**1**

**Watch the required videos on AWS code pipeline and AWS data pipeline and read AWS's literature on the products**.

**In your own words:**

**Compare AWS data pipeline with off-the-shelf airflow in Kubernetes, and AWS code pipeline with Jenkins. How does their feature set differ and how is it the same?**

AWS vs Airflow/K8s:

1. Both AWS and Airflow with k8’s allow for scaling, although AWS is theoretically unlimited, whereas k8’s would of course be limited by available resources. Both offer a GUI for ease of use, and both also manage pipelines across multiple resources, including use of customized code.

AWS pipelines are fairly easy to implement, have a minimal need of configuration and work seamlessly with other AWS services.

Airflow is highly configurable, supports some additional plugins and can be air gapped from the internet if needed.

1. As with the above, both Code Pipelines and Jenkins can accomplish code releases however, there are advantages and disadvantages of both.

Code pipelines integrate with other AWS resources quite well but are limited (as of now) to using GitHub or S3 buckets for versioning. Also, as a SaaS offering Code Pipelines have a minimalist approach to configuration, helping you get code up and running in a short period of time.

Jenkins does essentially the same thing as Code Pipeline but also enjoys a large library of plugins to support just about any need one could think of. Jenkins does require a little more configuration out of the box, this is a pro or con depending on your use case.

One interesting, although maybe an edge case, is that Code Pipeline can use Jenkins in an integrated deployment cycle.

**What are the advantages and disadvantages to using an AWS service vs the setup we have used in Kubernetes?**

1. AWS usually requires much less configuration to get running
2. AWS can tightly and easily couple other AWS resources – ensuring that your build deploy environment is on the “same page”.
3. AWS can theoretically be infinitely scaled both up and out
4. AWS charges only for compute time, while this might be 24/7 for something like an application server, it may only minutes or hours a month for something like Deploy and Code pipelines. This could be quite a bit of savings, as well as not having idle resources
5. A configuration of Airflow and Jenkins on K8s is also very robust, allowing scaling to the degree of available resources.
6. K8s deployments are highly configurable and there is great value to having the ability to “tweak” the tools under the covers.
7. Jenkins’s use of plugins can save a lot of development time and solve many of the common problems found in a typical (and even atypical) CI/CD setup

**If you were building a data pipeline, which would you choose?**

1. If I were looking a low usage patterns or regulated use patterns, I would most likely choose AWS to keep costs down. This is especially true if my targets are in the cloud in the first place!
2. For heavy usage patterns and development tempo CI/CD, I would most likely choose something like Jenkins and Airflow. Although, I might also suggest that it be housed as cloud native systems. I also may be biased here, as the last three large companies I worked for did exactly this with great success.

**2**

**Configure a pipeline in airflow to run "extract\_mysql\_fulll.py", as done last week, but make some changes:**

**● extract\_mysql\_full.py should store the csv file in s3. You can copy the code to do that from extract\_mysql\_incremental.py (be sure to pull the latest version from GitHub as it was modified recently). The name of your generated s3 object should be:**

**week13-<username>.csv in bucket UML.**

**● The dag file should be loaded into airflow from your GitHub repository. Update the helm configuration as described in lab 1 and the helm documentation**.

**Submit**

**1. Airflow logs of your run**

**2. Output of the AWS CLI command: aws s3 ls UML**

**3. Changes to the "dags:" section of the helm config file**

**4. Your modified python program**

**5. A description of your solution: how you overcame problems, and how you tested.**