Demystifying CATs
Michael Xu and Tim Hausler, Software Engineers, *Pivotal*
Motivation

• Short Story
• You want to contribute
• Maybe you have!
• You understand that you have to test
• This is the primary audience
• You want to learn a bit more about how CF releases are validated

---

CF Acceptance Tests (CATs)

This suite exercises a Cloud Foundry deployment using the `cf` CLI and `curl`. It is scoped to testing user-facing, end-to-end features.

For example, one test pushes an app with `cf push`, hits an endpoint on the app with `curl` that causes it to crash, and asserts that we see a crash event in `cf events`.

Tests that won’t be introduced here include things like basic CRUD of an object in the Cloud Controller. Such tests belong with the component they relate to.

These tests are not intended for use against production systems. They're meant for acceptance environments used by people developing Cloud Foundry’s releases. While these tests attempt to clean up after themselves, there's no guarantee that they won't change the state of your system in an undesirable way. For lightweight system tests that are safe to run against a production environment, please use the CF Smoke Tests.

NOTE: Because we want to parallelize execution, tests should be written to be executable independently. Tests should not depend on state in other tests, and should not modify the CF state in such a way as to impact other tests.

1. Test Setup
   i. Install Required Dependencies
   ii. Test Configuration
CATs

= 

(cf-acceptance-tests)
Who this talk is for

- You want to contribute
- You want to learn a bit more about how CF releases are validated
Why are we talking about CATs?

Iteration in CATs is painfully slow

One test  
One suite  
Many suites
Talk Outline

- What is CATs?
- How do I set it up?
- How do I troubleshoot it?
- Demo
What is CATs?
CATs is intended to demonstrate that the platform is capable of doing the things it is supposed to do.
What is CATs?

• Set of tests that run to validate the correctness of Cloud Foundry code
• Written in Go, maintained by #release-integration
• Intended to validate releases of Cloud Foundry
  • Acceptance tests
  • Use CLI to perform basic workflows
What does CATs NOT do?

- Not an SLI for your platform
  - Runs an inhuman number of app pushes
- Not a smoke test for your platform
  - Only run against development environments
- Validates software, not deployments
How do you set up and run CATs?
integration_config is the source of truth
integration_config is the source of truth

```json
{
  "api": "api.bosh-lite.com",
  "apps_domain": "bosh-lite.com",
  "admin_user": "admin",
  "admin_password": "meowmeow",
  "skip_ssl_validation": true,
  ...
}
```

CF deployment info
integration_config is the source of truth

```json
{
    ...
    "include_apps": true,
    "include_backend_compatibility": true,
    "include_container_networking": false,
    "include_detect": true,
    "include_security_groups": true,
    "include_services": false,
    "include_ssh": true,
    "include_tasks": true,
    ...
}
```
integration_config is the source of truth

```json
{
  ...
  "use_existing_user": true,
  "existing_user": "basel-dev",
  "use_existing_organization": true,
  "use_existing_organization": "cf-summit",
  "cf_push_timeout": "90",
  "long_curl_timeout": 30,
}
```

CATs runtime settings
Which ones do I run?
<table>
<thead>
<tr>
<th>DEFAULT</th>
<th>GA FEATURES</th>
<th>SPECIAL DEPLOYMENT</th>
</tr>
</thead>
<tbody>
<tr>
<td>apps</td>
<td>backend_compatibility</td>
<td>container_networking</td>
</tr>
<tr>
<td>detect</td>
<td>docker</td>
<td>credhub</td>
</tr>
<tr>
<td>persistent_app</td>
<td>internet_dependent</td>
<td>isolation_segments</td>
</tr>
<tr>
<td>routing</td>
<td>private_docker_registry</td>
<td>privileged_container_support</td>
</tr>
<tr>
<td></td>
<td>route_services</td>
<td>routing_isolation_segments</td>
</tr>
<tr>
<td></td>
<td>security_groups</td>
<td></td>
</tr>
<tr>
<td></td>
<td>services</td>
<td>zipkin</td>
</tr>
<tr>
<td></td>
<td>ssh</td>
<td></td>
</tr>
<tr>
<td></td>
<td>sso</td>
<td></td>
</tr>
<tr>
<td></td>
<td>tasks</td>
<td></td>
</tr>
<tr>
<td></td>
<td>v3</td>
<td></td>
</tr>
<tr>
<td>SUITES TO RUN</td>
<td>SPECIAL CASES</td>
<td></td>
</tr>
<tr>
<td>-----------------------</td>
<td>------------------------------------</td>
<td></td>
</tr>
<tr>
<td>apps</td>
<td>container_networking</td>
<td></td>
</tr>
<tr>
<td>backend_compatibility</td>
<td>credhub</td>
<td></td>
</tr>
<tr>
<td>detect</td>
<td>isolation_segments</td>
<td></td>
</tr>
<tr>
<td>docker</td>
<td>private.docker_registry</td>
<td></td>
</tr>
<tr>
<td>internet_dependent</td>
<td>privileged_container_support</td>
<td></td>
</tr>
<tr>
<td>persistent_app</td>
<td>routing_isolation_segments</td>
<td></td>
</tr>
<tr>
<td>route_services</td>
<td>zipkin</td>
<td></td>
</tr>
<tr>
<td>routing</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Let’s run CATs
But... CATs take a while
But... CATs take a while

Lots of tests

- Lots of CF pushes
- Lots of strain on the system
Quick iteration and feedback

- Focusing a test or a suite
- More nodes, faster runs!
  - `bin/test --nodes=8`
  - Vanilla CF handles 6 - 8 nodes
How do we troubleshoot errors?
Troubleshooting errors

- Inconsistent runs:
  - Decrease nodes
  - Specify longer timeouts:
    - curl_timeout
    - cf_push_timeout
- Bottlenecks:
  - Diego cells
  - API VMs
Troubleshooting errors

- CATs print out every CLI command and its output
CATs are not just local
How do we run CATs at Pivotal?

- Focusing a suite locally
- Large subset of suites locally
- Team concourse pipelines
- Release-Integration master pipeline
CATs Demo

- Demonstrate testing cycle for a PR that requires a CATs change
- Thanks to Alex Blease, Jen Spinney, and Sam Gudunarat for the contribution
A bit later...