Docker Service Discovery

Tools for building distributed systems and service-oriented architectures in Docker

Andy Shinn

Principal at Glider Labs

http://gliderlabs.com/

andy@gliderlabs.com

Email the team at team@gliderlabs.com

Demo URL

github.com/andyshinn/sample-webapp

Why service discovery?

- Automatically configure applications
- Aides in high availability and fault tolerance
- Helps to scale horizontally

Service discovery components

- Directory (consistent key value stores)
- Registration / Deregistration (adding services to the directory and removing when unhealthy or gone)
- Lookup (discovering and using the registered services)
- Health checks and monitoring

Key / value stores

- ZooKeeper
- etcd
- Consul
- Doozer

Registration and deregistration

- progrium/registrator
- BlueDragonX/beacon
- jwilder/docker-register
- Custom scripts:

```
IP=$(ip addr show eth0 | awk '$1 == "inet" {gsub(/\/.*$/, "", $2); print $2}')
PORT=$(docker inspect -f '{{range $i, $e := .NetworkSettings.Ports }}{{$p := index $e 0}}{{$p.
HostPort}}{{end}}' container-name)

while netstat -lnt | grep -q ":$PORT"; do
  etcdctl set /application/service/container-name $IP:$PORT --ttl 300
  sleep 200
done
```

Discovery and lookup

- confd
- Sentinel
- envconsul (and envetcd)
- vulcand (router with in-process lookup)
- your own app in-process:

```
client = etcd.Client(host=os.environ.get('ETCD_HOST', '10.1.42.1'))
key = str(client.read('/app/services/redis')._children[0]['value'])
redis_url = 'redis://{0}/0'.format(key)
count = redis.StrictRedis.from_url(redis_url).incr("counter")
```

Health checks and monitoring

Active

- Consul health checks

Passive

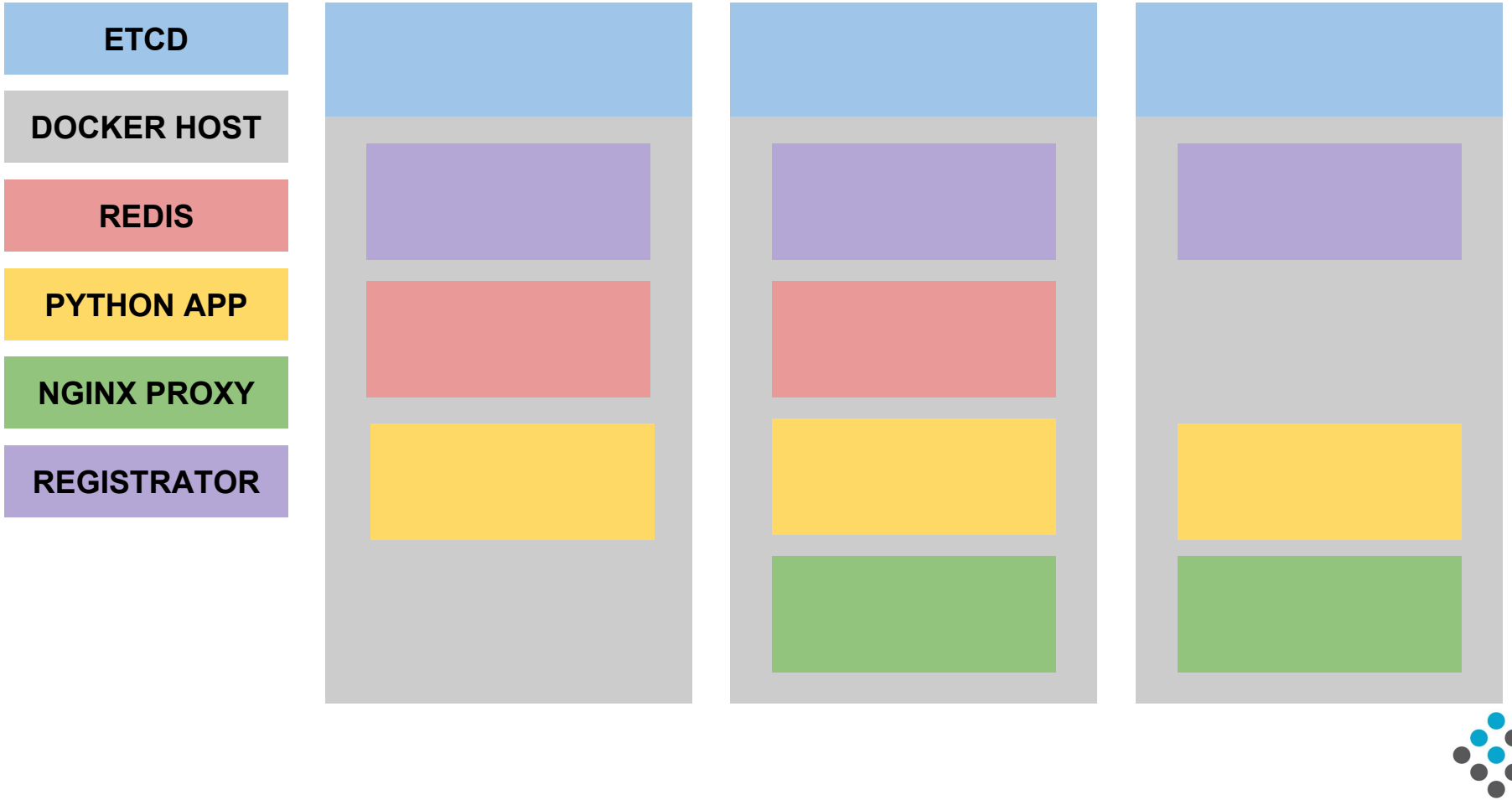
- Heartbeats with TTLs

Other tools

- https://github.com/jwilder/docker-gen
- https://github.com/airbnb/synapse
- https://github.com/skynetservices/skydns

Demo time!

- Registrator, Python app, Redis, nginx reverse proxy
- 2 services, Redis and the Python app
- Show Redis registration
- Show Python app using Redis via in-process discovery
- Show nginx proxying to Python app using co-process discovery


ETCD
DOCKER HOST
REDIS
PYTHON APP
NGINX PROXY
REGISTRATOR

References

- http://jasonwilder.com/blog/2014/02/04/service-discovery-in-the-cloud/
- http://progrium.com/blog/2014/07/29/understanding-modern-service-discovery-with-docker/
- http://progrium.com/blog/2014/09/10/automatic-docker-service-announcement-with-registrator/
- http://www.activestate.com/blog/2014/05/service-discovery-solutions
- https://github.com/jwilder/docker-gen