持		
备注			
测试人		**测试时间**	

1.14 BC-T114：对等网络

1.14.1 点对点之间的通信

<table>
<tr><td>用例编号</td><td>BC-T114-C1</td><td>用例名称</td><td>点对点之间的通信</td></tr>
<tr><td>用例类型</td><td colspan="3">必选</td></tr>
<tr><td>测试目标</td><td colspan="3">检测是否能够使用点对点之间的通信</td></tr>
<tr><td>前置条件</td><td colspan="3">1) 已经进入系统</td></tr>
<tr><td rowspan="2">测试过程</td><td colspan="2">步骤描述（参考）</td><td>输入数据</td></tr>
<tr><td colspan="2">1) 查看开发文档
2) 区块链各节点间网络是否互通。发包，ping（抽样），单节点之间的互通</td><td>无</td></tr>
<tr><td>预期结果</td><td colspan="3">采用抽样的方式从一个 ping 其他节点，ping 通过</td></tr>
<tr><td>测试结果</td><td colspan="3">通过</td></tr>
<tr><td>备注</td><td colspan="3">从节点 1 orderer1.signatorychain.com ping 节点 2 orderer2.signatorychain.com
 

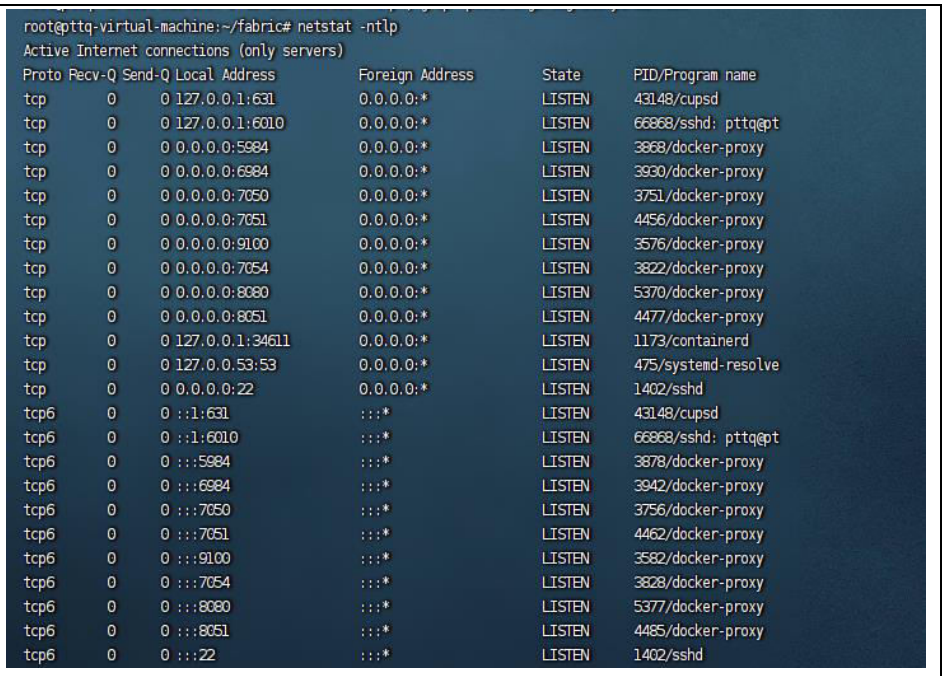
从节点 1 orderer1.signatorychain.com ping 节点 3 orderer3.signatorychain.com
 

从节点 1 orderer1.signatorychain.com ping 节点 4</td></tr>
</table>

	orderer4.signatorychain.com  节点之间能相互 ping 通		
测试人		**测试时间**	

1.14.2 点对点之间的通信接口

用例编号	BC-T114-C2	**用例名称**	点对点之间的通信接口
用例类型	必选		
测试目标	区块链各节点指定的端口号是否正常开启		
前置条件	1) 已经进入系统		
测试过程	**步骤描述（参考）**		**输入数据**
	1) 查看开发文档 2) 区块链各节点指定的端口号是否正常开启		无
预期结果	用端口扫描工具扫描区块链节点端口，开放端口与文档说明一致		
测试结果	通过		

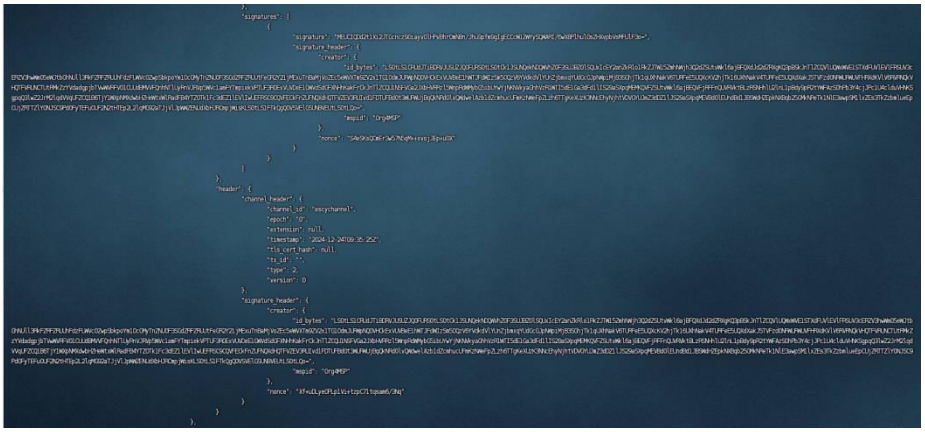
备注	 一个节点有 2 个端口，一个监听端口，一个发送端口，端口处于监听状态		
测试人		**测试时间**	

1.14.3 点对点之间的安全通信

用例编号	BC-T114-C3	**用例名称**	点对点之间的安全通信
用例类型	必选		
测试目标	检测区块链网络中各节点通信是否安全		
前置条件	1) 已经进入系统		
测试过程	**步骤描述（参考）**	**输入数据**	
	1) 查看开发文档 2) 区块链网络中各节点通信是否有安全机制，如使用 TLS 或 SSL 证书	无	
预期结果	配置中有开发文档中的安全机制		
测试结果	通过		

备注	 使用 TLS 证书进行各节点之间的安全通信
测试人		**测试时间**	

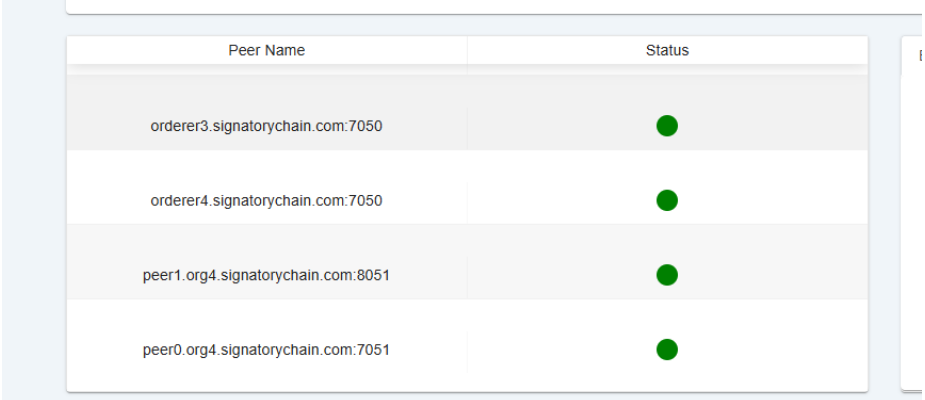
1.14.4 点对点通信多播能力

用例编号	BC-T114-C4	**用例名称**	点对点通信多播能力
用例类型	必选		
测试目标	检测是否能够提供点对点通信基础上的多播能力		
前置条件	1) 已经进入系统		
测试过程	**步骤描述（参考）**	**输入数据**	
	1) 查看开发文档 2) 采用抽样方法选取节点向网络中多播数据包	无	
预期结果	所有节点能正确收到多播消息		
测试结果	通过		
备注			

	通过查询区块详情信息，里面包含 4 个共识节点的签名信息，证明共识节点之间存在通信多播能力		
测试人		**测试时间**	

1.14.5 支持对节点的动态添加的识别

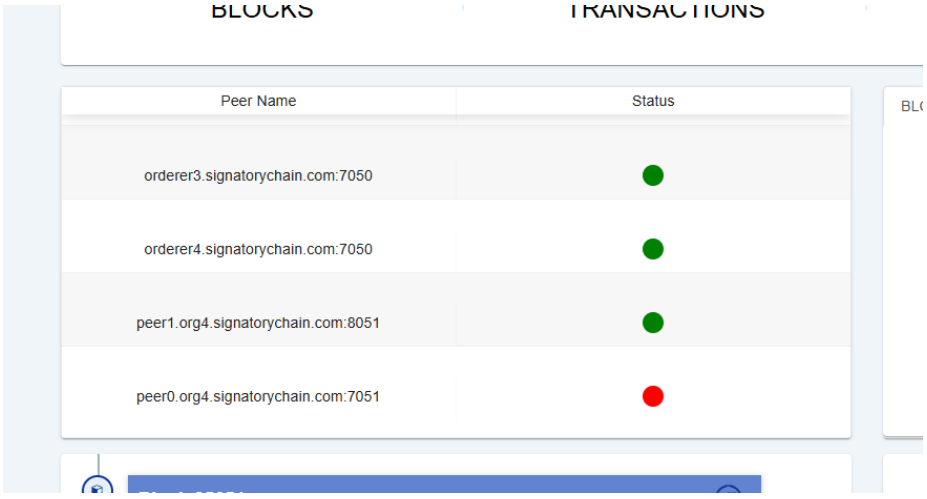
用例编号	BC-T114-C5	**用例名称**	支持对节点的动态添加的识别
用例类型	必选		
测试目标	检测能否部署新节点加入已有区块链网络		
前置条件	1) 已经进入系统		
测试过程	**步骤描述（参考）**	**输入数据**	
	1) 部署新节点加入已有区块链网络	无	
预期结果	新节点与已有区块链网络成功建立连接		
测试结果	通过		

备注	部署新节点加入已有区块链网络
	`pttq@pttq-virtual-machine:~/fabric$ sudo docker start peer0.org4.signatorychain.com`
	`peer0.org4.signatorychain.com`
	查询节点状态，新节点与已有区块链网络成功建立连接
	 		

测试人		**测试时间**	

1.14.6 支持对节点的动态减少的识别

<table>
<tr><td>用例编号</td><td>BC-T114-C6</td><td>用例名称</td><td>支持对节点的动态减少的识别</td></tr>
<tr><td>用例类型</td><td colspan="3">必选</td></tr>
<tr><td>测试目标</td><td colspan="3">检测能否在已有区块链网络中随机选取节点退出网络</td></tr>
<tr><td>前置条件</td><td colspan="3">1) 已经进入系统</td></tr>
<tr><td rowspan="2">测试过程</td><td colspan="2">步骤描述（参考）</td><td>输入数据</td></tr>
<tr><td colspan="2">1) 在已有区块链网络中随机选取节点退出网络</td><td>无</td></tr>
<tr><td>预期结果</td><td colspan="3">退出后与区块链网络无连接</td></tr>
<tr><td>测试结果</td><td colspan="3">通过</td></tr>
<tr><td>备注</td><td colspan="3">在已有区块链网络中随机选取节点退出网络
 

查询节点状态，退出后与区块链网络无连接
 
</td></tr>
</table>

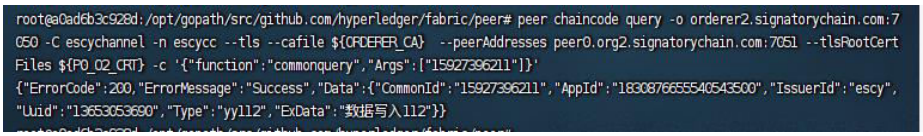
测试人		**测试时间**	

1.15 BC-T115：储存

1.15.1 对等网络能够被每个节点部署并使用

用例编号	BC-T115-C1	**用例名称**	对等网络能够被每个节点部署并使用
用例类型	必选		
测试目标	检测能否在选取的节点上写入若干区块		
前置条件	1) 已经进入系统		
测试过程	**步骤描述（参考）**	**输入数据**	
	1) 查看开发文档 2) 区块链节点能否正确将数据写入本地存储	无	
预期结果	在抽样选取的节点上写入若干区块，能成功写入		
测试结果	通过		
备注	在节点 1 数据上链成功 		
测试人		**测试时间**	

1.15.2 对等网络能够被每个节点查询

用例编号	BC-T115-C2	**用例名称**	对等网络能够被每个节点查询
用例类型	必选		
测试目标	检测能否在选取的节点上查询区块信息		
前置条件	1) 已经进入系统		
测试过程	**步骤描述（参考）**		**输入数据**
	1) 查看开发文档 2) 区块链节点能否正确将本地数据读取		无
预期结果	在抽样选取的节点上查询区块信息，能成功查询		
测试结果	通过		
备注	在节点 2 上能成功查询到区块信息 		
测试人		**测试时间**	

1.15.3 能够提供高效稳定的数据服务

用例编号	BC-T115-C3	**用例名称**	能够提供高效稳定的数据服务
用例类型	必选		
测试目标	检测能否对存储进行读写操作，分别记录读写速率		
前置条件	1) 已经进入系统		
测试过程	**步骤描述（参考）**		**输入数据**
	1) 查看开发文档 2) 存储读 / 写是否满足区块链系统		无

预期结果	对存储使用读写操作，分别记录读写速率		
测试结果	通过		
备注	通过监控界面查询系统读写速率 		
测试人		**测试时间**	

1.15.4 能够提供安全的数据服务

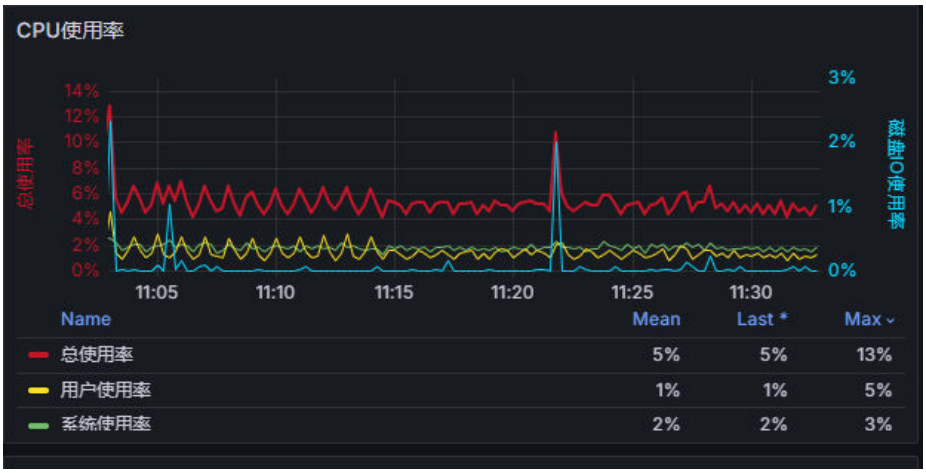
用例编号	BC-T115-C4	**用例名称**	能够提供安全的数据服务
用例类型	可选		
测试目标	存储 RTO / RPO 测试，测试其 RTO 和 RPO 是否与文档说明一致		
前置条件	1) 已经进入系统		
测试过程	**步骤描述（参考）**	**输入数据**	
	1) 查看开发文档 2) 存储 RTO / RPO 测试	无	
预期结果	对存储 RTO / RPO 测试，能够满足说明文档要求		

测试结果	通过/此选项不支持		
备注			
测试人		**测试时间**	

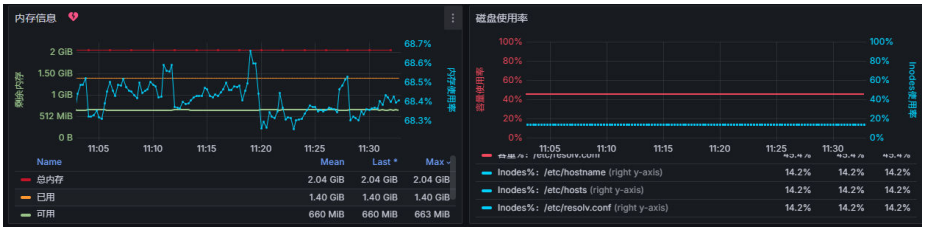
1.16 BC-T116：计算

1.16.1 区块链节点运行环境监控

用例编号	BC-T116-C1	**用例名称**	区块链节点运行环境监控
用例类型	必选		
测试目标	检测单一节点在区块链系统运行时的 CPU 使用率以及内存外存使用情况		
前置条件	1) 已经进入系统		
测试过程	**步骤描述（参考）**		**输入数据**
	1) 查看开发文档 2) 检测单一节点在区块链系统运行时的 CPU 使用率以及内存外存使用情况		无
预期结果	区块链系统运行一段时间后该节点无明显异常		
测试结果	通过		
备注	通过监控系统查询 CPU 使用率		

	
测试人	**测试时间**	

1.16.2 区块链节点计算能力

用例编号	BC-T116-C2	**用例名称**	区块链节点计算能力
用例类型	必选		
测试目标	检测对等网络中，计算能力能否满足每个节点要求		
前置条件	1) 已经进入系统		
测试过程	**步骤描述（参考）**	**输入数据**	
	1) 抽样检测部分节点在区块链系统运行时的CPU使用率以及内存外存使用情况	无	
预期结果	区块链系统运行一段时间后被抽样的节点无明显异常		
测试结果	通过		
备注	通过监控系统查询内存外存使用率 		

测试人		**测试时间**	