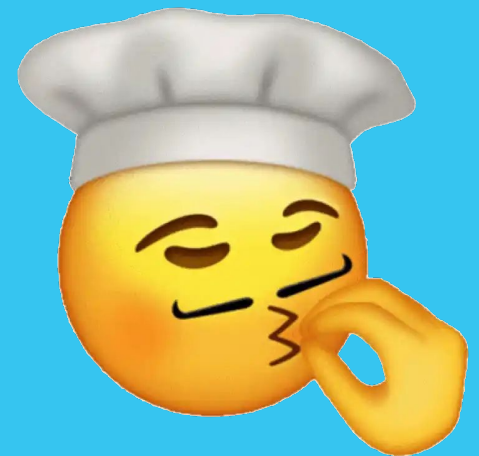


Brett Pemberton / Leigh London  
SREcon 2023 EMEA

# Scaling Chef, Emotionally





## **Brett Pemberton**

Staff Software Engineer  
Cloud Foundations - Slack



## **Leigh London**

Senior Software Engineer  
Cloud Foundations - Slack





## **Brett Pemberton**

Staff Software Engineer  
Cloud Foundations - Slack



## **Leigh London**

Senior Software Engineer  
Cloud Foundations - Slack



# 1 A brief history of time (and Chef)

Where we came from

## 2 Growth

Happy plants

## 3 Dark clouds

Some foreshadowing

## 4 Solidify the fundamentals

A set of changes

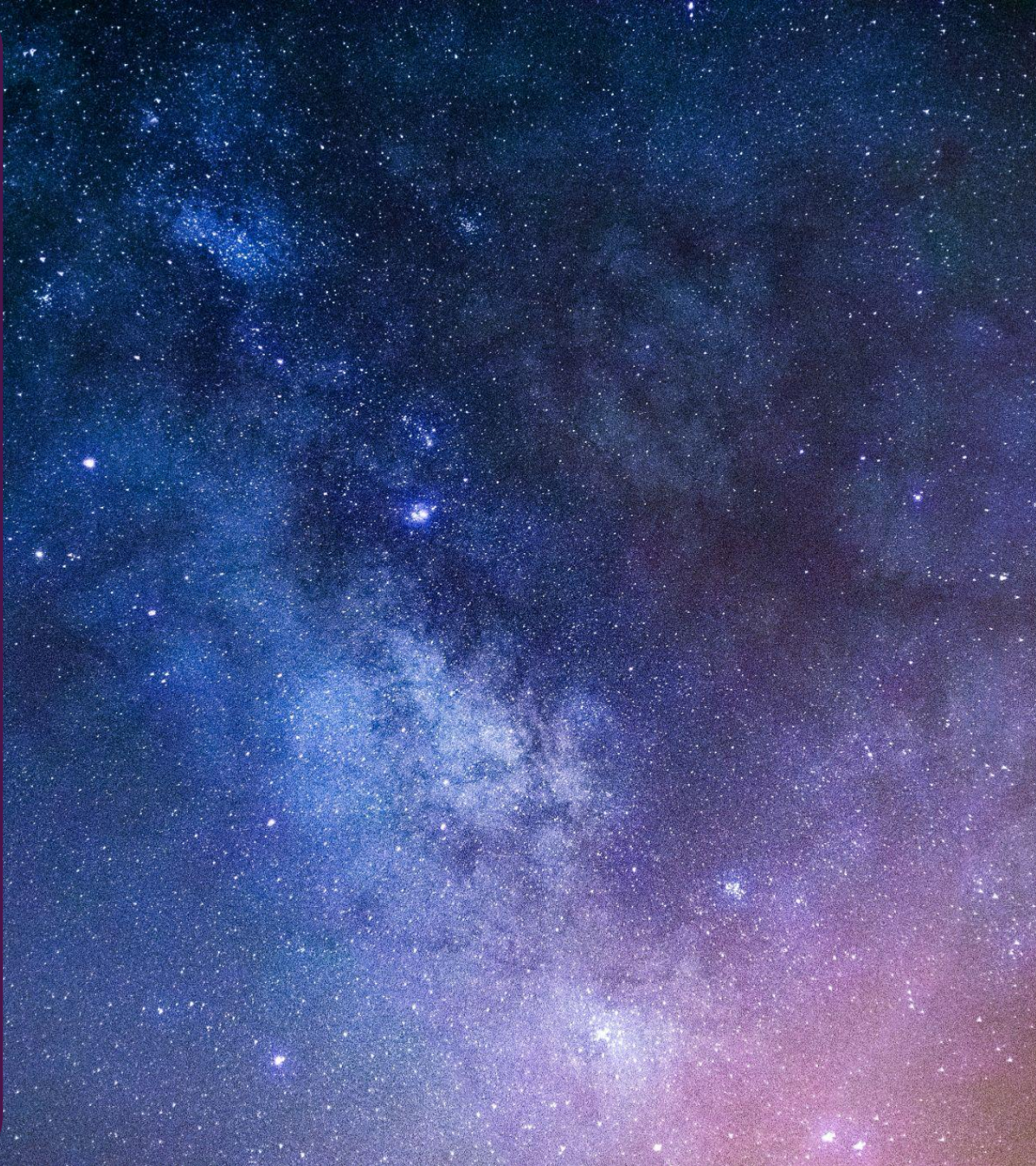
## 5 Modern era

Where we're going



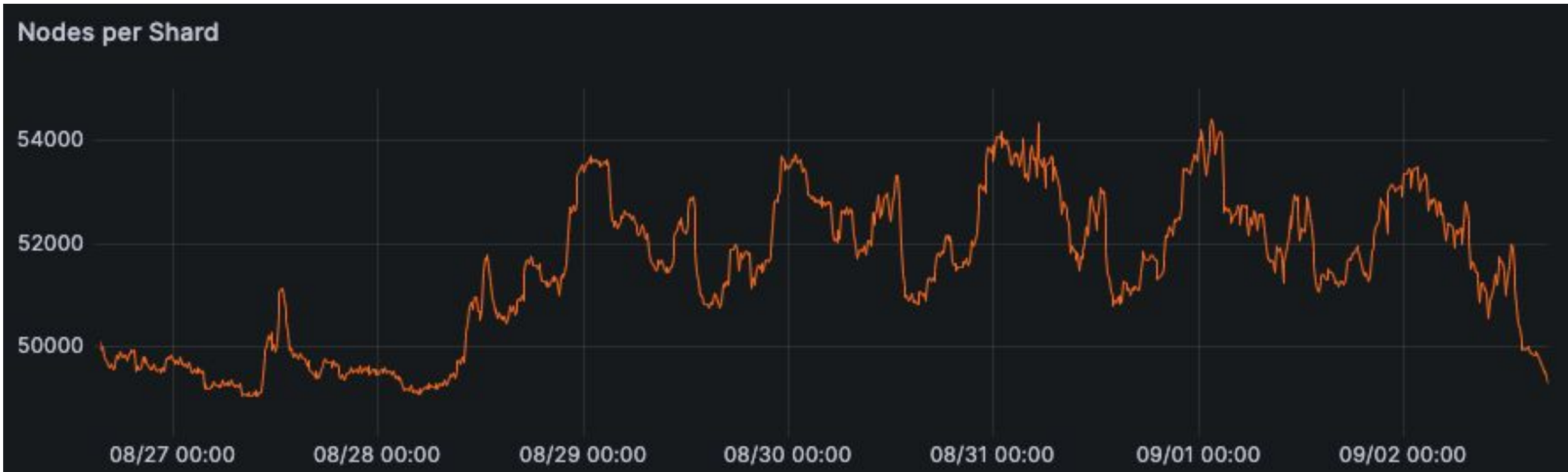


# A brief history of time (and Chef)





EC2 Instance count over a week



## s/xmpp/xmpp #1004

 Merged **george** merged 1 commit into `master` from `fix-xmpp-typo`  on Nov 14, 2015

 Conversation **1**

 Commits **1**



 Checks **0**

 Files changed **1**



**george** commented on Nov 14, 2015

*No description provided.*

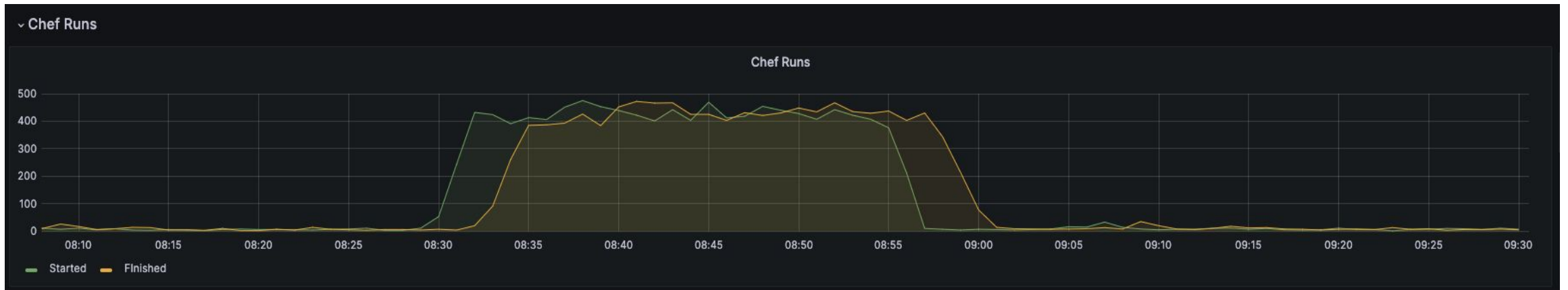
  s/xmpp/xmpp



**oof** commented on Nov 14, 2015



# A brief history of time (and Chef)







**Provisioner** APP 7 years ago

dsmith provisioned `chef-be1` as `slack-base`





**Provisioner** APP 7 years ago

dsmith provisioned `chef-be1` as `slack-base`

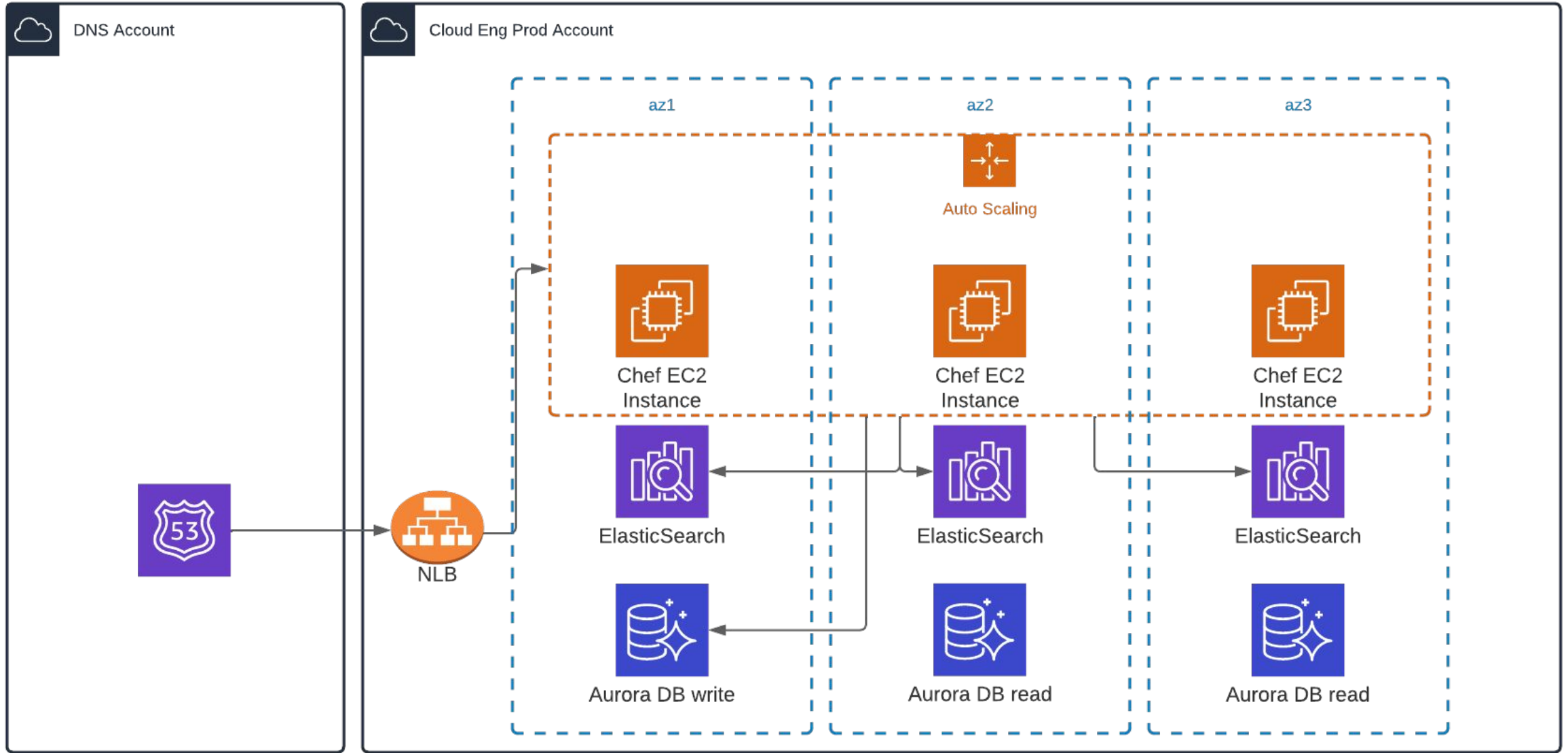


**Provisioner** APP 5 years ago

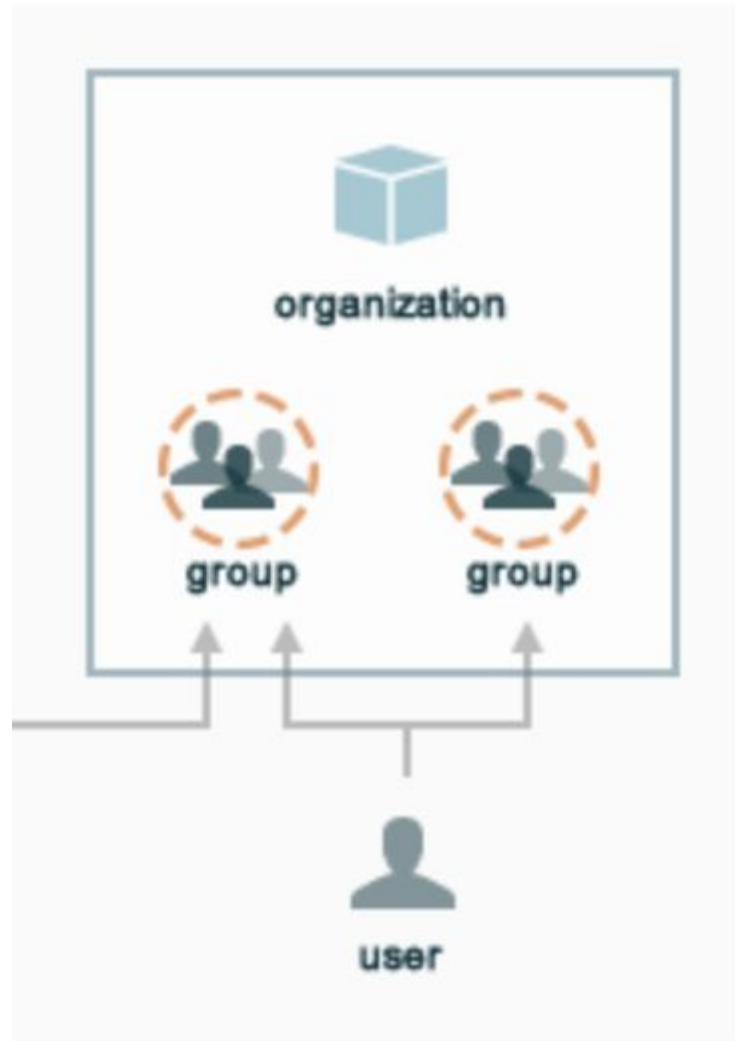
provision-service deprovisioned `i-d977c049` (`chef-be1`, `us-east-1d`, `ec2`)



# A brief history of time (and Chef)



A brief history of time (and Chef)





# Growth

A brief relaxing interlude



# Lots of growth

- The company is growing
- New teams are being formed to run new infra
- Services evolve over time
- New features tends to mean new infra to support these features... which has to run somewhere
- We have an increase in Chef nodes over time as well as infra paradigms for how some services run themselves



Growth



# Feature flags

- We can add a small bit of safety with some feature flagging within the Chef code
- It's honestly simpler than it sounds





```
extend Feature
if feature_enabled('hack_the_gibson')
  file '/etc/hack-the-planet.conf'
  action :create
  contents 'hack_the_planet = true\n'
  owner root
  group root
end
else
  file '/etc/hack-the-planet.conf'
  action :delete
end
end
```



```
default['feature']['hack_the_gibson_dri'] = '#devel-cloud-foundations'  
default['feature']['hack_the_gibson'] = {  
  'env' => {  
    'sandbox' => true,  
    'dev' => true,  
    'prod' => 25,  
  }  
}
```



# Zonal Chef

- One way to reduce our load a bit on the daily run is to splay the run... more... and run more days in a single day



## module ZonalChef

```
# Map each az to a chef run slot
# This is used only for prod, we chef dev all at once
@azid_to_chef_slot = {
  'afs1-az1' => '6',
  'afs1-az2' => '1',
  'afs1-az3' => '3',
  'apne1-az1' => '1',
  'apne1-az2' => '2',
  'apne1-az3' => '3',
  'apne1-az4' => '4',
  'apne2-az1' => '3',
```





# Dark clouds



# Not everything is a happy story

- Eventually we start to see some problems which are harder to just “scale” out of
- We have accidentally DOS’d ourselves a few times now with well-intentioned changes





**Stuart Williams** 5:07 PM

```
psql (13.6)
SSL connection (protocol: TLSv1.2, cipher: ECDHE-RSA-AES256-GCM-SHA384, bits: 256, compression: off)
Type "help" for help.

opscode_chef=> select count(*) from cookbook_version_dependencies;
 count
-----
 84719
(1 row)
```

My memory is somewhat fuzzy, but that exploded last time at 20k or something, didn't it?



1





**Stuart Williams** 5:07 PM

```
psql (13.6)
SSL connection (protocol: TLSv1.2, cipher: ECDHE-RSA-AES256-GCM-SHA384, bits: 256, compression: off)
Type "help" for help.

opscod_chef=> select
count
-----
84719
(1 row)
```

My memory is somewhat



1



The number of cookbooks per organization, across all versions, on the Chef Infra Server is **under 500**.

(Multiple organizations with fewer than 500 cookbooks each, that still add up to a total of more than 500 cookbooks server-wide, is fine.)

[https://docs.chef.io/server/capacity\\_planning](https://docs.chef.io/server/capacity_planning)



# Problems with scale

- The scale that we operate at means that we have to be careful of doing big changes and blowing things up



S2

S2

S1





## Infrastructure Freeze Window

Starting immediately, as of today, we will pause chef and AMI changes while we come up with a new safe way to make infrastructure changes in production. Please visit [#infrastructure-safety](#) for more!

### What are we doing

- We are freezing all chef  and AMI  changes. This means all cookbooks used by all teams will be frozen. New AMI versions will not be created.





### Infrastructure Free

Starting immediately, a  
with a new safe way to  
[#infrastructure-safety](#)

### What are we doing

- We are freezing  
all teams will be



anges while we come up  
Please visit

cookbooks used by





# Solidify the fundamentals



# Solidify the fundamentals

- We have pretty standard things in software engineering that we can use to help in this
  - Standardise review processes and questions (slightly painful)
  - Continuous Integration and automated testing (delightful)
  - Tracking progress and targeting the next fixes for biggest wins (simple)





# CAB process

Oh no



# CAB process

Oh no

#infra-change-control





# CAB process

Oh no



**Standard Change** WORKFLOW 16:25



**@brett** is requesting approval for a standard change.



1



# CAB process

Oh no



**Low Risk Change** WORKFLOW 01:00

 [@mccv](#) is requesting approval for a low-risk change.

 1  1 



# CAB process

Oh no



**Notify of Exempt Change** WORKFLOW 01:34

@md5 is proceeding with an exempted change.



3 replies Last reply 2 years ago





## Fundamentals

### Labels

**cab-dep-low-blast-radius**

**cab-dep-low-risk**

### Labels

**cab-dep-high-blast-radius**

**cab-dep-standard-risk**



Fundamentals

Labels

cab-dep-low-blast-radius

cab-dep-low-risk

Labels

cab-dep-high-blast-radius

cab-dep-standard-risk

Labels

cab-dep-emergency-change

cab-dep-high-blast-radius

cab-dep-standard-risk

Labels

cab-dep-ff-removal

cab-dep-ff-rollout

cab-dep-high-blast-radius

cab-dep-standard-risk

cab-dep-test-kitchen

Labels

cab-dep-ff-rollout

cab-dep-low-blast-radius

cab-dep-low-risk





Fundamentals

Labels

cab-dep-low-blast-radius

cab-dep-low-risk

Labels

cab-dep-high-blast-radius

cab-dep-standard-risk

Labels

cab-dep-emergency-change

cab-dep-high-blast-radius

cab-dep-standard-risk

Labels

cab-dep-ff-removal

cab-dep-ff-rollout

cab-dep-high-blast-radius

cab-dep-standard-risk

cab-dep-test-kitchen

Labels

cab-dep-ff-rollout

cab-dep-low-blast-radius

cab-dep-low-risk

Labels

cab-dep-failed-change

cab-dep-high-blast-radius

cab-dep-standard-risk

Labels

cab-dep-high-blast-radius

cab-dep-standard-risk

spicy

Labels

cab-dep-low-blast-radius

die-jenkins-die

cab-dep-low-risk



# Automate and make it easy to do the right thing

- Sounds kinda complicated? But it's not really
- Testing: make it easy to test changes in a standard way
  - **test-kitchen** and what we call **gitchen**
- Analytics: look at the code and track it
  - Honestly we do this pretty simply. Basically **find** and **grep**
- Tracking progress: make smart data-driven decisions



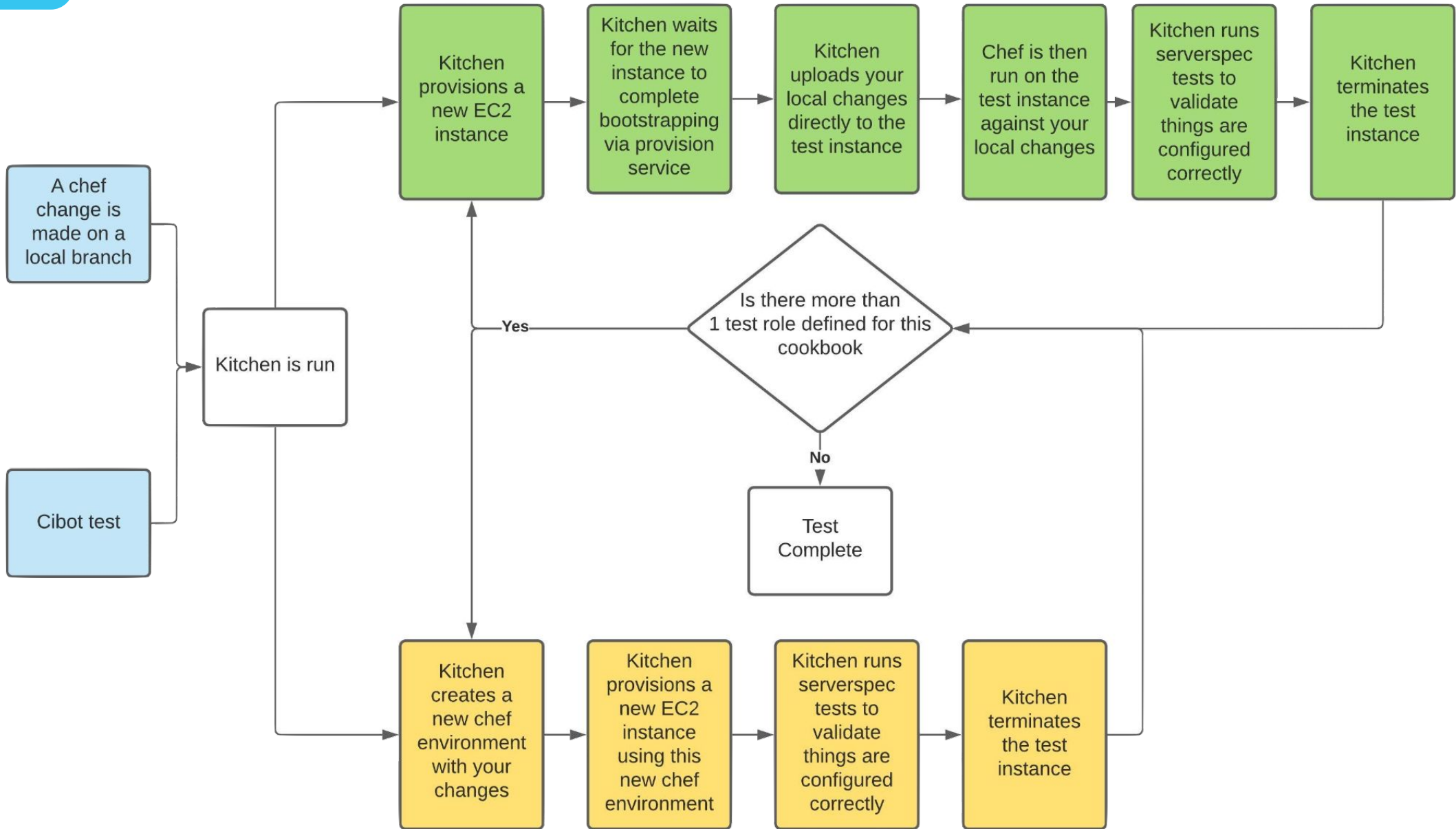
# What's in the kitchen?

We call it a “gitchen”

- The open-source Chef project for testing stuff is **test-kitchen** and it's honestly pretty great
- We have a pretty complex environment and a *ton* of common **kitchen.yaml** boilerplate so we made a bit of a wrapper that takes a smaller/different yaml and generates **kitchen.yaml** and executes it internally
- We call it **gitchen** because it's a **Go** wrapper program for **kitchen**
  - Nothing to do with **git**, we're just “really good” at naming things



## Fundamentals



# Validation, because Chef is scary

## Exit codes can lie

- Just because a Chef run passes doesn't mean that *end-user* impact is going to be all good
- We can use another open-source project to do some testing-validation-code-stuff for us!
- We can use ServerSpec to let teams write some pretty straightforward tests
- After Chef runs we have a step that will invoke ServerSpec and can emit errors or even terminate the host if they don't pass



```
slack_validation_serverspec_test 'atop_enable_atop' do
  raw_test <<-EOF
describe file('/etc/default/atop') do
  it { should exist }
end
describe service('atop') do
  it { should be_enabled }
  it { should be_running }
end
  EOF
  instance_test true
end
```





# Safety Score

- Figuring out the “safety” and “tracking it” was important because we had to be able to communicate it
- We developed an **Infra Safety Score** to do with checking how many cookbooks have test coverage, how much of the fleet was checked with these kitchen tests, etc
- It is scored automatically in our data warehouse & analytics dashboards, via data we extract from our code and systems
- Important benefit - we didn't to track the score by having people update a JIRA ticket or a spreadsheet

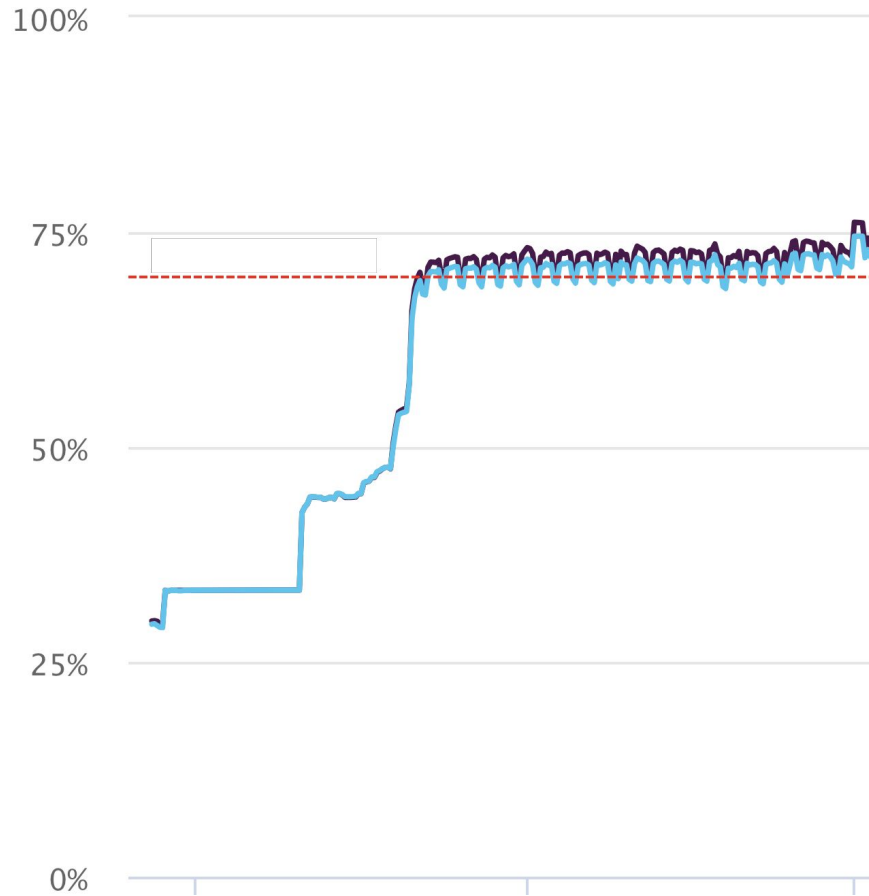


# Safety Score

#	Team	Safety Score	# of Services	# of Chef Roles	ServerSpec Tests Coverage	ServerSp
1	Flannel	97	1	1	100	100
2	Cloud Foundations	97	4	8	100	100
3	Cloud Eng	97	1	1	100	100
4	Bedrock Core	96	2	2	100	100



# Safety Score



Infrastructure Safety Score (All services)

# 76.22%



# Neat

- With a few straightforward tricks we now have a more scalable system that is giving us more safety
- We have more levers to lean on for reliability and testing



**NICE**



**Modern era**





▼ Pins



**Patrick Bernal**

 proj-chef-client-upgrades (archived) | Dec 3rd, 2021 at 5:15 AM

Hello Chef DRIs! 

Back in [January 2016](#) we moved from Chef Client 11 to Chef Client 12. Since then, we've stayed at that release while the world moved on, and this release became unsupported and EOL. But now, the time has come for all services to upgrade their Chef clients!



**TL;DR**

- **What:** Upgrade Chef client to 16.x.





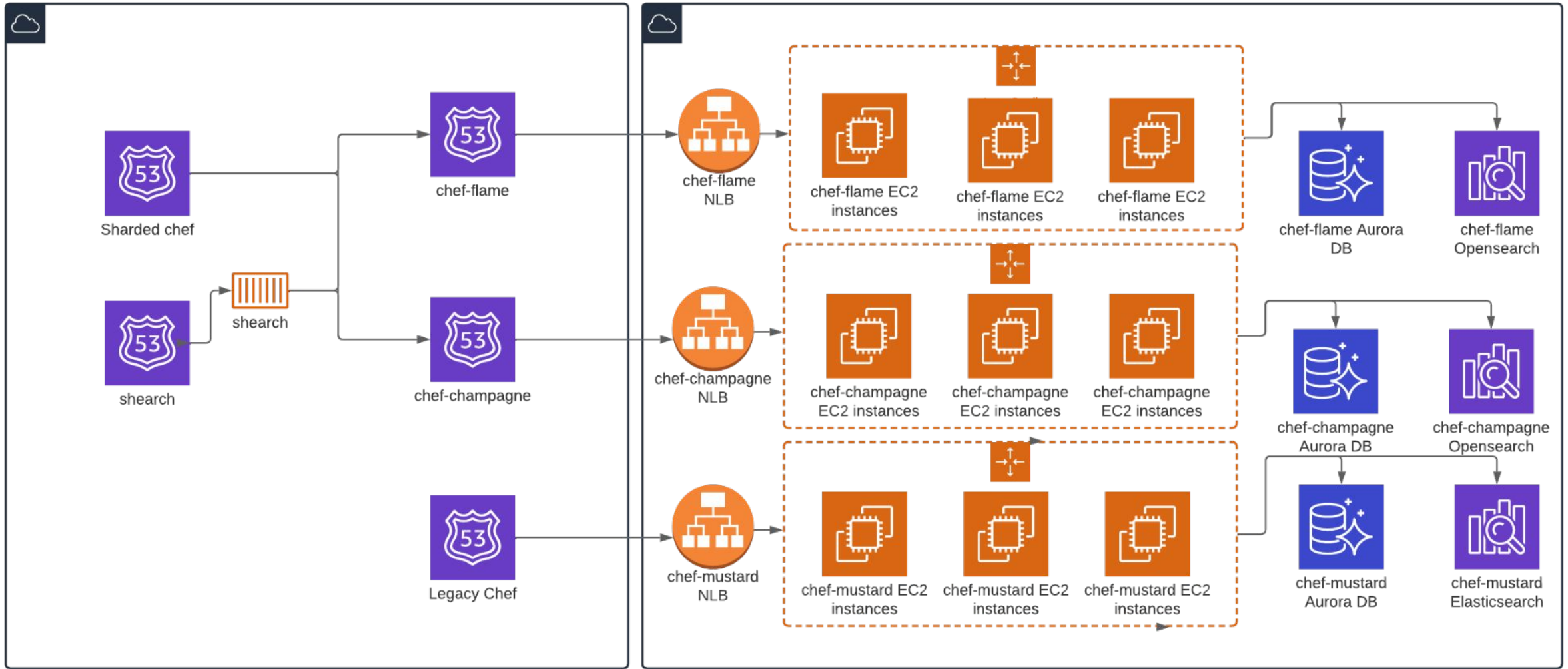
## All checks have passed

25 successful checks

✓		<b>buildkite/slack-chef-repo-chef-syntax</b> — Build #13778 passed (5 seco...	<a href="#">Details</a>
✓		<b>buildkite/slack-chef-repo-chef-test-kitchen</b> — Build #8318 passed (3...	<a href="#">Details</a>
✓		<b>buildkite/slack-chef-repo-chef-test-kitchen-ubuntu-2204</b> — Build #8...	<a href="#">Details</a>
✓		<b>buildkite/slack-chef-repo-chefspec</b> — Build #13551 passed (36 secon...	<a href="#">Details</a>
✓		<b>buildkite/slack-chef-reno-code-freezes</b> — Build #35215 passed (4 sec	<a href="#">Details</a>



Modern era





# Takeaways



## Growth

Things take time

Make it easy

Numbers

### **Growth:**

- Incremental growth makes incremental scaling easy
- Triggers for architectural changes to support hypergrowth are harder to come by



Growth

**Things take time**

Make it easy

Numbers

**Things take time:**

- We've only been talking for a few minutes now but this talk is spanning literal years of evolution
- An acceptable system is just that - **acceptable** - and we kept iterating



Growth

Things take time

**Make it easy**

Numbers

**Make it easy:**

- Make the good thing to do the easy thing to do
- Simplifying the process and shifting it closer to code made it way less management overhead and busywork
- Automatic dashboarding, if it's available to you, is phenomenal for this type of work





Growth

Things take time

Make it easy

**Numbers**

**Numbers:**

- Keep cookbook numbers down like Chef recommends and don't accidentally blow up your system
- And don't have this exact incident happen multiple times



# Thanks!

