VCs on line 1

Scaling under pressure with **Chef, Packer** and **Terraform**

Opex Founders
Sanju Burkule & Gunanand Nagarkar

15-20 years Global Experience

(US, Europe, India)

Monitoring &

infrastructure domain

R&D background

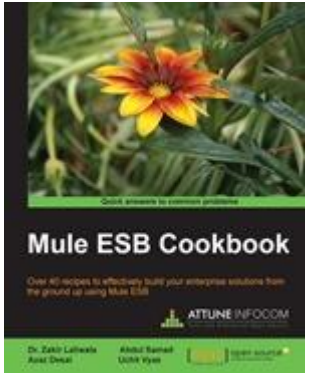
Config Management Camp

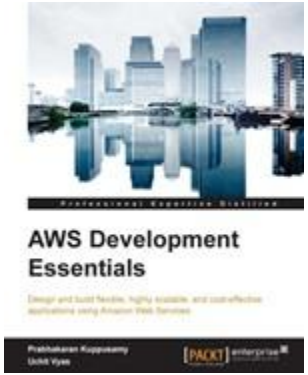
Who are we: Uchit Vyas

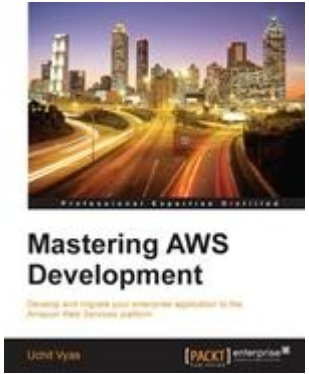
6 years of exciting experience

Infrastructure automation domain

Leading Cloud Infrastructure automation in Opex Software







Introduction

Opex

Single metric we measure: Speed *with correctness*

Mission: Automate IT. Create Time

Focus: SAAS applications

Config Management Camp

SAAS Startup Dream
Scaling with SPEED

How to make this happen, technically?

Images: 4actionmarketing.net, dreamstime.com, thenextweb.com

Pressure from investors

How to make this happen, technically?

Images: startupinitiative.com

SAAS is value: Step 1 of 3

Clear value is delivered

SAAS is value: Step 2 of 3

Clear value is delivered

Onboarding new customers *rapidly*

SAAS is value: Step 3 of 3

Clear value is delivered

Onboarding new customers *rapidly*

Upgrade fast, maintain lead,
make value irresistible

Mortality Rate

Images: jeepneymanilaph.files.wordpress.com

Execution Problems

All good

Execution Problems

All good

Harden the OS *skill not available immediately*
Automate app deploy/cfg *after thought*
Automate Testing *fully* *after thought*

Execution Problems

All good

Harden the OS *skill not available immediately*
Automate app deployment *after thought*
Automate app configuration *after thought*

Microservices *independently upgradable?*
Lead *starting to lose speed, competition!*

Images: avopress.com, animationmagazine.net, thenextweb.com

DevOps + Strong Ops

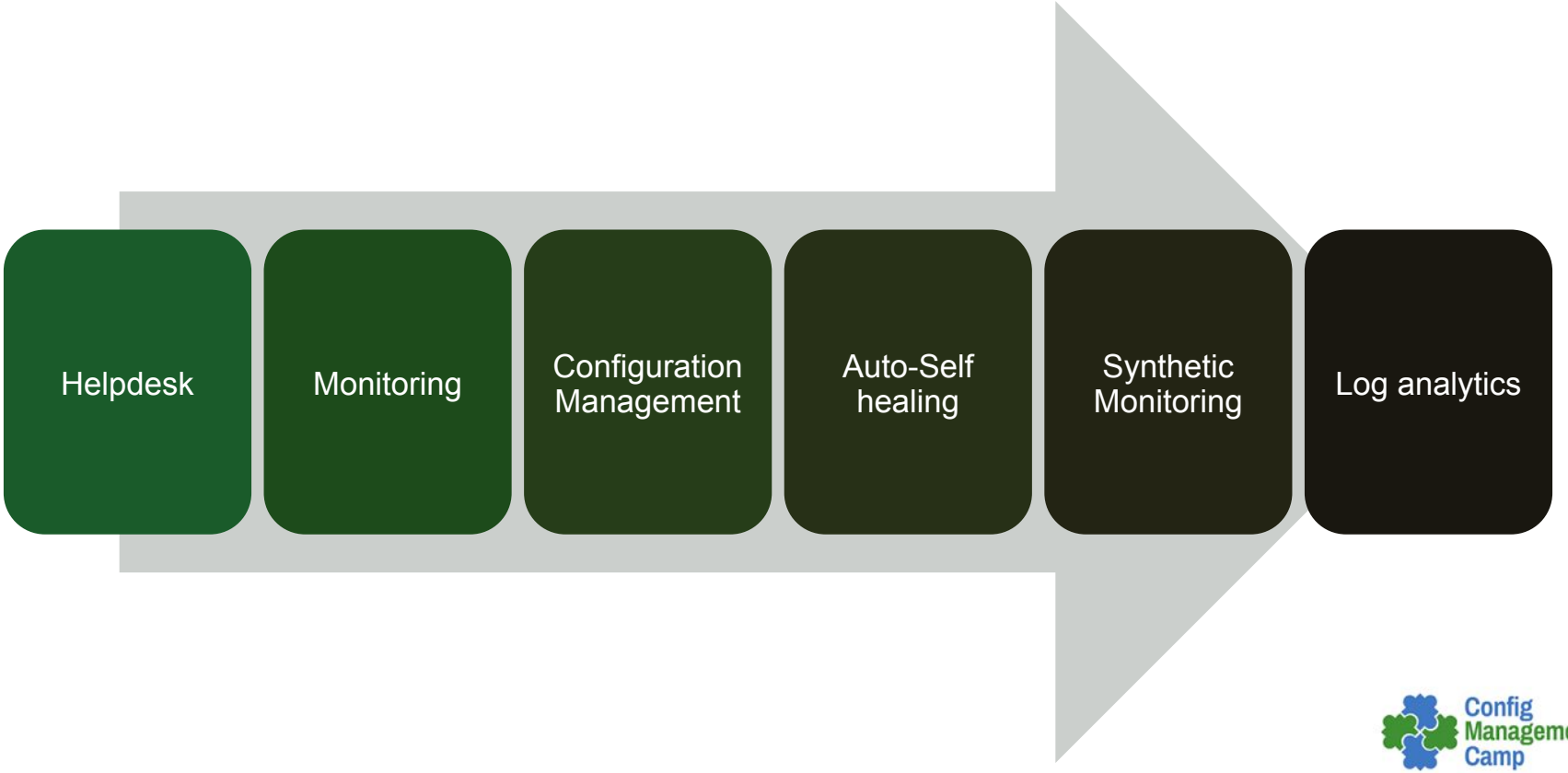
Transform test automation for DevOps

Test bed Infrastructure creation Automation

Test Automation (Selenium, QTP etc)

Automated Test Result Analytics

Strong Ops to scale



Good News

Opex

Config Management Camp

OS Hardening: Why is it imp?

- protecting IP
- unfair competition
- cybersecurity
- private data
- direct or indirect
- attacks via cloud

President Barack Obama delivers remarks at the Business Roundtable offices in Washington September 16, 2015.

OS Hardening

Use Chef templates and Terraform

Include following in server configuration definition

- Security Rules, Password policies,
- Secure SSH, Compliance policy
- Important agents (AV, monitoring)

Use “Chef-Vault” for storing secrets

Sample code: login definition

```
template '/etc/login.defs' do
 source 'login.defs.erb'
 mode '0444'
 owner 'root'
 group 'root'
 variables(
   additional_user_paths: node['env’] …
 )
end
```

Sample code: erb file

'sample.erb'

```
<% if @port != 80 -%>
 Listen <%= @port %>
<% end -%>
```

Two simple, but powerful concepts

a) Expression evaluation

b) Variable value replacement

Sample code: login definition

```
variables(
   password_max_age: node['auth']['pw_max_age'],
   password_min_age: node['auth']['pw_min_age'],
   login_retries: node['auth']['retries'],
   login_timeout: node['auth']['timeout'],
   chfn_restrict: '', # "rwh"
   allow_login_without_home: node['auth']
['allow_homeless'],

 …
 )
```

Quality gates: Serverspec

Suse 11 Hardening

☑ Passed ☑ Failed ☑ Pending

1262 examples, 68 failures
Finished in 4003.69032 seconds

File "/etc/ntp.conf"	
should exist	1.20977s
should be file	1.16359s
File "/etc/pam.d/login"	
should exist	1.45530s
should be file	1.17264s
File "/etc/motd"	
should exist	1.14274s
should be file	1.23089s
File "/etc/nsswitch.conf"	
should exist	1.11210s
should be file	1.14907s
File "/etc/resolv.conf"	
should exist	1.18897s
should be file	1.14980s
File "/etc/vimrc"	
should exist	1.17678s
should be file	1.14994s

High speed lab creation

- Terraform is parallelized
- Auto-sequencing based on graphs
- Terraform can integrate with any layer of the stack
- Lab setup on pre-existing servers, provisioning servers from scratch - both are supported

Opex

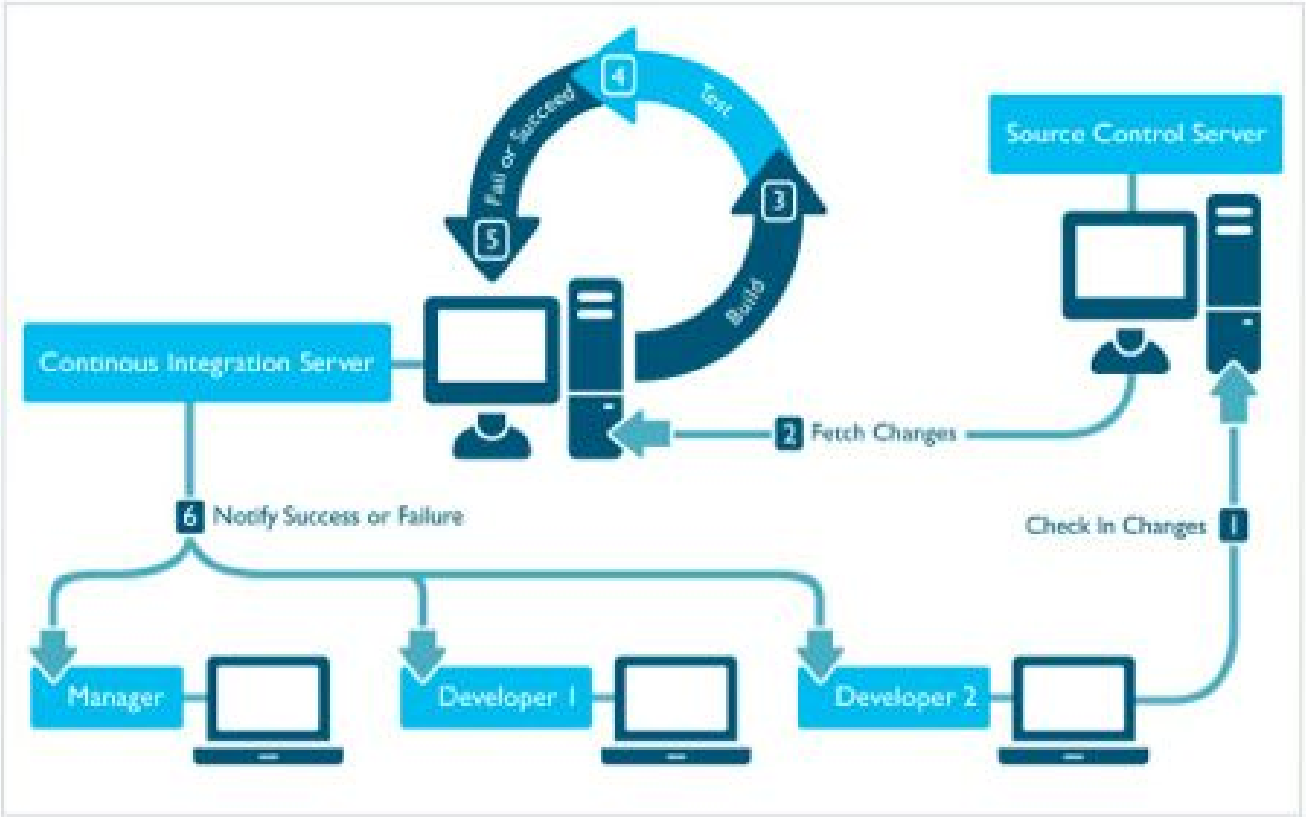
Config Management Camp

Speed of provisioning

Why we chose terraform for high-speed scaling

Number of Machines	Chef-metal	Terraform
10	2.7 minutes	1.20 minutes
30	3.9 minutes	3.08 minutes
40	5.6 minutes	3.36 minutes
60	9.4 minutes	7.08 minutes
100	15.2 minutes	7.41 minutes

CI Server in DevOps



TestNow
For DevOps

Development
(local, remote, outsourced)

Test/QA
(local, remote, outsourced)
- Load testing

Production
(local, remote, outsourced)
- Cloud load testing
- Monitoring

TestNow For DevOps

Dashboard | URL Test | Manual Test | Automated Test | ScaleNow | Infra | Docs

URL Test

Select Browsers: Chrome: 0 | Firefox: 0 | IE: 0 | Opera: 0 | Android: 0 | AndroidChrome: 0 | Safari: 0 | UIPerfAnalysis: 0

- Check All
- Uncheck All
- 43
- 42
- 41
- 40
- 39
- 38
- 37
- 36

Application URL: http://yourdomain

Text To Search: Text to search in page.

Timeout (mins): 10

Execution Strategy: Efficient / Boost

Test Now

TestNow For DevOps

Dashboard | URL Test | Manual Test | Automated Test | ScaleNow | Infra | Docs

ScaleNow

1 SELECT BROWSER — 2 CODEBASE — 3 EXECUTION DETAILS

Select Browser: Chrome: 0 | Firefox: 0 | Opera: 0

Application Name: Application Name

Application URL: http://yourdomain

Parallel Users Count: 10

Hours to Run: 1

☑ Incremental Load

Increase Users with: 1

Interval(Min): 2

Next

Select Browsers	
Application Name	
Application URL	http://yourdomain
Parallel VMs Count	
Hours to Run	
Git Repo	
User ID	ganesh.khakare@opexsoftware.com
Git branch	
Test Run Command	
Pre - script	echo "Executing PreScript"
Post - s	

TestNow
For DevOps

Dashboard | URL Test | Manual Test | Automated Test | ScaleNow | Docs

89 min
Last Job Execution Time

Last Fail-Fast Time

2125778 of 2860530
Test Cases Passed

132 of 157
Releasable builds

Consolidated Report - JobID 8cdd30ca - 450_Users_WithLB_and_without_onboaring

Consolidated Report

Provisioning 0 | Configuring 0 | Running 0 | Success 150 | Failed 0

Offline

Distributing load elegantly

Opex

Next few slides explain the code.

Key highlighted portions are in a red rectangle

Use cases 1: Synthetic monitoring

Use case 2: Load balancing across all regions in a cloud

How to scale across AZs? How to scale across clouds?

Config Management Camp

.tf Provisioning (AWS multi AZs)

```
provider "aws" {
  region = "us-west-2"
  access_key = "XXXXXXX"
  secret_key = "XXXXXXXXX"
}
```

Variables for AZs and AMIs

```
variable "region"
{
  default = "us-west-2"
}
variable "region_az" {
  default = {
  "us-east-1" = "us-east-1a,us-east-1c,us-east-1d,us-east-1e"
  "us-west-1" = "us-west-1a,us-west-1b,us-west-1c"
  "us-west-2" = "us-west-2a,us-west-2b,us-west-2c"
  "eu-west-1" = "eu-west-1a,eu-west-1b,eu-west-1c"
  "eu-central-1" = "eu-central-1a,eu-central-1b"
  "ap-southeast-1" = "ap-southeast-1a,ap-southeast-1b"
  "ap-northeast-1" = "ap-northeast-1a,ap-northeast-1b,ap-northeast-1c"
  "ap-southeast-2" = "ap-southeast-2a,ap-southeast-2b"
  "sa-east-1" = "sa-east-1a,sa-east-1b,sa-east-1c"
  }
}
```

Lets use this region to start

Region based AZ map

```
variable "region_az" {
  default = {
  "us-east-1" = "us-east-1a,us-east-1c,us-east-1d,us-east-1e"
  "us-west-1" = "us-west-1a,us-west-1b,us-west-1c"
  "us-west-2" = "us-west-2a,us-west-2b,us-west-2c"
  "eu-west-1" = "eu-west-1a,eu-west-1b,eu-west-1c"
  "eu-central-1" = "eu-central-1a,eu-central-1b"
  "ap-southeast-1" = "ap-southeast-1a,ap-southeast-1b"
  "ap-northeast-1" = "ap-northeast-1a,ap-northeast-1b,ap-northeast-1c"
  "ap-southeast-2" = "ap-southeast-2a,ap-southeast-2b"
  "sa-east-1" = "sa-east-1a,sa-east-1b,sa-east-1c"
  }
}
variable "ami"
{
  default = ...
```

used in lookup

Region based Ubuntu AMI map

```
variable "ami"
{
  default =
{
    "description" = "Ubuntu server 14.04 ami id"
    "us-west-1" = "ami-df6a8b9b"
    "us-west-2" = "ami-5189a661"
    "us-east-1" = "ami-d05e75b8"
    "eu-west-1" = "ami-47a23a30"
    "eu-central-1" = "ami-accff2b1"
    "ap-northeast-1" = "ami-936d9d93"
    "ap-southeast-1" = "ami-96f1c1c4"
    "ap-southeast-2" = "ami-69631053"
    "sa-east-1" = "ami-4d883350"
  }
}
```

Resource declaration

```
resource "aws_instance" "web" {
  ami = "${lookup(var.ami, var.region)}"
  instance_type = "${var.instance_type}"
  count = "${var.servers}"
  availability_zone =
        "${element(split(",",lookup(var.region_az,
 var.region)),
        count.index%length
         (split(",",lookup(var.region_az, var.
 region))
          ))}"
```

Scaling and iterating

```
resource "aws_instance" "web" {
  ami = "${lookup(var.ami, var.region)}"
  instance_type = "${var.instance_type}"
  count = "${var.servers}"
  availability_zone =
        "${element(split(",",lookup(var.region_az,
var.region)),
        count.index%length
         (split(",",lookup(var.region_az, var.
region))
          ))}"
```

Scaling to 100s of servers

Math library added from terraform v0.4
Simulating iteration

Multi-cloud distribution

Step 1
Creating an image for each cloud

Step 2
Using that image in the code that we just saw

Creating image using Packer

```
"provisioners": [{
    "type": "shell",
    "inline": [
      "sleep 30",
      "sudo apt-get update",
      "sudo apt-get install -y redis-
server"
    ]
  }]
```

Provisioning Redis server

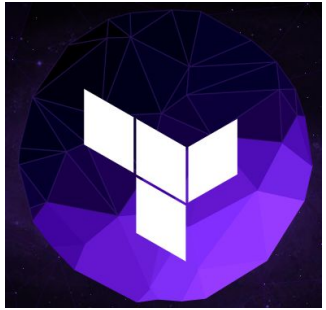
Strong Ops - Chef provisioner

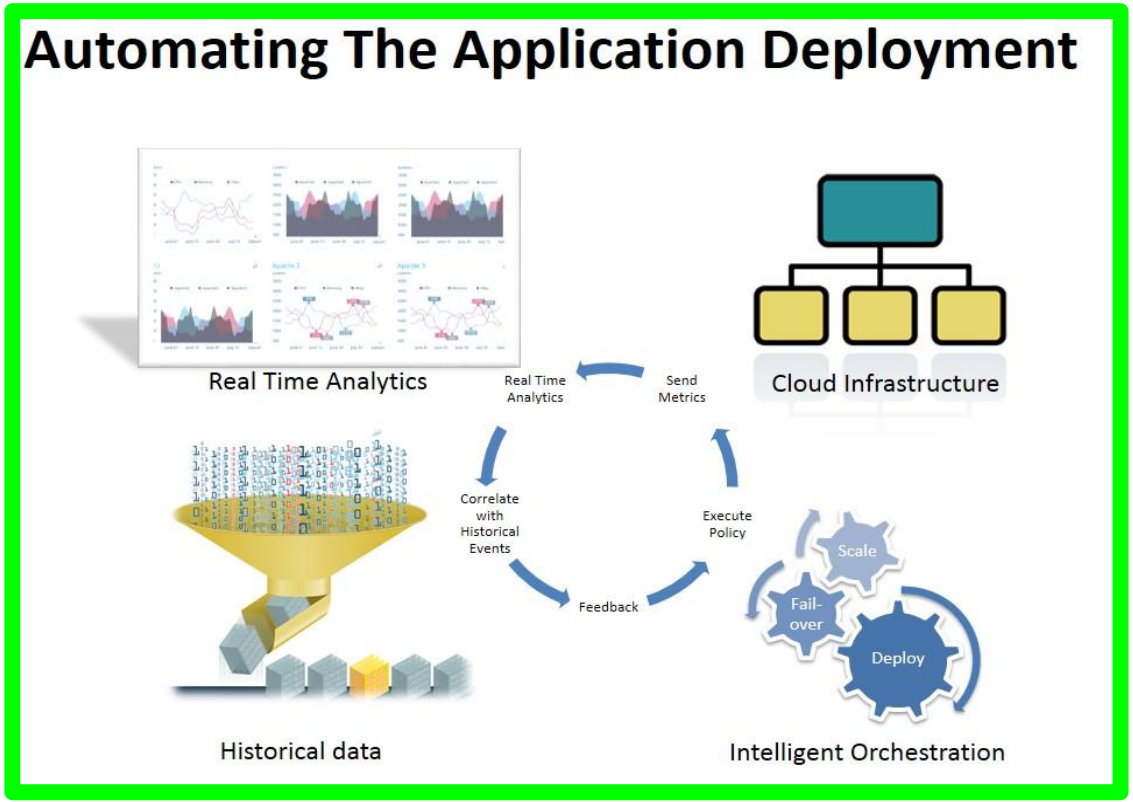
```
"provisioners": [{
    "type": "chef-client",
    "server_url”: “https://mychefserver.com/”
  }]
```

Using Chef client-server

Cookbooks and Recipes:
Helpdesk, monitoring server, monitoring clients, antivirus agents, auto-healing servers, analytics data couriers...

App context: Data bags





Creating image using Packer

```
"builders": [{
   "type": "amazon-ebs",
   "access_key": "{{user `aws_access_key`}}",
   "secret_key": "{{user `aws_secret_key`}}",
   "region": "us-east-1",
   "source_ami": "ami-de0d9eb7",
   "instance_type": "t1.micro",
   "ssh_username": "ubuntu",
   "ami_name": "packer-example {{timestamp}}"
 },
```

Build in Amazon Cloud

Same code for multi-cloud

```
"builders": [{
    "type": "amazon-ebs",
    "access_key": "{{user `aws_access_key`}}",
    "secret_key": "{{user `aws_secret_key`}}",
    "region": "us-east-1",
    "source_ami": "ami-de0d9eb7",
    "instance_type": "t1.micro",
    "ssh_username": "ubuntu",
    "ami_name": "packer-example {{timestamp}}"
  },
```

Build in Amazon Cloud

```
{
    "type": "digitalocean",
    "api_token": "{{user `do_api_token`}}",
    "image": "ubuntu-14-04-x64",
    "region": "nyc3",
    "size": "512mb"
  }],
```

Build in Digital Ocean Cloud

Beautiful multi-color output

```
amazon-ebs output will be in this color.
digitalocean output will be in this color.

==> digitalocean: Creating temporary ssh key for droplet...
==> amazon-ebs: Prevalidating AMI Name...
==> amazon-ebs: Inspecting the source AMI...
==> amazon-ebs: Creating temporary keypair: packer 55f6c5e5-2b50-c8c3-5e37-7d246b6f0bca
==> amazon-ebs: Creating temporary security group for this instance...
==> amazon-ebs: Authorizing access to port 22 the temporary security group...
==> amazon-ebs: Launching a source AWS instance...
==> digitalocean: Creating droplet...
==> digitalocean: Waiting for droplet to become active...
```

Hope this helps!

Pilot of great idea - clear value delivered - all good

Onboarding new customers *rapidly* without sacrificing quality

- Harden the OS - chef-templates, packer, Terraform
- Automate app deployment - Terraform + Chef
- Automate app configuration - Terraform + Chef + Data bags

Pivot fast, maintain the lead, make value irresistible

- DevOps for building new features fast - Adopt DevOps early
- Upgrade customers **DevOps pipeline + Roles + Environments**

Why DevOps projects fail

Opex

The 4th question in daily agile scrums that was never asked

4th question: Are you blocking Ops?

Meaning:

a) Are there any new binaries created/deleted?
b) Are there any configuration files that changed?
c) Are there any configuration attributes that changed?

Config Management Camp

Recap, takeaway

Easy to code infrastructure
Easy to replicate infrastructure
Easy to distribute infrastructure across AZs
Easy to balance infrastructure across regions
Easy to balance infrastructure across clouds,countries
Easy to integrate with other apps
The 4th scrum question

Opex forecast: Next few years

Opex

Cheaper, stable, scalable, elastic infra needed by millions of businesses to win

&

Opex is poised to provide innovative solutions

Config Management Camp

Opex Software Thanks You

AUTOMATE IT. CREATE TIME

Config Management Camp