

## 一、创建配置应用用户

1. 所有节点新建appuser用户

```
adduser appuser
```

2. 配置appuser用户环境变量，在.bash\_profile中增加，

```
export HADOOP_USER_NAME=impala
export HADOOP_OPTS=-Djava.security.egd=file:/dev/./dev/urandom
```

1) 指定impala作为hadoop系统默认用户， 2) 解决sqoop连接oracle时连接中断问题

3. 上传配置文件

将beeline-file-1.txt和sqoop-import-options-file-1.txt 上传到目录 /home/appuser/app-env

## 二、创建业务数据库

使用impala 用户登录hue 创建数据库

```
create database if not exists zhbd_dw;
create database if not exists zhbd_mid;
create database if not exists zhbd_rds;
create database if not exists zhbd_summary;
create database if not exists zhbd_tmp;
```

## 三、azkaban配置

1. 创建数据库

```
create database azkaban DEFAULT CHARACTER SET utf8;
grant all on azkaban.* TO 'azkaban'@'%' IDENTIFIED BY 'azkaban';
```

2. 初始化数据

```
CREATE TABLE active_executing_flows (
    exec_id INT,
    update_time BIGINT,
    PRIMARY KEY (exec_id)
);

CREATE TABLE active_sla (
    exec_id INT NOT NULL,
    job_name VARCHAR(128) NOT NULL,
    check_time BIGINT NOT NULL,
    rule TINYINT NOT NULL,
    enc_type TINYINT,
    options LONGBLOB NOT NULL,
    primary key(exec_id, job_name)
);

CREATE TABLE execution_flows (
    exec_id INT NOT NULL AUTO_INCREMENT,
    project_id INT NOT NULL,
    version INT NOT NULL,
    flow_id VARCHAR(128) NOT NULL,
    status TINYINT,
    submit_user VARCHAR(64),
    submit_time BIGINT,
    update_time BIGINT,
    start_time BIGINT,
```

```

        end_time BIGINT,
        enc_type TINYINT,
        flow_data LONGBLOB,
        executor_id INT DEFAULT NULL,
        PRIMARY KEY (exec_id)
    );

CREATE INDEX ex_flows_start_time ON execution_flows(start_time);
CREATE INDEX ex_flows_end_time ON execution_flows(end_time);
CREATE INDEX ex_flows_time_range ON execution_flows(start_time, end_time);
CREATE INDEX ex_flows_flows ON execution_flows(project_id, flow_id);
CREATE INDEX executor_id ON execution_flows(executor_id);
CREATE INDEX ex_flows_staus ON execution_flows(status);
CREATE TABLE execution_jobs (
    exec_id INT NOT NULL,
    project_id INT NOT NULL,
    version INT NOT NULL,
    flow_id VARCHAR(128) NOT NULL,
    job_id VARCHAR(128) NOT NULL,
    attempt INT,
    start_time BIGINT,
    end_time BIGINT,
    status TINYINT,
    input_params LONGBLOB,
    output_params LONGBLOB,
    attachments LONGBLOB,
    PRIMARY KEY (exec_id, job_id, attempt)
);

CREATE INDEX exec_job ON execution_jobs(exec_id, job_id);
CREATE INDEX exec_id ON execution_jobs(exec_id);
CREATE INDEX ex_job_id ON execution_jobs(project_id, job_id);

CREATE TABLE execution_logs (
    exec_id INT NOT NULL,
    name VARCHAR(128),
    attempt INT,
    enc_type TINYINT,
    start_byte INT,
    end_byte INT,
    log LONGBLOB,
    upload_time BIGINT,
    PRIMARY KEY (exec_id, name, attempt, start_byte)
);

CREATE INDEX ex_log_attempt ON execution_logs(exec_id, name, attempt);
CREATE INDEX ex_log_index ON execution_logs(exec_id, name);
CREATE INDEX ex_log_upload_time ON execution_logs(upload_time);

CREATE TABLE executor_events (
    executor_id INT NOT NULL,
    event_type TINYINT NOT NULL,
    event_time DATETIME NOT NULL,
    username VARCHAR(64),
    message VARCHAR(512)
);

CREATE INDEX executor_log ON executor_events(executor_id, event_time);
CREATE TABLE executors (
    id INT NOT NULL PRIMARY KEY AUTO_INCREMENT,
    host VARCHAR(64) NOT NULL,
    port INT NOT NULL,
    active BOOLEAN DEFAULT false,
    UNIQUE (host, port),
    UNIQUE INDEX executor_id (id)
);

```

```

CREATE INDEX executor_connection ON executors(host, port);

CREATE TABLE project_events (
    project_id INT NOT NULL,
    event_type TINYINT NOT NULL,
    event_time BIGINT NOT NULL,
    username VARCHAR(64),
    message VARCHAR(512)
);

CREATE INDEX log ON project_events(project_id, event_time);

CREATE TABLE project_files (
    project_id INT NOT NULL,
    version INT not NULL,
    chunk INT,
    size INT,
    file LONGBLOB,
    PRIMARY KEY (project_id, version, chunk)
);

CREATE INDEX file_version ON project_files(project_id, version);
CREATE TABLE project_flows (
    project_id INT NOT NULL,
    version INT NOT NULL,
    flow_id VARCHAR(128),
    modified_time BIGINT NOT NULL,
    encoding_type TINYINT,
    json BLOB,
    PRIMARY KEY (project_id, version, flow_id)
);

CREATE INDEX flow_index ON project_flows(project_id, version);

CREATE TABLE project_permissions (
    project_id VARCHAR(64) NOT NULL,
    modified_time BIGINT NOT NULL,
    name VARCHAR(64) NOT NULL,
    permissions INT NOT NULL,
    isGroup BOOLEAN NOT NULL,
    PRIMARY KEY (project_id, name)
);

CREATE INDEX permission_index ON project_permissions(project_id);

CREATE TABLE project_properties (
    project_id INT NOT NULL,
    version INT NOT NULL,
    name VARCHAR(255),
    modified_time BIGINT NOT NULL,
    encoding_type TINYINT,
    property BLOB,
    PRIMARY KEY (project_id, version, name)
);

CREATE INDEX properties_index ON project_properties(project_id, version);

CREATE TABLE project_versions (
    project_id INT NOT NULL,
    version INT not NULL,
    upload_time BIGINT NOT NULL,
    uploader VARCHAR(64) NOT NULL,
    file_type VARCHAR(16),
    file_name VARCHAR(128),
    md5 BINARY(16),
    num_chunks INT,
    resource_id VARCHAR(512) DEFAULT NULL,

```

```

        PRIMARY KEY (project_id, version)
    );

    CREATE INDEX version_index ON project_versions(project_id);

    CREATE TABLE projects (
        id INT NOT NULL PRIMARY KEY AUTO_INCREMENT,
        name VARCHAR(64) NOT NULL,
        active BOOLEAN,
        modified_time BIGINT NOT NULL,
        create_time BIGINT NOT NULL,
        version INT,
        last_modified_by VARCHAR(64) NOT NULL,
        description VARCHAR(2048),
        enc_type TINYINT,
        settings_blob LONGBLOB,
        UNIQUE INDEX project_id (id)
    );

    CREATE INDEX project_name ON projects(name);

    CREATE TABLE properties (
        name VARCHAR(64) NOT NULL,
        type INT NOT NULL,
        modified_time BIGINT NOT NULL,
        value VARCHAR(256),
        PRIMARY KEY (name, type)
    );

    CREATE TABLE triggers (
        trigger_id INT NOT NULL AUTO_INCREMENT,
        trigger_source VARCHAR(128),
        modify_time BIGINT NOT NULL,
        enc_type TINYINT,
        data LONGBLOB,
        PRIMARY KEY (trigger_id)
    );

```

### 3. 应用程序安装

管理节点 安装 azkaban-web-server-3.32.1

1) tar zxvf azkaban-web-server-3.32.1.tar.gz

受管节点 安装 azkaban-exec-server-3.32.1

1) tar zxvf azkaban-exec-server-3.32.1.tar.gz

### 4. 启动顺序

启动执行器

```

cd /home/appuser/azkaban-exec-server-3.32.1
bin/start-exec.sh

```

修改数据库

```

update executors set active=1

```

启动管理器

```

cd /home/appuser/azkaban-web-server-3.32.1
bin/start-web.sh

```

访问地址

