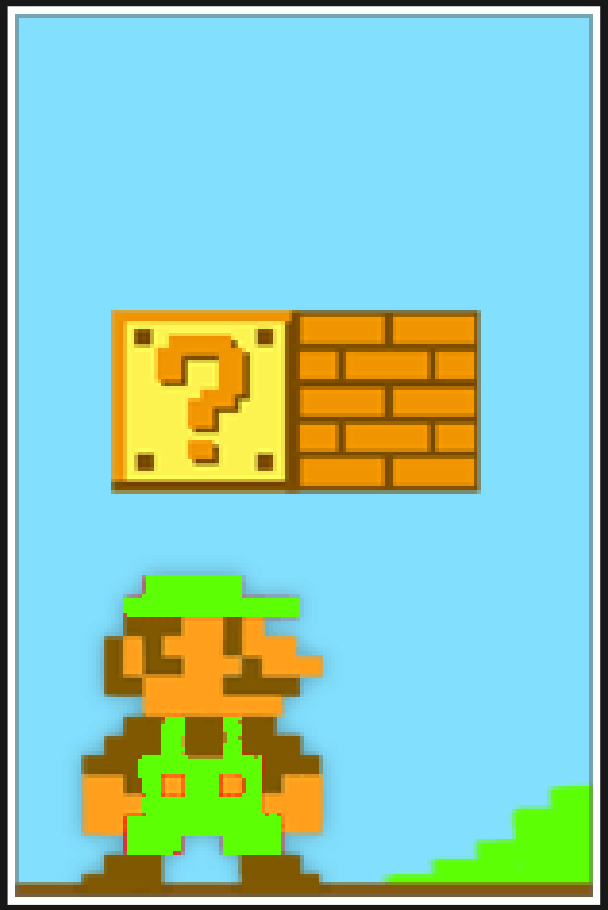
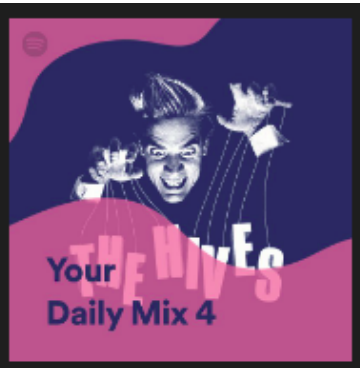
Luigi



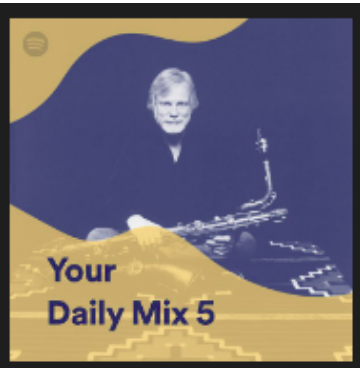
Daily Mix 3

The Fall, IDLES, Gang Of Four and more



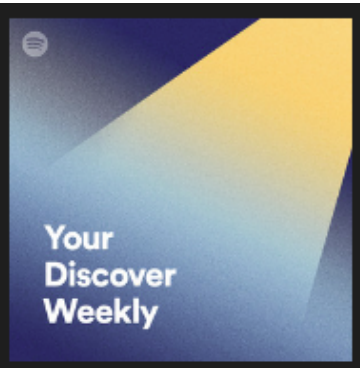
Daily Mix 4

The Hives, Joy Division, The Kinks and more



Daily Mix 5

Gerry Mulligan, Paul Desmond, Dave Brubeck and more



Discover Weekly

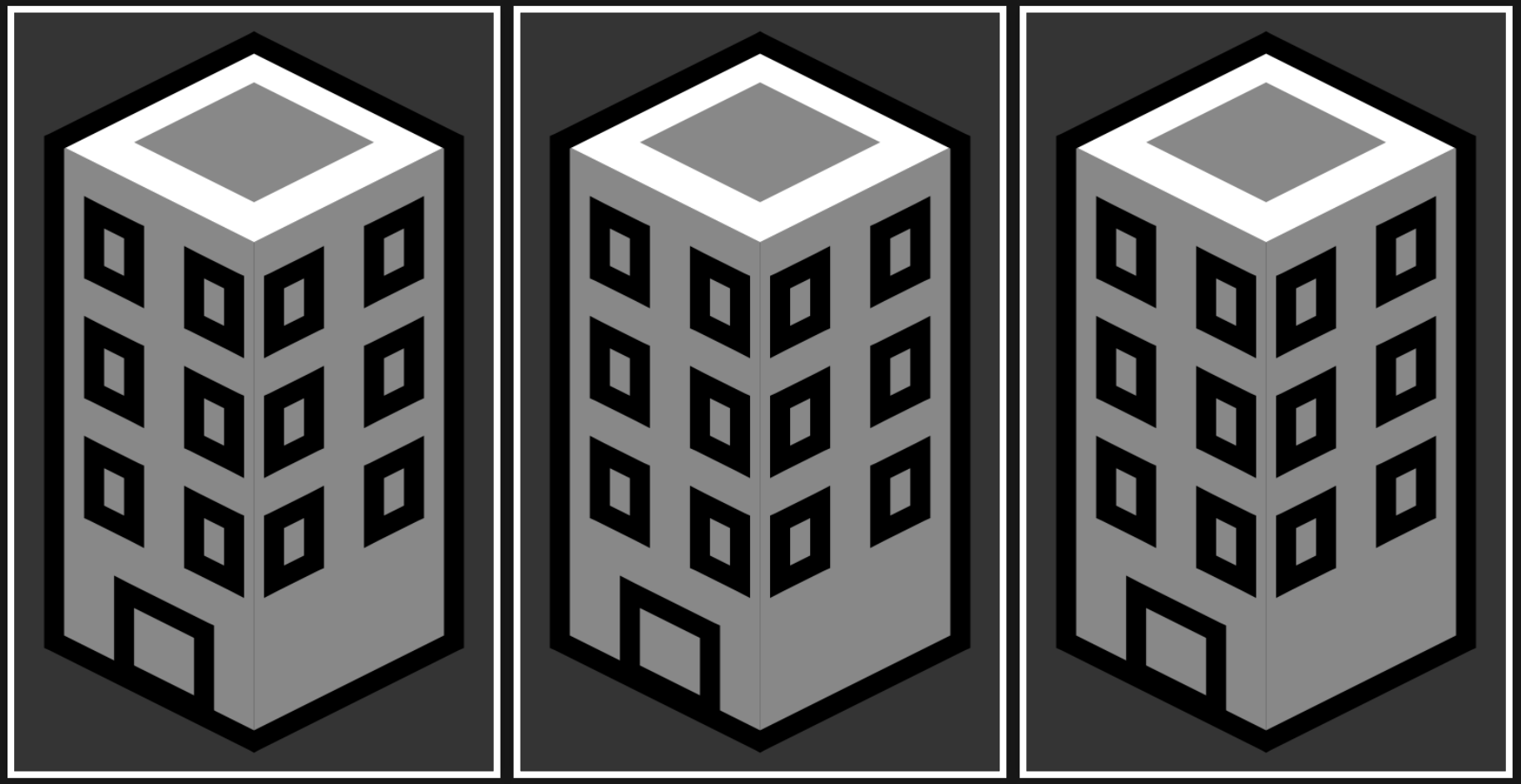
Your weekly mixtape of fresh music. Enjoy new discoveries and deep cut...

PLAYLIST • BY SPOTIFY

Release Radar

Never miss a new release! Catch all the latest music from artists you follow,...

PLAYLIST • BY SPOTIFY



Wigit

```
pip install luigi
```

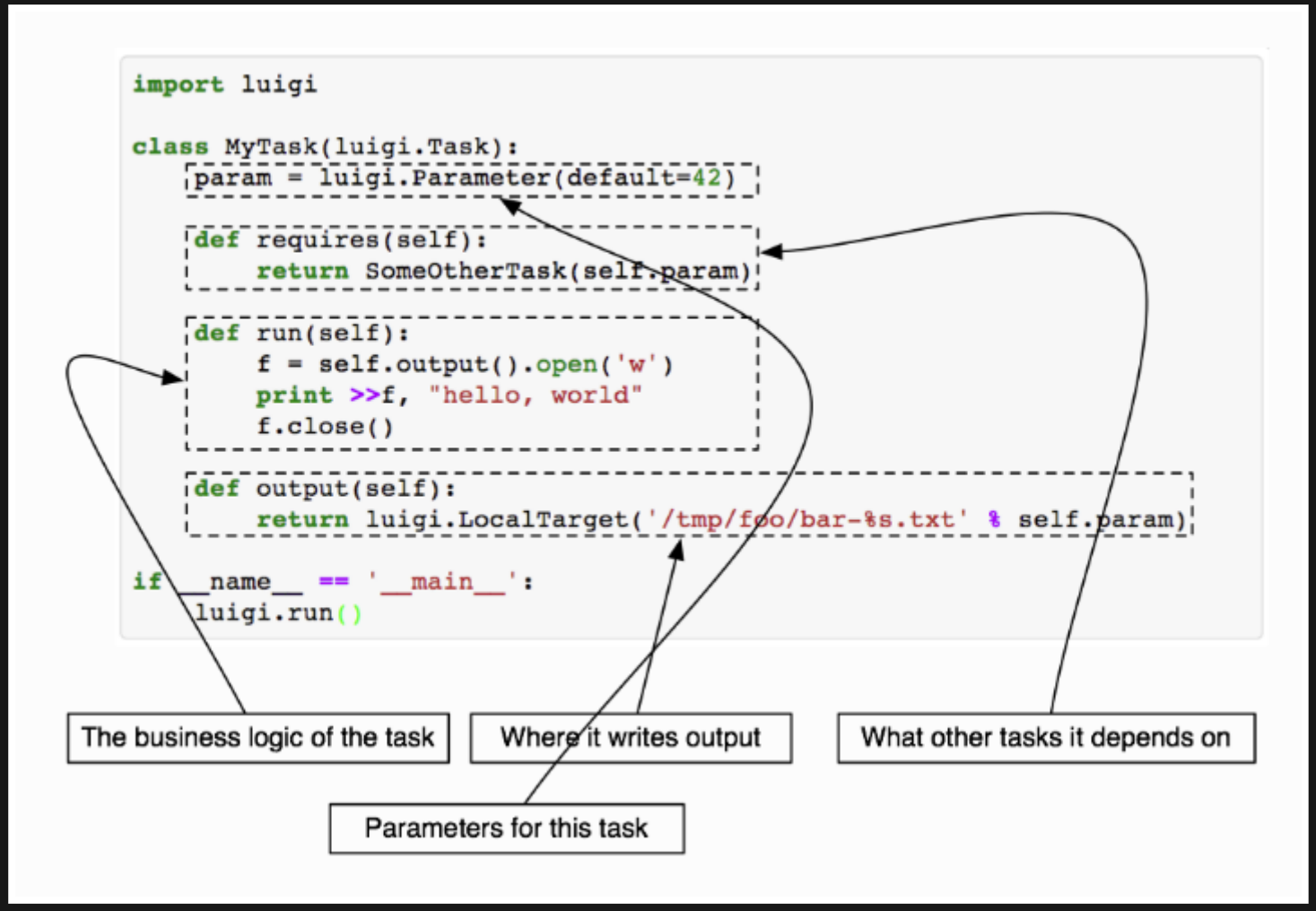
```
luigid --background --port=8082 --logdir=logs
```

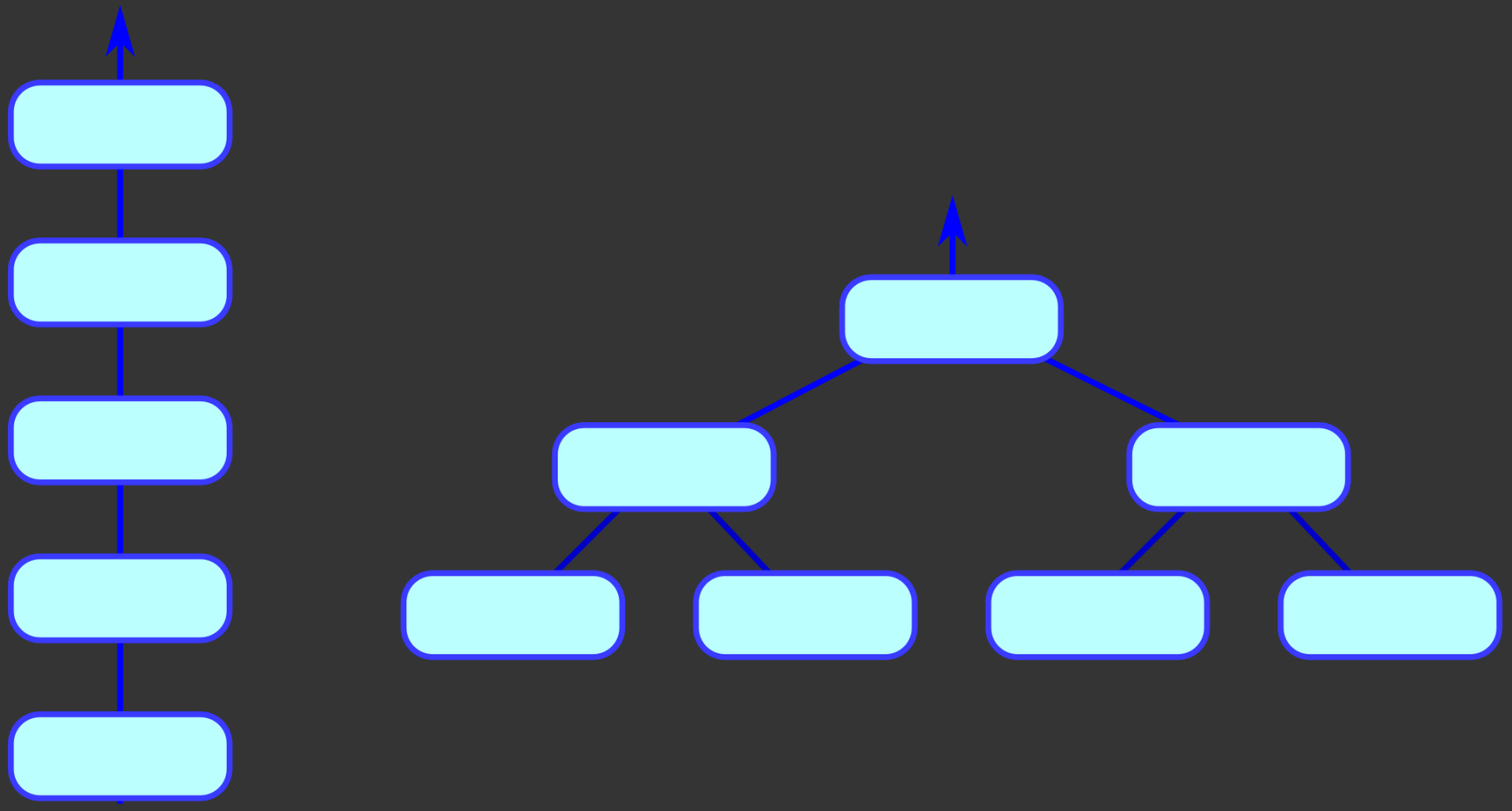
```
class MyTask(luigi.Task):
        myvar1 = luigi.Parameter()
        myvar2 = luigi.Parameter()
        ...
```

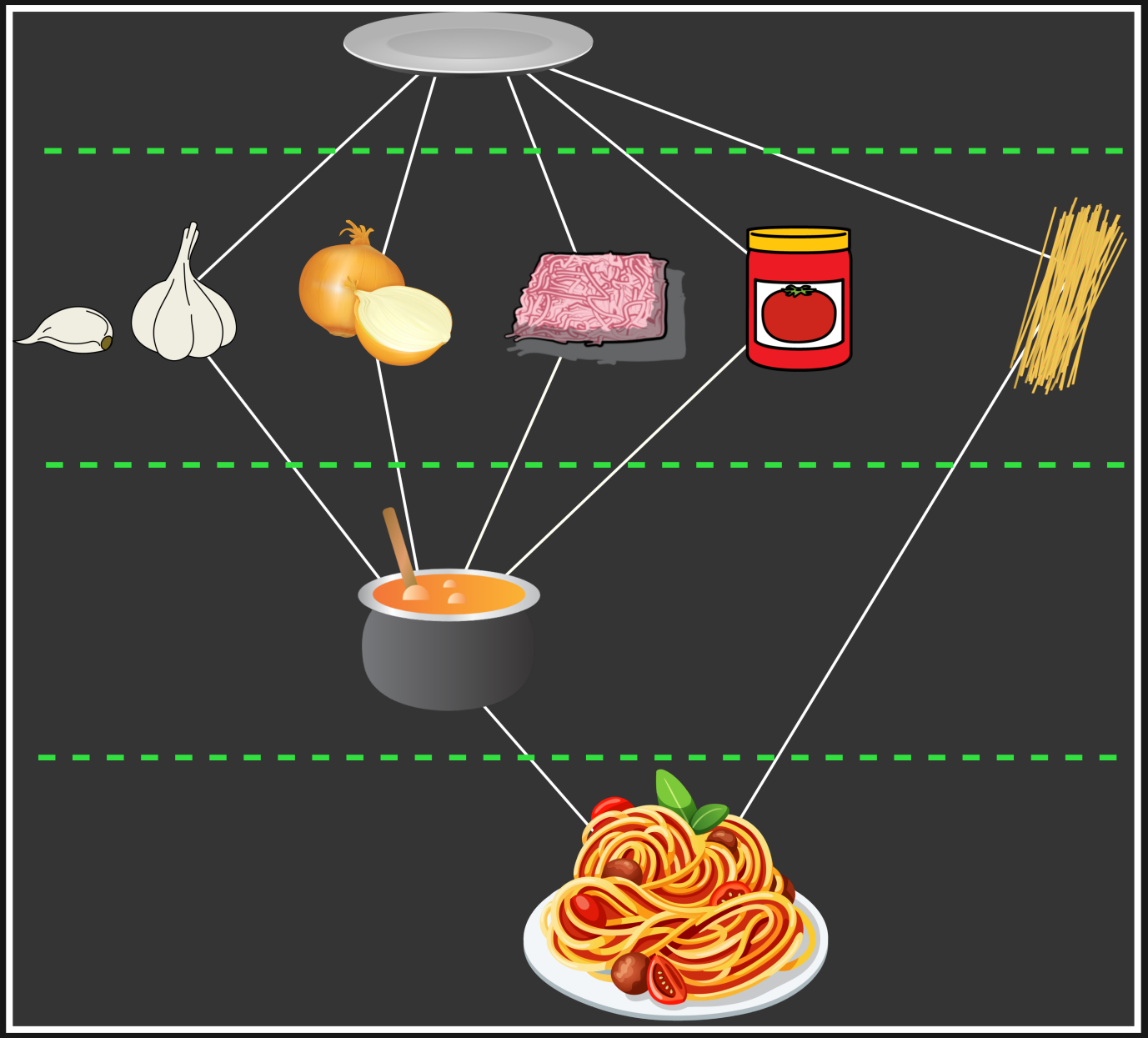
```
...
def requires(self):
return [
ReqTask1(param1='a', param2='b'),
Re1Task2()
]
...
```

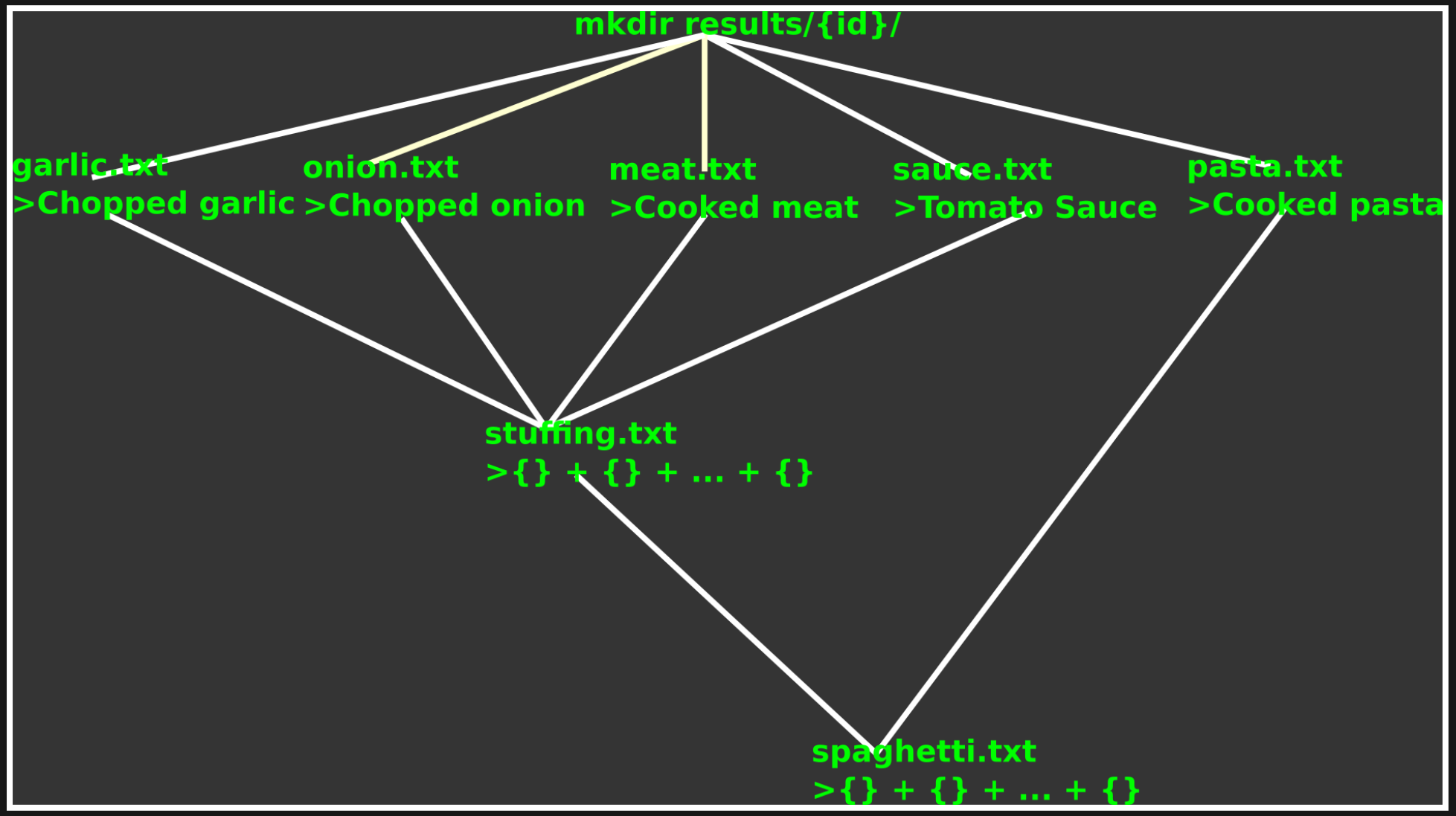
```
...
def run(self):
doSomething()
...
```

```
...
def output(self):
return luigi.LocalTarget(path)
...
```


import luigi
class MyTask(luigi.Task):
param = luigi.Parameter(default=42)
def requires(self):
return SomeOtherTask(self.param)
def run(self):
f = self.output().open('w')
print >>f, "hello, world"
f.close()
def output(self):
return luigi.LocalTarget('/tmp/foo/bar-%s.txt' % self.param)
if __name__ == '__main__':
luigi.run()
The business logic of the task
Where it writes output
What other tasks it depends on
Parameters for this task






mkdir results/{id}/
garlic.txt
>Chopped garlic
onion.txt
>Chopped onion
meat.txt
>Cooked meat
sauce.txt
>Tomato Sauce
pasta.txt
>Cooked pasta
stuffing.txt
>{} + {} + ... + {}
spaghetti.txt
>{} + {} + ... + {}

PREPARE PLATE

```
class PreparePlate(luigi.Task):
    path = luigi.Parameter()
    def run(self):
        time.sleep(5)
        os.makedirs(self.path)

    def output(self):
        return luigi.LocalTarget(self.path)
```

COOK PASTA

```
class CookPasta(luigi.Task):
    path = luigi.Parameter()
    def requires(self):
        return [PreparePlate(path=os.path.dirname(self.path))]

    def run(self):
        time.sleep(5)
        with open(self.path, 'w') as pasta:
            pasta.write('Cooked Pasta')

    def output(self):
        return luigi.LocalTarget(self.path)
```

CHOP

```
class Chop(luigi.Task):
    path = luigi.Parameter()
    ingredient = luigi.Parameter()

    def run(self):
        time.sleep(5)
        with open(self.path, 'w') as ingredient:
            ingredient.write(f'Chopped {self.ingredient}')

    def output(self):
        return luigi.LocalTarget(self.path)

    def requires(self):
        return [PreparePlate(path=os.path.dirname(self.path))]
```

MAKE STUFFING

```
class MakeStuffing(luigi.Task):
    path = luigi.Parameter()
    id = luigi.Parameter(default='test')

    def requires(self):
        return [Chop(path=f'results/{self.id}/garlic.txt',
                     ingredient='garlic'),
                MakeTomatoSauce(
                    path=f'results/{self.id}/sauce.txt'
                    ),
                Chop(path=f'results/{self.id}/onion.txt',
                     ingredient='onion'),
                CookMeat(path=f'results/{self.id}/meat.txt')
            ]
```

```
def run(self):
    time.sleep(5)
    with open(self.input()[0].path, 'r') as garlic_file:
        garlic = garlic_file.read()
    with open(self.input()[1].path, 'r') as sauce_file:
        sauce = sauce_file.read()
    with open(self.input()[2].path, 'r') as onion_file:
        onion = onion_file.read()
    with open(self.input()[3].path, 'r') as meat_file:
        meat = meat_file.read()
    with open(self.path, 'w') as stuffing:
        stuffing.write(f'{garlic} + {sauce} + {onion}')
```

```
def output(self):
    return luigi.LocalTarget(self.path)
```

MAKE SPAGHETTI

```
class MakeSpaghetti(luigi.Task):
    id = luigi.Parameter(default='test')

    def requires(self):
        return [
            CookPasta(path=f'results/{self.id}/pasta.txt'),
            MakeStuffing(
                path=f'results/{self.id}/stuffing.txt',
                id=self.id
            )
        ]
```

```
def run(self):
    time.sleep(5)
    with open(self.input()[0].path, 'r') as pasta_file:
        pasta = pasta_file.read()
    with open(self.input()[1].path, 'r') as stuffing_file:
        stuffing = stuffing_file.read()
    with open(self.output().path, 'w') as spaghetti:
        spaghetti.write(f'{pasta} + {stuffing}')

def output(self):
    path = f'results/{self.id}/spaghetti.txt'
    return luigi.LocalTarget(path)
```

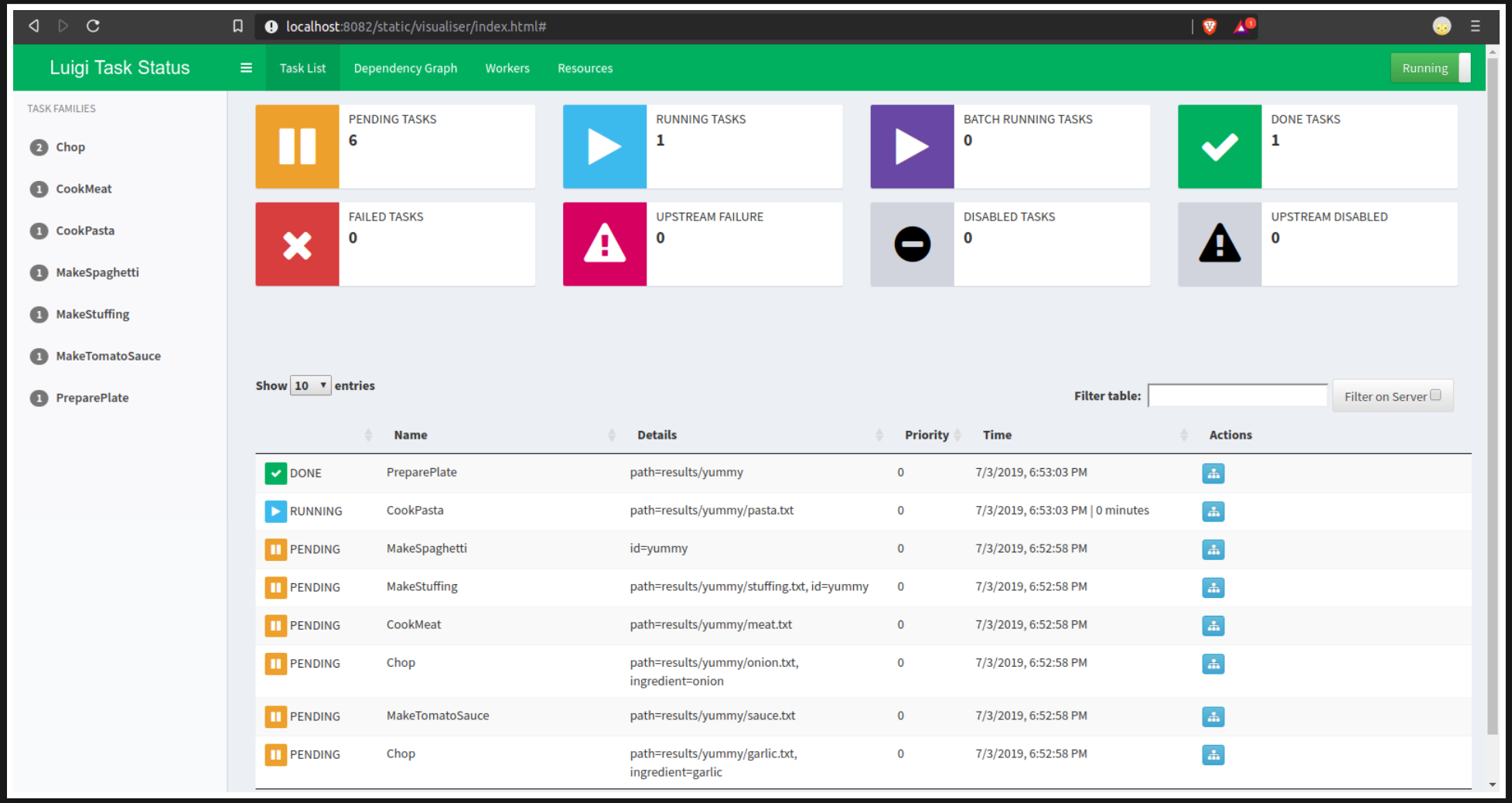
RUN IT!

```
python3 luigi-spaghetti.py MakeSpaghetti --id yummy
```

```
===== Luigi Execution Summary =====
Scheduled 8 tasks of which:
* 8 ran successfully:
- 2 Chop(path=results/yummy/garlic.txt, ingredient=garlic)
    and Chop(path=results/yummy/onion.txt, ingredient=onion)
- 1 CookMeat(path=results/yummy/meat.txt)
- 1 CookPasta(path=results/yummy/pasta.txt)
- 1 MakeSpaghetti(id=yummy)
- 1 MakeStuffing(path=results/yummy/stuffing.txt, id=yummy)
...
This progress looks :) because there
were no failed tasks or missing dependencies
===== Luigi Execution Summary =====
```

```
File: garlic.txt
|Chopped garlic
File: meat.txt
|Cooked Meat
File: onion.txt
|Chopped onion
File: pasta.txt
|Cooked Pasta
File: sauce.txt
|Tomato Sauce
File: spaghetti.txt
|Cooked Pasta + Chopped garlic + Tomato Sauce + Chopped onion
File: stuffing.txt
|Chopped garlic + Tomato Sauce + Chopped onion
```

Dashboard


localhost:8082/static/visualiser/index.html#
Luigi Task Status
Task List
Dependency Graph
Workers
Resources
Running
TASK FAMILIES
2 Chop
1 CookMeat
1 CookPasta
1 MakeSpaghetti
1 MakeStuffing
1 MakeTomatoSauce
1 PreparePlate
PENDING TASKS 6
RUNNING TASKS 1
BATCH RUNNING TASKS 0
DONE TASKS 1
FAILED TASKS 0
UPSTREAM FAILURE 0
DISABLED TASKS 0
UPSTREAM DISABLED 0
Show 10 entries
Filter table:
Filter on Server
Name
Details
Priority
Time
Actions
DONE PreparePlate path=results/yummy 0 7/3/2019, 6:53:03 PM
RUNNING CookPasta path=results/yummy/pasta.txt 0 7/3/2019, 6:53:03 PM | 0 minutes
PENDING MakeSpaghetti id=yummy 0 7/3/2019, 6:52:58 PM
PENDING MakeStuffing path=results/yummy/stuffing.txt, id=yummy 0 7/3/2019, 6:52:58 PM
PENDING CookMeat path=results/yummy/meat.txt 0 7/3/2019, 6:52:58 PM
PENDING Chop path=results/yummy/onion.txt, ingredient=onion 0 7/3/2019, 6:52:58 PM
PENDING MakeTomatoSauce path=results/yummy/sauce.txt 0 7/3/2019, 6:52:58 PM
PENDING Chop path=results/yummy/garlic.txt, ingredient=garlic 0 7/3/2019, 6:52:58 PM

Thanks!

github.com/Dysproz/luigi-spaghetti