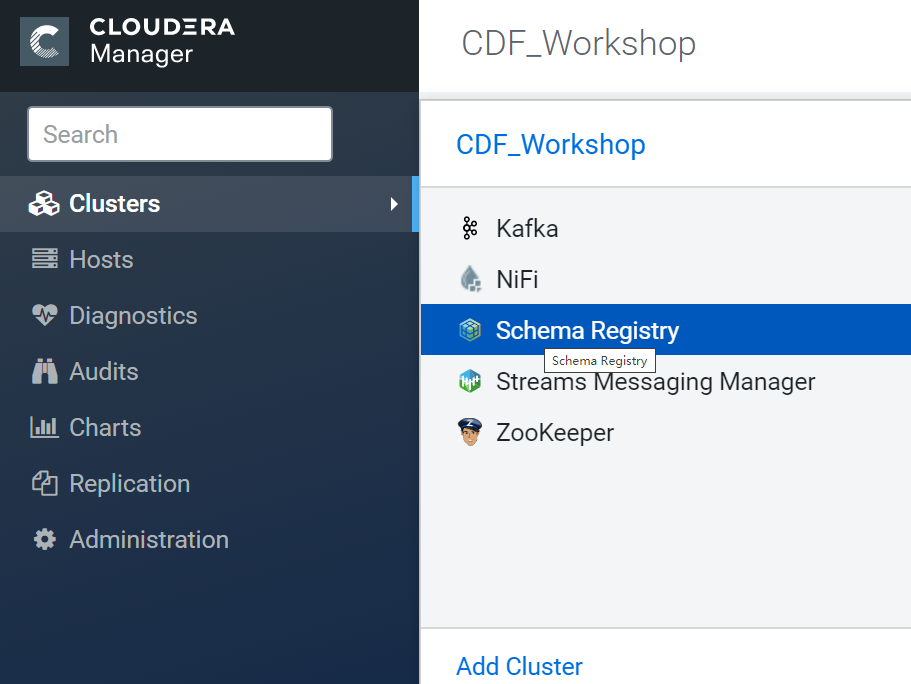
**Lab 4**

在本次实验中，我们将学到:

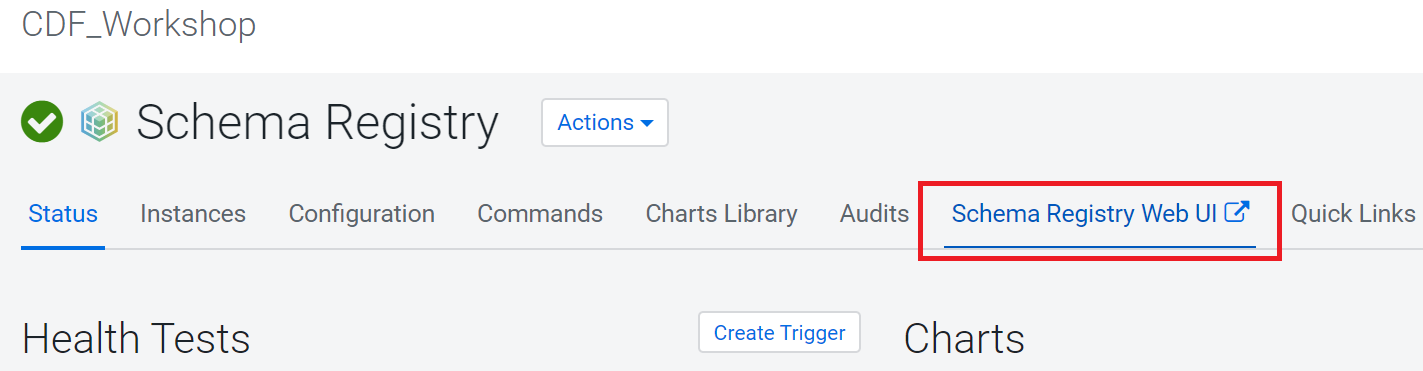
* 使用Schema Registry 来创建一个Kafka Message Schema
* 将Meetup RSVP 的数据通过Schemazhua转换后发送至该Kafka Topic
* 使用Streaming Message Manager 查看该Topic的数据

**使用**Schema Registry **管理Kafka消息的Schema**

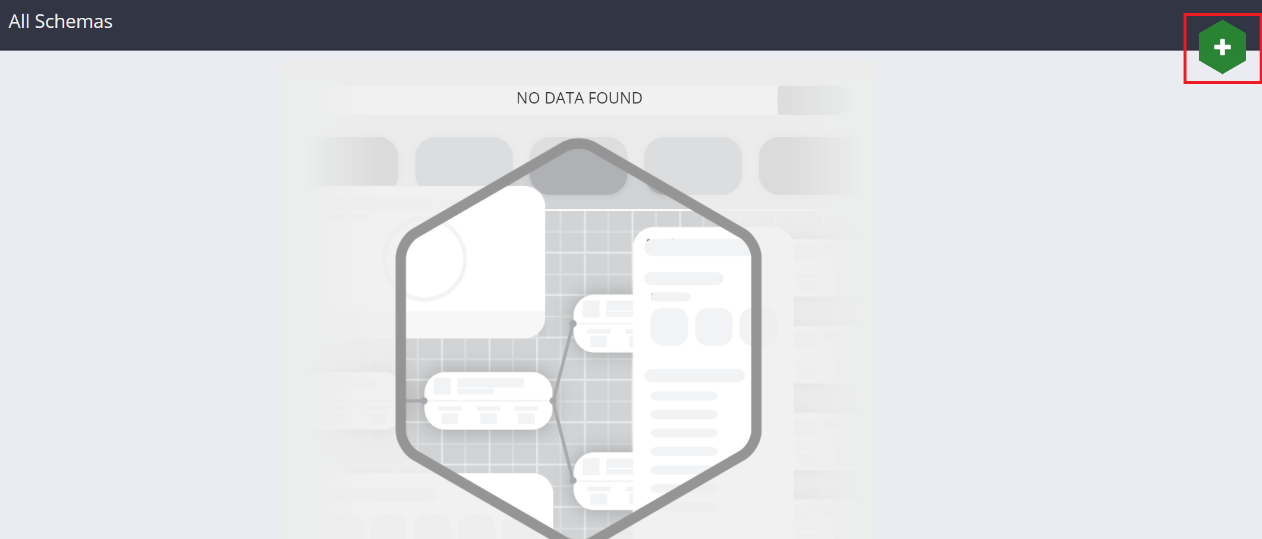
* 步骤1: 登录Cloudera Manager
  + 在浏览器中输入你的IP:7180打开Cloudera Manager登录界面
  + 输入用户名 admin , 密码 Cl0udera 登录Cloudera Manager
  + 选择Schema Registry服务



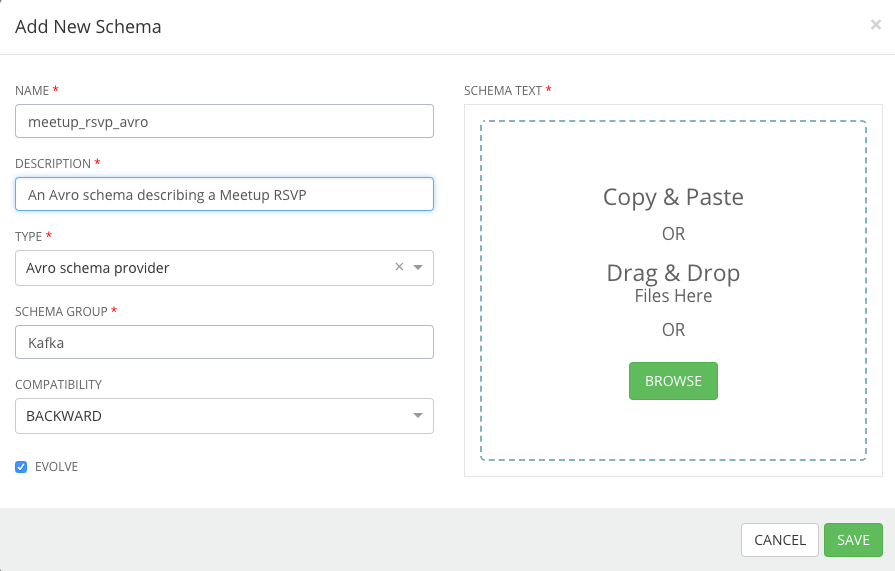
* 步骤2: 登录Schema Registry 服务
  + 选择通过Schema Registry WebUI 快速链接打开Schema Registry



* + 点击+ 按钮



* + 输入如下信息并保存来完成添加一个新的Schema

Name : meetup\_rsvp\_avro

Description：meetup\_rsvp\_avro

Type : Avro Schema Proovider

Schema Group: Kafka

Compatibility 选择 Backward

{

Schema Text 中输入如下信息

"type": "record",

"name": "meetup\_rsvp\_avro",

"fields": [

{

"name": "event\_name",

"type": "string",

"default": "no\_name"

},

{

"name": "event\_url",

"type": "string",

"default": "no\_url"

},

{

"name": "venue",

"type":

{

"type": "record",

"name": "venue",

"fields": [

{

"name": "lat",

"type": "float",

"default": 0.0

},

{

"name": "lon",

"type": "float",

"default": 0.0

},

{

"name": "name",

"type": "string",

"default": "no\_venue\_name"

}

]

}

},

{

"name": "group",

"type":

{

"type": "record",

"name": "group",

"fields": [

{

"name": "group\_city",

"type": "string",

"default": "no\_group\_city"

},

{

"name": "group\_country",

"type": "string",

"default": "no\_group\_country"

},

{

"name": "group\_name",

"type": "string",

"default": "no\_group\_name"

},

{

"name": "group\_state",

"type": "string",

"default": "no\_group\_state"

},

{

"name": "urlkey",

"type": "string",

"default": "no\_group\_urlkey"

},

{

"name": "topic\_name",

"type": "string",

"default": "no\_group\_topic\_name"

}

]

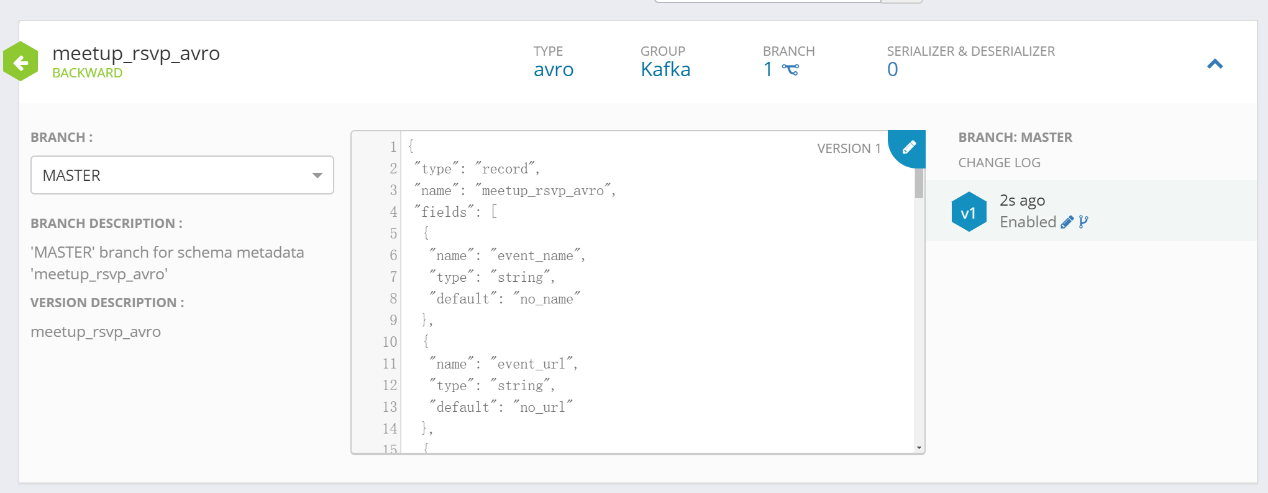
}

}

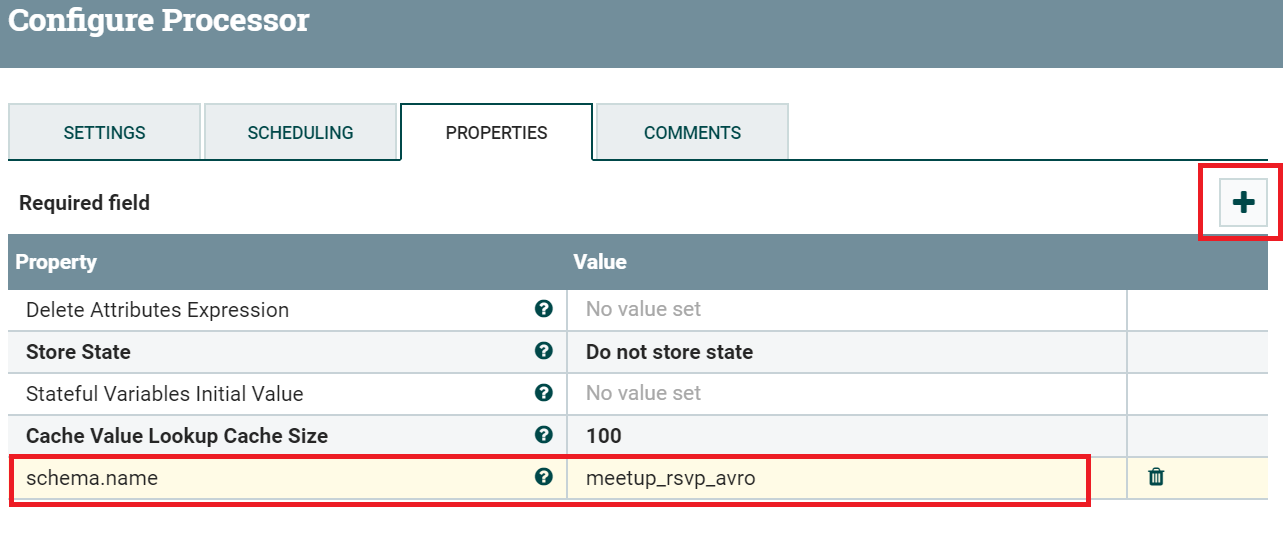
]

}

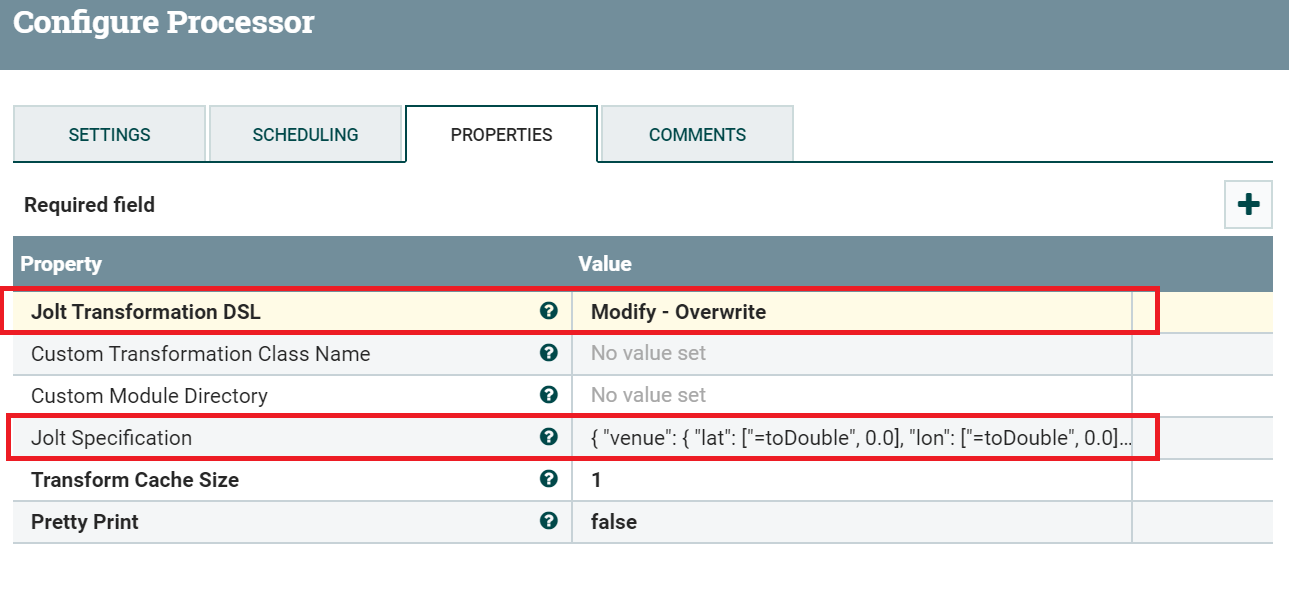
* + 点击Save 按钮并确认你的Schema 正确创建



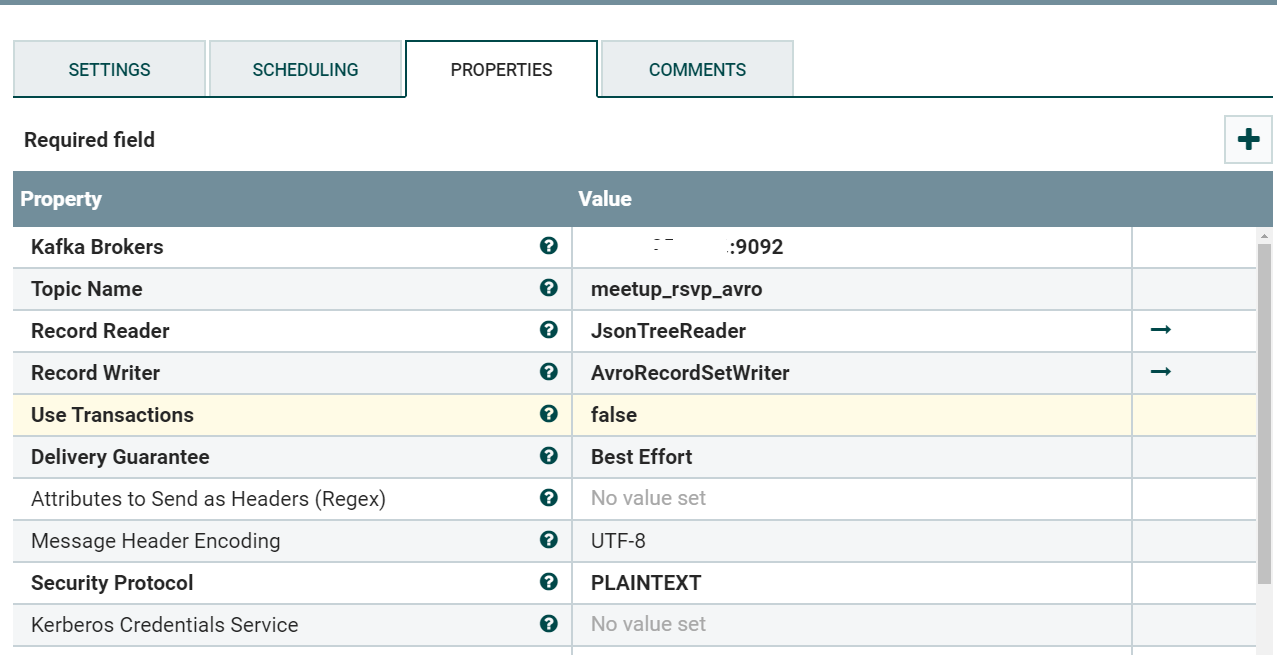
* 步骤3：通过Streaming Message Manager创建一个新的Topic，
  + 名为meetup\_rsvp\_avro, 具体方法请参考Lab 3.
* 步骤4：返回NIFI页面，并添加一个Update Attribute处理器
  + 添加一个属性 schema.name : meetup\_rsvp\_avro



* + 添加一个JoltTransformation处理器



* 步骤5：添加一个Publish\_Kafka\_Record\_2.0 处理器



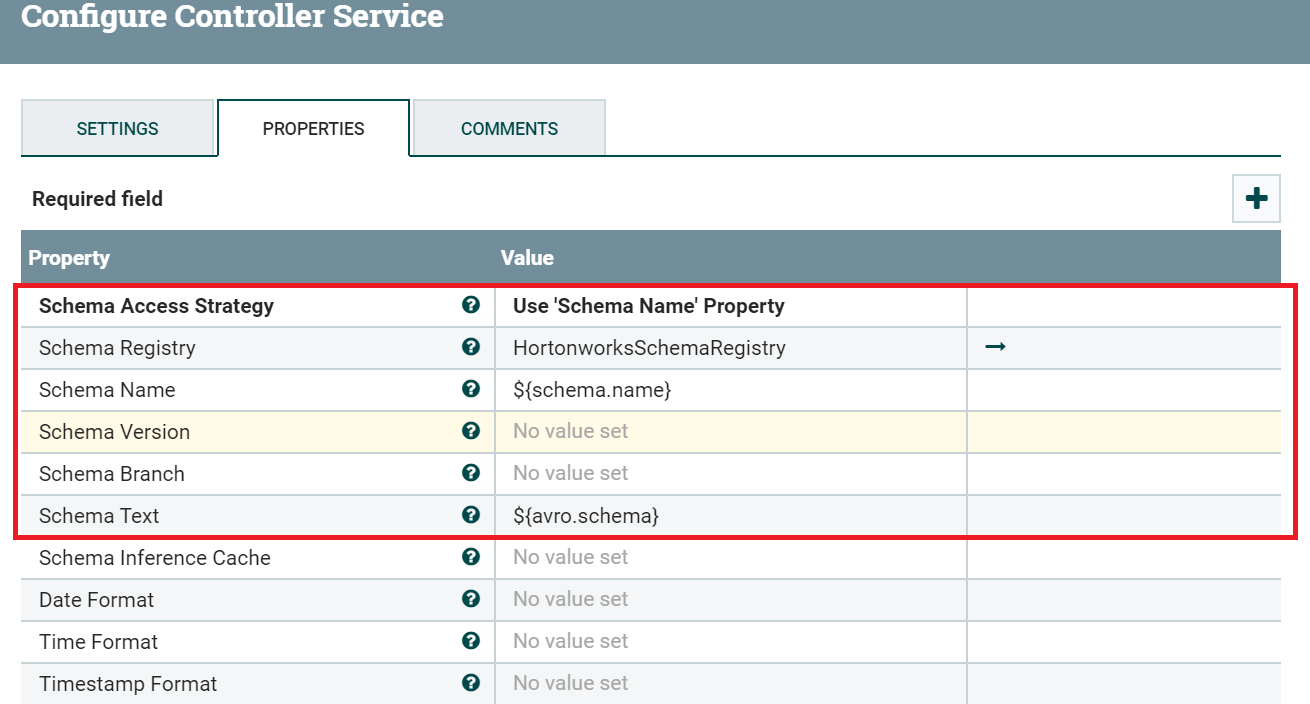
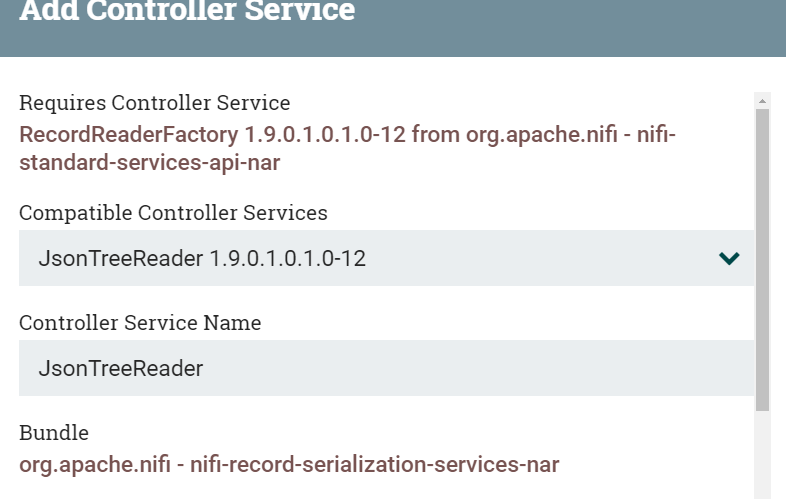
Topic为 meetup\_rsvp\_avro

Record Reader处选择JsonTreeReader 服务

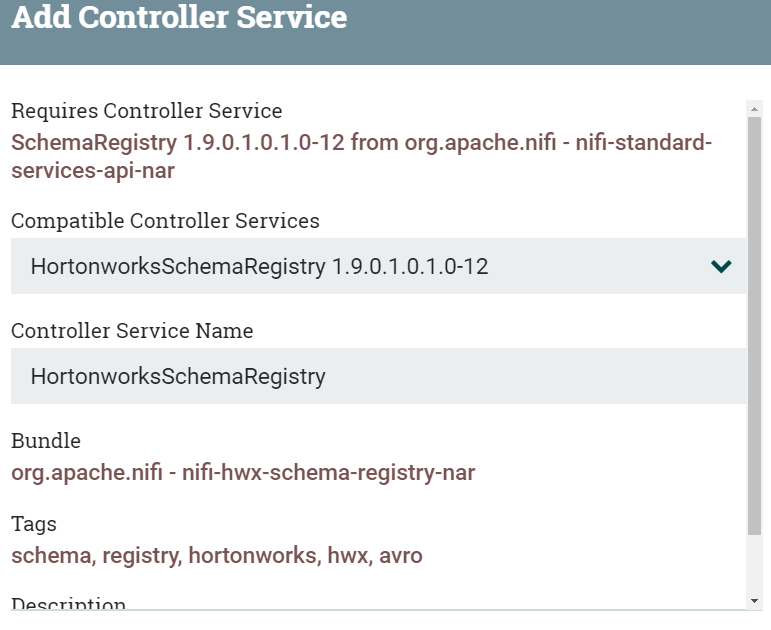
Record Writer 选择AvroRecordWriter服务

Use Transaction 为 False

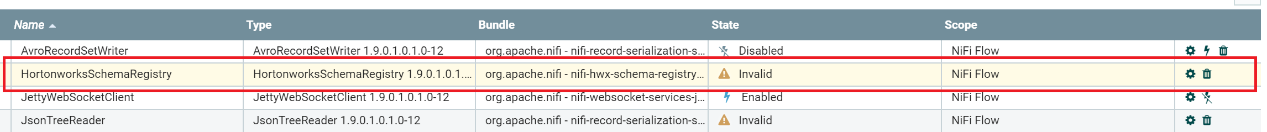
Delivery Guarantee为Best Effort



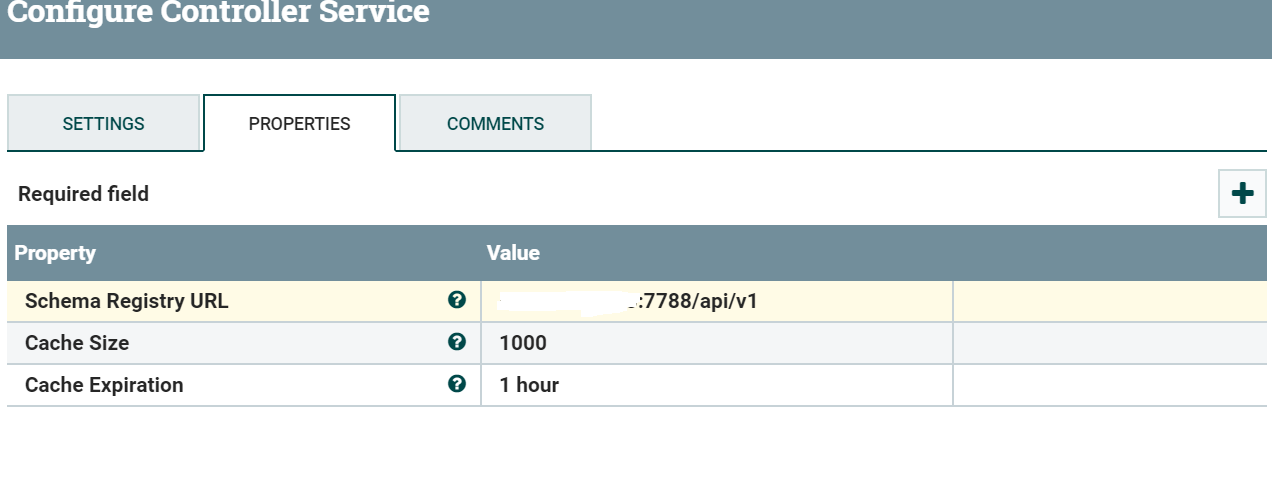
* + 配置Schema Access Strategy为 Use’Schema Name’ Property
  + JsonTreeReader服务还引用了SchemaRegistry服务，需要新建一个SchemaRegistry服务



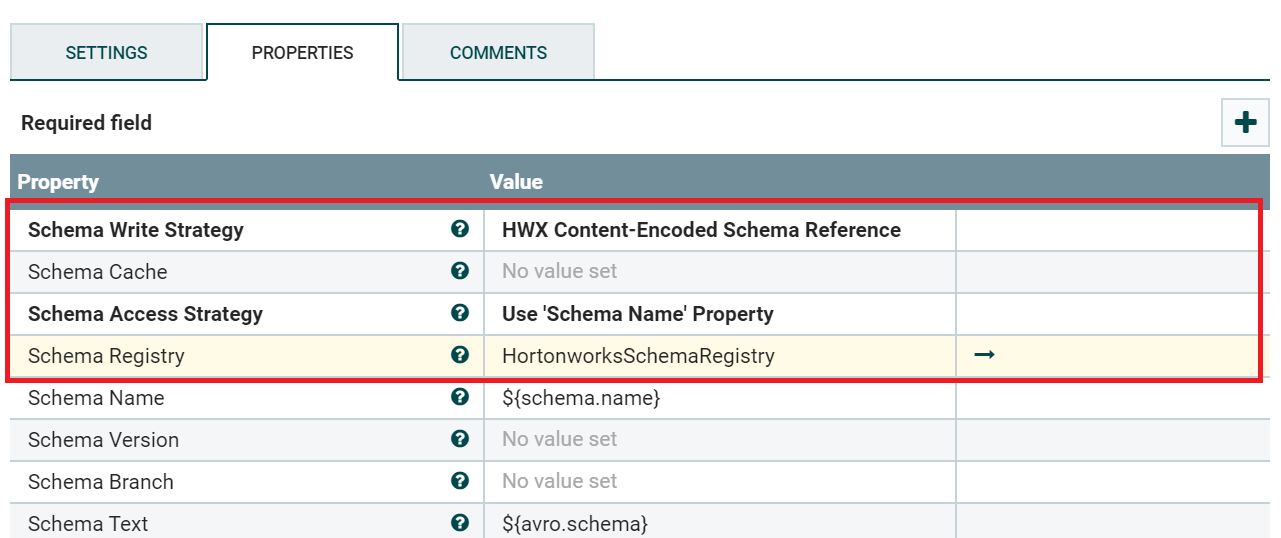
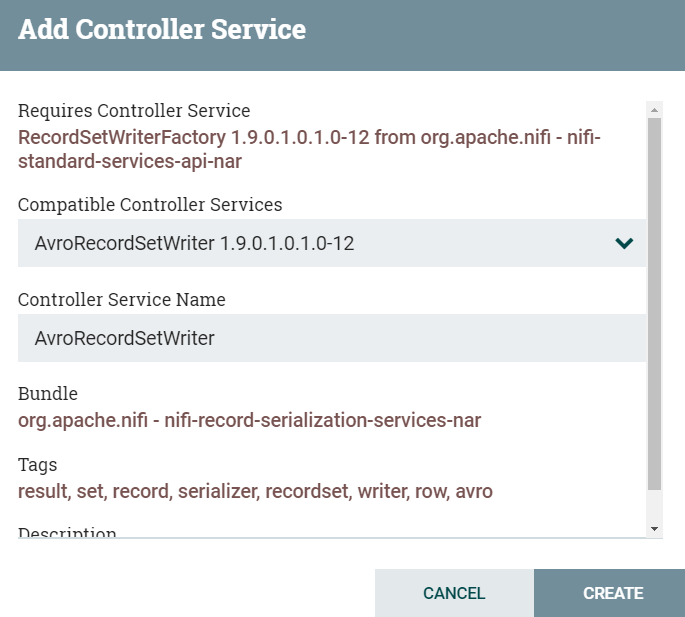
* + 配置Hortonworks Schema Registry 服务



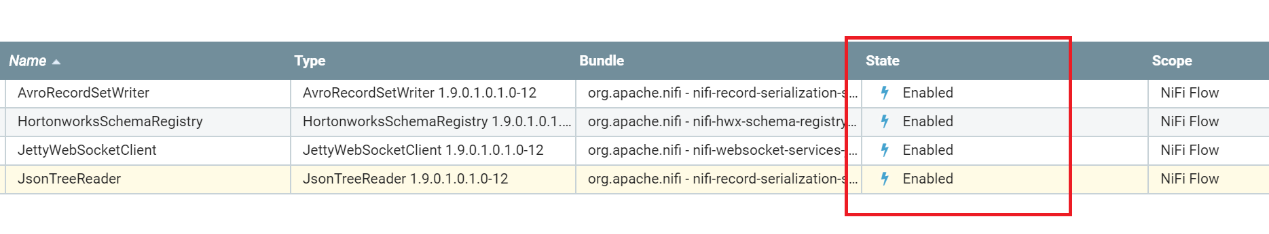
* + Schema Registry URL为http://你的IP:7788/api/v1



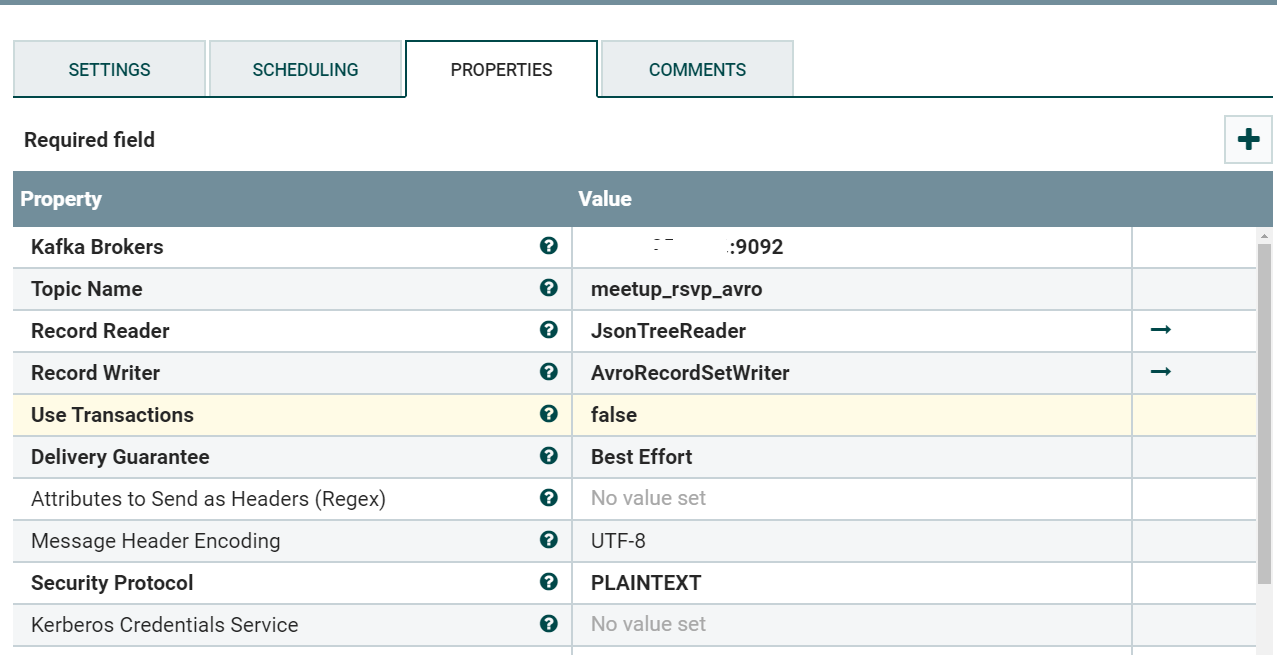
* + 配置AvroRecordSetWriter 服务



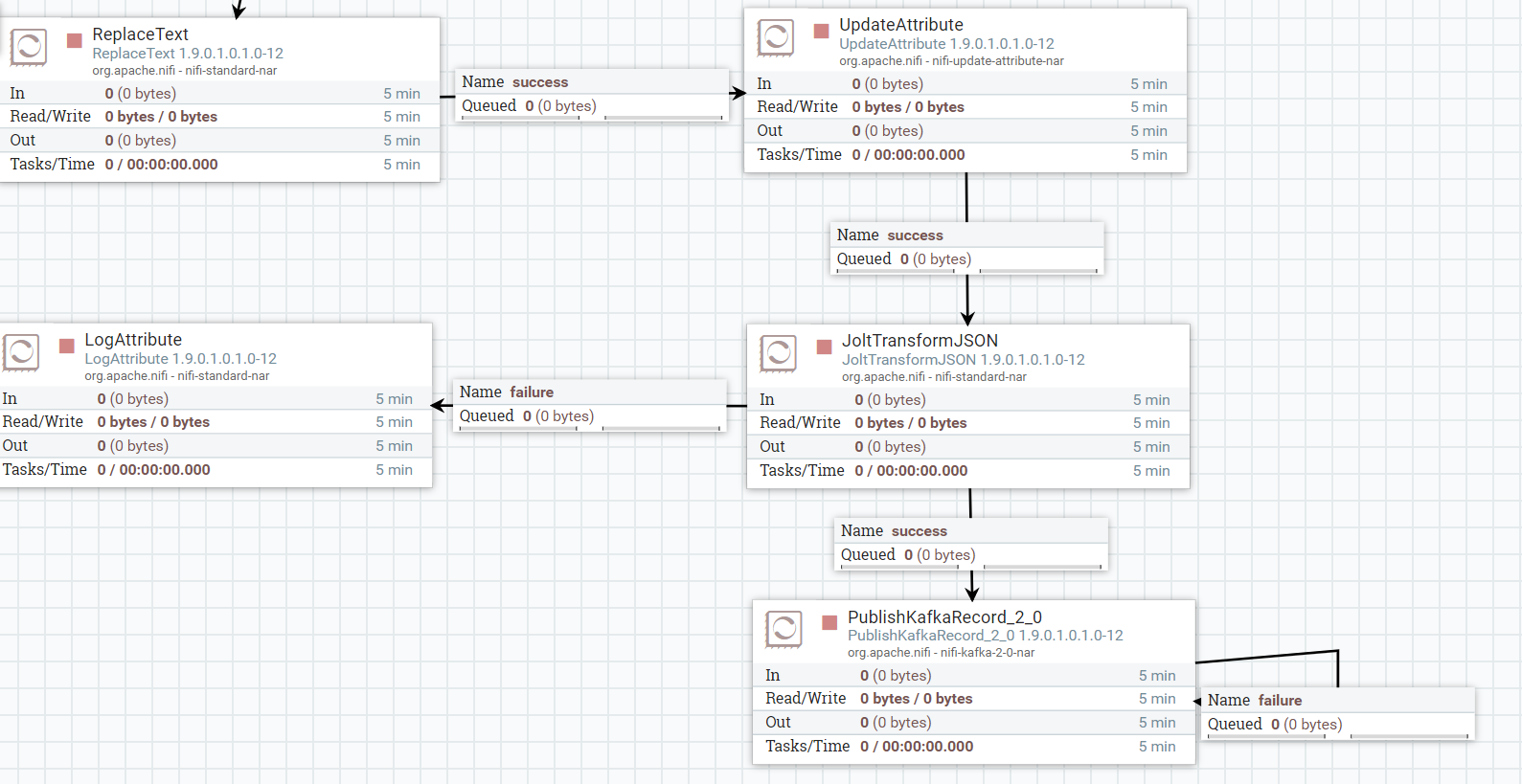
启动这些公共服务，并确定这些服务是出于Enabled状态



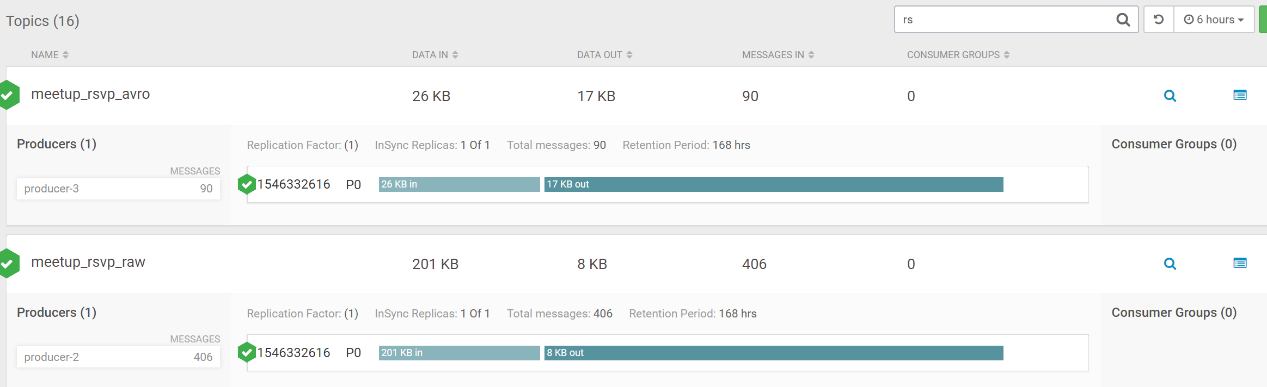
* + 打开Publish\_Kafka\_Record\_2.0 处理器并确认各项配置无误



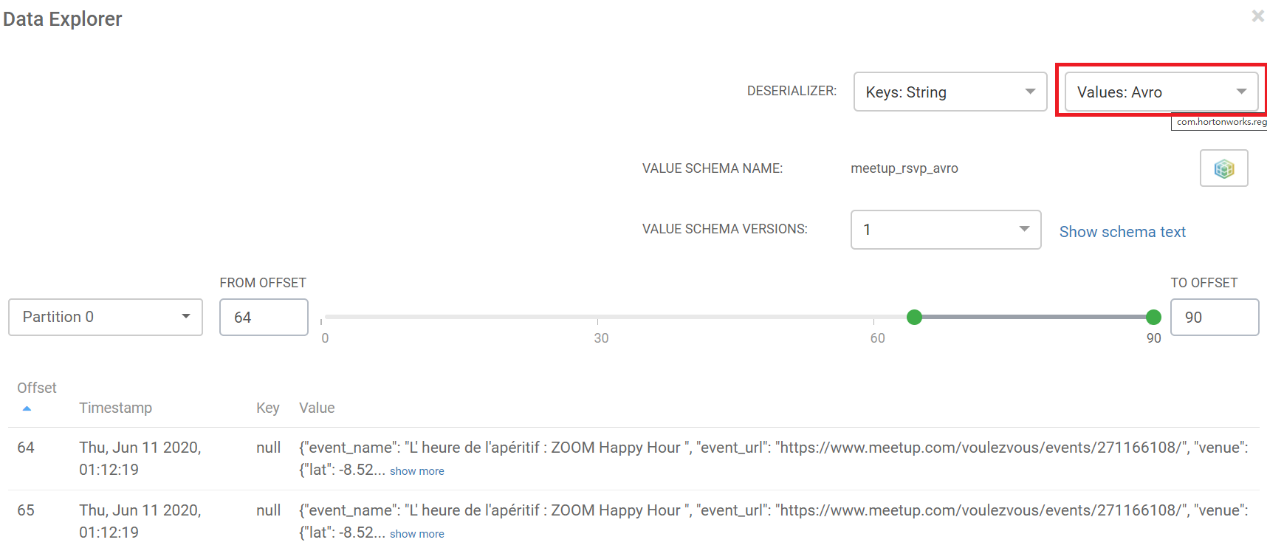
* + 你的新添加的数据流应该和下图类似



* 步骤6：启动这个数据流，并登录Streaming Message Manager 查看该Topic 是否有数据



* + 点击放大镜按钮，查看Topic中的数据,并在Value Deserializer选择 Avro和String 进行不同模式的查看



本次Lab 4实验到此完成

**思考问题**

1. 为什么要使用Schema Registry?