# eSDK Huawei Storage Kubernetes CSM Plugins V2.1.0

# 用户指南

**文档版本** 01

发布日期 2024-12-25





#### 版权所有 © 华为技术有限公司 2025。 保留一切权利。

非经本公司书面许可,任何单位和个人不得擅自摘抄、复制本文档内容的部分或全部,并不得以任何形式传播。

#### 商标声明



HUAWE和其他华为商标均为华为技术有限公司的商标。

本文档提及的其他所有商标或注册商标,由各自的所有人拥有。

#### 注意

您购买的产品、服务或特性等应受华为公司商业合同和条款的约束,本文档中描述的全部或部分产品、服务或 特性可能不在您的购买或使用范围之内。除非合同另有约定,华为公司对本文档内容不做任何明示或暗示的声 明或保证。

由于产品版本升级或其他原因,本文档内容会不定期进行更新。除非另有约定,本文档仅作为使用指导,本文档中的所有陈述、信息和建议不构成任何明示或暗示的担保。

# 华为技术有限公司

地址: 深圳市龙岗区坂田华为总部办公楼 邮编: 518129

网址: <a href="https://e.huawei.com">https://e.huawei.com</a>

# 安全声明

#### 产品生命周期政策

华为公司对产品生命周期的规定以"产品生命周期终止政策"为准,该政策的详细内容请参见如下网址: https://support.huawei.com/ecolumnsweb/zh/warranty-policy

#### 漏洞处理流程

华为公司对产品漏洞管理的规定以"漏洞处理流程"为准,该流程的详细内容请参见如下网址:

https://www.huawei.com/cn/psirt/vul-response-process

如企业客户须获取漏洞信息,请参见如下网址:

https://securitybulletin.huawei.com/enterprise/cn/security-advisory

#### 华为初始证书权责说明

华为公司对随设备出厂的初始数字证书,发布了"华为设备初始数字证书权责说明",该说明的详细内容请参见如下网址:

https://support.huawei.com/enterprise/zh/bulletins-service/ENEWS2000015766

#### 华为企业业务最终用户许可协议(EULA)

本最终用户许可协议是最终用户(个人、公司或其他任何实体)与华为公司就华为软件的使用所缔结的协议。最终用户对华为软件的使用受本协议约束,该协议的详细内容请参见如下网址: https://e.huawei.com/cn/about/eula

#### 产品资料生命周期策略

华为公司针对随产品版本发布的售后客户资料(产品资料),发布了"产品资料生命周期策略",该策略的详细内容请参见如下网址:

https://support.huawei.com/enterprise/zh/bulletins-website/ENEWS2000017760

# 前言

# 读者对象

本文档主要适用于以下读者对象:

- 技术支持工程师
- 运维工程师
- 具备存储,Kubernetes和CSI基础知识的工程师

# 符号约定

在本文中可能出现下列标志,它们所代表的含义如下。

符号	说明
▲ 危险	表示如不避免则将会导致死亡或严重伤害的具有高等级风险的危害。
▲ 警告	表示如不避免则可能导致死亡或严重伤害的具有中等级风险的危害。
▲ 注意	表示如不避免则可能导致轻微或中度伤害的具有低等级风险的危害。
须知	用于传递设备或环境安全警示信息。如不避免则可能会导致设备 损坏、数据丢失、设备性能降低或其它不可预知的结果。 "须知"不涉及人身伤害。
□ 说明	对正文中重点信息的补充说明。 "说明"不是安全警示信息,不涉及人身、设备及环境伤害信 息。

# 修改记录

文档版本	发布日期	修改说明
01	2024-12-25	第一次正式发布。

# 目录

<u> </u>	iii
1 产品描述	1
2 兼容性和特性	
2.1 Kubernetes 兼容性	
2.2 华为存储兼容性	
2.3 csm-prometheus 特性说明	
2.4 csm-storage 特性说明	
2.5 约束说明	
3 安装前准备	8
3.1 获取工具、软件包	8
3.2 上传镜像到镜像仓库	g
3.3 安装 Helm	10
3.4 准备配置文件	10
4 安装部署	17
4.1 Helm 安装部署	17
4.1.1 安装软件	17
4.1.2 CCE Agile 安装软件	18
4.1.2.1 制作 Helm 安装包	18
4.1.2.2 安装华为 CSM	18
4.1.3 升级软件	20
4.1.4 回退软件	21
4.1.5 卸载软件	
4.1.6 CCE Agile 卸载软件	
4.2 手动安装部署	
4.2.1 手动安装软件	
4.2.2 手动更新软件	
4.2.3 手动卸载软件	25
5 Prometheus 配置	
5.1 安装 Prometheus	
5.2 配置 Prometheus 服务	
5.3 配置 Prometheus 的仪表盘	32

6 Grafana 配置	33
6.1 安装 Grafana	33
6.2 使用 Grafana	33
7 常用运维指导	39
7.1 日志收集	
7.1.1 <u>前置检查</u>	
7.1.2 使用 oceanctl 收集 CSM 日志	
7.2 查看版本信息	
8 附录	41
8.1 使用非 root 用户访问 Kubernetes	
8.2 csm-prometheus 配置 HTTPS 服务	42
8.2.1 配置 csm-prometheus 的 HTTPS 服务证书	42
8.2.2 删除 csm-prometheus 的 HTTPS 服务证书	43
8.2.3 更新 csm-prometheus 的 HTTPS 服务证书	43
8.3 存储证书管理	44
8.4 权限矩阵	44
9 FAQ	46
9.1 Pod 状态为 OOMKilled	
9.2 手动调整数据抓取请求并发量	
9.3 存储侧 Pod 标签残留	47

# 1 产品描述

CSM(Container Storage Monitor)是Kubernetes容器场景下对华为存储的资源和 Kubernetes中的资源进行可视化展示的一个工具。该工具能够将PV/Pod与LUN/文件系统之间的关系通知存储,从而在存储上完成展示,提供给存储管理员查看;也能将 LUN/文件系统的性能,容量,IOPS等数据上传至三方网管,提供给应用管理员。从而 满足用户在运维场景下的可用性,提升容器场景下运维的可用性短板。

- 2.1 Kubernetes兼容性
- 2.2 华为存储兼容性
- 2.3 csm-prometheus 特性说明
- 2.4 csm-storage特性说明
- 2.5 约束说明

# 2.1 Kubernetes 兼容性

表 2-1 支持的容器管理平台

容器管理平台	版本
Kubernetes	1.16~1.27, 1.31
CCE Agile	22.3.2

# 2.2 华为存储兼容性

#### □说明

- 使用CSM对接存储时,仅支持对CSI发放的LUN/文件系统进行展示,不支持Dtree。
- CSM-Storage标签功能仅OceanStor 6.1.7及以上版本和OceanStor Dorado 6.1.7及以上版本支持。

# 表 2-2 CSM-Storage 兼容性

存储版本	Huawei CSI版本
OceanStor 6.1.7/6.1.8/ V700R001C00	4.6.0
OceanStor Dorado 6.1.7/6.1.8/ V700R001C00	

# 表 2-3 CSM-Prometheus 兼容性

存储版本	Promet heus版 本	Grafa na版 本	Huawei CSI版本
OceanStor 6.1.3/6.1.5/6.1.6/6.1.7/6.1.8/ V700R001C00	2.25.0 - 2.43.0	Grafa na 7.0.4	4.6.0
OceanStor Dorado 6.1.0 <sup>1</sup> /6.1.2/6.1.3/6.1.5/6.1.6/6 .1.7/6.1.8/V700R001C00			

• 注释1 使用OceanStor Dorado 6.1.0版本存储,Prometheus性能监控指标随并发量上升可能会出现断点以及不连续的现象。

# 2.3 csm-prometheus 特性说明

csm-prometheus采集存储监控数据,暴露给Prometheus平台采集,支持的监控指标如下表:

表 2-4 集中式存储监控对象支持的对象指标

对象类型	对象指标
存储	<ul><li>基本信息</li><li>监控状态</li><li>运行状态</li></ul>
控制器	<ul> <li>CPU利用率</li> <li>内存利用率</li> <li>健康状态</li> <li>运行状态</li> </ul>

对象类型	对象指标
存储池	● 总容量
	● 剩余容量
	● 使用容量
	● 容量利用率
LUN	● 总容量
	● 容量利用率
文件系统	● 总容量
	● 容量利用率
PV	<ul><li>◆ 总容量</li></ul>
	● 容量利用率

# 表 2-5 集中式存储监控对象支持的性能指标

对象类型	性能指标
控制器	● 21: 带宽(MB/s)
	● 23: 读带宽(MB/s)
	● 26:写带宽(MB/s)
	• 22: IOPS
	● 25: 读IOPS
	• 28: 写IOPS
	● 370: 平均I/O响应时间(μs)
存储池	● 21: 带宽(MB/s)
	• 22: IOPS
	● 370: 平均I/O响应时间(μs)
LUN	● 21: 带宽(MB/s)
	• 22: IOPS
	● 370: 平均I/O响应时间(μs)
文件系统	• 182: OPS
	● 524: 平均读OPS响应时间(μs)
	● 525: 平均写OPS响应时间(μs)

对象类型	性能指标	
PV	<ul><li>LUN带宽:存储上类型为LUN的PV的带宽(MB/s)</li><li>LUN IOPS:存储上类型为LUN的PV的IOPS</li></ul>	
	<ul> <li>LUN平均I/O响应时间:存储上类型为LUN的PV的平均I/O响应 时间(μs)</li> </ul>	
	• 文件系统OPS:存储上类型为文件系统的PV的OPS	
	<ul><li>文件系统平均读OPS响应时间:存储上类型为文件系统的PV的 平均读OPS响应时间(μs)</li></ul>	
	<ul><li>文件系统平均写OPS响应时间:存储上类型为文件系统的PV的 平均写OPS响应时间(μs)</li></ul>	

# 2.4 csm-storage 特性说明

#### 须知

CSM拓扑服务的内存资源配额默认为512 Mi,拓扑服务本身的内存占用会随集群上PV、Pod的数量线性增加。当集群上资源规模较大时,可手动修改拓扑服务内存资源配额,以保证csm-storage特性的正常使用。具体配置请参考表3-7。

csm-storage会将集群上的PV及与关联的Pod、文件系统、LUN的拓扑关系上报给存储。存储界面的资源展示效果如下:



# 2.5 约束说明

# 性能约束

#### 表 2-6 CSM 使用规格管理

使用规格管理		推荐值
存储管理	接入存储设备数量	5
	单存储设备对接CSM数量	<=3
监控项管理	单个存储支持的最大监控项数量	15000
	所有存储最大监控项总数	40000
抓取间隔管理	对象数据抓取间隔( <mark>表2-4</mark> 中的指标)	300秒
	性能数据抓取间隔(表2-5中的指标)	300秒

#### 须知

- 数据抓取间隔时间不宜配置过小,否则会导致存储后端压力增大,影响存储后端的运行。
- 对象数据抓取间隔和性能数据抓取间隔建议设置为不同的值,防止同时查询时存储 后端压力过大而导致查询失败。
- 单个存储资源(LUN或者文件系统)不支持重复建立拓扑关系。
- 同一台存储上不同租户的同名资源,暂不支持上报拓扑关系。

# 表 2-7 CSM 推荐数据抓取请求并发量

存储文件系统和LUN资源总数量	推荐请求并发量
< 2000	20
2000-5000	10
> 5000	5

# 须知

CSM数据抓取请求并发量默认为20,当存储资源数量增加时,建议按对应推荐请求并发量进行配置,减缓存储查询的压力。配置请求并发量请参考**9.2 手动调整数据抓取请求并发量**。

# **3** 安装前准备

- 3.1 获取工具、软件包
- 3.2 上传镜像到镜像仓库
- 3.3 安装Helm
- 3.4 准备配置文件

# 3.1 获取工具、软件包

# 工具

软件安装、配置和调测过程中,需要准备必要的工具,如表3-1所示。

表 3-1 工具列表

工具名称	工具说明	获取方式
PuTTY	跨平台远程访问工具 用于在软件安装过程中在Windows 系统上访问各节点。	您可以访问chiark主页下载 PuTTY软件。 低版本的PuTTY软件可能导 致登录存储系统失败,建议 使用最新版本的PuTTY软 件。
WinSCP	跨平台文件传输工具,请使用5.7.5或 更高版本,并在传输文件时选用SCP 协议。 用于在Windows系统和Linux系统间 传输文件。	您可以访问WinSCP主页下载 WinSCP软件。

# 软件包

在开始部署服务前,用户需要准备好用于安装的CSM软件安装包,如表3-2所示。

#### 表 3-2 软件包列表

软件包名称	说明	获取地址
eSDK_Huawei_Storage_C SM_V2.1.0_X86_64.zip eSDK_Huawei_Storage_C SM_V2.1.0_ARM_64.zip	CSM软件安装包	https://github.com/ Huawei/csm/releases

# 山 说明

为了防止软件包在传递过程或存储期间被恶意篡改,下载软件包时需下载对应的数字签名文件用于完整性验证。

在软件包下载之后,请参考《OpenPGP签名验证指南》,对从Support网站下载的软件包进行PGP数字签名校验。如果校验失败,请不要使用该软件包,先联系华为技术支持工程师解决。使用软件包安装/升级之前,也需要按上述过程先验证软件包的数字签名,确保软件包未被篡改。

- 运营商客户请访问: https://support.huawei.com/carrier/digitalSignatureAction
- 企业客户请访问: https://support.huawei.com/enterprise/zh/tool/pgp-verify-TL1000000054

# 3.2 上传镜像到镜像仓库

# 前提条件

- 已获取对应的CSM软件包,获取方式请参考3.1 获取工具、软件包。
- 已准备一个镜像仓库,镜像仓库保持与工作节点通信正常,且获取镜像仓库IP地 址以及帐号和密码。
- 已完成在镜像仓库创建项目。
- 已准备一台已安装Docker的Linux主机,且该主机支持访问镜像仓库。

# 操作步骤

**步骤1** 使用远程访问工具(以PuTTY为例),通过管理IP地址,登录安装Docker的Linux主机节点。

步骤2 使用"WinSCP"工具,将软件包上传至"/opt"目录。

步骤3 执行unzip /opt/软件包名称命令,解压软件包。

其中,软件包名称为获取到的CSM软件包。以 eSDK\_Huawei\_Storage\_CSM\_V2.1.0\_X86\_64.zip为例。 # unzip /opt/eSDK\_Huawei\_Storage\_CSM\_V2.1.0\_X86\_64.zip.zip -d /opt/huawei-csm

步骤4 执行docker login < 镜像仓库IP地址>命令,输入帐号和密码,登录镜像仓库。

步骤5 执行以下命令,上传CSM镜像。

chmod +x /opt/huawei-csm/helm/huawei-csm/upload-image.sh; ./opt/huawei-csm/helm/huawei-csm/upload-image.sh --imageRepo <镜像仓库项目名称>

#### ----结束

#### 须知

- upload-image.sh脚本使用Bash(Bourne-Again Shell)解释执行,执行前请确保 当前系统支持Bash类型的Unix shell。
- CCE Agile平台请参考该平台用户手册完成镜像导入和上传。

# 3.3 安装 Helm

#### 山 说明

当前仅支持Helm 3。

Helm是Kubernetes生态系统中的一个软件包管理工具,类似Ubuntu的APT(Advanced Packaging Tool)、CentOS的YUM(Yellowdog Updater, Modified)、或Python的PIP(Package Installer for Python)一样,Helm专门负责管理Kubernetes的应用资源。使用Helm可以对Kubernetes应用进行统一打包、分发、安装、升级以及回退等操作。

- Helm的获取、安装:点此前往。
- Helm的其他信息请参考:点此前往。

# 3.4 准备配置文件

在使用Helm时,需要根据部署时对接的华为存储以及需要使用的特性准备values.yaml文件。

# 操作步骤

**步骤1** 使用远程访问工具(以PuTTY为例),通过管理IP地址,登录Kubernetes集群的任意master节点。

步骤2 使用"WinSCP"工具,将软件包上传至"/opt"目录。

步骤3 执行unzip /opt/软件包名称命令,解压软件包。

其中,软件包名称为获取到的软件包。以 eSDK\_Huawei\_Storage\_CSM\_V2.1.0\_X86\_64.zip为例。 # unzip /opt/eSDK\_Huawei\_Storage\_CSM\_V2.1.0\_X86\_64.zip -d /opt/huawei-csm

步骤4 执行cd /opt/huawei-csm/helm/huawei-csm命令,进入到Helm的工作目录。

步骤5 执行vi values.yaml命令,配置values.yaml中的配置项参数。修改完成后,按Esc,并输入:wq!,保存修改。相关参数说明如表3-3、表3-4、表3-5和表3-6所示。

global配置项主要配置系统需要的全局信息。

表 3-3 global 配置项

参数	描述	必选参数	默认值
replicaCount	CSM部署的 Deployment对 应Pod副本数。 当Pod副本数大于1时,系统自动启用 leaderElection 功能。	否	1,建议不超过2。 Pod副本数为1时:保证Kubernetes中CPU核数不低于4,内存不低于4GB。 Pod副本数为2时:保证Kubernetes中CPU核数不低于8,内存不低于8GB。  勿知 - 若部署时使用的是单副本,后续想要更新部署方式为多副本,请参考4.1.3升级软件更新replicaCount参数。 - 禁止使用其他方式修改CSM部署的Deployment对应Pod副本数。若使用其他方式,例如直接编辑CSM部署的Deployment中的Spec.replicas参数,将会导致多副本功能异常。
imageRepo	镜像仓库名称	是	无,与 <mark>步骤5</mark> 的镜像仓库名称保持一致。
logging.modul e	日志模式	是	file 可选值: file, console
logging.level	日志级别	是	info 可选值:debug,info,warning,error
logging.fileSiz e	日志文件大小	是	20 MB
logging.maxBa ckups	日志最大备份数	是	9
leaderElection. leaseDuration	领导者持续时 间。仅多副本下 生效	否	8s

参数	描述	必选参数	默认值
leaderElection. renewDeadlin e	领导者重新选举 时间。仅多副本 下生效	否	6s
leaderElection. retryPeriod	领导者选举重试 时间。仅多副本 下生效	否	2s
balancedDepl oy	当设置为true 时,安装CSM将 倾向于将不同服 务的Pod调度到 不同节点上。	否	true balancedDeploy调度与nodeSelector调 度冲突,则balancedDeploy的调度不会 生效。

features配置项主要是配置特性开关。

表 3-4 features 配置项

参数	描述	必选参 数	默认值	备注
prometheusColle ctor.enabled	是否开启 Prometheus 收集服务。	是	true	-
prometheusColle ctor.nodePort	Prometheus 的 nodePort。	否	30074	默认值为30074,此端口占 用的是主机端口,如果有冲 突,注意修改。
prometheusColle ctor.csiDriverNa me	配置CSI注册 的驱动名 称。	是	csi.hua wei.co m	<ul> <li>直接使用默认值。</li> <li>对于CCE Agile平台,需要修改该字段,例如: csi.oceanstor.com。</li> </ul>
prometheusColle ctor.prometheus CollectorSSL.ena bled	是否开启 HTTPS服 务,开启后 Prometheus 插件将提供 HTTPS服 务。	是	true	注意开启后证书路径为必填。 证书相关的填写参考:配置 HTTPS服务

参数	描述	必选参 数	默认值	备注
prometheusColle ctor.prometheus CollectorSSL.cert Path	开启后 Prometheus 插件将提供 HTTPS服务 后,HTTPS 的证书路 径。	当 prome theusC ollecto r.prom etheus Collect orSSL. enable d为 true 时,必 选	-	注意该路径需为Helm工作路径下的相对路径。
prometheusColle ctor.prometheus CollectorSSL.key Path	开启后 Prometheus 插件将提供 HTTPS服务 后,HTTPS 的密钥路 径。	当 prome theusC ollecto r.prom etheus Collect orSSL. enable d为 true 时,必	-	注意该路径需为Helm工作路径下的相对路径。
prometheusColle ctor.nodeSelector	csm- prometheus -service的节 点选择器。 配置后csm- prometheus -service仅会 调度到存在 该标签的节 点上。	否	-	节点选择器的详细说明请参考: 将 Pod 分配给节点
storageTopo.ena bled	是否开启存 储拓扑展示 服务。	是	true	仅支持OceanStor 6.1.7/ OceanStor Dorado 6.1.7以 及之后的存储版本。
storageTopo.rtRe tryMaxDelay	topo资源同 步任务的最 大重试延迟	否	5m	建议使用默认配置,最小重 试延迟为5秒

参数	描述	必选参 数	默认值	备注
storageTopo.pvR etryMaxDelay	PV资源创建、删除topo资源任务的最大重试延迟	否	1m	建议使用默认配置,最小重 试延迟为5秒
storageTopo.pod RetryMaxDelay	Pod资源更 新topo资源 标签任务的 最大重试延 迟	否	1m	建议使用默认配置,最小重 试延迟为5秒
storageTopo.resy ncPeriod	刷新topo资 源的时间间 隔	否	15m	建议使用默认配置,最小刷 新间隔为5分钟
storageTopo.nod eSelector	csm- storage- service的节 点选择器。 配置后csm- storage- service仅会 调度到存在 该标签的节 点上。	否	-	节点选择器的详细说明请参考: <b>将 Pod 分配给节点</b>

images配置项主要配置CSM需要的镜像信息。

**表 3-5** image 配置项

参数	描述	必选参 数	默认值
prometheusColle ctor	Prometheu s存储数据 采集插件镜 像。	是	csm-prometheus-collector:2.1.0
topoService	资源拓扑服 务镜像。	是	csm-topo-service:2.1.0
containerMonito rInterface	容器监控接 口镜像。	是	csm-cmi:2.1.0
livenessProbe	探活服务镜 像。	是	csm-liveness-probe:2.1.0

cluster配置项主要关于Kubernetes集群的信息。

表 3-6 cluster 配置项

参数	描述	必选参数	默认值
cluster.name	自定义集群名称	是	kubernetes

containerResoucesSet配置项用于配置各Pod的容器资源配置。

表 3-7 containerResoucesSet 配置项

参数	描述	必选参数	默认值
prometheusService.livenes sProbe.requests.memory	Prometheus Pod中 livenessProbe容器最小内 存资源。	是	128Mi
prometheusService.livenes sProbe.limits.cpu	Prometheus Pod中 livenessProbe容器最大 CPU资源。	是	100m
prometheusService.livenes sProbe.limits.memory	Prometheus Pod中 livenessProbe容器最大内 存资源。	是	128Mi
prometheusService.prome theusCollector.requests.cp u	Prometheus Pod中 prometheusCollector容 器最小CPU资源。	是	50m
prometheusService.prome theusCollector.requests.m emory	Prometheus Pod中 prometheusCollector容 器最小内存资源。	是	128Mi
prometheusService.prome theusCollector.limits.cpu	Prometheus Pod中 prometheusCollector容 器最大CPU资源。	是	300m
prometheusService.prome theusCollector.limits.mem ory	Prometheus Pod中 prometheusCollector容 器最大内存资源。	是	512Mi
prometheusService.cmiCo ntroller.requests.cpu	Prometheus Pod中 cmiController容器最小 CPU资源。	是	50m
prometheusService.cmiCo ntroller.requests.memory	Prometheus Pod中 cmiController容器最小内 存资源。	是	128Mi
prometheusService.cmiCo ntroller.limits.cpu	Prometheus Pod中 cmiController容器最大 CPU资源。	是	300m

参数	描述	必选参数	默认值
prometheusService.cmiCo ntroller.limits.memory	Prometheus Pod中 cmiController容器最大内 存资源。	是	512Mi
storageService.livenessPro be.requests.cpu	storage Pod中 livenessProbe容器最小 CPU资源。	是	10m
storageService.livenessPro be.requests.memory	storage Pod中 livenessProbe容器最小内 存资源。	是	128Mi
storageService.livenessPro be.limits.cpu	storage Pod中 livenessProbe容器最大 CPU资源。	是	100m
storageService.livenessPro be.limits.memory	storage Pod中 livenessProbe容器最大内 存资源。	是	128Mi
storageService.cmiControl ler.requests.cpu	storage Pod中 cmiController容器最小 CPU资源。	是	50m
storageService.cmiControl ler.requests.memory	storage Pod中 cmiController容器最小内 存资源。	是	128Mi
storageService.cmiControl ler.limits.cpu	storage Pod中 cmiController容器最大 CPU资源。	是	300m
storageService.cmiControl ler.limits.memory	storage Pod中 cmiController容器最大内 存资源。	是	512Mi
storageService.topoServic e.requests.cpu	storage Pod中 topoService容器最小CPU 资源。	是	50m
storageService.topoServic e.requests.memory	storage Pod中 topoService容器最小内 存资源。	是	128Mi
storageService.topoServic e.limits.cpu	storage Pod中 topoService容器最大CPU 资源。	是	300m
storageService.topoServic e.limits.memory	storage Pod中 topoService容器最大内 存资源。	是	512Mi

# ----结束

**4** 安装部署

#### 4.1 Helm安装部署

4.2 手动安装部署

# 4.1 Helm 安装部署

# 4.1.1 安装软件

# 前提条件

- master节点已完成Helm 3的安装。
- 已完成values.yaml文件的配置,详情请参考3.4 准备配置文件。

# 操作步骤

**步骤1** 使用远程访问工具(以PuTTY为例),通过管理IP地址,登录Kubernetes集群的任意master节点。

步骤2 执行cd /opt/huawei-csm/helm/huawei-csm命令,进入到Helm的工作目录。

**步骤3** 执行helm install huawei-csm ./ -n huawei-csm --create-namespace命令,安装 CSM服务。

# helm install huawei-csm ./ -n huawei-csm --create-namespace NAME: huawei-csm LAST DEPLOYED: Tue Aug 8 23:11:18 2023 NAMESPACE: huawei-csm STATUS: deployed REVISION: 1 TEST SUITE: None

步骤4 执行kubectl get pod -n huawei-csm命令,检查服务是否启动。

# kubectl get pod -n huawei-csm
NAME READY STATUS RESTARTS AGE
csm-prometheus-service-86c795d68-b5xjg 2/2 Running 0 5s
csm-storage-service-85485fd75f-9wg8m 2/2 Running 0 4s

#### ----结束

# 4.1.2 CCE Agile 安装软件

# 4.1.2.1 制作 Helm 安装包

CCE Agile平台安装华为CSM需要制作Helm安装包,本章节介绍如何制作Helm安装包。

# 前提条件

- 节点服务器已经安装部署了Helm 3。
- 已经准备安装CSM所需的values.yaml文件。具体信息请参考**3.4 准备配置文件**章 节说明。

# 操作步骤

- **步骤1** 使用远程访问工具(以PuTTY为例),通过管理IP地址,登录已部署Helm的任意节点。
- 步骤2 执行cd /opt/huawei-csm/helm/命令,进入到Helm的工作目录。
- **步骤3** 执行helm package huawei-csm/ -d ./命令制作Helm安装包,该命令会将安装包生成到当前路径下。

# helm package huawei-csm/ -d ./ Successfully packaged chart and saved it to: huawei-csm-2.1.0.tgz

----结束

# 4.1.2.2 安装华为 CSM

# 前提条件

华为CSM Helm安装包已制作完成,具体信息请参考4.1.2.1 制作Helm安装包。

# 操作步骤

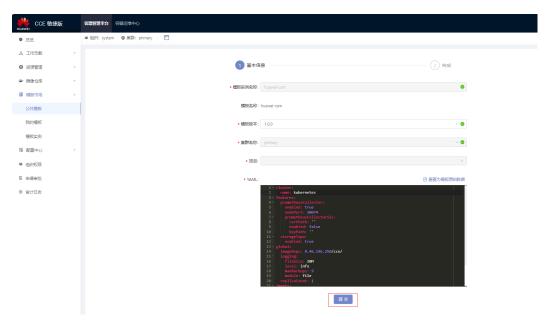
- **步骤1** 使用远程访问工具(以PuTTY为例),通过管理IP地址,登录已部署CCE Agile平台 master的任意节点。
- **步骤2** 执行**kubectl create namespace** *huawei-csm*命令创建部署华为CSM的命名空间,huawei-csm为自定义的命名空间。

# kubectl create namespace huawei-csm

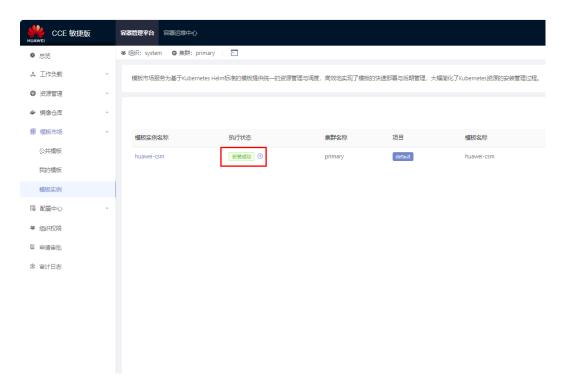
- 步骤3 导出Helm安装包,具体请参考4.1.2.1 制作Helm安装包。
- 步骤4 在主页单击"模板市场> 我的模板>上传模板",进入上传模板对话框。将导出的 Helm安装包导入CCE Agile平台。



**步骤5** 安装包上传完毕,在主页单击"模板市场>我的模板",进入我的模板页面,单击"安装>提交"。其中模板实例名称可自定义填写。



**步骤6** 在主页单击"模板市场>模板实例",选择安装时指定的项目(例如样例中的项目是"default")。安装成功后执行状态将回显为"安装成功"。



----结束

# 4.1.3 升级软件

# 使用场景

当升级CSM服务版本时,请使用本章节进行配置。

# 前提条件

已使用Helm 3完成CSM的部署。

# 注意事项

升级时如果values.yaml文件和update-value.yaml文件含有相同参数的配置,优先使用update-value.yaml内参数。

# 操作步骤

- **步骤1** 使用远程访问工具(以PuTTY为例),通过管理IP地址,登录Kubernetes集群的任意master节点。
- 步骤2 将新的镜像导入相应节点,详情请参考3.2 上传镜像到镜像仓库。
- 步骤3 (可选)执行kubectl delete validatingWebhookConfiguration toposervice.xuanwu.huawei.io命令,删除webhook资源。若toposervice.xuanwu.huawei.io资源不存在,可跳过本步骤。

# kubectl delete validatingWebhookConfiguration topo-service.xuanwu.huawei.io

- 步骤4 执行cd /opt/huawei-csm/helm/huawei-csm/命令,进入到Helm的工作目录。
- 步骤5 执行kubectl apply -f crds/命令, 更新crd资源。

# kubectl apply -f crds/ customresourcedefinition.apiextensions.k8s.io/resourcetopologies.xuanwu.huawei.io configured

**步骤6** 执行helm get values huawei-csm -n huawei-csm -a > update-value.yaml命令, 获取原有服务配置文件。

步骤7 执行vi update-value.yaml命令打开文件,根据更新需要更新参数值。修改完成后,按Esc,并输入:wq!,保存修改。配置详情请参考3.4 准备配置文件。

步骤8 执行helm upgrade huawei-csm ./ -n huawei-csm -f ./values.yaml -f update-value.yaml --wait --timeout 2m命令,升级CSM服务。回显中有Release "huawei-csm" has been upgraded,则表示升级CSM服务成功。

# helm upgrade huawei-csm ./ -n huawei-csm -f ./values.yaml -f update-value.yaml --wait --timeout 2m Release "huawei-csm" has been upgraded. Happy Helming!

NAME: huawei-csm

LAST DEPLOYED: Wed Aug 9 04:19:10 2023

NAMESPACE: huawei-csm

STATUS: deployed

REVISION: 3

TEST SUITE: None

步骤9 (可选)执行kubectl get rt | grep topo- | awk '{print "kubectl delete rt "\$1}' | sh命令,删除旧版本残留的topo资源。若集群上没有旧版本残留的topo资源,可跳过本步骤。

----结束

# 4.1.4 回退软件

# 前提条件

- 已使用Helm 3完成CSM的部署。
- 已使用Helm 3完成CSM的升级。

# 操作步骤

**步骤1** 使用远程访问工具(以PuTTY为例),通过管理IP地址,登录Kubernetes集群的任意master节点。

步骤2 执行cd /opt/huawei-csm/helm/huawei-csm/命令,进入到Helm的工作目录。

**步骤3** 执行helm history huawei-csm -n huawei-csm命令,查看Helm部署服务的历史版本。

# helm history huawei-csm -n huawei-csm
REVISION UPDATED STATUS CHART APP VERSION
DESCRIPTION
1 Tue Aug 8 23:11:18 2023 superseded huawei-csm-2.1.0 1.0.0 Install complete
2 Wed Aug 9 04:19:10 2023 deployed huawei-csm-2.1.0 1.0.0 Upgrade complete

**步骤4** 执行helm rollback huawei-csm revision-number -n huawei-csm --wait -- timeout 2m命令,回退CSM服务到指定版本。回显中有Rollback was a success,则表示回退CSM服务到指定版本成功。

其中,revision-number为<mark>步骤3</mark>查询到的版本号。例如版本为: 1。

# helm rollback huawei-csm 1 -n huawei-csm --wait --timeout 2m Rollback was a success! Happy Helming!

----结束

# 4.1.5 卸载软件

# 前提条件

- 已使用Helm 3完成CSM的部署。
- 使用CSM创建的资源不再需要,且已完成删除。

# 操作步骤

**步骤1** 使用远程访问工具(以PuTTY为例),通过管理IP地址,登录Kubernetes集群的任意master节点。

步骤2 执行cd /opt/huawei-csm/helm/huawei-csm/命令,进入到Helm的工作目录。

**步骤3** 执行**helm uninstall huawei-csm -n huawei-csm**命令,卸载CSM服务。回显中有 "release "huawei-csm" uninstalled ",则表示卸载服务成功。

# helm uninstall huawei-csm -n huawei-csm release "huawei-csm" uninstalled

步骤4 (可选)执行kubectl delete validatingWebhookConfiguration toposervice.xuanwu.huawei.io命令,删除webhook资源。若toposervice.xuanwu.huawei.io资源不存在,可跳过本步骤。

kubectl delete validatingWebhookConfiguration topo-service.xuanwu.huawei.io

步骤5 (可选)执行kubectl delete -f crds/命令,清理crd资源。

#### 须知

- 如果后续不再使用华为CSM,且已经清理华为CSM在环境中的相关资源对象,则执 行该步骤。否则请跳过该步骤。
- 删除crd资源前,确保通过华为CSM创建的资源已经清理干净,包含的crd资源类型可查看/opt/huawei-csm/helm/huawei-csm/crds目录下的文件。
- 删除crd资源会将该crd关联的所有资源进行清理,请慎重执行该操作。
- 如果出现类似的提示: <Error from server (NotFound): error when deleting "crds/xuanwu.huawei.io\_resourcetopologies.yaml": customresourcedefinitions.apiextensions.k8s.io "resourcetopologies.xuanwu.huawei.io" not found>,则说明该crd已经卸载,请忽略该提示。

# kubectl delete -f crds/ customresourcedefinition.apiextensions.k8s.io "resourcetopologies.xuanwu.huawei.io" deleted

步骤6 执行kubectl delete ns huawei-csm命令,删除命名空间。

#### 须知

删除命名空间会将该命名空间的所有资源进行清理,请慎重执行该操作。

#### ----结束

# 4.1.6 CCE Agile 卸载软件

本章节介绍如何在 CCE Agile平台卸载华为CSM,以CCE Agile v22.3.2为例。

# 操作步骤

步骤1 登录CCE Agile平台。

步骤2 在主页单击"模板市场>模板实例",进入模板实例页面。

步骤3 选择华为CSM模板实例,单击"卸载",在弹出的提示框中单击"确定"。



----结束

# 4.2 手动安装部署

# 4.2.1 手动安装软件

# 操作步骤

**步骤1** 使用远程访问工具(以PuTTY为例),通过管理IP地址,登录Kubernetes集群的任意master节点。

**步骤2** 执行**kubectl create namespace huawei-csm**命令,创建huawei-csm命名空间。
# kubectl create namespace huawei-csm
namespace/huawei-csm created

步骤3 执行cd /opt/huawei-csm/manual/huawei-csm命令,进入到华为CSM安装目录。

步骤4 执行kubectl apply -f crds/命令,安装crd资源。

# kubectl apply -f crds/ customresourcedefinition.apiextensions.k8s.io/resourcetopologies.xuanwu.huawei.io created

步骤5 执行cd /opt/huawei-csm/manual/huawei-csm/templates命令,进入到华为CSM的工作目录。

**步骤6** ( 可选 ) 采用单副本部署方式,请跳过本步骤。采用多副本部署方式时,需手动修改 templates目录下所有文件中Deployment资源的副本数,并将所有enable-leaderelection参数设置为true 。

apiVersion: apps/v1 kind: Deployment metadata: labels: app: csm-storage-service name: csm-storage-service

```
namespace: huawei-csm
spec:
progressDeadlineSeconds: 600
replicas: 2
...
args:
---enable-leader-election=true
...
```

**步骤7** ( 可选 )如果需要配置CSM服务的节点选择器,则需手动将templates目录下所有文件 中Deployment资源的nodeSelector字段取消注释,并配置相应的node匹配标签 。

```
apiVersion: apps/v1
kind: Deployment
metadata:
name: csm-prometheus-service
...
spec:
# uncomment if you wish to configure selection constraints for csm-prometheus-service pods
nodeSelector:
kubernetes.io/hostname: <host-name>
...
```

步骤8 (可选)如果需要配置Prometheus插件提供HTTPS服务,请参照表4-1手动配置证书 文件prometheus-ssl.yaml。执行命令kubectl apply -f prometheus-ssl.yaml创建 证书文件。然后手动将csm-prometheus.yaml文件中证书相关字段取消注释,并将 use-https参数修改为true。如果使用默认的HTTP服务,请跳过本步骤。

# kubectl apply -f prometheus-ssl.yaml secret/prometheus-ssl created

#### 表 4-1 prometheus-ssl 参数

参数	描述	必选参 数	默认值
data.tls.crt	Base64编码 的证书信息	是	-
data.tls.key	Base64编码 的私钥信息	是	-

```
prometheus-ssl.yaml
   containers:
     - name: prometheus-collector
        - --use-https=true # modify the value to "true" if configured the SSL cert
      volumeMounts:
# uncomment if configured the SSL cert
       - name: secret-volume
         mountPath: /etc/secret-volume
         readOnly: true
      livenessProbe:
       failureThreshold: 5
       httpGet:
# uncomment if configured the SSL cert
         scheme: HTTPS
         path: /healthz
         port: 8887
   volumes:
```

# uncomment if configured the SSL cert

- name: secret-volume

secret:

secret Name: prometheus-ssl

defaultMode: 0400

# 步骤9 执行kubectl apply -f csm-prometheus.yaml命令,部署csm-prometheus服务。

# kubectl apply -f csm-prometheus.yaml

serviceaccount/csm-prometheus-sa created

clusterrole.rbac.authorization.k8s.io/prometheus-collector-role created

clusterrolebinding.rbac.authorization.k8s.io/prometheus-collector-binding created

clusterrole.rbac.authorization.k8s.io/cmi-collector-role created

clusterrolebinding.rbac.authorization.k8s.io/cmi-collector-binding created

deployment.apps/csm-prometheus-service created

service/csm-prometheus-service created

# 步骤10 执行kubectl apply -f csm-storage.yaml命令, 部署csm-storage服务。

# kubectl apply -f csm-storage.yaml

serviceaccount/csm-storage-sa created

clusterrole.rbac.authorization.k8s.io/topo-service-role created

clusterrole.rbac.authorization.k8s.io/cmi-controller-role created

clusterrolebinding.rbac.authorization.k8s.io/topo-service-binding created

clusterrolebinding.rbac.authorization.k8s.io/cmi-controller-binding created

deployment.apps/csm-storage-service created

service/csm-storage-service created

#### **步骤11** 执行**kubectl get pod -n huawei-csm**命令,查看Pod创建情况。回显如下,表示创建 华为CSM服务成功。

# kubectl get pod -n huawei-csm

NAME READY STATUS RESTARTS AGE

csm-prometheus-service-7c8bd8fd89-lbfvn 3/3 Running 0 3m21s csm-storage-service-747dbc5cbd-n24j8 3/3 Running 0 2m19s

----结束

# 4.2.2 手动更新软件

使用远程访问工具(以PuTTY为例),通过管理IP地址,登录Kubernetes集群的任意master节点。

步骤1 参考4.2.3 手动卸载软件手动卸载旧版本华为CSM。

**步骤2** 进入到新安装包的/opt/huawei-csm/manual/huawei-csm目录下,执行**kubectl apply -f** *crds*/更新crd资源。

# kubectl apply -f crds/

customresourcedefinition.apiextensions.k8s.io/resourcetopologies.xuanwu.huawei.io configured

步骤3 参考4.2.1 手动安装软件安装当前版本华为CSM。

----结束

# 4.2.3 手动卸载软件

#### 前提条件

华为CSM已经安装成功,且华为CSM服务正常运行。

# 卸载 csm-prometheus 服务

**步骤1** 使用远程访问工具(以PuTTY为例),通过管理IP地址,登录Kubernetes集群的任意master节点。

**步骤2** 执行cd /opt/huawei-csm/manual/huawei-csm/templates命令,进入到华为CSM 的工作目录。

步骤3 执行kubectl delete -f csm-prometheus.yaml命令, 卸载csm-prometheus服务。

# kubectl delete -f csm-prometheus.yaml serviceaccount "csm-prometheus-sa" deleted clusterrole.rbac.authorization.k8s.io "prometheus-collector-role" deleted clusterrolebinding.rbac.authorization.k8s.io "prometheus-collector-binding" deleted clusterrole.rbac.authorization.k8s.io "cmi-collector-role" deleted clusterrolebinding.rbac.authorization.k8s.io "cmi-collector-binding" deleted deployment.apps "csm-prometheus-service" deleted service "csm-prometheus-service" deleted

步骤4 (可选)若没有配置证书文件请跳过本步骤。执行kubectl delete -f prometheus-ssl.yaml命令,删除证书secret文件。

# kubectl delete -f prometheus-ssl.yaml secret "prometheus-ssl" deleted

----结束

# 卸载 csm-storage 服务

**步骤1** 使用远程访问工具(以PuTTY为例),通过管理IP地址,登录Kubernetes集群的任意 master节点。

**步骤2** 执行cd /opt/huawei-csm/manual/huawei-csm/templates命令,进入到华为CSM 的工作目录。

步骤3 执行kubectl delete -f csm-storage.yaml命令,卸载csm-storage服务。

# kubectl delete -f csm-storage.yaml
serviceaccount "csm-storage-sa" deleted
clusterrole.rbac.authorization.k8s.io "topo-service-role" deleted
clusterrole.rbac.authorization.k8s.io "cmi-controller-role" deleted
clusterrolebinding.rbac.authorization.k8s.io "topo-service-binding" deleted
clusterrolebinding.rbac.authorization.k8s.io "cmi-controller-binding" deleted
deployment.apps "csm-storage-service" deleted
service "csm-storage-service" deleted

----结束

#### 其他资源卸载

**步骤1** 使用远程访问工具(以PuTTY为例),通过管理IP地址,登录Kubernetes集群的任意master节点。

步骤2 执行cd /opt/huawei-csm/manual/huawei-csm命令,进入到华为CSM安装目录。

步骤3 执行kubectl delete configmap -n huawei-csm huawei-csm-version命令,删除 configmap资源。

# kubectl delete configmap -n huawei-csm huawei-csm-version configmap "huawei-csm-version" deleted

步骤4 (可选)若采用单副本部署华为CSM,请跳过本步骤。执行kubectl delete lease -n huawei-csm resource-topology命令,删除lease资源。

**步骤5** (可选)执行kubectl delete -f crds/命令,清理crd资源。

#### 须知

- 如果后续不再使用华为CSM,且已经清理华为CSM在环境中的相关资源对象,则执 行该步骤。否则请跳过该步骤。
- 删除crd资源前,确保通过华为CSM创建的资源已经清理干净,包含的crd资源类型可查看/opt/huawei-csm/manual/huawei-csm/crds目录下的文件。
- 删除crd资源会将该crd关联的所有资源进行清理,请慎重执行该操作。
- 如果出现类似的提示: <Error from server (NotFound): error when deleting "crds/xuanwu.huawei.io\_resourcetopologies.yaml": customresourcedefinitions.apiextensions.k8s.io "resourcetopologies.xuanwu.huawei.io" not found>,则说明该crd已经卸载,请忽略该提示。

# kubectl delete -f crds/ customresourcedefinition.apiextensions.k8s.io "resourcetopologies.xuanwu.huawei.io" deleted

步骤6 执行kubectl delete ns huawei-csm命令,删除命名空间。

#### 须知

删除命名空间会将该命名空间的所有资源进行清理,请慎重执行该操作。

----结束

# **5** Prometheus 配置

- 5.1 安装Prometheus
- 5.2 配置Prometheus服务
- 5.3 配置Prometheus的仪表盘

# 5.1 安装 Prometheus

Prometheus是一款开源的监控系统和警报工具软件,Prometheus于2016年加入云原生计算基金会,成为继Kubernetes之后的第二个托管项目。

Prometheus的获取、安装:点此前往。

Prometheus的其他信息请参考:点此前往。

# 5.2 配置 Prometheus 服务

# 配置说明

Prometheus主要通过调用插件提供的接口进行对存储的数据监控,无论Prometheus部署在Kubernetes中或者是单独的部署在一个服务器上,均需要有配置文件完成Prometheus的启动与对插件的监控。

本章节仅描述如何在Prometheus配置文件中添加本插件相关的配置。其他的配置内容可参考Prometheus官方文档:点此前往。

**步骤1** 修改Prometheus的配置文件,在"scrape\_configs"配置段中添加通过CSM创建的存储后端监控项。

#### □ 说明

该配置文件为Prometheus平台的配置文件,不同的部署方式存在的位置不同,可以参考 Prometheus官方文档: **点此前往** 

#### 步骤2 添加资源监控项。配置模板如下所示:

scrape\_configs:
- job\_name: \*\*\*\*\*
scrape\_interval: \*\*\*
scrape\_timeout: \*\*\*

```
metrics_path: /object/***
scheme: http/https
params:
    controller: ["]
    storagepool: ["]
    filesystem: ["]
    lun: ["]
    array: ["]
    pv: ["]
    tls_config:
    ca_file: ******
    cert_file: ******
    key_file: ******
static_configs:
    - targets: [*:*.**:<端口号>']
```

# 表 5-1 参数说明

参数	说明	示例
job_name	监控任务名称。	job_name: OceanStor- Monitor
scheme	Web请求方式。如果不设置,则使 用默认值http。插件侧配置 tls_config后,此处需要设置为 https。	scheme: https
scrape_interv al	数据抓取间隔。	scrape_interval: 15m
scrape_timeo ut	数据抓取超时时间。推荐与 "scrape_interval"的值配置一 致。	scrape_timeout: 15m
metrics_path	数据抓取路径,格式为/object/ ***, ***对应于插件配置文件中的存储后端名。 说明 该后端需要使用存储系统组拥有对应 权限角色的用户才能查询到指标数据。	metrics_path: /object/ backend1
	有权限的角色包括超级管理员,管理 员,监控管理员。	

<sup>&</sup>quot;scrape\_configs"配置段中的参数说明如表5-1所示。

参数	说明	示例
params	集中式存储监控对象。当前支持如下类型:  controller: 控制器  storagepool: 存储池  filesystem: 文件系统  lun: LUN  array: 阵列  pv: Kubernetes中PV数据  说明  如果不关注某类对象,可删减。 对象参数值必须填["]。 对象指标数据请参考表2-4	controller: ["] storagepool: ["] filesystem: ["] lun: ["] array: ["] pv: ["]
tls_config	当scheme设置为https时的TLS配置。scheme设置为https时必填。  ca_file:用于验证API服务器证书的CA证书路径。  cert_file:用于向服务器进行客户端证书身份验证的证书文件路径。  key_file:用于向服务器进行客户端证书身份验证的秘钥文件路径。	ca_file: /opt/huawei/ca.crt cert_file: /opt/huawei/ client.crt key_file: /opt/huawei/ client.key <b>说明</b> 路径自定义。
targets	插件监听地址及端口。 对Kubernetes部署默认端口为 30074,具体端口为部署插件时填 写的端口。	['192.168.1.1:30074']

# 🗀 说明

关于"scrape\_configs"配置参数更加详细的描述,可参考Prometheus官网链接:点此前往。

# 步骤3 添加性能监控项。配置模板如下所示:

```
scrape_configs:
- job_name: ******
scrape_interval: ***
scrape_timeout: ***
metrics_path: /performance/***
params:
controller: ['21,22,370']
storagepool: ['21,22,370']
lun: ['21,22,370']
filesystem: ['182,524,525']
pv: ['filesystem,lun'] #
tls_config:
ca_file: ******
cert_file: ******
key_file: ******
```

static\_configs: - targets: ['\*.\*.\*.'<端口号>']

#### 表 5-2 参数说明

参数	说明	示例
job_name	监控任务名称。	job_name: OceanStor- Monitor
scheme	Web请求方式。如果不设置,则使 用默认值http。可设置为https。	scheme: https
scrape_interv al	数据抓取间隔。	scrape_interval: 15m
scrape_timeo ut	数据抓取超时时间。推荐与 "scrape_interval"的值配置一 致。	scrape_timeout: 15m
metrics_path	数据抓取路径,格式为/performance/***, ***对应于插件配置文件中的存储后端名。 说明 该后端需要使用存储系统组拥有对应权限角色的用户才能查询到指标数据。 有权限的角色包括超级管理员,管理员,监控管理员。	metrics_path: /performance/ backend1
params	<ul> <li>当前支持如下类型:</li> <li>controller: 控制器</li> <li>storagepool: 存储池</li> <li>filesystem: 文件系统</li> <li>lun: LUN</li> <li>pv: Kubernetes的PV数据</li> <li>说明</li> <li>如果不关注某类对象,可删减。</li> <li>对象参数值用于指定该对象的性能指标项,格式为'指标1,指标2'。如果不关注某项指标,可删减。</li> <li>性能指标数据以及对应关系请参考表2-5</li> </ul>	controller: ['21,22,23,25,26,28,370'] storagepool: ['21,22,370'] lun: ['21,22,370'] filesystem: ['182,524,525'] pv: ['filesystem,lun']

<sup>&</sup>quot;scrape\_configs"配置段中的参数说明如表5-2所示。

参数	说明	示例
tls_config	当scheme设置为https时的TLS配置。scheme设置为https时必填。  ca_file:用于验证API服务器证书的CA证书路径。  cert_file:用于向服务器进行客户端证书身份验证的证书文件路径。  key_file:用于向服务器进行客户端证书身份验证的秘钥文件路径。	ca_file: /opt/huawei/ca.crt cert_file: /opt/huawei/ client.crt key_file: /opt/huawei/ client.key <b>说明</b> 路径自定义。
targets	插件监听地址及端口。 对Kubernetes部署默认端口为 30074,具体端口为部署插件时填 写的端口。 对非Kubernetes部署无默认端口, 具体的端口为部署插件时填写的端 口。	['192.168.1.1:30074']

#### 山 说明

关于"scrape\_configs"配置参数更加详细的描述,可参考Prometheus官网链接:点此前往。

步骤4 如果要监控多个存储后端,重复执行步骤2和步骤3。

步骤5 重新启动Prometheus服务。

#### 山 说明

Prometheus为开源组件,不同的部署方式启动方式不同,可以参考Prometheus官方文档:点此前往

----结束

## 5.3 配置 Prometheus 的仪表盘

**步骤1** 在浏览器地址栏中输入Prometheus服务的IP地址及端口(端口默认为9090),登录Prometheus监控界面。

步骤2 单击 "Graph"标签页,在搜索栏中输入"huawei",选择关注的监控项。

# **6** Grafana 配置

- 6.1 安装Grafana
- 6.2 使用Grafana

## 6.1 安装 Grafana

Grafana是一个开源的可视化平台,并提供了对Prometheus的完整支持。

#### □ 说明

- Grafana 2.5.0 (发布于2015年10月28日) 起支持Prometheus作为Grafana数据源。当前华为存储仅支持对接Grafana 7.0.4。
- CCE 敏捷版自带Grafana组件且具有默认Prometheus数据源,可以直接跳至**使用Grafana监控存储**进行下一步配置。

关于如何安装Grafana,请参阅Grafana 官方文档。

## 6.2 使用 Grafana

#### 山 说明

默认情况下,Grafana将监听3000端口。

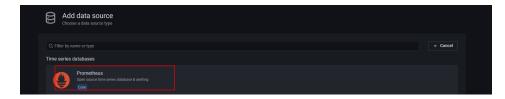
## 添加 Prometheus 并配置数据源

**步骤1** 在左侧菜单栏中选择"Configuration > Data sources",单击"Add data source"添加数据源。



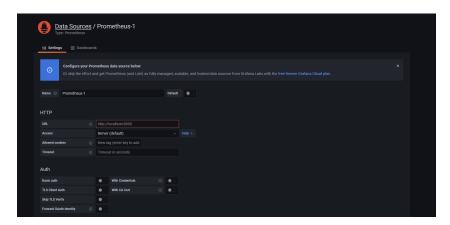


#### 步骤2 选择"Prometheus",添加数据源。



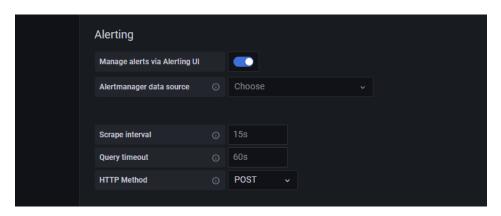
#### 步骤3 配置数据源信息。

- Name: 设置数据源的名称。
- URL: Prometheus服务器的IP地址。
- Access: 根据实际情况进行配置:
  - 配置为"Server"表示获取数据时先由浏览器发请求到Grafana的服务器上, 然后再由Grafana的服务器向Prometheus服务器发请求获取数据。
  - 配置为"Browser"表示获取数据直接由浏览器发请求到Prometheus服务器上(需要解决跨域问题)

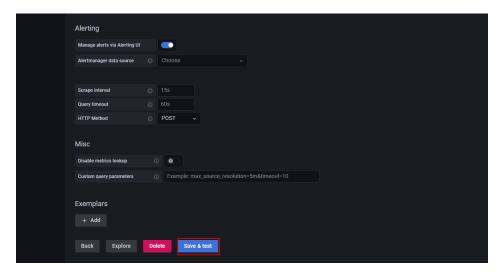


#### 步骤4 设置HTTP方法。

• POST: 推荐在查询大量数据时使用。



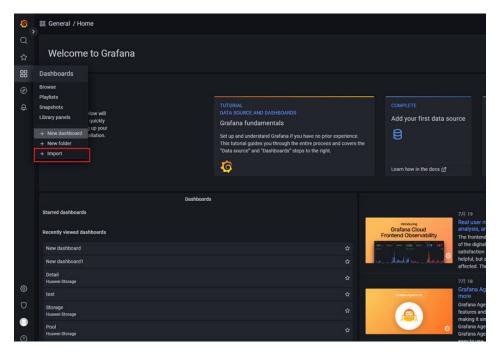
步骤5 单击"Save & Test",保存新的数据源。



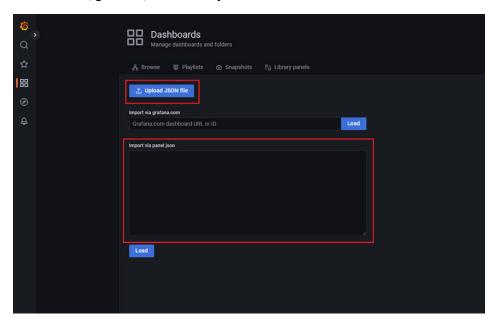
----结束

## 使用 Grafana 监控存储

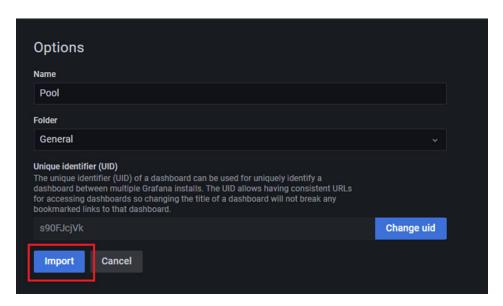
步骤1 在左侧菜单栏中选择"Dashboards > Import"。



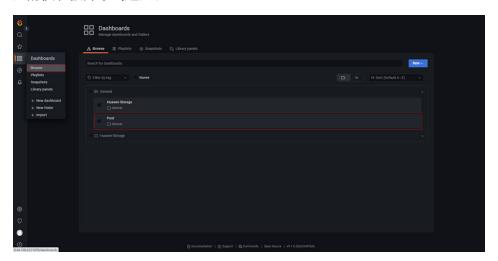
步骤2 单击"Upload JSON file"上传JSON文件或者直接粘入JSON数据即可载入现有的仪表板,载入成功后可以在Browse中查看,JSON文件位于CSM软件包解压路径下的helm/huawei-csm/grafana/OceanStor.json目录下。此JSON文件为给出的样例模板。



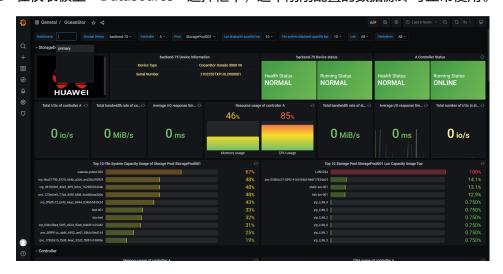
步骤3 设置完仪表板名称及所属文件夹后,单击"Import"导入。



**步骤4** 在左侧菜单栏中选择"Dashboards > Browse",在"Browse"界面中,找到刚刚导入的仪表板并单击进入。



步骤5 在仪表板上"DataSource"选择框中,选中刚刚配置的数据源即可正常使用。



## 🗀 说明

模板左上角的华为Logo需要在公网环境下才能加载。

## **了** 常用运维指导

- 7.1 日志收集
- 7.2 查看版本信息

## 7.1 日志收集

#### 须知

使用oceanctl工具收集日志可能会导致master节点CPU使用率短时间内明显增加。请根据master节点业务负载以及CPU使用情况选择使用本功能。也可在各节点的日志归档目录"/var/log/huawei-csm/"下手动收集CSM日志。

## 7.1.1 前置检查

**步骤1** 使用远程访问工具(以PuTTY为例),通过管理IP地址,登录Kubernetes集群内有 oceanctl工具的节点。

步骤2 执行oceanctl version命令,显示版本号为v4.3.0。

```
$ oceanctl version
+-----+
| OCEANCTL VERSION |
+------+
| v4.3.0 |
```

#### 步骤3 执行oceanctl --help命令,返回信息如下。

```
$ oceanctl --help
A CLI tool for Ocean Storage in Kubernetes

Usage:
oceanctl [command]

Available Commands:
collect collect messages in Kubernetes
create Create a resource to Ocean Storage in Kubernetes
delete Delete one or more resources from Ocean Storage in Kubernetes
get Get one or more resources from Ocean Storage in Kubernetes
help Help about any command
update Update a resource for Ocean Storage in Kubernetes
```

```
version Print the version of oceanctl

Flags:
-h, --help help for oceanctl

Use "oceanctl [command] --help" for more information about a command.
```

**步骤4** 执行**kubectl get deploy -n** *\${NAMESPACE}*命令,检查Pod是否正常启动,其中,\$ {NAMESPACE}为CSM安装的命名空间,以**huawei-csm**为例。

```
$ kubectl get deploy -n huawei-csm
NAME READY UP-TO-DATE AVAILABLE AGE
csm-prometheus-service 1/1 1 1 3h23m
csm-storage-service 1/1 1 1 3h23m
```

----结束

## 7.1.2 使用 oceanctl 收集 CSM 日志

- **步骤1** 使用远程访问工具(以PuTTY为例),通过管理IP地址,登录**7.1.1 前置检查**章节中检查的节点。
- **步骤2** 执行**oceanctl collect logs -n huawei-csm -a**命令,收集集群内所有CSM容器所在节点的CSM日志。

步骤3 检查/tmp目录下生成的日志压缩包,可以使用unzip \${zip\_name} -d collect\_logs解 压日志压缩包,其中\${zip\_name}为压缩包的名字。

```
$ date
Fri Sep 8 22:14:57 CST 2023
$ ls /tmp
huawei-csm-2023-09-08-22:13:01-all.zip
```

----结束

## 7.2 查看版本信息

- **步骤1** 使用远程访问工具(以PuTTY为例),通过管理IP地址,登录Kubernetes集群的任意master节点。
- 步骤2 执行kubectl get cm -n \${NAMESPACE} huawei-csm-version -o yaml命令。其中,\${NAMESPACE}表示的是命名空间名称,以huawei-csm为例。

```
$ kubectl get cm -n huawei-csm huawei-csm-version -o yaml apiVersion: v1 data:
    cmi-controller: 2.1.0 liveness-probe: 2.1.0 prometheus-collector: 2.1.0 topo-service: 2.1.0 kind: ConfigMap metadata:
    name: huawei-csm-version namespace: huawei-csm
```

# **8** <sub>附录</sub>

- 8.1 使用非root用户访问Kubernetes
- 8.2 csm-prometheus配置HTTPS服务
- 8.3 存储证书管理
- 8.4 权限矩阵

## 8.1 使用非 root 用户访问 Kubernetes

## 前提条件

该非root用户具有使用/bin/cp和/bin/chown的sudo权限。

## 操作步骤

- **步骤1** 使用远程访问工具(以PuTTY为例),通过管理IP地址,登录Kubernetes集群的任意 master节点。
- **步骤2** 执行**mkdir -p \$HOME/.kube**命令,创建Kubernetes集群的认证文件存放目录。 \$ mkdir -p \$HOME/.kube
- **步骤3** 执行**sudo cp -i /etc/kubernetes/admin.conf \$HOME/.kube/config**命令,拷贝Kubernetes集群的认证文件。

其中,/etc/kubernetes/admin.conf修改为实际使用的认证文件。 \$ sudo cp -i /etc/kubernetes/admin.conf \$HOME/.kube/config

**步骤4** 执行sudo chown \$(id -u):\$(id -g) \$HOME/.kube/config命令,修改认证文件的用户与用户组。

\$ sudo chown \$(id -u):\$(id -g) \$HOME/.kube/config

步骤5 执行以下命令,配置当前用户的KUBECONFIG环境变量,以Ubuntu 20.04为例。

\$ echo "export KUBECONFIG=\$HOME/.kube/config" >> ~/.bashrc
\$ source ~/.bashrc

## 8.2 csm-prometheus 配置 HTTPS 服务

## 8.2.1 配置 csm-prometheus 的 HTTPS 服务证书

#### 前提条件

- Kubernetes正常运行。
- 用户已获取证书和秘钥。

### 操作步骤

**步骤1** 使用远程访问工具(以PuTTY为例),通过管理IP地址,登录Kubernetes集群放置 CSM的helm文件夹的节点。

步骤2 执行cd /opt/huawei-csm/helm/huawei-csm命令,进入到Helm的工作目录。

步骤3 创建保存证书的文件夹,mkdir cert。

步骤4 将证书文件和秘钥文件放置到/opt/huawei-csm/helm/huawei-csm/cert文件夹下。

-rw-r--r-- 1 root root 4.5K Sep 8 23:00 server.crt -rw-r--r-- 1 root root 1.7K Sep 8 23:00 server.key

#### 步骤5 执行vim /opt/huawei-csm/helm/huawei-csm/values.yaml:

- 配置features.prometheusCollector.prometheusCollectorSSL.enabled为 true。
- 配置features.prometheusCollector.prometheusCollectorSSL.certPath为 "cert/server.crt"。
- 配置features.prometheusCollector.prometheusCollectorSSL.keyPath为 "cert/server.key"。

```
# all supported features
 # prometheusCollector: allowed prometheus use the storage to collect metrics
 prometheusCollector:
  # Allowed values:
  # true: enable prometheus collect feature
  # false: disable prometheus collect feature
  # Default value: false
  enabled: true
  # nodePort: port the containers are provided to the prometheus
  # Default value: 30074
  nodePort: 30074
  # prometheusCollectorSSL: parameters required to start https
  # Default value: 30074
  prometheusCollectorSSL:
    # Allowed values:
    # true: enable https, when set it certPath and keyPath must set
    # false: disable https, use http
    # Default value: true
    enabled: true
    # The Path of cert, need to be placed in the huawei-csm directory
    certPath: "cert/server.crt"
    # The Path of key, need to be placed in the huawei-csm directory
    keyPath: "cert/server.key"
```

步骤6 按4.1.1 安装软件完成安装。

## 8.2.2 删除 csm-prometheus 的 HTTPS 服务证书

#### 前提条件

- Kubernetes正常运行。
- 用户已获取证书和秘钥。

## 操作步骤

- **步骤1** 使用远程访问工具(以PuTTY为例),通过管理IP地址,登录Kubernetes集群放置 CSM的helm文件夹的节点。
- 步骤2 执行cd /opt/huawei-csm/helm/huawei-csm命令,进入到Helm的工作目录。
- **步骤3** 执行helm get values huawei-csm -n huawei-csm -a > update-value.yaml命令,获取原有服务配置文件。

步骤4 执行vim /opt/huawei-csm/helm/huawei-csm/update-value.yaml:

- 配置features.prometheusCollector.prometheusCollectorSSL.enabled为 false。
- 配置features.prometheusCollector.prometheusCollectorSSL.certPath为""。
- 配置features.prometheusCollector.prometheusCollectorSSL.keyPath为""。

features:
 prometheusCollector:
 csiDriverName: csi.huawei.com
 enabled: true
 nodePort: 30074
 prometheusCollectorSSL:
 enabled: false
 certPath: ""
 keyPath: ""

**步骤5** 执行helm upgrade huawei-csm ./ -n huawei-csm -f ./values.yaml -f update-value.yaml --wait --timeout 2m命令,升级CSM服务。回显中有Release "huawei-csm" has been upgraded,则表示升级CSM服务成功。

----结束

## 8.2.3 更新 csm-prometheus 的 HTTPS 服务证书

- **步骤1** 使用远程访问工具(以PuTTY为例),通过管理IP地址,登录Kubernetes集群放置 CSM的helm文件夹的节点。
- 步骤2 执行cd /opt/huawei-csm/helm/huawei-csm命令,进入到Helm的工作目录。
- 步骤3 创建保存证书的文件夹,mkdir cert。
- **步骤4** 将新的证书文件和秘钥文件放置到/opt/huawei-csm/helm/huawei-csm/cert文件夹下。

```
-rw-r--r- 1 root root 4.5K Sep 8 23:00 server.crt
-rw-r--r- 1 root root 1.7K Sep 8 23:00 server.key
```

**步骤5** 执行helm get values huawei-csm -n huawei-csm -a > update-value.yaml命令,获取原有服务配置文件。

步骤6 执行vim /opt/huawei-csm/helm/huawei-csm/update-value.yaml:

- 配置features.prometheusCollector.prometheusCollectorSSL.enabled为 true。
- 配置features.prometheusCollector.prometheusCollectorSSL.certPath为 "cert/server.crt"。
- 配置features.prometheusCollector.prometheusCollectorSSL.keyPath为 "cert/server.key"。

#### features:

prometheusCollector:

csiDriverName: csi.huawei.com

enabled: true nodePort: 30074

prometheusCollectorSSL:

enabled: true

certPath: "cert/server.crt" keyPath: "cert/server.key"

步骤7 执行helm upgrade huawei-csm ./ -n huawei-csm -f ./values.yaml -f update-value.yaml --wait --timeout 2m命令,升级CSM服务。回显中有Release "huawei-csm" has been upgraded,则表示升级CSM服务成功。

----结束

## 8.3 存储证书管理

参考 Huawei Storage Kubernetes CSI 用户指南中的"存储后端管理"章节。

## 8.4 权限矩阵

表 8-1 权限矩阵

ClusterRol e	ApiGroups	Resources	Verbs
prometheu s-collector- role	-	"persistentvolumes"," persistentvolumeclai ms","pods"	"get","list"
	"xuanwu.hua wei.io"	"storagebackendclaim s"	"get","list"
cmi- collector- role	"xuanwu.hua wei.io"	"storagebackendclaim s"	"get"
	-	"secrets"	"get"
	-	"configmaps"	"create", "get", "update"
topo- service-role	_	"secrets", "events", "configmaps"	"create", "get", "update", "delete"
	"coordination. k8s.io"	"leases"	"create", "get", "update", "delete"

ClusterRol e	ApiGroups	Resources	Verbs
	"xuanwu.hua wei.io"	"resourcetopologies", "resourcetopologies/ status"	"create", "get", "list", "watch", "update", "delete"
	11*11	11*11	"get", "list", "watch"
cmi- controller- role	"xuanwu.hua wei.io"	"storagebackendclaim s"	"get"
	-	"secrets"	"get"
	-	"configmaps"	"create", "get", "update"

# **9** FAQ

- 9.1 Pod 状态为OOMKilled
- 9.2 手动调整数据抓取请求并发量
- 9.3 存储侧Pod标签残留

## 9.1 Pod 状态为 OOMKilled

## 现象描述

查看Pod,Pod状态为OOMKilled,查看Pod中有打印告警事件: unable to set memory limit to xxx (current usage: xxx, peak usage: xxx): unknown。

[root@k8s-master1-yqz CSMImage]# kubectl get pod -n huawei-csm csm-prometheus-service-65b446d597-ljx7q NAME READY STATUS RESTARTS AGE csm-prometheus-service-65b446d597-ljx7q 2/3 00MKilled 3 (<invalid> ago) 43s

Marming Failed <invalid> kubelet Error: failed to create containerd task: failed to create shim task: OCI runtime create failed: runc create failed: unable to start container process: error during container init: error setting cgroup config for procisoss process: unable to set memory limit to 5242880 (current usage: 6651904, peak usage: 7139228): unknown

#### 根因分析

Pod中容器内存使用达到上限,需要增加可使用内存上限。

## 解决措施

修改对应容器资源配置,并更新CSM。详细步骤请参考4.1.3 升级软件。

## 9.2 手动调整数据抓取请求并发量

- **步骤1** 使用远程访问工具(以PuTTY为例),通过管理IP地址,登录Kubernetes集群的任意master节点。
- 步骤2 执行命令kubectl edit deployments.apps -n huawei-csm csm-prometheus-service,编辑deployments配置项。
- 步骤3 以配置请求并发量10为例,向name为cmi-controller的容器中添加args列表项--client-max-threads=10,如下所示。

```
apiVersion: apps/v1
kind: Deployment
metadata:
 name: csm-prometheus-service
 namespace: huawei-csm
spec:
 template:
  metadata:
   creationTimestamp: null
   labels:
    app: csm-prometheus-service
  spec:
   containers:
    - args:
     - --cmi-address=$(ENDPOINT)
     - --cmi-name=cmi.huawei.com
     - --page-size=100
     - --backend-namespace=huawei-csi
     - --log-file-dir=/var/log/huawei-csm/csm-prometheus-service
     - --log-file=cmi-service
     - --logging-module=file
     - --log-level=info
     - --log-file-size=20M
     - --max-backups=9
     - --client-max-threads=10
     name: cmi-controller
```

步骤4 添加完成后输入:wq命令保存配置并退出。

步骤5 执行命令kubectl get pod -n huawei-csm等待容器重启完成。

步骤6 执行命令kubectl get deployments.apps -n huawei-csm csm-prometheusservice -o yaml | grep client-max-threads查看是否配置成功。

----结束

## 9.3 存储侧 Pod 标签残留

### 现象描述

在CSM版本从1.x.x升级到2.x.x后,存储上的资源有Pod拓扑关系残留,且无法清理。

## 根因分析

在1.x.x版本中,topo资源的名称来源于存储的资源名称,而在2.x.x版本中topo资源的名称来源更改为集群上的PV名称,并将前缀从topo-更换为rt-以做区分。

#### 1.x.x版本:

```
NAME PROVISIONER VOLUMEHANDLE

STATUS AGE
topo-pvc-0a8f7871-f26e-4665-b4be-53dfa9c878cb cmi.huawei.com 181-iscsi.pvc-0a8f7871-f26e-4665-b4be-53dfa9c878cb Normal 353d
topo-pvc-3b6b1dd6-b3dd-4394-8251-4bd6009411cf cmi.huawei.com 181-iscsi.pvc-3b6b1dd6-b3dd-4394-8251-4bd6009411cf Normal 354d
```

#### 2.x.x及之后版本:

NAME PROVISIONER VOLUMEHANDLE

STATUS AGE

rt-pvc-0a8f7871-f26e-4665-b4be-53dfa9c878cb cmi.huawei.com 181-iscsi.pvc-0a8f7871-f26e-4665-b4be-53dfa9c878cb Normal 353d

rt-pvc-3b6b1dd6-b3dd-4394-8251-4bd6009411cf cmi.huawei.com 181-iscsi.pvc-3b6b1dd6-b3dd-4394-8251-4bd6009411cf Normal 354d

在升级到2.x.x版本后,CSM拓扑服务会根据集群上的资源信息创建出新的topo资源,并将新的topo资源与存储上旧版本拓扑关系绑定。

若在关闭CSM拓扑服务的情况下删除已上报拓扑关系到存储的Pod,那么CSM拓扑服务重新启用后,无法根据集群上已存在的资源创建出相应的topo资源绑定存储上旧的拓扑关系。

此时存储上旧的拓扑关系残留,且无法清理。

## 解决措施或规避方法

出现上述存储拓扑关系残留的情况,仅能通过删除存储资源的方式清理残留的拓扑关系。

在CSM版本1.x.x升级到2.x.x的过程中,若旧版本启用CSM拓扑服务,那么在升级过程中也启用CSM拓扑服务可以有效防止上述场景出现。