

PingCAP 中文技术文档风格指南

v1.1

1. 关于本指南	4
1.1 目的	4
1.2 适用范围	4
1.3 使用原则	4
1.4 用词说明	4
2. 语言风格	5
2.1 对话式	5
2.2 客观礼貌	5
2.3 简洁清晰	5
2.4 通俗易懂	6
2.5 用户导向	6
2.6 用词恰当	6
2.6.1 禁用词	6
2.6.2 常用词	6
3. 文档结构样式	8
3.1 元数据	8
3.2 标题	9
3.2.1 标题的层级	10
3.2.2 标题的描述	11
3.2.3 使用标题的注意事项	11
3.3 目录	12
3.4 段落	12
3.5 句子	13
4. 文档内容元素	13
4.1 空白符号	13
4.1.1 空白的使用	13
4.1.2 Tab 和空格的使用	14
4.2 列表	14
4.2.1 无序列表和有序列表	14
4.2.2 列表的使用	15

4.3 表格	17
4.4 图形和图片	18
4.5 注意和说明	18
4.6 版本标签	19
4.7 代码块	20
4.8 代码注释	23
4.9 链接	24
4.9.1 链接的描述	24
4.9.2 链接的路径	25
4.9.3 PingCAP 文档链接规则	25
4.10 引用	26
4.11 缩略语	27
4.11.1 汉语缩略语	27
4.11.2 英语缩略语	27
4.12 数字	28
4.12.1 基本规则	28
4.12.2 多位数值	28
4.12.3 数字范围	28
4.12.4 倍数表达	29
4.12.5 时间与日期	29
4.12.6 物理量	30
4.12.7 非物理量	30
4.13 单位符号	30
5. 标点符号	30
5.1 中文标点使用	30
5.2 中英文混排格式	31
5.3 常用中文标点符号	31
5.3.1 句号	31
5.3.2 逗号	32
5.3.3 顿号	32
5.3.4 分号	32
5.3.5 冒号	33
5.3.6 引号	33
5.3.7 括号	34
5.3.8 书名号	35
5.3.9 连接号	35
5.3.10 破折号	36
5.3.11 省略号	36

5.3.12 感叹号	36
5.3.13 斜杠与反斜杠	37
5.3.14 反引号	37
6. 名称与命名	38
6.1 文件命名	38
6.2 产品命名	38
6.3 名称使用	39
7. 拼写与语法	39
7.1 拼写	39
7.2 语法	40
7.3 文档质量检查工具	41
8. 附录资料及相关说明	41
8.1 术语表和缩略语表	41
9. 参考资料	41

1. 关于本指南

本指南为 PingCAP 公司 (以下简称 “PingCAP”) 中文技术文档的编写提供了一套风格指南规范。具体而言，本指南在中文技术文档的**语言风格、结构样式、内容元素、标点符号、格式排版**等方面给出明确要求或参考建议。

1.1 目的

- 提高中文文案可读性
- 统一文档风格，保证公司对外输出形象一致
- 避免不同的文档作者对同一问题反复作出决策，降低与文档相关的沟通成本

1.2 适用范围

- 为编写中文文档的作者 (主要是产品研发人员、DBA 等) 提供规范或建议
- 审校文档过程中争议问题的裁决
- 也可供软件界面、帮助文档等资料开发人员参考

1.3 使用原则

- 如对本指南提供的风格有任何疑问，请咨询 I18N 团队成员。
- 如遇本指南未提及的风格，请向 I18N 团队进行确认，以免出现风格不统一的情况。
- 本指南保持更新，欢迎使用者提出任何改进意见。

1.4 用词说明

本指南使用的表示“要求”的全部关键词已在下表第二列列出，具体释义请参见 [RFC2119](#) 对相应词语做出的相关规定：

RFC2119 中定义的关键词	对应的中文关键词	释义说明
MUST/REQUIRED/SHALL	强制/必须/务必/只能	强制性规则，表示绝对要求这样做
MUST NOT/SHALL NOT	禁止/不能/不要	强制性规则，表示绝对禁止这样做
SHOULD/RECOMMENDED	应/应当/应该/建议/推荐	非强制性规则，表示一般情况下应该这样做，但在知悉全部后果的前提下，可以选择不这样做

SHOULD NOT/NOT RECOMMENDED	不应当/不应该/不建议/不推荐	非强制性规则，表示一般情况下不应该这样做，但在知悉全部后果的前提下，可以选择这样做
MAY/OPTIONAL	可以/可选	非强制性规则，表示这个要求是可选的，可以这样做，也可以不这样做

注：RFC (Request For Comments) 指关于互联网标准的正式文件，在这些文件的表述过程中，必须严格区分哪些是“建议” (suggestion)，哪些是“要求” (requirement)。所以，RFC2119 专门对五个关键词的涵义作出了规定，分别表示“要求”的严格程度。

2. 语言风格

本章对 PingCAP 中文技术文档中使用的语言风格作出了统一规范。

2.1 对话式

技术文档中应使用对话式的、平易近人的语气。由于技术文档传达的技术信息本身已较为枯燥难懂，因此不建议使用过于正式、无聊乏味的语气。

使用对话式的语气并不意味着完全用日常说话的方式来写文档。一般而言，口语存在冗长啰嗦、缺乏逻辑性的倾向，在编写技术文档时应极力避免该问题。

2.2 客观礼貌

技术文档中应保持客观礼貌的语气，这样最容易拉近与读者的距离。具体要求有：

- 客观地传达技术信息，而不是在推销产品，否则易引起读者反感。
- 保持一种友好礼貌的语气，不要显得强硬粗鲁。
- 文档中不要穿插太多玩笑，偶尔滑稽一次是可取的，这样能适当展现作者和公司的个性，使人印象深刻。但必须记住，技术文档的首要目的是向读者传达技术信息，不能为了追求轻松愉快的文档风格而使读者不明所以。

2.3 简洁清晰

技术文档中应使用精练的语言。**建议作者在完成初稿后再通篇读一遍文档**，将文中所有对表达意思没有明显作用的字、词、句删去，在不影响表达效果的前提下把文案长度减到最短。具体要求有：

- 禁止啰嗦冗长
- 禁止逻辑混乱
- 同一文档中勿重复表达同一事物
- 尽量用主动时态，尤其要**阐述清楚主语和宾语**

2.4 通俗易懂

技术文档中不推荐使用只有特定人群才了解的语词。具体要求有：

- 不推荐使用行话黑话、俚语、脏话等，比如“魔改”（即做特殊的优化）、“CPU 打到 60%”（即 CPU 使用率为 60%）等行话黑话
- 不推荐使用网络流行语，比如“墙裂”、“童鞋们”等流行语中故意的谐音错别字

2.5 用户导向

技术文档中应以用户为导向。为编写出可用性较高的文档，文档作者应尽量站在用户的角度思考问题。具体要求有：

- 文档作者应充分考虑文档受众的技术水平分布以及实际操作中可能出现的问题，尽可能全面、清晰地将技术信息普及给大众。
- 对于操作型技术文档，除语言审校外，应继续进行“文档可用性测试”（如 I18N 部门发起的 Queeny's Test）——由一位无技术背景的测试人员参照文档进行完整操作，如操作顺利成功，则该文档可用性测试通过；如失败，则需要继续修改完善文档。

2.6 用词恰当

用词恰当体现在两个方面：用词正确及用词统一。本节从禁用词和常用语两方面介绍了相应规范。

2.6.1 禁用词

用词正确体现在不使用有冒犯性的禁用词语。技术文档中的禁用词可参考[新华社 2015 年 11 月发布的《新华社新闻报道中的禁用词（第一批）》](#)。技术文档虽不是新闻报道，但作为技术传播领域的大众传播物，应同样考虑文档传播带来的影响。避免使用具有冒犯性的词语，能为个人或公司节省不必要的麻烦。

以下是《新华社新闻报道中的禁用词（第一批）》中比较适用于技术文档的几点：

- 报道各种事实特别是产品、商品时不使用“最佳”、“最好”、“最著名”、“最新技术”、“最高水平”、“最先进水平”等具有强烈评价色彩的词语。
- 对有身体伤疾的人士不使用“残废人”、“独眼龙”、“瞎子”、“聋子”、“傻子”、“呆子”、“弱智”等蔑称，而应使用“残疾人”、“盲人”、“聋人”、“智力障碍者”等词语。

2.6.2 常用词

常用词是指在编写一篇或一系列技术文档时，经常被使用的词语，如人称代词、指示代词、行文语态助词、操作动词、技术术语等。

技术文档中必须正确使用各种常用词。具体要求有：

1. 必须用对“的”、“地”、“得”，不能乱用。

- 【正确示例一】PD 会将数据迁移到其他的 TiKV 节点上。（形容词 + 的 + 名词）
- 【正确示例二】TiDB 可以显式地使用事务。（副词 + 地 + 动词）
- 【正确示例三】这个值不宜调得过大。（动词 + 得 + 副词）

2. 必须明确“其”、“该”、“此”、“这”等代词指代的内容，保证不造成歧义。

- 【示例一】
 - 【错误】从管理系统可以监视中继系统和受其直接控制的分配系统。
 - 【正确】从管理系统可以监视两个系统：中继系统和受中继系统直接控制的分配系统。
- 【示例二】
 - 【错误】如果希望从本地已编译好的 binary 文件构建 PD、TiKV 或 TiDB 镜像，需要将其 `image` 字段留空。
 - 【正确】如果希望从本地已编译好的 binary 文件构建 PD、TiKV 或 TiDB 镜像，需要将相应镜像的 `image` 字段留空。

3. 不建议使用表示程度、强调语气的词，因为这类词词义通常比较模糊，或者显得主观绝对，建议用具体的数据或实例代替。例如，以下类型的词建议避免使用。

- 表示程度的词：很好、很多、较多、较好、完全地、基本地、决定性地等
 - 【错误示例】很好地提升了性能。
 - 【分析】诸如“很好地”这样的词含义既模糊又主观，建议用具体的数据代替。另外，性能的含义也比较广泛，可以详细说明。
 - 【修改建议】性能提升了 50% 或者 .99 延迟从 10ms 降为 1ms。
- 表示量的词：有些、非常、大量、一些、少许、部分、几乎、数倍等
 - 【错误示例】...建表语句执行时间会是关闭该变量的数倍。
 - 【分析】对于用户来说，“数倍”的含义很模糊，三倍和八倍的差别是巨大的。这种情况下，可以说明一句影响建表语句执行时间的因素有哪些，并举一两个具体的实例加以说明。
 - 【修改建议】... 建表语句执行时间会是关闭该变量的数倍，具体多少倍取决于硬件和 PD 参数的配置。如当硬件为 ... 且 PD 参数配置为 ... 时，建表语句执行时间会是关闭该变量的 ... 倍。

4. 不建议使用冷僻、生造或者文言文的词语，应该使用现代汉语的常用表达方式。

- 【错误示例一】这是唯二的快速启动的方法。
- 【正确示例一】这是仅有的两种快速启动的方法。

5. 禁止使用过多的形容词或词组修饰名词。

- 【错误示例】根据表名恢复被删除的表，会找到最近历史 DDL JOB 中的第一个是 DROP TABLE 类型的 DDL 且 DROP TABLE 的表名等于 RECOVER TABLE 语句中指定的表名的表进行恢复。
- 【分析】这句话有太多修饰的词组，读起来很拗口，令人费解。这种情况下，建议适当断句以明晰语义。
- 【正确示例】根据表名恢复被删除表的过程是：首先找到最近历史 DDL JOB 中的第一个是 DROP TABLE 类型的 DDL 语句，并且该 DROP TABLE 语句中指定的表名等于 RECOVER TABLE 语句中指定的表名，再对这张表进行恢复。

另外，同一篇或同一系列技术文档中应尽可能统一用词，以降低阅读理解的难度。具体要求有：

1. 必须保证全文人称代词一致，人称不能反复改变。

- 推荐使用“您”或“你”来指称文档读者或用户，两者皆可，但同一文档内禁止混用
- 推荐使用“作者”、“文档作者”等第三人称形式来代指文档作者，不推荐使用“我”来代指文档作者，这样容易显得主观，也会在读者心中引起不必要的疑惑
- 可以使用“我们”来代称整个公司，但建议少用

2. 建议尽量使用主动语态，不使用被动语态。

- 【错误示例】`auth_string` 会被 TiDB 经过加密存储在 `mysql.user` 表中。
- 【正确示例】TiDB 会将 `auth_string` 加密存储在 `mysql.user` 表中。

【注意】汉语中使用**被字句**跟英语中使用**被动式**是不同的。在英语中，使用被动式的目的是为了**避免提及施事者**，但在汉语中的被字句往往带有被动的负面含义。另外，在中文文档中使用主动语态能帮助**明确句子主语和宾语**，这对后续的技术翻译工作极为重要。

3. 文档结构样式

本章对构成一篇中文技术文档的结构样式作出了统一规范。

3.1 元数据

PingCAP 中文技术文档采用 Markdown 语言编写，最终将发布到官方网站上，因此我们需要元数据 (metadata) 来做搜索引擎优化 (SEO)，让用户在谷歌等搜索引擎更容易搜索到相关文档。

PingCAP 中文文档的元数据必须包含三个属性：title、summary（摘要）和 category。标准格式如下。

```
---
title: 从 Amazon Aurora MySQL 迁移数据
summary: 使用 DM 从 Amazon Aurora MySQL 迁移数据。
category: how-to
---
```

中文文档的元数据应遵循以下格式规范。

- 元数据必须加在文档最开头，并由“---”包裹起来。
- 元数据写完后，空一行后接一级标题。
- 元数据中的 title 必须与正文的一级标题保持一致。
- title、summary、category、aliases 等属性名称全小写，后接英文冒号并空一格后，才能写具体的属性内容。
- 严禁自行添加除 title、category、aliases、summary 等之外的元数据信息。
- aliases 不是必需属性，新增的文档或者无需重定向的文档中禁止添加 aliases 属性。

中文文档的元数据应遵循以下内容规范。

- title 的写法：参考[标题的描述](#)一节，因为 title 与正文标题必须一致。
- summary 的写法：
 - 和 title 不同的是，summary 是全文的一句话简单摘要，所以必须是一个完整的句子（后面加句号）。
 - 建议 summary 尽可能多地包含本文的关键词，这有利于 SEO。
- category 的写法：根据文档所在目录一般有 benchmark、FAQ、how-to、reference、Releases、tispark、introduction 等。如这几种皆不适用，请咨询 I18N 部门。
- 如需添加多个 alias，则用英文逗号分隔不同 alias，逗号两边不空格，内容示例如下：
aliases: ["/docs/op-guide/ansible-deployment/", "/docs/op-guide/aaa-bbb/"]。

附：

有关 aliases 的更多说明，请参考 [Content Strategist Guide - Add aliases to a document](#)。

有关元数据撰写的更多规范，请参考[官网中英文上传文章 metadata 规范汇总](#)。

3.2 标题

标题在技术文档中的地位非常重要，文档作者要设计合理的标题层级和标题描述，从而帮助读者理清整篇文档的逻辑，使文章结构一目了然。

3.2.1 标题的层级

技术文档中使用标题最多不超过四级。标题从一级开始递增使用，禁止跳级使用。例如：一级标题下面不能直接使用三级标题；二级标题下面不能直接使用四级标题。

- 一级标题：文章标题
- 二级标题：文章正文部分的标题
- 三级标题：二级标题下面一级的小标题
- 四级标题：三级标题下面某一方面的小标题

下图为在 Markdown 技术文档中使用标题的示例，左侧是编辑文字，右侧是预览效果：



为避免出现过于复杂的章节，若无特殊需要，不建议使用四级标题。如果三级标题下有并列性的内容，建议使用列表 (list) 代替四级标题。如下图中，若内容 A、B、C 的篇幅不长，则右侧的标题样式比左侧的标题样式要好。

三级标题

这是三级标题的内容。

四级标题 A

这是四级标题的内容 A。

四级标题 B

这是四级标题的内容 B。

四级标题 C

这是四级标题的内容 C。

三级标题

这是三级标题的内容。

1. 内容 A

2. 内容 B

3. 内容 C

3.2.2 标题的描述

技术文档中的标题包括但不限于以下几种描述：

- 名词词组，如“...概述”、“背景”、“原理”
- 主题词+动词，如“A 工具安装”、“A 工具部署”、“A 工具配置”
- 动词+主题词，如“配置 MySQL 数据库”
- 定语+主题词，如“A 工具的安装”，“A 工具的架构”
- 介词+定语+主题词，如“对机器配置的要求”

标题描述的设计并无严格的模板，只要遵循以下几个原则即可：

- 标题能够概括反映本章节的中心内容。
- 标题简洁扼要、涵义明确。
- 同级别的标题尽量使用相同的结构。
- 标题描述**操作任务**时建议使用“动词+主题词”结构，不建议使用名词词组。

3.2.3 使用标题的注意事项

技术文档中使用标题主要有以下几个注意事项：

- 一级标题必须与文首元数据的 title 保持一致。
- 下级标题禁止重复上一级标题的内容。
- 禁止标题以标点符号（如句号、冒号、问号等）结尾，FAQ 文档除外（可以用问号结尾）。
- 禁止在标题中解释缩略语。
- **标题与标题之间一定要有引导介绍性的句子。**例如，一级标题和二级标题之间应有引言，二级标题和三级标题之间应有正文内容。
- 标题要避免孤立编号（即同级标题只有一个），正文不要有孤立的三级标题和四级标题。

- 项目列表（包括无序和有序列表）是最小编号单位，因此项目列表下禁止嵌套任何级别的标题。

3.3 目录

目录通过各级标题自动生成，帮助用户快速浏览全文结构和定位章节。

对于一本技术手册而言，必须提供总目录（包含所有章节及附录）。如果是安装手册等还需要提供图目录、表目录。（目前不适用单独的一篇技术文档）

对于一篇用 Markdown 语言编写的技术文档而言，除 TOC.md 文件外，一般不需要在文档中任何位置提供目录。禁止在非 TOC.md 的文档中手动生成或添加目录。

【注意】如果一篇文档是新增的，并且需要在官网显示该文档，则必须手动添加该文档目录到相应根目录下的 TOC.md，如 dev/TOC.md 或 v3.0/TOC.md 或 v2.1/TOC.md，如下：

```
v3.0 ▸ TOC.md ▸ abc # TiDB 中文用户文档
1  #·TiDB·中文用户文档
2
3  ##·目录
4
5  +·[关于·TiDB](overview.md)
6  +·主要概念
7  |···[整体架构](architecture.md)
8  |···+·[核心特性](key-features.md)
9  |···|···[水平扩展](key-features.md#水平扩展)
10 |···|···[高可用](key-features.md#高可用)
11 +·操作指南
12 |···+·快速上手
13 |···|···+·创建集群
14 |···|···|···+·使用·Kubernetes
15 |···|···|···|···[DinD](how-to/get-started/deploy-tidb-from-kubernetes-dind.md)
16 |···|···|···|···[Minikube](how-to/get-started/deploy-tidb-from-kubernetes-minikube.md)
17 |···|···|···|···[GKE](how-to/get-started/deploy-tidb-from-kubernetes-gke.md)
```

3.4 段落

段落是正文部分的基本构成单元之一，由多个句子组成。PingCAP 段落写作要求如下：

- 段落之间使用一个空行隔开。
- 段落开头不需要缩进，顶格开始即可。
- 一个段落只能有一个主题，或一个中心句子。
- 一个段落里避免只有一个句子。如果句子很长，要避免“一逗到底”的情况，合理断句。
- 一个段落的长度建议在 50~200 字之间，尽量不要超过 250 字。（Word 里统计字数）
- 段落的句子语气应该使用陈述和肯定语气，避免使用感叹语气。

- 对于技术描述类主题，应考虑先图表，后句子的原则，不要单一的使用段落来陈述主题。
- 段落的中心句子建议放在段首，对全段内容进行概述。后面陈述的句子为核心句服务。

3.5 句子

句子以句号结尾，句号表示句子意思已完成。PingCAP 中文技术文档句子写作要求如下：

- 句子要避免使用长句。一个句子建议不超过 100 字（正文 3 行）。
- 句子要使用简单句和并列句，避免使用复合句。
- 善于断句，避免“一逗到底”的现象。

【错误示例】原因是 DM 需要保存同步的 binlog position 信息，但是 MySQL binlog position 官方定义使用 uint32 存储，所以超过 4G 部分的 binlog position 的 offset 值会溢出，就会存储的是一个错误的 binlog position，在重启 task 或者 dm-worker 后，需要使用该 binlog position 重新解析 binlog/relay log，进而出现上面的错误。

【分析】以上句子为多个分句构成的复合句。“一逗到底”的情况增加了理解整体句群含义的难度。这种情况下，应该在适当的地方进行断句，并添加“这”、“其”等代词，合理切分句与句之间的逻辑。

【修改建议】由于 DM 需要保存同步的 binlog position 信息，且 MySQL binlog position 官方定义使用 uint32 存储，因此超过 4G 部分的 binlog position 的 offset 值会溢出。**这会导致存储的是一个错误的 binlog position。**在重启 task 或者 dm-worker 后，需要使用该 binlog position 重新解析 binlog/relay log，进而出现上面的错误。

4. 文档内容元素

本章对 PingCAP 中文技术文档中包含的内容元素作出了统一规范。

4.1 空白符号

空白符号包括空格、空行等，其中空格分为半角空格和全角空格。

4.1.1 空白的使用

PingCAP 技术文档中使用空白符号应遵循以下规范。

空格	<ul style="list-style-type: none"> • 禁止使用全角空格，一律使用半角空格。 • 中文字符（包括汉字和中文标点符号）和中文字符之间禁止空半角空格。 • 中文标点符号前、后禁止空格。 • 对于英文字符和阿拉伯数字，应使用半角空格包围，以下情况例外：
----	--

	<ul style="list-style-type: none"> ○ 位于句首时，左边空格省略。 ○ 其右侧为全角标点时，右边空格省略。 ● 除表示缩进、列表级别、代码块中固有空格外，禁止连续出现两个及以上的半角空格。 ● 禁止使用 Tab (制表符) 替换空格。
空行	<ul style="list-style-type: none"> ● 不同段落间必须使用一个空行隔开。 ● 插入表格、图片等元素时，插入的语句前后需有一个空行。 ● 不同排版格式之间使用一个空行隔开，如标题后的正文，正文中的代码块等。 ● 禁止连续出现两个及以上的空行。

4.1.2 Tab 和空格的使用

技术文档中经常使用 Tab 键和空格键进行缩进和对齐。由于在不同的编辑器里 Tab 的默认长度可能不一致，用 Tab 键设置缩进可能导致格式混乱。如果使用空格键设置缩进，则用任何编辑器打开文档都会显示一样的对齐效果。

因此，PingCAP 中文技术文档中**必须使用空格键**而不用 Tab 键进行缩进或对齐。

【建议】如使用 Visual Studio Code 等编辑器编写文档，可以统一设置一个 Tab 等于四个半角空格。

4.2 列表

当有 3 项或更多重要信息需要展示时，纵向列表是最清晰且吸引眼球的方式。但如果项目少于 3 项且无需特别强调，将其直接放在句子中通常效果更好。

也可以创建多级嵌套列表，在某一级别下另起一行，缩进**四个空格**即可开始更低级别的列表。

4.2.1 无序列表和有序列表

技术文档中的列表分为有序列表 (ordered list) 和无序列表 (unordered list) 两种。一般而言，当列表项之间的顺序不重要时，使用无序列表；当各项之间的顺序很重要时，使用有序列表。

【无序列表示例】

目前，TiDB 使用了以下组件：

- Prometheus Server：用于收集和存储时间序列数据。
- Client 代码库：用于定制程序中需要的 Metric。
- Alertmanager：用于实现报警机制。

【有序列表示例】

解决办法：

1. 编辑数据源文件。
2. 手动创建所有的表。
3. 设置参数跳过检查。

有序列表的使用场景较少。当列表项的内容是以下几种时，应该使用有序列表。

- 必须按顺序操作的步骤（最常用）
- 需要进行排名的多项内容
- 需要在下文进行引用的规则或其它信息（比如下文需要引用该列表的第 3 项时可以说“规则 3”）

【重要原则】除非顺序很重要，否则不要使用有序列表！

4.2.2 列表的使用

PingCAP 技术文档中使用列表应遵循以下规范。

使用列表的规范	错误案例	正确案例
1. 并列列表项中尽量使用相似的句子结构。	SQL 查询优化器： <ul style="list-style-type: none"> ● 支持 eager aggregate ● 更详细的 explain 信息 ● union 算子并行化 ● 子查询性能优化 ● 优化 CBO 框架 	SQL 查询优化器： <ul style="list-style-type: none"> ● 支持 eager aggregate ● 支持更详细的 explain 信息 ● 支持 union 算子并行化 ● 优化子查询性能 ● 优化 CBO 框架
2. 每一项的长度尽量接近。	在 GitHub 上提交的新 Issue 分为以下几种： <ul style="list-style-type: none"> ● 如果您发现了 bug 需要报告 ● 请求开发一项新功能 ● 常规问题 ● 为解决或提升性能提的 Issue 	在 GitHub 上提交的新 Issue 分为以下几种： <ul style="list-style-type: none"> ● 错误报告 ● 功能请求 ● 常规问题 ● 性能问题
3. 避免在每一项开头重复相同的词语。	TiDB 与 MySQL 安全特性的差异： <ul style="list-style-type: none"> ● 不支持外部身份验证方式。 ● 不支持列级别权限设置。 	相较于 MySQL 安全特性，TiDB 不支持的功能有如下几种： <ul style="list-style-type: none"> ● 外部身份验证方式 ● 列级别权限设置 ● 证书验证身份方式

	<ul style="list-style-type: none"> 不支持使用证书验证身份。 	
4. 使用清晰的、描述性的句子或短语来引出列表。	<p>我们可以通过这个 store 的 state_name 来确定这个 store 的状态：</p> <ul style="list-style-type: none"> Up：这个 store 正常服务 Disconnected：当前没有检测到这个 store 的心跳，可能是故障或网络连接中断 Down：超过一小时没有收到 store 心跳，此时 PD 会为此 store 上的数据添加副本 	<p>通过该 store 的 state_name 确定其状态：</p> <ul style="list-style-type: none"> Up：该 store 正常服务。 Disconnected：当前没有检测到该 store 的心跳，可能出现故障或网络连接中断。 Down：超过一小时没有收到 store 心跳，此时 PD 会为此 store 上的数据添加副本。
<p>5. 并列列表项中保持标点符号一致。</p> <p>若列表项均为句子，那么每一项建议以句号结尾；</p> <p>若列表项均为词组，则不能以任何标点结尾。</p> <p>若列表项既有词组又有句子，则统一以句号结尾。</p>	<p>【例一】TiDB Binlog 支持以下功能场景：</p> <ul style="list-style-type: none"> 数据同步。 实时备份和恢复。 <p>【例二】指定数据源的相关信息：</p> <ul style="list-style-type: none"> 在 Name 处，为数据源指定一个名称。 在 Type 处，选择 Prometheus。 在 URL 处，指定 Prometheus 的 IP 地址。 其它字段 	<p>【例一】TiDB Binlog 支持以下功能场景：</p> <ul style="list-style-type: none"> 数据同步 实时备份和恢复 <p>【例二】指定数据源的相关信息：</p> <ul style="list-style-type: none"> 在 Name 处，为数据源指定一个名称。 在 Type 处，选择 Prometheus。 在 URL 处，指定 Prometheus 的 IP 地址。 其它字段。
6. 不要滥用无序列表，否则会导致它们失去应有的效果。	无	无
<p>7. 避免过多嵌套使用列表，这样通常会显得冗长复杂。</p> <p>如果一定要表现多层级的列表，要保证最多不超过 3 级，且每一级都要用不</p>	无	无

同样式的小圆点。		
8. 一个操作任务的步骤描述通常要使用有序列表，为方便用户记忆，建议严格限制列表项在 7 个以下，最多不要超过 9 个步骤。	无	无

4.3 表格

PingCAP 技术文档中使用表格应遵循以下规范。

1. 样式规范

- 单元格中的所有内容都保持**左对齐**。
- 所有表格都必须有**表头描述**，即表格第一行中描述各列内容的短语。
- 尽量保证表格同一行的单元格不跨页。（非 .md 文件需要考虑）
- 每个表格下都要空一行，这一空行的样式为正文。如遇到换页的情况，表格后的空行在新页的第一行，则这行空行必须删除。

2. 表头规范

同一技术文档或产品手册中相同类型表格的表头描述需要保持一致。

【注意】对参数进行说明的表，应该统一表头描述，不要出现“参数说明”、“参数解释”、“参数含义”、“参数意义”等不同的描述。

【示例】同一文档中类似如下结构的表，第一列的表头不能既有“变量名”又有“变量”，或者第二列的表头不能既有“作用”又有“功能”。

变量名	作用	单位	默认值
tidb_mem_quota_query	配置整条 SQL 的内存使用阈值	Byte	32 << 30
tidb_mem_quota_hashjoin	配置 Hash Join 的内存使用阈值	Byte	32 << 30
tidb_mem_quota_mergejoin	配置 Merge Join 的内存使用阈值	Byte	32 << 30

3. 单元格内容规范

- 表格中的内容应该尽量简练，文字表述风格保持一致。避免长篇大段的说明，建议单元格中的内容所占行数不要超过 6 行。

- 不出现空白的单元格。建议无内容单元格填写“无”或特定含义符号（如“/”）。若使用特定含义符号，需说明符号代表的意义。
- 单元格内容避免重复。如果多个单元格中的内容相同，建议将内容复制或者采用多个单元格合并的方式，不要使用“同上”。

4. 表说明规范

表说明即表格之前的一行表描述，通常带有序号或编号，主要是为了简要描述表格内容。

- 表说明尽量采用名词词组形式。
- 表说明简明扼要，长度不能超过表的宽度。
- 同类表格的表说明在全文中风格保持一致。

4.4 图形和图片

PingCAP 技术文档中插入图形与图片应遵循以下规范。

- 必须使用清晰可辨的图形图片。
- 只能插入 .png（推荐）和 .jpg 格式的图片，**禁止使用 .svg 格式图片**，否则会导致官网 PDF 构建失败。
- 中文文档里图形图片上的文字尽量都用中文，如果原图文字是其他语言，应先做好图片本地化工作。
- 同一文档内图形图片上的中文应该统一字体（如“宋体”），英文和数字也要统一字体（如“Arial”）。
- 图形图片中避免出现大段文字，描述性语言建议放到图外，用编号替代。
- 图形图片中包含缩略语时，需要在图说明中对缩略语进行解释。
- 图说明和图尽量保证在同一页中显示。
- 图片的 ALT 属性为图片的替换文字，在 Markdown 格式中位于感叹号之后的中括号中，如 `![ALT]()`，图片的 ALT 属性必须为图片的简短描述，不可为空，这有利于搜索引擎优化。
 - 【示例】`![TiDB architecture platform](/media/tidb-architecture-platform.png)`
- 图片的命名必须使用描述性的文字，这有利于搜索引擎优化。

4.5 注意和说明

注意和说明内容在技术文档中起强调效果。PingCAP 技术文档中使用注意和说明应遵循以下规范。

- 根据提示内容的级别和分类使用不同的文字描述。PingCAP 中文文档中的分类包括“警告”、“注意”等。
 - 警告和注意的格式一致，只需选择相应的词语：

```
> **警告/注意：**
```

```
>
> 警告内容或注意内容.....
```

- 官网渲染后的警告效果：

警告：

该文档中部署 TiDB 的操作指导**不适用于**在生产或研发环境中部署 TiDB 的情况。

- 官网渲染后的注意效果：

注意：

目前 TiDB 尚不支持强制验证客户端身份，即服务端对客户端的身份验证是可选的。若客户端在 TLS 握手时未出示自己的身份证书，也能正常建立 TLS 连接。

- 注意和说明中不能包含表格和图形。
- 注意和说明的内容不能过长，建议不要超过 4 行。
- 不要滥用注意或说明文字。建议不要连用两段注意或说明。

4.6 版本标签

PingCAP 技术文档中支持为某个变量、配置项或功能添加版本标签。

例如，“tidb_allow_remove_auto_inc”系统变量是从 v3.0.4 版本开始引入，则可以在该变量后添加一行版本说明：

```
<span class="version-mark">从 v3.0.4 版本开始引入</span>
```

```
###·tidb_allow_remove_auto_inc·<span class="version-mark">从 v3.0.4 版本开始引入</span>
```

作用域：SESSION

默认值：0

这个变量用来控制是否允许通过 `ALTER TABLE MODIFY` 或 `ALTER TABLE CHANGE` 来移除某个列的 `auto_increment` 属性。默认为不允许。

官网渲染效果：

tidb_allow_remove_auto_inc  从 v3.0.4 版本开始引入

作用域：SESSION

默认值：0

这个变量用来控制是否允许通过 `ALTER TABLE MODIFY` 或 `ALTER TABLE CHANGE` 来移除某个列的 `auto_increment` 属性。默认为不允许。

PingCAP 技术文档中添加版本标签应遵循以下规范。

- 版本标签加在变量、配置项或功能等名称之后。
- 同一文档或同一类型的版本标签内容尽量统一风格，比如统一写成“从...版本开始引入”、“仅在...版本支持”等。

4.7 代码块

PingCAP 技术文档使用 Markdown 语言编写，插入及高亮代码块的常用方式有两种：

- 用“`”包裹语句中的某个参数名或关键字，如 ``key_word``（`中内容自动高亮）。
- 用“```”包裹多行代码，如需高亮代码块，只需在第一个“```”之后加上相应的语法名称。
 - 常用的语法名称有 `sql`, `bash`, `shell`, `sh`, `vim`, `yaml`, `toml`, `ruby`, `rust`, `go`, `json` 等
 - 【示例一】

```
```sql
create table t (c int);
```
```

- 【示例二】

```
```shell
(gdb) set sysroot /proc/${pid}/root/
```
```

PingCAP 技术文档中插入代码块应遵循以下规范。

- 代码块前后必须加上一行空行。
- 代码块要注意缩进。主要有下面两种情况。
 - a. 如果当前代码块**属于正文内容**，则该代码块无需缩进。例如：

<顶格>使用该参数来尽量避免在恢复数据时报以下错误：

```
<顶格>```
<顶格>packet for query is too large.
<顶格>```
```

- b. 如果当前代码块从属于列表项下的内容，则要在该列表项的缩进基础上再缩进四格。例如（列表项的基础缩进为零，则从属于该列表项的代码块需缩进四格）：

```
<顶格>1. 打开连接后，即可通过本地的对应端口访问 PD 服务和 TiKV 节点。
```

```
<空四格>{{< copyable "shell-regular" >}}
```

```
<空四格>```shell
<空四格>tikv-ctl --host 127.0.0.1:20160 <subcommands>
<空四格>```
```

- 如果代码块里的内容是用户不需要复制的（比如代码执行结果、无需复制的配置文件片段等），则不用在代码块前加上任何 code snippet。
- 如果代码块里的内容是用户需要复制的（比如可执行的 shell/bash 命令、SQL 语句等），则为了在渲染官网时代码块前能自动添加相应命令前缀（\$、# 或 >），你必须在代码块前添加渲染程序可识别的 code snippet。有以下几种情况：

- a. 如果代码块内是 SQL 语句，则在代码块前添加 {{< copyable "sql" >}}。

■ 【示例】

```
{{< copyable "sql" >}}

```sql
SELECT * FROM t1;
```
```

- 【效果】渲染程序将自动在代码块前添加前缀 > 符号。

```
> SELECT * FROM t1; Copy
```

- b. 如果代码块内是普通用户执行的 shell 或 bash 命令，则在代码块前添加 {{< copyable "shell-regular" >}}。

■ 【示例】

```
{{< copyable "shell-regular" >}}
```

```
```shell
mysql -h 127.0.0.1 -P 4000 -u root
```
```

- 【效果】渲染程序将自动在代码块前添加前缀 \$ 符号。

```
$ mysql -h 127.0.0.1 -P 4000 -u root
```

Copy

- 如果代码块内是 **root 用户** 执行的 shell 或 bash 命令，则在代码块前添加 {{< copyable "shell-root" >}}。

- 【示例】

```
{{< copyable "shell-root" >}}
```

```
```bash
useradd -m -d /home/tidb tidb
```
```

- 【效果】渲染程序将自动在代码块前添加前缀 # 符号。

```
# useradd -m -d /home/tidb tidb
```

Copy

- 如果代码块内既不是 SQL 语句又不是 bash/shell 命令，但仍然支持用户复制，比如在 yaml 配置文件中新增的一行配置，则在代码块前添加 {{< copyable "" >}}。

- 【示例】

```
{{< copyable "" >}}
```

```
```yaml
tidb:
 service:
 type: NodePort
 # externalTrafficPolicy: Cluster
 # annotations:
 # cloud.google.com/load-balancer-type: Internal
```
```

- 【效果】渲染程序不会在代码块前添加任何符号，但支持复制。

```
tidb:
  service:
    type: NodePort
    # externalTrafficPolicy: Cluster
    # annotations:
    # cloud.google.com/load-balancer-type: Internal
```

[Copy](#)

【注意】

1. {{< copyable "...>}} 这行 code snippet 和之后的代码块之间**必须空一行**。
2. 可供用户直接复制执行的代码块里尽量**不要添加代码注释**。
3. **一个代码块内尽量只写一行命令**（多行命令需拆到多个代码块里），这样方便解释每一条命令的作用，对小白用户比较友好。
4. 如果必须将多行 bash/shell 命令放在同一代码块内进行复制，可以通过 **&& 或 &** 实现多步命令一次性复制并执行的效果。

○ 【示例】

```
{{< copyable "shell-regular" >}}
```

```
```shell
yum -y install epel-release git curl sshpass &&
yum -y install python2-pip
```
```

或者：

```
{{< copyable "shell-regular" >}}
```

```
```shell
yum -y install epel-release git curl sshpass &
yum -y install python2-pip
```
```

- && 的含义是前一步命令执行不成功，后一步命令不执行；& 的含义是前一步命令即使执行不成功，后一步命令也会执行。

5. 多行 SQL 语句无需加 && 或 &，否则会报错。

4.8 代码注释

PingCAP 中文技术文档中的代码注释一般出现在由三个反引号包裹的多行代码块中。

但根据现有的代码块书写规范，代码块内应**尽量减少**直接用“#”或“//”等注释符进行注释的行为，应该尝试将代码注释的内容写到正文中，以解释代码块。这是为了方便用户在阅读官网文档时能直接复制并执行代码块中的语句或命令。（详情请参见相关的 [Jira Issue](#)）

对于必须要添加的代码注释，应遵循以下规范。

- 一行注释不能太长，太长时应适当进行断句并切分到下一行。
- 一行注释末尾必须有标点符号，一般是句号；与下一行的内容承接时，用逗号。
- 必须根据``后定义的语法选择相应的注释符，不能自创注释符，一般使用“#”或“//”进行注释。

附：有关 PingCAP 代码注释的规范，还可参考 I18N 陈纯写的 [Code Comment Style](#)。

4.9 链接

技术文档中的链接将用户引导至同一文档中的其他位置、其他相邻文档或外部站点。本节主要介绍了在 PingCAP 的 Markdown 技术文档中使用链接应遵循的规范。

【Markdown 链接示例】

- 链至同一文档中的其他位置：
 - [产品架构](#产品架构)
 - [使用 dmctl 加密上游 MySQL 用户密码](#使用-dmctl-加密上游-mysql-用户密码)
 - [选项 1：使用单节点上单个 DM-Worker 实例的集群拓扑](#选项-1使用单节点上单个-dm-worker-实例的集群拓扑)
- 链至其他相邻文档：[产品架构](/dev/how-to/maintain/tidb-binlog.md)
- 链至外部站点：[贡献者指南](<https://docs.microsoft.com/zh-cn/contribute/>)

4.9.1 链接的描述

Markdown 链接中方括号 [] 里的内容即为该链接的描述性文本。链接的描述需要符合以下规范。

- 链接描述必须能概括所链文档或页面的大致内容，这有利于搜索引擎优化。例如，链接描述可以是所链页面的标题。
 - **【错误示例】**
 - 详情参见 trouble-shooting.md
 - 详情请点击[此处](trouble-shooting.md)
 - **【正确示例】**
 - 详情参见[故障诊断文档](trouble-shooting.md)。

- 同类型的链接描述尽量统一风格。例如：同一文档内不宜多次出现“详情参见”、“详情参阅”、“具体见”、“具体请见”等表达相同意思的不同描述。

4.9.2 链接的路径

Markdown 链接中圆括号 () 里的内容即为该链接的路径。链接的路径需要符合以下规范。

- 如链至其他相邻文档，且链接的文档篇幅较长，应该链接至锚点。链接至锚点即链接至某级标题处。Markdown 支持在链接路径的文件名后加“#标题名称”，即可以链接至该文件的特定标题处。
 - 【示例】在使用 metrics 接口前，需先[部署 Prometheus 和 Grafana](/dev/how-to/deploy/orchestrated/ansible#在中控机上配置部署机器-ssh-互信及-sudo-规则)。
- 链接路径应统一风格，不要混用相对路径和绝对路径。PingCAP 中文文档中链接至非外部站点时必须统一使用相对路径，如 /dev/how-to/maintain/tidb-binlog.md。
- 建议减少将用户链至外部站点的次数，以免外部站点的页面失效而影响读者体验。

4.9.3 PingCAP 文档链接规则

本节详细介绍了适用于 GitHub pingcap/docs 和 pingcap/docs-cn 内文档的链接路径规则，如不遵守，会导致相应的 Pull Request 无法通过 CI 死链检查，或者可能导致 PingCAP 官网文档中出现死链。

规则一：任何链至其他文件的相对路径都必须从 /dev/ 或 /v3.0/ 或 /v2.1/ 或其他同级目录开始写起。（注意不能省去最前面的 / 符号）

- 【错误示例】
/reference/configuration/tidb-server/configuration-file.md#slow-threshold
- 【正确示例】
/v3.0/reference/configuration/tidb-server/configuration-file.md#slow-threshold

【注意】

- 本规则不适用于链至同一文件内的相对路径。如需链至同一文件的某级标题处，只需在链接路径处直接写上“#标题名称”即可。

【示例】详细信息参见[第 3 步：部署 Drainer](#第-3-步部署-drainer)。

- 本规则不影响使用绝对路径的外部链接写法。

【示例】[TiDB-Ansible](https://github.com/pingcap/tidb-ansible) 是正确的。

对于链至某级标题（即锚点）的相对链接，“#”后的锚点链接必须遵循以下 Github Flavored 标题锚点规范：

规则二：锚点链接里的字母必须全部改成小写形式。

- 【示例】[使用 dmctl 加密上游 MySQL 用户密码](#使用-dmctl-加密上游-mysql-用户密码)

规则三：锚点链接里的空格必须使用连字符“-”代替。

- 【示例】[使用 Ansible 滚动升级](#使用-ansible-滚动升级)

规则四：锚点链接里的字母、数字、下划线“_”、连字符“-”及 [CJK 常用中文 4E00-9FFF](#) 保留原样，其余特殊符号（如顿号、逗号、冒号、引号、反引号、问号、括号等标点，&、/、=、% 等符号）都直接删去。

- 【示例一】[TiKV_channel_full_total](#tikv_channel_full_total)
- 【示例二】[--config](#--config) 一节。
- 【示例三】[分配机器资源，编辑 inventory.ini 文件](#分配机器资源编辑-inventoryini-文件)。
- 【示例四】[TiDB DM (Data Migration)](#tidb-dm-data-migration)
- 【示例五】[2.2.1.4 supervise/svc/svstat 服务具体起什么作用？](#2214-supervisesvcsvstat-服务具体起什么作用)
- 【示例六】[2.2.8 使用 TiDB Ansible 部署 TiDB 集群的时候，遇到 `UNREACHABLE! "msg": "Failed to connect to the host via ssh: "` 报错是什么原因？](#228-使用-tidb-ansible-部署-tidb-集群的时候遇到-unreachable-msg-failed-to-connect-to-the-host-via-ssh--报错是什么原因)

4.10 引用

PingCAP 技术文档中使用引用应遵循以下规范。

- 当某内容在其他地方已经详细描述过、不适合在正文中再次介绍时，可以引用。
- 对于必须引用但内容很少（少于 100 字）的情况，建议直接在该处重新描述一遍。
- 必须保证引用的位置准确。

技术文档中引用第三方内容应遵循以下规范。

- 引用他人的语句时，应注明出处。
 - 【示例】One man's constant is another man's variable. — Alan Perlis

- 全篇转载时，应在全文开头显著位置注明作者和出处，并链接至原文。
 - 【示例】本文转载自 WikiQuote
- 使用外部图片时，必须在图片下方或文末标明来源。
 - 【示例】本文部分图片来自 Wikipedia

4.11 缩略语

中文技术文档中的缩略语有两种：汉语缩略语和英语缩略语。下文分别介绍了两者的概念、特点及使用规范。

4.11.1 汉语缩略语

汉语缩略语是由较长的中文语词缩短省略而成的语词，如“人大”、“重启”、“停机”、“绑核”等。

由于汉语缩略语一般数量较少、含义明确，在技术文档中只要保证该缩略语通俗易懂、不造成歧义，原则上不限制使用次数。如果某词在文档中必须大量使用，但其缩略语不常见，建议在该词第一次出现时说明情况，提示读者下文中将以缩略语的形式称呼该词。

4.11.2 英语缩略语

英语缩略语数量巨大、种类繁多，出现在中文技术文档中的主要有三种：首字母缩略词 (acronym)、字母词 (initialism) 和缩写词 (shortened word)。三者的区别是：

- 首字母缩略词：由每个词的首字母组成，以词的形式发音。
 - 【示例】NATO 代表 North Atlantic Treaty Organization
- 字母词：也是由每个词的首字母组成，但按字母逐字发音。
 - 【示例】FBI 代表 Federal Bureau of Investigation
- 缩写词：由较长的中文语词缩短省略而成的语词。
 - 【示例】App 代表 Application；demo 代表 demonstration

在 PingCAP 技术文档中使用英语缩略语应遵循以下规范。

- 不建议在标题中解释英文缩略语，以免造成标题冗长。
- 建议在正文中第一次出现缩略语的地方解释其完整含义。

- 某词用缩略语进行代称时，必须在该词第一次出现时说明情况，提示读者下文中将以缩略语的形式称呼该词。
- 禁止使用不规范的缩略语，例如，用“16c32g”表示“16 核、32 GB”。

4.12 数字

PingCAP 技术文档中书写和使用数字应遵循以下规范。

4.12.1 基本规则

数字的格式应遵循如下规范。

- 数字与英文字母一样，一律使用半角形式，不得使用全角形式。
 - 【错误示例】每 2 分钟导入一个 2 5 6 MB 的数据文件。
 - 【正确示例】每 2 分钟导入一个 256 MB 的数据文件。
- 要求半角数字两旁各空一个空格。
 - 【错误示例】每2分钟导入一个 256MB 的数据文件。
 - 【正确示例】每 2 分钟导入一个 256 MB 的数据文件。

数字的书面写法有两种：文字型和阿拉伯型。具体使用哪种写法应遵循如下规范。

- 在统计表中必须使用阿拉伯数字，如正负整数、小数、百分比、分数、比例等。
- 固定表达式的词组、术语、带有修饰色彩的词语中应该使用文字型数字，如三三制、五局三胜制、第三季度。
- 当与“几”、“余”、“多”搭配表示约数、概数时，一律使用文字型数字，不得使用阿拉伯数字，如“几十个”、“一百多个”、“十几天”。

4.12.2 多位数值

技术文档中书写多位数值应遵循以下规范。

- 数值巨大的精确数字，建议使用“亿、万”作单位。如，10 亿 231 万 3245 人。
- 数值为千位以上，建议添加半角逗号作为分节符。多位整数从右到左分节，如 2,345,567,456。
- 使用阿拉伯型表示的数字不能断开移行。

4.12.3 数字范围

技术文档中书写数字范围应遵循以下规范。

- 表示数值范围时，用“~”连接。

- 数字带有国际单位和表示量的数词时，建议两个数字都加上数值和单位。
 - 【错误示例】132~234 kg、67~89%、64~128 MB。
 - 【正确示例】132 kg~234 kg、67%~89%、64 MB~128 MB。

4.12.4 倍数表达

技术文档中表达倍数应遵循以下规范。

- 数值的增加必须明确使用“**增加了**”或“**增加到**”，不能只使用“增加”。“了”表增量，“到”表定量。
 - 【错误示例】增加两倍
 - 【正确示例一】增加了两倍——即过去为一，现在为三。
 - 【正确示例二】增加到过去的两倍——即过去为一，现在为二。
- 数值的减少必须明确使用“**降低了**”或“**降低到**”，不能只使用“降低”。“了”表增量，“到”表定量。
 - 【错误示例】降低 80%
 - 【正确示例一】降低了 80%——即原来是一百，现在是二十。
 - 【正确示例二】降低到 80%——即原来是一百，现在是八十。
- 不能用“降低 N 倍”或“减少 N 倍”的表示法，要用“**降低百分之几**”或“**减少百分之几**”。

4.12.5 时间与日期

技术文档中使用阿拉伯数字表示时间与日期的情况如下。

- 公历世纪、年代、年、月、日应该使用阿拉伯数字。如，20 世纪 80 年代，2003 年 7 月 30 日。（一般而言，“1990 年”在技术文档中不能简写成“90 年”）
- 在表格、索引、年表中，年月日可以表示为“2003-07-30”。月和日为个位数时，要加“0”补充。
- 时、分、秒。正式情况，采用如下格式：14 时 12 分 36 秒。其他情况，按照 GB-T 7408-94-5.3.1.1 要求，使用扩展格式：14:12:16。

使用文字型数字表示时间与日期的情况如下。（技术文档中应避免使用）

- 农历年份。如正月初五、丙寅年十月十五日；
- 含有时间简称来表示时间、节日的词组。如“九一三”事件、五四运动。

4.12.6 物理量

物理量量值必须用阿拉伯数字表示，并正确使用法定计量单位。[物理量](#)包括：长度、时间、速度、面积、体积、频率等。

4.12.7 非物理量

一般情况下非物理量使用阿拉伯数字表示，比如 45.6 万元、1480 人、100 TB。

在没有统计意义的词组中，整数一到十可以使用文字型数字，也可以用阿拉伯数字，但全文必须保持一致。比如一个人、三本书、六条语句；翻译专业学生 6 个、全职员工 14 人。

4.13 单位符号

技术文档中使用单位符号应遵循以下规范。

- 为避免指代不明，中文文档中不建议用单位符号代替汉字名称，如 mm 代替毫米、h 代替小时、min 代替分钟。
- 大多数情况下，数值与单位符号之间需要空一个半角空格，如 7 kg、8 MB、10 GB。但有几种例外：
 - 单位是角度、摄氏度或百分号时，数值和单位之间不空格。如：60°、37°C、100%
 - 单位是英尺符号和英寸符号时，数值和单位之间不空格。如：4'5" 表示 4 英尺 5 英寸

5. 标点符号

技术文档中的标点符号极易用错，文档作者必须牢记规范，保证文档的美观性和可读性。

本章对 PingCAP 技术文档中使用的标点符号作出了统一规范。

5.1 中文标点使用

使用中文标点符号应遵循以下规范。

- **中文语句中的标点符号一律使用全角形式**（即中文输入法下的标点符号），不得使用半角形式（即英文输入法下的标点符号）。
- 中文全角标点符号两旁禁止空半角空格。示例：
 - **【错误示例】**如果 CPU 设有限额（从 K8S 指定的上限），则需要手动调整。

- 【正确示例】如果 CPU 设有限额（从 K8S 指定的上限），则需要手动调整。

5.2 中英文混排格式

中文技术文档中不仅会出现中文标点，也可能出现英文标点，因此在中英文混排时应着重注意中英文标点的使用。

中英文混排时使用标点符号应遵循以下规范。

- **括号里全为英文时**使用半角括号，并在括号前后各空一个半角空格，括号和括号内的英文之间不需要空格。
 - 【错误示例一】数据定义语言 (DDL) 是一种..... (使用了全角括号)
 - 【错误示例二】数据定义语言(DDL)是一种..... (半角括号前后未空格)
 - 【错误示例三】数据定义语言 (DDL) 是一种..... (半角括号和半角括号内的英文之间空了一格)
 - 【正确示例】数据定义语言 (DDL) 是一种.....
- **括号里既有中文又有英文（即只要括号内包含任何中文）时**使用全角括号，括号前后不空格。
 - 【错误示例】斜杠 (slash 或 forward slash) 和反斜杠 (backslash) 是两种符号。
 - 【正确示例】斜杠 (slash 或 forward slash) 和反斜杠 (backslash) 是两种符号。
- **全角标点与英文或数字之间不加空格。**
 - 【错误示例】我刚买了台 iPhone，好开心！
 - 【正确示例】我刚买了台 iPhone，好开心！

5.3 常用中文标点符号

技术文档中容易用错的标点符号主要有：句号、逗号、顿号、分号、冒号、引号、括号、书名号、连接号、破折号、省略号、感叹号、斜杠与反斜杠。下文介绍了它们的使用规范。

5.3.1 句号

句号的形式为“。”，常用于陈述句末尾的停顿。

句号表示一个句子的意思已经完整，技术文档中应善用句号切分语意，帮助用户理清逻辑。

- 【错误示例】原因是 DM 需要保存同步的 binlog position 信息，但是 MySQL binlog position 官方定义使用 uint32 存储，所以超过 4G 部分的 binlog position 的 offset 值会

溢出，就会存储的是一个错误的 binlog position，在重启 task 或者 dm-worker 后，需要使用该 binlog position 重新解析 binlog/relay log，进而出现上面的错误。

- **【正确示例】**由于 DM 需要保存同步的 binlog position 信息，且 MySQL binlog position 官方定义使用 uint32 存储，因此超过 4G 部分的 binlog position 的 offset 值会溢出。这会导致存储的是一个错误的 binlog position。在重启 task 或者 dm-worker 后，需要使用该 binlog position 重新解析 binlog/relay log，进而出现上面的错误。

5.3.2 逗号

逗号的形式为“，”，表示句子内部的一般性停顿。

逗号虽然没有特殊、专门的意义，使用也最普遍，但是不能滥用。技术文档中不要出现“一‘逗’到底”的错误，即整个段落除了段落结尾外，全部停顿都使用逗号。

- **【错误示例】**首先判断出错发生在 relay log 写入还是 binlog replication/syncer unit 同步（通过日志出错信息中的 component 信息即可判断），如果错误发生在 relay log 模块，binlog replication/syncer unit 保存的断点都是正确的情况，可以先停止任务，停止 DM-worker，手动调节 relay meta 的 binlog-position 到 4，重启 DM-worker 重新拉取 relay log，relay log 写入正常后启动任务会自动从断点继续同步。
- **【正确示例】**首先通过日志出错信息中的 component 信息判断是在 relay log 写入还是 binlog replication/syncer unit 同步时出错。如果错误发生在 relay log 模块，且 binlog replication/syncer unit 保存的断点都正确，则可以先停止任务和 DM-worker，手动调节 relay meta 的 binlog-position 到 4，重启 DM-worker 重新拉取 relay log。relay log 写入正常后启动任务会自动从断点继续同步。

5.3.3 顿号

顿号的形式为“、”，表示句子内部并列词语之间的停顿。中文中表述三者或三者以上的并列情况时，必须用顿号，而不用逗号表示。

- **【错误示例】**此操作会关闭 TiDB，Pump，TiKV，PD 服务，并清空 Pump，TiKV，PD 数据目录。
- **【正确示例】**此操作会关闭 TiDB、Pump、TiKV 和 PD 服务，并清空 Pump、TiKV 和 PD 数据目录。

【注意】英文中没有顿号，常用逗号来表示并列词语的停顿。

5.3.4 分号

分号的形式为“；”，表示复句内部并列分句之间的停顿。一般情况下，并列分句有三句或超过三句时，建议使用分号表示停顿。

【示例】Placement Driver (简称 PD) 是整个集群的管理模块，其主要工作有三个：一是存储集群的元信息（某个 Key 存储在哪个 TiKV 节点）；二是对 TiKV 集群进行调度和负载均衡（如数据的迁移、Raft group leader 的迁移等）；三是分配全局唯一且递增的事务 ID。

5.3.5 冒号

冒号的形式为“：”。在技术文档中常用在需要解释的词语后边，表示引出解释和说明。

冒号在技术文档中常用来引出列表。

【示例】设备分为以下几个部分：

- 机柜
- 单板
- 外接设备

建议不要连续使用冒号。

- 【错误示例】计费策略：表示计费：1，表示不计费：2。
- 【正确示例】计费策略：1 表示计费，2 表示不计费。

5.3.6 引号

引号分为直角引号和弯引号。直角引号为「」，弯引号有双引号和单引号两种形式。PingCAP 中文技术文档中统一使用弯引号，禁止使用直角引号。

本节只讨论弯引号使用的一些规范，下文中使用的“引号”含义即为“弯引号”含义。

- 引号里面还要用引号时，外面一层用双引号，里面一层用单引号。
- 技术文档出现报错信息、特定操作或名称、缩略语提示、特殊名词等时，建议使用引号。
 - 【正确示例一】启动集群或者升级集群过程中出现“Timeout when waiting for search string 200 OK”是什么原因？
 - 【正确示例二】Sysbench 按照“建表->插入数据->创建索引”的顺序导入数据。
 - 【正确示例三】如果 `tidb-lightning` 曾经异常退出，集群可能仍留在“导入模式” (import mode)，不适合在生产环境工作。此时需要强制切换回“普通模式” (normal mode)。
 - 【正确示例四】作为 NewSQL 数据库，TiDB 兼顾了传统关系型数据库的优秀特性以及 NoSQL 数据库可扩展性，以及跨数据中心（下文简称“中心”）场景下的高可用。
 - 【正确示例五】以下操作可能会形成一个“关系环”。

5.3.7 括号

括号的常用形式包括圆括号“()”、方括号“[]”、方头括号“【】”以及尖括号“<>”（也称单书名号）。除 markdown 特有的格式需求（如用 []() 引用链接）外，PingCAP 中文技术文档正文中常出现的是圆括号和尖括号。

1. 使用圆括号的场景

中文文档中一个词语或句子后紧跟的注释性文字，用圆括号进行注释。注释词语时，圆括号放在被注释词语之后；注释整个句子的，圆括号放在句末标点之后。

【正确示例一】TiDB 使用周期性运行的 GC（Garbage Collection，垃圾回收）来进行清理。

【正确示例二】写技术说明类文档和文学创作不同，不能摊开稿纸搞“即兴”。（其实文学创作也要有素养才能有“即兴”。）

中文文档中可能用到全角圆括号，也可能用到半角圆括号。具体规范如下。

- **括号里全为英文时**使用半角括号，并在括号前后各空一个半角空格，括号和括号内的英文之间不需要空格。
 - 【错误示例一】数据定义语言（DDL）是一种……（使用了全角括号）
 - 【错误示例二】数据定义语言(DDL)是一种……（半角括号前后未空格）
 - 【错误示例三】数据定义语言 (DDL) 是一种……（半角括号和半角括号内的英文之间空了一格）
 - 【正确示例】数据定义语言 (DDL) 是一种……
- **括号里既有中文又有英文（即只要括号内包含任何中文）**时使用全角括号，括号前后不空格。
 - 【错误示例】斜杠 (slash 或 forward slash) 和反斜杠 (backslash) 是两种符号。
 - 【正确示例】斜杠（ slash 或 forward slash ）和反斜杠 (backslash) 是两种符号。

2. 使用尖括号的场景

由于 bare URL 在 pingcap/docs-cn 仓库中会触发 markdownlint 报错，所以必须用**尖括号包裹 bare URL**，这也方便用户直接点击并打开该 URL：

- 【错误示例】访问集群 Grafana 监控页面：http://localhost:3000。
- 【正确示例】访问集群 Grafana 监控页面：<<http://localhost:3000>>。

5.3.8 书名号

书名号的形式为双书名号“《》”。书名、篇名、报纸名、刊物名等，用书名号标示。英文手册名称用双引号或斜体表示，不用书名号标示。

【正确示例】详情见《TiDB-Lightning 表库过滤》。

书名号中的文档名称必须使用全称，不能使用简称。

- 【错误示例】具体操作请参见《S8016 安装手册》。
- 【正确示例】具体操作请参见《Quidway S8016 路由交换机安装手册》。

5.3.9 连接号

连接号的常见形式为“—”，占一个字的位置。连接号还有另外四种形式，即长横“——”（占两个字的位置）、半字线“-”（占半个字的位置）、短划线“-”（占1/3个字的位置）和浪纹“~”（占一个字的位置）。

几种连接号的用法及插入方式如下表所示。

| 符号名称 | 符号 | 举例 | 插入方式 |
|------|----|--|---|
| 一字线 | — | 常用于汉语名词之间：
压力—温度曲线、化学—物理反应 | Word 里选择[插入/特殊符号]，选择特殊符号中的“长划线”。 |
| 半字线 | - | 常用于汉字和数字(字母)之间：
铀-235、惠普-5000 | Word 里选择[插入/特殊符号]，选择特殊符号中的“短划线”。 |
| 短划线 | - | 一些外来的词汇中常用短划线：
E-Mail、CD-ROM、ITU-T、8-bit、RJ-45 插头、STM-16

电话号码中的连接符也统一用短划线：025-12345678 | Word 里选择[插入/特殊符号]，选择特殊符号中的“不间断连字符”，或者在英文输入法中按减号键输入。 |
| 浪纹 | ~ | 用于表示数量的范围：1~3 | 英文输入法中，通过键盘 <shift+~> 输入。 |
| 长横 | —— | 常用于地名之间：北京——天津 | Word 里选择[插入/特殊符号]，选择特殊符号中的“长划线”，连续输入两次。 |

5.3.10 破折号

破折号的形式为“——”，占两个字的位置。技术文档中破折号常用于引出注释和说明部分。

【示例】一种有机地结合了高速大容量光纤传输技术和智能网络技术的新体制——光同步传送网应运而生。

使用破折号要注意以下几点。

- 破折号前后不空格。
 - 【错误示例】一种新体制 —— 光同步传送网应运而生。
 - 【正确示例】一种新体制——光同步传送网应运而生。
- 破折号与连接号中的长横都是占两格的长横，符号形式相同。但两者含义不同，破折号用来解释、说明前文；连接号是把前后相关词语连成一个整体。

5.3.11 省略号

中文省略号的形式为“……”，有六个小圆点，占两个字的位置。一般而言，中文语句中禁止使用三个小圆点“...”（英文省略号），必须使用六个小圆点“……”。

省略号主要在以下两种场景中使用。

- 引文的省略，用省略号标明。
 - 【示例】中文风格指南中要求：“句号常用于陈述句末尾的停顿。在技术文档中常用于简单句和复合句的结尾，表示句子意思已完整……”
- 列举的省略，用省略号标明。
 - 【示例】分支选择支持所有类型的分区表，无论是 range 分区或是 hash 分区等。对于 hash 分区，如果没有指定分区名，会自动使用 `p0`、`p1`、`p2`、……、或 `pN-1` 作为分区名。

5.3.12 感叹号

技术文档中使用感叹号应遵循以下规范。

- 尽量使用平静的语气叙述，避免使用感叹号。
- 禁止多个感叹号连用，比如“！！”和“!!!”。

5.3.13 斜杠与反斜杠

斜杠 (slash 或 forward slash) 的符号是“/”，表示分隔符。反斜杠 (backslash) 的符号是“\”，表示转义字符。反斜杠一般只出现在代码中，因此本节主要讨论斜杠的使用。

【注意】在 Github docs-cn repo 里开启新的 Pull Request 时，PR 标题中严禁使用反斜杠，否则会导致 PingCAP 官网构建失败。详见 [PR 标题写作规范](#)。

斜杠有如下几种用法。

- 表示除法，如 $120/60 = 2$
- 表示单位（除法的变形），如 60 km/h
- 表示“或”，如他/她。注意：如果需要表示多个并列项，禁止使用“/”连接，必须使用顿号连接。
 - 【错误示例】高可用是 TiDB 的另一大特点，TiDB/TiKV/PD 这三个组件都能容忍部分实例失效，不影响整个集群的可用性。
 - 【正确示例】高可用是 TiDB 的另一大特点，TiDB、TiKV 和 PD 这三个组件都能容忍部分实例失效，不影响整个集群的可用性。

斜杠的两旁禁止加空格。

- 【错误示例】do / ignore 规则
- 【正确示例】do/ignore 规则

5.3.14 反引号

反引号 (backtick) 的符号是“`”，一般在英文输入法下由键盘左上角 Tab 键上方的按键键入。

反引号有如下两种用法。

| | 1. 单个反引号包裹单行元素 | 2. 三个反引号包裹多行元素 |
|------|--|--|
| 作用 | 强调包裹的部分，与正文内容区分 | <ul style="list-style-type: none">• 便于展示多行内容• 强调包裹的部分，与正文内容区分• 前面的```后若指明代码块中的语法，可高亮代码块（详见代码块一节） |
| 用法示例 | <code>`time_zone`</code> 的默认值是 <code>`System`</code> | <pre>```sql create table t (ts timestamp, dt datetime); ```</pre> |

| | | |
|------|--|---|
| 显示效果 | <code>time_zone</code> 的默认值是 <code>System</code> | <pre>create table t (ts timestamp, dt datetime); OK - 0 rows affected (0.02 sec)</pre> |
| 适用情况 | <p>以下元素需要使用单个反引号包裹：</p> <ul style="list-style-type: none"> • 变量名 • 变量的值 • SQL 语法名称，如 <code>`SELECT`</code> • 单行 SQL 语句 • 配置项名 • 字段名 • 其他需要强调的名称，如文件名、表名、列名、库名、数据类型名称等 | <p>以下元素需要使用三个反引号包裹：</p> <ul style="list-style-type: none"> • 多行代码块（多行 SQL 语句、多行 shell 命令等） • 代码执行结果 • 配置文件中的多行配置 |

6. 名称与命名

本章对 PingCAP 技术文档的相关名称与命名作出了统一规范。

6.1 文件命名

本节主要介绍 PingCAP 技术文档的 Markdown 文件命名规范，如下所示。

- 文件名应该能对文档内容进行简要概括。如：`destroy-tidb-cluster.md`
- 文件名不宜过长，由一个或多个英文单词组成。
- 除 `TOC.md`、`CONTRIBUTING.md` 和 `README.md` 等文件外，其他文件名必须统一使用**全小写字母**。同一文件名中禁止大小写混用。
- 当文件名由多个英文单词组成时，单词中间应当由**短划线“-”**隔开。
- 不要在文件名中使用下划线“`_`”。在 URL 中出现下划线不利于搜索引擎优化，且下划线在部分 Markdown 实现中是修饰字符。
- 文件后缀名必须使用小写形式。
- Markdown 文件必须使用“`.md`”后缀。

6.2 产品命名

PingCAP 公司有多产品，其研发人员分布在 TiDB、TiKV、Cloud、Tools 等多个部门。在命名产品时，应遵循一定的风格规范，不可随意取名。

以下总结了一些 PingCAP 现有的中文产品名称。

| 产品名称 | 相关文档 | 常见错误名称 |
|------|------|--------|
|------|------|--------|

| | | |
|---------------------------|--|----------------|
| DM/TiDB DM/Data Migration | Data Migration Overview | TiDB-DM、dm |
| Binlog/TiDB Binlog | TiDB Binlog Cluster Overview | TiDB-Binlog |
| Lightning/TiDB Lightning | TiDB Lightning Overview | TiDB-Lightning |
| | | |

参考：[《TiDB 周边工具输出名称确认》最新版](#)

6.3 名称使用

对于有既定写法的名称，在技术文档中应准确指称，尤其注意大小写问题。

- 【错误示例】mysql、Linkedin、postgresql
- 【正确示例】MySQL、LinkedIn、PostgreSQL

7. 拼写与语法

技术文档中，应严格保证无任何拼写和语法错误。本章对技术文档的拼写和语法作出了统一规范。

7.1 拼写

中文技术文档主要有以下几种拼写错误，应严格避免。

- 简体中文和繁体中文不能混用。
- 禁止出现中英文错别字。错别字包括错字和别字，两者均不能出现。
 - 错字是将某个字写错，这个错字是人为编造出来的，无任何意义。如将 regulate 写成 regluate。
 - 别字是指写出的字与正确的字形近或音近，但意思却不同。如将 cooperate 写成 corporate，将 authorization 写成 authentication。
- 英文大小写形式不能写错。如不能写成 PingCap 公司或者 mysql 数据库，应该写成 PingCAP 公司和 MySQL 数据库。

7.2 语法

本节列举了中文技术文档中常见的几种语法错误类型，其中提供的相应示例均来源于真实文档写作案例，类似的错误望各位文档作者及审校者着重注意。

- 成分残缺

- 【示例一】会话保持：在应用程序没有提供会话保持的功能下，HAProxy 可以提供该项功能。

【建议】会话保持：在应用程序没有提供会话保持功能的情况下，HAProxy 可以提供该项功能。

- 搭配不当

- 【示例一】HAProxy 是由 Linux 内核的核心贡献者 Willy Tarreau 于 2000 年编写，**并仍然负责**该项目的维护，该在开源社区提供免费和版本迭代。

【存在的问题】“并仍然负责”的主语是 Willy Tarreau，不是 HAProxy。

【建议】HAProxy 是由 Linux 内核的核心贡献者 Willy Tarreau 于 2000 年编写，他现在仍然负责该项目的维护，并在开源社区免费提供版本迭代。

- 用词不当

- 【示例一】表示开始**打开**记录http请求的日志功能

【建议】表示开始**启用**记录 HTTP 请求的日志功能

- 成分多余

- 【示例一】根据官方建议，目前**稳定版本**的 HAProxy 为**稳定版** [2.0 特性](#)。

【解释】官方目前建议使用 HAProxy 稳定版本 2.0，2.0 特性可以参考此链接。

【建议】官方目前建议使用 [HAProxy 稳定版本 2.0](#)。

- 句式杂糅

7.3 文档质量检查工具

一名成熟的文档作者应该配置自己的文档编辑器，保证其能对文档自动进行拼写和语法检查。推荐以下几种对文档自动进行质量检查的工具。

- Grammarly
- Visual Studio Code 插件
 - Code Spell Checker
 - Markdownlint
- [LanguageTool](#)
- [Vale](#)

8. 附录资料及相关说明

针对编写的一系列技术文档，应提供相应的术语表和缩略语表作为附录资料，方便读者查阅。

同时，根据技术文档发布平台的不同，应在阅读文档的醒目处提示读者有关文档的侵权说明、商标说明等信息。（暂不适用）

8.1 术语表和缩略语表

本节提供了从 PingCAP 中文技术文档总结出的术语表和缩略语表。

PingCAP 术语表已于 2019 年 11 月发布至官网：[术语表 | TiDB 官方用户文档](#)（更新中，欢迎提 issue 补充）。

缩略语表链接：待补充。

9. 参考资料

以下提供了部分本指南参考的资料：

[Google Developer Documentation Style Guide](#)

[PingCAP Markdown 编码规范](#)

[Microsoft Docs 参与者指南概述](#)

[中文技术文档的写作规范](#)

[豌豆荚文案风格指南](#)

[LeanCloud 文案风格指南](#)

[Lengoo 简体中文规范指南](#)

[知乎专栏 - 写给大家看的中文排版指南](#)

[Markdown 书写风格指南](#)

[中文文案排版指北](#)

正确使用 Tab 和空格：

- [缩进与对齐——正确地使用Tab和空格](#)
- [使用tab, 还是空格, 小小编程习惯却有巨大的差别](#)
- [写代码时，缩进使用tab 还是空格？ - 知乎](#)

列表：

[写给初学者的 Markdown 教程](#)

[英语技术文档中如何正确使用无序列表和有序列表？](#)

[使用无序列表的 7 项注意](#)

引号：

[为什么在知乎上表引用时，用直角引号（「」）而不是弯引号（“”）？](#)