



Faça login em Medium com o Google



Gustavo Santos

progustavosantos@gmail.com

Continuar como Gustavo

Certificação Airflow



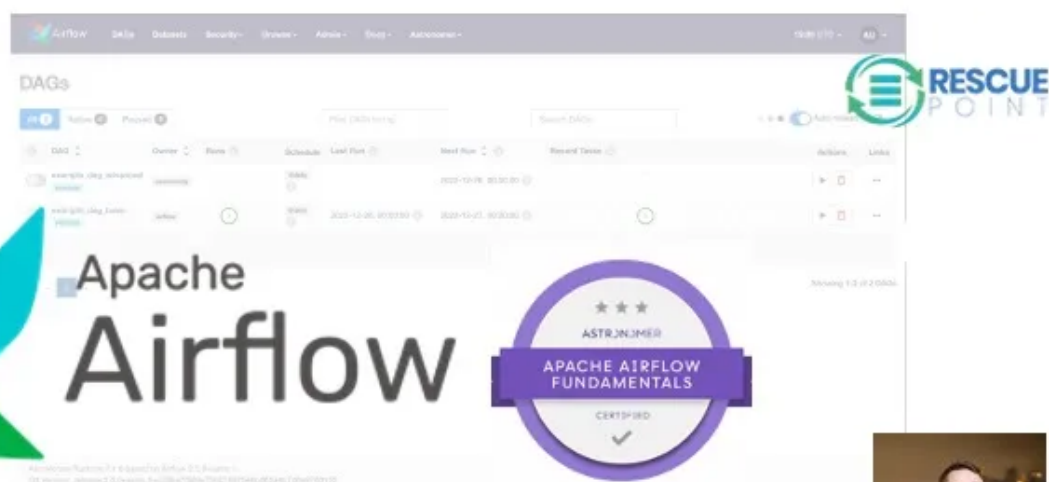
Anselmo Borges · Follow

Published in Rescue Point

14 min read · Dec 27, 2022

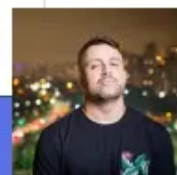


Share



Certificação Airflow Fundamentals (Parte 1/2)

Um guia com um resumo do material para a certificação de Airflow da Astronomer, na primeira de 2 partes, como subir o ambiente, arquitetura do Airflow, overview sobre o Funcionamento e visualizações de suas DAGs de forma eficiente na UI.



Anselmo Borges
DataOps Engineer
Microsoft MCT

Primeira parte do meu resumo pra certificação Airflow fundamentals.

Primeiramente

Esse é um resumo foi feito por mim com base no curso do curso de Airflow Fundamentals da Astronomer, que é uma empresa que tem uma versão suportada do AirFlow, lembrando que o AirFlow pode ser instalado Open Source pois é um projeto da Apache e assim como outras tecnologias como o Spark, Cassandra, Hadoop e outros existem empresas que encapsulam a solução e vendem o suporte.

Porque isso existe?

Porque se você coloca algo open source na sua empresa você precisa segurar o B.O de suporta-la e isso implica em de vez em quando cair em alguns Bugs e depender da comunidade pra resolver. Imagina, você colocou em produção, economizou uma bala, seu chefe pirou, mas você pegou um bug de desenvolvimento. Pra quem você vai chorar? Esperar a comunidade te ajudar no tempo deles? Sem SLA? As vezes o que você não paga de licença corre um risco como esse e tudo isso tem que ser posto na balança durante a implantação.

Varias empresas fazem isso como Cloudera (Hadoop e outras soluções), DataStax (Cassandra), Confluent (Kafka), Databricks (Spark e Delta Lake) e uma das mais conhecidas pra que você saiba que isso existe faz tempo é a Red Hat (Linux), só fechando o assunto, eles encapsulam o Open Source em versões estáveis e cobram o suporte caso queira usá-las.



Tem que explicar né? rs

Fiz questão de deixar isso claro pois tem gente que nem faz idéia disso, o cara ouve o nome e quer implantar, rs.

O curso

Esse curso se encontra no site da [Astronomer.io](https://astronomer.io) conforme disse anteriormente, é em inglês e pra fazer a prova de certificação fiz um resumo em português e pode ser útil caso você tenha dificuldade na língua inglesa. Dividi esse material em 2 partes, até pra que o post não vire uma obra de Tolkien. Peço desculpas caso você encontre algo errado, pois posso ter entendido errado, se puder leia esse post e faça o treinamento pra que fique tudo certinho, blza?

Como instalar na sua máquina?

Pra que você possa realizar fazer esse mini curso de forma síncrona e consiga testar todas as etapas, recomendo a configuração do `astro cli` inicialmente e ter o docker instalado na sua maquina pois o astro cli vai fazer a instalação dos containers necessários pra que você possa realizar o treinamento.

Instalando o Astro CLI conforme seu sistema operacional

Segue o link da documentação que ensina como instalar o `astro cli` no seu computador de acordo com o sistema operacional que está usando.

Install the Astro CLI | Astronomer Documentation

Install the Astro CLI on a Mac operating system with a single command.

docs.astronomer.io

Instalando o Docker

Como disse você vai precisar do Docker instalado no seu computador pra que consiga subir o Airflow devidamente configurado para esse treinamento, mas lembrando que vai ser um Single node com tudo configurado, apenas pra fins de treinamento. Em ambientes produtivos a arquitetura é completamente outra e darei uns exemplos mais a diante.

Segue um link com uma documentação que te ajuda a instalar o Docker Desktop no seu computador.

Get Docker

Docker Desktop terms Commercial use of Docker Desktop in larger enterprises (more than 250 employees OR more than \$10...

docs.docker.com

Preparando o ambiente

Com o Docker e o Astro CLI instalados no seu ambiente você vai rodar os seguintes comandos abaixo (coloquei em shell mais você adapta pro seu sistema operacional ae):

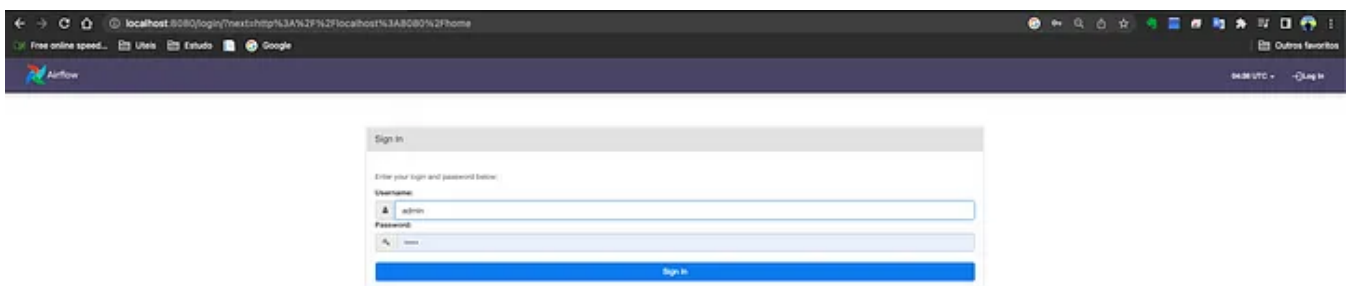
```
# Crie uma nova pasta chamada astro e entre nela
mkdir astro
cd astro

# Rode o comando abaixo e ele vai baixar os arquivos para que a instalação
# do Airflow seja realizada (lembre-se que você precisa do Docker)
astro dev init

# Rode o comando abaixo e com base nos arquivos baixados no comando anterior
# será iniciado o download das imagens e subir os containers no Docker
astro dev start

# Rode o comando abaixo para ver se está tudo rodando conforme deve
astro dev ps
```

Feito isso você pode acessar o Airflow do seu navegador usando o endereço <http://localhost:8080> e a telinha abaixo será exibida



para logar use admin como usuário e admin como senha

Apache Airflow

O Airflow é um orquestrador de jobs de dados ETL(extract, transform and load), com uma ampla variedades de soluções para que você consiga alem de executar o seu job de forma correta poder fazer isso de forma escalavel, monitoravel e controlavel de diversos modos:

- Via UI (User Interface)
- Via CLI (command line interface)
- Via Rest API (requisições usando HTTP)

Um dos exemplos bem legais usados no curso é que se eu tiver um front end, posso criar um botão que chama uma REST API que pode controlar meu job conforme a minha necessidade.

Componentes básicos do AirFlow

Existem componentes básicos do AirFlow:

- **Web Server:** Ele é o responsável pela exibição do UI pra que você controle e monitore seus jobs via interface Web.
- **Scheduler:** O Agendador do Airflow e ele é muito importante por que sem ele você não consegue agendar ou executar nenhum job no AirFlow por isso no caso ele é o coração da solução já que ela é feita pra isso, orquestrar seu jobs. Existe inclusive a possibilidade de você ter mais de um Scheduler rodando no mesmo cluster o que traz uma alta disponibilidade já que se um scheduler não estiver disponível, você pode executar em outro.
- **MetaData Database:** Todos os dados relacionados aos usuários, jobs, conexões, qualquer dado relacionado ao AirFlow, está armazenado no Metadata Database e qualquer base que suporta SQL pode ser usada como metadata database, no meu caso estou usando postgres mas poderia ser MySQL, Oracle, SQL Server e ate um mongo DB que não é um banco parecido com os demais por isso não é 100% recomendado devido a algumas limitações.
- **Executor:** Define como suas tasks serão executadas pelo AirFlow, por exemplo, se você tem um servidor Kubernetes e precisa executar sua task nele, você vai precisar de um Kubernetes executor, se for rodar em um Celery cluster com multiplas maquinas você vai ter que usar o Celery executor, se você tem uma maquina extremamente poderosa e quer executar varias tasks nela, você vai precisar de um local executor, ou seja, pra cada ambiente que você for rodar sua task um executor especifico será necessário. Um ponto importante do executor que ele executa sequencialmente, um após o outro.

- **Worker:** Quando o executor é definido o worker é o cara que está executando ela onde você definiu, ele pode ser um processo ou um subprocesso. Um exemplo é quando você por exemplo rodou uma execução no kubernetes, dentro de um POD a PID é o processo do worker que está executando a sua task. Outro exemplo, caso esteja executando uma task localmente que está usando múltiplos processos, cada processo desse é um worker.
- **Queue:** Um serviço de fila onde eu gerencio a ordem de execuções dessas atividades, pode ser um RabbitMQ ou um Redis, no ambiente de exemplo não configuramos um.

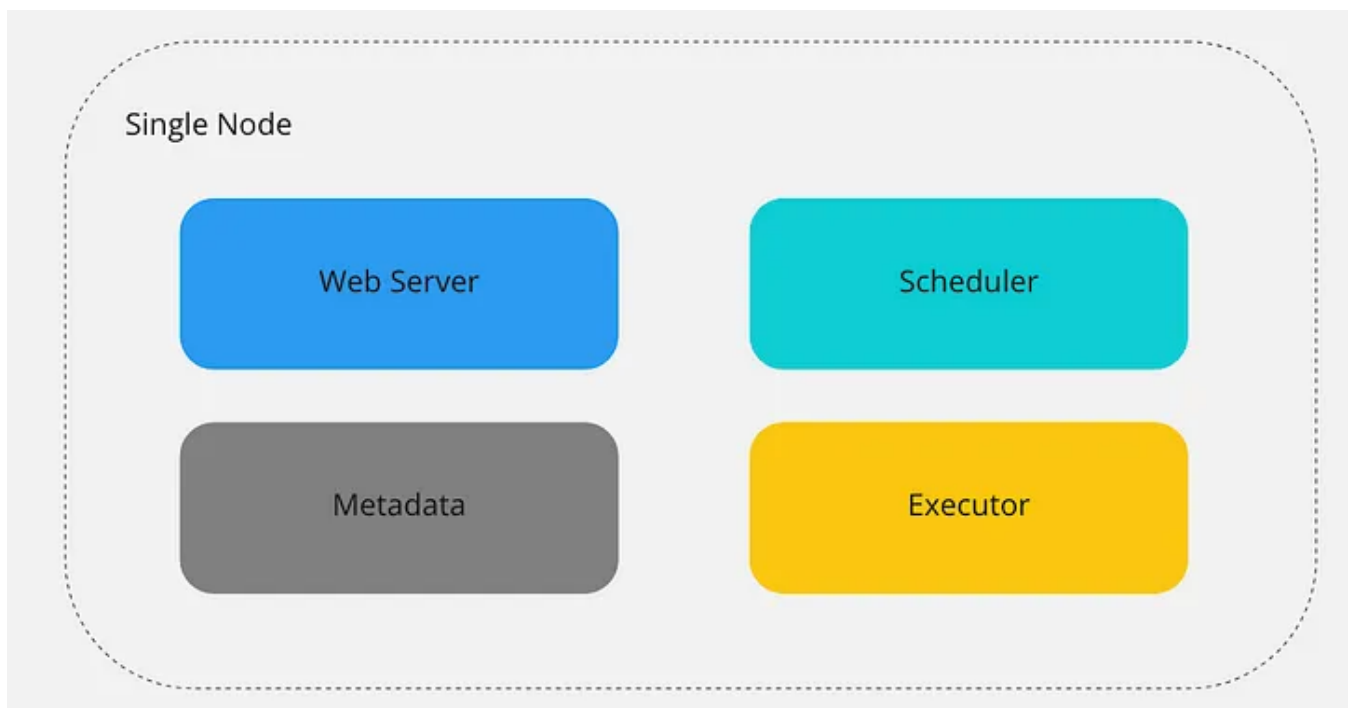
Como eles funcionam juntos?

Nesse ponto vou explicar a arquitetura de como esses componentes trabalhando em conjunto. Existem algumas possibilidades mas os próximos exemplos vão se basear em somente 2 delas

Single node

Quando todos os componentes estão sendo usados na mesma máquina e seriam WebServer, Scheduler, Metadata e o Executor (o Worker não aparece pois ele é o fruto da execução).

O Webserver, Scheduler e Executor estão sempre interagindo com o Metadata, pois o Webserver precisa de informações dele pra exibição e controle, o Scheduler as informações dos jobs, se eles rodaram ou não, parâmetros entre outras informações e por sua vez o executor também consulta e grava informações sobre andamentos dos jobs e coisas do gênero. Pra começar o uso é legal uma arquitetura dessa mas tudo no mesmo cluster gera um risco em ambientes de produção, caiu o cluster, acabou a brincadeira.

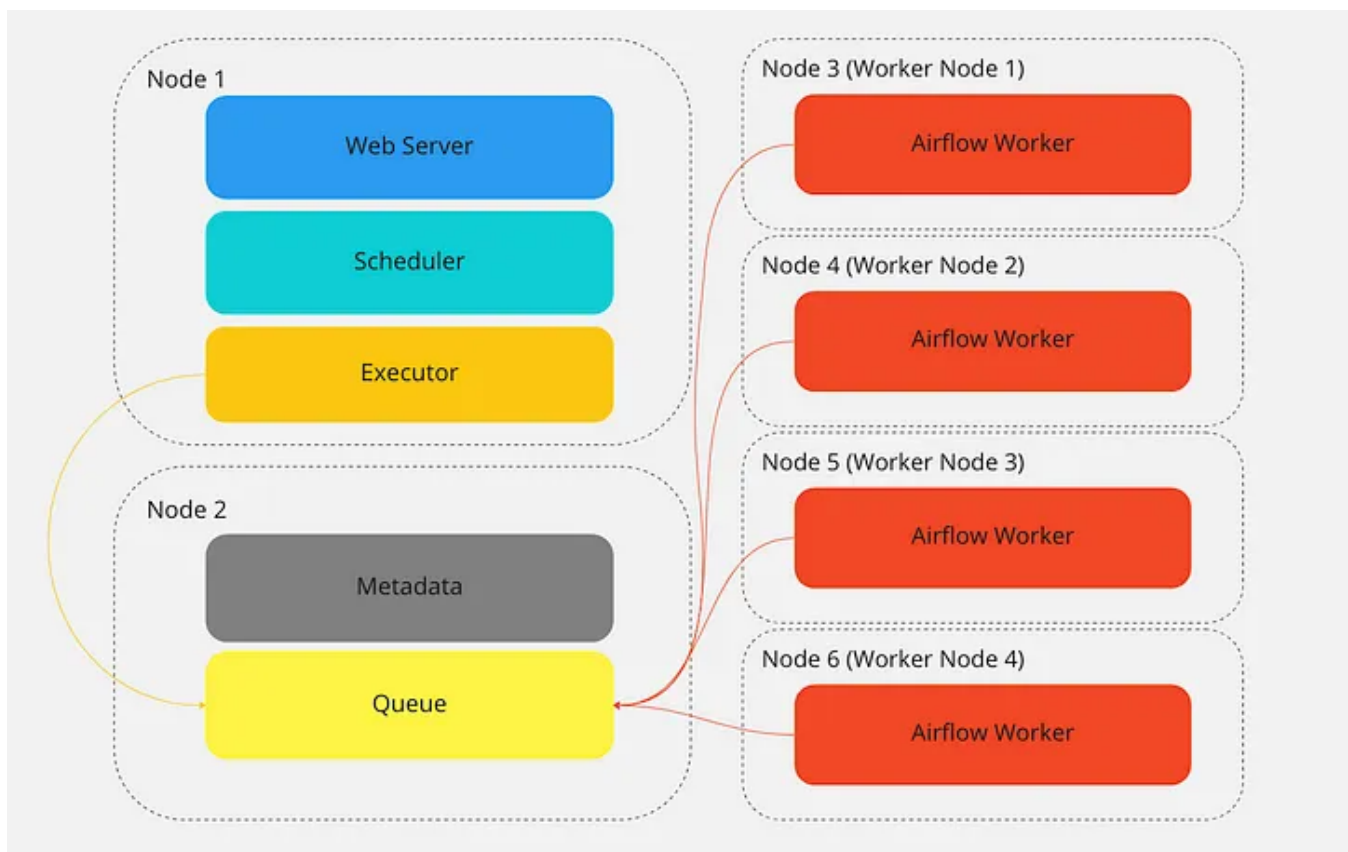


Nesse exemplo todos os componentes encontram-se na mesma maquina (node)

Multi nodes

Uma arquitetura bem mais voltada a ambientes produtivos, com alta disponibilidade, escalabilidade, onde consigo dividir os meus core components em mais de um node e até replica-los entre eles pra que eu crie algum tipo de redundância. Esse modelo é chamado de celery conforme citado anteriormente. Nos nodes convencionais posso separa os core components por exemplo no node 1 posso ter o Webserver, o Scheduler e o Executor, no node 2 posso ter o Metadata e o serviço de Queue.

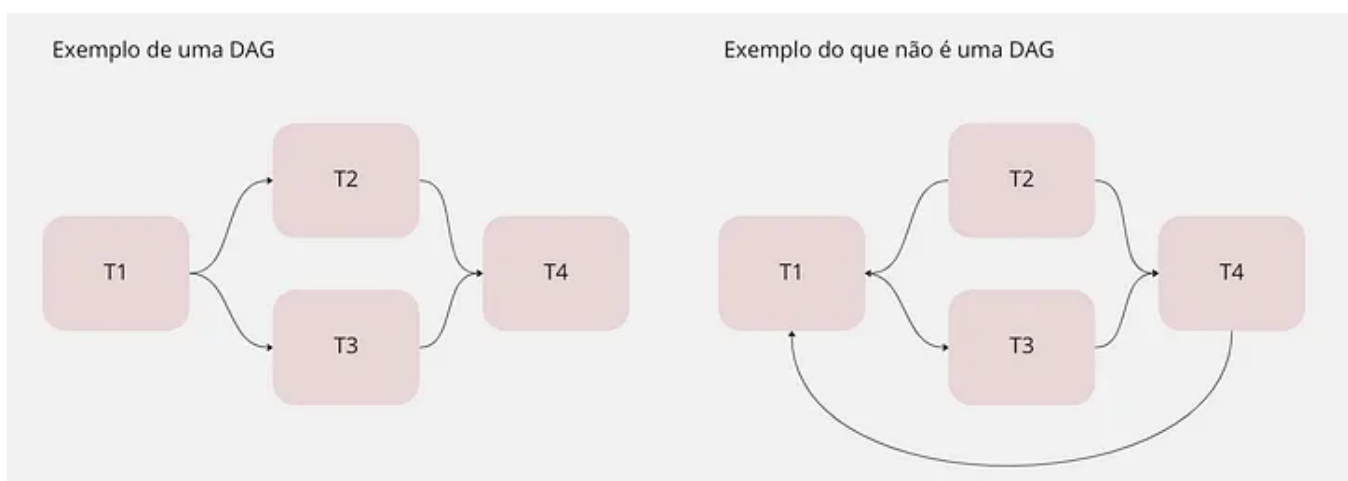
Posso executar os Workers nesse nodes, mas não ficaria performático, por isso existe a possibilidade de nesse cluster eu ainda ter os Worker nodes, maquinas especificas para execução dos workers solicitados e que me daria a possibilidade de dividi-los entre maquinas evitando sobrecargas.



nessa imagem uma arquitetura com 6 nodes dividindo recursos entre eles inclusive a fila (queue)

DAGs

DAG (Direct Acyclic Graph), uma DAG no Airflow é um Data pipeline, se você cria um Data Pipeline no Airflow, automaticamente você está criando uma DAG. Uma DAG basicamente é um fluxo de ações direcionadas, que por mais que suas ações se dividam, ele tem um ponto final onde ele se encontra, outro ponto é que o DAG nunca é um loop, ele sempre tem um destino final que orienta o caminho. Podemos falar de dependências sequenciais também pois no caso uma depende do termino da anterior para que a sequencia flua. No seu DAG sempre haverão steps dependentes um dos outros conforme desenho, esses steps são chamados de Operators



Operators

Um operador é tipo um objeto, vamos supor que ao instanciar esse objeto você diz que ele vai ler um arquivo e printar o conteúdo desse arquivo na tela. Note que esse operador é uma task do seu Data Pipeline/DAG. Os tipos de operators possíveis que são:

- **Action Operators:** Permite que você execute algo em seu Data Pipeline, por exemplo, se vc precisar executar uma função Python, você usa o python operator, bash operator para processos bash e SQL Operator para queries SQL, lembrando que todos esses como ações (actions).
- **Transfer Operators:** Como o próprio nome já diz trata-se de transferencia de dados de uma fonte para um destino definido, por exemplo a transferencia de um dado do MySQL para Presto DB usa-se o Presto Operator.
- **Sensor Operators:** É o operador responsável por aguardar um resultado para que a tarefa possa seguir. Por exemplo, o File sensor pode aguardar que um arquivo seja gerado em uma determinada pasta para que o processo tenha inicio.

Quando o Operator está atribuído a uma DAG ele se torna automaticamente uma **Task**, quando essa task é schedulada, ela se torna uma **task instance operator**.

Dependencies

Para que o Data Pipeline funciona tenho que atribuir ordem na execução dela, por sua vez, uma task deve geralmente numa sequencia iniciar a primeira por um scheduler e as demais pelas dependências da conclusão ou não da task anterior, lembre-se, DAG é um pipeline orientado que tem uma direção definida. A ordem que vincula essas dependencias são:

- `set_downstream (<<)`
- `set_upstream (>>)`

Voltaremos a falar deles mais pra frente

Workflow

Workflow é a combinação de todos esses pontos anteriores a DAG que contem operators ordenados por suas dependências, o conjunto todo compõe um belo

Workflow.

Task Life Cycle

O ciclo de vida de uma task no Airflow passa pelos seguintes componentes:

1. Vamos dar um exemplo em que tenho uma pasta chamada DAGs onde coloco meu pipelines lá dentro, vamos supor que joga um arquivo python chamado DAG.py.
2. Quando você joga o arquivo nessa pasta ele vai ser consumido por 2 dos seus recursos, o Webserver e o Scheduler. Um ponto importante desses 2 recursos referente ao arquivo que jogamos lá na pasta é que o Webserver por padrão pesquisa por novos arquivos nela a cada 30 segundos, enquanto o Scheduler faz a verificação de novos arquivos nela em um padrão de 5 minutos. Esses valores podem ser ajustados mas o padrão inicial é esse.
3. Quando a DAG é consumida pelo Web Server e pelo scheduler ela está pronta pra uso e pode ser iniciada, sendo assim o scheduler manda gravar no MetaStore um DAGRun, mas sem nenhuma informação de status pois ele não rodou ainda. Com isso uma task instance vai ser designada para execução e aguardar. Nesse ponto a task está no status de “queued”, ou seja, está na fila pronta pra ser executada.
4. Com a task instance criada, o scheduler vai mandar ela pro executor conforme o agendamento do start dela, é nesse momento que o recurso é alocado no Worker, conforme falado anteriormente. Quando ele vai pra esse ponto, ela passa do status de “queued” para “running” mostrando que agora a task está em execução.
5. Com o término da execução o executor atualiza o status da task instance no metastore para “done”, isso caso ela tenha sido executada com sucesso e com os demais status caso haja algum problema.
6. O scheduler fica como responsável em ver se a task rodou com sucesso no Metastore, se não teve nenhum problema e então o Webserver é atualizado com o novo status da Task no Metastore, basta dar um refresh na tela e pronto.

Extras e providers

Quando você instala o Airflow, ele vem com as funcionalidades necessárias pra que você use de forma básica, porém caso precise configurar algo em específico, como

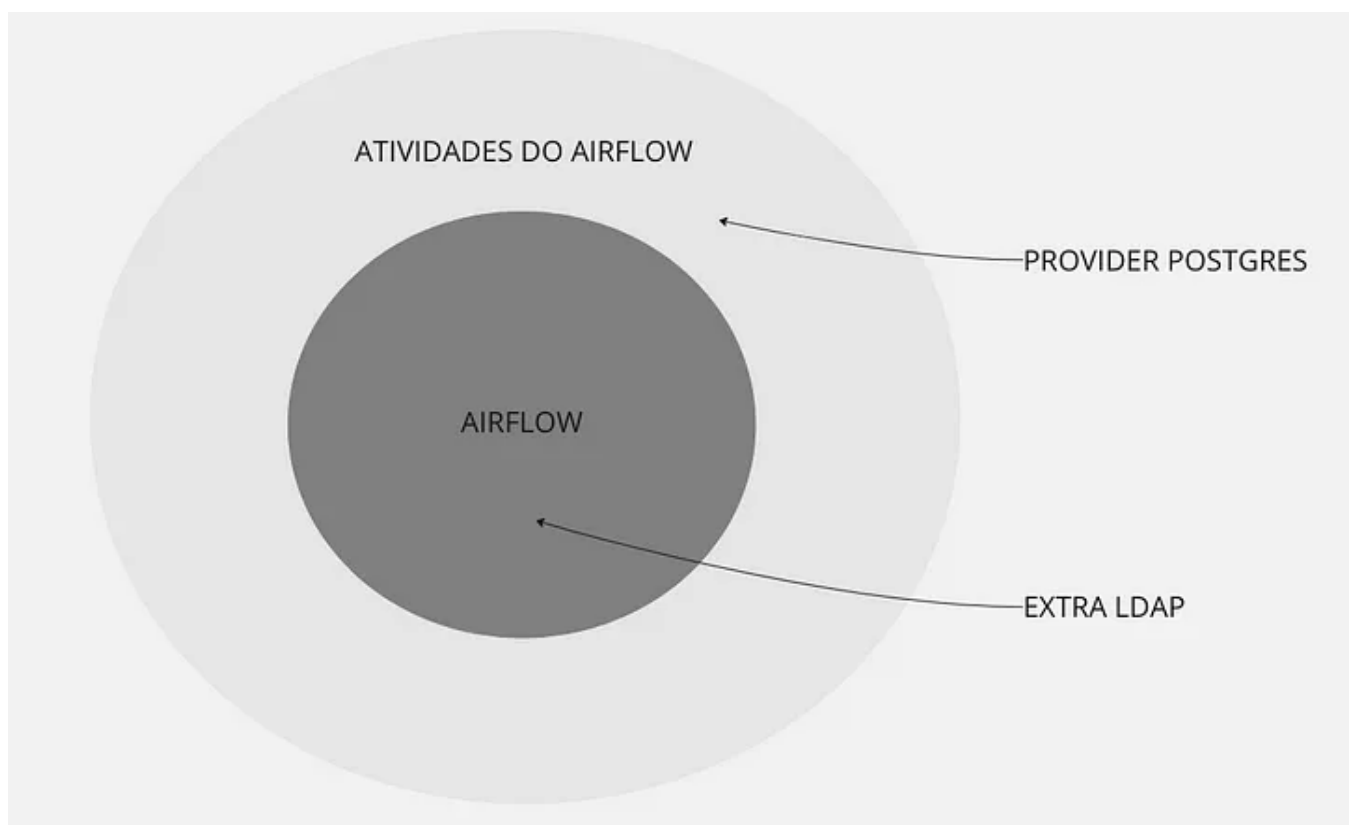
por exemplo autenticar os usuários do LDAP no Airflow, ou executar suas tasks em um celery, pra esses casos você precisa instalar extras para estender as funcionalidades padrões do Airflow.

- **Extras:** Pacotes que instalam as funcionalidades que você precisa para o funcionamento do Airflow.

Agora vamos falar de funcionalidades do seu DAG, por exemplo como a conexão com um banco Postgres (isso tem a ver com o seu Data Pipeline e não as funcionalidades core do Airflow). Pra esses casos fazemos a instalação de um provider Postgres.

- **Providers:** Não tem a ver com o core do AirFlow e sim com funcionalidades da sua DAG e assim como os Extras podem ser instalados conforme a sua necessidade.

As vantagens dos Providers é que eles podem ser atualizados de forma independente do Airflow no geral. Isso te livra da dependência da atualização ou até parada do AirFlow para suas configurações.



Exemplo de extras e providers

Meios de controle dos Data Pipelines no Airflow

Existem 3 meios de manipulação e controle dos seus data pipelines no AirFlow:

1. **UI (User interface):** Usando a interface web através do navegador é possível gerenciar e monitorar seus data pipelines no Airflow, por exemplo, se você quiser checar os logs da sua task ou se quiser ver o histórico dos seus diagrams você pode usar o UI. De longe é o método mais usado para esses tipos de tarefas no Airflow.
2. **CLI (Command line Interface):** Podemos usar ele também para alguns casos acima mas ele é extremamente útil caso você queira testar as suas tasks, caso necessite atualizar ou até inicializar o Airflow. Caso você não tenha acesso ao User Interface é uma alternativa de se usar o AirFlow.
3. **Rest API (Requisições HTTP):** Util quando você que precisa criar algo no AirFlow ou usar seu próprio front end como dado num exemplo anterior, criando um botão na sua pagina que executa alguma ação ou até iniciar uma DAG no AirFlow. Em resumo o Rest API é extremamente útil quando você quer que outras soluções interajam com o Airflow.

Vamos dar uma visão mais aprofundada de como usar cada um deles:

UI (User Interface)

Fiz um vídeo por que acho mais fácil de entender do que se eu escrevesse cada uma das funções que serão citadas, logo, segue abaixo uma visão inicial sobre a aba **DAGs**. Logue no endereço do Webserver <http://localhost:8080> com o user `admin` e a senha `admin`. Feito isso, automaticamente você vai cair na view de DAGs.

Aba DAGs

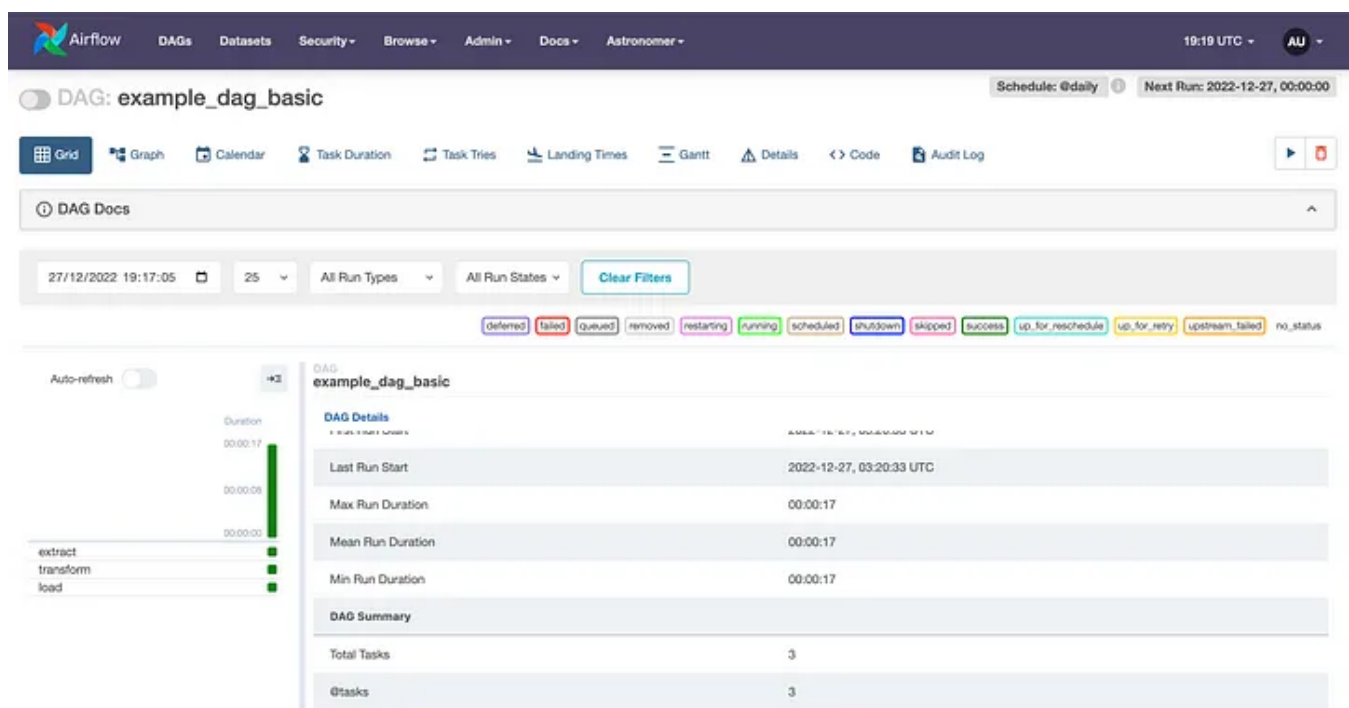
Logo que logamos caímos por padrão nessa aba, ela mostra as informações principais sobre as DAGs conforme video abaixo:

Overview sobre a aba DAGs

Como foi visto, dentro da Aba DAGs eu tenho algumas possibilidade de ver informações mais detalhadas sobre meus Data Pipelines, vamos dar uma aprofundada em cada um deles.

Grid View

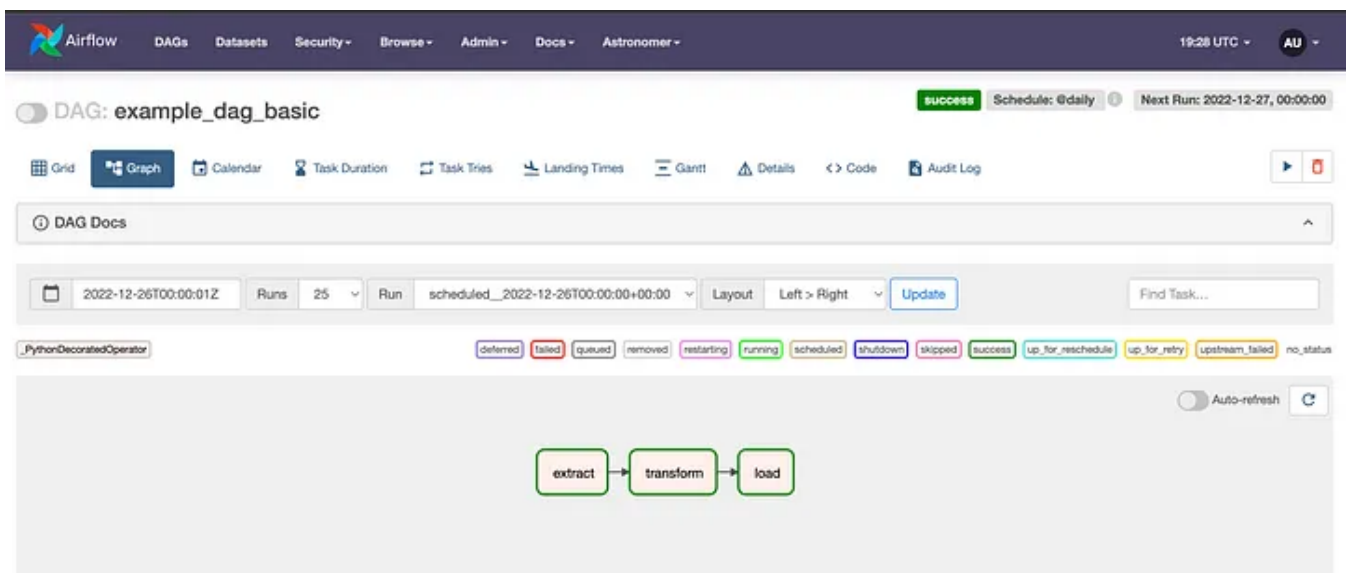
Centraliza as informações principais sobre a DAG, onde você consegue filtrar as execuções por datas, saber o numero de vezes que rodou, se foi com sucesso ou não, bem útil para uma avaliação inicial e menos detalhada das suas DAGs.



Aba DAG — Grid View

Graph View

Essa visualização é boa pra você checar todas as dependências do seu Data Pipeline e ver por etapa como está ou foi o andamento dela, lembra que falamos que a DAG é um conjunto de tasks executadas num sentido, certo?

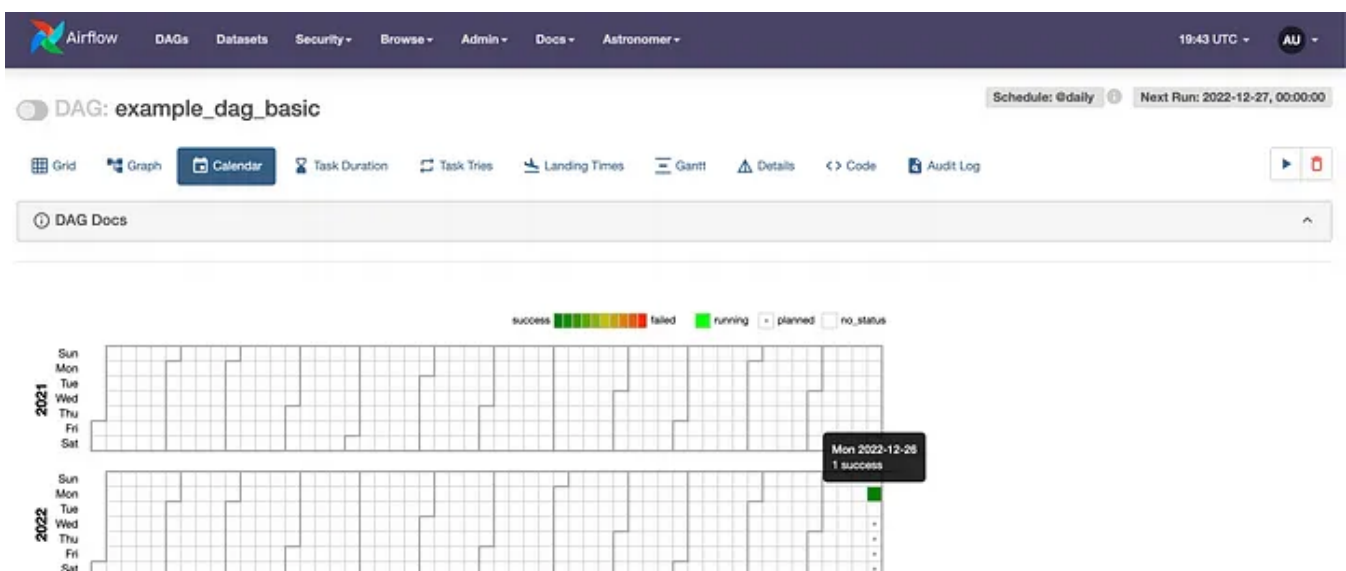


Aba DAG — Graph View

Nesse tipo de visão podemos segmentar por data e os filtros de cores serve para cada um dos status possíveis, como podem ver aí, o verdinho é pro status de "success" que no meu caso só rodei uma vez e com sucesso. Além dos status no lado esquerdo posso filtrar pelo tipo de operator que estou usando (no caso só um, o `_pythonDecoratedOperator`).

Calendar

Nessa aba como o próprio nome já diz podemos ver o status das nossas execuções de acordo com o calendário e junto com ele o status das execuções.



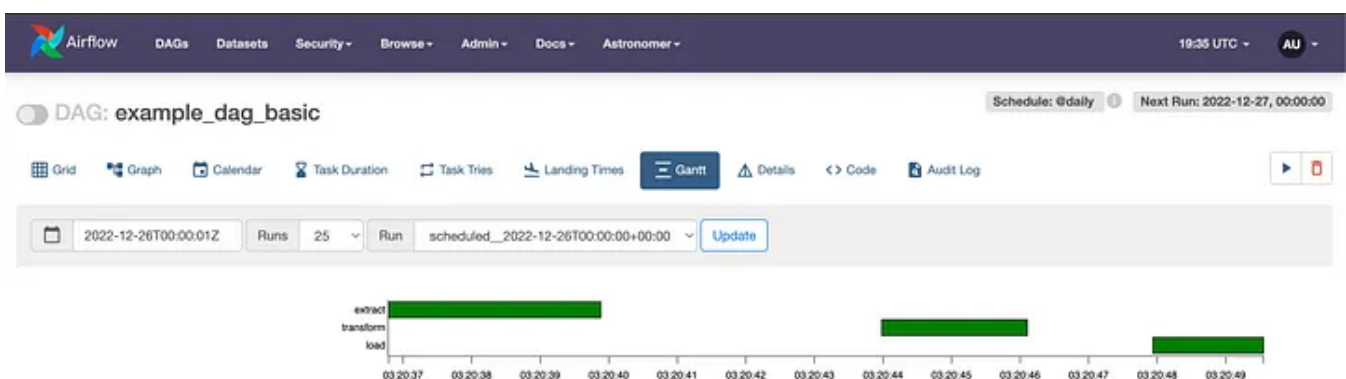
tenho apenas 1 execução e com sucesso em Monday

Essa view pode parecer um pouco estranha mas ela tira a média das execuções, por exemplo, caso você tenha tido problemas nas execuções da terça por exemplo, ele

tira uma média naquela escala de cores e coloca no dia, caso não esteja verde escura, vale a pena dar uma olhada no que rolou. Uma outra vantagem é que as vezes posso ter erros por dias da semana, ou fim de mês, devido a uma concorrência ou algo do tipo, esse é um bom método pra pegar esse tipo de anomalia.

Gantt View

O Gantt View (Baseado no gráfico de Gantt) é bem útil quando quero saber o tempo que cada uma das tasks da DAG rodou, até pra saber qual parte do data pipeline está consumindo mais recursos, ou precisa ser ajustado, ou as vezes até dividido em novas tasks.



Note no exemplo o tempo de execução de cada uma das tasks da minha DAG de exemplo

Details

Precisa de alguma informação mais específica sobre a DAG é aqui que vai achar. Infos como timezone, qual é o arquivo relacionado a DAG, dono e muitas outras informações que você pode não encontrar em outras views.

Airflow	
example_dag_basic	
Schedule: Weekly Next Run: 2022-12-27, 05:00:00	
DAG Details	
Schedule Interval: Daily	
Catchup: False	
Start Date: None	
Next Active Run: 2022-12-27 05:00:00	
Concurrency: 16	
Default Args: {'owner': 'me'}	
Tasks Count: 8	
Task File: [airflow, transformers, test]	
Python File Location: example_dag_basic.py	
Owner: airflow	
Owner Link: None	
Email Run To: None	
Tags: [example]	
DAG Metadata Information	
Attribute Value	
Name: /usr/local/airflow/dags/example_dag_basic.py	
Has Import Errors: False	
Has Task Concurrency Limits: False	
Is Active: True	
Is Paused at Creation: True	
Is Waiting: False	
Last Updated: None	
Last Pending Time: 2022-12-27 14:09:11.402220+00:00	
Last Pending Reason: None	
Metadata: Metadata {}	
Next Dagrun: 2022-12-27 05:00:00+00:00	
Next Dagrun Reason: After	
Next Dagrun Interval: 2022-12-27 05:00:00+00:00	
Next Dagrun Interval End: 2022-12-27 05:00:00+00:00	
Next Dagrun Interval Start: 2022-12-27 05:00:00+00:00	
Parent Dag: None	
Parent ID: None	
Owner Email: airflow@airflow.apache.org	

Aba DAGs — View Details

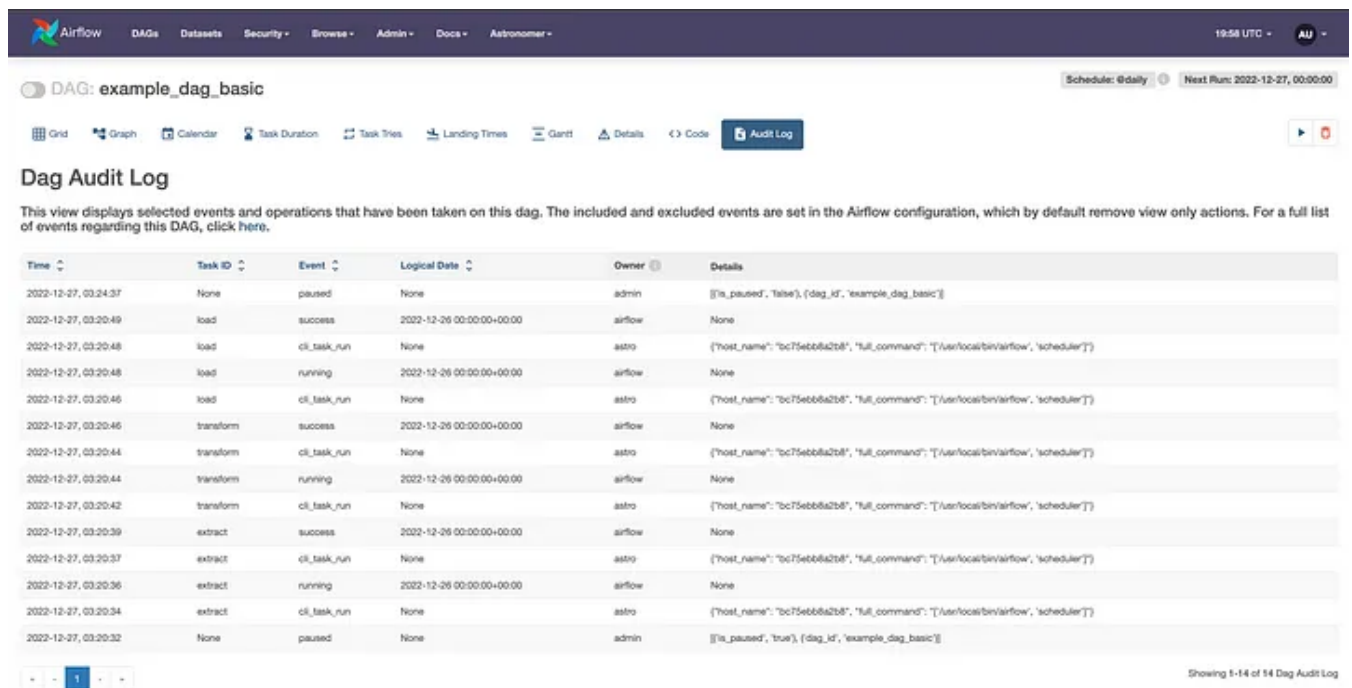
Code

Como o nome já diz, se preciso ver o código da DAG é aqui e consigo visualizar, sem a necessidade de ir no arquivo, ter acesso a pasta e etc, outra vantagem é que você sempre vai ver o mais atualizado caso esteja trabalhando com controle de versão no Git.

```
1 import sys
2 from airflow.decorators import task, task_group
3 from airflow.decorators import dag, task_group
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
2090
2091
2092
2093
2094
2095
2096
2097
2098
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2179
2180
2181
2182
2183
2184
2185
2186
2187
2188
2189
2190
2191
2192
2193
2194
2195
2196
2197
2198
2199
2200
2201
2202
2203
2204
2205
2206
2207
2208
2209
2210
2211
2212
2213
2214
2215
2216
2217
2218
2219
2220
2221
2222
2223
2224
2225
2226
2227
2228
2229
2230
2231
2232
2233
2234
2235
2236
2237
2238
2239
2240
2241
2242
2243
2244
2245
2246
2247
2248
2249
2250
2251
2252
2253
2254
2255
2256
2257
2258
2259
2260
2261
2262
2263
2264
2265
2266
2267
2268
2269
2270
2271
2272
2273
2274
2275
2276
2277
2278
2279
2280
2281
2282
2283
2284
2285
2286
2287
2288
2289
2290
2291
2292
2293
2294
2295
2296
2297
2298
2299
2300
2301
2302
2303
2304
2305
2306
2307
2308
2309
2310
2311
2312
2313
2314
2315
2316
2317
2318
2319
2320
2321
2322
2323
2324
2325
2326
2327
2328
2329
2330
2331
2332
2333
2334
2335
2336
2337
2338
2339
2340
2341
2342
2343
2344
2345
2346
2347
2348
2349
2350
2351
2352
2353
2354
2355
2356
2357
2358
2359
2360
2361
2362
2363
2364
2365
2366
2367
2368
2369
2370
2371
2372
2373
2374
2375

```


Deu erro em alguma das suas tasks? Quer entender o porque? Basta entrar na view AuditLogs e filtrar por qual step quer ver os logs, qual data. Mas nessa primeira etapa mostra logs bem superficiais e sinceramente, achei que a busca poderia ser melhor mas é isso que tem, rs.

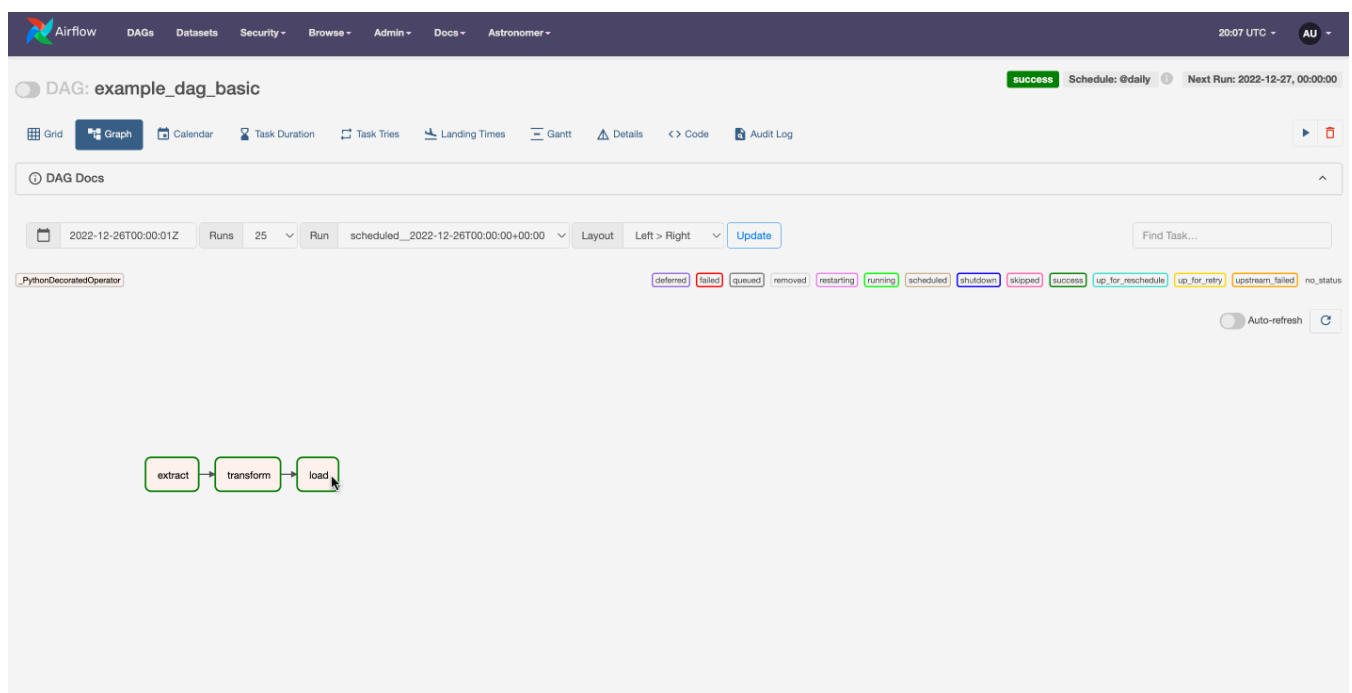


The screenshot shows the 'Dag Audit Log' view in the Airflow web interface. It displays a table of events for the DAG 'example_dag_basic'. The table has columns for Time, Task ID, Event, Logical Date, Owner, and Details. The events include 'paused', 'load', 'cli_task_run', 'transform', and 'extract' tasks, with their respective statuses and owners.

Time	Task ID	Event	Logical Date	Owner	Details
2022-12-27, 03:24:37	None	paused	None	admin	['is_paused', 'false'], {'dag_id', 'example_dag_basic'}
2022-12-27, 03:20:49	load	success	2022-12-26 00:00:00+00:00	airflow	None
2022-12-27, 03:20:48	load	cli_task_run	None	astro	['host_name', 'bc75ebb8a2b8', 'full_command', 'T:/usr/local/bin/airflow', 'scheduler']
2022-12-27, 03:20:48	load	running	2022-12-26 00:00:00+00:00	airflow	None
2022-12-27, 03:20:48	load	cli_task_run	None	astro	['host_name', 'bc75ebb8a2b8', 'full_command', 'T:/usr/local/bin/airflow', 'scheduler']
2022-12-27, 03:20:46	transform	success	2022-12-26 00:00:00+00:00	airflow	None
2022-12-27, 03:20:44	transform	cli_task_run	None	astro	['host_name', 'bc75ebb8a2b8', 'full_command', 'T:/usr/local/bin/airflow', 'scheduler']
2022-12-27, 03:20:44	transform	running	2022-12-26 00:00:00+00:00	airflow	None
2022-12-27, 03:20:42	transform	cli_task_run	None	astro	['host_name', 'bc75ebb8a2b8', 'full_command', 'T:/usr/local/bin/airflow', 'scheduler']
2022-12-27, 03:20:39	extract	success	2022-12-26 00:00:00+00:00	airflow	None
2022-12-27, 03:20:37	extract	cli_task_run	None	astro	['host_name', 'bc75ebb8a2b8', 'full_command', 'T:/usr/local/bin/airflow', 'scheduler']
2022-12-27, 03:20:36	extract	running	2022-12-26 00:00:00+00:00	airflow	None
2022-12-27, 03:20:34	extract	cli_task_run	None	astro	['host_name', 'bc75ebb8a2b8', 'full_command', 'T:/usr/local/bin/airflow', 'scheduler']
2022-12-27, 03:20:32	None	paused	None	admin	['is_paused', 'true'], {'dag_id', 'example_dag_basic'}

Aba DAGs — AuditLogs

Existe uma outra forma de acessar seus logs que por sua vez te dão bem mais informações sobre sua task, por exemplo, usando o **Graph View**, clicando em uma das tasks da DAG. Se liga.



nesse exemplo mostro detalhes da task e os logs

Não foram abordados todos os temas mesmo porque não tenho informações de logs ou execuções o suficiente pra isso mas já dá pra viver. Se ficou duvidas segue um videozinho abaixo recapitulando todos os pontos discutidos na aba DAGs.

Resumo sobre as visualizações de DAGs na UI do Airflow

Como disse pra não ficar muito pesado, optei por quebrar esse material em 2, no próximo post iremos abordar de temas como:

- Comando uteis no Airflow CLI
- Agendando uma DAG
- Mais informações sobre os operators
- Definir o Path de suas DAGs
- Fazermos alguns testes com dados
- Como fazer a prova de certificação.

Se curtiu esse material, deixa a palminha ae, deixa um like nos vídeos, se inscreva no canal, essa métrica ajuda a saber se esse material está sendo relevante e se devo continuar postando coisas desse tipo.

Te aguardo no próximo!

Anselmo Borges

Apache Airflow

Astronomer

Data Engineering

Certification

Rescue Point



Follow

Written by Anselmo Borges

491 Followers · Editor for Rescue Point

Bigdata Engineer, Cloud Architect, Nerd, Alcoholic, Brazilian JiuJitsu Black belt and hide and seek World champion.

More from Anselmo Borges and Rescue Point



 Anselmo Borges in Rescue Point

Certificação Airflow Fundamentals (Parte 2/3)

Continuando nosso material sobre a certificação básica em Apache Airflow da Astronomer, com um material bem rico com fotos, videos e...

9 min read · Dec 30, 2022

 66 

Open in app 

Sign up

Sign In

  Search Medium



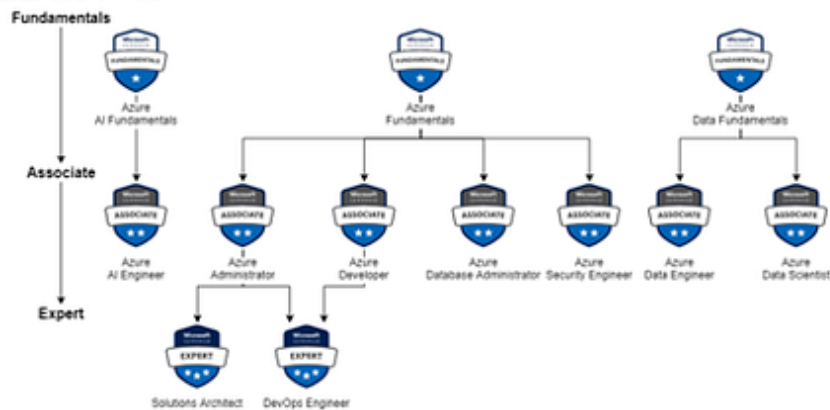
 Anselmo Borges in Rescue Point

Certificação Airflow Fundamentals (Parte 3/3)

Ultimo post da certificação de Apache Airflow básica pra você se certificar ainda esse mês e de GRAÇA!

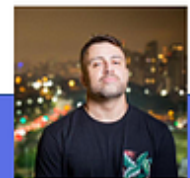
6 min read · Jan 4

👏 74 💬



Certificações Microsoft Azure na faixa

Um overview sobre as certificações e como obter vouchers para as provas, que vão de 50 a 100% do valor da prova.



Anselmo Borges
DataOps Engineer
Microsoft MCT

Anselmo Borges in Rescue Point

Como se certificar em Microsoft Azure Gratuitamente?

5 formas de arrumar vouchers de certificações em Microsoft Azure que podem variar de 50% a 100% dependendo da prova.

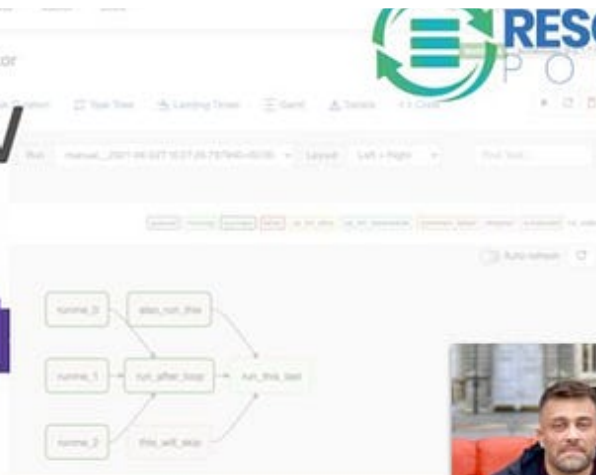
6 min read · May 31, 2022

👏 75 💬 2





Apache
Airflow



Anselmo Borges
Data Engineer Specialist
Microsoft MCT

Compilado certificação Airflow Fundamentals

Um compilado de 3 posts feitos com base no material da Astronomer que vão te ajudar a passar nessa prova que está de graça!



Anselmo Borges in Rescue Point

Compilado Certificação Airflow

Compiladão dos 3 posts que te ajudam a tirar a certificação de Airflow Fundamentals da Astronomer, aproveite, a prova está de graça denovo!

2 min read · Aug 25



49




See all from Anselmo Borges

See all from Rescue Point

Recommended from Medium



 Khoulood El Alami in Towards Data Science

Don't Start Your Data Science Journey Without These 5 Must-Do Steps From a Spotify Data Scientist

A complete guide to everything I wish I'd done before starting my Data Science journey, here's to acing your first year with data

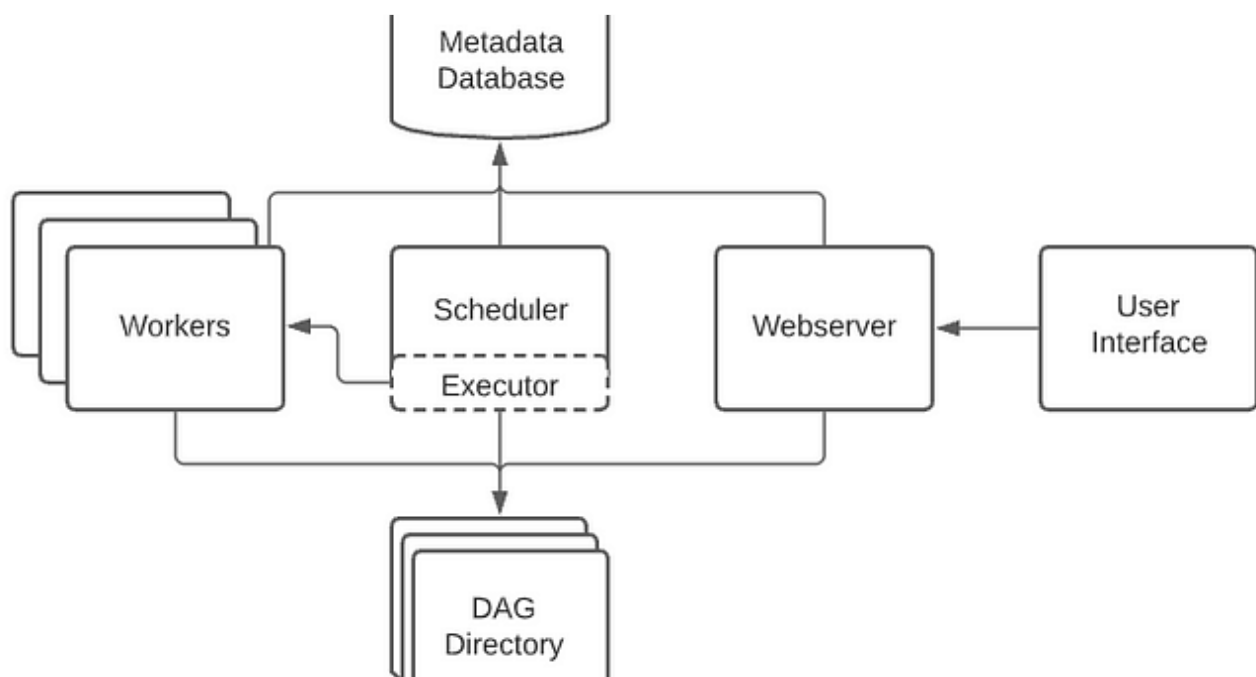
18 min read · Sep 24




1.3K



10



 Himanshu Kumar in Naukri Engineering

Scaling and Achieving High Availability (HA) in Airflow using Local Executor

Airflow is a workflow management system that lets the user define workflows programmatically and monitor them using a web interface...

4 min read · Jun 14



Lists



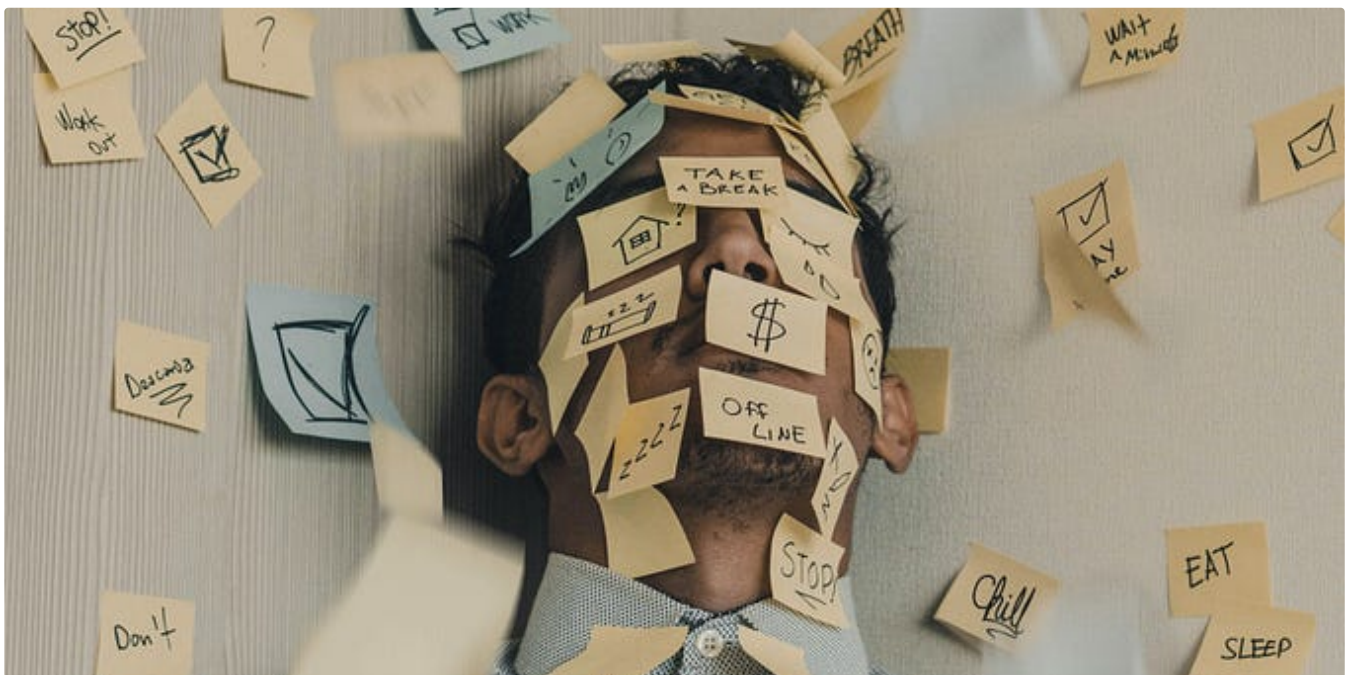
New_Reading_List

174 stories · 128 saves



Natural Language Processing

666 stories · 269 saves



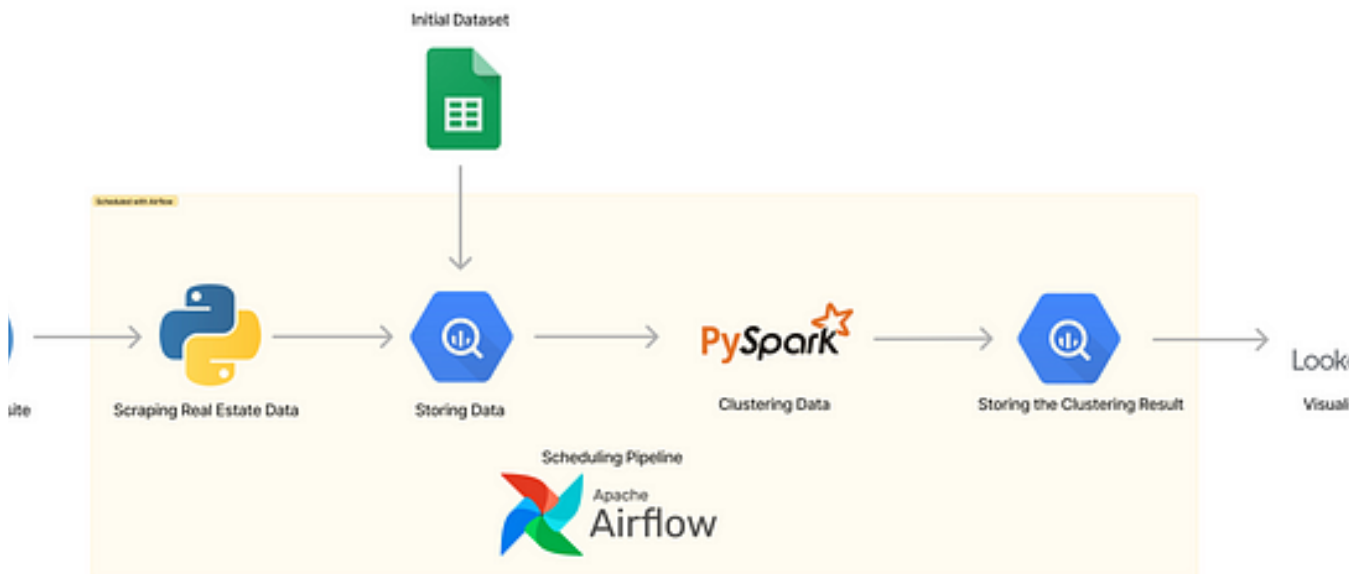
Orchestra in Orchestra's Data Release Pipeline Blog

Why organisations fail data teams

Gartner reported in 2017 that upwards of 85% of data projects fail. Today, it seems the figure is similar for AI projects. These statistics...

6 min read · 5 days ago





Dana Fatadilla Rabba in Towards Dev

Building End-to-end Data Pipeline with Airflow, PySpark, and BigQuery

In today's data-driven world, extracting, transforming, and loading (ETL) processes play a vital role in handling and analyzing large...

4 min read · May 12



61



Apache Airflow

How to Install Apache Airflow In Docker Container on EC2 Machine.

In this article, we'll cover how to install Airflow Docker container on ec2 machine in simple 3 steps.

4 min read · May 30



Understanding dbt Python Models on Snowflake

Traditionally, dbt models are defined using SQL. But in early 2022, dbt Labs made an internal push to extend the dbt framework to support...

2 min read · 3 days ago



See more recommendations