代码托管(CodeArts Repo)

产品介绍

文档版本 01

发布日期 2025-11-17

华为云计算技术有限公司

商标声明


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云计算技术有限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，华为云计算技术有限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。

华为云计算技术有限公司

地址： 贵州省贵安新区黔中大道交兴功路华为云数据中心 邮编：550029

网址： https://www.huaweicloud.com/

目 录

1 图解代码托管（CodeArts Repo）

2 什么是代码托管（CodeArts Repo）

什么是代码托管（CodeArts Repo）

代码托管（CodeArts Repo）是面向软件开发者的基于Git的在线代码托管服务，是具备安全管控、成员/权限管理、分支保护/合并、在线编辑、统计服务等功能的云端代码仓库，旨在解决软件开发者在跨地域协同、多分支并发、代码版本管理、安全性等方面的问题。

- 在线代码阅读、修改、提交，随时随地开发，不受地域限制。
- 在线分支管理，包含分支新建、切换、合并，实现多分支并行开发，效率高。
- 分支保护，可防止分支被其他人提交或误删。
- IP白名单地域控制和支持HTTPS传输，拦截不合法的代码下载，确保数据传输安全性。
- 支持重置密码，解决用户忘记密码的问题。

为什么选择代码托管（CodeArts Repo）

代码托管（CodeArts Repo）提供高效安全的代码托管服务，确保代码端到端的可追溯。

- 全栈自研，安全无忧。
- 高效代码协同开发。
- 多层级代码质量防护。
- 以代码为中心的研发资产追溯。

3 产品优势

- **统一代码仓平台**
  - MR开发模式: 即合并请求模式，是业界主流的开发模式，以提交MR(PR)为主，类似GitLab MR/GitHub PR工作流。
- **极致安全**

从传输安全、精细化权限管控、安全策略、存储加密、备份恢复、代码安全检测、安全审计等多维度构筑安全防御机制，提供极致韧性和安全的代码托管能力，代码核心资产安全无忧

- **内置规范**

将代码库配置管理、分支开发规范、代码Review规范、Committer工程实践等多种标准规范和实践内置于其中，帮您建立标准、规范、高效的代码开发流程。

- **高效协同**

提供基于Git的多种开发协作模式，同时支持分支开发模式和Fork社交编程开发模式，支持业界Git Flow、集中式工作流和多功能分支工作流等常用分支开发模型，既适合中小企业灵活开发模式，也支持中大型企业的复杂开发协作模式。

- **代码高质量**
  - 多层级、细粒度代码上库质量门禁。
  - 集成代码规范检查、安全检查、代码重复率和圈复杂度检查等自动化检测，保障代码高质量。
  - 多形式代码检视，提升代码质量、传递技术经验。
- **一站式DevSecOps**

与需求管理（ CodeArts Req ）、CI/CD等无缝衔接，提供一站式DevSecOps软件开发工具链。

4 应用场景

异地协同开发

- 场景描述：面向中小企业、孵化中心，协同合作。
- 场景特点：用户群体对开发工作的推进效率，敏捷度要求更高，需要高效的协作管理方式和更低开发成本。面临异地开发协同效率低、代码合并冲突频繁的难题。
- 适用场景：云端代码托管服务，实现协同开发。多分支管理功能和合并请求功能，彻底解决代码合并冲突的难题。

高校教学

- 场景描述：高校教师与学生，学习与授课。
- 场景特点：目前缺少功能完备的研发工具链，搭建研发工具环境耗费大量时间，环境维护耗费精力，现有的研发工具上手慢，学习成本高，不利于教学。
- 适用场景：代码托管服务提供完整的代码托管服务，以及丰富的代码仓库模板，使学生可以迅速上手。

项目外包

- 应用：开发类外包项目，需要多组织协同的项目。
- 场景特点：在外包项目管理中，目前普遍存在代码仓库无精细化权限管控、代码单一分支、提交历史可追述性差等情况。
- 适用场景：代码托管服务提供健全的权限控制功能，多分支的协同开发环境，基于代码提交情况的统计分析。

5 产品功能

代码托管服务支持的主要功能如下。关于各功能支持的地域（Region）信息，可通过控制台查询详情。

个人环境配置

用户在代码托管进行代码开发前，需要完成个人的环境配置。

作为代码托管的基础工具，用户需要先完成本地 Git 环境的搭建。

当用户需要将代码推送到云端仓库或从云端仓库下拉代码时，云端仓库需要验证用户的身份与权限，代码托管提供了SSH 密钥、HTTPS密码、访问令牌和GPG公钥的验证方法，以此保障代码访问的安全性。

在提交代码或创建代码评审（MR）时，系统需要使用提交邮箱来标识提交者，确保每个提交都能准确追溯到具体的提交者，因此建议每位开发者在进行开发前，配置提交邮箱。

当代码仓涉及大文件存储时，用户可配置Git LFS。

有关更多信息，请参阅**环境和个人配置**。

配置仓库设置

代码托管支持用户对项目下或者代码组下的所有仓库设置相同的仓库设置。

如果对项目或者代码组下的仓库进行统一设置，可以提升代码仓信息的批量管理效率，确保项目或者代码组下仓库规则的一致性。

有关更多信息，请参阅**配置项目级仓库设置**、**配置代码组的仓库设置**和**配置仓库设置**。

配置保护分支规则

在软件开发过程中，团队成员频繁地向主分支提交代码，这可能导致代码质量下降和安全问题。

为了确保代码质量和安全性，可以配置保护分支规则，这些规则可以保证分支的安全性，允许开发人员使用合并请求合入代码，同时阻止管理者以外的人推送代码，阻止任何人强行推送到此分支，以及阻止任何人删除此分支。

通过合理设置项目级或者仓库级的保护分支规则，可以有效防止未授权的代码提交，确保代码审查过程的严格执行，从而提高代码质量和安全性。

有关更多信息，请参阅**配置项目级保护分支规则**、**配置代码组的保护分支**和**配置仓库级保护分支规则**。

配置合并请求规则

在软件开发团队中，为了确保代码质量和团队协作效率，通常需要对代码合并请求进行严格的评审。然而，在实际操作中，由于缺乏有效的规则设置，可能会导致代码质量下降、团队协作效率低下等问题。

研发团队可以通过配置合并请求规则来解决这些问题，如果研发的评审问题未全部解决时，不能合入合并请求，单子也不能关闭。

有关更多信息，请参阅**设置项目级合并请求规则**、**配置代码组的合并请求规则**和**配置仓库级合并请求规则**。

配置 E2E 设置

E2E设置是代码托管的一项核心功能，用于关联代码合入与相关工作项，便于追溯管理。

有关更多信息，请参阅**设置项目级E2E设置**、**配置代码组的E2E设置**和**配置仓库级E2E设置**。

配置 Webhook

在现代软件开发流程中，开发团队经常需要将代码托管仓库与CI/CD工具或其他第三方系统进行集成，以实现自动化构建、测试和部署。

然而，手动触发这些流程不仅耗时，还容易出错。配置Webhook可以解决这一问题，通过实现代码托管仓库与第三方系统的无缝衔接，当订阅的分支推送、Tag推送等事件发生时，自动发送 POST 请求触发第三方系统的通知弹窗、构建、镜像更新、部署等操作，减少人工干预，提升开发部署全流程的效率。

代码托管支持用户根据项目实际需求选择订阅的事件类型，还可通过分支过滤正则规则精准定位触发条件，避免无关事件的无效触发，确保资源聚焦于关键操作，满足不同项目的个性化需求。

代码托管提供Token鉴权功能，鉴权信息通过HTTP请求头传输，同时要求用户对URL中的敏感数据自行加解密，双重保障接口调用和数据传输的安全性，降低信息泄露风险。

代码托管的Webhook支持配置自定义的第三方系统URL，且Token类型除预设选项外还可自定义，能够灵活对接各类自建系统或主流 CI/CD 工具，适配不同技术栈和业务场景的集成需求。

有关更多信息，请参阅**配置项目级的Webhook设置**、**配置代码组级的Webhook**和**配置代码仓的Webhook设置**。

新建代码仓库

代码托管提供了不同方式进行新建仓库，目前代码托管服务提供以下几种仓库创建方式：

- 自定义新建代码仓库。当团队启动全新项目，并且需要个性化配置仓库结构和规则时，可以自定义新建代码仓库，此创建方式的灵活性较高。
- 按模板新建代码仓库。团队可基于系统提供的预设模板创建仓库，不需要从零开始搭建基础配置，此创建方式可以保证项目初始化的统一性，减少重复配置工作。
- Fork仓库。当开发者需要在原仓库的基础上进行二次开发和贡献代码，或者学习原项目代码的场景，可以基于已存在的目标仓库创建派生仓库，该方式会复制原仓库的代码、分支和提交历史等，且后续可通过合并请求与原仓库同步更新。
- 迁移代码仓库。团队可以从其他代码托管平台迁移到代码托管，即可保留项目历史数据，当前支持迁移GitHub仓、GitLab仓、自建GitLab仓、Gitee仓、Coding仓、Codeup仓、Bitbucket仓库和Gerrit仓。

有关更多信息，请参阅**管理Repo代码仓库**和**迁移代码与同步仓库**。

Fork 仓库

在软件开发领域，大型项目通常包含多个子项目，开发过程中频繁的代码提交和测试可能导致项目版本仓库（源仓库）的不稳定。Fork仓库提供了一种解决方案，它允许开发者基于源仓库创建一个完全相同的镜像仓库。在镜像仓库中，开发者可以自由地进行开发和测试，而不会对源仓库产生任何影响。

只有当新特性被充分测试并确认无误后，才会通过合并请求将这些修改合并回源仓库。这种模式有效地解决了大型项目开发中的代码管理问题，提高了团队协作效率。通过使用Fork仓库，开发者可以在独立的环境中进行开发，确保源仓库的稳定性和安全性。

有关更多信息，请参**Fork仓库**。

备份仓库

在软件开发过程中，为了确保代码的安全性和可恢复性，开发团队通常需要定期对代码仓库进行备份。然而，当开发团队尝试将仓库备份到华为云的其他区域或本地计算机时，可能会遇到操作复杂、连通性问题等挑战。您可以选择将仓库备份到华为云的其它区域，这本质上是一次导入外部仓库的操作，将一个区域的仓库备份到另一个区域中。

此外，您也可以选择将仓库备份到本地计算机，通过使用HTTPS或SSH两种clone形式生成clone命令，只需粘贴进本地Git客户端并执行即可，但需确保仓库的连通性。

有关更多信息，请参阅**备份仓库**。

管理分支

在软件开发过程中，团队成员经常需要同时处理多个功能开发或缺陷修复任务，这可能导致代码冲突和版本管理混乱。为了解决这一问题，可以使用版本管理工具中的分支功能。

分支是版本管理工具中最常用的一种管理手段，使用分支可以把项目开发中的几项工作彼此隔离开来使其互不影响，当需要发布版本之前再通过分支合并将其进行整合。

在代码托管服务的Git仓库创建之初，通常会默认生成一条名为master（生产）的分支，一般作为最新版本分支使用，开发者可以随时手动创建自定义分支以应对实际开发中的个性场景。

团队可以通过合理规划分支策略，如master（生产）、develop（开发）、feature（功能）、release（发布）和hotfix（紧急修复）分支，可以确保每个开发任务独立进行，减少代码冲突，提高开发效率。

有关更多信息，请参阅**开发协作工作流**和**管理分支**。

管理 Tag

在软件开发过程中，团队成员经常需要回溯到特定的版本进行问题排查或功能回滚。然而，由于每次提交（commit）的ID是一长串编码，相对于我们熟知的“V1.0.0”这样的版本号，Commit ID不便于记忆，同时也不具备可识别性，这导致了在需要快速定位特定版本时的困难。为帮助团队成员能够更高效地进行版本管理和问题排查，Tag是Git提供的帮助团队进行版本管理的工具，用户可以使用Git Tag标记提交，从而将项目中的重要的版本管理起来，以便日后精确检索历史版本。

Tag会指向一个commit，无论后续版本怎么变化，它永远指向这个commit不会变化，相当于一个被永远保存的版本快照（只有手动删除时才会被剔除版本库）。

通过给重要的版本打上Tag，给它一个相对友好的名字（比如“ myTag_V1.0.0 ”、“首个商业化版本”），团队成员可以更容易记住和追述这些版本，从而提高工作效率。

有关更多信息，请参阅**管理Tag**。

仓库网络

在软件开发过程中，开发团队经常需要查看代码的提交历史，以了解代码的变更情况。然而，传统的文件页签中的历史记录仅能展示单个文件的变更，无法直观地展现不同提交之间的关系，这给团队成员理解代码变更的全貌带来了困难。为帮助开发更高效地进行代码审查和协作，代码托管提供了“仓库网络”，以流向图的形式展现了某条分支或Tag的整个提交（commit）历史（包括动作、时间、提交者、提交系统生成备注和手动填写备注）以及提交历史的关系。

相对于文件页签中的历史而言，提交网络具备展现提交之间关系的优势，帮助团队成员更直观地理解代码变更的全貌，提高协作效率。

有关更多信息，请参阅**查看仓库网络**。

版本管理

在软件开发过程中，团队成员经常需要对同一项目进行不同的修改和优化，这可能导致代码版本混乱，影响开发效率。为了解决这一问题，CodeArts Repo提供了版本管理功能，该功能可以记录、跟踪、维护和控制产品或系统系列的变更情况，包括但不限于分支、Tag、差异对比等。

通过使用版本管理，团队可以有效地管理代码变更，提高协作效率，确保项目的顺利进行。

有关更多信息，请参阅**管理代码文件**。

代码文件的差异对比

在软件开发过程中，团队成员经常需要对代码的不同版本进行比较，以确保代码质量和团队协作效率。然而，手动对比代码版本耗时且容易出错。“差异对比”可以帮助开发者直观地查看到各个代码版本之间的差异，是本服务提供的版本管理手段之一。

在仓库控制台中，可以对任意的分支、Tag、某次提交进行代码差异的对比，对比出的差异包括变更、新增、删除三个种类，不同种类的差异会以不同样式进行显示以方便辨识。

通过使用差异对比功能，开发者可以快速识别代码变更，提高代码审查和合并的效率。

有关更多信息，请参阅**管理仓库文件**、**在控制台管理Tag**。

Cherry-Pick

在软件开发过程中，团队成员经常需要从其他分支中引入特定的修复或功能，但又不希望引入该分支上的所有更改。当需要在本地合入其他分支的提交时，如果采用合并整个分支的方式，可能会引入不必要的代码变更，导致代码库混乱。用户可以使用git cherry-pick命令，该命令允许开发者将某一次提交取出，并将它覆盖到某个版本上，从而实现精准的代码变更引入。

有关更多信息，请参阅**历史页签：查看分支或Tag版本的提交历史**。

克隆或者下载代码仓到本地

在软件开发过程中，开发人员经常需要在本地计算机上进行代码修改，以便利用本地开发环境的便利性和灵活性。然而，当开发人员需要将本地修改的代码同步到云端代码托管服务时，可能会遇到同步过程复杂、效率低下的问题。云端代码版本和本地代码版本可以通过上传、克隆或者下载的操作在彼此间进行同步，代码托管提供了多种克隆或者下载方式，包括但不限于使用Git命令行工具、图形界面客户端等，以满足不同用户的需求，简化同步流程，提高开发效率。

有关更多信息，请参阅**克隆/下载代码仓库到本地**。

在线开发代码

在软件开发过程中，开发人员经常需要在不同的设备上进行代码编辑，但传统的开发环境限制了这一需求，导致代码修改不及时，影响开发效率。代码托管服务内置了IDE Online功能，支持用户在线编辑代码，用户可以在控制台修改代码，提升问题修复速度和操作的灵活性。

有关更多信息，请参阅**在Repo编辑并创建合并请求**。

解决代码冲突

在多人团队协作开发中，当多个开发者同时修改同一个文件时，可能会遇到代码提交冲突的问题，导致代码推送失败。代码托管支持解决代码冲突，保障团队开发的高效性。解决方法包括：

- 用户将远程代码仓库拉取到本地仓库的工作区，利用Git自动合并可以合并的修改，并手动解决冲突部分，然后通过add > commit > push的方式，再次提交代码。
- 用户也可以利用代码托管服务提供的在线冲突解决工具，直接在线解决冲突。

有关更多信息，请参阅**解决合并请求的代码冲突**。

关联工作项

在软件开发过程中，团队成员需要频繁地提交代码以实现新功能或修复bug，但如何确保这些提交与具体的工作任务准确关联，成为了项目管理中的一个挑战。代码托管服

务可以将每一次代码提交（commit）关联到需求管理的工作项中，帮助开发者精确记录每一次修复bug、提交新特性时所对应的工作任务。

关联工作项还可以帮助项目管理者查看每一个需求、bug修复时，所涉及修改内容的提交人、提交内容等信息，从而提高项目透明度和管理效率。

在代码托管控制台中，对文件的任何操作在保存时都会要求必须填写一个提交信息才能保存，可以理解为控制台的每一次保存都是一次commit操作，其必填的提交信息对应了commit命令的-m内容，代码托管服务从-m（提交信息）中捕获关键字的方式来自动关联工作项，解决了代码提交与工作任务脱节的问题。

有关更多信息，请参阅**可集成系统**。

查看&评论提交记录

在团队协作开发软件时，开发人员经常需要回顾代码提交记录以追踪问题或理解代码变更的原因。然而，当团队成员众多且提交频繁时，找到特定的提交记录和理解其背景信息变得困难。代码托管服务支持查看提交历史的详细信息以及其涉及的文件变更。

用户可以在仓库的提交网络、仓库文件列表的历史页签中，查看提交历史的清单，单击某次提交历史可以进入查看此次提交提交人、提交号、父节点、此条提交下评论的数量、代码变更对。可以对提交内容进行评论，也可对评论内容进行跟帖。这不仅帮助团队成员快速定位问题，还促进了团队内部的沟通与协作。

有关更多信息，请参阅**在仓库动态页查看提交历史**。

配置消息通知

在使用代码仓库进行项目协作时，团队成员需要及时了解仓库的状态变化，以确保项目的顺利进行。然而，当仓库出现欠费冻结、被手动关闭或因冻结时间超过保留期而被彻底删除等情况时，团队成员可能无法及时获知，导致项目进度受阻。

代码托管支持通过消息通知，可以在仓库冻结、关闭、合并请求或删除等关键事件发生时，自动向指定的仓库成员发送邮件通知，确保信息的及时传达，帮助团队成员快速响应仓库状态变化，保障项目顺利进行。

有关更多信息，请参阅**设置Repo的仓库和合并请求通知**。

成员管理

在企业级项目管理中，为了确保代码仓库的安全性和协作效率，成员管理成为了一个关键环节。然而，当项目成员需要跨租户协作时，可能会遇到权限管理和成员邀请的复杂性问题，导致协作效率低下。

代码托管通过明确仓库成员来源于其所属项目的项目成员，并允许项目创建者所在租户邀请其他租户下的IAM账号加入项目，同时规定只有仓库创建者（所有者）和仓库管理员才能对仓库人员进行变动，其他人员仅能浏览仓库成员列表，可以跨租户成员管理，提高协作效率。

有关更多信息，请参阅**管理Repo成员权限**。

配置 IP 白名单

在企业日常运营中，为了保护敏感数据不被未授权访问，通常需要对仓库访问进行严格控制。然而，如果没有配置IP白名单，任何IP地址都可能尝试访问仓库，这将带来潜在的安全风险。

通过设置项目或者仓库的IP白名单，可以确保只有特定的IP地址能够访问仓库，从而提高安全性。

只有在白名单范围内的IP地址发起的访问才会被允许，其他所有IP地址的访问请求将被拒绝，从而大大增强了仓库的安全性。

有关更多信息，请参阅**设置IP白名单**。

风险操作

代码托管支持管理员更改代码仓库所有者等操作，但这些操作存在风险，请谨慎操作。

有关更多信息，请参阅**管理代码文件**。

审计日志

代码托管通过使用支持多维度查询的审计日志系统，可以根据操作时间、操作人、具体行为，以及行为所影响的分支、Tag等信息进行快速查询，有效提升问题排查效率和代码管理的安全性。

有关更多信息，请参阅**审计日志**。

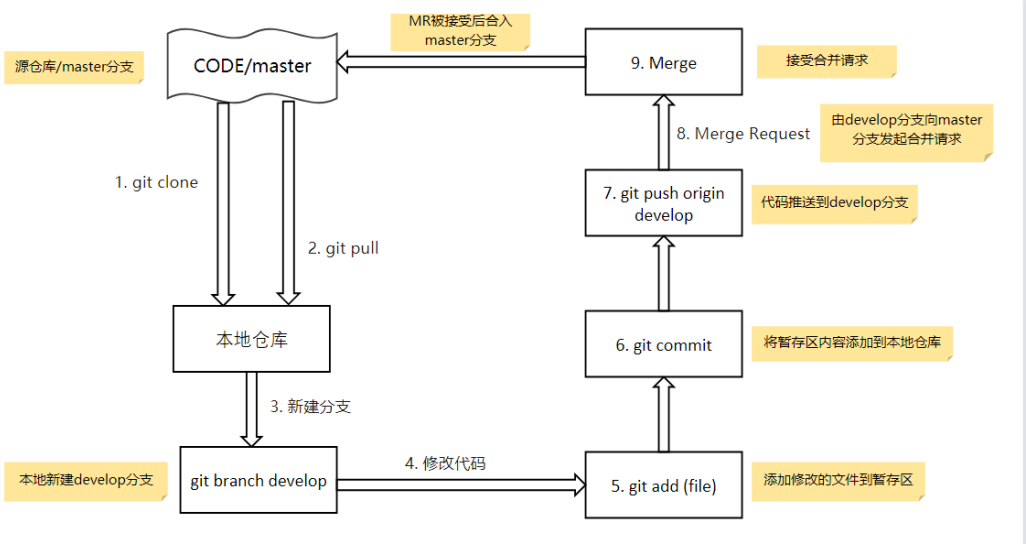
6 原理介绍

6.1 作业流原理介绍

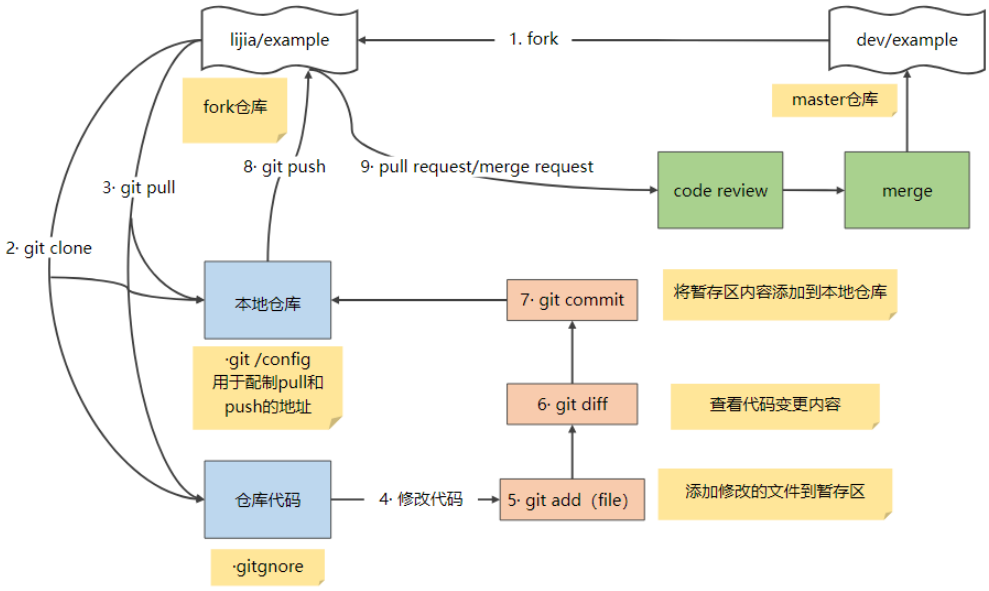
作业流（Workflow）是对作业流程及其各操作步骤之间业务规则的抽象、概括描述。作业流提供了一种很好的工程化的方式来解决业务问题，使得业务抽象、流程格式化、易维护和易拓展，实现一定程度的业务可视化。

下面将介绍两种开发模式的作业流。

- **分支开发模式：**是采用直接clone源项目中心仓的方式，由新建分支向目标分支发起合并请求提交代码，让任何一个开发者都可以方便地向开源项目贡献代码。该模式没有代码评审的机制，开发者之间的协作与交流少而不顺畅，因此多适用于小团队开发。对于较大的团队，建议使用fork开发模式。



- **Fork开发模式：**是一种社交编程，是利用群体的智慧来进行合作编程的一种工作模式，采用派生/合并请求的方式，让任何一个开发者都可以方便地向开源项目贡献代码。这也是一种优秀的代码评审机制，使开发者之间的交流与协作更加顺畅而灵活，开发工作更加高效。



6.2 代码存储原理介绍

CodeArts Repo使用华为云弹性文件服务SFS进行代码数据存储，在CodeArts Repo创建文件时，系统会默认选择开启加密功能。

加密文件系统使用的是密钥管理服务（KMS）提供的密钥，CodeArts Repo会利用弹性文件服务SFS的云原生能力自行构建和维护密钥管理基础设施，安全便捷。

CodeArts Repo在写入或读取弹性文件服务SFS中的代码仓库数据时，SFS存储会从KMS获取加解密密钥（AES-256加密算法），自动将数据进行加密或解密操作。

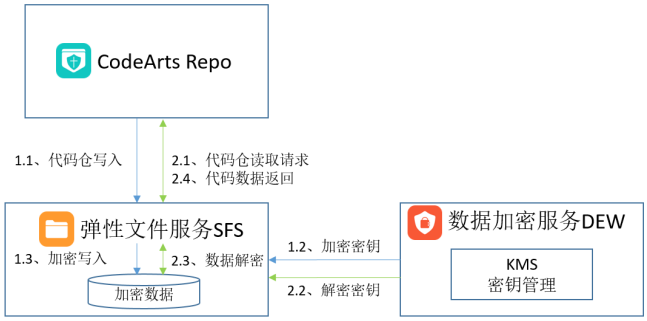
如下图所示，代码数据的加密写入流程为“写入代码仓>获取加密密钥>加密写入”“”，代码数据的加密读取流程“代码仓读取请求>解密密钥>数据解密>代码数据返回”。

其中，代码数据加密写入的各个环节工作方式如下：

- 代码仓写入：用户提交推送代码数据时，CodeArts Repo Server会将用户代码数据写入到SFS弹性文件云服务。
- 获取加密密钥：SFS从KMS读取存储的加密密钥，用于对代码数据进行加密。
- 加密写入：使用加密密钥，对代码数据进行加密并将加密后的数据存储到SFS上。

代码数据加密读取的各个环节工作方式如下：

- 代码仓读取请求：由用户发起下载代码仓的请求。
- 解密密钥：SFS从KMS读取存储的解密密钥，用于对代码数据进行解密。
- 数据解密：使用解密密钥，对代码数据进行解密。
- 代码数据返回：将解密后的代码数据返回给用户。



7 安全

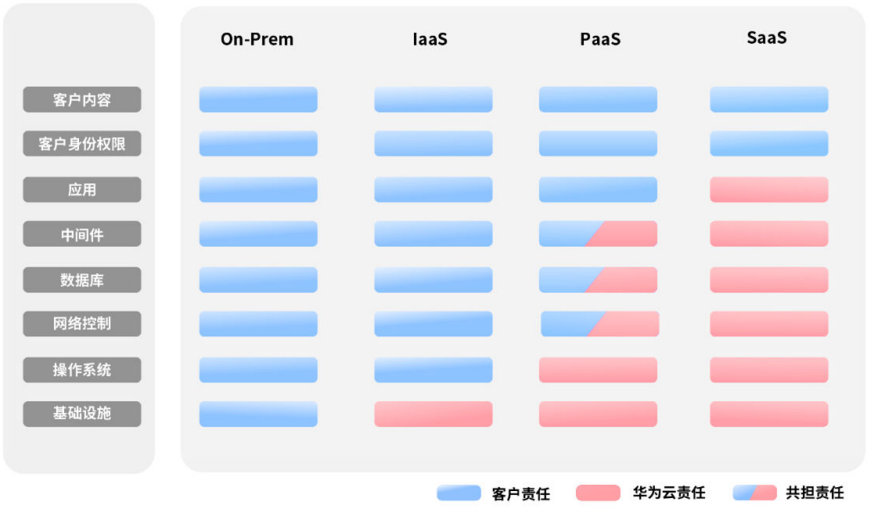
7.1 责任共担

华为云秉承“将公司对网络和业务安全性保障的责任置于公司的商业利益之上”。针对层出不穷的云安全挑战和无孔不入的云安全威胁与攻击，华为云在遵从法律法规业界标准的基础上，以安全生态圈为护城河，依托华为独有的软硬件优势，构建面向不同区域和行业的完善云服务安全保障体系。

与传统的本地数据中心相比，云计算的运营方和使用方分离，提供了更好的灵活性和控制力，有效降低了客户的运营负担。正因如此，云的安全性无法由一方完全承担，云安全工作需要华为云与您共同努力，如图7-1所示。

- **华为云**：无论在任何云服务类别下，华为云都会承担基础设施的安全责任，包括安全性、合规性。该基础设施由华为云提供的物理数据中心（计算、存储、网络等）、虚拟化平台及云服务组成。在PaaS、SaaS场景下，华为云也会基于控制原则承担所提供服务或组件的安全配置、漏洞修复、安全防护和入侵检测等职责。
- **客户**：无论在任何云服务类别下，客户数据资产的所有权和控制权都不会转移。在未经授权的情况，华为云承诺不触碰客户数据，客户的内容数据、身份和权限都需要客户自身看护，这包括确保云上内容的合法合规，使用安全的凭证（如强口令、多因子认证）并妥善管理，同时监控内容安全事件和账号异常行为并及时响应。

图 7-1 华为云安全责任共担模型



云安全责任基于控制权，以可见、可用作为前提。在客户上云的过程中，资产（例如设备、硬件、软件、介质、虚拟机、操作系统、数据等）由客户完全控制向客户与华为云共同控制转变，这也就意味着客户需要承担的责任取决于客户所选取的云服务。如图7-1所示，客户可以基于自身的业务需求选择不同的云服务类别（例如IaaS、PaaS、SaaS服务）。不同的云服务类别中，每个组件的控制权不同，这也导致了华为云与客户的责任关系不同。

- 在On-prem场景下，由于客户享有对硬件、软件和数据等资产的全部控制权，因此客户应当对所有组件的安全性负责。
- 在IaaS场景下，客户控制着除基础设施外的所有组件，因此客户需要做好除基础设施外的所有组件的安全工作，例如应用自身的合法合规性、开发设计安全，以及相关组件（如中间件、数据库和操作系统）的漏洞修复、配置安全、安全防护方案等。
- 在PaaS场景下，客户除了对自身部署的应用负责，也要做好PaaS服务中间件、数据库、网络控制的安全配置和策略工作。
- 在SaaS场景下，客户对客户内容、账号和权限具有控制权，客户需要做好自身内容的保护以及合法合规、账号和权限的配置和保护等。

传统本地部署(On-Prem)：由客户在自有数据中心内部署和管理软件及IT基础设施，而非依赖于远程的云服务提供商；

基础设施即服务(IaaS)：由云服务提供商提供计算、网络、存储等基础设施服务，如**弹性云服务器**、**虚拟专用网络**、**对象存储服务**；

平台即服务(PaaS)：由云服务提供商提供应用程序开发和部署所需要的平台，客户无需维护底层基础设施，如**AI开发平台ModelArts**、**云数据库 GaussDB**；

软件即服务 (SaaS)：由云服务提供商提供完整应用软件，客户直接应用软件而无需安装、维护应用软件及底层平台和基础设施，如**华为云会议**。

7.2 身份认证与访问控制

身份认证

无论通过管理控制台或API接口访问CodeArts Repo，CodeArts Repo使用统一身份认证服务IAM进行认证鉴权。

CodeArts Repo支持两种认证方式：

- **Token认证：**通过Token认证调用请求。
- **AK/SK认证：**通过AK（Access Key ID）/SK（Secret Access Key）加密调用请求。推荐使用AK/SK认证，其安全性比Token认证要高。

访问控制

1. **IAM权限管理**

   权限管理是基于角色与权限的细粒度授权，即根据不同角色的工作需要分配不同的操作权限，用户只可访问被授权资源。

   CodeArts Repo中的角色有产品经理、测试经理、运维经理、系统工程师、Committer、开发人员、测试人员、参与者、浏览者和自定义角色。

2. **IP白名单控制**
   - IP白名单是对IP范围开设的白名单，通过设置IP白名单能极大增强您的仓库的安全性。
   - 只有在IP白名单范围内的IP才可以访问仓库。除此之外其他IP发起的访问将被拒绝。
   - IP白名单包括租户级IP白名单和仓库级IP白名单，并可配置优先级。

   关于IP白名单的详细配置方法，请参见**设置IP白名单**。

3. **锁定仓库**

   为防止任何人破坏即将发布版本的代码仓库，管理员可以锁定仓库，在锁定仓库后，任何人都无法向任何分支提交代码（包括管理员本人）。

   关于锁定仓库的详细操作方法，请参见**锁定仓库**。

4. **保护分支管理**

   分支保护，可防止分支被其他人提交或误删。
   - 保证分支的安全性，允许开发人员使用合并请求合入代码。
   - 阻止管理者以外的人推送代码。
   - 阻止任何人强行推送到此分支。
   - 阻止任何人删除此分支。

   有关更多信息，请参阅**配置项目级保护分支规则**、**配置代码组的保护分支**和**配置仓库级保护分支规则**。

5. **运维SOD**

   为规范开发、测试、发布上线全流程运维脚本（包含脚本开发、代码检视、手动测试、集成验收、发布审核、脚本上线、版本管理等），推行和加强标准化作业的管理，保证流程合规、安全合规、质量合规。

6. **防护墙和VPC隔离**

   CodeArts Repo通过防护墙和VPC隔离支持租户间网络和资源隔离。

7.3 数据保护技术

CodeArts Repo通过多种手段保护数据安全。

数据保护手段	简要说明	详细介绍
传输加密（HTTPS）	通过在云端对托管在CodeArts Repo的代码库进行落盘加密，可以有效避免数据拥有者之外的人接触到用户的明文数据，避免数据在云端发生泄露。同时，代码加密过程对用户完全透明，用户可以使用任意官方Git端来访问CodeArts Repo上的代码仓库。	关于HTTPS密码的详细介绍及获取方式，请参考**配置HTTPS密码**
密钥管理	通过SSH密钥和部署密钥管理，确保请求发起是请求发起方，让用户只能浏览被授权的数据，保证数据安全。	关于SSH密钥详细介绍及获取方式，请参阅**配置SSH密钥**。
git-crypt加密传输与存储	git-crypt是一款第三方开源软件，可以用于对Git仓库中的文件进行透明化的加密和解密。	其可对指定文件、指定文件类型等进行加密存储，开发者可以将加密文件（如机密信息或敏感数据）与可共享的代码存储在同一个仓库中，并如同普通仓库一样被拉取和推送，只有持有对应文件密钥的人才能查看到加密文件的内容，但并不会限制参与者对非加密文件读写。 关于git-crypt加密传输与存储详细介绍及使用方式，请参见**在Git客户端使用git-crypt传输敏感数据**。
敏感数据匿名和高价值数据加密	CodeArts Repo在利用统一、准确的数据支撑应用程序和服务的同时充分保障了数据安全性和隐私性。	日志和数据库中无可避免有一些敏感数据，包含但不限于密钥，账号信息等等。为防止敏感数据泄露造成安全问题，CodeArts Repo先把这些数据进行匿名或者加密处理，其原理是哈希函数，是对一段信息产生信息摘要，以防止被篡改。

数据保护手段	简要说明	详细介绍
防DDOS工具	DDoS高防（Anti-DDoS）是防护DDoS攻击的工具。当您的互联网服务器遭受大流量的DDos攻击时，DDoS高防可以保护其应用服务持续可用。	DDoS高防支持通过DNS解析和IP直接指向两种引流方式，实现网站域名和业务端口的接入防护。根据您在DDoS高防中为业务配置的转发规则，DDoS高防将业务的DNS域名解析或业务IP指向DDoS高防实例IP或CNAME地址进行引流。 来自公网的访问流量都将优先经过高防机房，恶意攻击流量将在高防流量清洗中心进行清洗过滤，正常的访问流量通过端口协议转发的方式返回给源站服务器，从而保障源站服务器的稳定访问。
流量限制	流量限制可以用来限制用户在给定时间内HTTP请求的数量，流量限制用来保护上游应用服务器不被同时太多用户请求所压垮。	CodeArts Repo的主要使用Nginx流控和APIGW流控。Nginx的流量限制使用漏桶算法，该算法在通讯和分组交换计算机网络中广泛使用，用以处理带宽有限时的突发情况。APIGW流控可限制单位时间内API的被调用次数，保护后端服务，提供持续稳定的服务。
容灾备份	容灾备份不仅保证数据不丢失，还要保证在服务器宕机后接管服务器的业务，保证业务连续性。保障用户可以不间断的使用应用服务，让用户的服务请求能够持续运行，保证信息系统提供的服务完整、可靠、一致。	-
Hash分片存储	Hash分片存储，即通过数据分片提高隐私性和私密性，就是按照一定的规则，将数据集划分成相互独立正交的数据子集。然后数据被随机分散到多个节点中，没有任何一个节点可以访问完整的数据，它们只包含数据的某一部分。	-
水印	为防止未经授权拍照、截图或其他手段随意传播公司核心资产，可以开启水印设置。	关于水印的详细介绍和使用方式，请参见**给仓库添加水印设置**。
备份	仓库备份操作保障代码安全，防止他人误删除，分为两种备份形式。 • 将仓库备份到华为云的其它区域。 • 将仓库备份到您本地计算机。	关于备份仓库的详细操作方法，请参阅**备份仓库**。

7.4 审计与日志

审计

云审计服务（Cloud Trace Service，CTS），提供对各种云资源操作记录的收集、存储和查询功能，可用于支撑安全分析、合规审计、资源跟踪和问题定位等常见应用场景。

用户开通云审计服务并创建和配置追踪器后，CTS可记录CodeArts Repo的管理事件和数据事件用于审计。

CTS的在CodeArts Repo的审计详情，请参阅**CTS审计日志**

日志

云日志服务（Log Tank Service）提供一站式日志采集、秒级搜索、海量存储、结构化处理、转储和可视化图表等功能，满足应用运维、网络日志可视化分析等保合规和运营分析等应用场景。

出于分析问题的目的，CodeArts Repo将系统运行的日志实时记录到LTS，并保存3天。

基于服务器、数据库等的日志进行监控，对触发监控规则的日志信息通过短信和邮件进行告警，确保现网故障和隐患能第一时间被发现并进行有效处理，保证用户的业务正常运转，做到问题的及时发现和处理，减少对用户业务的影响。

7.5 监控安全风险

WAF 应用防护系统

CodeArts Repo对接WAF应用防护系统。Web应用防护系统也称为网站应用级入侵防御系统。

WAF通过对HTTP(S)请求进行检测，识别并阻断SQL注入、跨站脚本攻击、网页木马上传、命令/代码注入、文件包含、敏感文件访问、第三方应用漏洞攻击、CC攻击、恶意爬虫扫描、跨站请求伪造等攻击，保护Web服务安全稳定。

WAF支持云模式、独享模式和ELB模式三种部署模式。

主机围栏

主机围栏可分别对PC设备接入设置、IP地址列表、PC设备标识列表进行安全围栏设置。

OS 加固和异常侦测

OS规范化脚本分为两个脚本，检查脚本（osstdchk.py）和修复脚本（osstdfix.py）。OS加固需根据华为云OS加固标准进行加固。

7.6 安全运维

变更作业流程

通过脚本在平台进行现网变更，避免在服务器控制台直接操作引发现网故障，并且执行平台操作需符合1+1检视流程，一人实施，另外一人监控和检查，保证流程合规、安全合规、质量合规。

提权操作的控制

依据风险分层分级和权限SOD原则，对权限以及授权过程进行控制。当遇到普通业务告警，需遵循高危和黑名单命令控制，即当进行变更操作时可对其中的命令进行实时监控，并且可通过配置规则对命令危险程度进行等级划分，若检测出高危和黑名单命令，系统会提供实时告警通知，避免违规操作造成业务中断。当遇到紧急业务告警时，提权需遵守规定，确保安全与效率的均衡。

变更操作的审视

变更实施前需进行申请、风险审核、专家组评估等流程。

变更实施过程中的每一步操作都必须检查、验证及监控业务情况，检查范围包括变更服务、周边服务及全局的监控告警、拨测及流量变化等，避免出现人为变更导致现网故障。

7.7 认证证书

合规证书

华为云服务及平台通过了多项国内外权威机构（ISO/SOC/PCI等）的安全合规认证，用户可自行**申请下载**合规资质证书。

图 7-2 合规证书下载

资源中心

华为云还提供以下资源来帮助用户满足合规性要求，具体请查看**资源中心**。

图 7-3 资源中心

合规资质证书

华为云安全服务提供了网络安全专用产品安全检测证书、软件著作权等证书，供用户下载和参考。具体请查看**合规资质证书**。

图 7-4 网络安全专用产品安全检测证书&软件著作权证书

8 约束与限制

本节介绍了代码托管中的限制，如下表所示。

表 8-1 使用限制说明

指标类型	指标项	体验版	基础版	专业版	企业版
单个仓库规格	单仓大小（含LFS）	<=1GB	<=10GB	<=20GB	<=30GB
	单文件上传大小限制（页面）	<=50MB	<=50MB	<=50MB	<=50MB
	单文件推送大小限制（本地）	<=200MB	<=200MB	<=300MB	<=300MB
	LFS单文件大小限制	<=1GB	<=1GB	<=2GB	<=2GB
	在线修改代码，单次保存行数限制	<=5000行	<=5000行	<=5000行	<=5000行
仓库总容量规格	仓库容量（含LFS，超出容量限制会导致仓库部分功能无法使用，如代码无法上传）	<=10GB	<=50GB	<=100GB	<=500GB
仓库数量规格	仓库数量	不限制	不限制	不限制	不限制
浏览器	类型	目前适配的主流浏览器类型包括： • Chrome（推荐） • IE10以上 • Edge（推荐） • Firefox • Safari			
分辨率	分辨率大小	推荐使用1920*1080及以上。			

当仓库容量超出限制、欠费冻结、违规冻结时，仓库部分功能不可执行。具体如表8-2。

当出现公安冻结，未实名冻结时，您仅具有查看权限，不具备操作权限。具体如表8-2。

表 8-2 功能约束限制

页签	功能	仓库容量超出限制、欠费冻结、违规冻结	公安冻结、未实名认证冻结
租户首页	新建仓库	×	×
仓库首页	• 关联工作项 • 成员管理 • 删除仓库	√	×
代码	• 新建/编辑/删除/重命名/上传文件 • 新建/删除目录 • 新建/删除子模块 • Cherry-Pick，revert文件	×	×
代码	新增/删除/编辑/回复/解决检视意见和评论	√	×
分支&tag	• 新建分支 • 分支合并 • 新建tag	×	×
分支&tag	• 编辑/删除分支 • 设置保护分支 • 删除tag	√	×
合并请求	• 新建/编辑/关闭/重开/合并合并请求 • Cherry-Pick，revert合并请求 • 合并请求解决代码冲突	×	×
合并请求	新增/删除/编辑/回复/解决检视意见	√	×
成员	新增/删除/编辑/审核成员	√	×
仓库	Fork仓库	×	×

页签	功能	仓库容量超出限制、欠费冻结、违规冻结	公安冻结、未实名认证冻结
设置	• 仓库设置 • 子模块设置 • 部署密钥同步功能 • 仓库加速 • 策略设置（全部） • 服务集成（全部） • 同步设置 • 同步仓库	√	×
设置	• 仓库信息 • 通知设置 • 仓库加速 • 仓库备份 • 合并请求模板 • 检视意见模板 • 部署密钥 • 租户和仓库级IP白名单 • 风险操作 • 水印设置 • 锁定仓库 • 审计日志 • 租户级用量管理	√	× **说明** 除了仓库备份、租户级用量管理外，其余所有设置项只能查看不能操作。

说明

当**关闭代码托管服务**，您不可进入仓库，界面并提示需开通服务，开启代码托管服务后，恢复仓库状态。关闭代码托管服务超过30天，系统自动删除仓库，不可恢复。

9 基本概念

- 项目管理员

  项目管理员，通常项目创建者默认为本项目的项目管理员。

  项目管理员拥有在本项目下的所有权限，且权限不得被移除或修改。

  项目创建者（同时也是项目管理员）可以赋予项目下其他成员进行权限管理的权限。
- 仓库所有者（仓库创建者）

  项目成员被赋予新建仓库的权限，当该成员创建仓库后，会被标记为仓库所有者，同时享有该仓库最大的权限。
- 仓库管理员

  仓库所有者、项目管理员、父代码组所有者均为仓库管理员。
- 代码组管理员

  代码组所有者、项目管理员、父代码组所有者均为代码组管理员。

