



Exam



You have passed the quiz!

Your score:

65 of 75 Correct (86%)

Elapsed time:

56 minutes

75 of 75 questions answered

[Hide Answers](#)

Question 1: ✓ Correct answer

Your DAG was paused during 5 days. Its schedule interval is defined to 10 mins. As you can imagine you will get many DAGRuns running. You schedule it again, which view do you think is the most useful one to check how your DAGRuns are going?

- ☐ ~~DAGs view~~
- ☐ ~~Gantt view~~
- ☒ **Tree view**

Question 2: ✓ Correct answer

Is it possible to have multiple connections with the same connection id?

- ☐ ~~Yes, why not?~~
- ☒ **No**

Question 3: ✓ Correct answer

By default with the local executor, how many tasks can you execute in parallel in Airflow?

- ☐ ~~46~~
- ☐ ~~8~~
- ☒ **32**

Question 4: ✔ Correct answer

You can't access the UI neither the REST API of Airflow. Can you still trigger a DAG?

☒ **Yes, through the command airflow dags trigger**

☐ ~~No, the UI and the REST API are the only ways~~

Question 5: ✔ Correct answer

Your DAG should only run for the next two months. Which parameter can help you with that?

☐ ~~period~~

☐ ~~start_date~~

☒ **end_date**

Question 6: ✔ Correct answer

You just wrote a DAG that you would like to trigger every 10 minutes, but not on weekends. What should the value of the schedule interval parameter be?

☐ ~~schedule_interval="*/10 * * * M-F"~~

☒ **schedule_interval="*/10 * * * 1-5"**

☐ ~~schedule_interval=timedelta(minutes=10)~~

Question 7: ✔ Correct answer

You have to fetch data from files coming from your different partners. Therefore, your DAG has to be triggered every day **at** 7:00 in the morning. The start date is defined to **datetime(2021, 1, 1, 1)**. How can you express that?

☐ ~~schedule_interval=timedelta(hours=7)~~

☐ ~~schedule_interval=timedelta(hours=6)~~

☒ **schedule_interval="0 7 * * *"**

Question 8: ✔ Correct answer

You need to fetch data from your Presto database. To do this, you create a connection, but when you look at the connection type:

Conn Id *	presto
Conn Type *	<div> Postgres <input type="text"/> </div> <div> <input type="text"/> </div> <div> HTTP </div> <div> Hashicorp Vault </div> <div> IMAP </div> <div> Kubernetes Cluster Connection </div> <div> Mesos Framework ID </div> <div> MySQL </div> <div> Postgres </div>
Description	
Host	
Schema	
Login	
Password	
Port	
Extra	

Save

you can't find Presto. What should you do?

- ☐ ~~pip install presto~~
- ☒ **pip install apache-airflow-providers-presto**
- ☐ ~~pip install presto-python-client~~

Question 9: ✓ Correct answer

If you want to know which tasks have failed across all DAGRuns of a given DAG, can you check this from the column "Recent Tasks" in the DAGs view?

- ☐ ~~Yes~~
- ☒ **No**

Question 10: ✓ Correct answer

With the Celery Executor, do you have to install Airflow on each machine/worker where your tasks are gonna be executed?

- ☐ ~~No, Celery is enough~~
- ☒ **Yes**

Question 11: ✗ Incorrect answer

On the DAGs view, what is "Last Run" column?

- ☒ ~~The date when the DAG is effectively triggered~~
- ☐ **The most recent execution date**
- ☐ ~~The interval of time between each DAGRun~~

Question 12: ✔ Correct answer

There is a file of 5 Gb you would like to process. Can you share it between your tasks through XCOMs?

- ☐ ~~Yes~~
- ☒ **No**

Question 13: ✔ Correct answer

You're ready to put Airflow in the real world. To start scaling and speed up the processing of your DAGs, you would like to execute multiple tasks at the same time. Constraint, you only have one machine to run Airflow and execute tasks. What would be the easiest to set up and more suitable executor to choose?

- ☒ **Go with the Local Executor**
- ☐ ~~Stay with the Sequential Executor~~
- ☐ ~~Go with the Celery Executor~~

Question 14: ✔ Correct answer

In case of a failure for **task_2** in the DAG below

```
6 ~ default_args = {
7     |     'retries': 5
8     | }
9
10 dag = DAG('my_dag', start_date=datetime(2020, 1, 1), schedule_interval='@daily', catchup=False)
11
12 ~ task_1 = BashOperator(
13     |     task_id='task_1',
14     |     bash_command='ls',
15     |     retry_delay=timedelta(minutes=3),
16     |     dag=dag
17     | )
18
19 ~ task_2 = BashOperator(
20     |     task_id='task_2',
21     |     bash_command='pwd',
22     |     retries=3,
23     |     retry_delay=timedelta(minutes=3),
24     |     dag=dag
25     | )
```

How many times the task will be retried before ending up with the FAILED status?

- ☒ **3**
- ☐ ~~5~~
- ☐ ~~4~~

Question 15: ✔ Correct answer

What is the most typical way to create a data pipeline in Airflow?

☒ **By creating a python file with an instantiated DAG object inside of it. Some expected arguments for the DAG object include the dag_id, start_date and the schedule_interval**

☐ ~~By creating a python file with a python function named DAG inside of it. This function must include some arguments such as a dag_id, start_date, and the schedule_interval.~~

☐ ~~By creating a yaml file with the DAG parameters specified. Some parameters include the dag_id, the start_date and the schedule_interval.~~

Question 16: ✔ Correct answer

When you install Airflow for the first time, you get the Sequential Executor. Are you able to execute multiple tasks at the same time?

☒ **No**

☐ ~~Yes~~

Question 17: ✔ Correct answer

You've defined dependencies between tasks of your DAG. You would like to check that they are correct. Which view appears to be the most suitable?

☐ ~~DAGs view~~

☐ ~~Tree view~~

☒ **Graph view**

Question 18: ✔ Correct answer

Once your DAG has a unique id, the next step is to define when it will start being scheduled. Let's say, you want to start scheduling the DAG as of 2021/01/01. How would you do that?

☐ ~~start_date="2021/01/01"~~

☐ ~~schedule_interval=datetime(2021, 1, 1)~~

☒ **start_date=datetime(2021, 1, 1)**

Question 19: ✔ Correct answer

You had to pause a DAG for auditing purpose. That DAG has a short schedule interval. Therefore, by scheduling it again you might end up with too many DAGRuns running in parallel. How can you limit that number of running DAGRuns?

☐ ~~By changing the start date to the current date~~

☐ ~~By setting the parameter catchup to False~~

☒ **With the parameter max_active_runs**

Question 20: ✔ Correct answer

You got them! You can see the files and folders of Airflow which means, Airflow is initialized! What is the next step? Running Airflow? Let's do it! By the way, what components Airflow needs to run with the default executor?

- ☐ ~~A web server and a database~~
- ☒ **A web server, a scheduler and a database**
- ☐ ~~A web server, a worker and a database~~

Question 21: ✓ Correct answer

Carefully look at the following DAG:

```
1 from airflow import DAG
2 from airflow.operators.bash import BashOperator
3
4 from datetime import datetime, timedelta
5
6 dag = DAG('my_dag', start_date=datetime(2021, 1, 1), schedule_interval='@daily', catchup=False)
7
8 task_1 = BashOperator(
9     task_id='task_1',
10    bash_command='ls',
11    start_date=datetime(2021, 1, 2),
12    dag=dag
13 )
14
15 task_2 = BashOperator(
16     task_id='task_2',
17     bash_command='pwd',
18     start_date=datetime(2020, 1, 3),
19     dag=dag
20 )
```

Is there something wrong?

- ☐ ~~Yes, too many start dates~~
- ☒ **No, you can have different start dates**

Question 22: ✓ Correct answer

Every morning you're anxious about your tasks having failed during the night. Indeed, data scientists, managers, data analysts count on you to get their data in time and do their analytics. Problem, every morning you repeat the same manual steps to check the files, the scripts, the outputs of those scripts and if everything is stored where it should be. Is this could be a use case for Airflow?

- ☒ **Absolutely!**
- ☐ ~~Grafana would be more suitable~~
- ☐ ~~No, I don't think so~~

Question 23: ✓ Correct answer

Looks like the execution date you get on the UI mismatch with your current date. Hm... is it normal?

- ☒ **Yes, dates are displayed in UTC by default**
- ☐ ~~No, there is something wrong~~

Question 24: ✓ Correct answer

Look at the DAG below:

```

1 from airflow import DAG
2 from airflow.operators.python import PythonOperator
3
4 from datetime import datetime
5
6 def _extracting():
7     return ['data']
8
9 def _processing():
10    return ['processing']
11
12 dag = DAG('my_dag', start_date=datetime(2020, 1, 1), schedule_interval='@daily', catchup=False)
13
14 task_1 = PythonOperator(
15     task_id='task_1',
16     python_callable=_extracting,
17     dag=dag
18 )
19
20 task_2 = PythonOperator(
21     task_id='task_2',
22     python_callable=_processing,
23     dag=dag
24 )

```

Is there a way to make the code cleaner by avoiding to add the DAG object into all operators?

- ☐ ~~No we can't~~
- ☐ ~~Yes, by creating a default argument dictionary~~
- ☒ **Yes, by instantiating the DAG object with the context manager "with"**

Question 25: ✔ Correct answer

Let's take a look at the beautiful DAG below:

```

1 from airflow import DAG
2 from airflow.operators.python import PythonOperator
3
4 from datetime import datetime
5
6 def _extracting():
7     return 'some data'
8
9 def _cleaning():
10    print('should clean the data here')
11
12 dag = DAG('my_dag', start_date=datetime(2020, 1, 1), schedule_interval='@daily', catchup=False)
13
14 task_1 = PythonOperator(
15     task_id='task_1',
16     python_callable=_extracting,
17     dag=dag
18 )
19
20 task_2 = PythonOperator(
21     task_id='task_2',
22     python_callable=_cleaning,
23     dag=dag
24 )
25
26 task_1 >> task_2

```

Is there something wrong in that DAG?

☒ **No**

☐ ~~Yes~~

Question 26: ✖ Incorrect answer

You've just discovered an issue in your DAG but 5 DAGRuns have already been completed. How can you run them again from the UI?

☐ **Browse → Admin → Dag Runs → Select the 5 DAGRuns → Actions → Clear the state**

☒ ~~Browse → Admin → Dag Runs → Select the 5 DAGRuns → Actions → Delete~~

☐ ~~We can't, we have to use the CLI~~

Question 27: ✔ Correct answer

You're writing a script that will download data from an API and store it in a database on a daily basis. Could this be a use case for Airflow?

☒ **Absolutely!**

☐ ~~Cron is better~~

☐ ~~No~~

Question 28: ✔ Correct answer

Let's say you live in New York, timezone UTC-5. If you define the start_date parameter to datetime(2021, 1, 1). Will the DAG be triggered the 2021/01/01 at 00:00 in New York?

☒ **No**

☐ ~~Yes~~

Question 29: ✔ Correct answer

You've just installed Airflow 2.0 with pip, well done! Now for security reasons, you can't use the default folder (~/airflow) for Airflow's home. You're only allowed to use /opt/. How can you change the home to the allowed path?

☐ ~~You can go to /opt/ then execute then initialize Airflow~~

☐ ~~You create a new user being able to create a folder airflow in~~

☒ **You export the environment variable AIRFLOW_HOME="/opt/"**

Question 30: ✔ Correct answer

Let's take a look at the following DAG:


```

1 from airflow import DAG
2 from airflow.operators.bash import BashOperator
3
4 from datetime import datetime, timedelta
5
6 default_args = {
7     'start_date': datetime(2020, 1, 1),
8     'retries': 3,
9     'retry_delay': timedelta(minutes=3)
10 }
11
12 with DAG('my_dag', schedule_interval='@daily', default_args=default_args, catchup=False) as dag:
13
14     cleaning = BashOperator(
15         task_id='cleaning',
16         bash_command='exit 1'
17     )

```

What will be the status of the task cleaning after having failed for the first time?

- ☐ failed
- ☐ ~~upstream_failed~~
- ☒ **up_for_retry**

Question 31: ✔ Correct answer

Airflow is up and running, the scheduler is ready to trigger tasks, the UI is accessible on port 8080 and the metadata database has been initialized. You even see data in some tables. However, something is missing. Your pipelines! Where do they go?

- ☐ ~~In the folder plugins/~~
- ☒ **In the folder dags/**
- ☐ ~~In the folder include/~~

Question 32: ✘ Incorrect answer

Let's assume we have a DAG that is scheduled to run daily. The start date is January 1st, 2021. What is the execution date for the first DAG Run?

- ☒ ~~2021/01/02 00:00~~
- ☐ **2021/01/01 00:00**
- ☐ ~~2021/01/02 23:59~~

Question 33: ✔ Correct answer

As your company grows, so your number of tasks. Recently, you added some tasks to one of your DAGs but now it is taking too much time to complete. What is the best view to spot any bottleneck in your DAG?

- ☐ ~~Tree view~~
- ☐ ~~Graph view~~
- ☒ **Gantt view**

Question 34: ✓ Correct answer

The Celery Executor is great to execute as many tasks as you want. If you need more resources, you add new machines. However, it comes at a price of more complexity. Indeed, do you remember what compose the Celery Executor?

☒ **In addition to the web server, scheduler and database, we have to set up a queue broker, a result backend and workers.**

☐ ~~In addition to the web server, scheduler and database, we have to set a queue broker~~

☐ ~~In addition to the web server, scheduler and database, we have to set a result backend and workers.~~

Question 35: ✓ Correct answer

One of your teammates coded a DAG but there is an issue. Indeed, it doesn't get triggered at the current date. We are the 1st of February 2021 and the start date is datetime(2021, 2, 1) with a schedule interval defined to 15 mins. Why the DAG doesn't get triggered?

☒ **In Airflow, a DAG is triggered after the start date + the schedule interval**

☐ ~~In Airflow, a DAG is triggered after the start date~~

☐ ~~In Airflow, we first have to trigger the DAG manually before it gets scheduled automatically~~

Question 36: ✓ Correct answer

What is the typical journey of a task?

☒ **No status → Scheduled → Queued → Running → Success**

☐ ~~Scheduled → Queued → Running → Success~~

☐ ~~Queued → Scheduled → Running → Success~~

Question 37: ✗ Incorrect answer

You've read carefully the documentation and you saw that you can't execute multiple tasks with the Sequential Executor due to the limitations of SQLite. What other database(s) could you use instead?

☒ **MariaDB (with some limitations)**

☒ **Postgres**

☒ ~~Cassandra~~

Question 38: ✓ Correct answer

You've got the DAG below:

```

1 from airflow import DAG
2 from airflow.operators.bash import BashOperator
3
4 from datetime import datetime, timedelta
5
6 dag = DAG('my_dag', start_date=datetime(2020, 1, 1), schedule_interval='@daily', catchup=False)
7
8 task_1 = BashOperator(
9     task_id='task_1',
10    bash_command='ls',
11    retries=3,
12    retry_delay=timedelta(minutes=3),
13    dag=dag
14 )
15
16 task_2 = BashOperator(
17     task_id='task_2',
18     bash_command='pwd',
19     retries=3,
20     retry_delay=timedelta(minutes=3),
21     dag=dag
22 )
23
24 task_1 >> task_2

```

As a meticulous engineer, you've noticed that the arguments `retries` and `retry_delay` share the same values for all operators.

Is there a way to avoid that and make the code cleaner?

- ☒ **Yes, by defining default arguments in the DAG objects**
- ☐ ~~Yes, by adding `retries` and `retry_delay` parameters in the DAG object~~
- ☐ ~~No, we there isn't~~

Question 39: ✔ Correct answer

Oops! There is a bug in your DAG and you need to fix it. You pause the DAG but unfortunately, the error took 3 days to fix. That DAG is scheduled to be triggered every day and has a start date 2021/01/05 00:00. The current date is the 2021/01/08 10:00. If you schedule the DAG again, how many running DAGRuns will you get?

- ☐ 4
- ☐ 2
- ☒ **3**

Question 40: ✔ Correct answer

You work in a Bank. In order to process awaiting transfers, your DAG has to be triggered every 4 hours. You've defined the start date to `datetime(2021, 1, 1)`. What would be the value of the schedule interval?

- ☐ ~~`schedule_interval="0 4 * * *"`~~
- ☒ **`schedule_interval=timedelta(hours=4)`**
- ☐ ~~`schedule_interval="4"`~~

Question 41: ✔ Correct answer

You start scheduling the following DAG for the first time:

```
1 from airflow import DAG
2 from airflow.operators.bash import BashOperator
3
4 from datetime import datetime, timedelta
5
6 dag = DAG('my_dag', start_date=datetime(2020, 1, 1), schedule_interval='@daily', catchup=False)
7
8 task_1 = BashOperator(
9     task_id='task_1',
10    bash_command='ls',
11    dag=dag
12 )
13
14 task_2 = BashOperator(
15     task_id='task_2',
16     bash_command='pwd',
17     dag=dag
18 )
19
20 task_1 >> task_2
```

How many DAGRuns will you end up with right after scheduling it?

- ☒ 1
- ☐ 0
- ☐ 400

Question 42: ✓ Correct answer

You just added a new DAG file into your dags folder, but it doesn't show up on the UI. The code is perfect, as it has passed a rigorous code review, and the reviewer even commented that you are the best Airflow engineer they have ever seen! What's going on?

- ☐ You have to restart Airflow for the DAG to show up on the UI
- ☐ Your reviewer lied. You made a mistake in the DAG code, and he wanted to see you panic
- ☒ You need to wait 30 seconds, as there can be up to a 30 second delay before the webserver acknowledges the new DAG file.

Question 43: ✓ Correct answer

The folder dags exists, you can put python files corresponding to your data pipelines in it. You are ready to create your first DAG. But, do you know exactly what a DAG is?

- ☐ DAG stands for Directed Acyclic Graph. It is a graph with nodes corresponding to the tasks, directed edges corresponding to the dependencies between tasks. A DAG can have loops and represents a data pipeline in Airflow
- ☒ DAG stands for Directed Acyclic Graph. It is a graph with nodes and directed edges, respectively defining the tasks and dependencies between those tasks. There is no loop. A DAG represents a data pipeline in Airflow.
- ☐ DAG stands for Directed Acyclic Graph. It is a graph with nodes and edges. Nodes are tasks, edges are dependencies. The dependencies do not define the order in which the tasks are executed.

Question 44: ✓ Correct answer

To avoid having too many DAGRuns running at the same time, you've decided to define the parameter `catchup_by_default` to `False`. Can you still backfill the data even if the catchup parameter is turned off?

- ☐ ~~Yes, with the REST API~~
- ☒ **Yes, with the command line interface**
- ☐ ~~No, when catchup is set to False, we can't backfill the missing DAGRuns.~~

Question 45: ✓ Correct answer

Let's say you have two tasks. Extracting your data and cleaning your data. Should put them into one operator or two distinct operators?

- ☐ ~~One operator~~
- ☒ **Two distinct operators**

Question 46: ✓ Correct answer

Super important concept, what is a DAGRun?

- ☒ **An instance of a DAG along with tasks to run and an execution date**
- ☐ ~~An instance of a task with an execution date~~
- ☐ ~~An object grouping all DAGs~~

Question 47: ✓ Correct answer

You've created an Airflow variable with a sensitive value. How can you hide from the UI?

- ☐ ~~We can't hide it~~
- ☒ **By putting "secret" in the name**
- ☐ ~~Whenever we have a sensitive value, it's better to create a connection~~

Question 48: ✓ Correct answer

What is an executor?

- ☐ ~~It's where tasks are executed~~
- ☒ **It defines how your tasks are executed, on which system**
- ☐ ~~It schedules your tasks~~

Question 49: ✓ Correct answer

Let's assume we have a DAG that is scheduled to run daily. The start date is January 1st, 2021. What is the execution date for the second DAGRun?

- ☒ **2021/01/02 00:00**
- ☐ ~~2021/01/01 00:00~~
- ☐ ~~2021/01/03 00:00~~

Question 50: ✓ Correct answer

You can control whether a DAG is paused or unpaused by clicking on the toggle of a specific DAG from the UI. Can you do it from the command line interface?

☒ **Yes**

☐ No

Question 51: ✓ Correct answer

When you create a DAG, there are some parameters that are absolutely crucial to understand. To illustrate the first one, what is the best practice around dag_id?

☐ ~~Nothing special, we can have multiple dags with the same dag id~~

☒ **We have to make sure the dag id is unique across all dags**

☐ ~~The dag id must be a number and not a string~~

Question 52: ✓ Correct answer

Your tasks are executed with the CeleryExecutor among 3 machines. Machines A, B and C. After having added a new task with the MySqlOperator you got a dependency issue on machine A. You installed the dependency on Machine A then, retry the task and got the same dependency issue. What the easiest way to fix the issue?

☒ **Install the dependency on all machines A, B and C**

☐ ~~Restart Airflow on all machines to get the modifications work~~

☐ ~~You have to create a queue to execute tasks only on machine A~~

Question 53: ✓ Correct answer

What is the role of a fernet key in Airflow?

☒ **Encrypt variables**

☒ **Encrypt connection passwords and extra values**

☐ ~~Access the UI~~

Question 54: ✗ Incorrect answer

Tasks of your DAGs are not getting executed. You've manually triggered it from the UI. You don't see any error, just the DAGRun is in green, like it is running, but no tasks get triggered. What could be the main cause?

☒ ~~There is an error in your code~~

☐ **You didn't turn on the toggle of the DAG**

☐ ~~The first task is getting too long~~

Question 55: ✓ Correct answer

There is a DAG that you doesn't need anymore. Therefore, you click on the red basket. What happens then?

☐ ~~All metadata related to the DAG is removed from the database. The file corresponding to the DAG is deleted.~~

☒ All metadata related to the DAG is removed from the database.

☐ The file corresponding to the DAG is deleted, only the history of the DAGRuns stay.

Question 56: ✔ Correct answer

You've just created a DAG and you need to process data one month before the current date. What is the best way to do that?

☒ airflow dags backfill

☐ airflow dags trigger

☐ we can't, it has to start at the current date

Question 57: ✔ Correct answer

It's a wonderful day! You've just got a call telling you that 5 machines are available for executing your tasks with Airflow. AWESOME! But wait a minute, which executor is the perfect one to benefit from those new machines?

☐ The Local Executor

☐ The Sequential Executor

☒ The Celery Executor

Question 58: ✔ Correct answer

To connect to your AWS S3 bucket you have to create a connection. One way is to configure that connection is by providing both an access key and a secret access key.

If you put the keys in the extra field:

Edit Connection

Conn Id *

Conn Type *

Description

Host

Schema

Login

Password

Port

Extra

Will the keys be hidden from the UI?

☐ Yes, like with password

☒ No, creating environment variables with the keys is more secure

Question 59: ✓ Correct answer

You have a use case where the first three tasks should be upstream to three downstream tasks. That means,

- A, B, C upstream to D;
- A, B, C upstream to E;
- A, B, C upstream to F.

What is the most efficient way to do that?

- ☐ ~~[A, B, C] >> [D, E, F]~~
- ☒ **cross_downstream([A, B, C], [D, E, F])**
- ☐ ~~[A, B, C] >> D; [A, B, C] >> E; [A, B, C] >> F~~

Question 60: ✗ Incorrect answer

You have two tasks. One downloading filenames and a second task in charge of getting the files corresponding to those filenames. As you already got the list of filenames from the first task, you want to push that list into the second task using XCOMs. What is the fastest and easiest way to do that? (Both tasks use the PythonOperator)

- ☐ **By returning the list from the python callable function**
- ☒ ~~By executing the method xcom_push~~
- ☐ ~~By executing the method xcom_pull~~

Question 61: ✓ Correct answer

When you create a connection in Airflow and the fernet key is undefined. Can you see the password and extra field values in plain text from the database?

- ☒ **Yes**
- ☐ ~~No~~

Question 62: ✓ Correct answer

This is it! You made your choice. You're going to install Airflow in your company! 🎉 But, wait a second. The DevOps team just sent you a message asking for the minimum requirements to get Airflow up and running on a Linux Ubuntu OS. What are they?

- ☐ ~~Python 2.7 is enough with pip and an Internet access. A constraint file for 2.7 and some system level packages updated/installed~~
- ☐ ~~At least python 3.6. Pip with an internet access. Some system level packages updated/installed~~
- ☒ **At least python 3.6. Pip with an internet access. Some system level packages updated/installed and preferably a constraint file**

Question 63: ✓ Correct answer

What is the equivalent of these dependencies A >> B >> C >> D

- ☐ ~~A << B << C << D~~
- ☐ ~~[A, B, C, D]~~

☒ D << C << B << A

Question 64: ✘ Incorrect answer

You've coded your first DAG! Well done!

```
1  from airflow import DAG
2  from airflow.operators.python import PythonOperator
3  from airflow.operators.bash import BashOperator
4
5  from datetime import datetime
6
7  def _check():
8      print('checking my data')
9
10 dag = DAG('my_dag', schedule_interval='@daily', catchup=False)
11
12 task_1 = PythonOperator(
13     task_id='task_1',
14     python_callable=_check,
15     dag=dag
16 )
17
18 task_2 = BashOperator(
19     task_id='task_2',
20     bash_command='echo "hello here"',
21     dag=dag
22 )
```

However, is there something wrong here?

☐ Yes

☒ No

Question 65: ✘ Incorrect answer

You've just created the following DAG:

```

1 from airflow import DAG
2 from airflow.operators.bash import BashOperator
3
4 from datetime import datetime, timedelta
5
6 default_args = {
7     'start_date': datetime(2020, 1, 1),
8     'retries': 3,
9     'retry_delay': timedelta(minutes=3),
10    'email_on_failure': True
11 }
12
13 with DAG('my_dag', schedule_interval='@daily', default_args=default_args, catchup=False) as dag:
14
15     cleaning = BashOperator(
16         task_id='cleaning',
17         bash_command='exit 1'
18     )

```

and you want to be alerted by email to handle task failures.

If the task "cleaning" fails, how many emails will you receive (look carefully) ?

- ☐ 0
- ☐ 3
- ☒ 4

Question 66: ✔ Correct answer

Once you've installed Airflow with Pip and defined the home of Airflow, there is one command that you have to execute first before any other. What is that command?

- ☒ **airflow db init**
- ☐ ~~airflow db check~~
- ☐ ~~airflow db reset~~

Question 67: ✘ Incorrect answer

You work in a Bank. Your goal is to catch possible fraudulent transfers in real time. As soon as a transfer comes in, your machine learning model defines if it should be blocked or not. Could this be a use case for Airflow?

- ☒ ~~Of course!~~
- ☐ **No**

Question 68: ✔ Correct answer

Based on the following DAG:

```

1 from airflow import DAG
2 from airflow.utils.dates import days_ago
3 from airflow.operators.bash import BashOperator
4
5 from datetime import datetime, timedelta
6
7 default_args = {
8     'start_date': days_ago(5),
9     'retries': 3,
10    'retry_delay': timedelta(minutes=3),
11    'email_on_failure': True
12 }
13
14 with DAG('my_dag', schedule_interval='@daily', default_args=default_args, catchup=True) as dag:
15
16     dummy = BashOperator(
17         task_id='dummy',
18         bash_command='exit 0'
19     )

```

How many DAGRuns will run as soon as the DAG is scheduled?

- ☐ 4
- ☐ 0
- ☒ 5

Question 69: ✓ Correct answer

Oh! You just got your first task in failure. What should you do to debug and retry it?

- ☒ Click on the task → Logs → Fix the issue → Click on the task → Clear
- ☐ Click on the task → Logs → Fix the issue
- ☐ Click on the task → Clear

Question 70: ✓ Correct answer

You work at Netflix (well done 😊). You are in charge of building a recommendation system. In order to recommend the next series/movie to watch for customers, you have to process terabytes of data. Is this could be a use case for Airflow?

- ☐ Yes, we can process this data in Airflow
- ☒ Yes, we can process this data by triggering Spark jobs
- ☐ No, we can't

Question 71: ✓ Correct answer

Can I use the email_on_success parameter to receive an email if a task succeeds?

- ☐ Yes, absolutely!
- ☒ No, it doesn't exist

Question 72: ✗ Incorrect answer

You want to be alerted when your tasks fail, what is/are the best option(s) for doing so?

- ☒ **email_on_failure**
- ☒ **on_failure_callback**
- ☐ ~~email_on_retry~~
- ☒ ~~sla~~

Question 73: ✔ Correct answer

Your company is running a website. Each time a customer hit a button to validate a form, a specific DAG gets triggered. Therefore, that DAG must not be scheduled. How can you do that?

- ☐ ~~schedule_interval="@once"~~
- ☒ **schedule_interval=None**
- ☐ ~~schedule_interval=""~~

Question 74: ✔ Correct answer

There is a new Airflow version available! What command should you run to update your instance?

- ☐ ~~airflow db init~~
- ☒ **airflow db upgrade**
- ☐ ~~airflow db reset~~

Question 75: ✔ Correct answer

There are different types of operators. You just got a use case where you have to wait for some files before moving to the processing task in your DAG. Which type of operator is the most appropriate?

- ☐ ~~Action Operators~~
- ☒ **Sensors**
- ☐ ~~Transfer Operators~~



PRODUCT	RESOURCES	COMPANY
Astro	Docs	About Us
Status	Registry	Careers
Support	Guides	Contact
Pricing	Webinars	Partners
Why Airflow	Academy	
	Forum	
	Blog	
	Weekly	
	Demo	

Apache Airflow, Airflow, and the Airflow logo are trademarks of the Apache Software Foundation.