UNIVERSITY OF SURREY

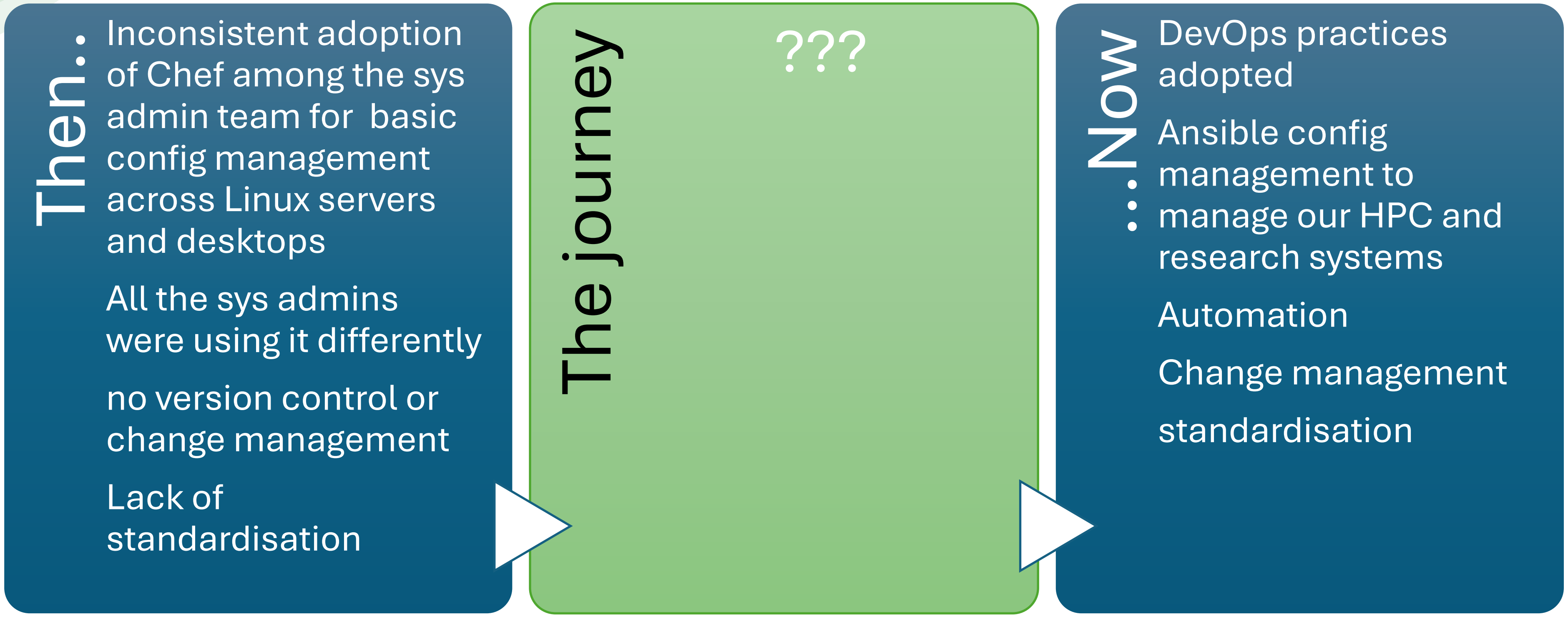
Dean Roe – Head of Research Computing

University of Surrey

Devops & Config management – Learning Journey

ACIT Hub

Our learning Journey



Then…

- Inconsistent adoption of Chef among the sys admin team for basic config management across Linux servers and desktops
- All the sys admins were using it differently
- no version control or change management
- Lack of standardisation

The journey

???

Now…

- DevOps practices adopted
- Ansible config management to manage our HPC and research systems
- Automation
- Change management
- standardisation

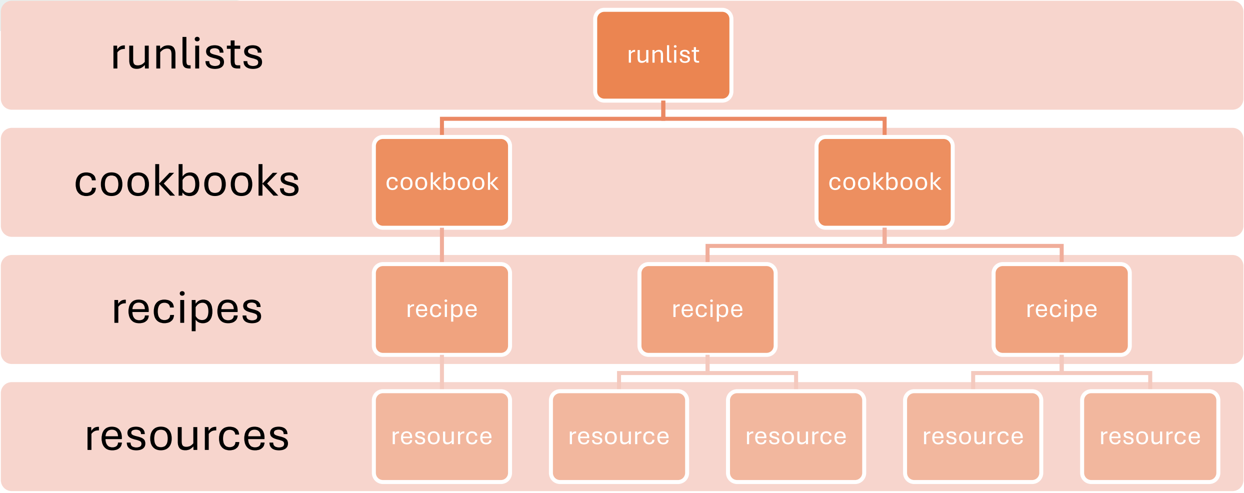
A bit about Chef

Chef is an infrastructure automation platform that treats system configurations as code. It ensures consistent, repeatable deployments across environments

🔑 **Key Features:**

- **Infrastructure as Code**: Define and version system states using code. Ruby based.
- **Policy Enforcement & drift correction**: Automatically converges systems to desired configurations. Detects and fixes deviations from intended setup (AKA Desired state configuration)
- **Cross-Platform Support**: Works across Linux, Windows, macOS, and cloud platforms.
- **Centralised Control**: One Chef server manages all nodes.
- **Requires an agent to be installed on the clients:** called chef-client
- **Pull based system:** agent on clients pulls latest run list from the server and runs chef-client on the client node at defined regular interval.
- **Code is written in Ruby**

Recipes and Cookbooks



- Runlist defines list of cookbooks to run on node
- Cookbooks are collections of recipes and other files that describe how to configure and manage a system.
- A recipe is a Ruby-based script that defines a set of resources and the order in which they should be applied to configure a system.
- Resources are the building blocks of recipes. They represent the desired state of an element in your system, such as a file, package, or service.

Resources

A typical resource in chef

```
file '/var/www/customers/public_html/index.php' do
  content '<html>This is a placeholder for the home page.</html>’
  mode '0755’
  owner 'web_admin’
  group 'web_admin'
end
```

This is **idempotent**

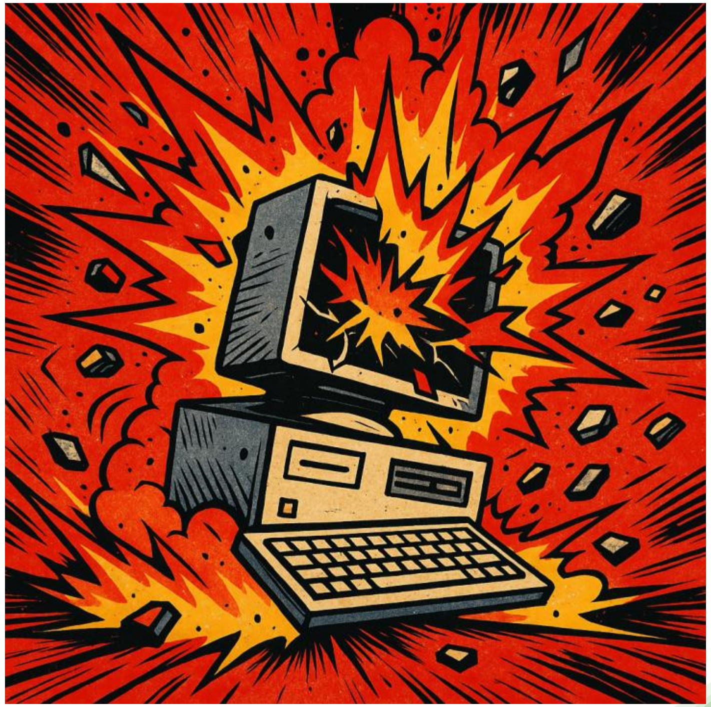
In code, idempotency means that applying an operation multiple times has the same effect as applying it just once. Essentially, if you repeat the operation, you don't get any additional or unexpected side effects beyond the initial execution.

Lesson 1 – use the tools properly.

Learn the tool and understand it!

How was chef being misused?

- ✗ Lots of pure Ruby
- ✗ Lots of hard coded variables (attributes in chef)
- ✗ Lots of resources set to “skip on failure”
- ✗ Monolithic cookbooks
- ✗ No consistency of code writing between authors
- ✗ No version control…



Lesson 2 – git is your friend

Version control and change management is essential.

- We set up Gitlab
- Put the chef cookbooks under version control
- Force the sys admins into that workflow
- Eventually this led to CI/CD pipelines
  - implement lint checkers and testing enforcing some standards and consistency in the code.
  - Automated deployment of update code

Lesson 3: platform lock-in or vendor lock-in is very risky

- Chef started raising their prices
- We has increasing numbers of nodes under Chef management
- Becoming financially unsustainable

We needed an exit strategy / plan B!

(it also would turn out that Chef would be sold later on)

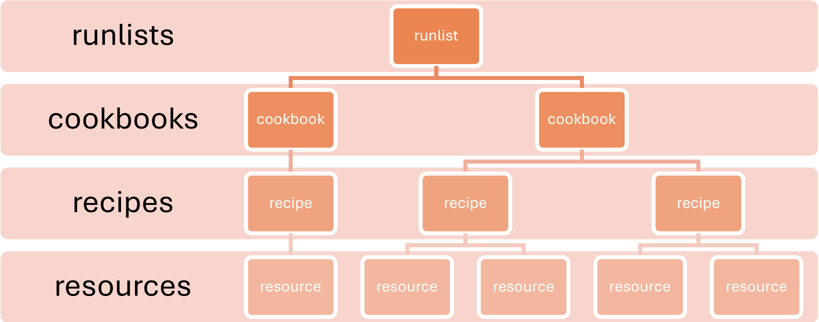
Ansible to the rescue

Ansible is an infrastructure automation platform that treats system configurations as code. It ensures consistent, repeatable deployments across environments

🔑 **Key Features:**

- **Infrastructure as Code**: Define and version system states using code. Ruby based.
- **Policy Enforcement & drift correction**: Automatically converges systems to desired configurations. Detects and fixes deviations from intended setup (AKA Desired state configuration)
- **Cross-Platform Support**: Works across Linux, Windows, macOS, and cloud platforms.
- **Centralised Control**: Node can all be managed from a central server
- **Agentless:** requires python and SSH on the endpoint to be configured
- **Push based system:** if using a server such as AWX or Ansible automation platform the solution is push based by default.
- **Code is written in YAML and is python based.**
- **Its free!** (mostly)

Ansible v Chef



runlists

runlist

cookbooks

cookbook

cookbook

recipes

recipe

recipe

recipe

resources

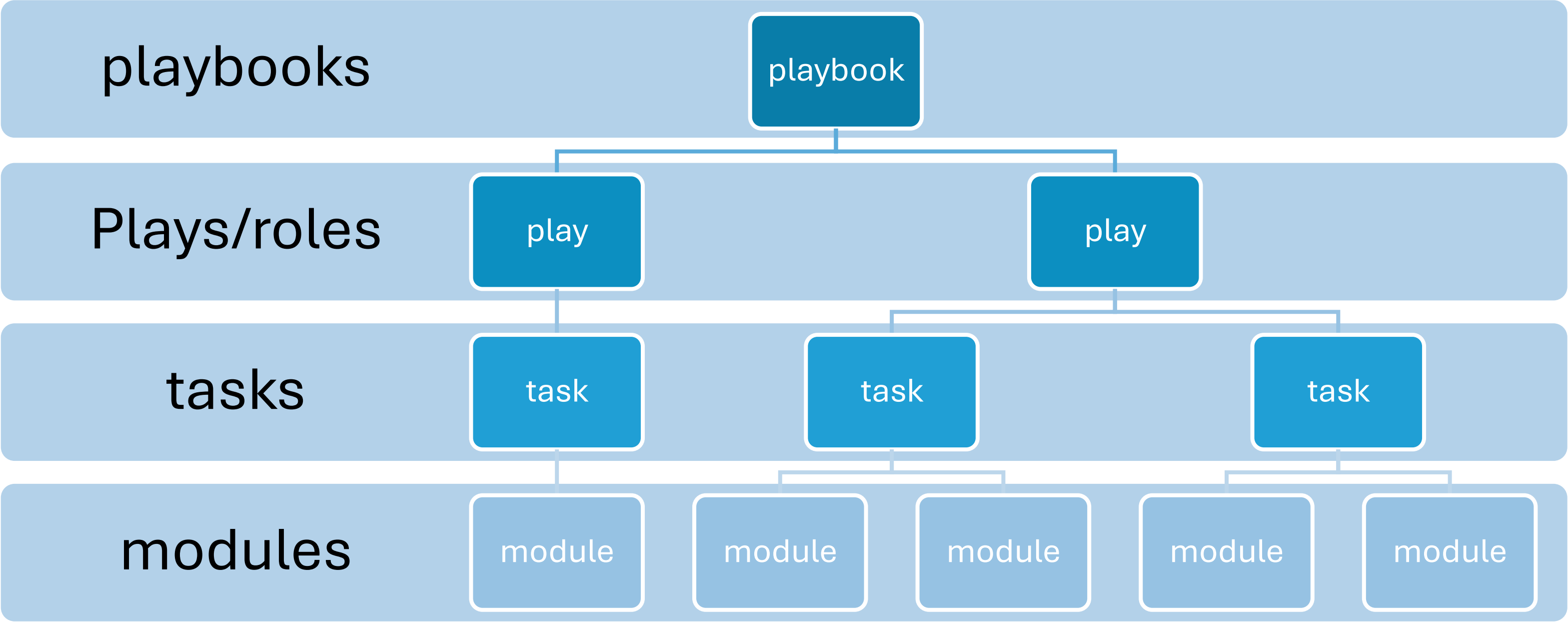
resource

resource

resource

resource

resource



playbooks

playbook

Plays/roles

play

play

tasks

task

task

task

modules

module

module

module

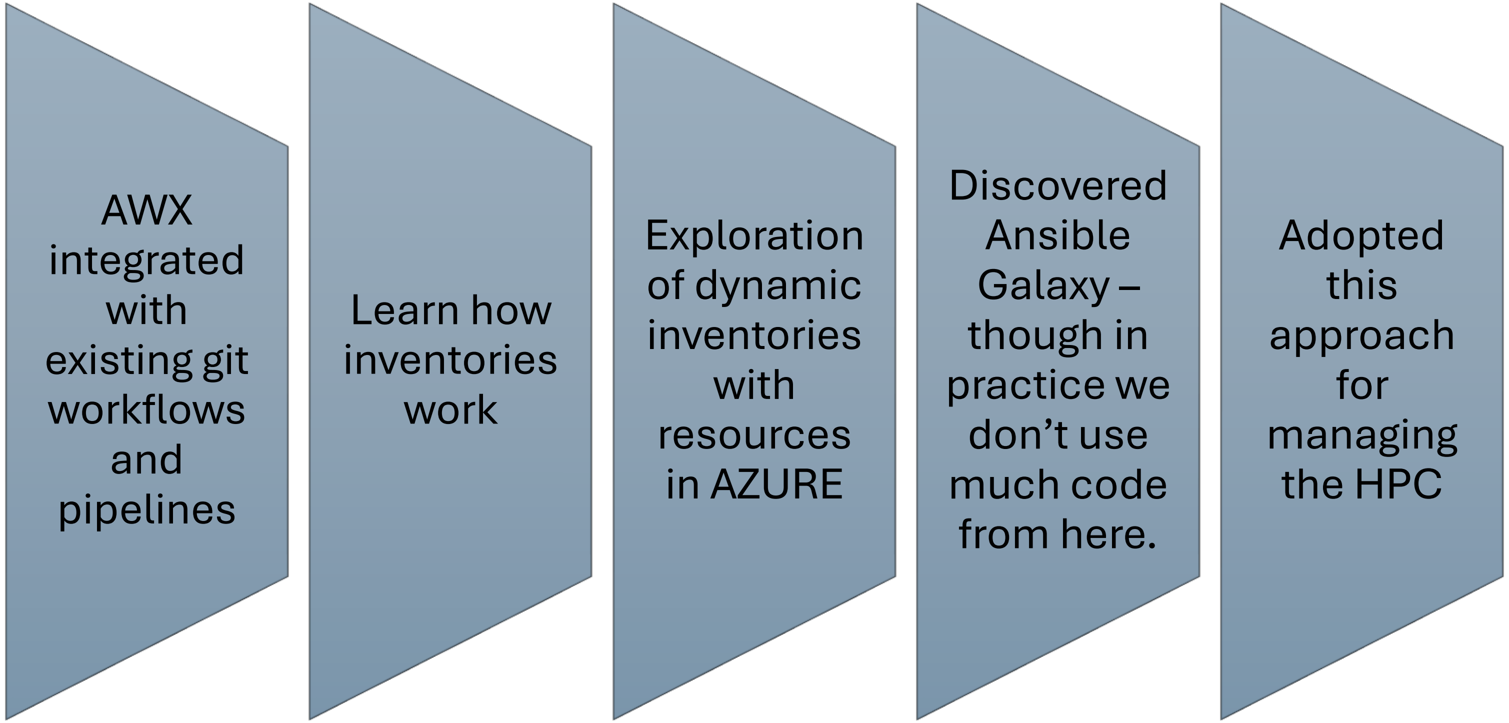
module

module

```
- name: Recursively change ownership of a directory
ansible.builtin.file:
path: /etc/foo
state: directory
recurse: yes
owner: foo
group: foo
```

ACIT Hub

What happened next?



AWX integrated with existing git workflows and pipelines

Learn how inventories work

Exploration of dynamic inventories with resources in AZURE

Discovered Ansible Galaxy – though in practice we don't use much code from here.

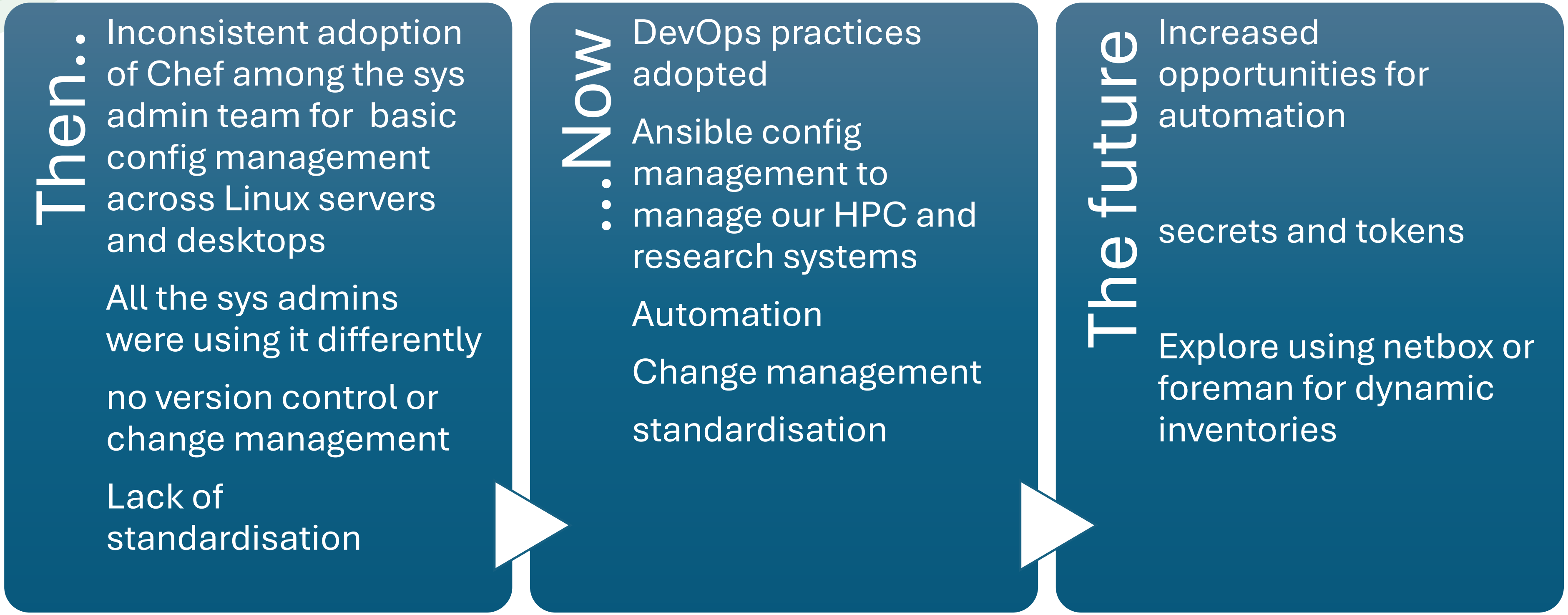
Adopted this approach for managing the HPC

ANSIBLE

UNIVERSITY OF SURREY

ACIT Hub

Our learning Journey



Then…

- Inconsistent adoption of Chef among the sys admin team for basic config management across Linux servers and desktops
- All the sys admins were using it differently
- no version control or change management
- Lack of standardisation

…Now

- DevOps practices adopted
- Ansible config management to manage our HPC and research systems
- Automation
- Change management
- standardisation

The future

- Increased opportunities for automation
- secrets and tokens
- Explore using netbox or foreman for dynamic inventories

UNIVERSITY OF SURREY