



Dagster Deep Dives: Configurations & Resources

Colton Padden

Developer Advocate - Dagster Labs

Overview



Dagster provides ***Configurations*** and ***Resources*** as fundamental building blocks to build data pipelines that are reusable, flexible, and extensible.



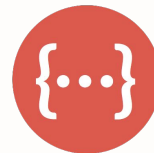
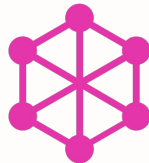
What we'll demonstrate...

1. Develop and test locally, and deploy to production with confidence (*emulate prod locally*)
2. Standardize your codebase in a reusable way; using configurations to modify how pipelines run
3. Leverage the power of the Launchpad for ad-hoc job execution



What's a *Resource*?

A standard way of defining external services, tools, and storage locations. These resources can be configured!





Integrate with the Modern Data Stack

Explore all Dagster integrations at dagster.io/integrations



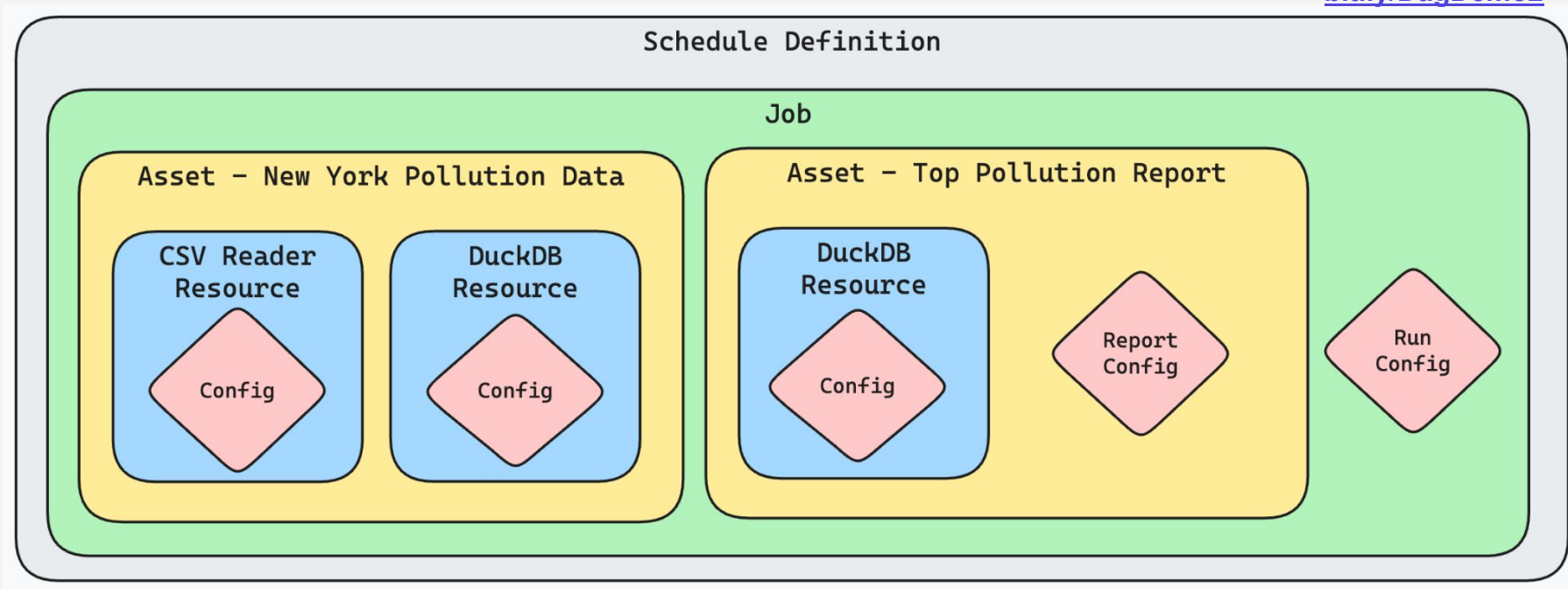


So, what's a **Configuration**?

A way to specify certain parameters to Assets, Ops, Runs, and more; they can be modified via the *Launchpad*.

What we'll build.

bit.ly/DagDemo2



Using Built-in and Custom Resources

```
import pandas as pd

from dagster import ConfigurableResource
from dagster_duckdb import DuckDBResource
from pydantic import Field
```

```
duckdb_resource = DuckDBResource(
    database=EnvVar("DUCKDB_DATABASE")
)
```


```
class CSVResource(ConfigurableResource):
    location: str = Field(description="Path to CSV (file:// or https://)")

    def load_dataset(self) -> pd.DataFrame:
        return pd.read_csv(self.location)
```

```
# .env.local
DUCKDB_DATABASE=local.duckdb
NY_AIR_QUALITY_CSV="data/ny-air-quality.csv"

# .env.staging
DUCKDB_DATABASE=local.duckdb
NY_AIR_QUALITY_CSV="https://url.com/data.csv"
```

Leverage environment variables w/ EnvVar



Check Out the DuckDB Integration

```
class DuckDBResource(ConfigurableResource):
    database: str = Field(
        description="Path to the DuckDB database."
    )
    connection_config: Dict[str, Any] = Field(
        description="DuckDB connection configuration options.",
        default={},
    )

    @contextmanager
    def get_connection(self):
        ...
        yield conn
        conn.close()
```

Using Resources

```
@asset
def ny_air_quality(
    database: DuckDBResource,
    air_quality_csv: CSVResource,
    config: NYAirQualityConfig
):

    df = air_quality_csv.load_dataset()

    df.columns = [c.lower().replace(" ", "_") for c in df.columns]

    with database.get_connection() as conn:
        conn.cursor().execute(
            """
            CREATE OR REPLACE TABLE ny_air_quality
            AS
            SELECT * FROM df
            """
        )
```

```
defs = Definitions(
    assets=[ny_air_quality, ny_air_quality_report],
    jobs=[air_quality_report_job],
    schedules=[report_hourly, report_daily],
    resources={
        "database": duckdb_resource,
        "air_quality_csv": air_quality_resource,
    },
)
```

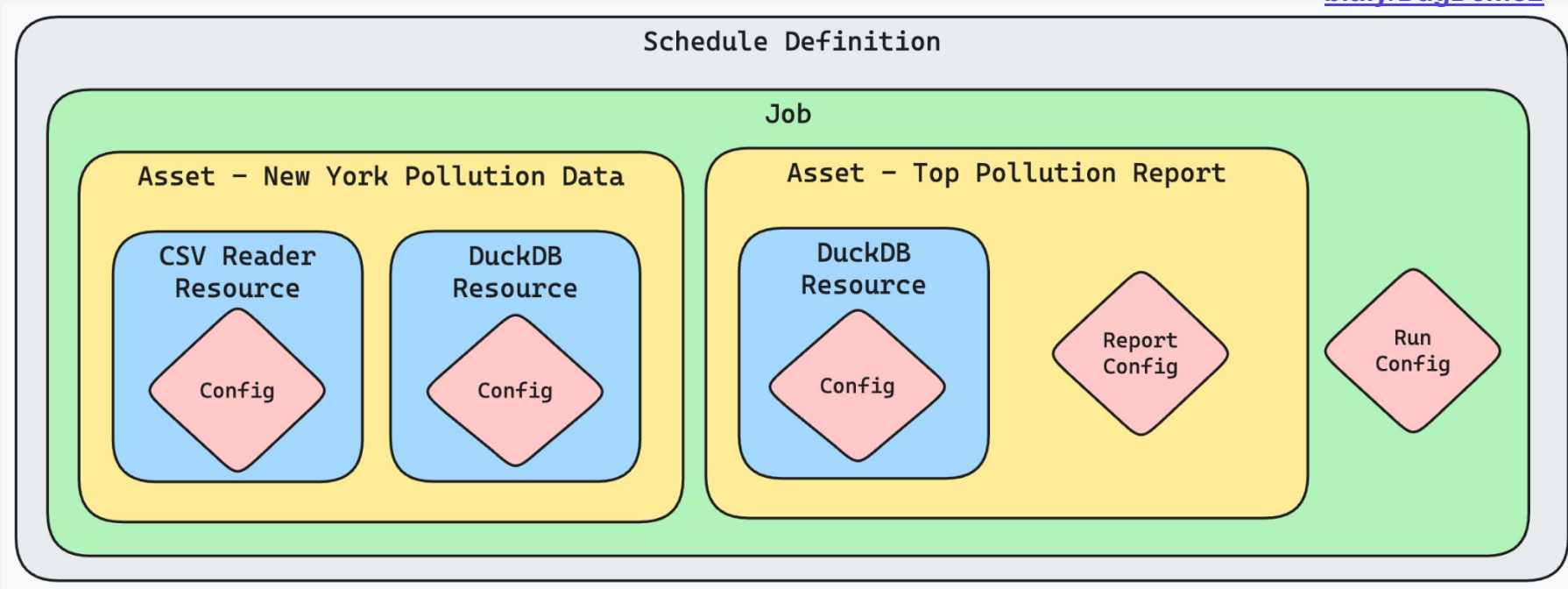
Configuring Scheduled Jobs

```
my_job_hourly = ScheduleDefinition(  
    name="custom_hourly_job",  
    job=air_quality_report_job,  
    cron_schedule="0 * * * *", # Every hour  
    run_config=RunConfig(ops={"ny_air_quality_report": ReportConfig()}),  
)
```

```
my_job_daily = ScheduleDefinition(  
    name="custom_daily_job",  
    job=air_quality_report_job,  
    cron_schedule="0 0 * * *", # At 12:00 AM UTC  
    run_config=RunConfig(  
        ops={  
            "ny_air_quality_report": ReportConfig(  
                limit=1000, destination_table="ny_annual_average_report_1000"  
            )  
        }  
    ),  
)
```

Review

bit.ly/DagDemo2



Let's Go!

github.com/dagster-io/devrel-project-demos

Next steps & resources



Join us in Slack

Connect with other data practitioners. Share knowledge or find help

bit.ly/DagSlack



Sign up for Dagster Cloud

Sign up for Dagster Cloud and get started with a free 30 day trial

bit.ly/DagCloud



Get the Newsletter

Stay up-to-date on the latest events and news

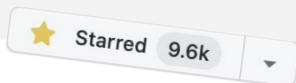
bit.ly/DagNews



Star the GitHub Repo

Let us know you enjoyed this presentation!

bit.ly/DagDemo



More

- Running unit tests for assets and resources
[test resources and configurations.py](#)
- Use DuckDB locally, and Snowflake when deployed
[conditional database resources.py](#)

Q / A

name	geo_place_name	measure_info	mean_value
Nitrogen dioxide (NO2)	Midtown (CD5)	ppb	34.93
Nitrogen dioxide (NO2)	Gramercy Park - Murray Hill	ppb	32.63
Nitrogen dioxide (NO2)	Chelsea - Clinton	ppb	30.66
Nitrogen dioxide (NO2)	Stuyvesant Town and Turtle Bay (CD6)	ppb	30.36
Nitrogen dioxide (NO2)	Chelsea-Village	ppb	29.53

NO2 standards set by the World Health Organization (WHO) and US Environmental Protection Agency (EPA):

Good (0-100 ppb)	Generally, no symptoms will be found in people when exposed to these levels.
Moderate (100-250 ppb)	Nitrogen Dioxide levels in this range have been shown to cause respiratory discomfort in people who have issues, such as those with asthma
Unhealthy (250-500 ppb)	This level of NO2 concentration will cause discomfort and potential health problems for any person regardless of health and age.

