# Deciding on how to run DBT on production

**Introduction**

* Introduce DBT

**Running DBT in production**

* Talk about what it means to run DBT in production
* <https://docs.getdbt.com/docs/running-a-dbt-project/running-dbt-in-production>
* setting up a system to run a dbt job on a schedule

**Different ways of scheduling DBT jobs**

* Dbt cloud
* Airflow
* Dagster
* Prefect

**What to consider when making the decision**

* **Scheduling - do you use airflow to orchestrate your dbt runs right now? How much of those runs involve also other non-dbt steps that you can't replace? This is, in my opinion, what will be the decisive point for you. If you run 100% dbt jobs then it's a matter of re doing the same scheduled jobs, but if you'll need to keep airflow it will get more complicated as you'll need to set up additional steps. There are a lot of resources and it now got easier with the official airflow operators, but probably still double the effort in time.**
* [**https://discourse.getdbt.com/t/dagster-vs-prefect/1833**](https://discourse.getdbt.com/t/dagster-vs-prefect/1833)
* **I would say Luigi is no longer a tool anyone uses. It hasn't really help up ever since Airflow came around. Airflow is great for batch jobs on a fixed schedule but starts to show a lot of pain when you are moving to event-based or ad-hoc scheduling. Dagster aims to make testing and development easier, and Prefect seemed more focused on data-science workflows. I've trialed Dagster back at 0.7 and it felt incomplete for me, but I know it's come a long way. If batch/scheduling cases fit 95% of your use-cases, my recommendation would be Airflow, the ecosystem around it is very strong and well-tested. I'd probably suggest** [**Astronomer.io**](http://astronomer.io/) **if you're looking for a hosted solution. It's not hard to host yourself, but it's one more thing to babysit.**
* **I will say for the reverse-etl use case of moving data from your warehouse into sales tools, my personal opinion is that** [**engineers shouldn't write reverse etl.**](https://hightouch.io/blog/reverse-etl-is-not-for-data-engineers-to-write) **I am definitely biased in that position, but it's one I've held before working at Hightouch too. Integrations can be really trick to do right, especially when you get into intermittent failures, logging, and monitoring.**

<https://getdbt.slack.com/archives/CBSQTAPLG/p1636666470241700>

https://getdbt.slack.com/archives/CMZ2Q9MA9/p1618961399039600