LUCY LINDETZ

Software Engineer

Head of Infrastructure at **IVI**



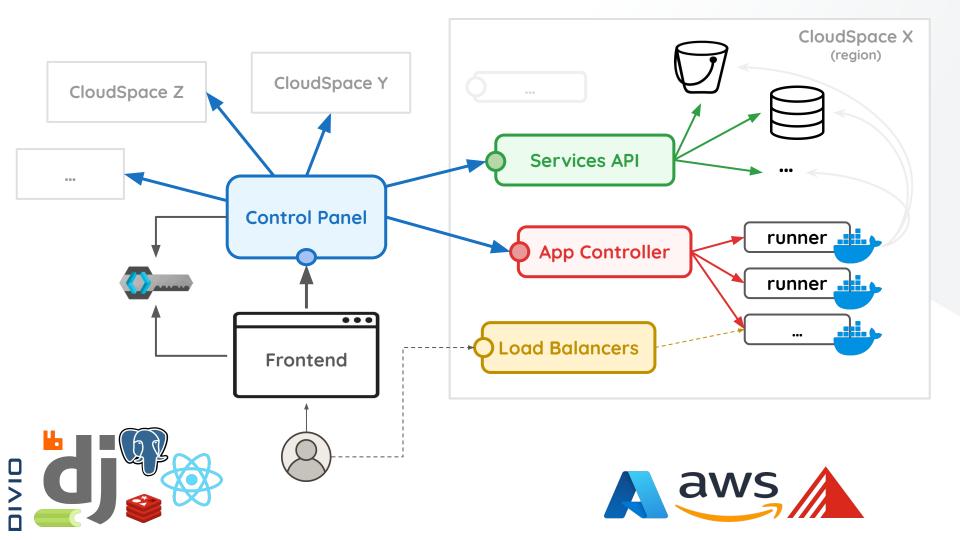
@ DETZLIN

https://derlin.ch

DIVIONAL PAAS



"PEACE OF MIND"





7'000+ applications

8 engineers

10'000+ environments

7'326
postgres 13

25'000+
services

Back to Postgres

PostgreSQL 13

24.09.2020

PostgreSQL 15

12.10.2022

FROM: Amazon Web Services, Inc.

ON: June 11, 2024

[Action required] Update Your Amazon RDS and Amazon Aurora SSL/TLS Certificates by August 22, 2024

. . .













III. config



IV. backing services



V. build|release|run



The Twelve-Factor App

https://12factor.net















Store config in the environment

A Divio App is ...

- . Dockerized code
- Running in stateless containers
- 3. Connected to backing services
- 4. Configured via Environment variables
- 5. And disposable (scalable!)



Execute the app as one or more stateless processes



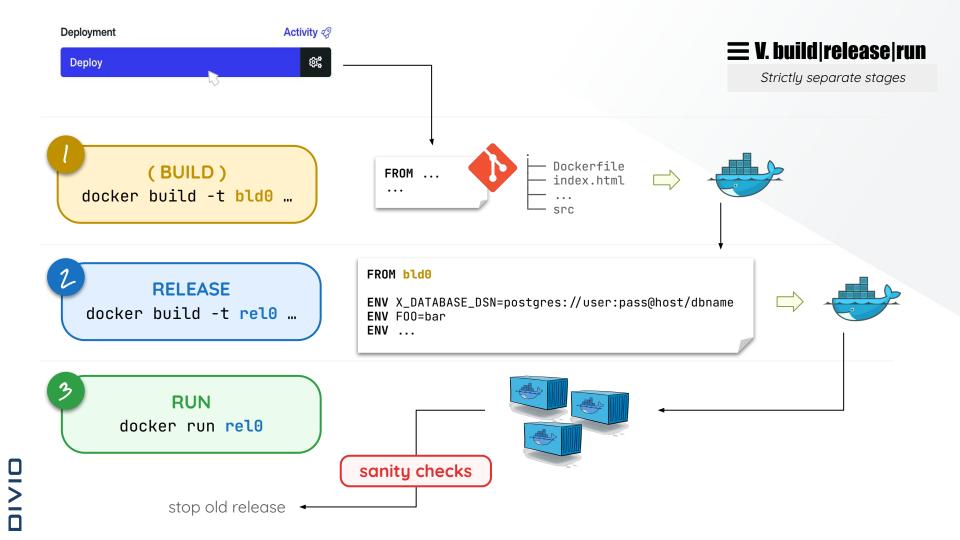
Treat backing services as attached resources

✓ VIII. Concurrency

Scale out via the process model



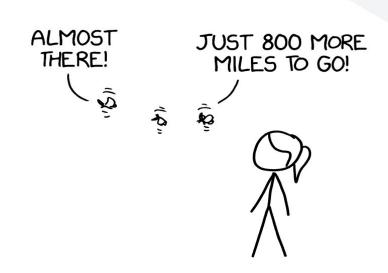
Maximize robustness with fast startup and graceful shutdown



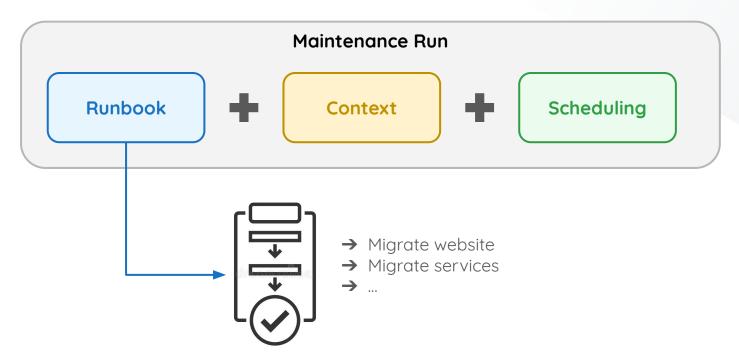
Basic Migration Process

- Stop writes
- ⊗ . Backup OLD (pg_dump)

- Resume writes
- ⊗ . Delete OLD



What's a maintenance?





Run admin/management tasks as one-off processes

base class

```
class Runbook:
   preconditions = None
   operations = None

def estimated_execution_time(self, run):
    # return a maintenance duration

def apply(self, run):
    # apply maintenance on each context
```

limit to pg13 apps

```
class PostgresRunbook(runbook.Runbook):
   preconditions = [
        conditions.HasServiceInstance(
            service_identifier="postgresql-database-13",
            provisioned=True,
        )
   ]
   operations = [
        ...
]
```

provision

```
operations.CollectServiceInstances(
  service_identifier="postgresql-database-13",
  provisioned=True,
                                                                               pause the app
),
operations.ReprovisionCollectedServiceInstances(
                                                                operations.WithMaintenancePage(
  service_identifier_overrides={
                                                                  pause=True,
      "postgresql-database-13": "postgresql-database-15"
                                                                  maintenance_app_pk=settings.MAINTENANCE_APP_ID,
  },
                                                                  operations=[
  copy_runtime_options=True,
),
                                                                ), # will unpause after the operations
                  Temporarily undergoing maintenance
                          We should be back online shortly
```

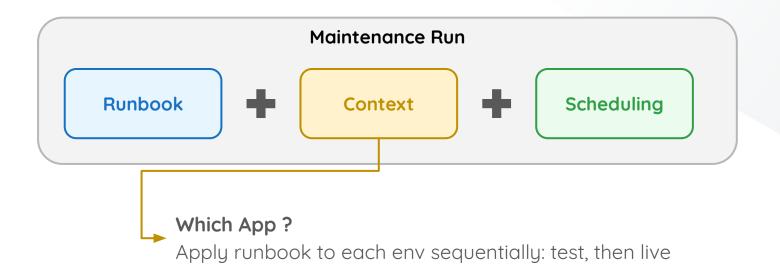
switch database

```
operations.BackupCollectedServices(
  notes=(
    "Backup taken prior to the migration of the database"
    " from Postgres 13 to Postgres 15."
  ),
),
operations.RestoreBackupToReprovisionedServices(),
operations.EnableReprovisionedServices(),
Available in the backups tab:)
```

redeploy

cleanup

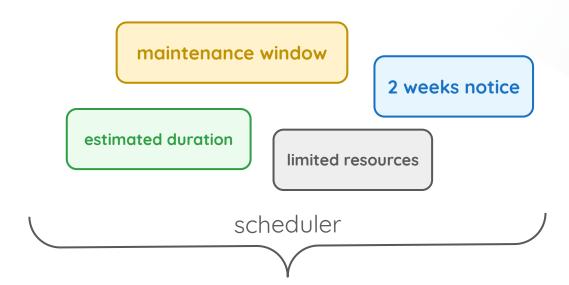
operations.DeprovisionCollectedServiceInstances(),

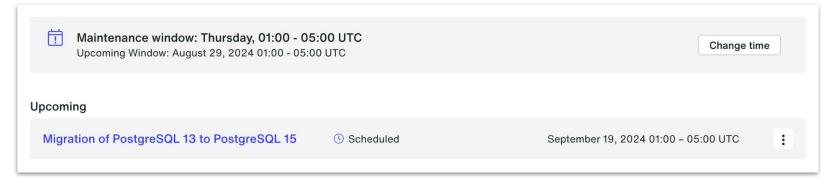




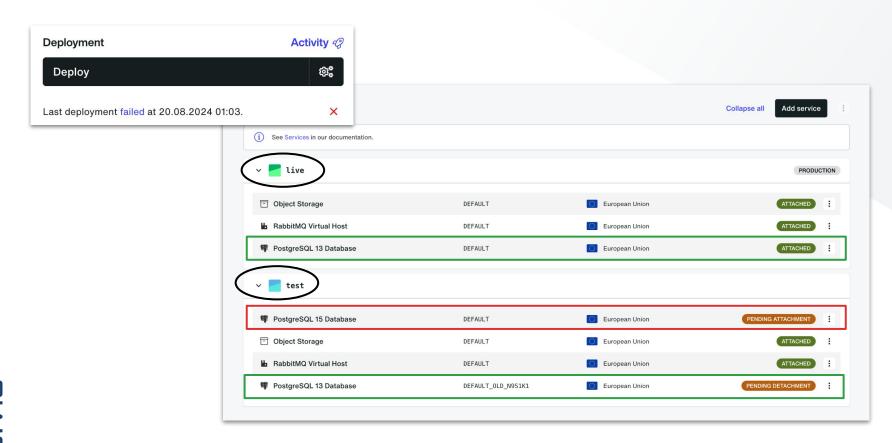








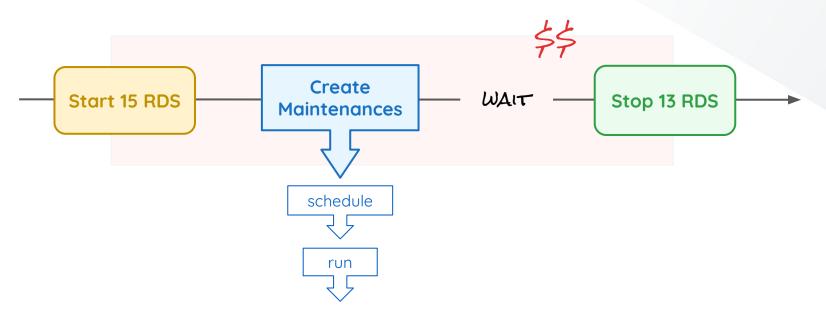
In case of failure...





What about the cost?

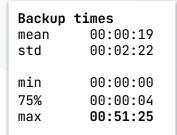
On each region (in parallel)



Now, the real world

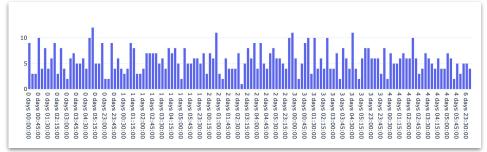








shuffle (almost randomly)



Let's do a first test ...

```
01:57:18.37
                          File "/usr/local/lib/python3.6/site-packages/django/db/backends/postgresql/base.py", line 178, in get_new_connection
              WEB
01:57:18.37
                            connection = Database.connect(**conn_params)
01:57:18.37
                          File "/usr/local/lib/python3.6/site-packages/psycopg2/__init__.py", line 122, in connect
01:57:18.37
                            conn = _connect(dsn, connection_factory=connection_factory, **kwasync)
01:57:18.37
                        django.db.utils.OperationalError: SCRAM authentication requires libpq version 10 or above
01:57:18.37
01:57:18.37
                        unable to load app 0 (mountpoint='') (callable not found or import error)
01:57:18.37
             WEB
                        *** no app loaded. GAME OVER ***
01:57:18.37
                        SIGINT/SIGQUIT received...killing workers...
01:57:18.38
                        gateway "uWSGI http 1" has been buried (pid: 17)
01:57:18.38
             DEPLOYMENT connection error.
```



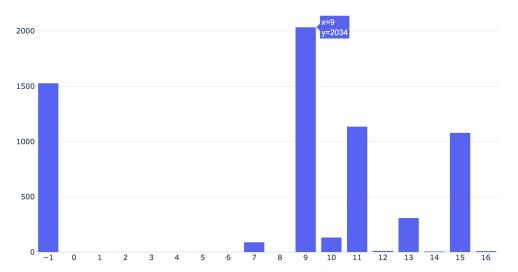
libpq

libpq is the C application programmer's interface to PostgreSQL.

PostgreSQL 14 Release Notes

- Change the default of the password_encryption server parameter to **scram-sha-256** (Peter Eisentraut)

libpq versions (6382)

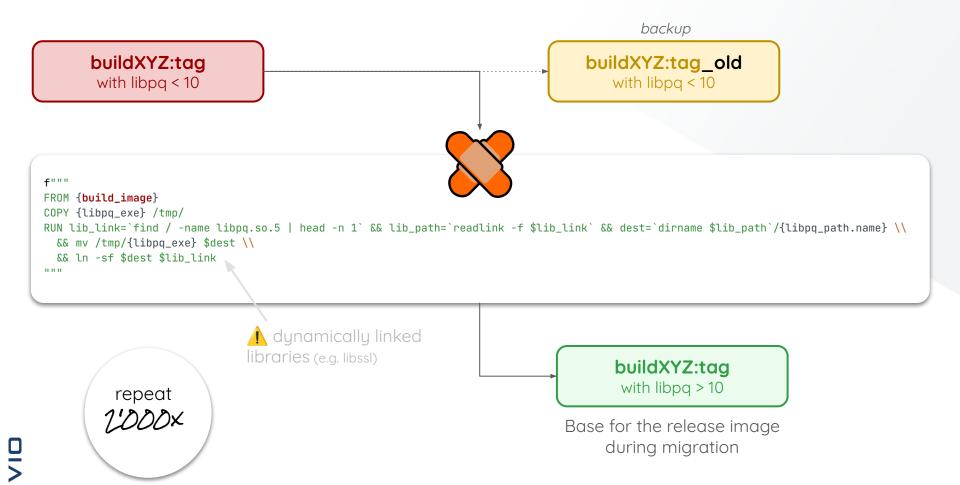




All build images are:

- x86_64 linux
- 1. stored in our registry
- 3. only rebuilt on code change





In 2 cases:

```
pg_restore [...]
```

ERROR: index row size 3048 exceeds btree version 4 maximum 2704 for index i

DETAIL: Index row references tuple (841,9) in relation t.

HINT: Values larger than 1/3 of a buffer page cannot be indexed. Consider a function index of an MD5 hash of the value, or use full text indexing.

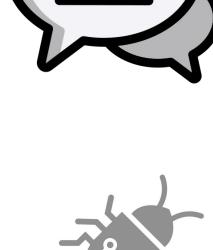
Command was: CREATE INDEX i ON public.t USING btree (bar);



And more ...













To sum up



5′000+

migrations

7'000+ databases

102

failures

99% libpq related 2 index related

99.98%

success rate















