

★完全公开

# 天擎 V10 集群部署 升级指导 V1.0

创建时间：2022 年 9 月 20 日

修改时间：2025 年 7 月 25 日

网络安全服务热线：95015

公司网址：<https://www.qianxin.com/>

联系地址：北京市西城区西直门外南路 26 号院 1 号楼

邮编：100044

## ● 版权声明

奇安信集团，保留所有权利。

奇安信集团及其关联公司对其发行的或与合作伙伴共同发行的产品享有版权，本文中出现的任何文字叙述、文档格式、插图、照片、方法、过程描述等内容，除另有特别注明外，所有版权均属奇安信集团及其关联公司所有；受各国版权法及国际版权公约的保护。

对于上述版权内容，任何个人、机构未经奇安信集团或其关联公司的书面授权许可，不得以任何方式复制或引用本文的任何片断；超越合理使用范畴、并未经上述公司书面许可的使用行为，奇安信集团或其关联公司均保留追究法律责任的权利。

由于产品版本升级或其它原因，本手册内容会不定期进行更新。除非另有约定，本手册仅作为使用指导，本手册中的所有陈述、信息和建议不构成任何明示或暗示的担保。

## ● 免责声明

本免责声明（“**本声明**”）适用于奇安信集团（包括但不限于奇安信科技集团股份有限公司、奇安信网神信息技术（北京）股份有限公司、北京网康科技有限公司，以及前述主体直接或者间接控制的法律实体）旗下推出的全部产品和服务（以下统称“**本产品**”）。如您使用前述产品，即表示您同意接受本声明的一切内容。如果您不同意接受，请立即停止使用相关产品。

奇安信集团有权随时自行决定修改、添加或删除本声明的全部或部分内容。您有责任定期检查免责声明部分的内容，以了解是否发生了变更。如您在我们发布变更后继续使用本产品，即表示您接受并同意这些变更。

1. 您明确理解并同意，**本产品按“现状”提供**，不存在任何形式的明示或暗示保证，并且在适用法律允许的最大范围内，奇安信集团不提供任何明示或暗示的陈述或保证，包括但不限于有关适销性、适用于特定目的以及不侵犯第三方权利的保证。奇安信集团不保证产品中所含的功能将满足您的全部要求，也不保证您对本产品的使用不会中断或出错。**选择本产品来达到预期结果，以及安装、使用本产品并获取结果所带来的所有责任和风险由您承担。**
2. 奇安信集团承诺致力于不断提升产品的质量，本产品是在现有技术水平基础上提供的，但奇安信集团无法保证您使用本产品将完全符合您的期望，包括但不限于不能保证您【通过使用产品能够发现所有的安全漏洞以及能检测到所有的入侵威胁，检测到的入侵威胁不保证完全正确】，**您理解并同意，出现前述不符合您对产品期望的情形不视为奇安信集团违约。**
3. 您明确理解并同意，您在使用本产品过程中可能发生不可抗力或不可预见的情形，包括但不限于：**1)被某些未经许可的个人、团体或机构通过某种渠道获得或篡改；2)因通信繁忙出现延迟，或因其他原因出现中断、停顿或数据不完全、数据错误等情况，从而使交易出现错误、延迟、中断或停顿；3)因地震、火灾、台风及其他各种不可抗力因素引起的停电、网络系统故障、电脑故障等；4)计算机系统可能因存在性能缺陷、质量问题、计算机病毒、硬件故障及其他原因；黑客攻击、计算机病毒侵入或发作等非可归责于奇安信集团的原因；5)政府管制、网络故障、国家政策变化、法律法规之变化等。**如发生不可抗力或不可预见的情形，奇安信集团将尽最大努力予以补救，但奇安信集团对于因不可抗力和不可预见的情形造成的各类直接或间接损失，均不承担任何责任。
4. 对于任何本产品的使用行为，包括但不限于您自身和/或任何第三方的行为，奇安信集团均不承担任何责任。
5. 对于从非奇安信集团指定途径以及从非奇安信集团发行的介质上获得的**本产品**，奇安信集团无法保证其是否感染计算机病毒、是否隐藏有伪装的特洛

伊木马程序或者黑客软件。使用此类产品，将可能导致不可预测的风险，建议用户不要轻易下载、安装、使用，奇安信集团不承担任何由此产生的一切法律责任。

6. 上述免责声明适用于因任何性能故障、错误、遗漏、中断、删除、缺陷、操作或传输延迟、电脑病毒、通信线路故障、失窃、毁坏、未经授权的访问、篡改或使用（无论是出于违约、侵权、疏忽或任何其他诉因）而导致的任何损害、责任或伤害。
  7. 奇安信集团保留在不发布通知的情况下随时采取以下行动的权利：在执行常规或非常规维护、错误纠正或其他更改所必需时，中断或修改本产品的任何组成部分的运行或功能。
  8. 本声明受中华人民共和国法律的约束并依据其解释。
  9. 在法律允许的最大范围内，本声明最终解释权归奇安信集团享有。
-

# 目录

|                                    |           |
|------------------------------------|-----------|
| <b>1 文档说明.....</b>                 | <b>6</b>  |
| 1.1 文档介绍 .....                     | 6         |
| 1.2 适用范围 .....                     | 6         |
| 1.3 名词解释 .....                     | 6         |
| 1.4 参考文档 .....                     | 6         |
| <b>2 升级准备.....</b>                 | <b>7</b>  |
| <b>3 环境检查.....</b>                 | <b>10</b> |
| <b>4 具体升级操作 .....</b>              | <b>13</b> |
| 4.1 上传安装包到指定目录.....                | 13        |
| 4.2 运行升级命令 .....                   | 13        |
| 4.3 升级成功 .....                     | 14        |
| <b>5 服务功能验证 .....</b>              | <b>15</b> |
| <b>6 应急预案.....</b>                 | <b>16</b> |
| 6.1 升级后，服务组件资源占用异常及接口占用异常 .....    | 16        |
| 6.2 升级后，客户端升级后带宽占满 .....           | 16        |
| <b>7 接入点升级 .....</b>               | <b>18</b> |
| 7.1 接入点 10.3.0.3201 之前版本升级.....    | 18        |
| 7.2 接入点 10.3.0.3201 及以后版本的升级 ..... | 18        |
| <b>8 FAQ .....</b>                 | <b>24</b> |
| 8.1 其它不支持的版本如何升级？ .....            | 24        |
| 8.2 如何理解标准安装部署？ .....              | 24        |
| 8.3 如果出现问题是否有日志？ .....             | 24        |

## 1 文档说明

### 1.1 文档介绍

本文档作为天擎 V10 产品集群部署升级时使用，用于确认升级操作完整、正确，从而提高升级部署工作的效率。

### 1.2 适用范围

适用奇安信内部服务体系升级产品时使用，作为内部文档用于产品升级阶段使用，以天擎产品 V10.0R7P2101 系列（10.7.0.2xxx 版本）为基准进行编写。本文档将集群部署升级要求项进行了整理，详细要求建议先通读天擎 V10 产品集群部署手册，基本可满足大多数项目升级实施场景。

### 1.3 名词解释

| 名词     | 名词说明                                                                                |
|--------|-------------------------------------------------------------------------------------|
| 天擎 V10 | 特指天擎终端安全管理系统的 V10.0 版本产品，作为该产品、系统的简称。                                               |
| 天擎     | 特指天擎终端安全管理系统的 V10.0 版本产品，作为该产品、系统的简称。                                               |
| 管理中心   | 天擎产品采用 B/S 架构设计，服务端通过 web 形式进行管理，特指服务器的 web。                                        |
| 管理系统   | 天擎产品的服务器端整体，包括 web 界面和后台组件和数据等整体内容。                                                 |
| 客户端    | 天擎产品的客户端，采用 C/S 架构，客户端用于部署到希望被管理、防护等业务诉求的操作系统上，通过管理中心进行统一管理、日志采集等需求的形式。             |
| 终端     | 泛指需要安装天擎客户端的 PC、服务器、虚拟机、自助等类型设备，架构上分为 X86 和信创终端，操作系统分为 Windows、Linux、macOS、类 Unix 等 |
| 许可证    | 由天擎产品所属公司提供用于天擎产品功能授权，购买产品后为客户提供唯一序列号和产品购买信息，通过激活和绑定后可下载许可证文件，导入管理中心使用。             |
| 单机部署   | 是指天擎管理系统部署在一台操作系统上，管理系统组件未进行分布式部署，产品设计上单机部署过程较为简化，部分模块不支持单机部署（例如高级版 EDR、DLP 事件溯源）。  |
| 集群部署   | 管理系统组件分别部署在不同的操作系统上，可采用分布式，支持高可用，目前仅支持 Linux 类的操作系统平台上部署，支持所有功能模块。                  |

### 1.4 参考文档

部署手册：[文档管理 - 文档中心 \(qianxin-inc.cn\)](#)

## 2 升级准备

1. 确认升级版本，以下不代表可以直接升级，具体以 KB 集群升级路线为准：

| 管理中心支持升级版本  | 管理中心升级后版本   |
|-------------|-------------|
| 10.7.0.2704 | 10.7.0.2800 |
| 10.7.0.2700 |             |
| 10.7.0.2602 |             |
| 10.7.0.2600 |             |
| 10.7.0.2500 |             |
| 10.7.0.2400 |             |
| 10.7.0.2310 |             |
| 10.7.0.2306 |             |
| 10.7.0.2302 |             |
| 10.7.0.2300 |             |
| 10.7.0.2200 |             |
| 10.7.0.2101 |             |
| 10.7.0.2000 |             |
| 10.7.0.1660 |             |
| 10.7.0.1600 |             |
| 10.7.0.1050 |             |
| 10.7.0.1000 |             |
| 10.6.0.2600 |             |
| 10.6.0.2504 |             |
| 10.6.0.2400 |             |
| 10.6.0.2000 |             |
| 10.6.0.1100 |             |
| 10.6.0.1000 |             |
| 10.5.0.3000 |             |
| 10.5.0.2000 |             |
| 10.5.0.1000 |             |
| 10.3.0.5000 |             |
| 10.3.0.4200 |             |
| 10.3.0.4100 |             |
| 10.3.0.4000 |             |
| 10.3.0.3200 |             |

此升级操作手册仅支持以上版本标准化安装部署后，升级到 **10.7.0.2800** 版本。升级路线以 **KB45768** 为准

标准化安装部署，指环境的第一个初始版本为如下版本：**10.3.0.3200**，**10.3.0.4000**，**10.3.0.4100**，**10.3.0.4200**，**10.3.0.5000**，**10.5.0.1000**，

**10.5.0.2000, 10.5.0.3000, 10.6.0.1000, 10.6.0.1100, 10.6.0.2000, 10.6.0.2400, 10.6.0.2504, 10.6.0.2600, 10.7.0.1000, 10.7.0.1050, 10.7.0.1600, 10.7.0.1660, 10.7.0.2000, 10.7.0.2101, 10.7.0.2200, 10.7.0.2300, 10.7.0.2302, 10.7.0.2306, 10.7.0.2310, 10.7.0.2400, 10.7.0.2500, 10.7.0.2600, 10.7.0.2602, 10.7.0.2700, 10.7.0.2704 支持升级到 10.7.0.2800 版本。**

不支持从其它低版本迭代升级上来的版本，再升级到此版本。例如：不支持从其它低版本升级到 **10.3.0.4200** 版本后，再升级到 **10.7.0.2800** 版本。

2. 其它低版本或升级上来的版本，按

<https://kb.qianxin.com/detail/505f176539f> 邮件申请规范，由三线专家远程操作支持升级；

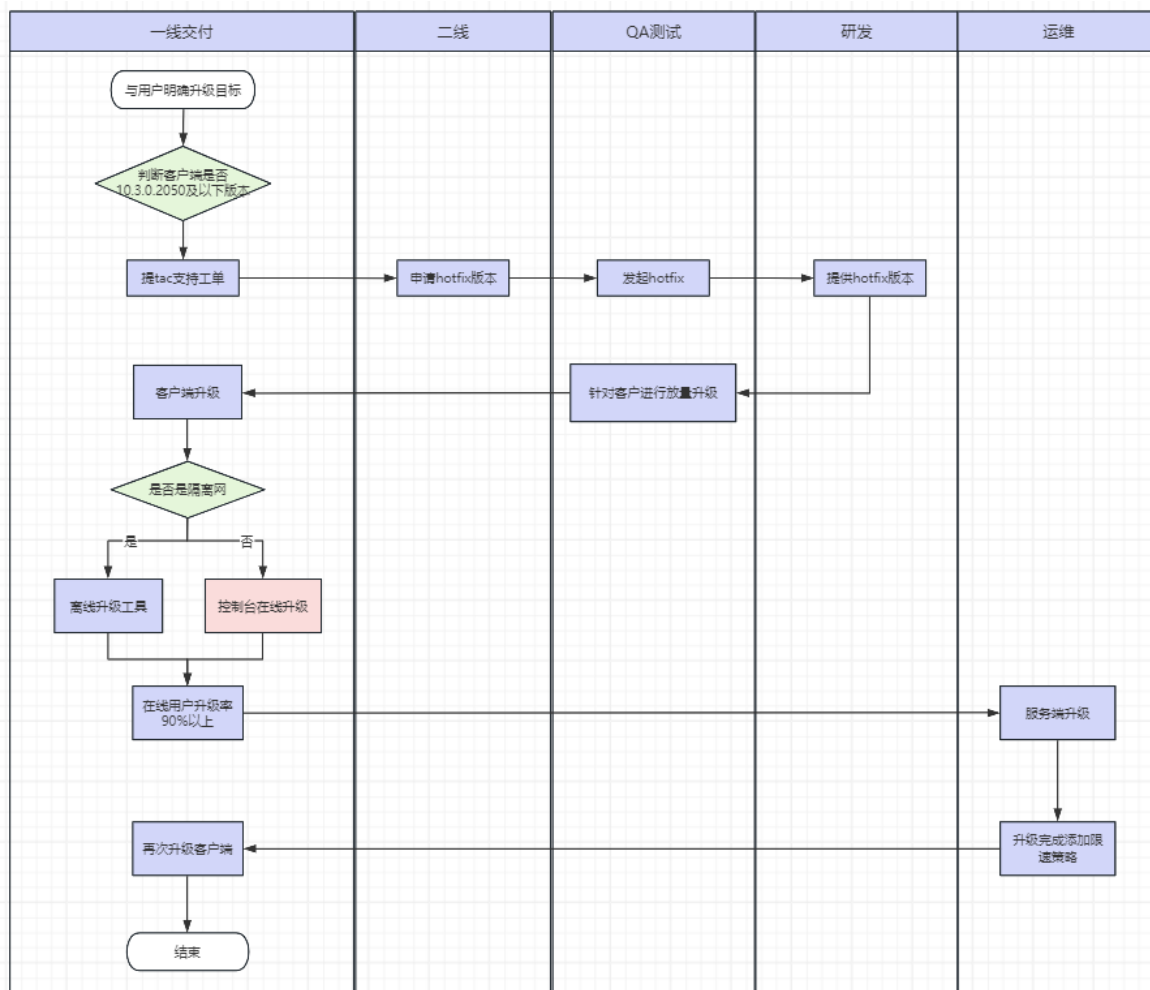
3. 此升级操作手册，适用于 3/4/6/9/12 台机器标准部署的环境；

4. 与客户明确具体升级日期和升级时间；

5. 从官网下载准备好 10.7.0.2800 版本安装包（集群版安装包较大，20GB 左右），将 10.7.0.2800 版本安装包上传到原来部署机（应用服务器的第一台）  
/qaxdata/cli/10.7.0.2800 目录下。

6. 如果星海版本小于 XH6.8.0, 并且没有配置 `t1l_only_drop_parts`、`materialize_t1l_after_modify` 参数的请提 TAC 寻求二线帮助，根据 wiki (XH6.8.0 以前给表配置数据清理策略) 做 ck 优化操作。

7. 升级前需检测客户端版本，如果客户端版本为 10.3.0.2050 及以下版本，需要先按下图流程，申请终端 hotfix 版本，将终端升级到指定 hotfix 版本后，才可进行服务端升级流程，参考 [KB62463](#)。



### 3 环境检查

#### 1. 确认具体备份程序执行主机信息

##### (1) 可用磁盘空间

a、df -Th 命令查看/qaxdata 目录所挂载的设备还剩多少空间，update.sh -d 参数指定备份目录位置，默认为/qaxdata 查看主要存放数据目录 qaxdata/s/services/mongo/data 自行对比有没有备份空间

b、/qaxdata 目录为 docker 数据目录，/qaxdata 目录已用磁盘空间加 70G 不要超 80%，否则会触发 docker gc 镜像机制

#### 2. 集群升级前，在天擎管理中心 > 更新管理 > 关闭终端版本的自动更新

3. 备份现有 deploy 信息，任意应用服务器执行 kubectl get deployment --no-headers > deploy\_当前时间.log

4. 在上个版本的 tq\_saas 目录（如：/qaxdata/cli/10.7.0.2500/tq\_saas）执行 darwin\_client 进行 paas 层检测

```
./darwin-client install --data.source=local --execute.type=k8s --  
template.file={template 文件} --actions=paas_check
```

```
例：./darwin-client install --data.source=local --  
execute.type=k8s --  
template.file=/qaxdata/cli/tq_paas/skylar/skylar_linuxserver_client/plan_c  
lient/darwin_template_2627e61b0c0004c2.yaml --actions=paas_check
```

注：管理中心升级过的环境是上次升级包中的 darwin\_update\_template.yaml（取最新日期的那个），未升级过的场景就是安装包路径下的 darwin\_template.yaml 文件。

如果检测中有 es、flink、sabre 有失败，则判断是否使用 edr 模块  
成功示例：

```
2023-09-20 18:15:15.124565 [INFO] <control.go:341> msg:===== 正在检查paas层基础服务 =====
2023-09-20 18:15:15.150264 [INFO] <paas_check.go:158> msg:-----> bn 校验通过
2023-09-20 18:15:15.156547 [INFO] <paas_check.go:158> msg:-----> redis 校验通过
2023-09-20 18:15:15.157492 [INFO] <paas_check.go:79> msg:-----> cdn 跳过校验
2023-09-20 18:15:15.157530 [INFO] <paas_check.go:79> msg:-----> edge 跳过校验
2023-09-20 18:15:15.157599 [INFO] <paas_check.go:79> msg:-----> cdn 跳过校验
add log to test
2023-09-20 18:15:15.158620 [INFO] <paas_check.go:158> msg:-----> minio 校验通过
2023-09-20 18:15:15.160024 [INFO] <paas_check.go:158> msg:-----> sxt_webrtc 校验通过
2023-09-20 18:15:15.161038 [INFO] <paas_check.go:79> msg:-----> web 跳过校验
2023-09-20 18:15:15.391531 [INFO] <paas_check.go:158> msg:-----> ch 校验通过
2023-09-20 18:15:15.406721 [INFO] <paas_check.go:158> msg:-----> mongo 校验通过
2023-09-20 18:15:15.407967 [INFO] <paas_check.go:79> msg:-----> pg 跳过校验
2023-09-20 18:15:15.408014 [INFO] <paas_check.go:79> msg:-----> pg 跳过校验
2023-09-20 18:15:15.427066 [INFO] <paas_check.go:158> msg:-----> kafka 校验通过
2023-09-20 18:15:15.481853 [INFO] <paas_check.go:158> msg:-----> noah 校验通过
2023-09-20 18:15:15.483229 [INFO] <paas_check.go:158> msg:-----> sabre 校验通过
2023-09-20 18:15:15.725750 [INFO] <paas_check.go:158> msg:-----> rh 校验通过
2023-09-20 18:15:17.054464 [INFO] <paas_check.go:158> msg:-----> es 校验通过
2023-09-20 18:15:17.055755 [INFO] <paas_check.go:79> msg:-----> bl 跳过校验
2023-09-20 18:15:17.055791 [INFO] <paas_check.go:79> msg:-----> caesar 跳过校验
2023-09-20 18:15:17.055822 [INFO] <paas_check.go:79> msg:-----> env 跳过校验
2023-09-20 18:15:17.061778 [INFO] <paas_check.go:158> msg:-----> etcd 校验通过
2023-09-20 18:15:17.063137 [INFO] <paas_check.go:79> msg:-----> registry 跳过校验
2023-09-20 18:15:17.063179 [INFO] <paas_check.go:79> msg:-----> registry 跳过校验
2023-09-20 18:15:17.063207 [INFO] <paas_check.go:79> msg:-----> registry 跳过校验
2023-09-20 18:15:17.063240 [INFO] <paas_check.go:79> msg:-----> registry 跳过校验
2023-09-20 18:15:17.063268 [INFO] <paas_check.go:79> msg:-----> registry 跳过校验
2023-09-20 18:15:17.063295 [INFO] <paas_check.go:79> msg:-----> registry 跳过校验
2023-09-20 18:15:17.063713 [INFO] <paas_check.go:158> msg:-----> flinkonk8s 校验通过
2023-09-20 18:15:17.069497 [INFO] <paas_check.go:158> msg:-----> etcd 校验通过
2023-09-20 18:15:17.074082 [INFO] <paas_check.go:158> msg:-----> redis 校验通过
2023-09-20 18:15:17.076671 I | structs.go:22: Connected to 10.58.128.118:2181
2023-09-20 18:15:17.079656 I | structs.go:22: authenticated: id=216178685606756373, timeout=100000
2023-09-20 18:15:17.079691 I | structs.go:22: re-submitting '0' credentials after reconnect
2023-09-20 18:15:17.082432 I | structs.go:22: recv loop terminated: err=E0F
2023-09-20 18:15:17.082455 I | structs.go:22: send loop terminated: err=<nil>
2023-09-20 18:15:17.082563 [INFO] <paas_check.go:158> msg:-----> zookeeper 校验通过
2023-09-20 18:15:17.084041 [INFO] <paas_check.go:158> msg:-----> tcp 校验通过
2023-09-20 18:15:17.097809 [INFO] <paas_check.go:158> msg:-----> schema-registry 校验通过
2023-09-20 18:15:17.099952 [INFO] <paas_check.go:79> msg:-----> darwin 跳过校验
2023-09-20 18:15:17.110862 [INFO] <paas_check.go:158> msg:-----> k8s 校验通过
2023-09-20 18:15:17.111974 [INFO] <paas_check.go:79> msg:-----> gossip-v2 跳过校验
2023-09-20 18:15:17.112026 [INFO] <control.go:362> msg:===== paas层基础服务已检查通过 =====
```

失败示例:

```
2023-09-20 18:35:10.911431 [INFO] <control.go:145> msg:===== 4ef06e7e0a00877开始安装=====
2023-09-20 18:35:10.914218 [INFO] <control.go:341> msg:===== 正在检查paas层基础服务 =====
2023-09-20 18:35:10.955878 [INFO] <paas_check.go:158> msg:-----> etcd 校验通过
2023-09-20 18:35:10.963873 [INFO] <paas_check.go:158> msg:-----> redis 校验通过
2023-09-20 18:35:10.968056 I | structs.go:22: Connected to 10.58.128.117:2181
2023-09-20 18:35:10.971559 I | structs.go:22: authenticated: id=144121091751870492, timeout=100000
2023-09-20 18:35:10.971595 I | structs.go:22: re-submitting '0' credentials after reconnect
2023-09-20 18:35:10.974969 I | structs.go:22: recv loop terminated: err=E0F
2023-09-20 18:35:10.979311 I | structs.go:22: send loop terminated: err=<nil>
2023-09-20 18:35:10.975133 [INFO] <paas_check.go:158> msg:-----> zookeeper 校验通过
2023-09-20 18:35:11.048784 [INFO] <paas_check.go:158> msg:-----> noah 校验通过
2023-09-20 18:35:11.233375 [INFO] <paas_check.go:158> msg:-----> ch 校验通过
2023-09-20 18:35:11.235024 [WARN] <paas_check.go:161> msg:-----> es校验, 第1次重试失败: dial tcp 10.58.128.116:9211: connect: connection refused
2023-09-20 18:35:21.235970 [WARN] <paas_check.go:161> msg:-----> es校验, 第2次重试失败: dial tcp 10.58.128.116:9211: connect: connection refused
2023-09-20 18:35:31.237472 [WARN] <paas_check.go:161> msg:-----> es校验, 第3次重试失败: dial tcp 10.58.128.116:9211: connect: connection refused
2023-09-20 18:35:41.238879 [WARN] <paas_check.go:161> msg:-----> es校验, 第4次重试失败: dial tcp 10.58.128.116:9211: connect: connection refused
2023-09-20 18:35:51.240287 [WARN] <paas_check.go:161> msg:-----> es校验, 第5次重试失败: dial tcp 10.58.128.116:9211: connect: connection refused
2023-09-20 18:35:51.241541 [INFO] <paas_check.go:158> msg:-----> sxt_webrtc 校验通过
2023-09-20 18:35:51.242567 [INFO] <paas_check.go:79> msg:-----> web 跳过校验
2023-09-20 18:35:51.242660 [INFO] <paas_check.go:79> msg:-----> darwin 跳过校验
2023-09-20 18:35:51.242694 [INFO] <paas_check.go:79> msg:-----> registry 跳过校验
2023-09-20 18:35:51.242741 [INFO] <paas_check.go:79> msg:-----> registry 跳过校验
2023-09-20 18:35:51.242792 [INFO] <paas_check.go:79> msg:-----> registry 跳过校验
2023-09-20 18:35:51.242842 [INFO] <paas_check.go:79> msg:-----> registry 跳过校验
2023-09-20 18:35:51.242910 [INFO] <paas_check.go:79> msg:-----> registry 跳过校验
2023-09-20 18:35:51.242958 [INFO] <paas_check.go:79> msg:-----> registry 跳过校验
2023-09-20 18:35:51.243502 [INFO] <paas_check.go:158> msg:-----> tcp 校验通过
2023-09-20 18:35:51.251315 [INFO] <paas_check.go:158> msg:-----> redis 校验通过
2023-09-20 18:35:51.252203 [INFO] <paas_check.go:79> msg:-----> edge 跳过校验
2023-09-20 18:35:51.252255 [INFO] <paas_check.go:79> msg:-----> env 跳过校验
2023-09-20 18:35:51.252529 [INFO] <paas_check.go:158> msg:-----> etcd 校验通过
2023-09-20 18:35:51.258111 [WARN] <paas_check.go:161> msg:-----> flinkonk8s校验, 第1次重试失败: dial tcp 10.58.128.119:31111: connect: connection refused
2023-09-20 18:36:01.258461 [WARN] <paas_check.go:161> msg:-----> flinkonk8s校验, 第2次重试失败: dial tcp 10.58.128.119:31111: connect: connection refused
2023-09-20 18:36:11.259796 [WARN] <paas_check.go:161> msg:-----> flinkonk8s校验, 第3次重试失败: dial tcp 10.58.128.119:31111: connect: connection refused
2023-09-20 18:36:21.260273 [WARN] <paas_check.go:161> msg:-----> flinkonk8s校验, 第4次重试失败: dial tcp 10.58.128.119:31111: connect: connection refused
2023-09-20 18:36:31.261481 [WARN] <paas_check.go:161> msg:-----> flinkonk8s校验, 第5次重试失败: dial tcp 10.58.128.119:31111: connect: connection refused
2023-09-20 18:36:31.275749 [INFO] <paas_check.go:158> msg:-----> k8s 校验通过
2023-09-20 18:36:31.277123 [INFO] <paas_check.go:79> msg:-----> cdn 跳过校验
add log to test
2023-09-20 18:36:31.278696 [INFO] <paas_check.go:158> msg:-----> minio 校验通过
```

```
2023-09-20 18:36:31.277123 [INFO] <paas_check.go:79> msg:-----> cdn 跳过校验
add log to test
2023-09-20 18:36:31.278696 [INFO] <paas_check.go:158> msg:-----> minio 校验通过
2023-09-20 18:36:31.280160 [INFO] <paas_check.go:79> msg:-----> bi 跳过校验
2023-09-20 18:36:31.280220 [INFO] <paas_check.go:79> msg:-----> cdn 跳过校验
2023-09-20 18:36:31.516168 [INFO] <paas_check.go:158> msg:-----> ch 校验通过
2023-09-20 18:36:31.526507 [INFO] <paas_check.go:158> msg:-----> mongo 校验通过
2023-09-20 18:36:31.527544 [INFO] <paas_check.go:79> msg:-----> pg 跳过校验
2023-09-20 18:36:31.527581 [INFO] <paas_check.go:79> msg:-----> pg 跳过校验
2023-09-20 18:36:31.543193 [INFO] <paas_check.go:158> msg:-----> kafka 校验通过
2023-09-20 18:36:31.544219 [INFO] <paas_check.go:79> msg:-----> gossin v2 跳过校验
2023-09-20 18:36:31.544455 [WARN] <paas_check.go:161> msg:-----> sabre校验, 第1次重试失败: dial tcp 10.58.128.119:31100: connect: connection refused
2023-09-20 18:36:41.545015 [WARN] <paas_check.go:161> msg:-----> sabre校验, 第2次重试失败: dial tcp 10.58.128.119:31100: connect: connection refused
2023-09-20 18:36:51.546324 [WARN] <paas_check.go:161> msg:-----> sabre校验, 第3次重试失败: dial tcp 10.58.128.119:31100: connect: connection refused
2023-09-20 18:37:01.547056 [WARN] <paas_check.go:161> msg:-----> sabre校验, 第4次重试失败: dial tcp 10.58.128.119:31100: connect: connection refused
2023-09-20 18:37:11.548037 [WARN] <paas_check.go:161> msg:-----> sabre校验, 第5次重试失败: dial tcp 10.58.128.119:31100: connect: connection refused
2023-09-20 18:37:11.569850 [INFO] <paas_check.go:158> msg:-----> schena-registrv 校验通过
2023-09-20 18:37:11.573171 [INFO] <paas_check.go:158> msg:-----> bn 校验通过
2023-09-20 18:37:11.574208 [INFO] <paas_check.go:79> msg:-----> caesar 跳过校验
2023-09-20 18:37:11.574303 [ERROR] <paas_check.go:103> msg:paas层基础服务检查失败: [1] es: dial tcp 10.58.128.116:9211: connect: connection refused, host: 10.58.128.116:9211, 10.58.128.117:9211, 10.58.128.118:9211, 10.58.128.119:9211. [2] flinkonk8s: dial tcp 10.58.128.119:31111: connect: connection refused, host: http://10.58.128.119:31111. [3] sabre: dial tcp 10.58.128.119:31100: connect: connection refused, host: 10.58.128.119:31100.
2023-09-20 18:37:11.580820 [ERROR] <control.go:529> msg:-----> 4ef06e7e0a008077安装失败=====
2023-09-20 18:37:11.580879 [ERROR] <control.go:530> msg:[1] es: dial tcp 10.58.128.116:9211: connect: connection refused, host: 10.58.128.116:9211, 10.58.128.117:9211, 10.58.128.118:9211, 10.58.128.119:9211. [2] flinkonk8s: dial tcp 10.58.128.119:31111: connect: connection refused, host: http://10.58.128.119:31111. [3] sabre: dial tcp 10.58.128.119:31100: connect: connection refused, host: 10.58.128.119:31100. ;
```

(1) 没使用 edr, 则执行以下命令忽略相关报错

```
sed -i 's/execute\.type=k8s/& --paas.check=false/'
/qaxdata/cli/10.7.0.2800/tq_paas/skyelar/skyelar_linuxserver_client/plan_client/update.sh
```

(2) 使用 edr, 检测并修复出问题的 paas 组件状态 (提 tac)

(3) 如果有 es、flink、sabre 以外的其他 paas 层组件检测未通过, 则检测并修复出问题的 paas 组件状态 (提 tac)

(4) 如果初始版本是 10.5.0.3000 之前的版本, sabre 会报错端口冲突

老版本 和新版本的 namespace 不一样

```
kubectl delete deploy sabre-server -n default-sabre-server
```

```
kubectl delete svc sabre-server-service -n default-sabre-server
```

先删除老版本的 sabre, 不然会导致和新版本的 sabre 端口冲突, deployment 也删除

4. 有条件, 尽量走停服关机**快照备份**。

(1) 如果需要虚拟机打快照进行备份, 请参考 8.5 章节。

(2) 如果不需要虚拟机打快照进行备份, 可以跳过此步骤。

5. 请确保所有服务器的系统日期、时间和时区与当地时间相同。

## 4 具体升级操作

下面命令中的版本号以升级到 **10.7.0.2800** 版本为例，具体以实际为准

### 4.1 上传安装包到指定目录

先创建目录 `mkdir /qaxdata/cli/10.7.0.2800`，再将 10.7.0.2800 版本安装包上传到原来部署机 `/qaxdata/cli/10.7.0.2800` 目录下

### 4.2 运行升级命令

1. `cd /qaxdata/cli/10.7.0.2800&& tar -xvf 2800 安装包`

```
[root@tq-update11-k8s-1 10.7.0.2100]# ls
ansible-rpms  ansible-rpms.tar.gz  install.sh  QI-ANXINTianqing-server-Linux-console_10.7.0.2100_232_61bd3d6aec7c7ee09ef80b5e301793b4.tar.gz  tools  tq_paas  tq_saas
[root@tq-update11-k8s-1 10.7.0.2100]#
```

2. `cd /qaxdata/cli/10.7.0.2800/tq_paas/skylar/skylar_linuxserver_client/plan_client/`

3. `sh update.sh -s 老包目录 -n 新包目录 -t darwinTemplate 路径 -u SSH 用户 -P SSH 密码 -p SSH 端口 -d 升级数据备份目录`

脚本参数说明：

-s 老包目录

-n 新包目录

-t darwinTemplate 路径 该路径为上一次安装/更新成功后生成的 `darwin_template_xxxxxxx.yaml/darwin_update_template_xxxxx.yaml` 文件路径（默认在：

`/qaxdata/cli/tq_paas/skylar/skylar_linuxserver_client/plan_client/`）

-u SSH 用户 （需要 `sudo` 权限或 `root` 用户）

-P SSH 密码

-p SSH 端口 （支持非默认端口）

-d 升级数据备份目录，不指定默认为 `/qaxdata` 为一级目录储存，`-d false` 为临时跳过数据备份

以下为命令示例：（执行此命令需要部署时的账号和密码）

**注：10.6.0.2600 是当前版本安装包所在目录，具体以实际为准**

```
sh update.sh -s /qaxdata/cli/10.6.0.2600 -n /qaxdata/cli/10.7.0.2800 -t
/qaxdata/cli/tq_paas/skylar/skylar_linuxserver_client/plan_client/darwin_template_26
285abe2200902d.yaml -u root -P 'password' -p 22
```

## 4.3 升级成功

升级过程大约需要执行 60 分钟，升级成功后如下图所示：

```
guser:DitBjLmHyCVnZ1VP_ZAa1@10.49.174.19:25432/da_skylar_zeus?alias=zeus&create_table=true;pg://pguser:DitBjLmHyCVnZ1VP_ZAa1@10.49.174.19:25432/edge?alias=edge&create_table=true;pg://postgres:DFa5hby2Quo1UQTP_ZAa1@10.49.174.19:25432/da_skylar_bi?alias=bi;pg://postgres:DFa5hby2Quo1UQTP_ZAa1@10.49.174.19:25432/da_skylar_noah?alias=noah

2022-11-24 21:35:07.072107 [INFO] <merge.go:765> msg:-->配置备份: /opt/cli/10.3.0.4200/tq_saas/_exec/event_4dbb88e7b200394d/onfig.after_back

2022-11-24 21:35:07.072219 [INFO] <control.go:526> msg:=====4dbb88e7b200394d已安装成功=====
```

【注】图片仅为示例，其中安装包名称以实际为准。

升级成功后需访问管理中心验证功能是否正常。

## 5 服务功能验证

1. 登录天擎管理中心，更新管理。调整天擎客户端的更新逻辑，针对部分可以进行功能验证的终端进行恢复终端版本更新验证。

2. 回滚计划安排

由于考虑回滚设施内容较多，有数据覆盖的操作。确认最终要进行版本回滚回退操作，联系产线支持处理。

## 6 应急预案

### 6.1 升级后，服务组件资源占用异常及接口占用异常

#### 1. 服务配置开关:

```
-control.type=config
```

#### 2. 全局限速配置:

```
etcdctl put "da.skylar.rate_limit.enable" "true"
```

```
etcdctl put "da.skylar.rate_limit.max_request_per_unit" "500"
```

```
#日志关键字 "rate limiter" - INFO
```

#### 3. 单服务限速:

(1) 服务启动参数添加: `-control.type="config"`

(2) etcd 添加限速策略:

```
etcdctl put <config_ns>.<project_name>.rate_limit.max_request_per_unit  
<limit_num>
```

```
<config_ns>.<project_name> = kubectl get deploy styx-file-assessment-go -  
o yaml |grep config.ns = da.skylar.project.zion.styx.file_assessment
```

#### 4. edge2 连接数限制:

```
-edge2_max_clientnum=1000
```

ps:

(1) 限制连接数后,可以通过 Edge 监控查看"Edge2 TCP connections"确认当前连接数状态,超过限制将握手失败,从服务表现来看就是到 edge 的 30080(6781)端口不通

(2) 需要结合 Edge 监控辅助查看相关信息,图表最下 > 单接口 RPC 请求速率

(3) 确认 mongo 慢日志查询状态: `grep protocol:op_msg mongod.log | awk '{print $1 ,'\t', $NF}'`

### 6.2 升级后，客户端升级后带宽占满

#### 1. 服务配置项目:

```
project.cdn.max_request (默认 3000,cdn 单容器的最大连接数)
```

project.cdn.disk\_space (默认 20G,容器缓存池大小)

## 2. darwin-cdn 缓存落地

(1)每台应用创建缓存落地目录: `mkdir /qaxdata/s/services/data/darwin-cdn-data -p`

(2)添加 hostPath 缓存数据持久化-任意一台应用执行即可: `kubectl patch deployment darwin-cdn -p='{ "spec": { "template": { "spec": { "volumes": [{ "name": "darwin-cdn-data", "hostPath": { "path": "/qaxdata/s/services/data/darwin-cdn-data", "type": "" } }, "containers": [{ "name": "darwin-cdn", "volumeMounts": [{ "name": "darwin-cdn-data", "mountPath": "/home/s/darwin-cdn/data/" } ] } ] } } } }`

结果返回: deployment.apps/darwin-cdn patched, 即成功

## 3. 添加反亲和规则:

任意应用服务器执行: `kubectl patch deployment darwin-cdn -p='{ "spec": { "template": { "spec": { "affinity": { "podAntiAffinity": { "requiredDuringSchedulingIgnoredDuringExecution": [ { "labelSelector": { "matchExpressions": [ { "key": "app", "operator": "In", "values": [ "$deployment_name" ] } ] }, "topologyKey": "kubernetes.io/hostname" } ] } } } } }`

结果返回: deployment.apps/darwin-cdn patched, 即成功

## 7 接入点升级

### 7.1 接入点 10.3.0.3201 之前版本升级

10.3.0.3201 之前的版本为 docker 部署，不能直接升级到 10.3.0.3201 及以后的版本，需要将有 docker 部署的接入点卸载后，按部署文档重新部署。接入点卸载方法：

1. `docker rm -f $(docker ps -aq)`
2. `rm -rf /data01/s/services`
3. `systemctl restart docker`

注意 /data01/s/services 为接入点的安装目录，由于前期版本的不规范，这个目录可能并不是 /data01/s/services 需要自行核对。

### 7.2 接入点 10.3.0.3201 及以后版本的升级

下载对应的接入点放入 /qaxdata/cli/x.x.x.x x.x.x.x 为对版本号，并解压安装包，下面以升级 10.6.0.2401 为例

```
tar -xvf QI-ANXINTianqing-node-  
Linux_10.6.0.2401_4_23d49f4bd28ed98c52b2c9305cc4c240.tar.gz
```

```
[root@install01v 10.6.0.2401]# pwd  
/opt/cli/10.6.0.2401  
[root@install01v 10.6.0.2401]# ls -lh  
total 25G  
-rw-rw-r-- 1 zhengchaoping zhengchaoping 25G Jun 16 21:34 QI-ANXINTianqing-node-Linux_10.6.0.2401_4_23d49f4bd28ed98c52b2c9305cc4c240.tar.gz  
[root@install01v 10.6.0.2401]# tar -xvf QI-ANXINTianqing-node-Linux_10.6.0.2401_4_23d49f4bd28ed98c52b2c9305cc4c240.tar.gz
```

```
cd tq_paas/skylar/skylar_linuxserver_client/plan_client/
```

```
cp /qaxdata/cli/tq_paas/skylar/skylar_linuxserver_client/plan_client/deploy-  
server.db ./
```

```
cp  
/qaxdata/cli/tq_paas/skylar/skylar_linuxserver_client/plan_client/third_party/ansible_  
inventory/* third_party/ansible_inventory/
```

```
[root@install01v 10.6.0.2401]# ls -lh
total 25G
-rw-r--r-- 1 root root 5.1K Jun 16 21:30 install.sh
-rw-rw-r-- 1 zhengchaoping zhengchaoping 25G Jun 16 21:34 01-ANXINTianqing-node-Linux_10.6.0.2401_4_23d49f4bd28ed98c52b2c9305cc4c240.tar.gz
drwxr-xr-x 5 root root 4.0K Jun 16 21:23 tools
drwxr-xr-x 3 root root 4.0K Jun 16 21:28 tq_paas
drwxr-xr-x 2 root root 4.0K Jun 27 20:56 tq_saas
[root@install01v 10.6.0.2401]# cd tq_paas/skylar/skylar_linuxserver_client/plan_client/
[root@install01v plan_client]# ls -lh
total 90M
-rw-r--r-- 1 root root 38M Nov 12 2021 ansible-rpms.tar.gz
-rw-r--r-- 1 root root 17M Nov 12 2021 deploy-id
-rw-r--r-- 1 root root 5.1K Feb 2 11:59 install.sh
-rw-r--r-- 1 root root 36M Jun 16 16:24 plan_client
-rw-r--r-- 1 root root 637 Apr 2 2022 plan.test.yaml
-rw-r--r-- 1 root root 678 Dec 14 2022 plan.yaml.template
drwxr-xr-x 5 root root 4.0K Jun 16 21:23 preCheck
-rw-r--r-- 1 root root 15K May 19 13:41 template.yaml
drwxr-xr-x 5 root root 4.0K Jun 16 21:28 third_party
drwxr-xr-x 5 root root 4.0K Jun 16 21:23 web
[root@install01v plan_client]# cp /opt/cli/tq_paas/skylar/skylar_linuxserver_client/plan_client/deploy-server.db ./
[root@install01v plan_client]# cp /opt/cli/tq_paas/skylar/skylar_linuxserver_client/plan_client/third_party/ansible_inventory/* third_party/ansible_inventory/
cp: overwrite 'third_party/ansible_inventory/README'? y
[root@install01v plan_client]#
```

```
cp /qaxdata/cli/10.6.0.2401/tq_saas/template.yaml.enc ./
```

```
cp ../darwin-client/darwin-client ./
```

```
chmod u+x *client
```

```
[root@install01v plan_client]# cp /opt/cli/tq_paas/skylar/skylar_linuxserver_client/plan_client/deploy-server.db ./
[root@install01v plan_client]# cp ../darwin-client/darwin-client ./
[root@install01v plan_client]# chmod u+x *client
[root@install01v plan_client]# ls -lh
total 198M
-rw-r--r-- 1 root root 38M Nov 12 2021 ansible-rpms.tar.gz
-rwxr--r-- 1 root root 108M Jun 27 21:01 darwin-client
-rw-r--r-- 1 root root 17M Nov 12 2021 deploy-id
-rw-r--r-- 1 root root 320K Jun 27 20:58 deploy-server.db
-rw-r--r-- 1 root root 5.1K Feb 2 11:59 install.sh
-rwxr--r-- 1 root root 36M Jun 16 16:24 plan_client
-rw-r--r-- 1 root root 637 Apr 2 2022 plan.test.yaml
-rw-r--r-- 1 root root 678 Dec 14 2022 plan.yaml.template
drwxr-xr-x 5 root root 4.0K Jun 16 21:23 preCheck
-rw-r--r-- 1 root root 15K May 19 13:41 template.yaml
-rw-r--r-- 1 root root 18K Jun 27 21:01 template.yaml.enc
drwxr-xr-x 5 root root 4.0K Jun 16 21:28 third_party
drwxr-xr-x 5 root root 4.0K Jun 16 21:23 web
[root@install01v plan_client]#
```

拷贝原来部署生成的 darwin\_template\_depoyID.yaml 来当前目录，注意这个 deployID 是申请许可证的 ID

```
cp /qaxdata/cli/tq_saas/darwin_template_267ae1aa3b0073b5.yaml ./
```

注意：如果/qaxdata/cli/tq\_saas/目录下没有 darwin\_template\_depoyID.yaml 文件，在/qaxdata/cli/tq\_paas/skylar/skylar\_linuxserver\_client/plan\_client/目录下查找

```
./darwin-client decrypt --file=template.yaml.enc
```

```
./plan_client update template --
```

src=darwin\_template\_267ae1aa3b0073b5.yaml --new=template.yaml --  
pass\_install\_registry

```
[root@install01v plan_client]# cp /opt/cli/tq_saa/darwin_template_267ae1aa3b0073b5.yaml ./
[root@install01v plan_client]# ./darwin-client decrypt --file=template.yaml.enc
WARNING: Package "github.com/golang/protobuf/protoc-gen-go/generator" is deprecated.
A future release of golang/protobuf will delete this package,
which has long been excluded from the compatibility promise.

2023-06-27 21:05:06.297631 [INFO] <framework.go:55> msg:Data Access Framework Init.
2023-06-27 21:05:06.432461 I | reg.da.go:13: Register PD[BuildID:bc3183878a62a17e0421465c64932f45] to DA framework for Client
2023-06-27 21:05:06.433440 [INFO] <sqlite3_checkpoint.go:106> msg:registeredDestroy for sqlite3
2023-06-27 21:05:06.433595 [DEBUG] <init.go:28> msg:register plugin(pub-sub) to da framework
2023-06-27 21:05:06.451050 [INFO] <decrypt.go:65> msg:解密文件: template.yaml
[root@install01v plan_client]# ./plan_client update template --src=darwin_template_267ae1aa3b0073b5.yaml --new=template.yaml --pass_install_registry
INFO[0000] 当前架构为amd64,skylar_linuxserver_client
INFO[0000] 更新成功
[root@install01v plan_client]# ls -lh
total 198M
-rw-r--r-- 1 root root 38M Nov 12 2021 ansible-rpms.tar.gz
-rwxr--r-- 1 root root 108M Jun 27 21:01 darwin-client
-rw-r--r-- 1 root root 15K Jun 27 21:04 darwin_template_267ae1aa3b0073b5.yaml
-rw-r--r-- 1 root root 15K Jun 27 21:07 darwin_update_template_20230627210707.yaml
-rw-r--r-- 1 root root 17M Nov 12 2021 deploy-id
-rw-r--r-- 1 root root 344K Jun 27 21:07 deploy-server.db
-rw-r--r-- 1 root root 5.1K Feb 2 11:59 install.sh
-rwxr--r-- 1 root root 36M Jun 16 16:24 plan_client
-rw-r--r-- 1 root root 637 Apr 2 2022 plan.test.yaml
-rw-r--r-- 1 root root 678 Dec 14 2022 plan.yaml.template
drwxr-xr-x 5 root root 4.0K Jun 16 21:23 preCheck
-rw-r--r-- 1 root root 13K Jun 27 21:05 template.yaml
-rw-r--r-- 1 root root 18K Jun 27 21:01 template.yaml.enc
drwxr-xr-x 5 root root 4.0K Jun 16 21:28 third_party
drwxr-xr-x 5 root root 4.0K Jun 16 21:23 web
[root@install01v plan_client]#
```

【注意】在这步，原有接入安装的为 10.7 以下的版本，升级到 10.7 以上的版本中，转换中报下面的错误

```
[root@install01v plan_client]# ./plan_client update template --src=darwin_template_267ae1aa3b0073b5.yaml --new=template.yaml --pass_install_registry
WARN[0000] 抽象类型:s3, 资源类型:minio_primary 在新template资源中没有匹配到, 跳过
INFO[0000] resource type: minio is not found. Next match all resources
ERROR[0000] 抽象类型:s3, 资源类型:minio 在所有旧资源类型中没有匹配到, 直接添加资源并青年空配置, 请检查是否需要安装后对配置进行手动修改"
INFO[0000] 更新成功
[root@install01v plan_client]#
```

则将 template.yaml 中的 s3 替换为  
darwin\_template\_267ae1aa3b0073b5.yaml 中的 s3 内容

```

- edge
- heka-relay
  address: 10.49.174.102:31120
  url: gossip.v2://10.49.174.102:31120?is_global=true
  options:
    is_global: "true"

s3:
  minio:
    tags:
      - darwincdn
    address: https://10.46.177.51:32185
    user: pfnJjCoizCCMQdL2qa
    password: 9Tdt0r08z60q71XwqwaTjaGy07q1_ZAa1
    url: https://pfnJjCoizCCMQdL2qa:9Tdt0r08z60q71XwqwaTjaGy07q1_ZAa1@10.46.177.51:32185?region=def
    options:
      region: default
    tags:
      - darwincdnlocal
    address: http://10.49.174.102:9345
    user: pfnJjCoizCCMQdL2qa
    password: GVRIaF8CfVpARugvHVd0xrHQrkSP_ZAa1
    url: http://pfnJjCoizCCMQdL2qa:GVRIaF8CfVpARugvHVd0xrHQrkSP_ZAa1@10.49.174.102:9345?region=def
    options:
      region: default

web:
  tags:
    - darwincdn
    address: https://10.49.174.102:30082
    url: https://10.49.174.102:30082

template.yaml
217,15

- darwincdn
- edge
- heka-relay
  address: 10.252.177.110:31120
  url: gossip.v2://10.252.177.110:31120?is_global=true
  options:
    is_global: "true"

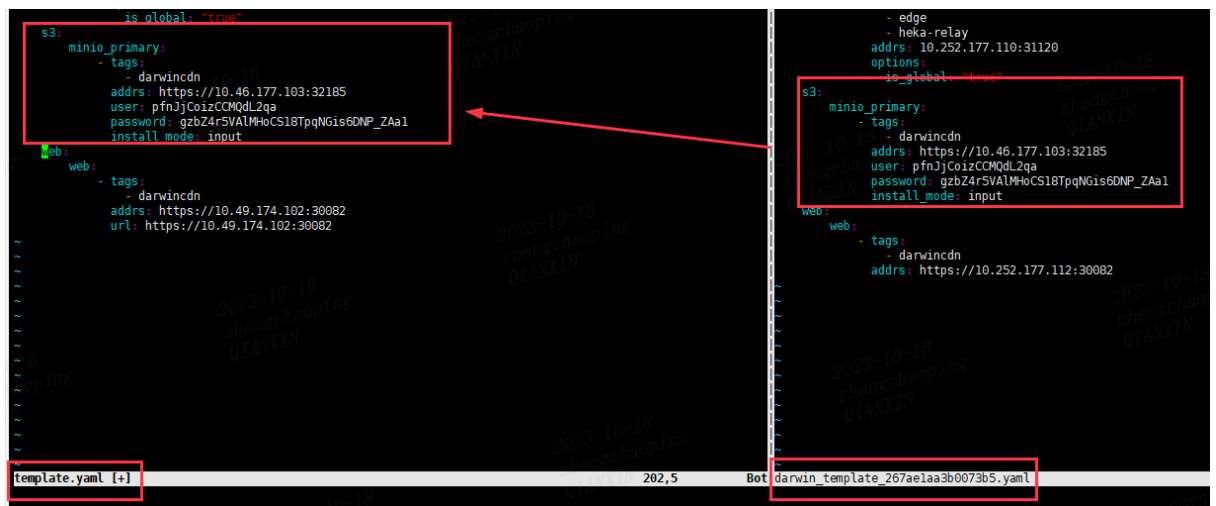
s3:
  minio_primary:
    tags:
      - darwincdn
    address: https://10.46.177.103:32185
    user: pfnJjCoizCCMQdL2qa
    password: gzbZ4r5VALMH0cS18TpqNGis6DNP_ZAa1
    install_mode: input

web:
  tags:
    - darwincdn
    address: https://10.252.177.112:30082

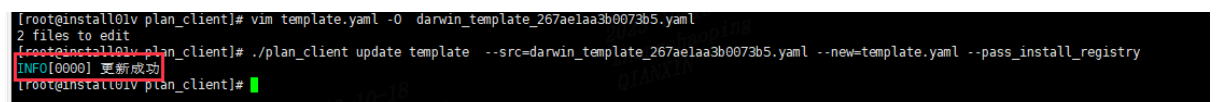
darwin_template_267ae1aa3b0073b5.yaml

```

template.yaml 的内容如下，注意格式。



然后运行：`./plan_client update template --src=darwin_template_267ae1aa3b0073b5.yaml --new=template.yaml --pass_install_registry`

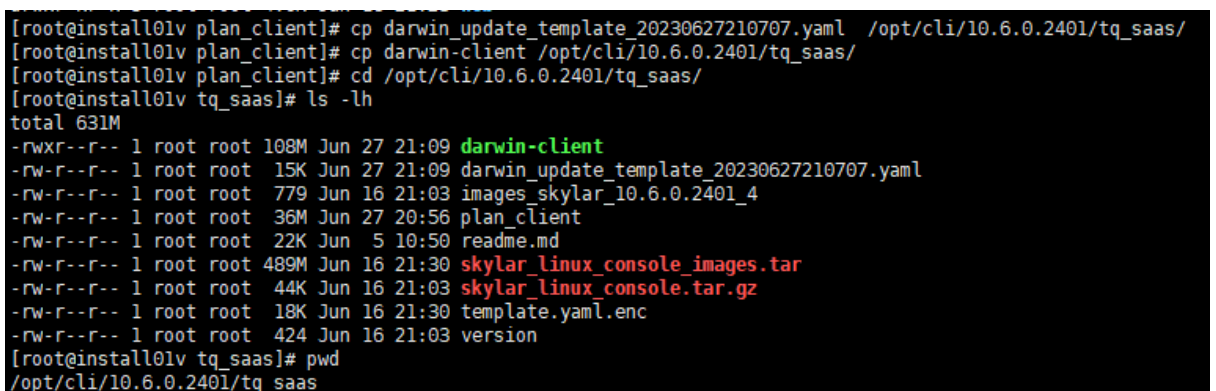


生成 `darwin_update_template.xxx.yaml` 文件，将文件拷贝到 `saas` 目录

`cp darwin_update_template_20230627210707.yaml /qaxdata/cli/10.6.0.2401/tq_saas/`

`cp darwin-client /qaxdata/cli/10.6.0.2401/tq_saas/`

`cd /qaxdata/cli/10.6.0.2401/tq_saas/`



接入点如果是集群，需要将 `skylar_linux_console_images.tar` 拷贝到每一台机器上并将镜像导入本地（每一台都要做这一步）

docker load -i skylar\_linux\_console\_images.tar (单机、集群都执行)

```
[root@install01v tq_saas]# pwd
/opt/cli/10.6.0.2401/tq_saas
[root@install01v tq_saas]# docker load -i skylar_linux_console_images.tar
515d02da915b: Loading layer [=====>] 10.24kB/10.24kB
8893c69b5f3: Loading layer [=====>] 10.24kB/10.24kB
```

【注意】：kubectl get svc 查看 proxy-pp 的服务与 cat darwin\_update\_template\_20230627210707.yaml |grep "proxy-pp"中的文件是否一致，如果 proxy-pp 中无 30083 而 darwin\_update\_template\_20230627210707.yaml 存在，则按下面操作，不是下图所示则不需要操作

```
-rw-r--r-- 1 root root 430 Sep 19 15:51 version
[root@whdev01v tq_saas]# kubectl get svc
NAME                TYPE        CLUSTER-IP      EXTERNAL-IP      PORT(S)                                     AGE
darwin-cdn          NodePort    172.17.3.182    <none>            32185:32185/TCP,32186:32186/TCP            8d
edge2-service       NodePort    172.17.190.22   <none>            6781:30180/TCP,6549:30373/UDP,26781:30181/TCP 8d
gossip-hub          NodePort    172.17.131.204  <none>            9120:31120/TCP                             8d
kubernetes           ClusterIP   172.17.0.1      <none>            443/TCP                                     8d
proxy-pp            NodePort    172.17.100.0    <none>            8080:30080/TCP,8081:30081/TCP             8d
remote-assistance-repeater NodePort    172.17.220.223  <none>            5901:30090/TCP,5500:30091/TCP             8d
trantor-fe          NodePort    172.17.233.175  <none>            8443:30082/TCP,8080:30083/TCP             8d
[root@whdev01v tq_saas]# cat darwin_update_template_20230627210707.yaml |grep "proxy-pp"
proxy-pp: nodeport,Local:edge-grpc,TCP,8080,30080;edge-agent,TCP,8081,30081;web-http,TCP,8082,30083;web-https,TCP,8083,30083 selector=app:proxy-pp
[root@whdev01v tq_saas]#
```

1. 下载 <https://pan.b.gianxin.com/s/rfTZdn4t9NaTntQ> 传到 /qaxdata/cli/10.6.0.2401/tq\_saas/ 目录中，
2. 检查 md5sum fixsvc.sh

```
[root@whdev01v tq_saas]# md5sum fixsvc.sh
181b6866856c777c6fde334755d4af61 fixsvc.sh
[root@whdev01v tq_saas]#
```

3. sh fixsvc.sh

```
[root@whdev01v tq_saas]#
[root@whdev01v tq_saas]# sh fixsvc.sh
daemonset.apps "proxy-pp" deleted
daemonset.apps/proxy-pp replaced
Command fix proxy-pp executed successfully.
service "proxy-pp" deleted
service "trantor-fe" deleted
Command delete svc executed successfully.
[root@whdev01v tq_saas]#
```

【注意】检查 darwin\_update\_template\_20230627210707.yaml 中 web 的 darwincdn 配置，如果不是如下图所示的 proxy-pp:8083，需要修改为 proxy-pp:8083 再进行操作

```
user: pfnJjCoizCCMqdl2qa
password: jnSWLClqXegreptGPiPPVP8ZQPB._ZAa1
url: http://pfnJjCoizCCMqdl2qa:GVRlAf8CfVPaRxgvHVdxx
options:
  region: default

web:
  web:
    - tags:
        - darwincdn
      addr: https://proxy-pp:8083
      url: https://10.49.174.102:30082
```

```
sudo ./darwin-client install --data.source=local --execute.type=k8s --  
template.file=darwin_update_template_20230627210707.yaml
```

```

[roote@install01v tq_saas]# ls -lh
total 631M
-rwxr--r-- 1 root root 108M Jun 27 21:09 darwin-client
-rw-r--r-- 1 root root 15K Jun 27 21:09 darwin_update_template_20230627210707.yaml
-rw-r--r-- 1 root root 779 Jun 16 21:03 images_skytar_10.0.0-2401_4
-rw-r--r-- 1 root root 36M Jun 27 20:56 plan_client
-rw-r--r-- 1 root root 22K Jun 5 10:50 readme.md
-rw-r--r-- 1 root root 489M Jun 16 21:30 skytar_linux_console_images.tar
-rw-r--r-- 1 root root 44K Jun 16 21:03 skytar_linux_console.tar.gz
-rw-r--r-- 1 root root 18K Jun 16 21:30 template.yaml.enc
-rw-r--r-- 1 root root 424 Jun 16 21:03 version
[roote@install01v tq_saas]# ./darwin-client install --data.source=local --execute.type=k8s --template.file=darwin_update_template_20230627210707.yaml

```

无报错，表示升级完成

[illegible]

## 8 FAQ

### 8.1 其它不支持的版本如何升级？

其它不支持的版本，请参考：<https://kb.qianxin.com/detail/505f176539f>

发送邮件申请升级，由产线同事远程操作支持。

### 8.2 如何理解标准安装部署？

标准安装部署，指通过天擎 V10 标准安装部署手册，首次安装成功的环境。不包括由某个低版本升级上来的环境。

### 8.3 如果出现问题是否有日志？

有。安装 saas 日志在/qaxdata/cli/10.7.0.2800/tq\_saas/install\_saas.log 文件中。

### 8.4 部署时使用 IP 后修改为域名集群，升级后 web 无法打开？

请参考：<https://kb.qianxin.com/detail/e1e459631165>

### 8.5 集群版如何停服关机打快照备份？

(1) 服务器停服步骤（二选一）

A：停服方法 1（初始版本 10.6 及以上）

① 执行每台应用机器上的服务关闭脚本，cd  
/qaxdata/s/services/data/install\_end/system-svc-tools && bash  
all\_stop.sh

② 在 BM 管理平台关闭所有服务，集群监控 -> 服务监测 -> 停止所有服务。保证所有服务状态都关闭（BM 页面右上角“控制台”查看进度）。

注：Cluster Client 和 ZooKeeper Client 会自动启动，可忽略。

③ 执行每台数据库和大数据机器上的服务关闭脚本，cd  
/qaxdata/s/services/data/install\_end/system-svc-tools && bash  
all\_stop.sh

B：停服方法 2（无上述关服务脚本）

① 每台应用机器上执行：systemctl stop docker && systemctl stop  
kubelet

② 在 BM 管理平台关闭所有服务，集群监控 -> 服务监测 -> 停止所有服务。保证所有服务状态都关闭（BM 页面右上角“控制台”查看进度）。

注：Cluster Client 和 ZooKeeper Client 会自动启动，可忽略。

③ 每台数据库机器上执行：`systemctl stop mongod27017 && systemctl stop redis16379 && systemctl stop redis16380 && systemctl stop postgres25432`

## （2）停服后的关机流程

集群服务关闭顺序如下：应用服务器 > 数据库服务器 = 大数据服务器

## （3）快照完成后的开机流程

启动顺序：

① 启动数据库服务器 和 大数据服务器

② 在 BM 管理平台启动所有服务，集群监控 -> 服务监测 -> 启动所有服务。保证所有服务状态都正常。

③ 启动应用服务器