



Testgres

мечтанием postgres как умеем

1st edition

```
SELECT pathman.create_hash_partitions('test.hash_rel', 'Value', 3);
create_hash_partitions
```

```
-----
                3
```

(1 row)

```
SELECT COUNT(*) FROM test.hash_rel;
count
```

```
-----
                3
```

(1 row)

```
SELECT COUNT(*) FROM ONLY test.hash_rel;
count
```

```
-----
                0
```

(1 row)

Как убедиться, что:

- воскресший узел может вернуться в мультимастер?
- pg_shardman умеет распределенный COPY?
- pg_probackup не перемалывает данные?
- pg_pathman работает на реплике?

*И таких вопросов **очень** много!*

Python

- hitchpostgres
- postgresfixture
- pytest-postgresql
- db_integration_test
- ...

Perl

- **PostgresNode.pm**
- pgTAP
- ...

PostgresNode.pm

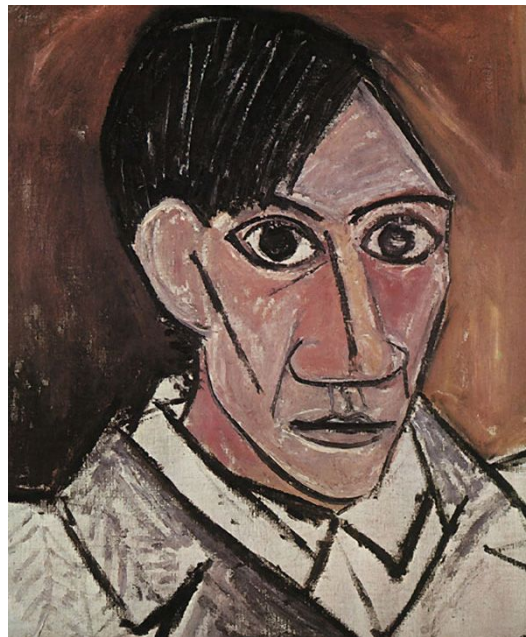
- Поддерживает только актуальный релиз PostgreSQL
- Результаты запросов в виде текста
- Сложная установка (достаем из “исходников”)
- Любые улучшения и исправления идут через сообщество
- И все-таки, это Perl :(

Но в остальном весьма хорош!

Сделаем свой, только лучше!*

- ~~Скопируем~~ украдем оригинальный API
- Перепишем на Python 2 / 3
- Добавим то, чего не хватает
- ...
- PROFIT!

**терминальная стадия NIH syndrome*



Un artista copia, un gran artista roba



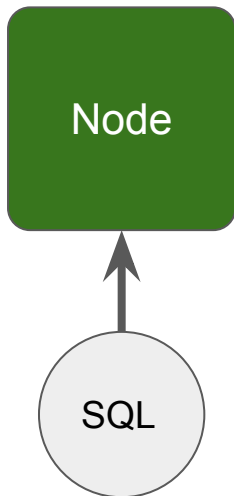
Node

```
node = testgres.get_new_node().init().start()
```

```
node.name  
    'testgres-<UUID>'
```

```
node.port  
    16271
```

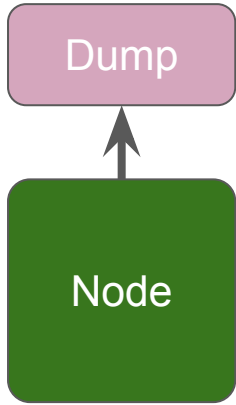
```
node.base_dir  
    '/tmp/tmp4z8zfepec'
```



```
node.safe_psql('SELECT 1')  
              b'1\n'
```

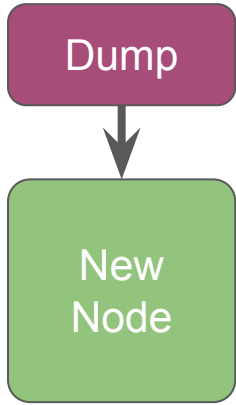
```
node.execute('SELECT 1')  
            [(1,)]
```

```
with node.connect('<DBNAME>') as con:  
    con.begin()  
    con.execute('CREATE TABLE t')  
    con.commit()
```

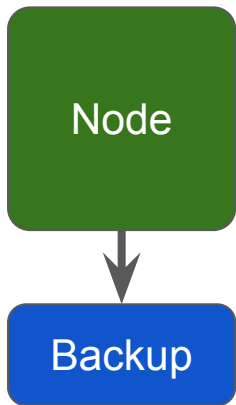



```
dump = node.dump()
```

```
node.dump(dbname='<DBNAME>',  
          username='<USER>',  
          filename='/path/to/file')
```



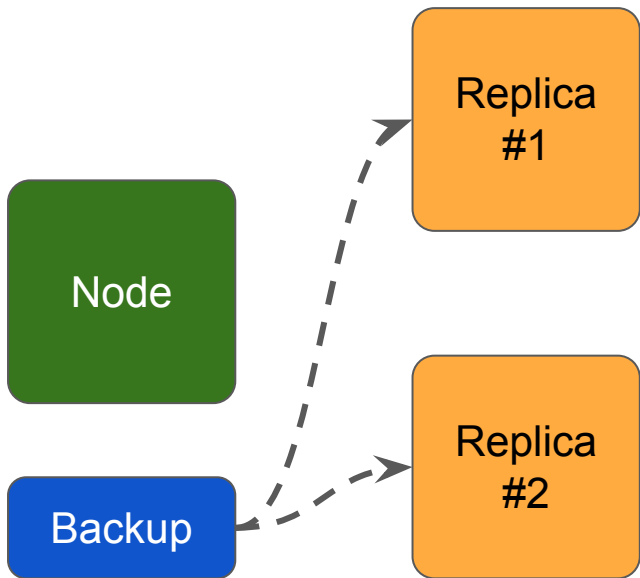
```
new_node = get_new_node().init().start()  
new_node.restore(filename='/path/to/file')
```



```
node = get_new_node() \  
        .init(allow_streaming=true) \  
        .start() as node:
```

```
backup = node.backup()
```

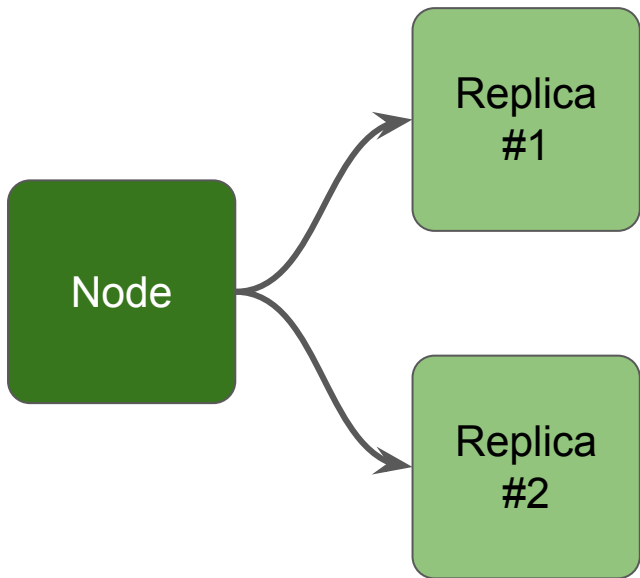
```
backup.spawn_primary()  
backup.spawn_replica()
```



```
replica_1 = backup.spawn_replica()  
replica_2 = backup.spawn_replica()
```

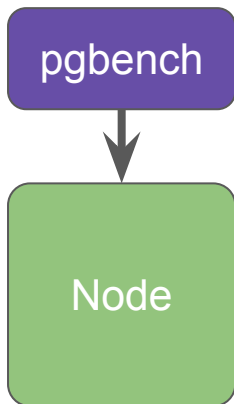
```
replica_1.status()  
NodeStatus.Stopped
```

```
replica_2.status()  
NodeStatus.Stopped
```



```
replica_1 = node.replicate().start()  
replica_2 = node.replicate().start()
```

```
replica_1.execute('select 1')  
[(1,)]
```



```
node = get_new_node().init().start()
```

```
node.pgbench_init(  
    scale=2,  
    foreign_keys=True,  
    options=[ '-q' ]).pgbench_run(time=2)
```

Что в планах?

- Добавить полезные обвязки (replication, generate data, etc)
- Параллельный запуск тестов
- Поддержка Windows
- Асинхронность (?)
- Запуск узлов в Docker (новый модуль?)



<https://github.com/postgrespro/testgres>