

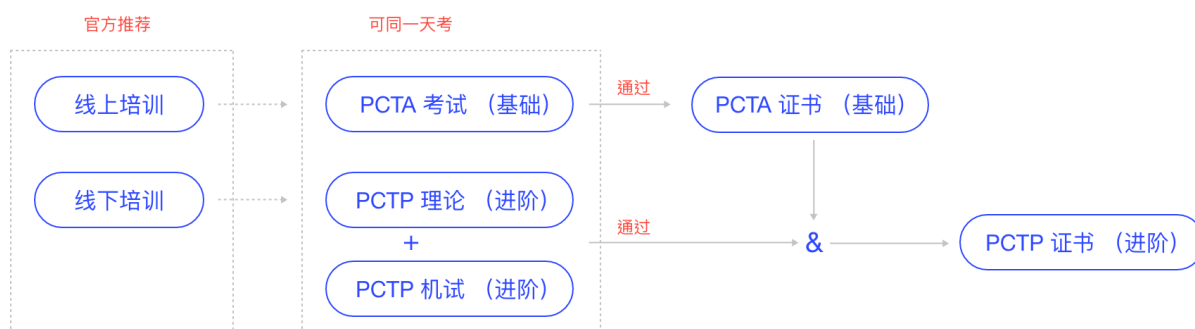
## PingCAP 官方认证 TiDB DBA

笔试+机试 基于 TiDB 3.0

### 顺利通过 TiDB DBA 官方认证考试，我能获得什么？

- 深度理解 TiDB 架构、原理及最佳实践
- 具备独立部署、运维及调优 TiDB 的能力
- 拥有分布式计算和存储领域的技术前瞻视野
- 获得来自 PingCAP 官方的权威技术能力认可

### 学习路径：



您即将参加 TiDB DBA 认证考试，我们希望您顺利通过，所以准备了考试大纲来帮助您全面学习、顺利备考。当您深刻理解以下所涉及的内容后，我们相信您已经为考试做好充分准备。预祝您顺利通过考试。

### 考试对象：

即将或已经从事 DBA、架构师等职位的人员，希望掌握和使用 TiDB，并以此谋求更高职业发展。

## 考核目标与要求：

**PCTA (PingCAP Certified TiDB Associate)** 是 PingCAP 公司认证 TiDB 数据库专员的缩写。PCTA 要求具备安装部署及日常运维分布式关系型数据库的能力。

PCTA 需要学习并熟练掌握 TiDB 架构原理、安装部署、周边工具等基础知识。

**PCTP (PingCAP Certified TiDB Professional)** 是 PingCAP 公司认证 TiDB 数据库专家的缩写。PCTP 要求具备管理大型分布式关系型数据库集群的能力。PCTP 需要学习并熟练掌握 TiDB 的深度原理及高级 Feature、性能调优、SQL 优化、Trouble Shooting 等进阶内容。要成为 PCTP 必须先获得 PCTA 认证。

## 考试规则：

- 1、考试期间不得交头接耳，互相串通答案等；理论考试需全程闭卷进行，手机、Pad、电脑等电子设备及书本、教材等纸质资料均不得带入考场，考试中心将提供草稿纸，考生只需携带身份证、准考证及答题用笔；上机考试统一使用考试中心用机和官方指定云环境，可以查阅官方文档、Asktug 和指定资料，谢绝自带资料或查阅其他网站内容，不得通过远程联系或咨询他人，否则视为作弊，将取消此次考试成绩；
- 2、考试时将当众启封考卷，以示公平；
- 3、题型包括单选（有且仅有一个正确选项）、多选（至少有2个正确选项）、不定项选择（有1个或者多个正确选项），每题分值相同，错选、漏选、多选均不得分；
- 4、通过线以答题正确率60%为基准，根据试卷难度系数，略有小幅波动

# PCTA 基础知识篇

## 考试形式和计分规则：

- 1、指定考试中心集中笔试；
- 2、考试时长60分钟，共60道题；

## 考试范围：

### 一、概述

#### 1. 数据库、大数据发展简史

- 数据库发展历史及主要产品
- 大数据发展历史及主要产品
- Spanner、TiDB 的由来及发展
  - 传统 MySQL 分库分表架构演变及遇到的问题
  - 分布式数据库概述
  - 分布式数据库的代表 Spanner、TiDB

#### 2. TiDB 概论、架构简介

- TiDB 整体架构
  - 存储引擎：TiKV 架构
  - 调度引擎：Placement Driver
  - SQL 计算引擎：TiDB Server
  - Spark 计算引擎：Tispark
- TiDB 生态组件矩阵
  - 数据库导入解决方案：Data Migration
  - 数据输出：TiDB Binlog
  - TiDB 备份及恢复工具：
    - Cloud TiDB
    - TiDB Ansible
    - Lightning

### 二、系统安装部署与管理

## 1. TiDB 集群管理概述

- 系统配置
  - 操作系统硬件、软件要求及相关配置
  - TiDB 参数配置
- 安装部署 TiDB Cluster
- TiDB 各文件介绍
- 启动、关闭 TiDB Cluster
- 升级 TiDB
- 单机多实例部署
- 在 TiDB 跑一个简单应用 (Sysbench)

## 2. TiDB 集群管理实操

- 集群安装部署
- 部署 HA Proxy
- 修改配置以及滚动升级 TiDB
- 扩容节点
- 多实例部署

## 三、使用管理手册

### 1. 常见的使用管理

- 用户及权限管理
- 表与系统视图
- 连接管理
- 常见的管理控制命令 (DCL)

### 2. SQL

- Query 处理
- DML 处理
- 支持 DDL 类型

3. 对象管理与维护
  - 常见的表类型
  - 支持的数据类型
  - JSON 的使用
  - 分区表的使用
  - 常见表管理命令
  - 索引 (B+Tree 与 LSM Tree)
  - 表、索引健康度及统计信息
4. TiDB 日志管理
  - TiDB 系统日志格式解读
  - TiKV 系统日志格式解读
  - PD 系统日志格式解读
  - Slow Log 日志格式解读
  - Slow Log 分析视图及工具
5. TiDB 使用实操
  - 创建自己的 Schema
  - TiDB-bench 使用，模拟 OLTP 场景
  - 测试 TPCH，对比 MySQL 的性能表现。

## 四、生态工具实操

1. DM 安装部署实操
  - DM 安装部署
  - DM 配置优化
  - DM 常见问题
  - DM 监控
2. TiDB-Binlog 安装部署实操
  - 部署要求及注意事项
  - 使用 TiDB-Ansible 部署
    - 下载 TiDB-Ansible

- 部署 Pump
  - 部署 Drainer
  - 使用 Binary 部署
    - 下载官方 Binary
    - 部署样例
    - 参数配置说明
3. TiDB-Lightning 安装部署实操
- 环境准备
    - 硬件需求
    - 导出数据
  - 使用 Ansible 部署 TiDB-Lightning
    - 部署 Lightning
    - 启动 tikv-importer
    - 启动 tidb-lightning
  - 手动部署 TiDB-Lightning
    - 下载 TiDB-Lightning 安装包
    - 启动 tikv-importer
    - 启动 tidb-lightning
    - 参数配置说明

## 五、备份恢复

1. 常见备份方案
2. 备份策略
3. 恢复策略和工具
4. 备份及恢复案例

## 六、TiDB 监控与告警

1. TiDB 集群监控

- Prometheus 原理
  - 集群监控架构图
  - 监控数据上传流程
  - 常见的 Dashboard 解读，重点 Metrics 解读
2. TiDB 报警系统
    - 如何配置 Alertmanager
    - 如何接入第三方报警平台
    - 常规报警处理流程
  3. 常见问题处理

## 七、业务开发优化

1. 业务开发最佳实践
  - 事务
  - TiDB 隔离级别详解
  - 显式事务中 DML 语句返回的 affected rows 不可信
  - 避开丢失更新影响的应用开发方法
  - 计数器，秒杀场景的处理方法
  - “嵌套事务”
  - 不支持 Spring 框架的 PROPAGATION\_NESTED 传播行为（依赖 savepoint 机制）
  - 大事务
  - 自增 ID
  - TiDB 中的自增 ID 分配原理
  - 自增 ID 设计最佳实践
  - TiDB 中的自增 ID 使用方法
  - 唯一性约束
  - 主键与唯一索引
  - 外键
  - INSERT 语句默认只在提交时进行唯一性约束校验
  - 索引

- TiDB 中的索引
  - 复合索引的设计
  - 批量计算场景的写入优化
  - 配置 SHARD\_ROW\_ID\_BITS 拆散写入热点
  - 分区表
2. TiDB 案例
  3. TiDB 性能 Benchmark
  4. TiDB 慢 SQL 与调优



# PCTP 高级进阶篇 - 理论笔试

## 考试形式和计分规则：

- 1、指定考试中心集中笔试；
- 2、考试时长75分钟，共70道题；

## 考试范围：

### 八、TiDB 计算内核体系

#### 1. TiDB Server

- TiDB Server 概览
- TiDB Server 的启动、停止
  - 对 TiDB Server 启动有影响的启动参数和它们的作用
  - 怎么知道 TiDB 已经启动成功了？
  - 集群里不同 TiDB 有什么区别？怎么找到 DDL/Stats/GC 的 owner？
- 客户端连接管理
  - 几种常见的连接方式（客户端/JDBC） / （TCP/Unix Socket）
  - SSL 的配置
  - 创建连接时，TiDB 都做了哪些工作？
  - 权限管理
  - Token Limit 的作用
  - TCP Keep Alive 的作用
  - prepare/execute 使用不同协议时的区别
  - 如何关闭连接
- 常用对外端口

#### 2. TiDB Parser

- Parser 基本原理
- Parser 相关问题处理

- 什么样的错误是 Parser 报错？
- Parser 报错信息怎么解读？

### 3. TiDB 关系型模型原理

- TiDB 中表的分类
  - 物理表
  - Information Schema
  - `mysql` database
- 关系型模型到 KV 的映射关系
- Table Partition
  - TiDB Table Partition 的原理
  - 支持的 partition 类型

### 4. DML 原理

- 数据写入原理
- 数据读取原理
- Misc.
  - 几种不同的 Insert 语句语义区别
  - Replace 和 Insert on duplicate 的区别
  - 何时检测冲突？Insert 语句的 lazy check
  - 怎么改写语句让执行更快？

### 5. DDL 原理

- TiDB DDL 概述
- TiDB drop table , truncate table , add column 等操作原理
- TiDB DDL 问题诊断以及工具
- TiDB Add index 操作原理以及调优

## 九、TiKV 存储内核体系

### 1. TiKV 整体架构

## 2. 存储引擎

- RocksDB 介绍及主要特性
- Column family 介绍
- LSM 原理及优缺点
- Compaction 原理
- RocksDB 数据写入流程
- Key Value 分离 Titan 的设计与实现
- RocksDB 主要参数介绍

## 3. Raft 复制

- Raft 介绍及主要特性
- Raft 各个角色及内部机制
- Raft 在 TiKV 中的使用
- TiKV 自动分片机制 (Region)
- Multi - Raft 设计与实现
- Raft 与 Paxos
- TiKV 中的 Raft 读写流程

## 4. Coprocessor

- 算子下推
- 执行引擎模型

## 5. Placement Driver 调度内核体系

- ETCD 原理介绍
- TiKV 元信息及管理
  - 节点, region
  - 元信息上报流程
- PD 调度算法
  - 调度策略解析
  - Region 迁移过程
  - 如果通过打 label 来增加调度策略
  - Region Split、Merge
  - 如何通过 pd-ctl 管理集群

## 十、生态工具原理

- MySQL 数据同步到 TiDB 解决方案
  - DM 设计原理
    - DM 设计原理
    - DM 功能介绍
    - DM 数据同步处理单元设计原理
    - Relay log 实现原理
    - Binlog Replication 同步原理
  - Sharding DDL的实现与 checkpoint 机制
  - 常见错误和异常场景
    - 常见问题排查手段（不兼容 DDL 处理，借助 checkpoint 定位同步状态等）
    - 同步异常定位和处理方法（sharding DDL 阻塞，relay-log 异常，同步连接中断等问题）
- TiDB 数据输出订阅解决方案
  - TiDB-Binlog 实现原理
    - TiDB binlog 概览
    - Pump 设计和实现
    - Drainer 设计和实现
  - Binlog 运维
    - pump/drainer 下线
    - Binlog 日志和监控详解
    - 常见的问题和处理手段
- TiDB 数据物理加载解决方案
  - Background
    - Why TiDB Lightning was created
    - Characteristics of TiDB Lightning
  - Import Process
    - Import mode
    - Core import process (Lightning, Importer)
    - Post-process (Checksum, Analyze, Auto-inc ID)

- Deployment
  - Hardware requirement
  - Configurations
- Extra Features
  - Table filter
  - Checkpoints
- Monitoring
  - Introducing Grafana panels

## 十一、事务

### 1. 数据库事务概述

- 事务概念介绍
- OCC
- 常见主流数据库的事务隔离级别及其行为介绍 (MySQL、Postgres、CockroachDB)
- TiDB 支持的事务隔离级别及其行为介绍
- MySQL Repeatable Read 与 TiDB 的区别
- 两阶段提交

### 2. 分布式事务原理

- Percolator 的原理
  - 简介
  - 写入过程
  - 读取过程
- TiDB 事务原理
  - TiDB 怎么写入
  - TiDB 怎么读取
  - TiDB 如何清理锁

### 3. TiDB 事务处理

- SQL 事务

- 事务启动
- 事务中的DML
- 语句提交和回滚
- 事务提交和回滚
- 常见提交错误
  - 写冲突处理
  - 读写冲突处理
  - 网络超时处理
- 事务重试处理
  - 事务重试问题和实现
  - 重试行为控制
  - 重试相关日志和监控
- select for update 处理
  - TiDB 和其他数据库处理区别
  - 处理 can not retry select for update

## 十二、系统性能优化

### 1. TiDB Server 性能相关原理

- TiDB 常见系统负载
- 读、写后台处理流程及主要性能瓶颈
- 读热点及解决思路
- 写热点及解决思路
- TiDB Server 性能相关监控
- 主要的性能相关参数

### 2. TiKV 性能相关原理

- 写入瓶颈分析和优化
- 数据在 TiKV 中的读取流程及常见瓶颈
- 读取瓶颈分析和优化及常见瓶颈
- RocksDB 性能优化
- RocksDB 主要的性能相关参数

3. 集群系统负载
  - 常见 OS 负载处理
  - 性能定位 trace
4. 常见性能优化 Case study
  - Sysbench 压测演练
  - 读热点定位和处理演练
  - 写热点定位和处理演练
  - 慢查询定位和处理演练
  - 其他瓶颈定位演练

## 十三、SQL 优化

1. 查询优化器原理
  - TiDB Logical Optimization，列举现有的优化规则
  - TiDB Physical Optimization，举例讲解物理优化过程中的 task type, expected row count, data order 等等物理属性
  - TiDB Ranger：Filter 条件如何构造成 Key Ranges
  - TiDB Plan Cache for Prepared statements
2. TiDB Statistics
  - 直方图和 CM-Sketch 详细介绍，Selectivity 估算原理
  - Analyze Table 语句如何更新直方图和 CM-Sketch
  - Query Feedback 工作原理
3. TiDB Execution Engine
  - Chunk 和 TiDB 向量化执行引擎原理介绍
  - TiDB 的表达式计算：向量化原理+SQL Mode Behaviors
4. 表关联算法
  - Hash Join, Merge Join
  - Index Join, NestedLoopApply

## 5. 其他算子

- TiDB 的 Aggregate 算法：Hash Aggregate, Stream Aggregate
- TiDB 的 Readers：IndexReader, TableReader, IndexLookupReader
- TiDB Insert/Update/Delete/Replace implementations
- DistSQL 中 Cop 请求如何发送的，CopIterator 如何工作的
- Elastic OLTP&OLAP, Batch Size Control

## 6. SQL 执行计划

- 如果查看执行计划
- 执行计划解读
- CBO 模型
- 驱动表
- 数据寻址算法
- 索引原理
- 索引使用

## 7. SQL 优化实践

# 十四、高可用及容灾

1. 高可用与思考(误区)
2. 基于 Raft 协议高可用(组件特性)
3. 基于 Binlog 数据备份同步架构
4. 高可用实操



# PCTP 高级进阶篇 - 上机考试

## 考试形式和计分规则：

- 1、指定考试中心集中上机考试，指定用机和云环境；
- 2、考试时长90分钟；
- 3、主要考核针对不同场景进行系统调优、SQL 优化、Trouble Shooting 等操作，强调解决问题的能力；
- 4、考试期间需要将过程命令行及结果截图复制在指定文档上；考试期间对操作步骤进行录屏；

## 考试环境：

- 1、官方考试用机为 UCloud 云环境 8-9台 CPU：2C，内存：4G，磁盘：50G

## 练习环境：

- 1、[点击](#)可获得 UCloud 云环境的折扣（教学专用）

- 2、考试题目举例：

TiKV 损坏场景恢复

场景：模拟多数 TiKV 损坏场景的 TiDB 集群恢复操作，在一套正常运行的集群下，手动删除其中两个 TiKV 部署目录、kill tikv-server 进程，进行灾难恢复

要求：写下具体操作步骤，并将命令返回的结果截图