Python, Docker, Kubernetes, and beyond?

Peter Bábics | EuroPython 2018

July 25, 2018
Based in Prague
Small team of developers
Developing a trading platform and strategies
Using open source
Our trading platform and tooling

- Python 3.6 + React.js
- asyncio
- Redis & TimescaleDB for storage
- Integrating third party libraries using Cython
- Dozens of processes
- Messaging - Kafka / RabbitMQ
In the beginning
There was chaos
In the beginning

- Applications deployed on physical servers
- Managed by circus
- Packages installed in virtualenv
- Under a single user
Pros
• Simple implementation and deployment

Cons
• Package versioning hell
• No failover
A wild blue whale appeared
The promise of a brighter future

- Unified environment
- Simple deployment
- Simple migrations
- Faster Continuous Integration (CI)
- Atomic releases
Migration challenges

- Image storage - GitLab registry
- Image caching
- Dedicated building environment

```yaml
build_job:
  script:
    - docker build -t $CI_IMAGE_NAME .

build_job:
  script:
    - 'docker run -d -p 9000:9000 -v "/:/data" sleep 1y'
```

- CI pipeline design
- Cleaning up old images
Migration highlights

- Unified, stable environment
- Fast builds
- Isolated environments
- Faster CI pipeline
Cons of plain Docker

- Known bugs
- No failover
- `dockerd` is a single point of failure
Docker gotchas

- PID 1 pitfall
- User permissions within containers
Warmly welcomed features

- Failover when a server fails
- Configuration stored in namespaces
- Service discovery

```bash
my-service.my-namespace.svc.cluster.local
```

- Ingress controller
- Deployment history

```bash
$ kubectl rollout undo deployment my-app
```
Where we are now

- Migration to Kubernetes is in progress
- Environment is configured by namespace variables
- Deployments are described in Jinja2 templates

```jinja
{% set env_type, profile_name = PROFILE.split('_', 1) %}
{% set namespace = "my-ns-" + env_type %}
{% set data_directory_mount = '/data' %}

{% if profile_name == 'cthulu' %}
  - name: API_KEY
    valueFrom:
      secretKeyRef:
        name: secret
        key: api_key
  - name: SUBSCRIPTION_INSTRUMENT_FILTER
    value: 'False'
{% endif %}
{% endif %}
```
Notable features

- Probes
- Update strategies
Update strategies - Rolling update

Rolling update

Version 1

Deploy version 2

Version 2

Version 2 available & stable
Update strategies - Recreate

Version 1

Deploy version 2

Version 2

Version 2 available & stable

Recreate
Thank you

peter.babics@quantlane.com

github.com/qntln