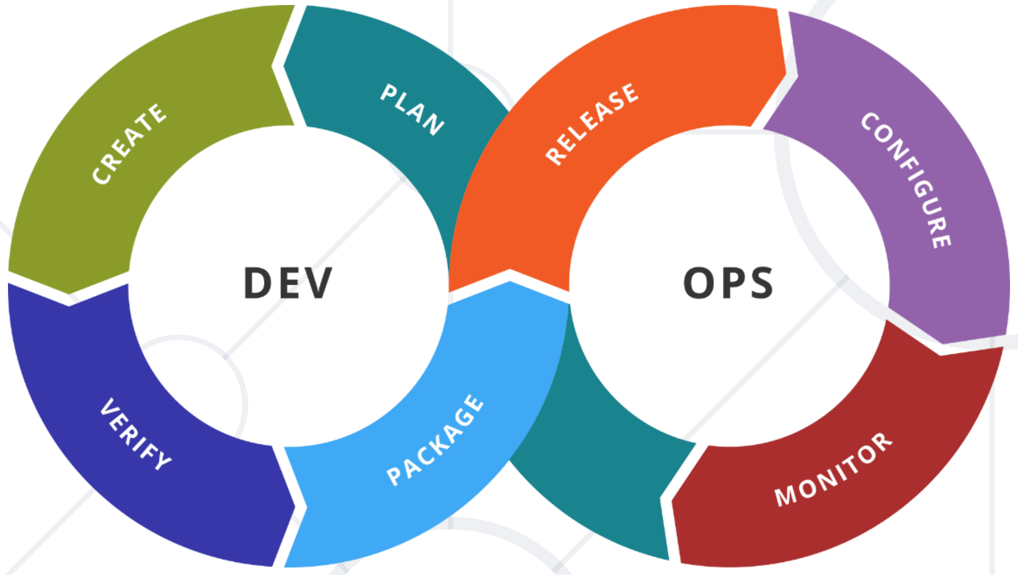
Chef

Introduction and Basic Techniques



SoftUni Team

Technical Trainers

Software University

SoftUni

Software University

https://softuni.org

You Have Questions?

Software University

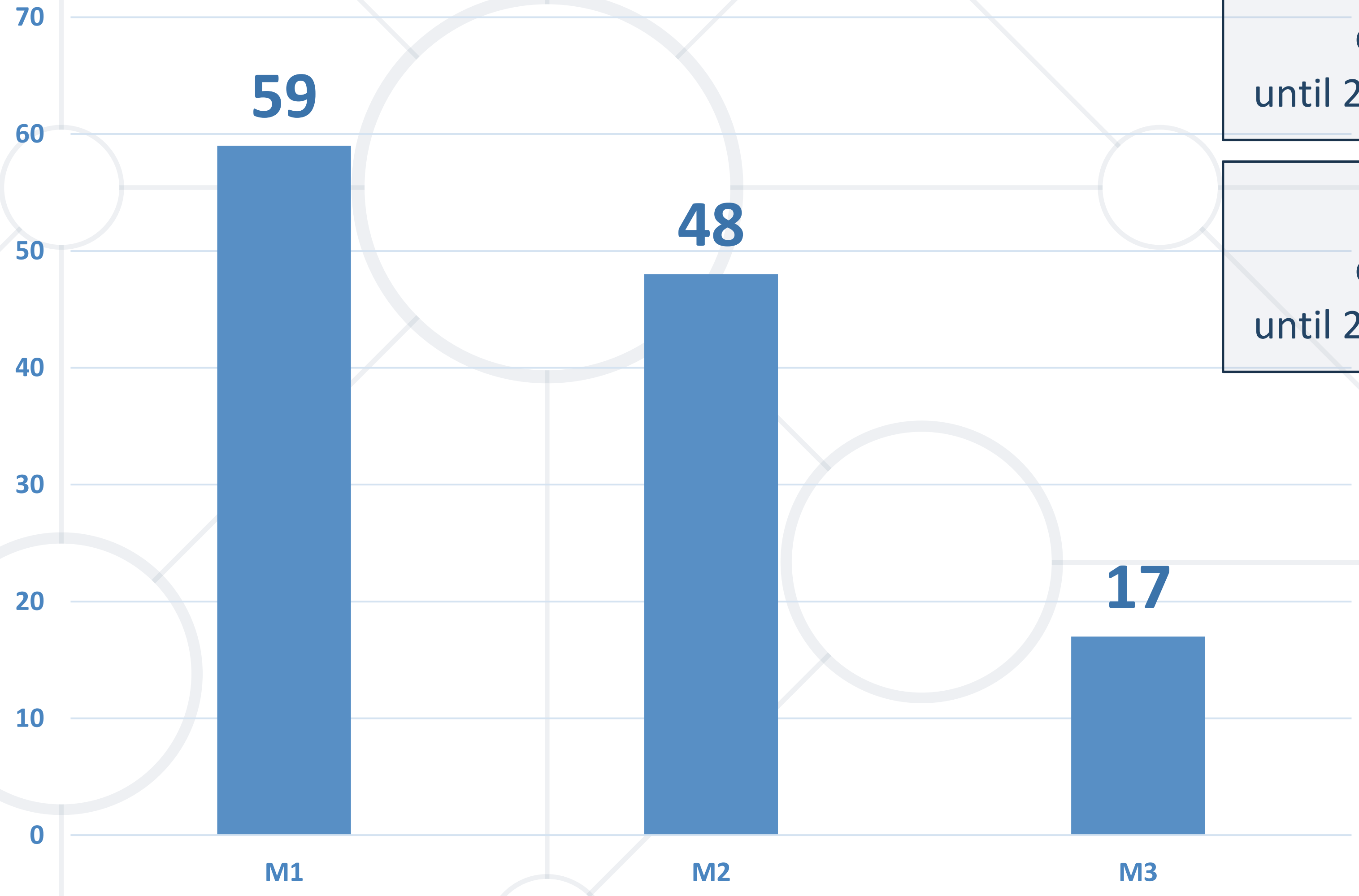
sli.do

#DevOps-23

facebook.com/groups

/DevOpsInfrastructureandConfigManagementApril2023

Homework Progress



Solutions for M3
can be submitted
until 23:59 on 26.04.2023

Solutions for M4
can be submitted
until 23:59 on 03.05.2023

Previous Module (M3)

Quick Overview

What We Covered

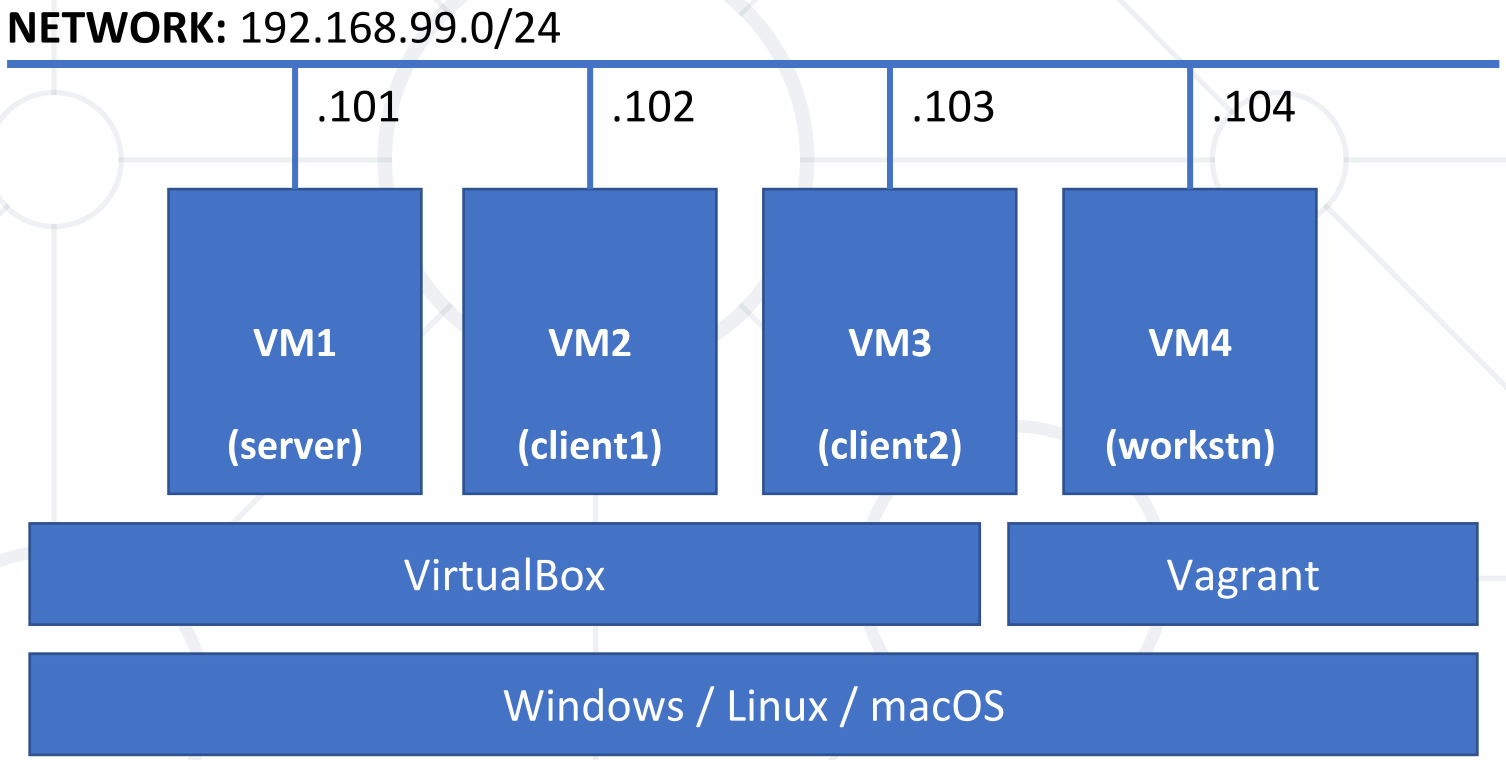
1. Introduction to Salt
   - Salt introduction and architecture
   - Installation and basic scenarios
2. Working with Salt
   - Basic scenarios and files
   - Pillars, filtering, and beacons
3. Advanced Salt
   - Custom modules

This Module (M4)

Topics and Lab Infrastructure

Table of Contents

Lab Infrastructure



Chef 101

Introduction. Architecture. Installation

Chef Products

- **Chef Infra**
  - Configure and manage infrastructure
- **Chef InSpec**
  - Build and run profiles for compliance automation
- **Chef Habitat**
  - Define, package, and deliver applications
- **Chef Automate**
  - Dashboards for operational visibility

Chef Enterprise Automation Stack (Chef EAS)

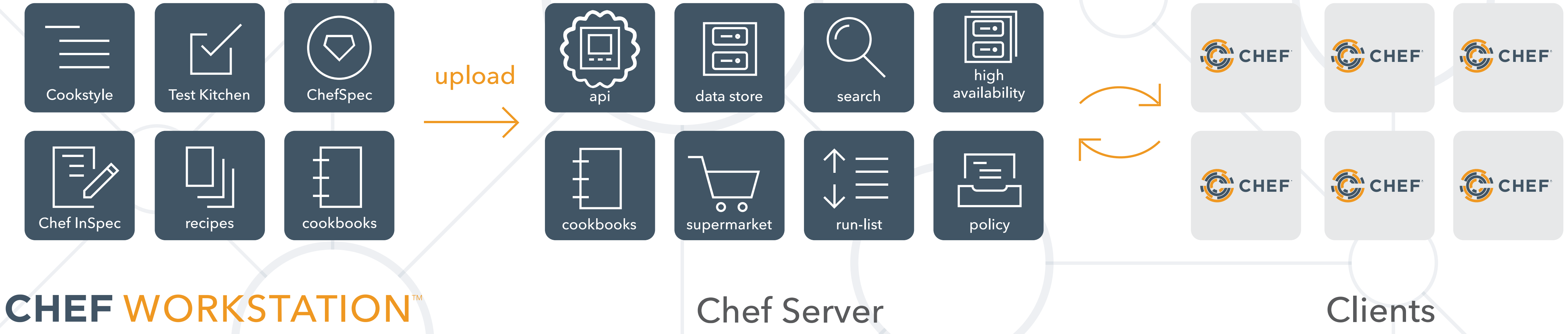
Introduction (Chef Infra)

- Solution for infrastructure and application automation
- Instructions are written in **Ruby DSL**
- Master-agent model, pull-based approach
- Server portion can be installed only on Linux
- Management part can be installed on Linux/macOS/Windows
- Client can be installed on Linux/Unix/Windows

Components and Workflow (1)

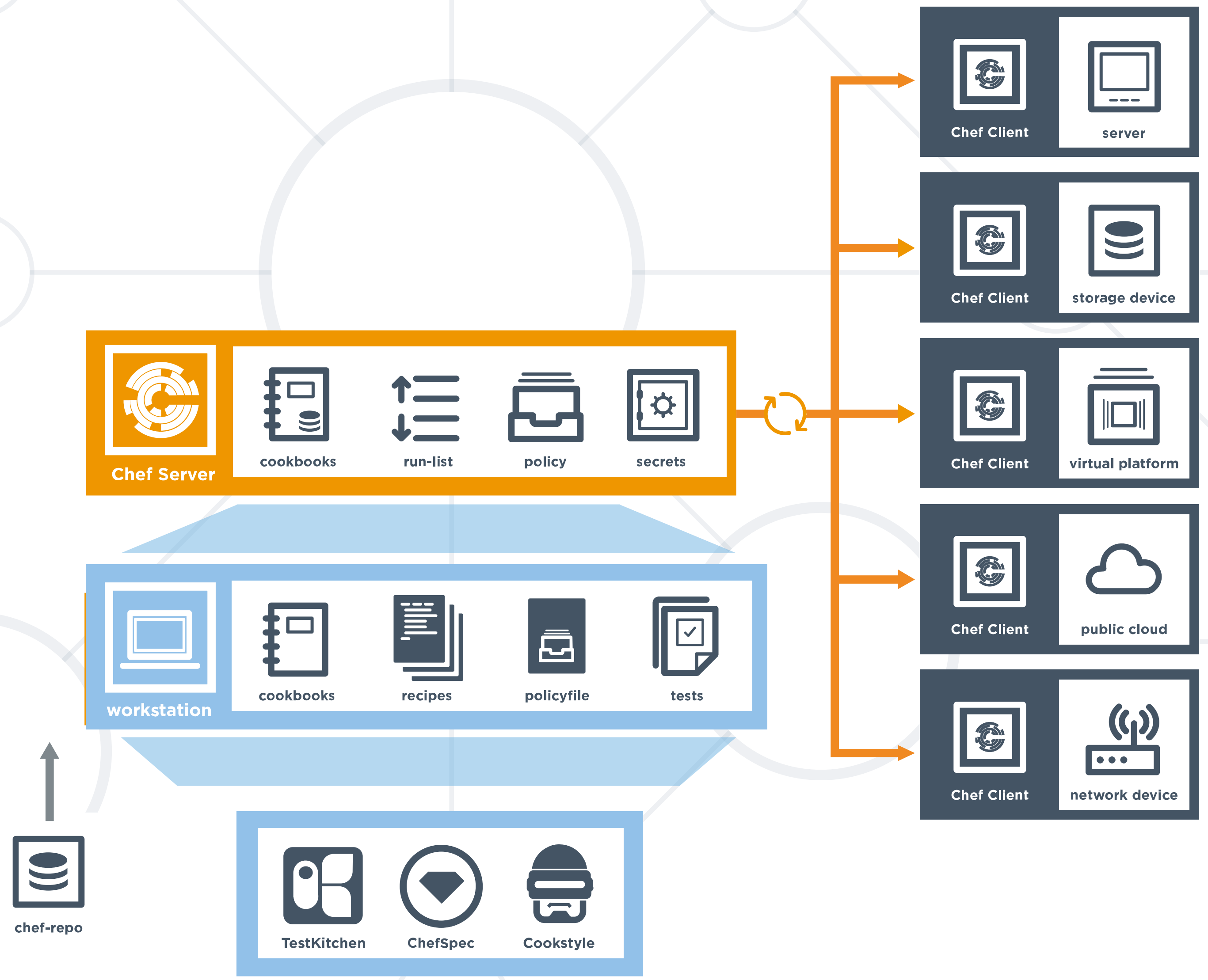
- **Chef workstation** is the point where users can author and test cookbooks and interact with the Chef server
- **Chef client nodes** are the machines that are managed by Chef
- **Chef client** is installed on each node and is used to configure the node to its desired state
- **Chef server** acts as a hub for cookbooks, policies, and metadata
- Nodes use the Chef client to ask the Chef server for configuration details, such as recipes, templates, and file distributions

Components and Workflow (2)



https://docs.chef.io/chef_overview/

Components and Workflow (3)



https://docs.chef.io/chef_overview/

Artifacts

- **Resource** is a statement of configuration policy that describes the desired state for a configuration item, declares the steps needed, artifact type, and set of properties
- **Recipes** specify the resources to use and the order in which they are to be applied
- **Cookbooks** are the fundamental unit of configuration and policy distribution. They define a scenario and contain everything that is required to support it

Practice: Chef 101

Live Demonstration in Class

DevOps

Chef 102

Resource (1)

- A statement of configuration policy
- Describes the desired state for a configuration item
- Declares the steps needed to bring that item to the desired state
- Specifies a resource type (such as package, template, or service
- Lists additional details (also known as resource properties), as necessary
- Resources are grouped into recipes, which describe working configurations

https://docs.chef.io/resource/

Resource (2)

- It is a Ruby block which has four components
  - a type
  - a name
  - one (or more) properties with values
  - one (or more) actions

```
type 'name' do
  attribute 'value'
  action :type_of_action
end
```

https://docs.chef.io/resource/

Recipes

- Collection of resources
- Helper code is added around resources using Ruby, when needed
- Must define everything that is required to configure part of a system
- Must be stored in a cookbook and may be included in another recipe
- May use the results of a search query and read the contents of a data
- May have a dependency on one (or more) recipes
- Must be added to a run-list before it can be used by Chef Infra Client
- Always executed in the same order as listed in a run-list

https://docs.chef.io/recipes/

Cookbooks

- Fundamental unit of configuration and policy distribution
- Define a scenario and contains everything that is required to support that scenario
- Recipes that specify which resources to use, as well as the order in which they are to be applied
- Attribute values, which allow environment-based configurations such as dev or production
- Custom Resources for extending Chef Infra beyond the built-in resources
- Files and Templates for distributing information to systems

https://docs.chef.io/cookbooks/

Cookbooks Folder Structure *

- **attributes** store additional settings and data in one or more files
- **files** store files that can be later distributed on nodes
- **recipes** store recipes each in separate file
- **templates** used to insert dynamic content to files
- **libraries** store Ruby code for new classes or extensions
- **metadata.rb** contains information about the cookbook

https://docs.chef.io/cookbooks/

Attributes

- An attribute is a specific detail about a node
- Determine the value that is applied to a node during run
- Used to understand the current state of a node, the state it was at the end of the previous run, and the state that it should has after the current run
- Defined by nodes, passed on the command line, cookbooks and policy files
- Attributes list is built during every run

https://docs.chef.io/attributes/

Attribute Types

- **default** is with lowest precedence and reset on every run
- **force_default** guarantees that a cookbook defined attribute will take precedence over an attribute set by role or environment
- **normal** is a setting that persists in the node object
- **override** is reset on every run and can be specified in recipe or attribute file for a role or environment
- **force_override** ensure that a cookbook defined attribute will take precedence over an override attribute set by role or environment
- **automatic** store data identified by Ohai at the beginning of every run and cannot be modified

https://docs.chef.io/attribute_types/

Templates

- An Embedded Ruby (ERB) template for dynamic generation of static text files
- May contain expressions and statements
- Expressions are delimited by open and close tags

```
<%= "I like #{$color} cars" %>
```

- Statements are delimited by a modifier like **`if`**, **`elsif`** and **`else`**

```
if condition
  # execute if true
else
  # execute if false
end
```

https://docs.chef.io/templates/

DevOps

Practice: Chef 102

Live Demonstration in Class

Chef 103

Custom Resources and Libraries. Tests

Custom Resources

- Ship directly in cookbooks
- Can utilize built-in resources and additional custom Ruby code
- Act like the existing built-in resources

```
provides :resource_name

property :property_name, RubyType, default: 'value'

action :action_a do
 # a mix of built-in Chef Infra resources and Ruby
 # this is the default action (provided first)
end

action :action_b do
 # a mix of built-in Chef Infra resources and Ruby
end
```

https://docs.chef.io/custom_resources/

Libraries

- Allow arbitrary Ruby code to be included in a cookbook
- Mostly used to write helpers that are used throughout recipes and custom resources
- Anything allowed by Ruby can take place in a library
- As well as extending built-in Chef classes

https://docs.chef.io/libraries/

Testing

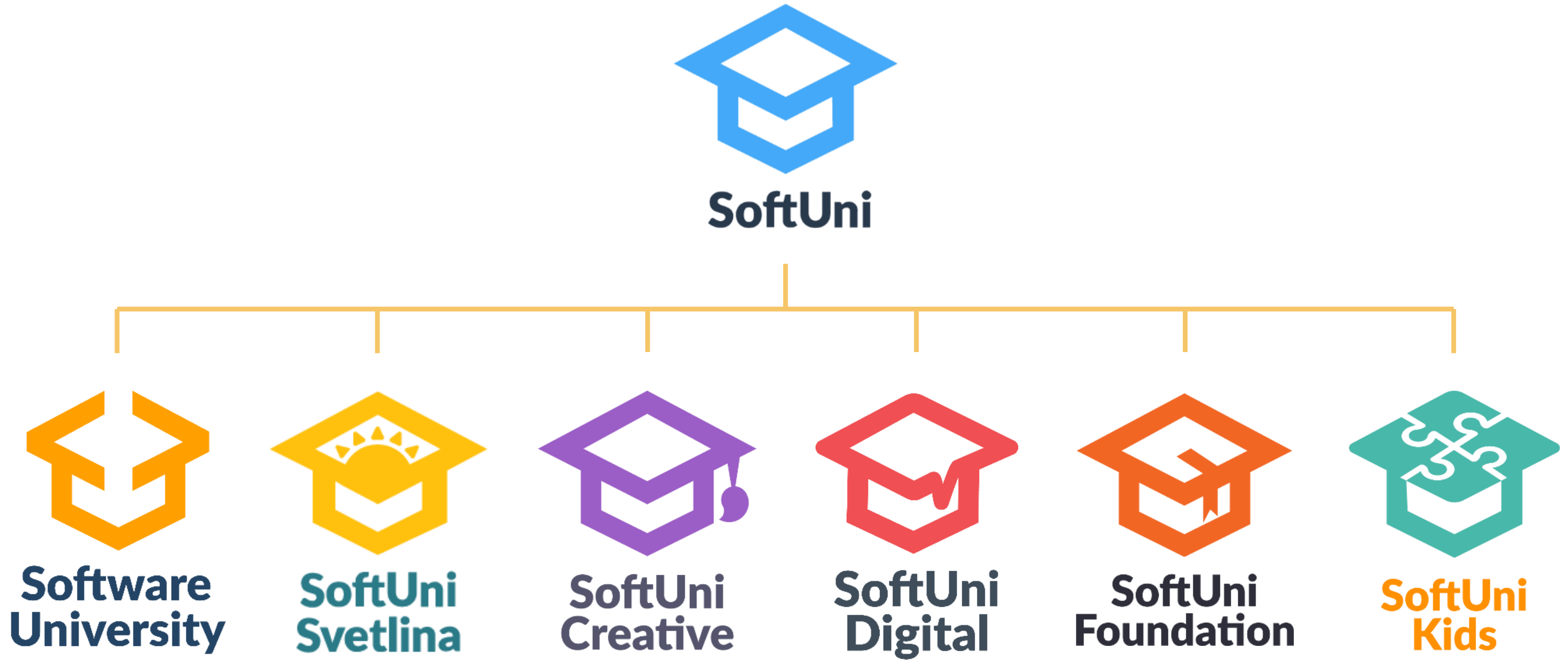
- **Test Kitchen** is easily activated on and used with Chef Workstation
- Driver plugin architecture is used to run code on various platforms
- Supported **drivers** are Vagrant, Amazon EC2, Docker, etc.
- Supported **transports** are SSH and WinRM
- Supported **provisioners** are Chef Infra, Shell, Ansible, etc.
- Supported **verifiers** include Chef InSpec, ServerSpec, Bats, etc.
- Managed via YAML configuration file (**kitchen.yml** or **kitchen.local.yml**)
- Used to be named with dot (.kitchen.yml or .kitchen.local.yml). Still available
- Controlled via the **kitchen** utility

https://kitchen.ci/

Practice: Chef 103

Live Demonstration in Class

Questions?


SoftUni
Software University
SoftUni Svetlina
SoftUni Creative
SoftUni Digital
SoftUni Foundation
SoftUni Kids

SoftUni Diamond Partners

Software University

SCHWARZ

LIDL

Kaufland

SCHWARZ Produktion

pre zero

Coca-Cola HBC Bulgaria

POKERSTARS | CAREERS

Postbank

Решения за твоето утре

AMBITIONED

DXC TECHNOLOGY

SOFTWARE GROUP

Bosch.IO

INDEAVR

Serving the high achievers

DRAFT KINGS

PHAR VISION

SmartIT

createX

SUPER HOSTING .BG

Educational Partners

CODE IT UP

License

Trainings @ Software University (SoftUni)

- Software University – High-Quality Education, Pr ofession and Job for Software Developers
  - [softuni.bg](), [softuni.org]()
- Software University Foundation
  - softuni.foundation
- Software University @ Facebook
  - facebook.com/SoftwareUniversity
- Software University Forums
  - forum.softuni.bg