组织信息同步-开发手册

1. 背景

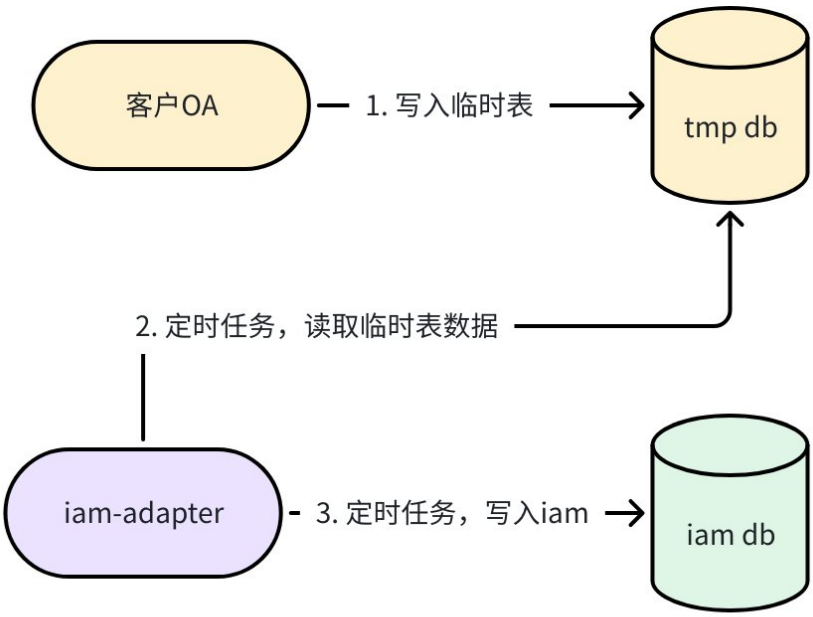
部分客户需要将组织信息同步至HiAgent。我们借助iam-adapter实现组织同步，以此避免修改HiAgent代码以对接非标准化需求。

> ❗ 适用于HiAgent 1.6.0之后的版本，sso 透传用户id在2.0.0版本开始支持

2. 架构图

iam-adapter为可插拔组件，与hiagent架构解耦。下图重点说明SSO流程及组织架构同步流程

2.1 组织架构同步流程



3. 开发指南

3.1 代码开发

可直接参考iam-adapter项目代码：https://code.byted.org/epscp/iam-adapter

外部伙伴代码获取：

iam-adapter-master.zip
335.26KB

注意事项：

1. 上面zip包代码为master分支代码，随迭代可能会有兼容更新，不影响老版本使用
2. 若涉及SSO代码开发，则需要完成开发后。将镜像提供至字节，进行内部镜像漏洞扫描。仅组织架构同步，无代码变更，无需漏扫。

3.1.1 功能特性

- 支持 SSO adapter，提供标准的 CAS 对接模版代码，开发人员可基于模版快速开发 SSO 对接代码
- 具备组织架构同步的能力，包含了临时表->生成表的数据同步定时任务

3.1.2 项目结构

代码块

```
.
├── Makefile              # Makefile文件，封装了项目的构建、运行、测试等命令
├── README.md             # 项目的 README 文件，介绍了项目的基本信息、功能特性、项目结构等
├── build
│   └── Dockerfile        # Dockerfile文件，封装了项目的构建镜像命令
├── cmd
│   ├── api
│   │   └── server.go     # 项目的主入口文件，负责启动项目的服务
│   ├── gen
│   │   └── generate.go   # 项目的代码生成文件，负责生成Mysql ddl model的代码
│   └── main.go           # 项目的主入口文件，负责启动项目的服务
├── common                # 项目的公共模块，包含了项目的公共函数、常量、结构体等
│   ├── tenant.go
├── config                # 项目的配置文件，包含了项目的配置信息、环境变量等
│   ├── adapterconf
│   │   └── Dev.yaml
│   ├── config.go
│   ├── iam.go
│   └── init.go
├── config.yaml
├── ddl                    # 项目的数据库文件，包含了项目的数据库表结构定义
│   └── init.ddl
├── go.mod
```

```
├── go.sum
├── idl                    # 项目的接口定义文件，包含了项目的接口定义和请求参数等
│   ├── base.thrift
│   ├── server.thrift
│   └── sync.thrift
├── manifests              # 项目的部署文件，包含了项目的部署配置和脚本等
│   └── iam-adapter
│       ├── Chart.yaml
│       ├── templates
│       │   ├── NOTES.txt
│       │   ├── _helpers.tpl
│       │   ├── app
│       │   │   ├── _helpers.tpl
│       │   │   ├── deployment.yaml
│       │   │   ├── hpa.yaml
│       │   │   └── service.yaml
│       │   ├── config.yaml
│       │   └── rbac.yaml
│       ├── values.yaml
│       └── vecompass-override.yaml
├── pkg
│   ├── cronjob            # 项目的定时任务模块，包含了项目的定时任务定义和执行逻辑
│   │   ├── init.go
│   │   ├── model.go
│   │   └── sync_job.go
│   ├── db
│   │   ├── gen            # 数据库model 生成代码，包含了项目的数据库访问函数和结构体定义，不需要变动
│   │   ├── init.go
│   │   └── model
│   ├── handler            # 项目的接口处理模块，包含了项目的接口处理函数和结构体定义
│   │   ├── callback.go
│   │   ├── run_job.go
│   │   ├── validate.go
│   ├── log
│   │   └── log.go
│   └── service            # 服务调用的包，主要包含对iam的调用
│       └── iam
└── thrift_gen
```

3.1.3 配置与开发

go 1.22 以上。可直接编译运行，或基于makefile进行构建。

IDE建议以goland进行开发。服务默认端口为6789，NodePort为30002.

配置文件:config/adapterconf/Dev.yaml

代码块

```yaml
Server:
  Port: 6789
  EnablePprof: false
Log:
  Level: info
  Mode: dev
#如果不需要同步组织机构，则SyncOrg里边的不用配置值
IsNeedSyncOrg: false
# 是否同步用户、组织ID
IsSyncID: false
# 是否为访客 false同步的用户是普通用户，true的话同步的是访客
IsVisitor: false
DefaultTenant: dev
SSO:
  HiAgentWebCallback: "http://33.234.30.131:32300/api/auth/callback"
  HiAgentOrigin: "http://33.234.30.131:32300"
  TokenURL: "http://127.0.0.1:18080/token"
  UserInfoURL: "http://127.0.0.1:18080/user_info"
SyncOrg:
  RootNode: "" #默认为空即可
  SyncCronTime:  "" #定时任务表达式
  SpecialHandlingNode: "" # 默认为空
  DBConfig:
    # 数据库类型：mysql\dameng\kingbase\postgresql
    DBType: dameng
    Endpoints: 33.234.30.131:31748
    User: root
    Password: ""
    DBName: iam
    ReadTimeout: "10s"
    Charset: utf8mb4
    TimeZone: UTC
  IAM: #top访问地址
    Host: top-server:8000
```

项目根目录下增加 .env文件，用于配置环境变量。

代码块

```
MYSQL_PASSWORD=""
```

helm chart 配置变更:chart配置在manifests/iam-adapter/下(需要学习helm基础)

配置文件定义：manifests/iam-adapter/values.yaml config代码块

覆写文件定义（用于部署覆盖配置）：manifests/iam-adapter/vecompass-override.yaml

组织同步开发说明

仅进行组织架构同步时，无需进行代码改动，重点关注上述标黄配置。需要在helm chart中同步修改上述配置。

3.1.4 组织架构同步

数据同步当前含三张表：临时用户表、临时组织表、临时组织用户关系表。表结构定义在ddl目录下。

其中，临时用户表、临时组织表、临时组织用户关系表的数据来源于客户系统，由客户自行开发对应的写入逻辑。

（临时空间表及空间关系表，可视客户需求来判断是否同步，非必要默认不同步）

数据表定义如下：

代码块

```
create table if not exists tmp_user
(
    id           varchar(64)                                not null comment '唯一随机ID'
        primary key,
    created_time timestamp        default CURRENT_TIMESTAMP not null comment '创建时间',
    updated_time timestamp        default CURRENT_TIMESTAMP not null on update CURRENT_TIMESTAMP comment '更新时间',
    tenant_id    varchar(64)      default '0'               not null comment '租户ID',
    user_name    varchar(128)     default ''                not null comment '用户名',
    description  varchar(255)     default ''                not null comment 'user描述',
    display_name varchar(255)     default ''                not null comment '展示名、昵称',
    email        varchar(256)     default ''                not null comment '邮箱',
    mobile       varchar(256)     default ''                not null comment '手机号',
    source       varchar(16)      default 'Platform'        not null comment '用户来源: Platform/CAS/Oauth',
    status       tinyint unsigned default '0'               not null comment '状态 1:active 0:inactive',
```

```
    is_deleted   tinyint unsigned default '0'               not null comment '是否删除 1:是 0:否',
    constraint uk_tenant_id_user_name
        unique (tenant_id, user_name)
)
    comment '临时用户表';

create table if not exists tmp_organization
(
    id           varchar(64)                            not null comment '唯一随机ID'
        primary key,
    created_time timestamp    default CURRENT_TIMESTAMP not null comment '创建时间',
    updated_time timestamp    default CURRENT_TIMESTAMP not null on update CURRENT_TIMESTAMP comment '更新时间',
    name         varchar(128) default ''                not null comment '组织名',
    org_code     varchar(128) default ''                not null comment '组织code',
    tenant_id    varchar(64)  default ''                not null comment '租户id',
    pid          varchar(64)  default ''                not null comment '父节点id',
    is_deleted   tinyint      default '0'               not null comment '是否删除 1:是 0:否',
    constraint uk_tenant_id_org_code
        unique (tenant_id, org_code)
)
    comment '临时组织表';

create table if not exists tmp_org_user_relation
(
    id           varchar(64)                           not null comment '唯一随机ID'
        primary key,
    created_time timestamp   default CURRENT_TIMESTAMP not null comment '创建时间',
    updated_time timestamp   default CURRENT_TIMESTAMP not null on update CURRENT_TIMESTAMP comment '更新时间',
    org_id       varchar(64)                           not null comment '组织id',
    user_id      varchar(64) default ''                not null comment '用户id',
    tenant_id    varchar(64) default ''                not null comment '租户ID'
)
    comment '临时用户组织关系表';
```

- 表字段说明
- tmp_user:
  - id: 唯一随机ID，必填（该id直接用客户业务系统中用户id填写）
  - user_name: 用户名，必填，唯一
  - tenant_id: 租户ID，必填。该信息可从客户部署的HiAgent环境中进行获取当前租户ID
  - display_name: 展示名，必填
  - source: 用户来源，必填，默认CAS
  - status: 用户状态，必填，默认1
  - is_deleted: 是否删除，必填，默认0
- tmp_organization:
  - id: 唯一随机ID，必填（该id直接用客户业务系统中组织id填写）
  - name: 组织名，必填
  - org_code: 组织code，必填，唯一
  - tenant_id: 租户ID，必填。该信息可从客户部署的HiAgent环境中进行获取当前租户ID
  - pid: 父节点id，必填 (根节点pid为空)
  - is_deleted: 是否删除，必填，默认0
- tmp_org_user_relation:
  - id: 唯一随机ID(该id非必填，可数据库自增)
  - org_id: 组织id，必填
  - user_id: 用户id，必填
  - tenant_id: 租户ID，必填。该信息可从客户部署的HiAgent环境中进行获取当前租户ID

同步处理逻辑：

4. 创建临时表，确定临时表创建所在库。可以在iam库中新建，或者由客户自行创建临时表，并提供数据库访问地址。

若在iam中创建临时表，则写入数据时，用户数据不带手机号和邮箱两个字段。

5. 临时表数据写入，该部分涉及到客户系统表数据适配并写入我们临时表，牵扯到客户自身的业务逻辑。由客户侧自行完成。
6. 配置定时任务，通过修改config.yaml文件，配置定时任务的执行时间。配置文件中 SyncCronTime 字段为配置定时任务。

3.2 测试用例（验证方法）

3.2.1 临时表数据验证

目的：在组织架构同步过程中，经常发现因为临时表数据不满足同步要求，导致同步失败，排查起来也比较费时费力，因此添加数据临时表数据验证功能，可以在组织架构同步前验证数据的正确性，有问题可以及时发现，及时进行相应的修改，提升组织架构同步的效率

方式：

代码块

```
# 进入adapter pod容器
kubectl exec -it <adapter-pod-name> -n <namespace> -- /bin/bash
# 调用检查临时表数据的接口
curl http://localhost:6789/CheckData
```

根据接口响应：有错误信息检查修改临时表数据，验证成功后进行组织架构同步

3.2.2 组织架构同步

代码块

```
# 手动组织架构同步，也需要进入adapter容器
curl --location --request POST 'http://localhost:6789/RunJob' --data ''
```

直接在客户环境进行组织同步

组织架构同步后，可在数据库中或HiAgent 租户管理页面组织管理中进行查看同步到的数据

3.3 部署方式

镜像构建通过makefile命令进行执行

代码块

```
make build
```

chart 包部署，通过helm进行部署至对应集群环境。部署前确认好镜像的地址以及chart配置

代码块

```
# 安装，项目根目录下执行，命名空间和kubeconfig 配置视具体情况而定
helm install iam-adapter -n vke-system manifest/iam-adapter --kubeconfig
~/.kube/{kubeconfig} -debug
```

```
# 更新
helm upgrade iam-adapter -n vke-system manifest/iam-adapter --kubeconfig ~/.kube/{kubeconfig} -debug
# 删除
helm uninstall iam-adapter -n vke-system manifest/iam-adapter --kubeconfig ~/.kube/{kubeconfig} -debug
```

3.4 HiAgent配置更新及影响范围

对HiAgent 平台功能无影响

4. 联系人

 李硕