# Architecture 101

# CLOUD FOUNDRY™

# Architecture 101

# Agenda

1. Let's talk about Deployment

2. What is Cloud Foundry?

3. What can be deployed to PaaS?

4. What happens when you run cf push? - Back-end overview

5. How are requests routed? - Front-end overview

6. Who deploys the deployer?

7. Q&A

# Let's talk about Deployment

## Current Dev Process

- Write provisioning scripts (CloudFormation/Terraform/Shell)

- Write configuration code (Chef/Puppet/Ansible/Shell)

- Establish some deployment system (Chef/Shipper/Capistrano/Shell)

- Hook Deployment Script to CI

**Time**: weeks to months

## Ideal Dev Process

- No provisioning

- No configuration

- Simple deploys

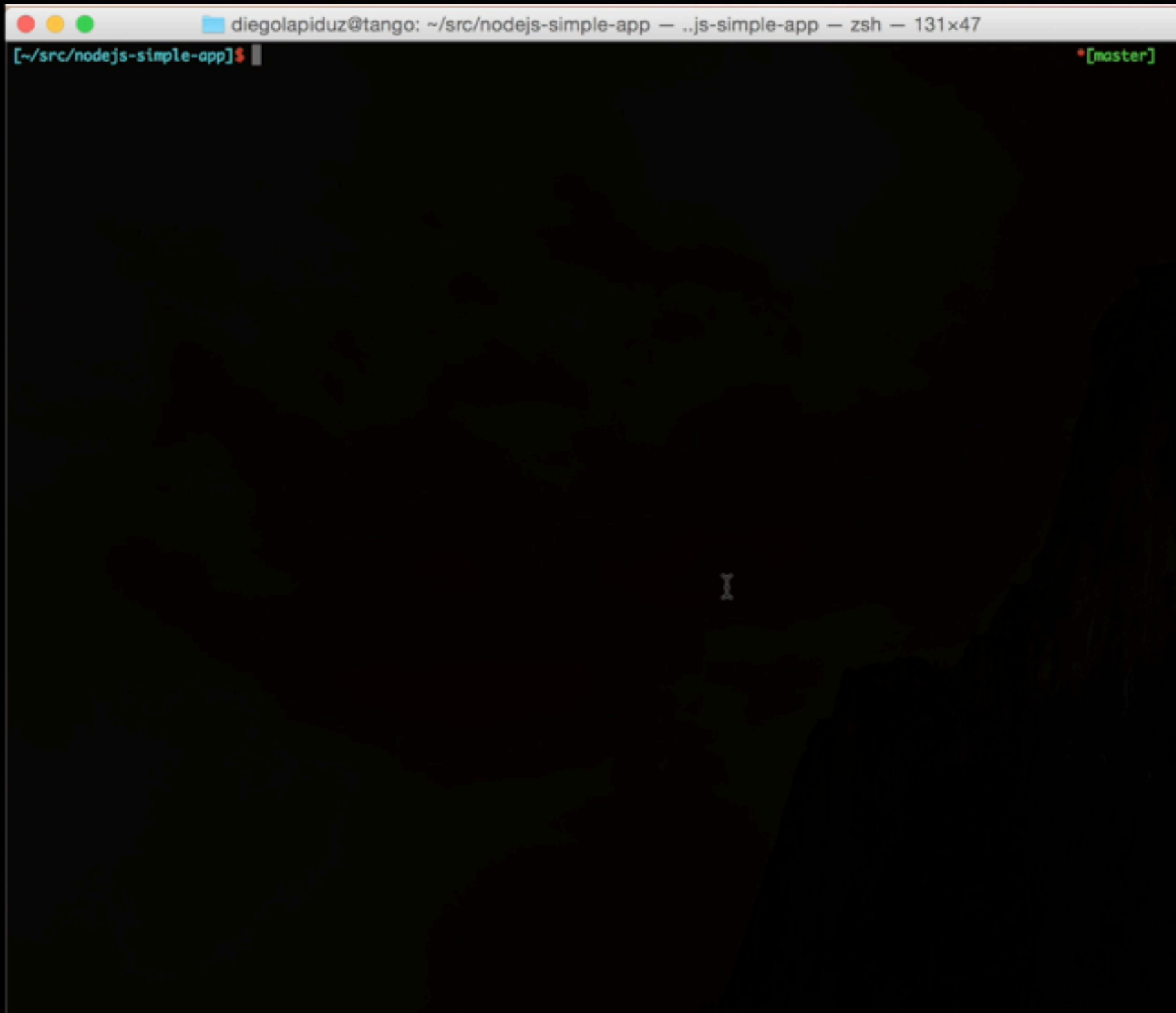- Simple CI Hook

**Time**: minutes to a day

# Proposed Solution: PaaS

## Pilot: Cloud Foundry

# What is Cloud Foundry?

- Self-hosted

- Open Source

- Multi-tenant

- Platform as a Service

```
[~/src/nodejs-simple-app]$                                                                                    *[master]
```

# What can be deployed to the PaaS?

12 factor apps (stateless)

# What can be deployed to the PaaS?
# 12 factor app

- **One codebase tracked in revision control**

- **Declare dependencies**

- **Config in the environment**

- Backing services = attached resources

- Separate build and run stages

- Execute the app as one or more stateless processes

- Export services via port binding

- Scale out via processes

- Fast startup / graceful shutdown

- Dev = Stage = Prod

- Logs as event streams

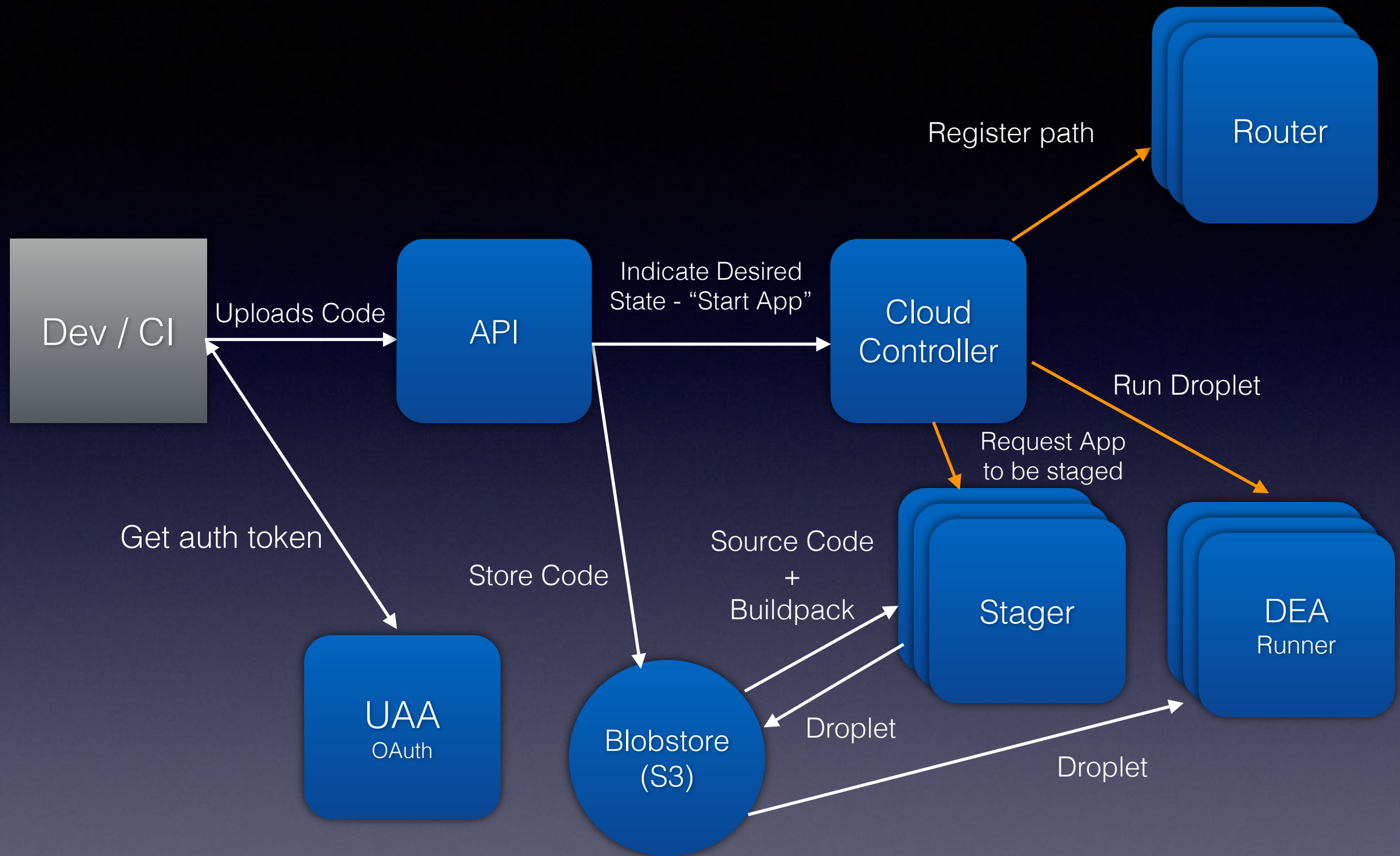- Run admin/management tasks as one-off processes

# What can be deployed to the PaaS?
# CF Default Buildpacks

- Ruby
- Node.js
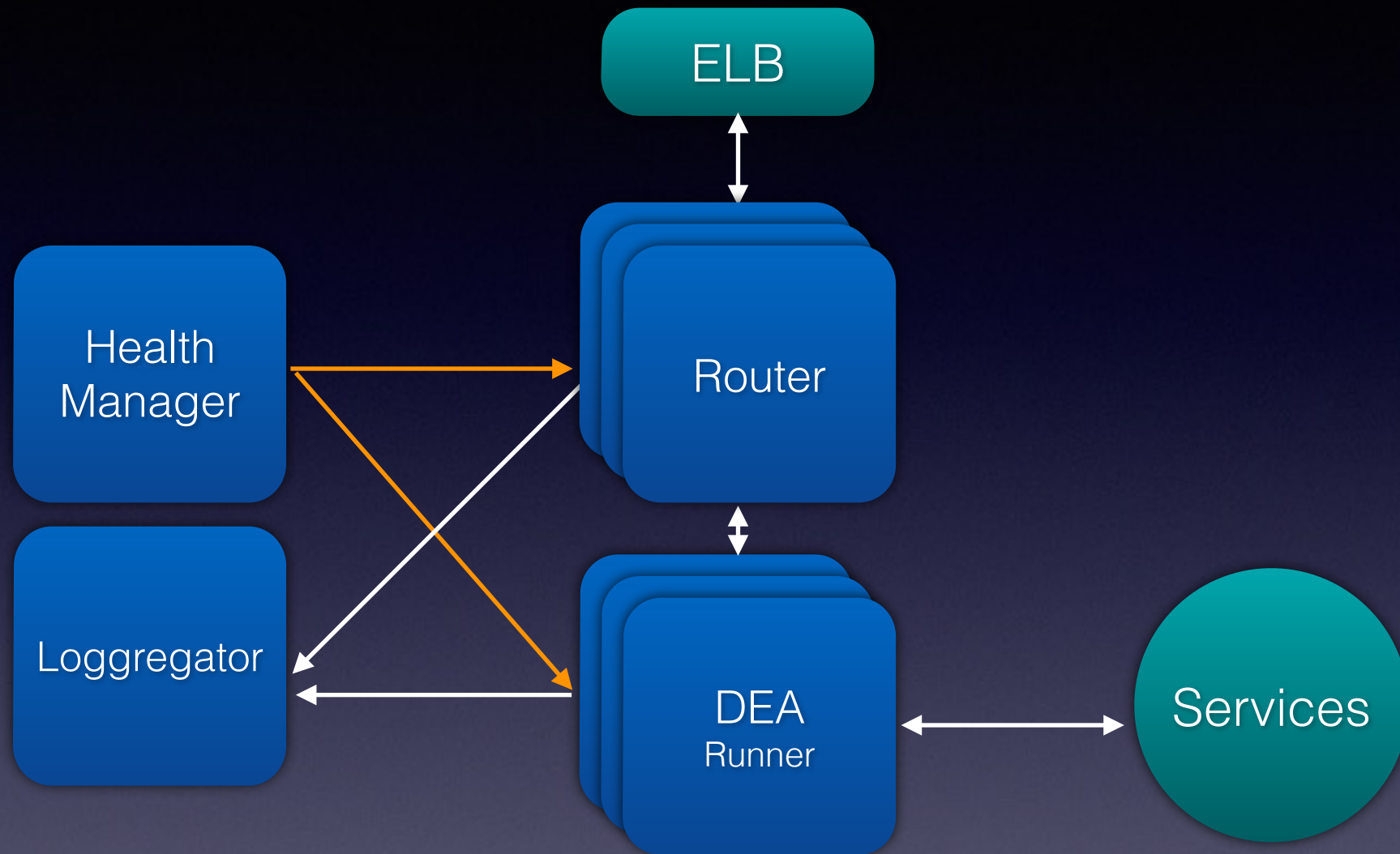- Java
- Go
- Python
- PHP

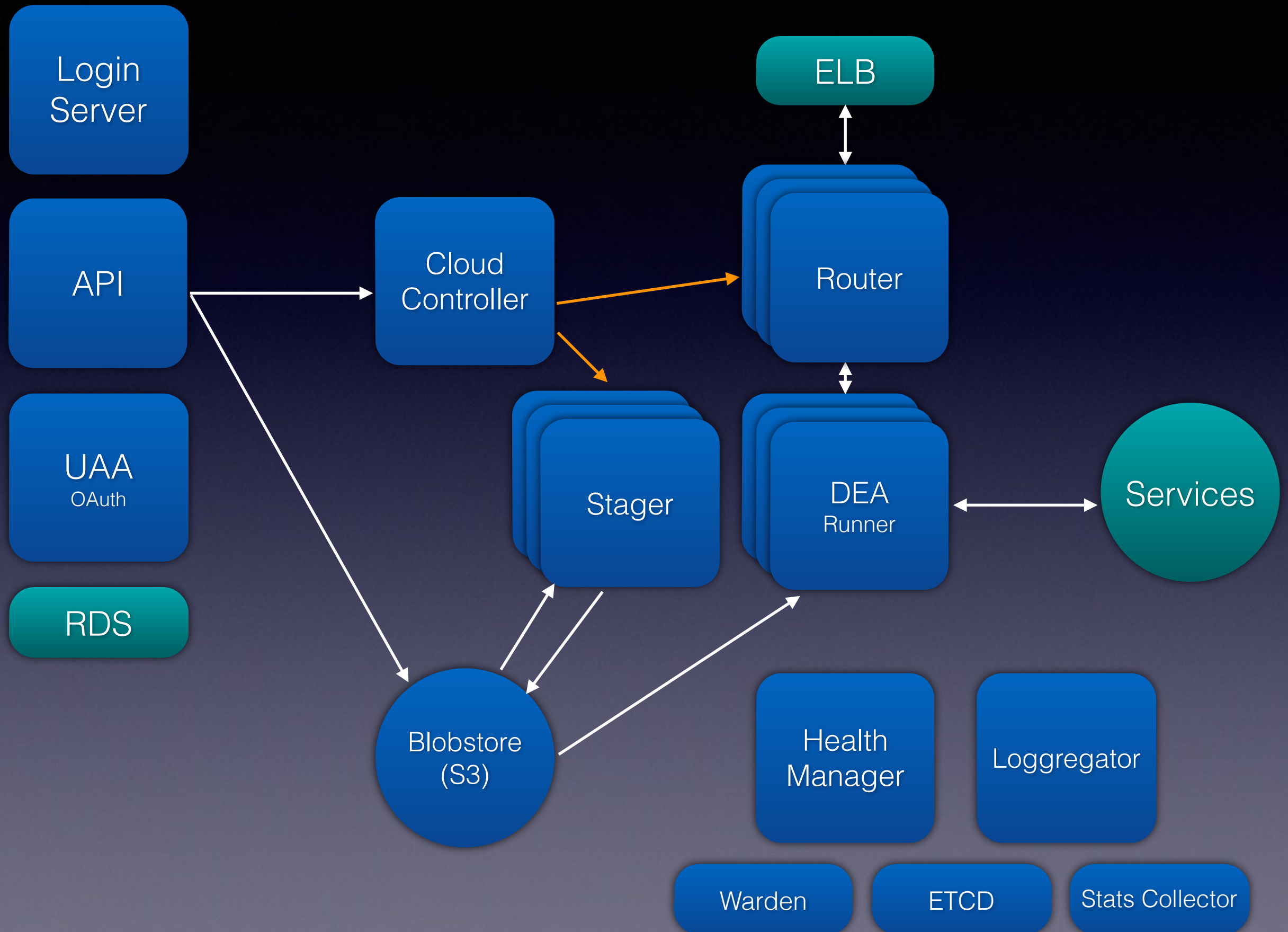- Custom ones also supported

What happens when you run `cf push`

# Request Routing

# From a developer perspective

- cf push myusa —no-start # Create application

- cf create-service postgresql default myusa_db # Setup db

- cf bind-service myusa myusa_db # bind the db

- cf push -c "bundle exec rake db:migrate" # migrate

- cf push -c "nil" # run the default start command

- cf scale -i 4 -m 1GB # Run 4 instances with 1GB of RAM each

- cf map-route myusa myusa.18f.us

# Considerations

- Similar architecture applies to every PaaS

- The system doesn't change when a new application is added

- It can be scaled up and out

- Billing, Account Management, Monitoring and Isolation not mentioned but are part of the CF back-end

- Services still need to be deployed and managed

Who deploys the deployer?

BOSH
The outer shell for cloud software

- Based on a VMWare project to deploy **stateful** distributed systems

- Spins vms

- Configures services

- Keeps an auditing log

- Rebuild dead vms

- Supports many targets (many "clouds")

- Supports many IaaS (AWS, OpenStack, vSphere)

Q&A

# Resources

- http://docs.cloudfoundry.org/concepts/architecture/

- http://12factor.net/

- http://bosh.io/docs