Solution:

1. **Creating a pipeline and storing data?**
    a. As described in setup part, we can create a kafka stream that will consume data from client keep storing them. After that, druid indexing service will consume data from kafka and will store them in druid database. As you can see in the **mockstagram_supervisor_spec.json** , we have defined all the dimensions that we will be storing for later purpose. And they are **pk,followerCount, followingCount, username, timestamp, followerRatio.** So, running the druid indexex with **mockstagram_supervisor_spec** will do our job.

2. **The most recent data of an influencer on the page?**
    a. It can be done using a druid query where we can query data of last minute or most recent. This can be done using druid time series query.

3. **The followerCount rank of an influencer?**
    a. Now to get the influence rank we need to run a query which will take most recent data of each user and then rank them according to the follower count. This can be done using druid time series query.

4. **The averagefollowerCount across all influencers?**
    a. This is similar think like above two. We need to write a timeseries query which will return the averageFollowerCount using most stats of each user.

5. **Bonus Task solution :**
    a. In our native database, where we are storing the user details, we can add one more field **isSuspicious**. And it will be updated only once in a day. So, we can configure a scheduler job which will hit the API which will return whether user is **suspicious** or not. And then we can update that user detail in the DB. So, after this as we can remove all the user while calculating the rank of influencer. Like, druid query will return the user's pk and rank but after that we can remove all the user from that list and will update the influencer's rank subsequently. This is solution if you are designing your own custom dashboard.

    b. In case, you want user rank with exclusion of suspicious user on Imply Dashboard only. One thing, we can do is add the **isSuspicious** field in kafka data itself and while querying the druid for latest influencerRank, we can exclude all the influencer with field **isSuspicious** true. And while storing data to kafka we should persist the last value of **isSuspicious** field and should only get updated once on the scheduler job.