DRUPALCAMP OTTAWA

#DCO2019

AUTOMATE ALL THE THINGS!

Christopher Gervais - @ergonlogic

openplus.ca™

Coldfront Labs Inc.

Platinum Sponsors

consensus ENTERPRISES

consensus ENTERPRISES

- ★ Veteran open source programmers and sysadmins
- ★ Specializing in Drupal™ 8 and ÆGIR
- ★ Experts in end-to-end application lifecycle
- ★ Focus on social enterprises, non-profits, and public sector

Some of our Partnerships

consensus ENTERPRISES

NDI NATIONAL DEMOCRATIC INSTITUTE

Health Canada

RNAO Registered Nurses' Association of Ontario L'Association des infirmières et infirmiers autorisés de l'Ontario

the YMCA

SEAL OF THE CITY AND COUNTY OF SAN FRANCISCO

HERITAGE TORONTO PEOPLE.PLACES.EVENTS.

CENTRE FOR SOCIAL INNOVATION

Drupal™ ASSOCIATION

BETTER MAKE ROOM

SODER

rabble.ca NEWS FOR THE REST OF US

McGill

Canadian Blood Services

HARVARD UNIVERSITY

KOBELCO KOBE STEEL, LTD.

OCA Ontario Council of Agencies Serving

Corporation des bibliothécaires professionnels du Québec Corporation of Professional Librarians of Quebec

Public Works and Government Services Canada

GIVING gateway™

CAMDEN Living Excellence

CANADIAN PUBLIC HEALTH ASSOCIATION ASSOCIATION CANADIENNE DE SANTÉ PUBLIQUE

What we'll discuss

A brief history of cloud computing
How did we get into this mess?

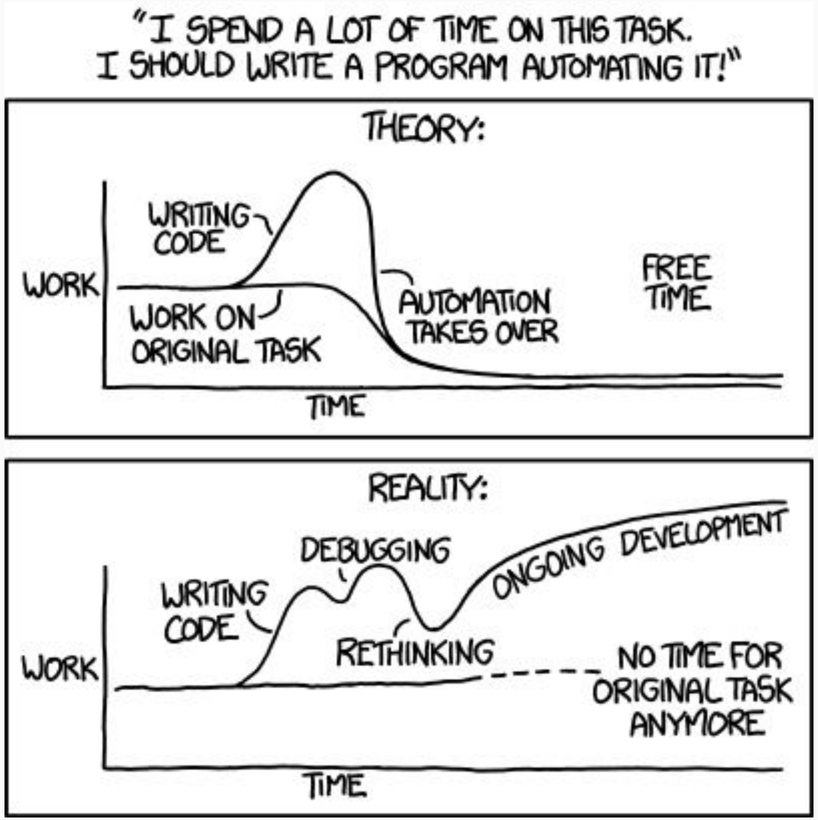
Principles and Practices of *infrastructure-as-code*
Why should we care?

How does Ansible support an *infrastructure-as-code* strategy?
Components and modules and providers; oh my!

Putting it all together
Demo time!

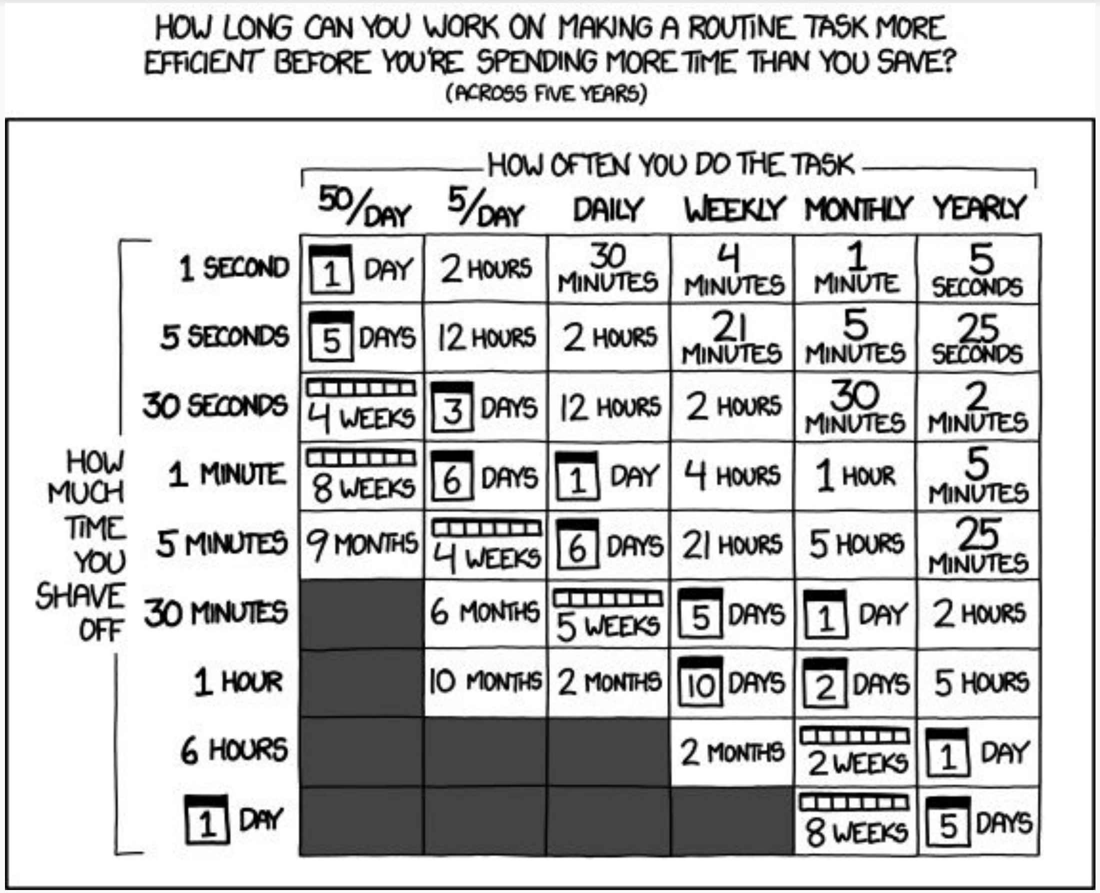
XKCD

… because, somehow, a webcomic provides the most succinct descriptions of the reality of automation.



consensus ENTERPRISES

You can never have too much *XKCD*!



HOW LONG CAN YOU WORK ON MAKING A ROUTINE TASK MORE EFFICIENT BEFORE YOU'RE SPENDING MORE TIME THAN YOU SAVE?
(ACROSS FIVE YEARS)

HOW MUCH TIME YOU SHAVE OFF	HOW OFTEN YOU DO THE TASK: 50/DAY	5/DAY	DAILY	WEEKLY	MONTHLY	YEARLY
1 SECOND	1 DAY	2 HOURS	30 MINUTES	4 MINUTES	1 MINUTE	5 SECONDS
5 SECONDS	5 DAYS	12 HOURS	2 HOURS	21 MINUTES	5 MINUTES	25 SECONDS
30 SECONDS	4 WEEKS	3 DAYS	12 HOURS	2 HOURS	30 MINUTES	2 MINUTES
1 MINUTE	8 WEEKS	6 DAYS	1 DAY	4 HOURS	1 HOUR	5 MINUTES
5 MINUTES	9 MONTHS	4 WEEKS	6 DAYS	21 HOURS	5 HOURS	25 MINUTES
30 MINUTES	■	6 MONTHS	5 WEEKS	5 DAYS	1 DAY	2 HOURS
1 HOUR	■	10 MONTHS	2 MONTHS	10 DAYS	2 DAYS	5 HOURS
6 HOURS	■	■	■	2 MONTHS	2 WEEKS	1 DAY
1 DAY	■	■	■	■	8 WEEKS	5 DAYS

consensus ENTERPRISES

A Brief history of Cloud Computing

Automate All the Things!

A brief history of cloud computing

- **Time-sharing**

  (government/academic)

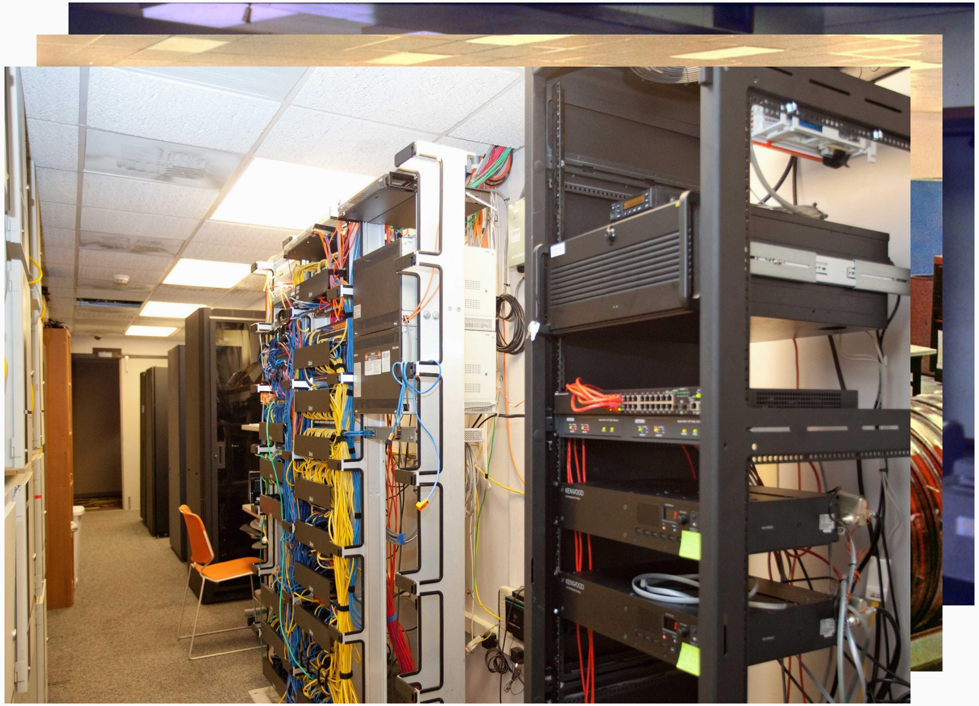
consensus ENTERPRISES

A brief history of cloud computing

- Time-sharing
  (government/academic)
- **Mainframes**
  (centralized/institutional)

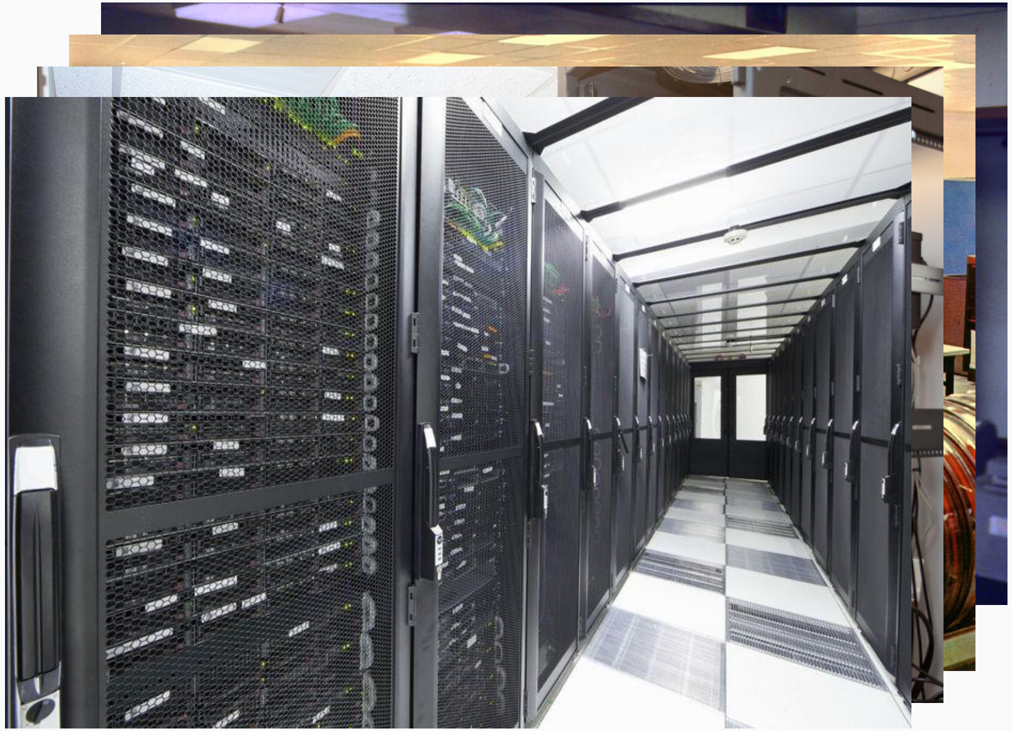
A brief history of cloud computing

- Time-sharing
  (government/academic)
- Mainframes
  (centralized/institutional)
- **Server rooms**
  (distributed/on-premise)



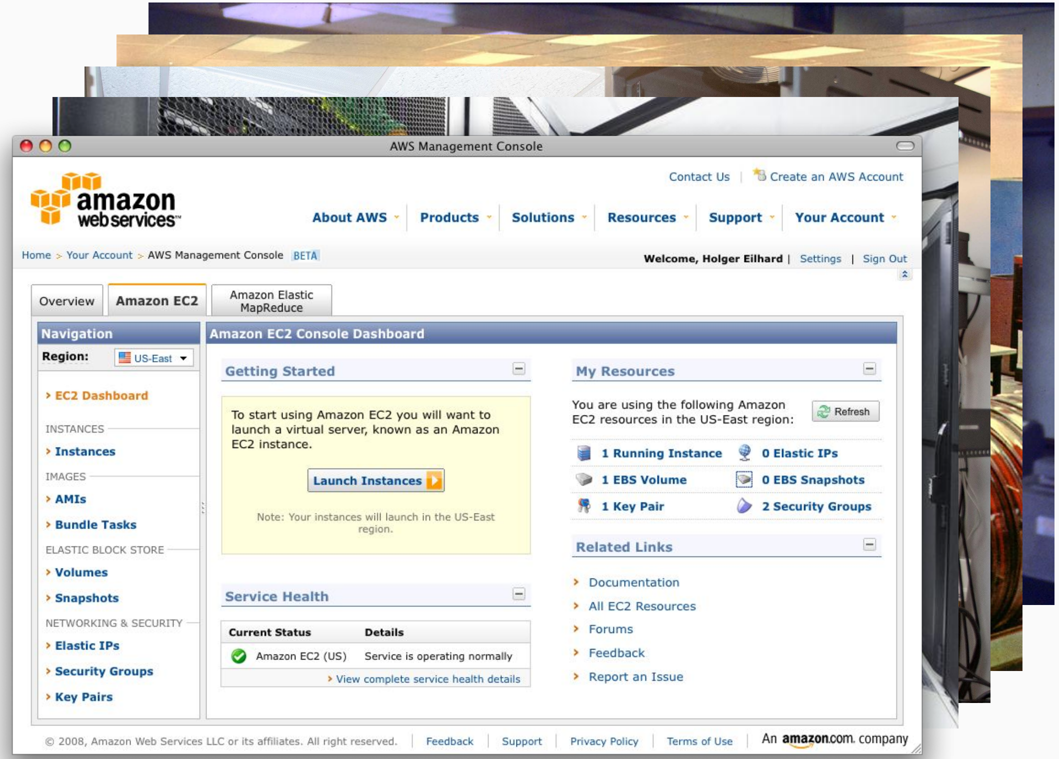
A brief history of cloud computing

- Time-sharing
  (government/academic)
- Mainframes
  (centralized/institutional)
- Server rooms
  (distributed/on-premise)
- **Datacenters**
  (co-location/hosted)



A brief history of cloud computing

- Time-sharing
  (government/academic)
- Mainframes
  (centralized/institutional)
- Server rooms
  (distributed/on-premise)
- Datacenters
  (co-location/hosted)
- **Cloud**
  (utility computing)



AWS Management Console

Contact Us | Create an AWS Account

amazon web services™

About AWS | Products | Solutions | Resources | Support | Your Account

Home > Your Account > AWS Management Console BETA

Welcome, Holger Eilhard | Settings | Sign Out

Overview | Amazon EC2 | Amazon Elastic MapReduce

Navigation

Region: US-East

- EC2 Dashboard

INSTANCES
- Instances

IMAGES
- AMIs
- Bundle Tasks

ELASTIC BLOCK STORE
- Volumes
- Snapshots

NETWORKING & SECURITY
- Elastic IPs
- Security Groups
- Key Pairs

Amazon EC2 Console Dashboard

Getting Started

To start using Amazon EC2 you will want to launch a virtual server, known as an Amazon EC2 instance.

Launch Instances

Note: Your instances will launch in the US-East region.

Service Health

Current Status	Details
✓ Amazon EC2 (US)	Service is operating normally

› View complete service health details

My Resources

You are using the following Amazon EC2 resources in the US-East region: Refresh

1 Running Instance	0 Elastic IPs
1 EBS Volume	0 EBS Snapshots
1 Key Pair	2 Security Groups

Related Links

- Documentation
- All EC2 Resources
- Forums
- Feedback
- Report an Issue

© 2008, Amazon Web Services LLC or its affiliates. All right reserved. | Feedback | Support | Privacy Policy | Terms of Use | An amazon.com. company

consensus ENTERPRISES

The era of cloud computing

Benefits	Challenges
Scalability	Controlling costs

The era of cloud computing

Benefits	Challenges
Scalability	Controlling costs
Flexibility	Increased complexity

The era of cloud computing

Benefits	Challenges
Scalability	Controlling costs
Flexibility	Increased complexity
Automation	Scarce expertise

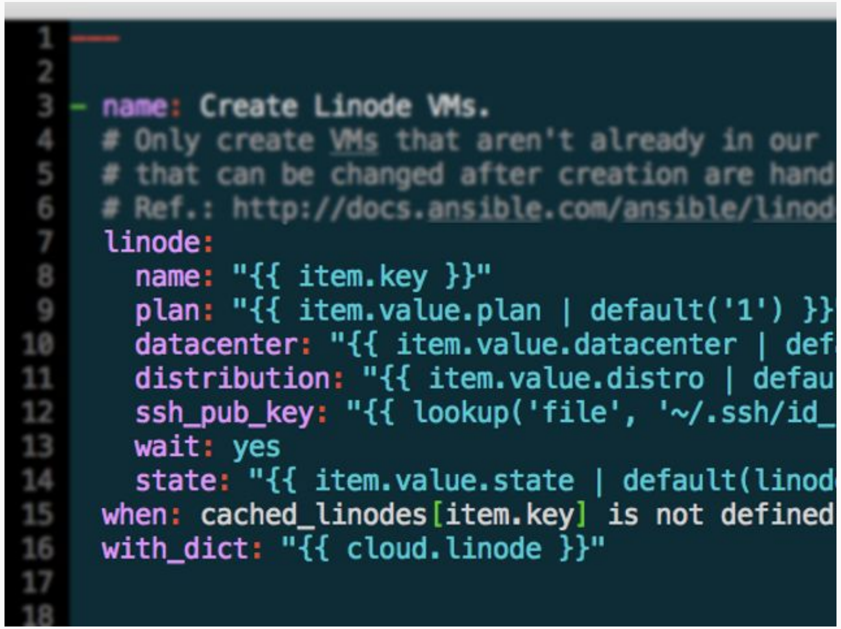
Principles and Practices of *infrastructure-as-code*

Automate All the Things!

consensus ENTERPRISES

Infrastructure-as-code Practices

- **Define resources in code**
  (avoid snowflake servers)



```
---

- name: Create Linode VMs.
  # Only create VMs that aren't already in our
  # that can be changed after creation are hand
  # Ref.: http://docs.ansible.com/ansible/linod
  linode:
    name: "{{ item.key }}"
    plan: "{{ item.value.plan | default('1') }}
    datacenter: "{{ item.value.datacenter | def
    distribution: "{{ item.value.distro | defau
    ssh_pub_key: "{{ lookup('file', '~/.ssh/id_
    wait: yes
    state: "{{ item.value.state | default(linod
  when: cached_linodes[item.key] is not defined
  with_dict: "{{ cloud.linode }}"
```

consensus ENTERPRISES

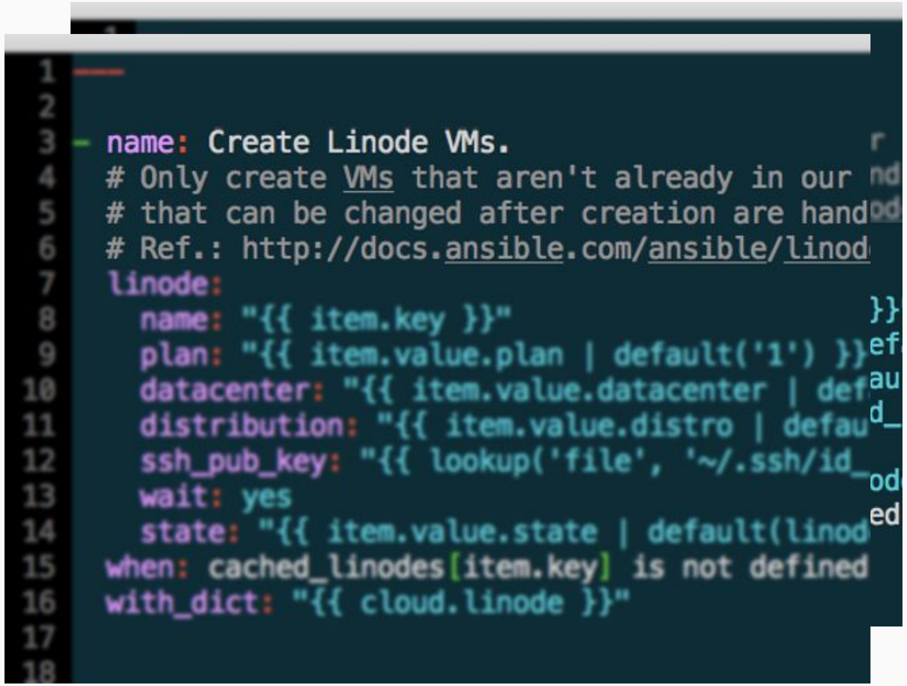
Infrastructure-as-code Practices

- Define resources in code

  (avoid snowflake servers)

- **Keep documentation inline**

  (self-documented systems)



```
---

- name: Create Linode VMs.
  # Only create VMs that aren't already in our
  # that can be changed after creation are hand
  # Ref.: http://docs.ansible.com/ansible/linod
  linode:
    name: "{{ item.key }}"
    plan: "{{ item.value.plan | default('1') }}
    datacenter: "{{ item.value.datacenter | def
    distribution: "{{ item.value.distro | defau
    ssh_pub_key: "{{ lookup('file', '~/.ssh/id_
    wait: yes
    state: "{{ item.value.state | default(linod
  when: cached_linodes[item.key] is not defined
  with_dict: "{{ cloud.linode }}"
```

consensus ENTERPRISES

Infrastructure-as-code Practices

- Define resources in code

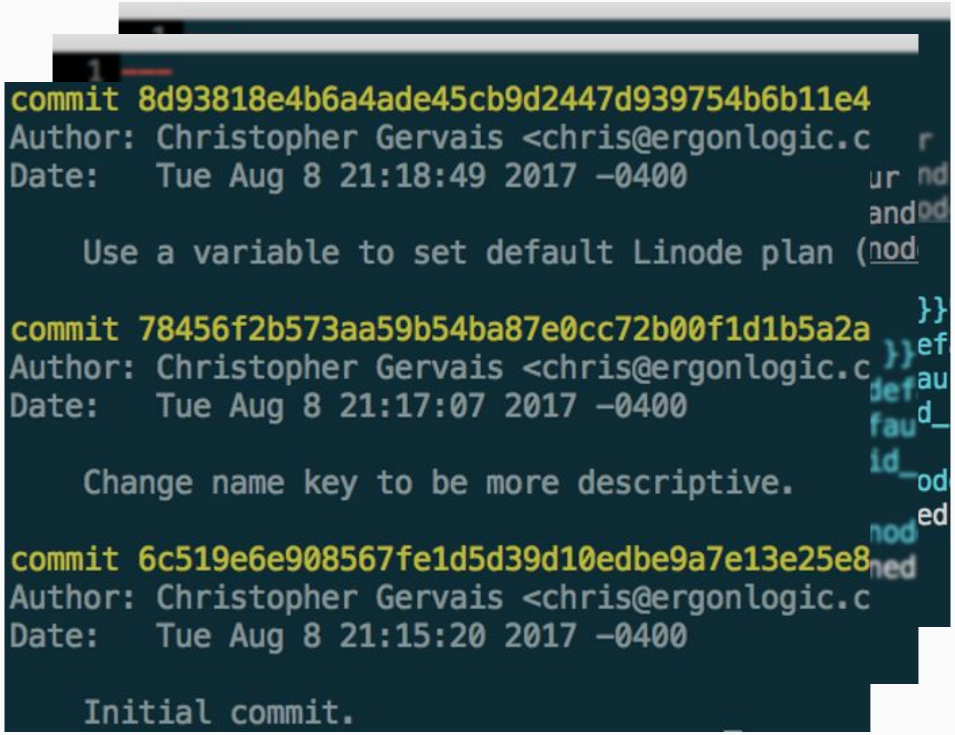
  (avoid snowflake servers)

- Keep documentation inline

  (self-documented systems)

- **Version-control everything**

  (audit trail and reproducible builds)

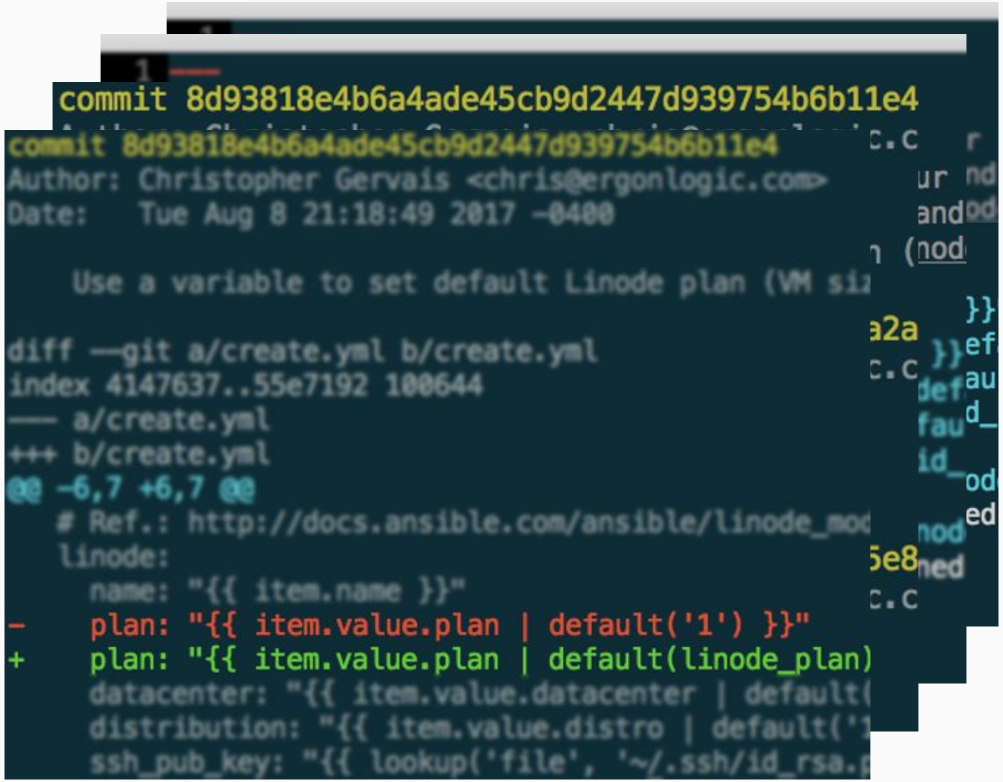


```
commit 8d93818e4b6a4ade45cb9d2447d939754b6b11e4
Author: Christopher Gervais <chris@ergonlogic.c
Date:   Tue Aug 8 21:18:49 2017 -0400

    Use a variable to set default Linode plan (nod

commit 78456f2b573aa59b54ba87e0cc72b00f1d1b5a2a
Author: Christopher Gervais <chris@ergonlogic.c
Date:   Tue Aug 8 21:17:07 2017 -0400

    Change name key to be more descriptive.

commit 6c519e6e908567fe1d5d39d10edbe9a7e13e25e8
Author: Christopher Gervais <chris@ergonlogic.c
Date:   Tue Aug 8 21:15:20 2017 -0400

    Initial commit.
```

consensus ENTERPRISES

Infrastructure-as-code Practices

- Define resources in code
  (avoid snowflake servers)
- Keep documentation inline
  (self-documented systems)
- Version-control everything
  (audit trail and reproducible builds)
- **Make small changes**
  (easier rollbacks)



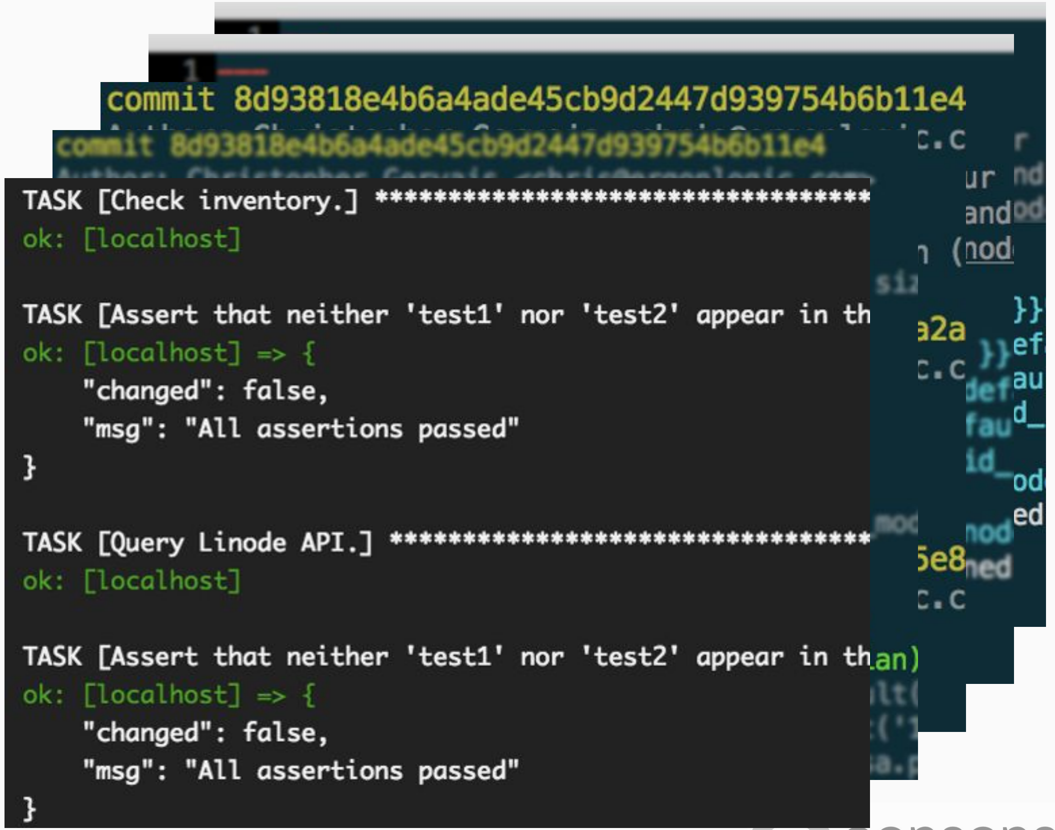
```
commit 8d93818e4b6a4ade45cb9d2447d939754b6b11e4
```

```
commit 8d93818e4b6a4ade45cb9d2447d939754b6b11e4
Author: Christopher Gervais <chris@ergonlogic.com>
Date:   Tue Aug 8 21:18:49 2017 -0400

    Use a variable to set default Linode plan (VM siz

diff --git a/create.yml b/create.yml
index 4147637..55e7192 100644
--- a/create.yml
+++ b/create.yml
@@ -6,7 +6,7 @@
   # Ref.: http://docs.ansible.com/ansible/linode_mod
   linode:
     name: "{{ item.name }}"
-    plan: "{{ item.value.plan | default('1') }}"
+    plan: "{{ item.value.plan | default(linode_plan)
     datacenter: "{{ item.value.datacenter | default(
     distribution: "{{ item.value.distro | default('1
     ssh_pub_key: "{{ lookup('file', '~/.ssh/id_rsa.p
```

consensus ENTERPRISES

Infrastructure-as-code Practices

- Define resources in code
  (avoid snowflake servers)
- Keep documentation inline
  (self-documented systems)
- Version-control everything
  (audit trail and reproducible builds)
- Make small changes
  (easier rollbacks)
- **Test continuously**
  (fail early)



```
commit 8d93818e4b6a4ade45cb9d2447d939754b6b11e4
commit 8d93818e4b6a4ade45cb9d2447d939754b6b11e4
```

```
TASK [Check inventory.] ********************************
ok: [localhost]

TASK [Assert that neither 'test1' nor 'test2' appear in th
ok: [localhost] => {
    "changed": false,
    "msg": "All assertions passed"
}

TASK [Query Linode API.] *******************************
ok: [localhost]

TASK [Assert that neither 'test1' nor 'test2' appear in th
ok: [localhost] => {
    "changed": false,
    "msg": "All assertions passed"
}
```

consensus ENTERPRISES

How does Ansible support an *infrastructure-as-code* strategy?

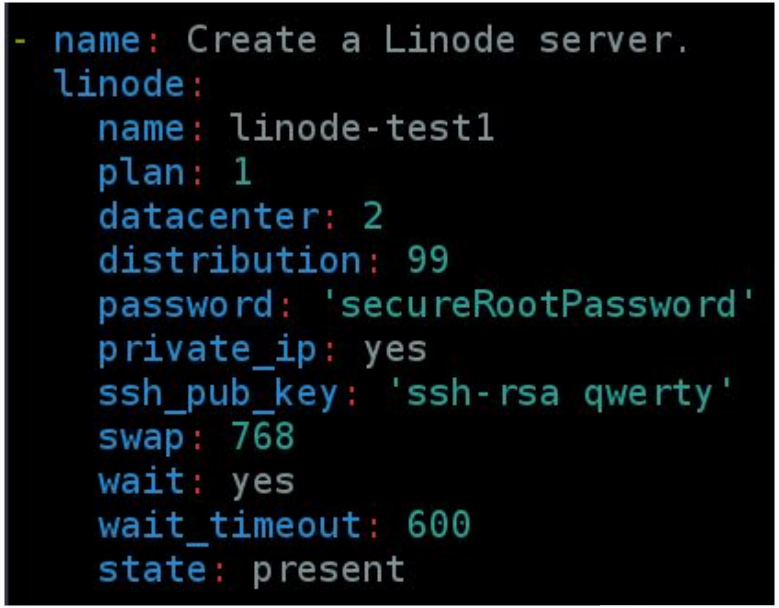
Automate All the Things!

consensus ENTERPRISES

How does Ansible support an *infrastructure-as-code* strategy?

Ansible allows us to define infrastructure components in a simple YAML syntax.

These files can, in turn, be committed into version control, and thus handled as software.



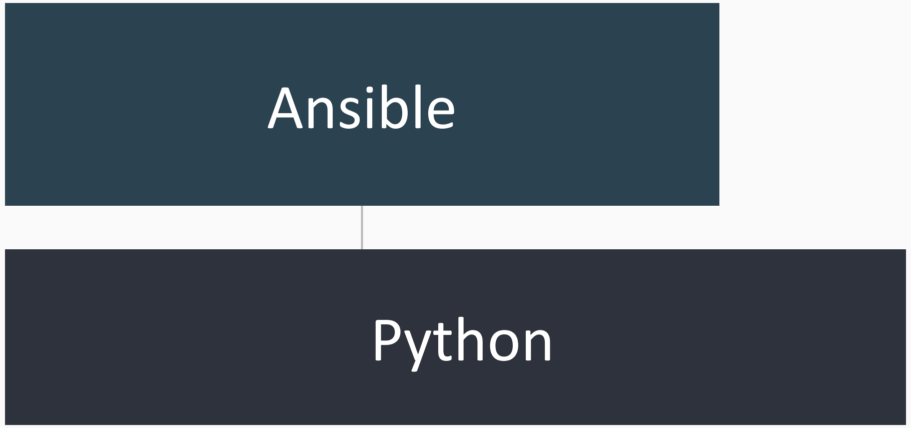
```yaml
- name: Create a Linode server.
  linode:
    name: linode-test1
    plan: 1
    datacenter: 2
    distribution: 99
    password: 'secureRootPassword'
    private_ip: yes
    ssh_pub_key: 'ssh-rsa qwerty'
    swap: 768
    wait: yes
    wait_timeout: 600
    state: present
```

Components

Custom *infrastructure-as-code* configuration depends on Ansible, roles and modules, which in turn depend on various Python libraries.

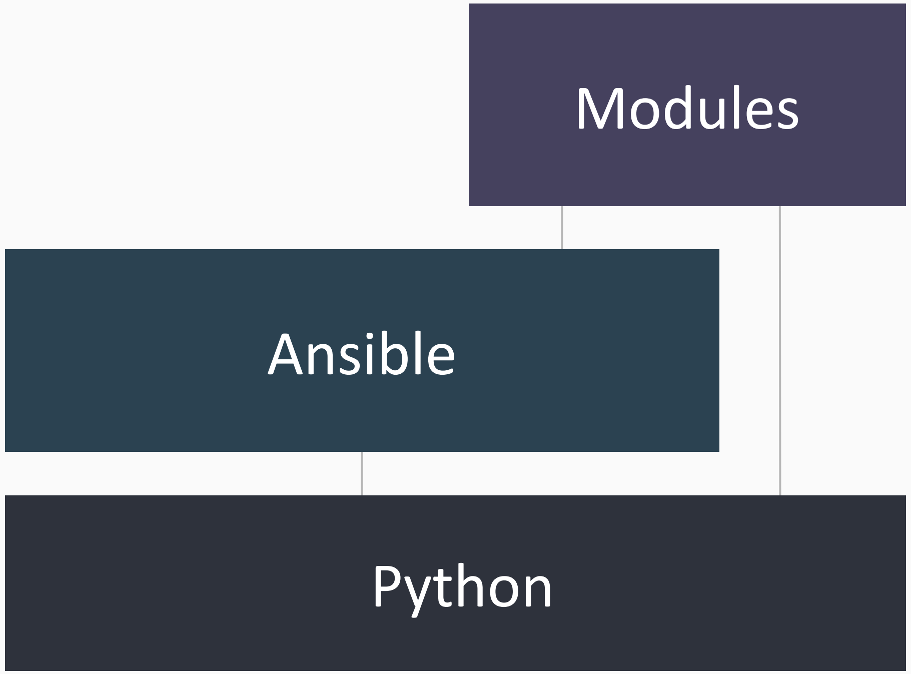
Components

Custom *infrastructure-as-code* configuration depends on Ansible, roles and modules, which in turn depend on various Python libraries.



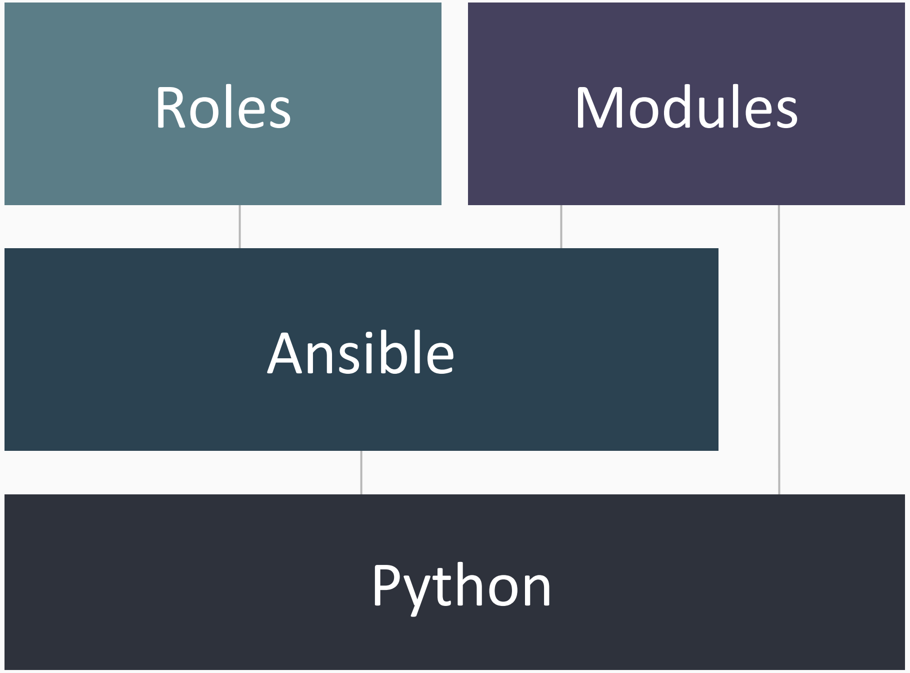
Components

Custom *infrastructure-as-code* configuration depends on Ansible, roles and modules, which in turn depend on various Python libraries.



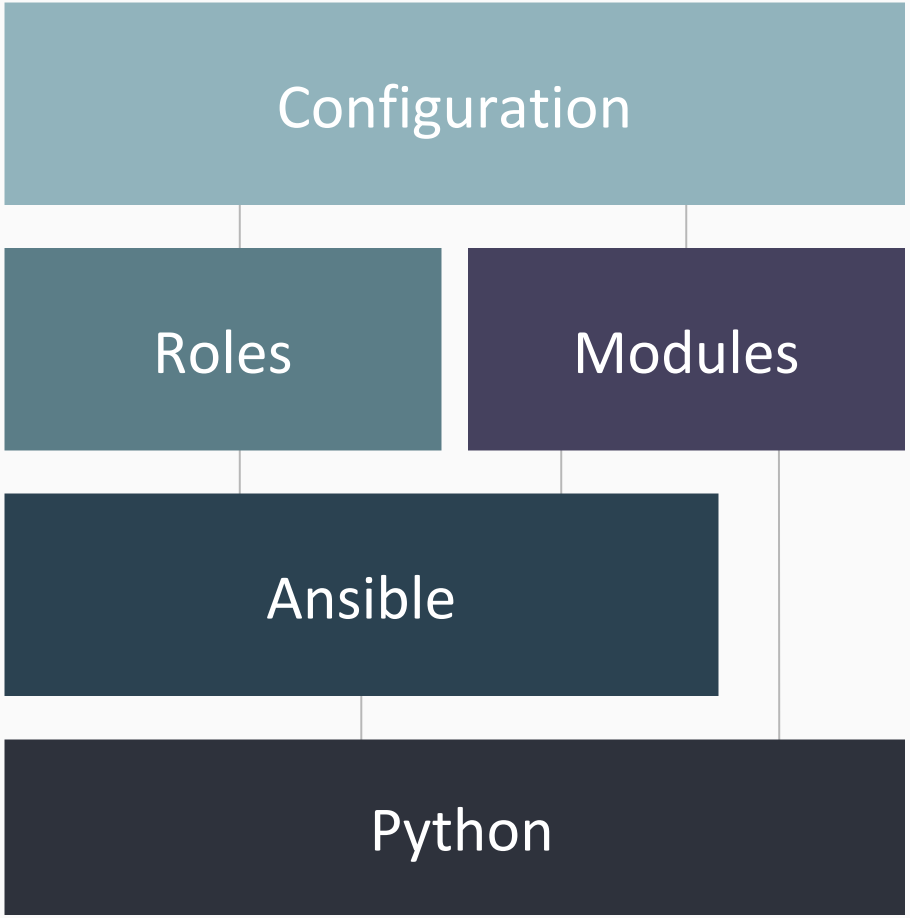
Components

Custom *infrastructure-as-code* configuration depends on Ansible, roles and modules, which in turn depend on various Python libraries.



Components

Custom *infrastructure-as-code* configuration depends on Ansible, roles and modules, which in turn depend on various Python libraries.



Providers vs. Modules

What’s the difference?

Providers: A cloud provider is (generally) a company that offers components of cloud computing (e.g. , IaaS).

Modules: These task plugins interact with providers’ APIs to create and manage various resources.

920+

That’s how many different cloud modules Ansible supports out-of-the-box. These range across 40+ cloud providers, from Amazon to XenServer.

consensus ENTERPRISES

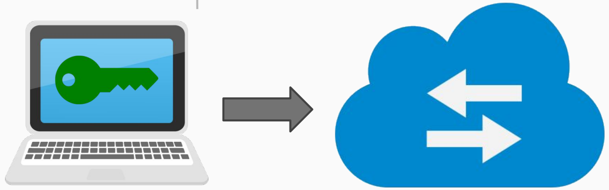
Authentication and Authorization

Authentication and Authorization

Step 1

User triggers

API call



consensus ENTERPRISES

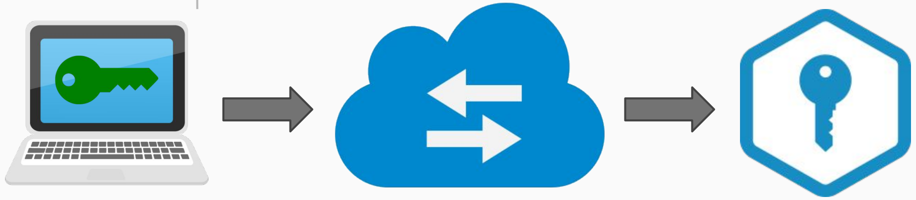
Authentication and Authorization

Step 1

User triggers API call

Step 2

API authenticates user's credentials



consensus ENTERPRISES

Authentication and Authorization

Step 1

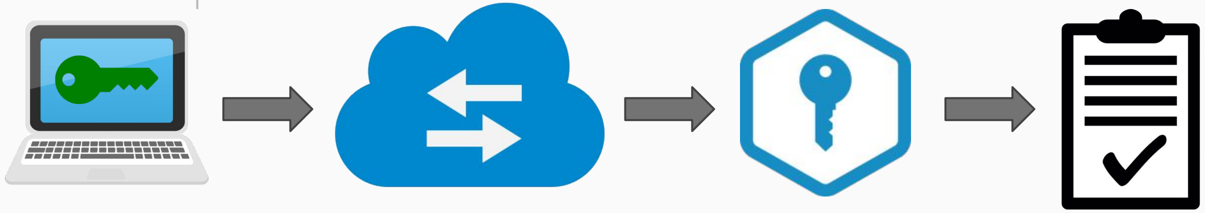
User triggers API call

Step 2

API authenticates user's credentials

Step 3

API checks that user is authorized for task(s)



Authentication and Authorization

Step 1

User triggers API call

Step 2

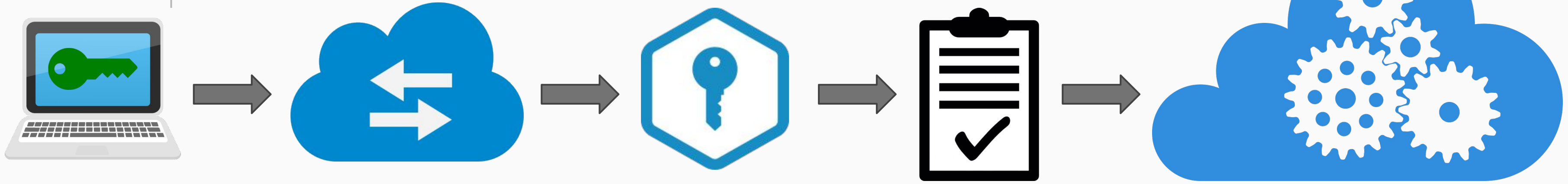
API authenticates user's credentials

Step 3

API checks that user is authorized for task(s)

Step 4

API executes task(s) in cloud infrastructure



consensus ENTERPRISES

Putting It All Together

Automate All the Things!

DRUPALCAMP OTTAWA

#DCO2019

QUESTIONS?

Christopher Gervais - @ergonlogic

openplus.ca™

Coldfront Labs Inc.

Platinum Sponsors

consensus ENTERPRISES