Python for Cloud services and Infrastructure management

Agenda

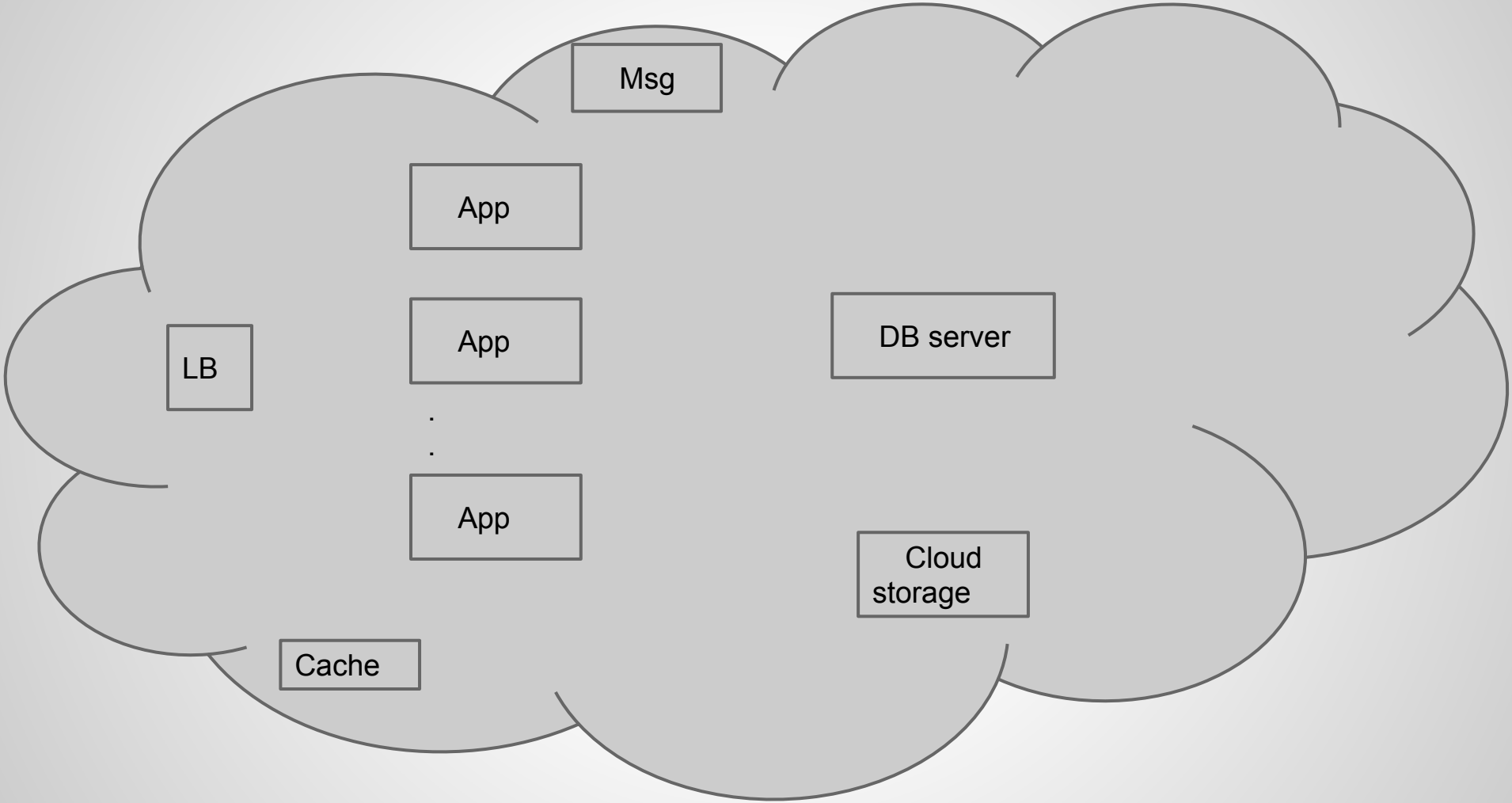
Python friendly services & Infrastructure practices

Things we use at Azoi

Deployment and configurations

Cloud for almost any web solution

- Load balancer
- App server
- Webserver
- DB server
- Internal Messaging
- Caching [Front end cache, backend cache]
- Distributed tasks
- Cloud storage [files, etc.]


Msg
App
App
LB
DB server
.
.
App
Cloud
storage
Cache

Deciding cloud infrastructure

Identify Usage

Identify performance requirement

Derive cloud instances requirement

Derive tools and technologies required

Cloud

Load balancer - Nginx/ ELB/ HAProxy

App server - uWsgi

Webserver - nginx

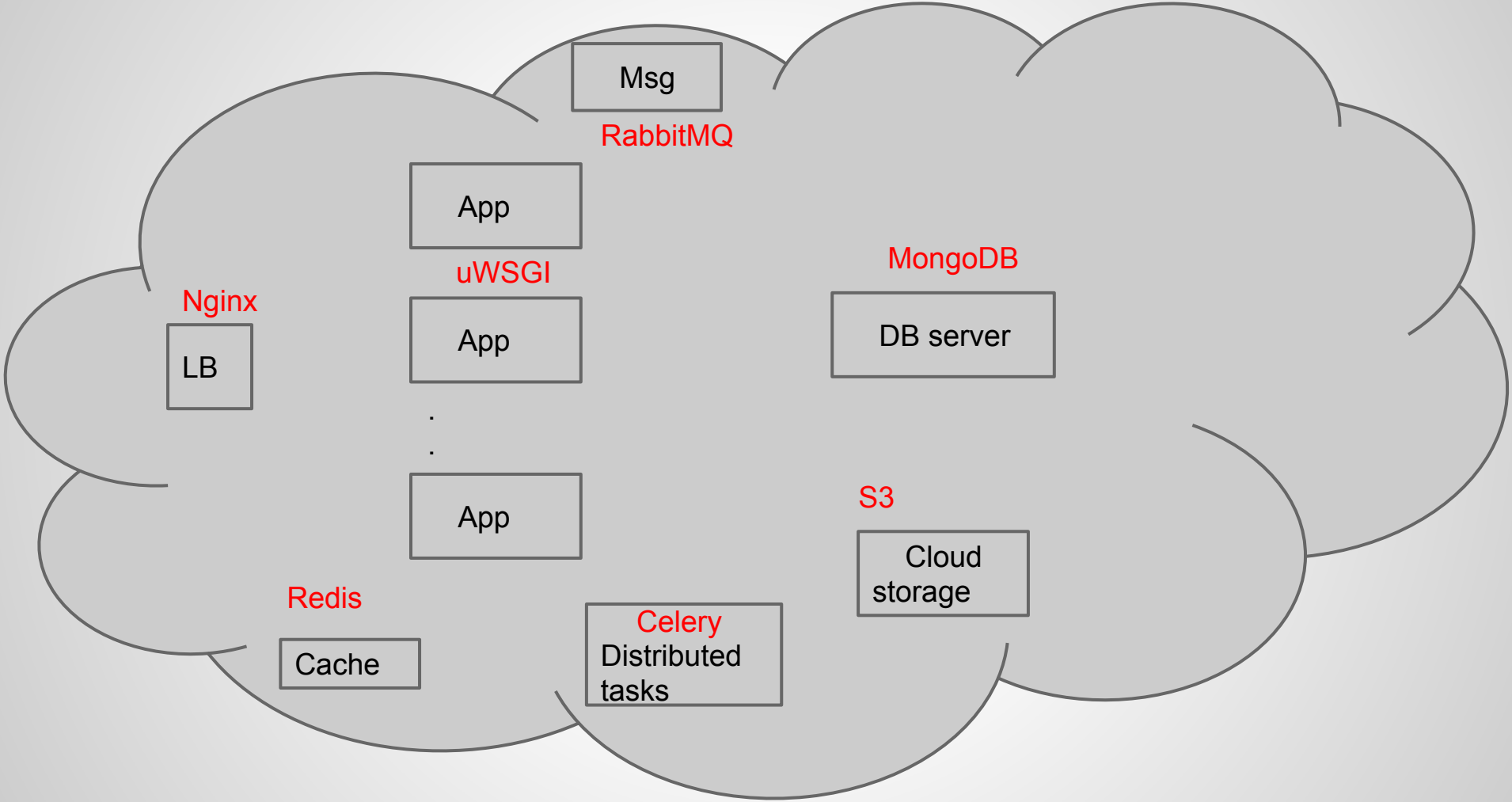
DB server - MongoDB

Internal Messaging - RabbitMQ

Caching - MemcacheD / Redis

Distributed tasks - Celery

Cloud storage [images, media, etc.] - S3


Msg
RabbitMQ
App
uWSGI
MongoDB
Nginx
App
DB server
LB
.
.
App
S3
Cloud
storage
Redis
Celery
Distributed
tasks
Cache

@ Azoi, we use

Load balancer - Nginx

App server - uWsgi

Webserver - nginx

DB server - NoSQL [MongoDB] and SQL [Postgres]

Internal Messaging - RabbitMQ

Caching - MemcacheD / Redis

Distributed tasks - Celery

Cloud storage [files, etc.] - S3

Monitoring - Nagios, Cloudwatch

Logging - Sentry [good match with Python based framework, we use **Django**]

#deployment and configurations

''' Also need to identify suitable tools for deployment and configurations. '''

@Azoi, we use **Fabric** for App deployment and admin tasks, advantage of writing deployment script in **python**.

We use **Jenkins** for integration.

Thank you