“ We're going to move your hub, the center of your digital life,
into the ***CLOUD***
...If you don't think we're serious about this, you're wrong. ”

- Steve Jobs, WWDC 2011

“ Gifts are easy …
Choices can be hard …
In the end, we are our choices.

Build yourself a great story.”

- Jeff Bezos, 2010

PYTHON 遇到 AMAZON

– 如何开发云时代的应用

费良宏 , AWS EVANGELIST / 2015.04

PYTHON 是什么?

- 一种通用的、高级程序设计语言
- 一个荷兰人(**Guido van Rossum**)的设计 / 1991年
- 一个榜单(**TIBOE**)上排名第8的语言 / 2015年4月
- 一大批公司在使用它

  Google, Youtube, Instagram, Pinterest, Bing, Reddit, Etsy, Dropbox, Quora, Yelp, Friendfeed, Trulia, Hunch, Blogger and Rdio…

- 一些软件用它开发

  Autodesk Maya, GIMP, BitTorrent, Blender 3D, Cinema 4D, Dropbox, OpenStack, YUM, OpenERP, Civilization IV, Matplotlib, FreeCAD, MySQL Workbench, Nuke…

开发者眼中的 **PYTHON**

- 在一些人眼中它像一把瑞士军刀，无所不能

  “ Python allows us to produce maintainable features in record times, with a minimum of developers.”
  - Cuong Do, Software Architect, Youtube

- 一些人觉得它像充满争议，并不完美

  Python 2 vs Python 3, Significant white-space, Performance, Global Interpreter Lock(GIL)…

PYTHON之禅

美胜于丑　规则胜于特例
显胜于隐　实用胜于单纯
简胜于繁　告错胜于沉默
繁胜于杂　沉默胜于吵闹
平胜于迭　拒绝胜于猜测
疏胜于密　唯一胜于显然
读胜于写　显然不是作者
现在胜于永不　永不胜于匆猝
值得说则必说　命名空间很好

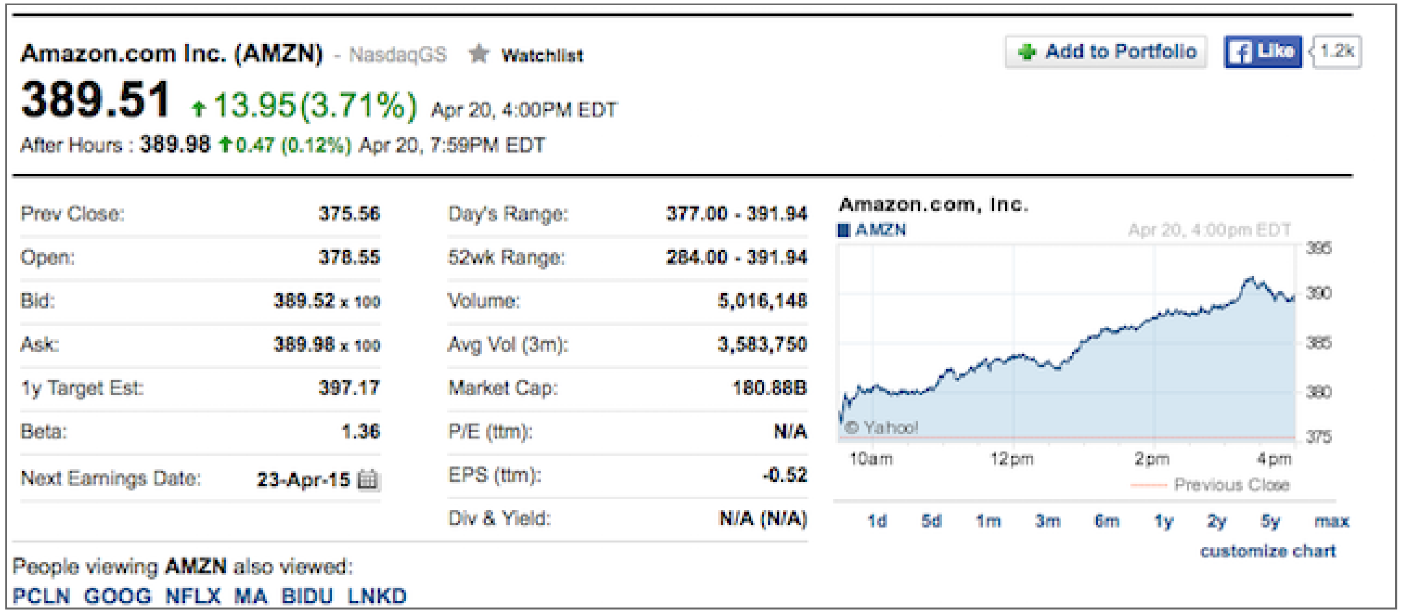
- *import this (PEP 20)*

爱上**PYTHON**的理由

让我保持专注
简化的原则，读比写要多
不会有向后兼容的痛苦
效率之上再谈性能
不把我当作傻瓜
不把我当作傻瓜
不假设如何发现错误
喜欢的人毋需多说
键入不多
合乎人性

- *Bruce Eckle*, "*Why I Love Python*"

一小段PYTHON代码可以说明



Amazon.com Inc. (AMZN) - NasdaqGS ★ Watchlist

Add to Portfolio | Like 1.2k

389.51 ↑13.95(3.71%) Apr 20, 4:00PM EDT

After Hours : 389.98 ↑0.47 (0.12%) Apr 20, 7:59PM EDT

Prev Close:	375.56	Day's Range:	377.00 - 391.94
Open:	378.55	52wk Range:	284.00 - 391.94
Bid:	389.52 x 100	Volume:	5,016,148
Ask:	389.98 x 100	Avg Vol (3m):	3,583,750
1y Target Est:	397.17	Market Cap:	180.88B
Beta:	1.36	P/E (ttm):	N/A
Next Earnings Date:	23-Apr-15	EPS (ttm):	-0.52
		Div & Yield:	N/A (N/A)

People viewing AMZN also viewed:
PCLN GOOG NFLX MA BIDU LNKD

```
import urllib.request
from bs4 import BeautifulSoup as bs4
htmlfile = urllib.request.urlopen("http://finance.yahoo.com/q?s=AMZN")
htmltext = htmlfile.read()
soup = bs4(htmltext)
for price in soup.find_all(id="yfs_l84_amzn"):
     print (price)
```

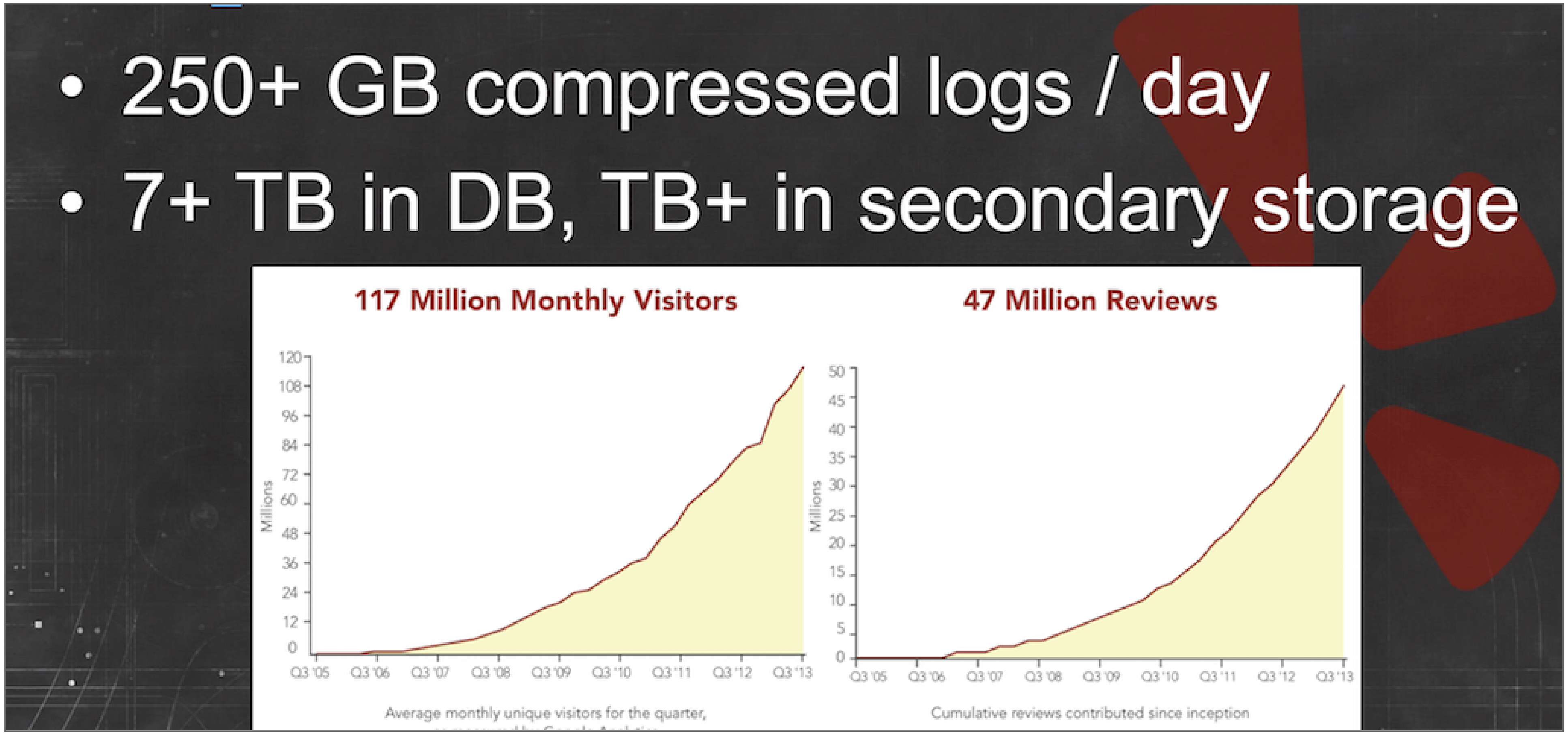
有没关于 PYTHON 的励志故事?

听过过<u>**YELP**</u>吗?

Yelp!

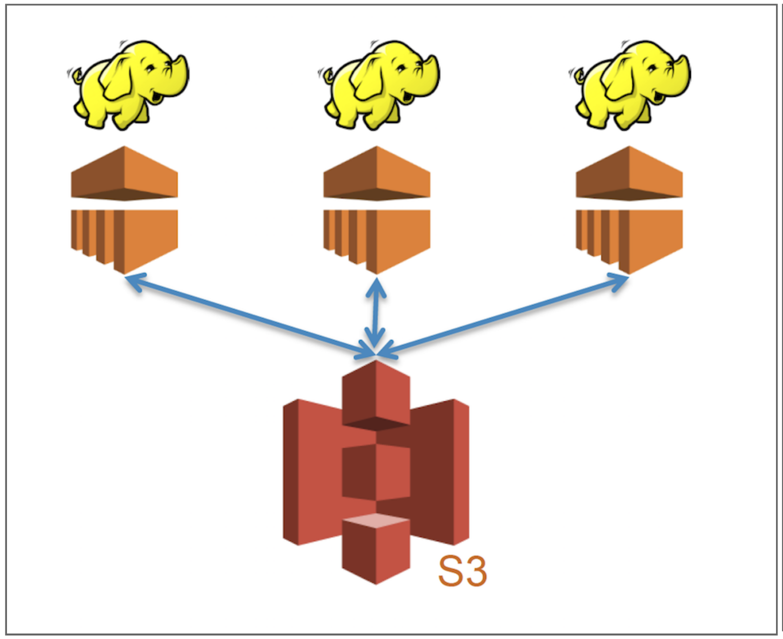
- Connects people with great local businesses
- Connects me with Thomas Keller

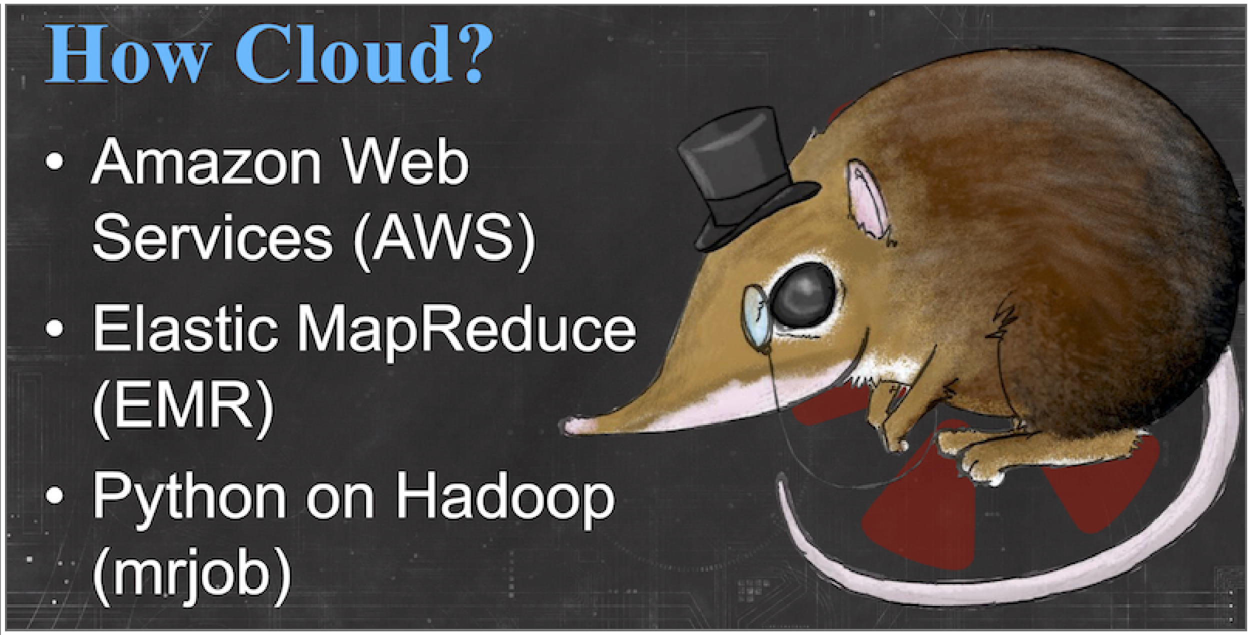
YELP的大数据难题



- 250+ GB compressed logs / day
- 7+ TB in DB, TB+ in secondary storage

YELP应对之道

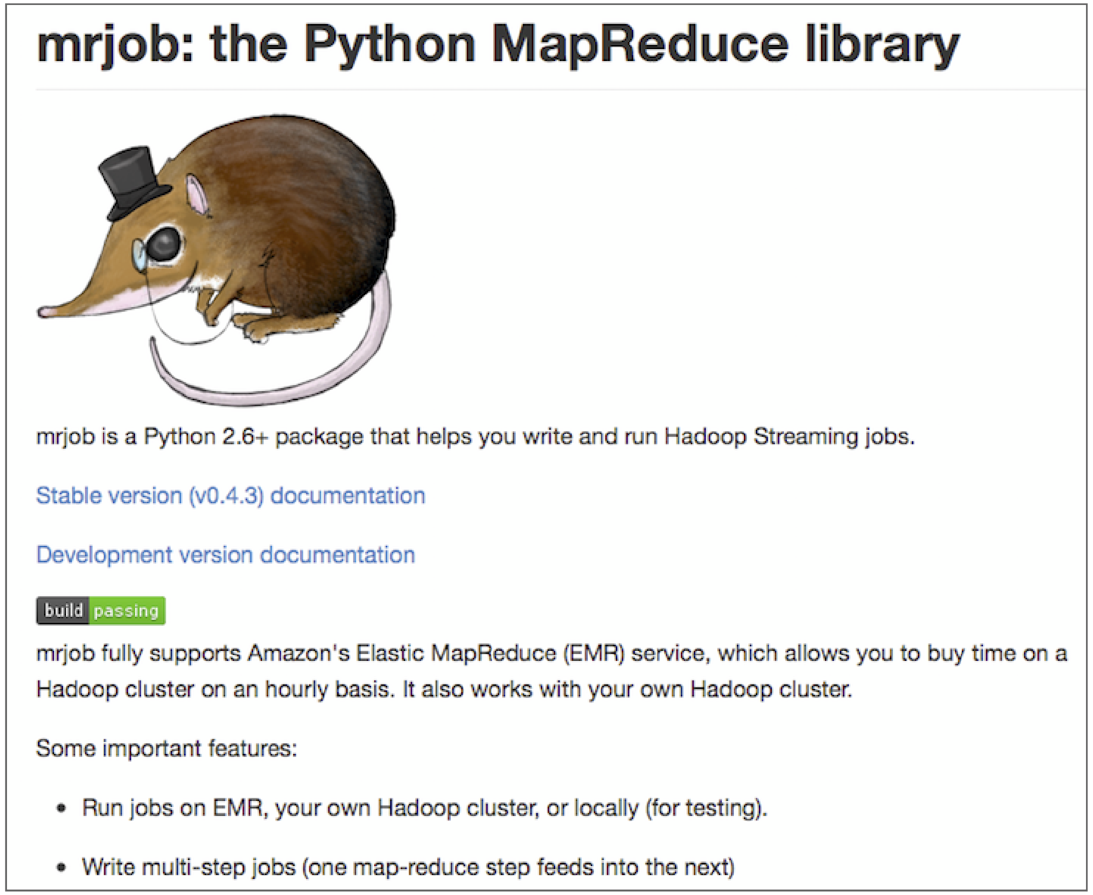




How Cloud?

- Amazon Web Services (AWS)
- Elastic MapReduce (EMR)
- Python on Hadoop (mrjob)

开源项目**MRJOB**



mrjob: the Python MapReduce library

mrjob is a Python 2.6+ package that helps you write and run Hadoop Streaming jobs.

Stable version (v0.4.3) documentation

Development version documentation

build passing

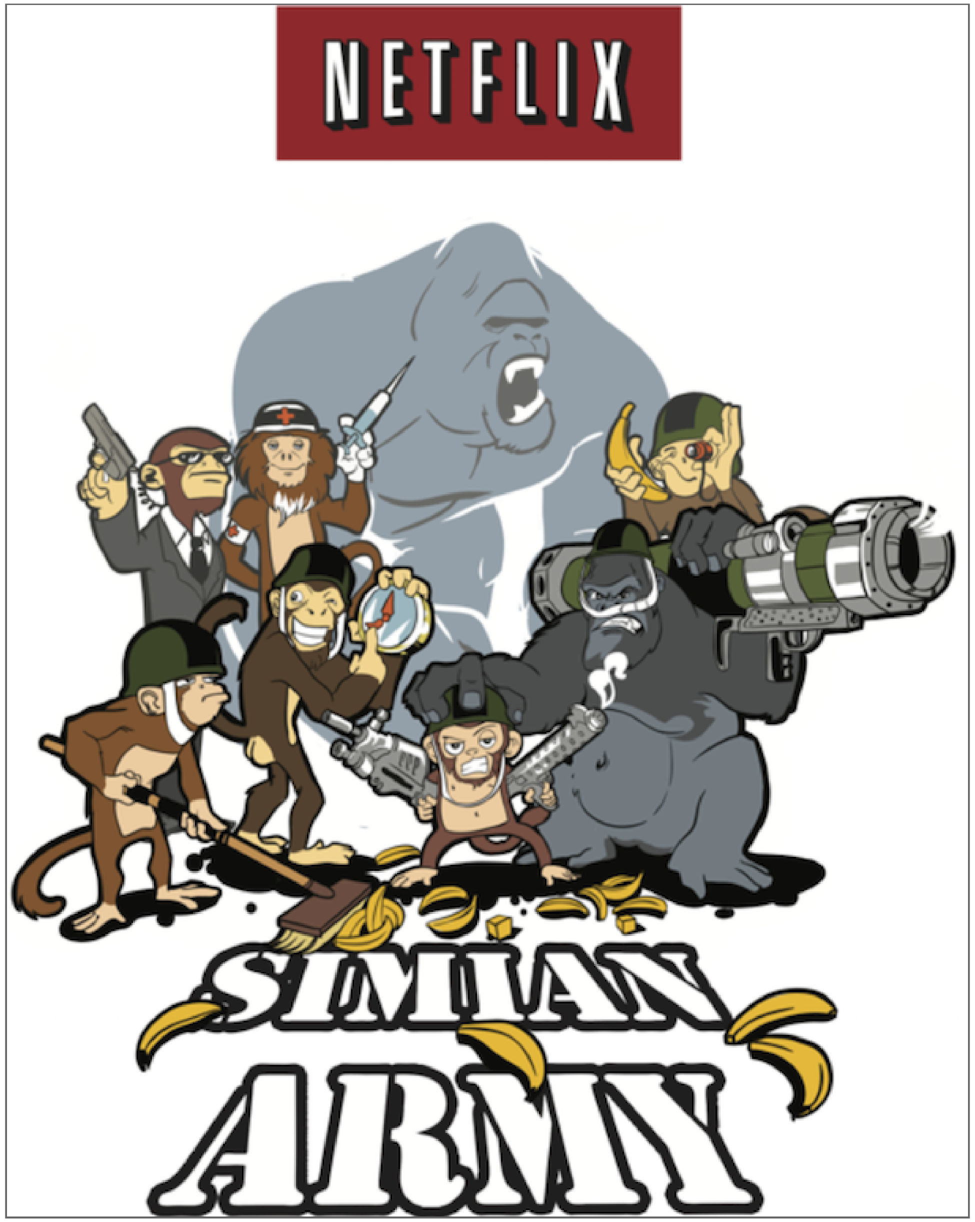
mrjob fully supports Amazon's Elastic MapReduce (EMR) service, which allows you to buy time on a Hadoop cluster on an hourly basis. It also works with your own Hadoop cluster.

Some important features:

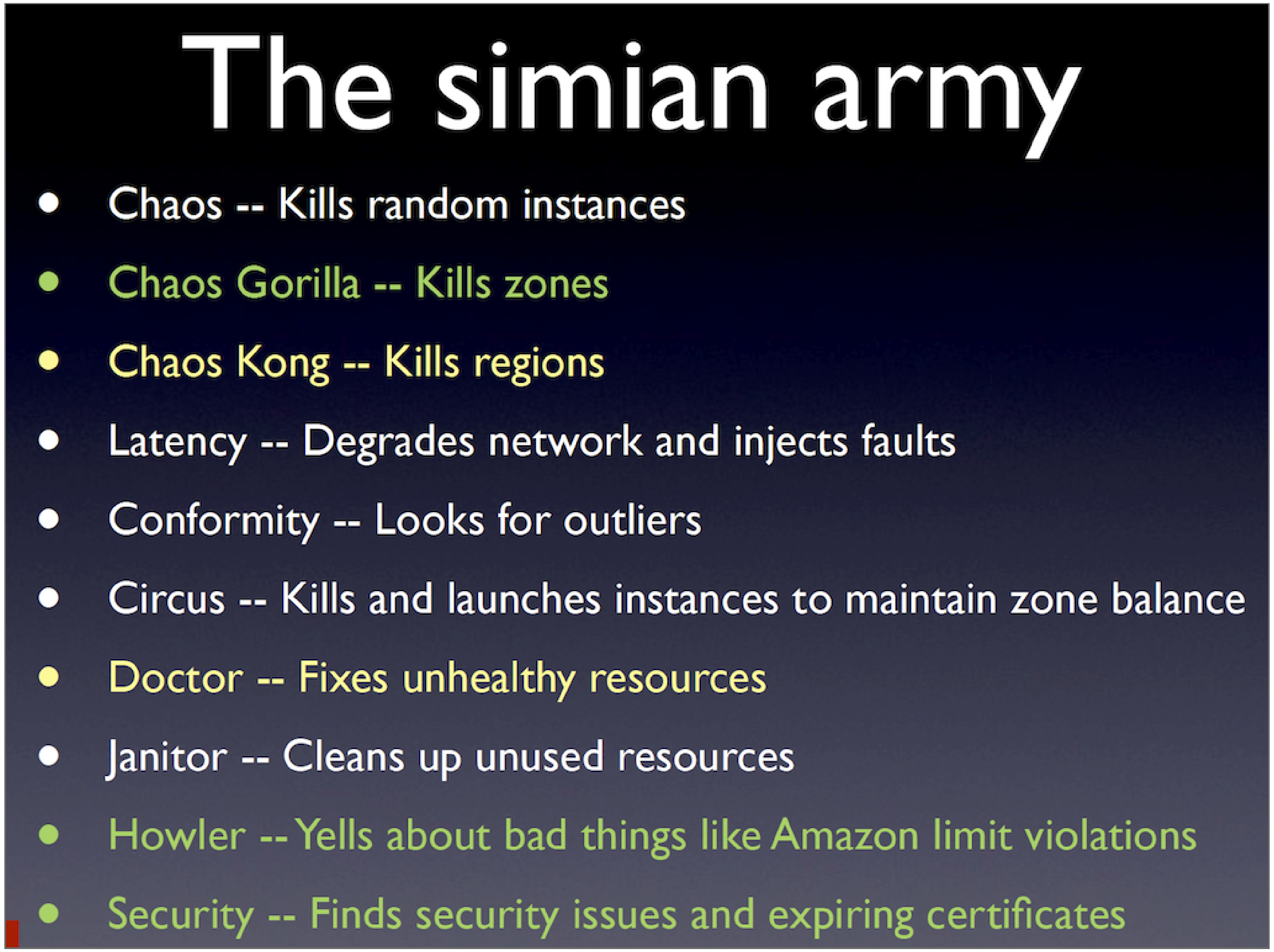
- Run jobs on EMR, your own Hadoop cluster, or locally (for testing).
- Write multi-step jobs (one map-reduce step feeds into the next)

```
"""The classic MapReduce job: count the frequency of words.
"""
from mrjob.job import MRJob
import re

WORD_RE = re.compile(r"[\w']+")


class MRWordFreqCount(MRJob):

    def mapper(self, _, line):
        for word in WORD_RE.findall(line):
            yield (word.lower(), 1)

    def combiner(self, word, counts):
        yield (word, sum(counts))

    def reducer(self, word, counts):
        yield (word, sum(counts))


if __name__ == '__main__':
     MRWordFreqCount.run()
```

你一定知道的NETFLIX



NETFLIX的PYTHON工具箱



The simian army

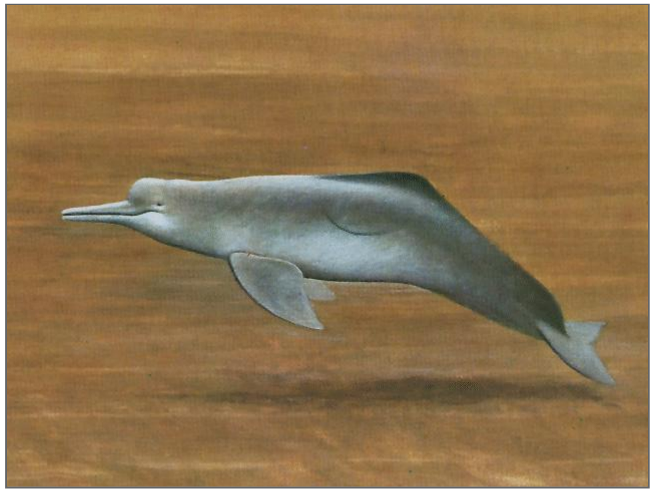
- Chaos -- Kills random instances
- Chaos Gorilla -- Kills zones
- Chaos Kong -- Kills regions
- Latency -- Degrades network and injects faults
- Conformity -- Looks for outliers
- Circus -- Kills and launches instances to maintain zone balance
- Doctor -- Fixes unhealthy resources
- Janitor -- Cleans up unused resources
- Howler -- Yells about bad things like Amazon limit violations
- Security -- Finds security issues and expiring certificates

项目地址 https://netflix.github.com/

成功的故事里面有<u>PYTHON</u>、有<u>AMAZON WEB SERVICE</u>，还有什么你不知道的…

<u>Boto</u>

云计算应用开发利器BOTO



什么是 Boto
- 面向 Python 开发的 AWS SDK

版本
- Botto 2 (Stable), Boto 3 (Preview)

功能
- 支持 40+ AWS 服务 (S3, EC2, EMR…)

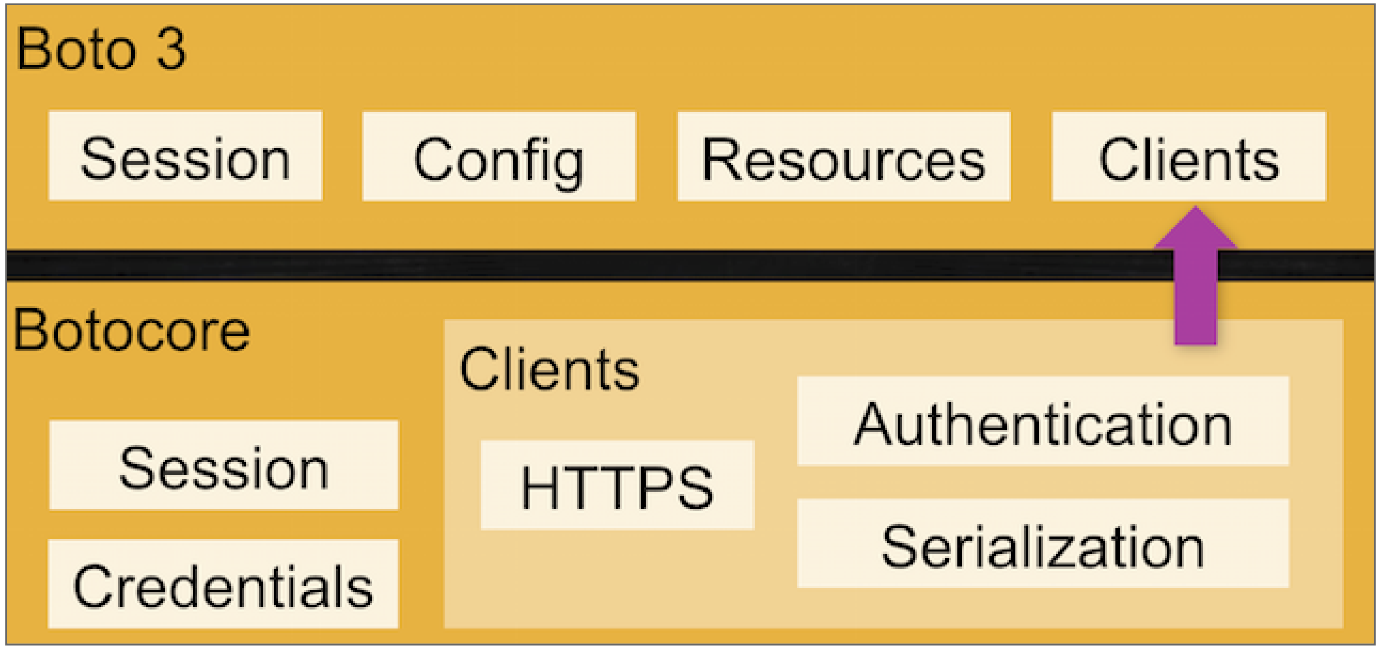
BOTO 3 的特性

- 设计之初已支持 **Python 2 & Python 3**
- 从底层支持的数据驱动
- 针对AWS 新的服务的支持
- 一致性的使用界面
- 现代风格的面向对象的 API
- 与 **Boto 2** 共存
- 可在现有的 **Boto 2** 的程序中使用
- 基于**Apache**许可的开源项目

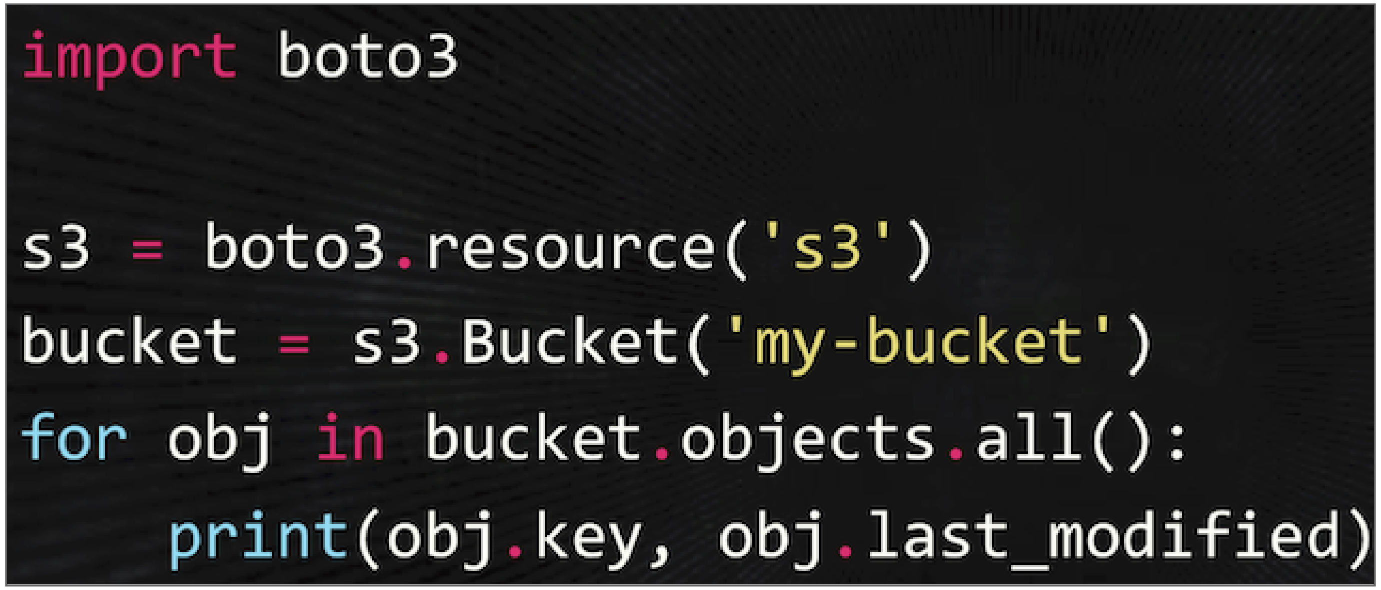
BOTO 3 中的概念

- 资源(Resource) - 高级的面向对象的接口
- 集合(Collection) - 迭代操作资源组的工具
- 客户端(Clients) - 低层次服务连接
- 分页器(Paginators) - 响应自动分页
- 阻塞(Waiters) - 一种阻塞的方法，直到特定的状态满足
- **Botocore** - 提供低级别的客户端、会话、凭据以及配置数据，与**AWS CLI** 共享

BOTO 3 的设计架构

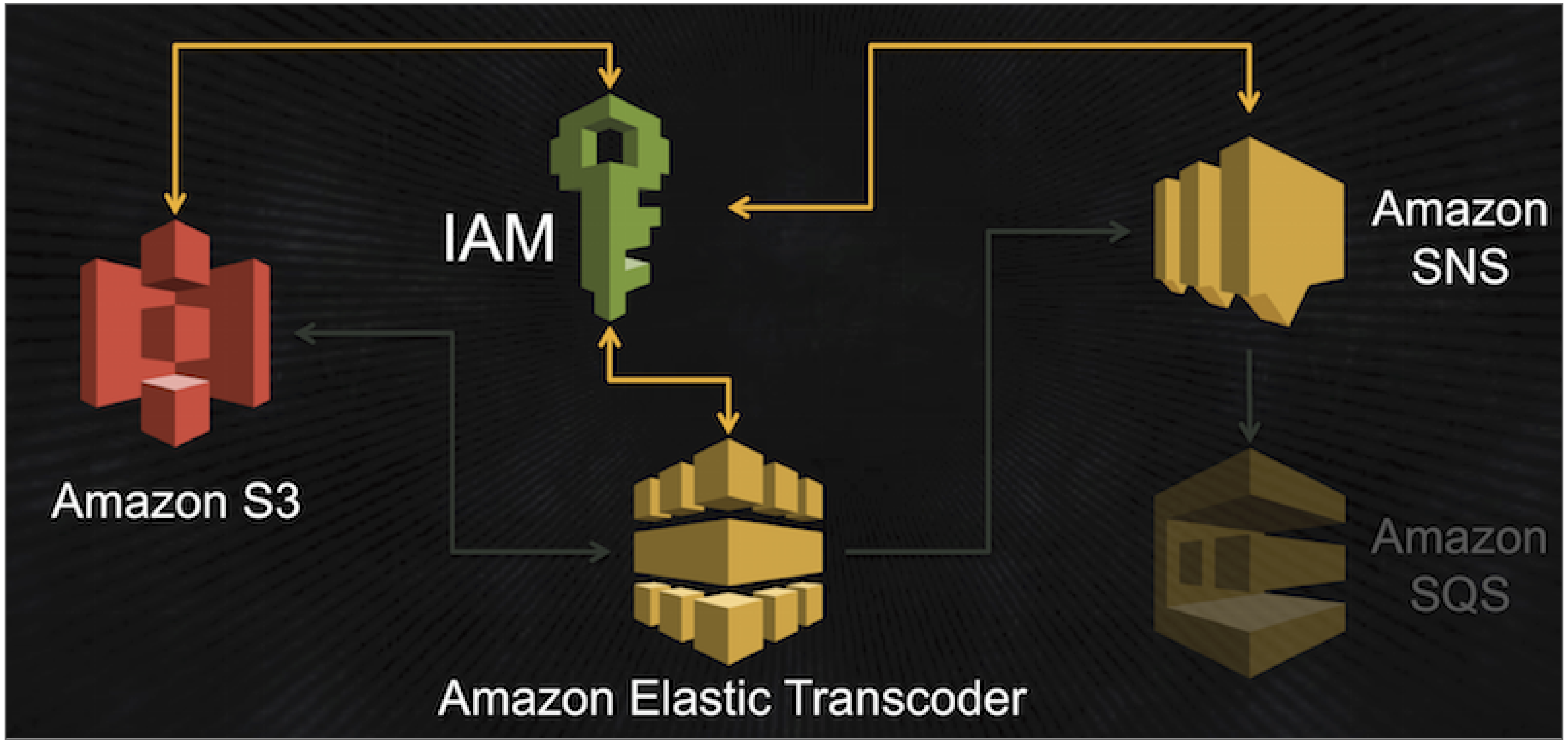

Boto 3
Session
Config
Resources
Clients
Botocore
Session
Credentials
Clients
HTTPS
Authentication
Serialization

BOTO 3 的样例代码



```python
import boto3

s3 = boto3.resource('s3')
bucket = s3.Bucket('my-bucket')
for obj in bucket.objects.all():
    print(obj.key, obj.last_modified)
```

BOTO 3 的应用实例－视频转码

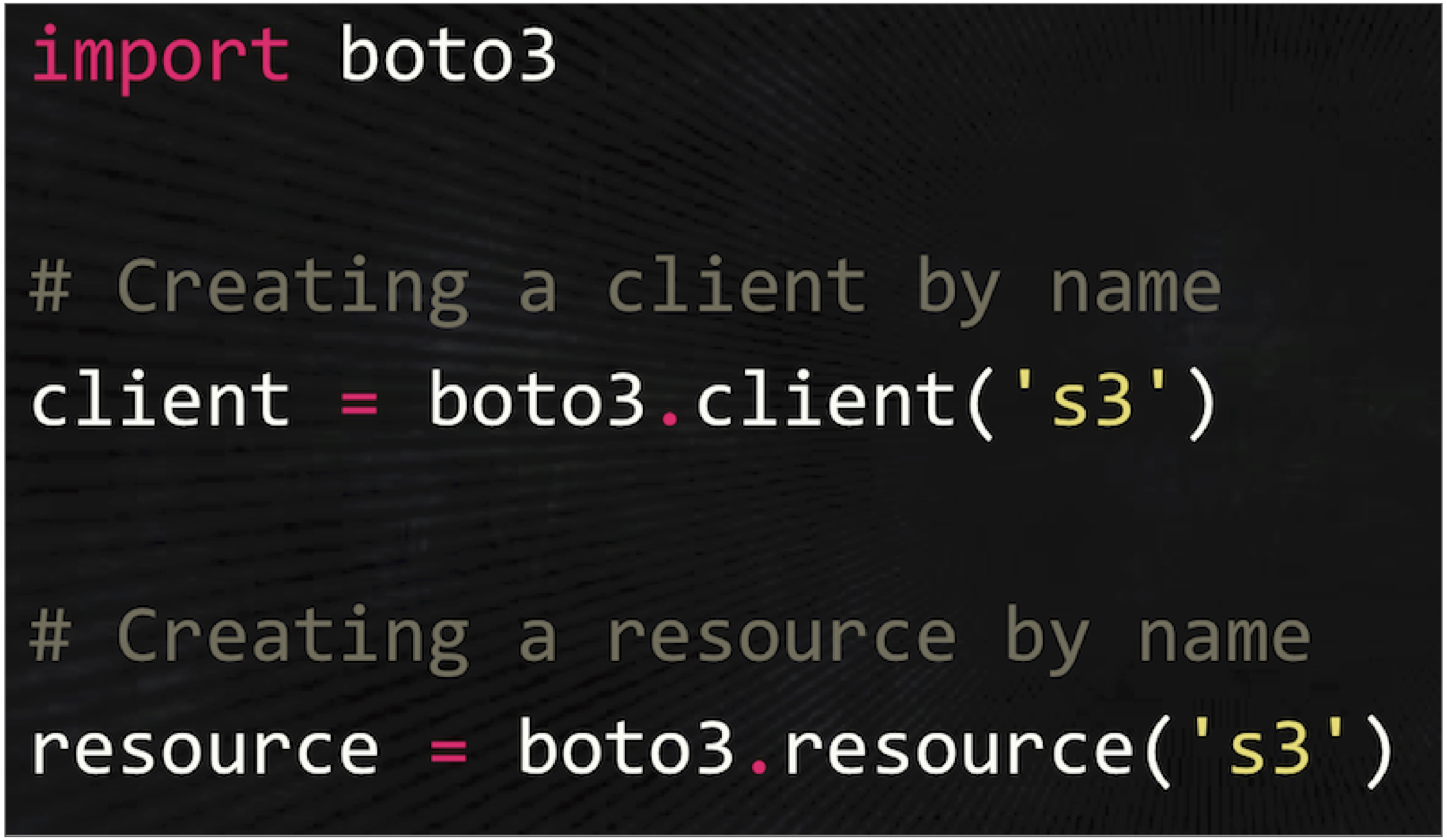


源代码- **https://github.com/boto/boto3-sample**

BOTO 3 应用－视频转码的基本功能

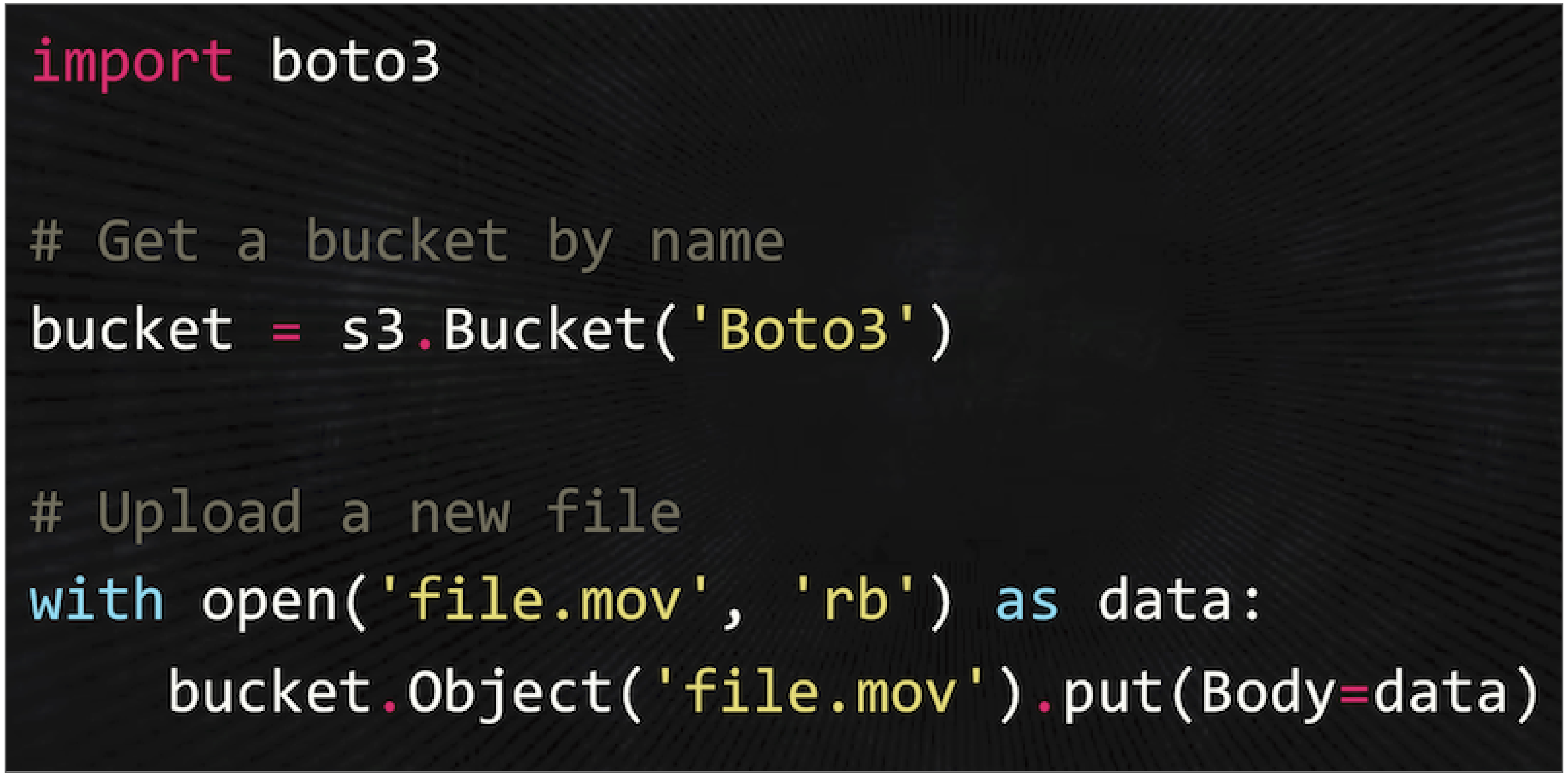
- **AWS Elastic Transcoder** requires configuration
- **AWS S3** bucket for input
- **AWS S3** bucket for output
- **AWS SNS** topic for notifications
- **IAM** role for access
- Transcoding pipeline
- Transcoding job

BOTO 3 视频转码－会话&连接



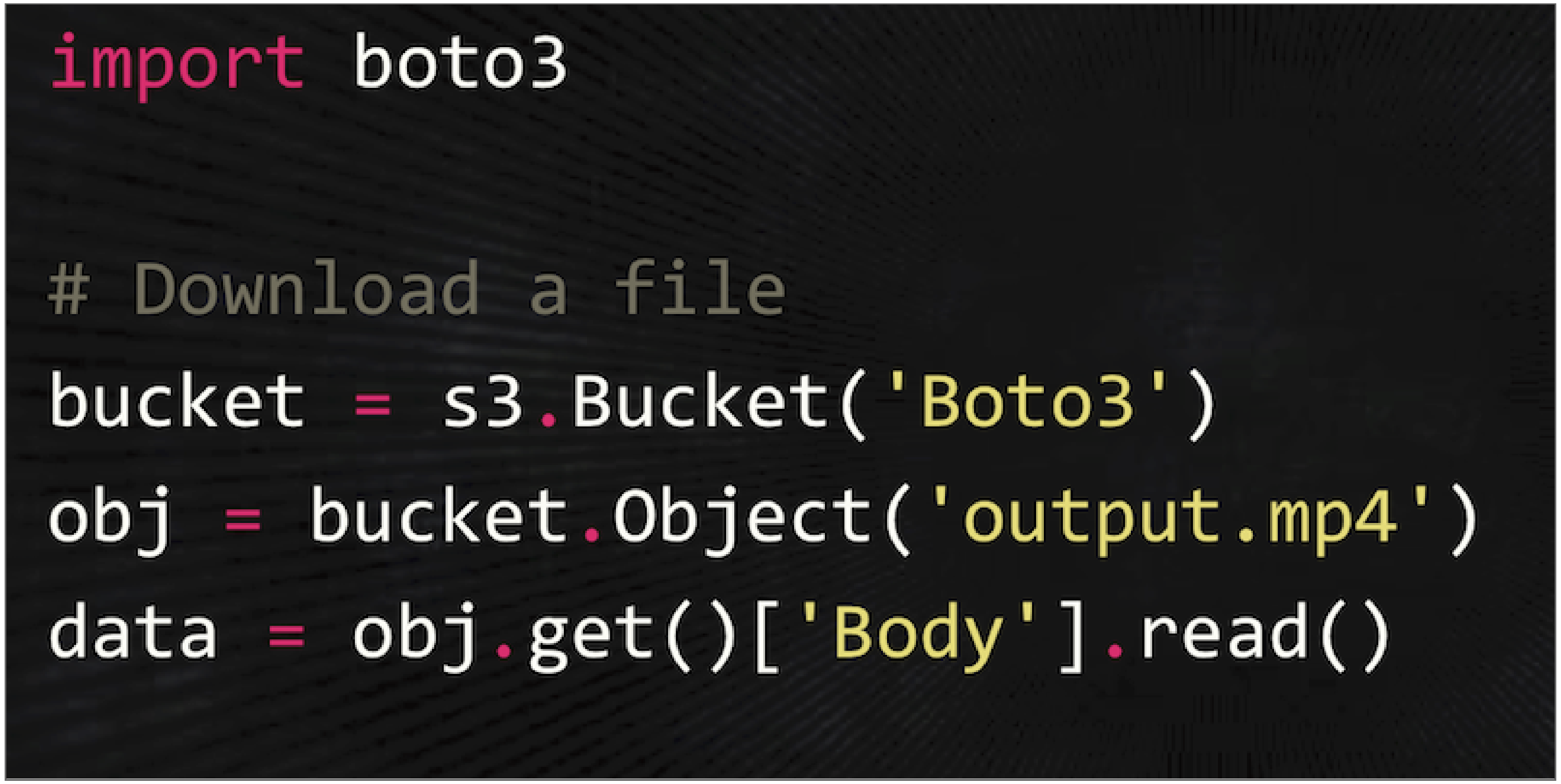
```
import boto3

# Creating a client by name
client = boto3.client('s3')

# Creating a resource by name
resource = boto3.resource('s3')
```

BOTO 3 视频转码－AWS S3 上传文件



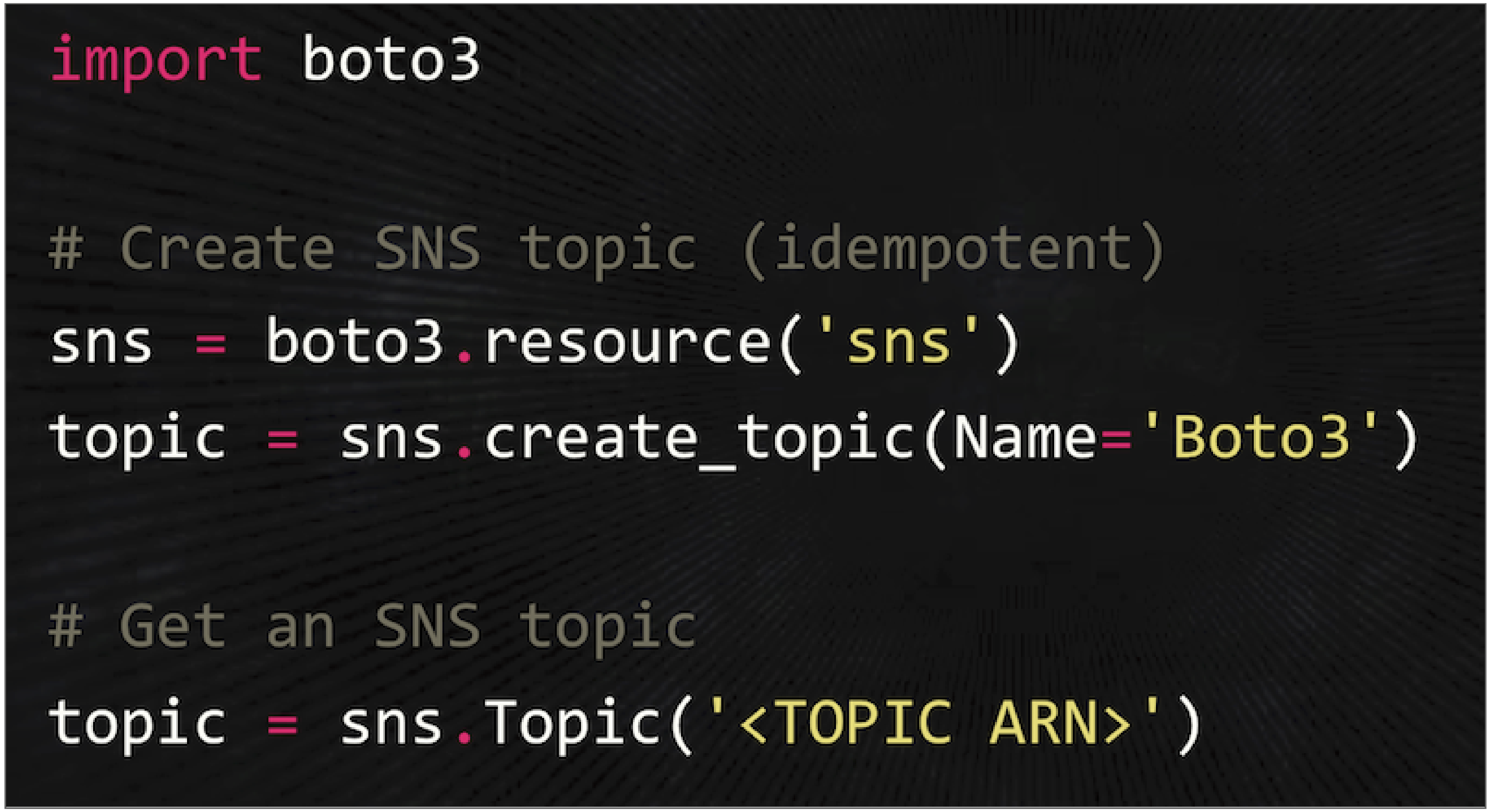
```python
import boto3

# Get a bucket by name
bucket = s3.Bucket('Boto3')


# Upload a new file
with open('file.mov', 'rb') as data:
    bucket.Object('file.mov').put(Body=data)
```

BOTO 3 视频转码－AWS S3 下载文件



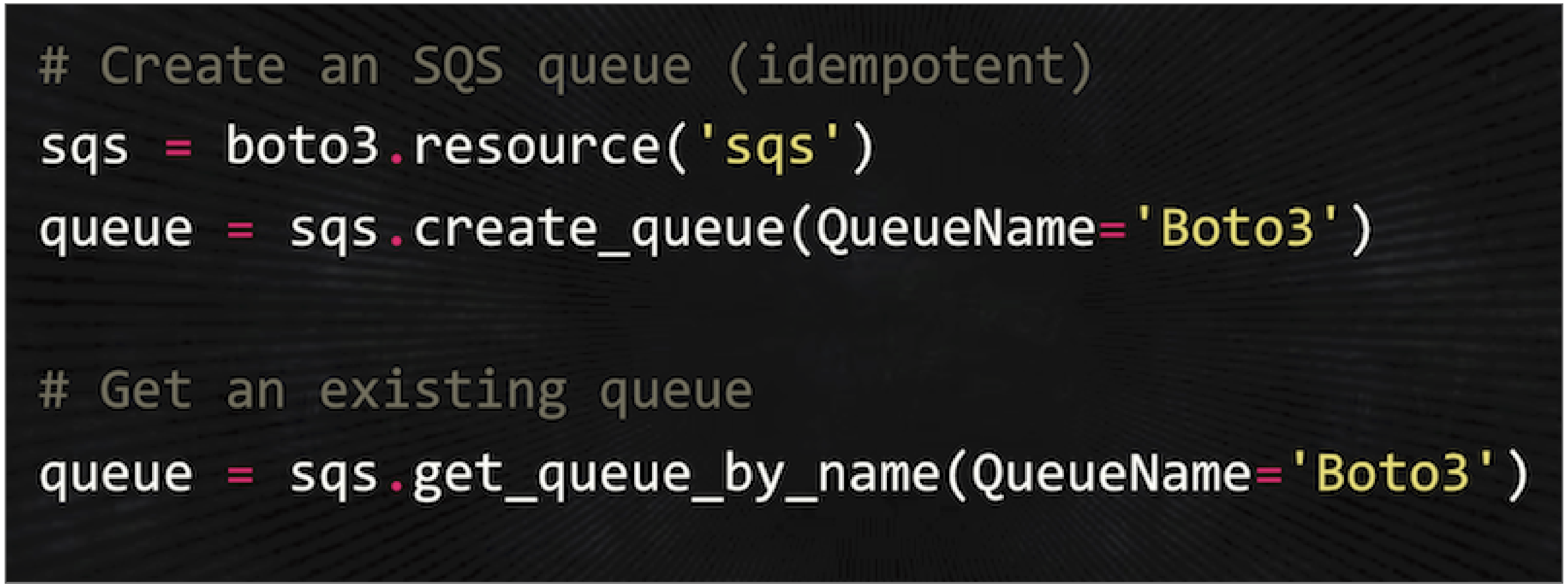
```
import boto3

# Download a file
bucket = s3.Bucket('Boto3')
obj = bucket.Object('output.mp4')
data = obj.get()['Body'].read()
```

BOTO 3 视频转码 – AWS SNS TOPIC



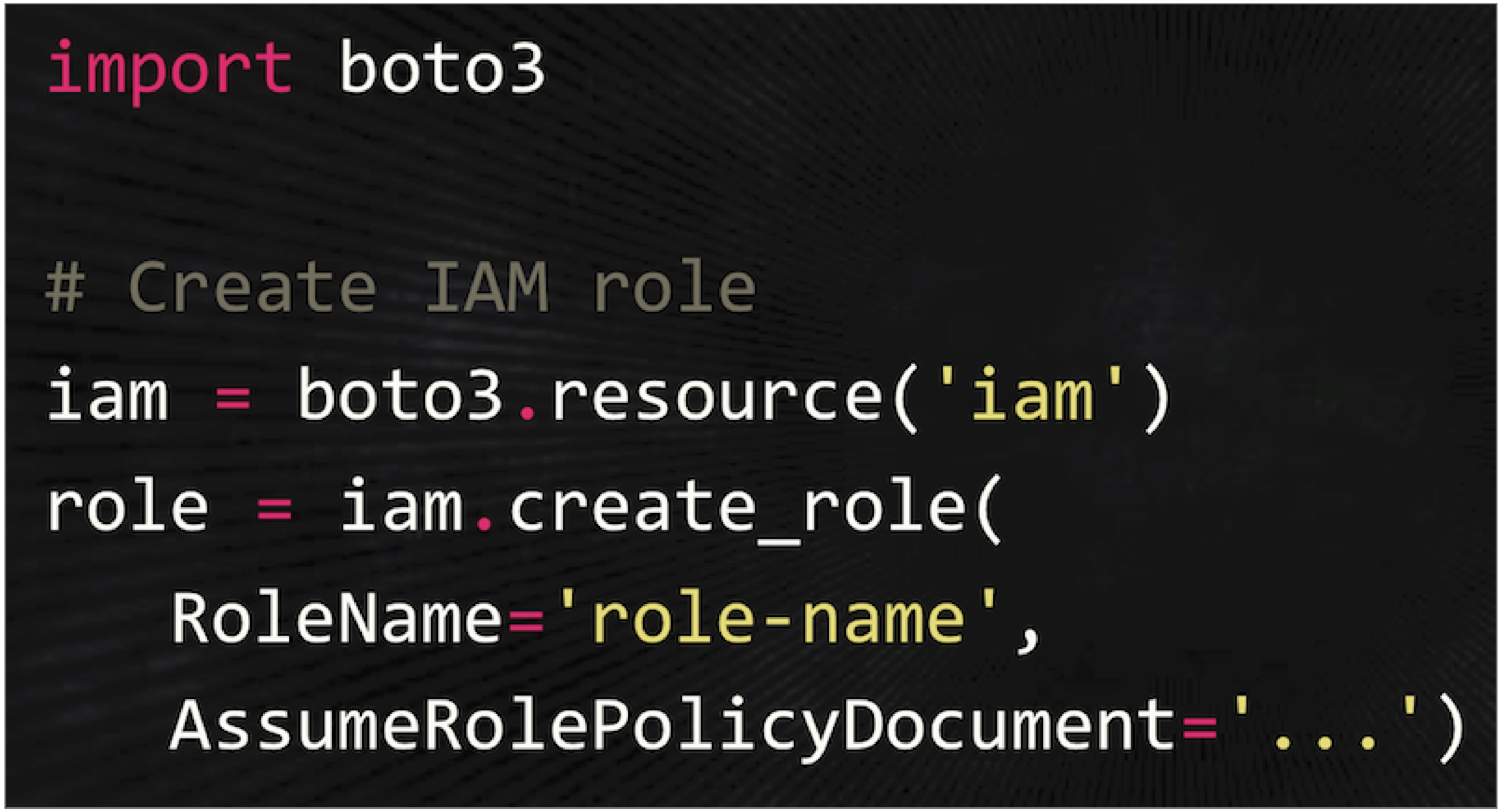
```python
import boto3

# Create SNS topic (idempotent)
sns = boto3.resource('sns')
topic = sns.create_topic(Name='Boto3')

# Get an SNS topic
topic = sns.Topic('<TOPIC ARN>')
```

BOTO 3 视频转码 – AWS SQS QUEUE



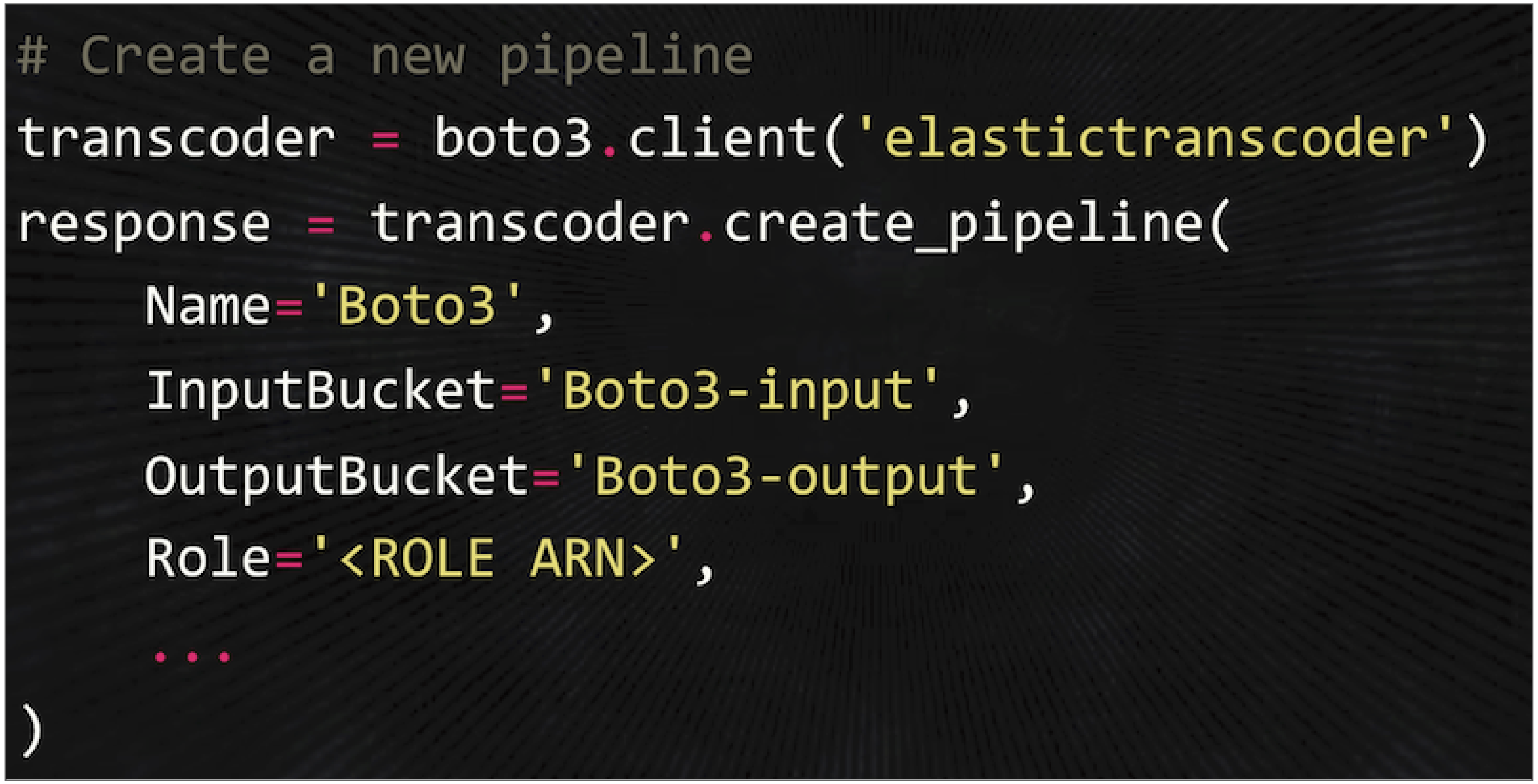
```
# Create an SQS queue (idempotent)
sqs = boto3.resource('sqs')
queue = sqs.create_queue(QueueName='Boto3')

# Get an existing queue
queue = sqs.get_queue_by_name(QueueName='Boto3')
```

BOTO 3 视频转码－AWS IAM ROLE



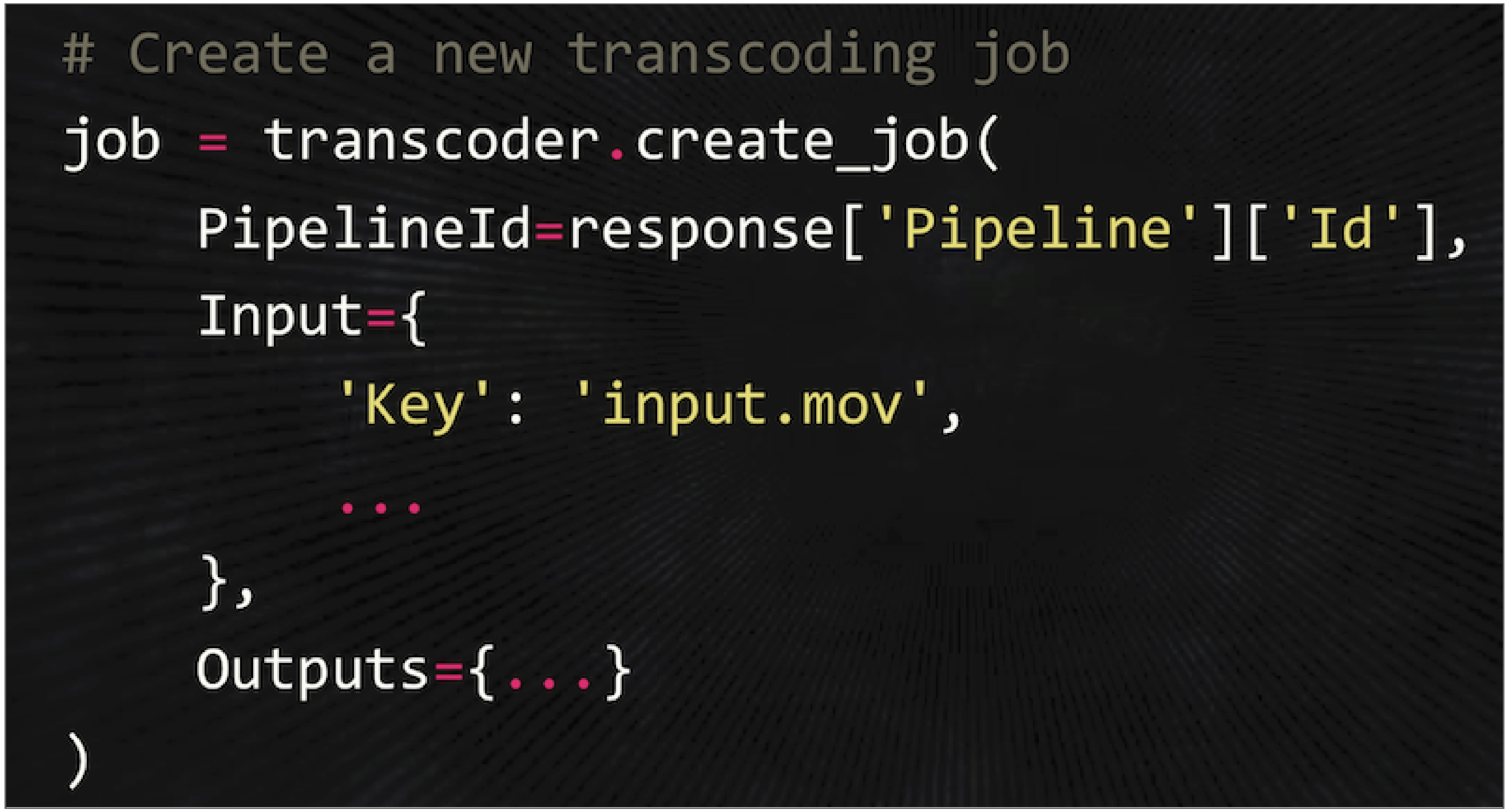
```python
import boto3

# Create IAM role
iam = boto3.resource('iam')
role = iam.create_role(
    RoleName='role-name',
    AssumeRolePolicyDocument='...')
```

BOTO 3 视频转码 – AWS TRANSCODING PIPELINE



```python
# Create a new pipeline
transcoder = boto3.client('elastictranscoder')
response = transcoder.create_pipeline(
    Name='Boto3',
    InputBucket='Boto3-input',
    OutputBucket='Boto3-output',
    Role='<ROLE ARN>',
    ...
)
```

BOTO 3 视频转码 – AWS TRANSCODING JOB



```
# Create a new transcoding job
job = transcoder.create_job(
    PipelineId=response['Pipeline']['Id'],
    Input={
        'Key': 'input.mov',
        ...
    },
    Outputs={...}
)
```

我的私人 PYTHON 收藏

- 数据分析
  - iPython, http://ipython.org/
  - Numpy, http://www.numpy.org/
  - Pandas, http://pandas.pydata.org/
  - Scipy, http://www.scipy.org/
  - Matplotlib, http://matplotlib.org/
- 机器学习
  - Scikit-learn, http://scikit-learn.org/stable/index.html
- Web 数据获取
  - Scrapy, http://http://scrapy.org/
- Web 应用框架
  - Flask, http://flask.pocoo.org/

…

谢谢！

- 体验 Amazon Web Services
- 开始你的 Boto 开发之旅