

 Engineering

Fast, safe, and reliable

The future of configuration



Qui Nguyen
Tech Lead

October 2019



How can we best **empower** developers?

Enable them to make changes as quickly as possible, while making it easy to stay safe.

Configuration

Values used internally by applications that you want to change over time.

Decouple the application from resource names like S3 buckets or logs.

Change the number of search results.

Tune the number of retries and timeouts for a request.

Update percentages or whitelists for an experiment.

Constants

config.py

```
TIMEOUT_SEC = 10
```

client.py

```
from config import TIMEOUT_SEC  
do_request(TIMEOUT_SEC)
```

+ Reusable

- Need to update code

- Can't vary across environments

Environment variables

```
client.py
-----
timeout = os.environ.get(
    'TIMEOUT_SEC'
)
do_request(timeout)
```

```
$ TIMEOUT_SEC=10 python client.py
```

- + Reusable
- + Can vary across environments
- Need to restart process

Read from files

```
client.yaml
```

```
-----
```

```
timeout: 10
```

```
client.py
```

```
-----
```

```
load_file_if_changed('client.yaml')
```

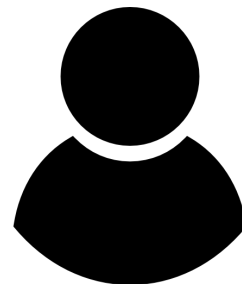
```
timeout = get_config('timeout')
```

```
do_request(timeout)
```

- + Reusable
- + Can vary across environments
- + Get new value if file changes

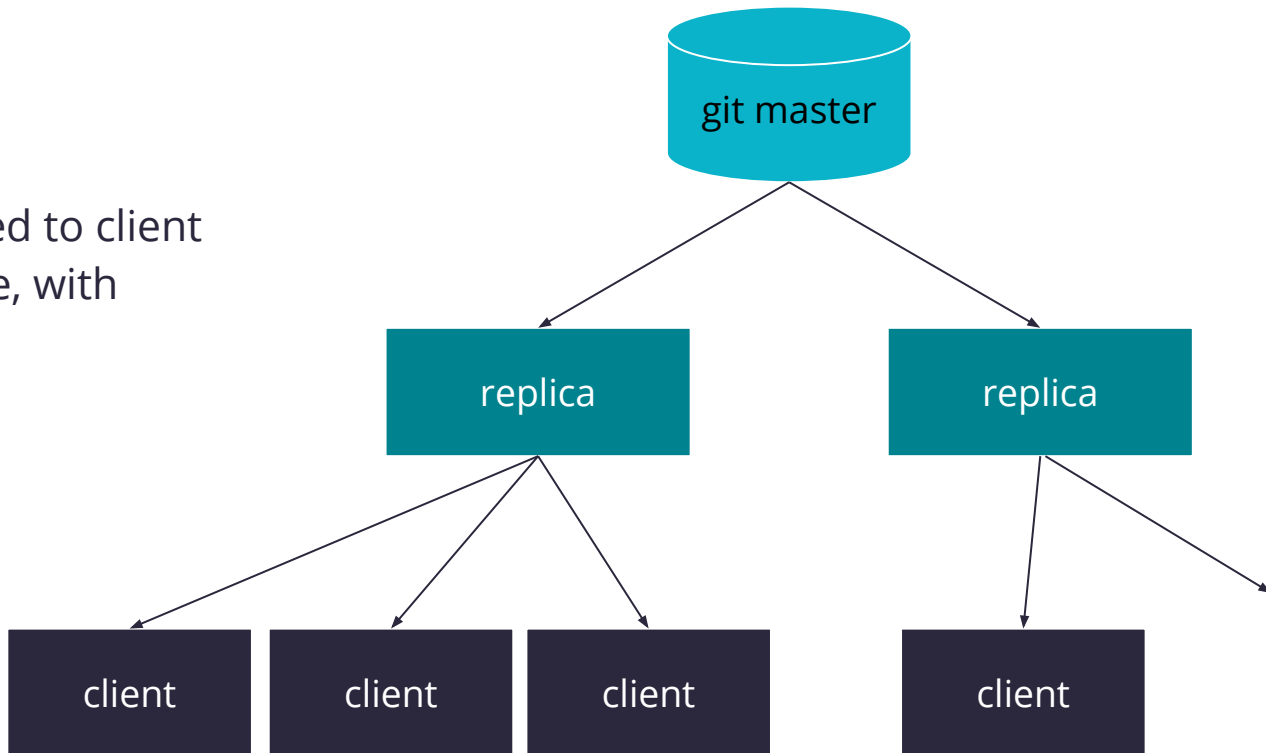
srv-configs

```
$ git push origin i/qui/timeout  
...  
remote: merged to master
```



srv-configs

- Files from git copied to client hosts every minute, with rsync



srv-configs

Varying files by environment

- Configs stored in git for all environments
- Clients materialize a view based on their location ("hiera-merging")

```
common/  
  foo.yaml  
ecosystem/  
  prod/  
    foo.yaml  
  dev/  
region/  
  ...
```

srv-configs

server1

ecosystem: prod
-> region: uswest1

common/foo.yaml

```
log_name: foo_errors  
max_timeout_secs: 5
```



ecosystem/prod/foo.yaml

```
max_timeout_secs: 1
```

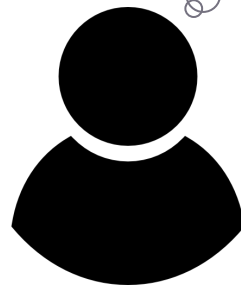


foo.yaml

```
log_name: foo_errors  
max_timeout_secs: 1
```

srv-configs

```
$ git push origin i/qui/timeout  
...  
remote: merged to master
```



I need to
decrease that
default timeout.

See changes in **2 minutes.**

Push a branch and your service will start using those values as soon as the files update, with no restart required.

Configuration brings down the site

experiments:

 buttons:

 active: True

 cohorts: [...]

```
cohort = get_cohort('buttons')
```

```
# modern_blue, layout_blue
```

```
use_blue_buttons = 'blue' in cohort
```



Make a Reservation

 Fri, Oct 18

 7:00 pm

 2 people

5:00 pm

9:00 pm

9:15 pm

9:30 pm

[See more availability](#)

 q-portland.com

 (503) 850-8915

Configuration brings down the site

```
experiments:  
  buttons:  
    active: False  
    cohorts: [...]
```

```
-----  
cohort = get_cohort('buttons')  
# modern_blue, layout_blue  
use_blue_buttons = 'blue' in cohort
```

**TypeError: argument of
type 'NoneType' is not
iterable**



The dog ate our website.

Ok, ok, ok we know that's not an excuse. We'll have it back up-and-running for you shortly.

Incidents

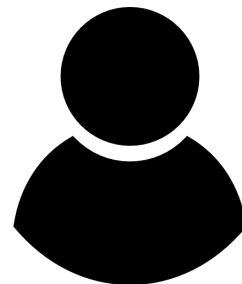
code changes

srv-configs



srv-configs deploy

```
$ git push origin i/qui/timeout  
...  
remote: deploy started
```



Make it **easy** to do the **right** thing.

Automatically stage changes, from stage to canary to production.

Roll back in one step.

srv-configs APP Today at 10:58 AM

public deploy of i/qui/demo
(540f76deb3b57cb0d5bcc36ba638ecaa46e46461) | state: deploying |
stage: 100%, dev: 100%, canary: 0%, prod: 0%

Rollback Now

2 replies

srv-configs APP < 1 minute ago

public deploy of i/qui/demo
(540f76deb3b57cb0d5bcc36ba638ecaa46e46461)

Deploying over next 180 seconds

State machine: deploying

Progress: stage: 100%, dev: 100%, canary: 0%, prod: 0%

Last operator action: None

srv-configs APP < 1 minute ago

Author: @qui

Files affected: ['common/compute_infra_test_service.yaml']

 Reply...



Also send to # test ?


Send

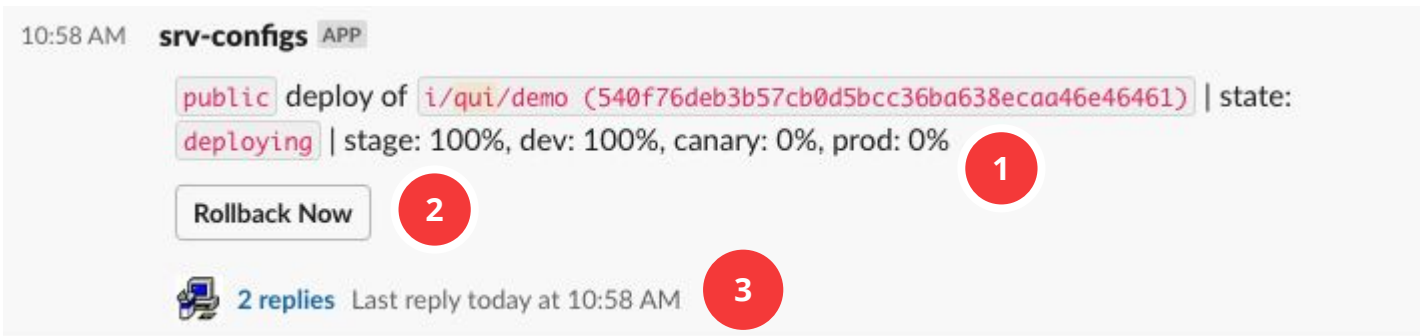
Interactive interface

10:58 AM **srv-configs** APP

`public` deploy of `i/qui/demo (540f76deb3b57cb0d5bcc36ba638ecaa46e46461)` | state:
`deploying` | stage: 100%, dev: 100%, canary: 0%, prod: 0%

[Rollback Now](#)

 [2 replies](#) Last reply today at 10:58 AM



1

Progress of deploy updated as it goes through stages.

2

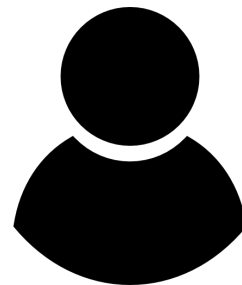
Prominent button to encourage rolling back in case of issues.

3

More details for authors in thread.

```
--- a/foo.yaml
+++ b/foo.yaml
@@ -1,2 +1,2 @@
  log_name: foo_errors
- max_timeout_secs: 5
+ max_timeout_secs: 2
```

```
$ git push origin i/qui/timeout
...
remote: deploy started
```



Implementing staged deployments

When an integration branch (i/) is pushed, git hooks:

- Generate a deployment version of each affected file
 - Update control file with rollout information for that branch
-

Implementing staged deployments

Generated by git hooks and synced to clients

main/foo.yaml

```
log_name: foo_errors
max_timeout_secs: 5
```

deploy/foo.yaml

```
log_name: foo_errors
max_timeout_secs: 2
```

branches.json

```
"i/qui/timeout": {
  "stage": 1.0,
  "canary": 0.0,
  "production": 0.0,
}, ...
```

New clientlib

Provides an interface between data (files) and logic

Before

```
load_file('my_service.yaml')  
watcher('my_service.yaml').reload_if_changed()
```

After

```
load_namespace('my_service')  
get_watcher('my_service').reload_if_changed()
```



**Consider
deploy files and
branches.json!**

Rolling back

When the rollback button is clicked:

- Delete deploy files
- Remove branch from branches.json

srv-configs APP Today at 11:25 AM
public deploy of i/qui/demo
(08acbb5bd5bd019ade5968110e0edeb35f2760a0) | state: cancelled |
Cancelled

3 replies

srv-configs APP < 1 minute ago
public deploy of i/qui/demo
(08acbb5bd5bd019ade5968110e0edeb35f2760a0)

Deploy cancelled.

State machine: cancelled

Progress: Cancelled

Last operator action: rollback

srv-configs APP < 1 minute ago
Author: @qui
Files affected: ['common/compute_infra_test_service.yaml']

srv-configs APP < 1 minute ago
qui pressed rollback

 Reply...



srv-configs APP Today at 10:58 AM

public deploy of i/qui/demo
(540f76deb3b57cb0d5bcc36ba638ecaa46e46461) | state: deploying |
stage: 100%, dev: 100%, canary: 0%, prod: 0%

Rollback Now

2 replies

srv-configs APP < 1 minute ago

public deploy of i/qui/demo
(540f76deb3b57cb0d5bcc36ba638ecaa46e46461)

Deploying over next 180 seconds

State machine: deploying

Progress: stage: 100%, dev: 100%, canary: 0%, prod: 0%

Last operator action: None

srv-configs APP < 1 minute ago

Author: @qui

Files affected: ['common/compute_infra_test_service.yaml']

 Reply...



Also send to # test ?

Send

Rolling back later

Tell the user how to rollback after the deploy is done, too.

srv-configs APP Today at 10:58 AM
public deploy of i/qui/demo
(540f76deb3b57cb0d5bcc36ba638ecaa46e46461) | state: complete |
stage: 100%, dev: 100%, canary: 100%, prod: 100%

3 replies

srv-configs APP 3 minutes ago
public deploy of i/qui/demo
(540f76deb3b57cb0d5bcc36ba638ecaa46e46461)

Merged into master.

State machine: complete

Progress: stage: 100%, dev: 100%, canary: 100%, prod: 100%

Last operator action: None

srv-configs APP 3 minutes ago
Author: @qui
Files affected: ['common/compute_infra_test_service.yaml']

srv-configs APP < 1 minute ago
Merged into master. Rollback Command (from your devbox):
srv-configs rollback
540f76deb3b57cb0d5bcc36ba638ecaa46e46461

🔗 Reply...

Also send to #test ?

Send

Automatic rollbacks

Can monitor SLOs or other metrics in deploy process

- Notify users and default to rolling back

12 replies

PaaSTA APP 22 days ago

Deploy of `9ddf07cd` of `bot_signal_platform` to `prod.canary`:

Finished rolling back to `3102dd56` in prod.canary

State machine: `abandon`

Progress: 100%

Last operator action: None

1 of 9 SLOs are failing:

`bot_signal_platform: get_guv_classification_v1 p99 > 50`

(duration=1h, percent_of_duration=95, paasta_cluster=norcal-prod)

PaaSTA APP 22 days ago

^ @pradeepn

PaaSTA APP 22 days ago

(Jenkins Job)

PaaSTA APP 22 days ago

Marked `9ddf07cd` for prod.canary

^ @pradeepn

PaaSTA APP 22 days ago

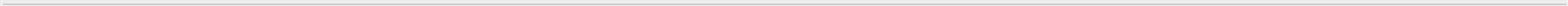
SLO started failing: `bot_signal_platform: get_guv_classification_v1 p99 > 50 (duration=1h, percent_of_duration=95, paasta_cluster=norcal-prod)`

PaaSTA APP 22 days ago

Will automatically roll back in 30 seconds, (at 19:15:05)! Click "Disable auto rollbacks 🙏" to cancel this!

This is cool...

but why files?



We should switch to a datastore!

Clients can watch for changes, instead of rsyncing everything

More natural API for automated changes

- Updating deploy progress
 - Updating config from other systems
-

...or not. Files have a lot of features.

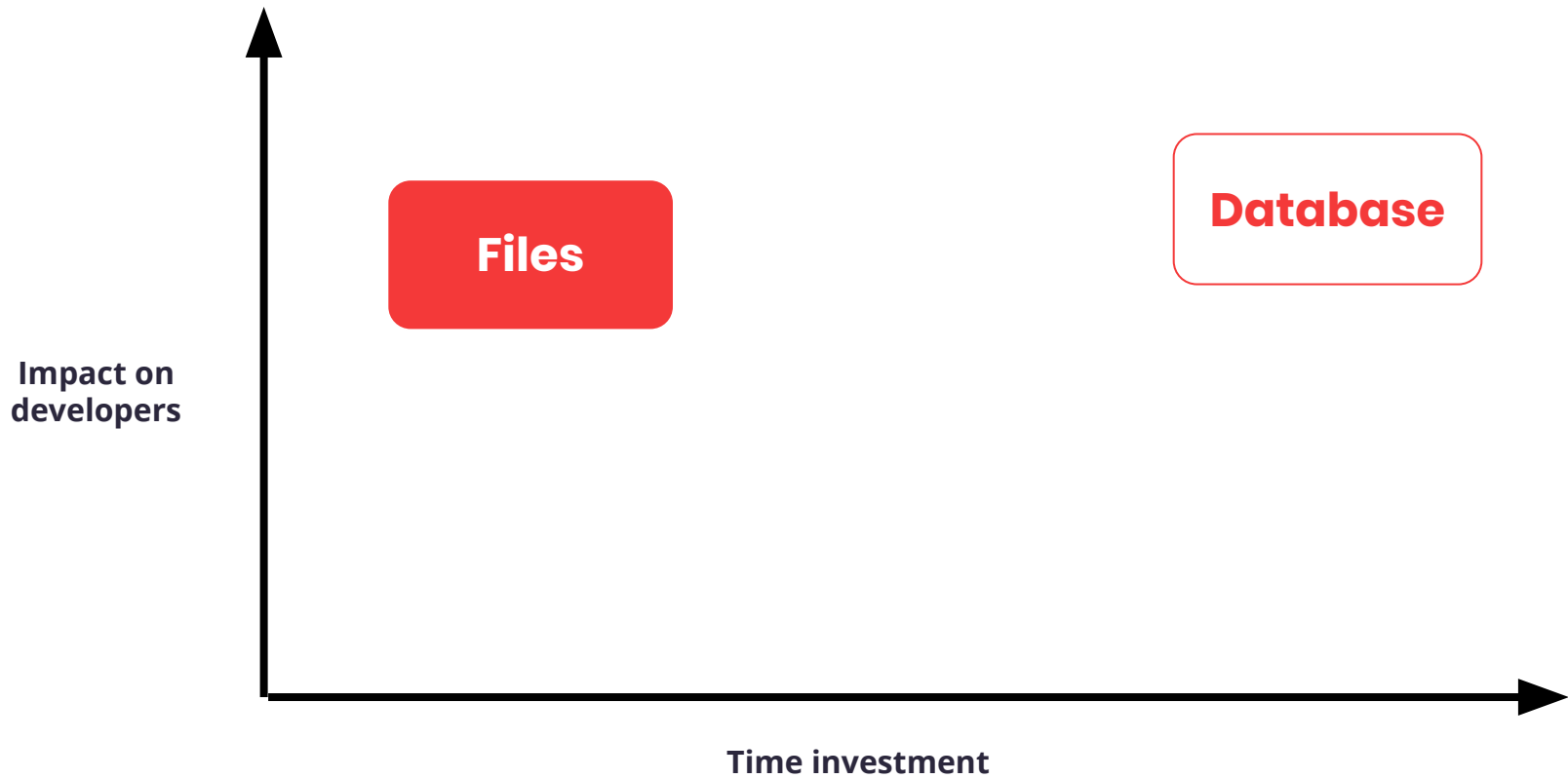
Can take advantage of existing workflows for code files

- Editing with text editors
- Searching, reviewing

Version control (git)

- Viewing history
- Reverting changesets

Easy to scale and handle master failures





Why don't we have both?

**Configs are as
powerful as
code.**

How can we best **empower** developers?

Dynamically loaded configuration from files allows developers to spend less time updating and deploying code.

Interactive deploy interface provides rich feedback and guides developers on best practices to keep the site safe.
