Jupyter as Your Incident Management Tool
DevOps as Science

Moshe Zadka – https://cobordism.com

2020
Acknowledgement of Country

Belmont (in San Francisco Bay Area Peninsula)  
Ancestral homeland of the Ramaytush Ohlone
Jupyter Notebooks

- Language agnostic console
- Saves inputs/outputs in JSON
- Can export to HTML, Markdown, etc.
Jupyter Kernels

- Language execution engine
- IPyKernel in a virtual environment
Data Science Origins

- Sharing work
- Executable format
- Exploration
Imaginary Stack

- Some vague "Service"
- Our team is in charge of its SLAs
- As luck would have it, a page comes in...
Symptom

- P90 latency is spiking
- Each server has different "typical behavior"
- Figure out which server is misbehaving and fix
Not a detective story!

Core file dumped resulting in some queries take too long
image = Image.open("prometheus.png")
width, height = image.size
image.resize((width//4, height//4))

Figure: png
Query Data

```python
now = datetime.datetime.now()
five_minutes_agp = (now - datetime.timedelta(minutes=5))
data = httpx.get(
    f"http://localhost:9090/api/v1/query_range""?query=level"
    f"&start={five_minutes_agp.timestamp()}"
    f"&end={now.timestamp()}&step=10s")
    ).json()

(data["data"]["result"][0]["metric"],
data["data"]["result"][0]["values"][2:2])

({'__name__': 'level',
  'instance': 'localhost:8080',
  'job': 'my_service',
  'which': 'object1'},
  [[1605230830.588, '0.11407403763077273'],
   [1605230840.588, '0.6268484040376028']])
```
Analyze Data

```python
amplitudes = {}
for result in data["data"]["result"]:  
    which = result["metric"]["which"]
    values = [float(value) 
        for when, value in result["values"]]
    amplitudes[which] = max(values) - min(values)
max(amplitudes.items(),
    key=operator.itemgetter(1))

('object4', 13.641418227003584)
```
We Know The Culprit

That dastardly object!
We Know The Culprit

That dastardly object! Now let’s find out what the problem is.
with open("aws_params.json") as fpin:
    params = json.loads(fpin.read())

ec2 = boto3.client("ec2", **params)
instances = ec2.describe_instances()
for reservation in instances['Reservations']:
    for instance in reservation['Instances']:
        for tag in instance.get('Tags', []):
            if (tag['Key'] == 'kind'
                and tag['Value'] == 'object4'):
                object4_instance = instance
                print(object4_instance['InstanceId'])

i-6e02ec9badfdcc9f5
We Know The Culprit

We Found the instance
We Know The Culprit

We Found the instance
i-something-or-other
We Know The Culprit

We Found the instance
i-something-or-other
Time to fix the issue
Connect to Host

```python
client = paramiko.SSHClient()
client.set_missing_host_key_policy(paramiko.client.WarningPolicy)
client.connect(
    object4_instance["PrivatelpAddress"],
    username="root")

/home/moshez/.virtualenvs/jupyter-incident/lib/python3.8/site-packages/paramiko/client.py:835: UserWarning: Unknown ssh ed25519 host key for 172.17.0.2: b'f8b3488df2a6cb911103b31345113e37'
warnings.warn(
```
Confirm the Problem

```
stdin, stdout, stderr = client.exec_command('ls')
stdin.close()
print(stdout.read().decode("utf-8"))
```
Fix the Problem

```
stdin, stdout, stderr = client.exec_command('rm ~/core')
stdin.close()
stdin, stdout, stderr = client.exec_command('ls')
print("−−−−−−")
    stdout.read().decode("utf-8"),
    "\n−−−−−−")

        
```

```
Post-Incident

- Duplicate notebook
Post-Incident

- Duplicate notebook
- Add notes in Markdown
Post-Incident

▶ Duplicate notebook
▶ Add notes in Markdown
▶ Export to HTML
▶ Attach HTML + Notebook to incident ticket
Final Thoughts

- Incident resolution is exploratory
Final Thoughts

- Incident resolution is exploratory
- This is not an auditing tool!
Final Thoughts

- Incident resolution is exploratory
- This is not an auditing tool!
- Share knowledge
Final Thoughts

- Incident resolution is exploratory
- This is not an auditing tool!
- Share knowledge
- Be prepared for retrospective