

Pragmatic Automation



Max Luebbe, Google
SRECon Americas 2019

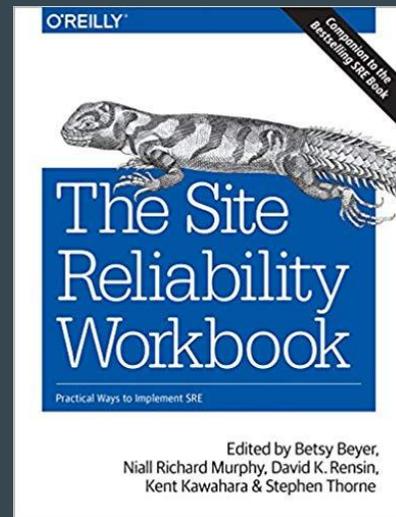
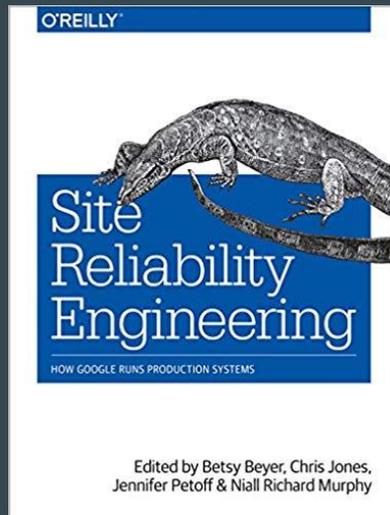
Max Luebbe

Site Reliability Engineer at Google since 2009
(Search Indexing -> Storage -> Cloud)

Google Chicago since 2012

College Graduate since 2018

@mluebbe



"Automate yourself out of a job"

What if we're asked to automate something we have no idea how to do?

Backstory

The Original Four!

`us-central1` (Iowa)

`us-east1` (South Carolina)

`eu-west1` (Belgium)

`asia-east1` (Taiwan)



Google Cloud Platform

```
$./build_new_region.sh
```

The End!

Thanks for coming to my talk!

Two Challenges

- 1) Define and automate the overall process to build a new Google Cloud Location
- 2) Define and automate the processes to deploy and configure every system required by a new Google Cloud Location

General Problem

A Common Failure Mode



https://en.wikipedia.org/wiki/List_of_waterfalls_that_empty_into_an_ocean

Another Common Failure Mode

EFFORT

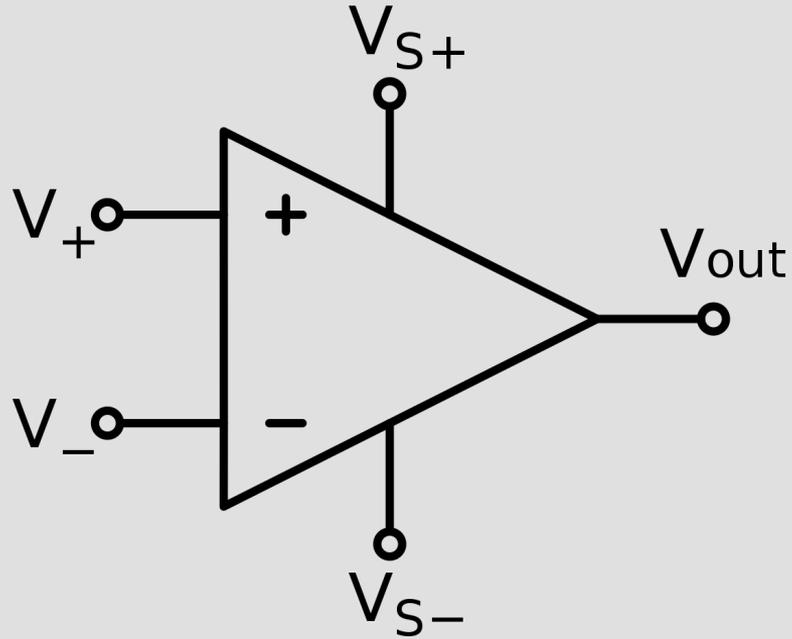
TO

AUTOMATE

Payoff.

Strategy

Effective Automation

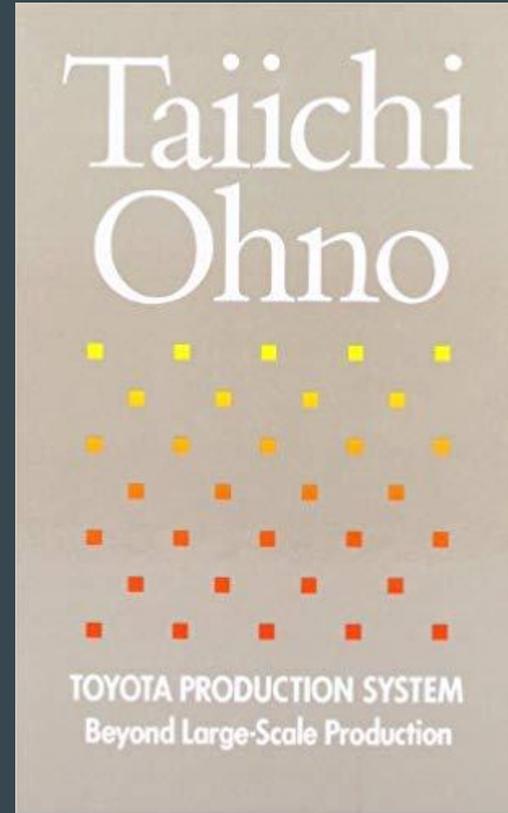


"Automation"

Automation with a human touch.

Human judgement is valuable

Standard Work Forms



"Automation" itself is not a goal



The simplest automation possible

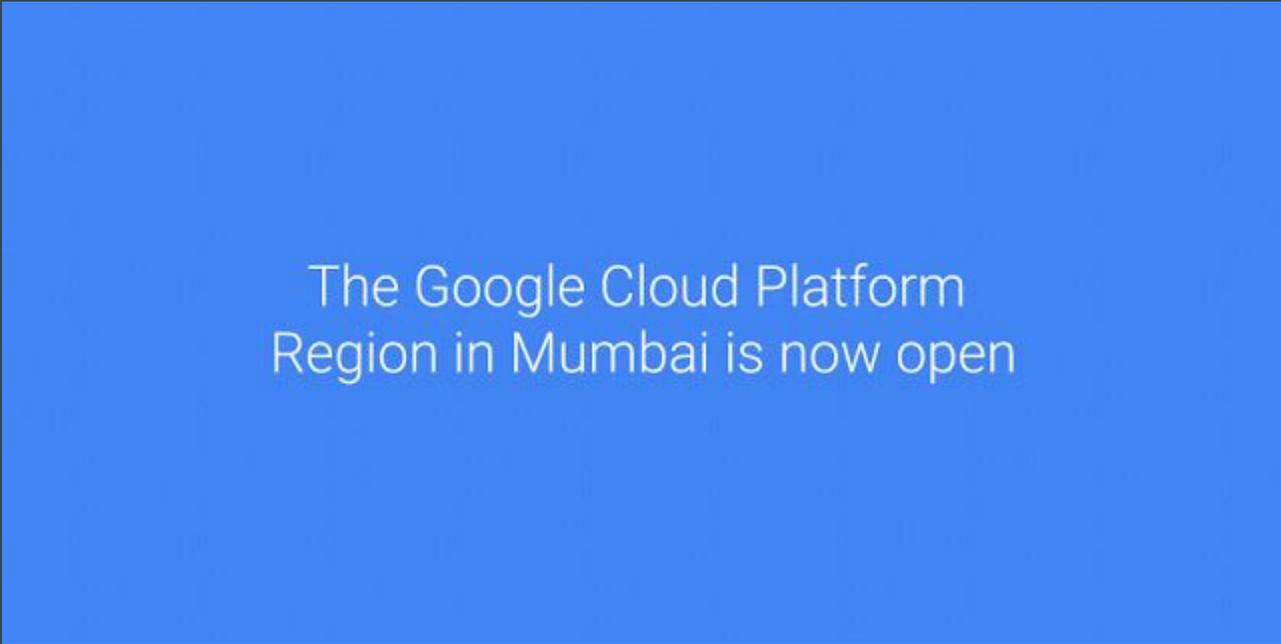


```
def do_everything(**every_option_imagineable):
```

- In addition to the instructions, we care about thinking about the interface.
- If we don't define an interface, collaboration becomes difficult.
- This is also essential to improving and replacing the implementation.

Case Study

Our actual goal wasn't automating region turnup



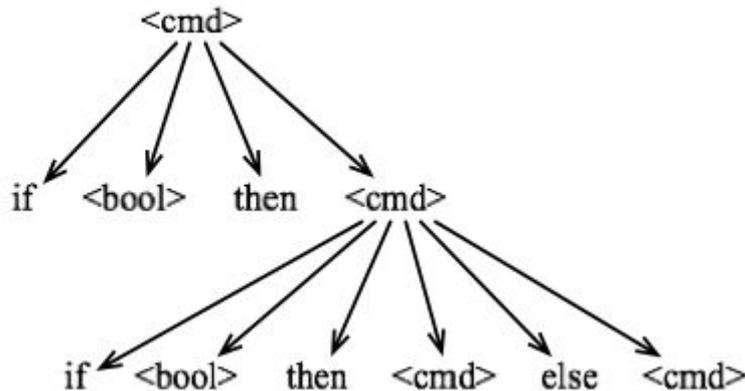
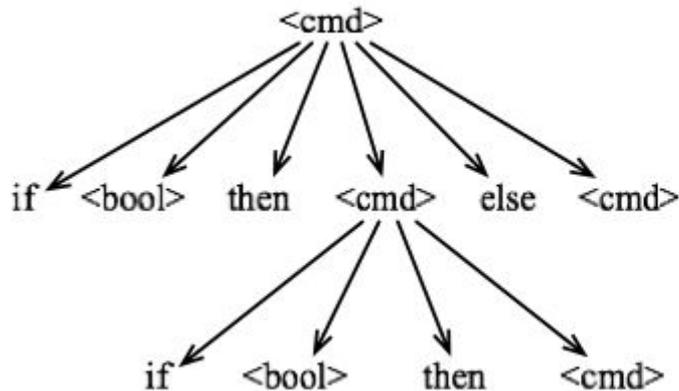
The Google Cloud Platform
Region in Mumbai is now open

(Example launch - updated map comes later on)

<https://cloud.google.com/blog/products/gcp/gcp-arrives-in-india-with-launch-of-mumbai-region>

Navigating Ambiguity

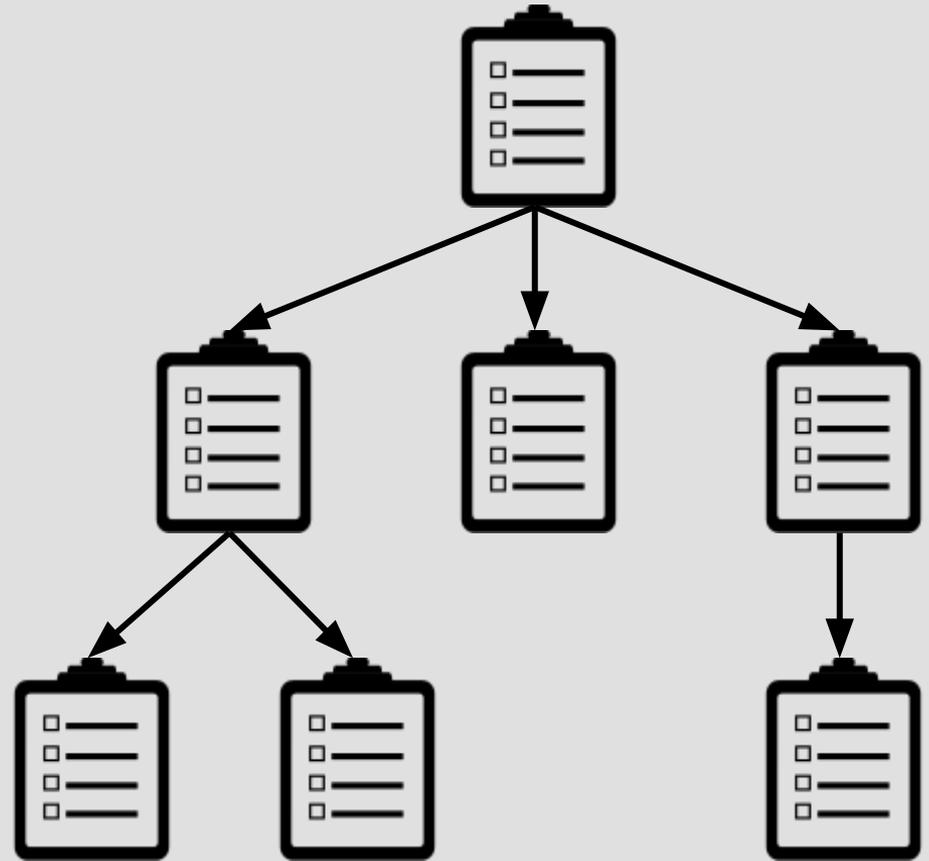
- The core team wouldn't know every detail, and that's fine!
- We defined our inputs and outputs (Really the only thing we did know)
- We worked backwards from the outputs asking product teams to do the same







VS



Incremental Steps

```
def turnup_GCE(region_name, num_zones, *compute_resources):  
    # We have no code, just a checklist.
```

We have interfaces and a service graph, but our implementation language is English.

Human RPCs

Google Search Issue Tracker

Issue Tracker Code font OFF

CREATE ISSUE

Assigned to me

Starred by me

CC'd to me

Reported by me

To be verified

Bookmark groups

Saved searches

Hotlists

Create hotlist

Create bookmark group

Browse components

Component *

Title *

Description

Create Create & Start Another Discard

Screenshot: <https://issuetracker.google.com/issues/new>

Creative Reuse of Existing Tools



Example Checklist Excerpt

FluffyCloud Regional Turnup

[Markdown](#)[Edit](#)

Edited 2019-05-09

This procedure describes how to turn up the (hypothetical) FluffyCloud service. The information required to turn up this service is as follows:

- **region**: Cloud region you're turning up.
- **datacenter**: Datacenter used to run for FluffyCloud backend for this region.
- **environment**: `prod` or `staging`.

These steps need to be performed in the order they're listed on this page.

Add this region and datacenter to the FluffyCloud configs

Edit the fluffycloud config file, which is either:

- `/fluffycloud/templates/prod_map.config`
- `/fluffycloud-staging/templates/staging_map.config`

To the backends map, add the following line:

```
"$region": "$datacenter",
```

Put these changes into a CL, have it reviewed by a teammate, and submit it.

Run the populate_database script

Run the following command, to set up FluffyCloud's database in this datacenter:

```
/fluffycloud/populate_database $environment $datacenter
```

Annotated Runbooks

```
# FluffyCloud Regional Turnup
This procedure describes how to turn up the (hypothetical)
FluffyCloud service.
```

```
The information required to turn up this service is as follows:
* `region`: Cloud region you're turning up.
* `datacenter`: Datacenter used to run for FluffyCloud backend for
this region.
* `environment`: `prod` or `staging`.
```

These steps need to be performed in the order they're listed on this page.

```
### Add this region and datacenter to the FluffyCloud configs
<!-- doctask add_configs -->
Edit the fluffycloud config file, which is either:
* `/fluffycloud/templates/prod_map.config`
* `/fluffycloud-staging/templates/staging_map.config`
To the backends map, add the following line:

...
    "$region": "$datacenter",
...

Put these changes into a CL, have it reviewed by a teammate, and
submit it.
<!-- doctask end -->

### Run the populate_database script
<!-- doctask populate_database_script depends_on add_configs -->
Run the following command, to set up FluffyCloud's database in this
datacenter:
...
/fluffycloud/populate_database $environment $datacenter
...

<!-- doctask end -->
```

Same interface, different implementation

FluffyCloud Sisyphus Freeze Sisyphus

Create a rollout Create Cancel

FluffyCloud Region Turnup	FluffyCloud Region Turnup Turnup a FluffyCloud region
Description The description of this rollout.	<input type="text" value="Turnup Antarctica"/>
Region Cloud region being turned up.	<input type="text" value="antarctica-north9"/>
Environment prod or staging	<input type="text" value="prod"/>
Datacenter Datacenter to run backend in	<input type="text" value="cold_datacenter"/>

Checklist Scaffolding

FluffyCloud Sisyphus Freeze Sisyphus

Active rollouts ▾

CREATE ROLLOUT

FluffyCloud Region Turnup

Turnup Antarctica ☆

Create user: maxluebbe
Creation time: 2019-05-09 12:34
Tasks: 4 todo – 0% completed

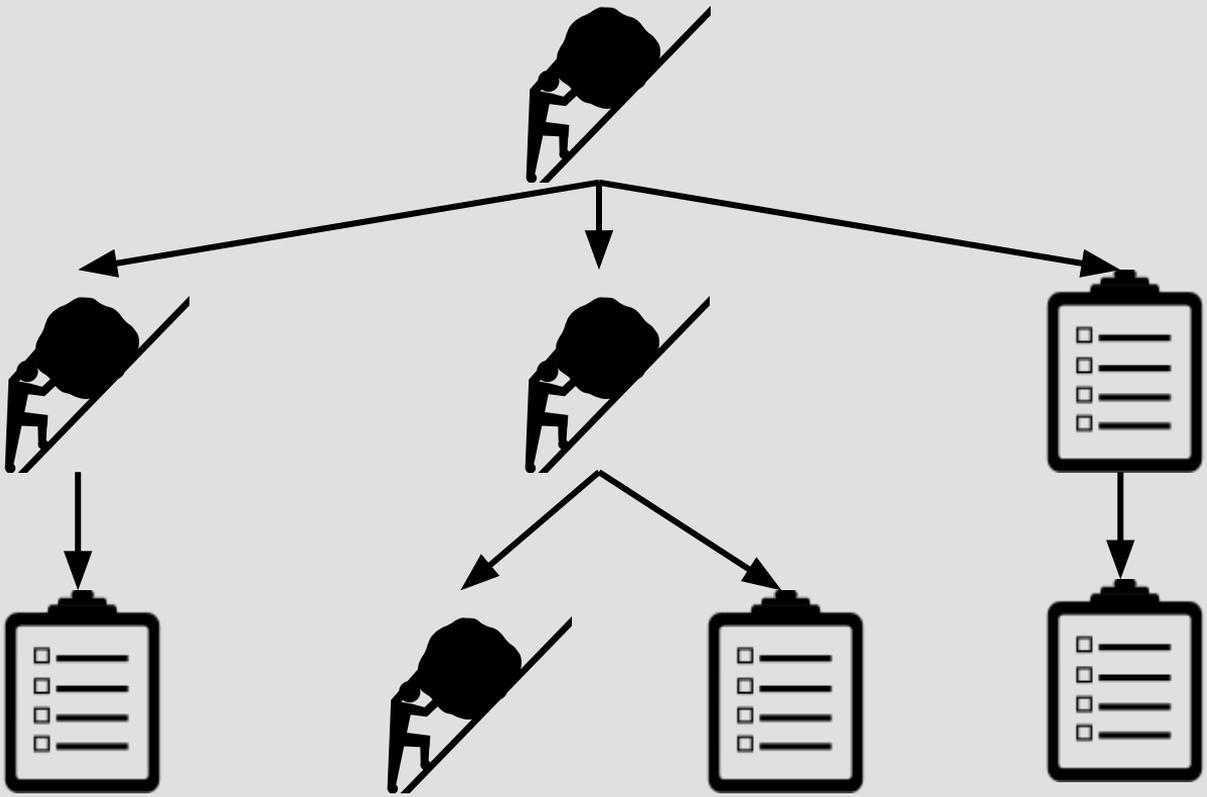
▸ Details
Scheduler: Unstarted (has not been started)

Start Delete Rollout Archive Rollout

Page 1 of 1

Tasks			Note	
▸	Add Configs	todo	Edit the fluffycloud config file, which is either: /fluffycloud/templates/prod_map.config /fluffycloud-staging/templates/staging_map.config To the backends map, add the following line: "\$region": "\$datacenter", Put these changes into a CL, have it reviewed by a teammate, and submit it.	Task Has Been Completed By A Human log more add
▸	Populate Database Script	todo	Run the following command, to set up FluffyCloud's database in this datacenter: /fluffycloud/populate_database \$environment \$datacenter	log more add
▸	Up Backend Jobs	todo	Run the following command, to bring up the backend job:: jobctl /fluffycloud/\$datacenter/fluffy.config start	log more add
▸	Add Lb Target	todo	Edit the FluffyCloud LB fragment: lb_cli edit cloud-fluffy-cloud Add the following line: BACKEND bk:cloud-fluffy-cloud \$datacenter 5000 For staging, substitute cloud-fluffy-cloud-staging for cloud-fluffy-cloud.	log more add

Hybrid Automation at Organization Scale



Wrapup

Google Cloud Platform circa March 2019



tl;dr:

- Automation isn't a goal, impact is.
- Design for incremental delivery
- Start with interfaces, checklists and SLOs
- Amplify human judgement, don't strictly look to remove it

Thanks!

@mluebbe