The Bits-Service Myth Buster
Agenda

1. What is the Bits-Service (and what it is not) ?
2. Why a Bits-Service ?
3. What did we do ?
4. Challenges we faced
5. Lessons we’ve learned
6. How does it affect you ? How does it help you ?
7. What will be next ?
8. Q&A
What is the Bits-Service?

- A true community incubation project (IBM/Pivotal) scoped around bits
- What are bits? (or BLOBs)
  - Application artifacts (package)
  - Compiled app (droplet)
  - Buildpacks
- Where does CF use bits?
  - Example: Push an App
Example: cf push --no-start
PUT /v2/apps/:app-uid/bits <zip-file>

assemble package

201

store file

201

store package

CC

Blobstore

cf
Where does Cloud Foundry store bits?

• Different object stores
  • Local disk
  • Amazon S3
  • Openstack Swift
  • etc.
So ... what is the Bits-Service?

- Cloud Foundry specific abstraction for storage backends
  - Clearly defined API
  - Encapsulates bits-related functionalities, e.g. resource matching
  - An additional service in a cloud foundry deployment

- What it is not
  - An Object Store
  - A generic Storage API
Why do we need a Bits-Service?

• CF Cloud Controller (CC) became a bit monolithic

• Idea: Refactor CC functionalities into separate µServices
  • independently scalable
  • clean API
  • better maintainability and operations
So ... what did we do?

- **Cookbook:**
  - Rip-out a resource (e.g. droplets) from the cloud controller
  - Make it work for all supported REST verbs
  - Make it accessible from the outside (nginx)
  - Make the cloud controller use it
  - Repeat until all resources were done

- **Add Breadth:**
  - Local filesystem first, then S3, ...

- **Add Depth:**
  - v3 API, Diego, ...
How will the Bits-Service change interaction?

Yesterday

CF CLI → CC → Blobstore

Today (temporary)

CF CLI → CC → Bits-Service → Blobstore

Tomorrow

CF CLI → CC → Bits-Service → Blobstore
Challenges we faced

• Often we had Fog 😊 (missing abstraction)
• Rather implicit and hidden bits-related flows in CC
• We were not alone: CC development continued
• Avoid breaking changes as far as possible because many involved teams:
  • CC
  • CLI
  • Diego
What did we learn?

• From working together
  • Company cultures are different 😊
  • High degree of remote pairing can work!
  • (PM) Synchronization with 9h time difference is tough

• CC can be split into microservices ... but it’s hard
  • Each resource works differently
  • Functionality is distributed across many places (v2 API is organically grown)
    • v3 would have been much better because it has a much cleaner REST model
How does it affect you? How does it help you?

• If you are a **cloud foundry operator**
  • Independent scalability of CC and Bits-Service operations
  • More VMs that you need to monitor etc.
  • And use anything else than S3 or Swift to store your bits: We need to talk!

• If you are an **application developer**
  • Cloud Controller getting more responsive
  • More efficient bits handling (faster push times, uploads etc.)

• If you are a **cloud foundry developer**
  • Eventually a clean API to code all of your bits operations
Future Plans

• Release bits-service and make it default in cf-release

• Add more operational stats

• Become independent of fog

• Implement more efficient resource matching

• Re-implement Bits-Service in Golang?

• Introduce a “backup blobstore” API?
Any Questions
Useful links

• CI: https://flintstone.ci.cf-app.com/?groups=all
• Tracker: https://www.pivotaltracker.com/n/projects/1406862
• Git: https://github.com/cloudfoundry-incubator/bits-service

Copyrights

• All pictures used in this presentation where obtained from pixabay.com and are under the CC0 public domain license.