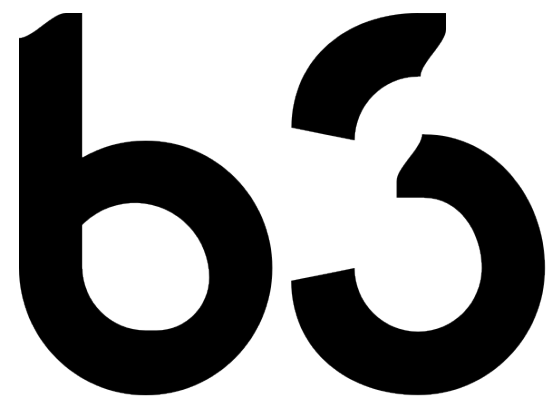


# Konfigurationshantering med Chef



**Creating  
possibilities  
together**

Jonas Linde <[jonas.linde@b3.se](mailto:jonas.linde@b3.se)>

# B3 Init AB

- IT-konsulter sedan 1994
- Infrastruktur
- Systemutveckling
- Ledning och Styrning
- DevOps
- Sedan 2017 en del av *B3 Consulting Group*



# Jonas Linde

- IT-konsult sedan 1997
- Infrastruktur
- Systemutveckling
- Ledning och Styrning
- DevOps



# Agenda

- Bakgrund
- Övning
- Uppbyggnad
- Exekvering
- Sammanfattning
- Övningar



# Bakgrund

# Alternativ

- CFEngine » 1993
- Puppet » 2005
- Chef » 2009
- Ansible » 2012
- Docker » 2013
- Serverless » 2014



# Chef

- OpsCode Inc » 2009
- Chef Inc » 2013
  - Chef InSpec » security & compliance
  - Chef Habitat » platform-independent build artifacts
  - Chef Infra » programvaran Chefs nya namn
  - Chef Automate » enterprise platform
  - Chef Workstation » command line tools
- Open Source » 2019
  - <https://downloads.chef.io>

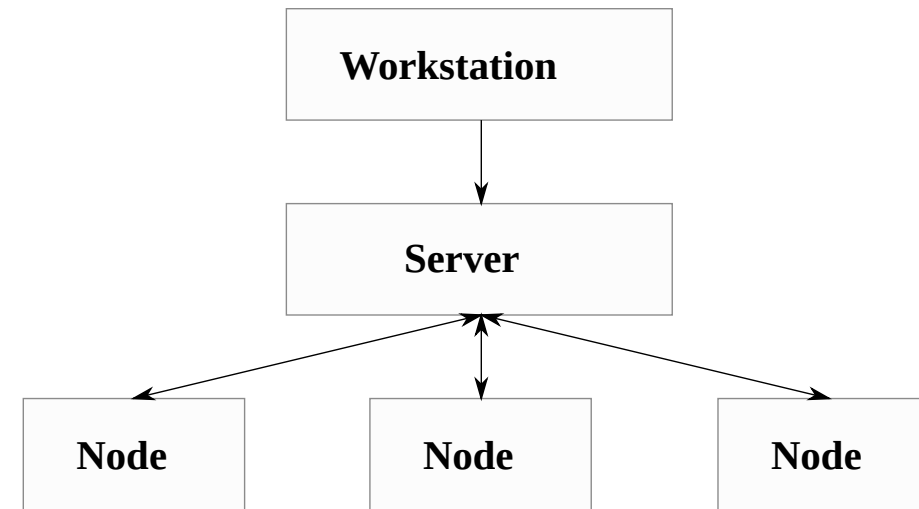


# Buzz words

- Systems Integration Framework
- Configuration Management and Automation Platform
- Configuration as Code

# Arkitektur

- workstation
- server
- nodes



# Chef Server

- Centralt lager för kokböcker och metadata
  - cookbooks
  - roles
  - environments
  - data\_bags
  - nodes
- Har ett WebUI
  - från början inte Open Source
  - men gratis upp till 25 noder

# Workstation

- Hanterar servern
- Använder `knife` för att kommunicera med servern och noderna
- Kan använda `chef-run` för att köra ad-hoc på noder
- `chef-workstation` är ett paket att ladda ner från <https://chef.io/downloads>
- `chefdk` är ett liknande paket byggt av communityn (deprecated)
- Med `knife` kan man även hämta community-skrivna kokböcker

```
knife supermarket search apt  
knife supermarket download aptly
```

# Node

- Kör **chef-client** regelbundet
- Hämtar konfiguration från servern
- Exekverar kokböcker
  - test and repair
- Med **chef-solo** kan man köra utan central server

# Övning

- Gör steg 1-3 i [Try Chef Infra](#)

# Uppbyggnad

# Basbegrepp

- chef - repo
- cookbook
- recipe
- resource
  - <https://docs.chef.io/resource.html>
- attribute

# chef-repo

- Innehåller
  - [roles/](#)
  - [environments/](#)
  - [data\\_bags/](#)
- Läs mer:
  - [http://docs.chef.io/chef\\_repo.html](http://docs.chef.io/chef_repo.html)
  - <http://docs.chef.io/roles.html>
  - <http://docs.chef.io/environments.html>
  - [http://docs.chef.io/data\\_bags.html](http://docs.chef.io/data_bags.html)

# Role

```
{
  "chef_type": "role",
  "json_class": "Chef::Role",
  "name": "webserver",
  "description": "The base role for systems that serve HTTP traffic",
  "default_attributes": {
    "apache2": {
      "listen_ports": [
        "80"
      ]
    }
  },
  "run_list": [
    "recipe[apache2]",
    "recipe[apache2::mod_ssl]",
    "role[monitor]"
  ]
}
```

# Environment

```
{
  "chef_type": "environment",
  "json_class": "Chef::Environment",
  "name": "dev",
  "description": "Dev Environment",
  "default_attributes": {
    "apache2": {
      "listen_ports": [
        "80"
      ]
    }
  },
  "cookbook_versions": {
    "couchdb": "= 11.0.0"
  }
}
```

# Data bag

```
data_bags/  
  sample_bag_1/  
    sample_item_1_1.json  
    sample_item_1_2.json  
    sample_item_1_3.json  
  sample_bag_2/  
    sample_item_2_1.json  
    sample_item_2_2.json  
  
{  
  // This is a comment  
  "id": "ITEM_NAME",  
  "anykey": "anyvalue"  
}
```

# Cookbook

- Innehåller
  - `README.md`
  - `metadata.rb`
  - `recipes/`
  - `attributes/`
  - `files/`
  - `templates/`
  - `extensions/`
- Läs mer: <http://docs.chef.io/cookbooks.html>

# Recipe

- En samling resurser
- Kan även innehålla styrkoder
  - villkor: `case`, `if`, `unless`
  - loopar: `each`
- Läs mer: <https://docs.chef.io/recipes.html>

# Exempelrecept

```
$ cat webserver/recipes/default.rb
#
# Cookbook:: webserver
# Recipe:: default
#
# Copyright:: 2018, The Authors, All Rights Reserved.
apt_update

package 'apache2'

template '/var/www/html/index.html' do
  source 'index.html.erb'
end

service 'apache2' do
  action [:enable, :start]
end
```

# Några resurser

- `cron`
- `execute`
  - `bash`
- `file`
- `package`
  - `apt_package`
  - `apt_update`
- `service`
- `template`
- `user`

# Resursexempel

```
package 'tar' do  
  version '1.16.1'  
  action :install  
end
```

eller

```
package 'tar' do  
  version '1.16.1'  
end
```

eller

```
package 'tar'
```

# Resursexempel

```
bash 'extract module' do
  cwd ::File.dirname(src_filepath)
  code <<-EOH
    mkdir -p #{extract_path}
    tar xzf #{src_filename} -C #{extract_path}
    mv #{extract_path}/*/* #{extract_path}/
  EOH
  not_if { ::File.exist?(extract_path) }
end
```

# Kedja

```
execute 'test-nagios-config' do
  command 'nagios3 --verify-config'
  action :nothing
end
```

```
template '/etc/nagios3/configures-nagios.conf' do
  # other parameters
  notifies :run, 'execute[test-nagios-config]', :delayed
end
```

eller

```
execute 'test-nagios-config' do
  command 'nagios3 --verify-config'
  action :nothing
  subscribes :run, 'template[/etc/nagios3/configures-nagios.conf]', :immediately
end
```

# Attributes

- Kommer från
  - `ohai` » verktyg som samlar fakta om noden
  - `attributes/` » i en kokbok
  - `recipes/` » i en kokbok
  - `environments/` » från `chef-repo`
  - `roles/` » från `chef-repo`

- Exempel:

```
$ cat cookbooks/starter/attributes/default.rb  
default["starter_name"] = "Sam Doe"  
override["system_name"] = "Chef Starter"
```

- Läs mer: <https://docs.chef.io/attributes.html>

# Attributordning

	Attribute Files	Node / Recipe	Environment	Role
default	1	2	3	4
force_default	5	6		
normal	7	8		
override	9	10	12	11
force_override	13	14		
automatic			15	

# Template

- Exempel:

```
$ cat webserver/templates/index.html.erb
<html>
  <head>
    <title>Learn Chef Demo</title>
  </head>
  <body>
    <h1>Hello Learn Chef</h1>
    <p>This is <%=node['hostname']%></p>
  </body>
</html>
```

# Exekvering

# Grundflöde

- På workstation skrivs kokböcker och roller m.m.
- Med `knife` laddas allt upp till servern
- Noderna kör `chef-client` som hämtar konfiguration från servern och exekverar recepten
- `cron` kan användas för att starta `chef-client` periodiskt
- Med `knife` kan man `bootstrap`a en nod

# Solo

- `chef-solo` är en lokal Chef-server
- Kommandot '`chef-client -z ...`' startar en lokal server
  - och kör sedan mot den

# Ad-hoc

- Kommandot `chef-run` kör enstaka recept på en eller fler noder

# Körlista

- Listan av recept som ska exekveras kan anges

- på kommandoraden:

```
chef-client --runlist "apache2,recipe[apache2::mod_ssl],role[webserver]"
```

- på kommandoraden i cron

- i en roll:

```
"run_list": [  
  "recipe[apache2]",  
  "recipe[apache2::mod_ssl]",  
  "role[webserver]"  
]
```

- Om ingen `run_list` anges, används senaste lyckade exekvering
  - sparas i `node`-objektet på servern

# Sammanfattning

# Configuration as Code

- Resurser specificeras i recept i kokböcker
- Environments och roller specificerar vilka recept som ska köras
- Attribut kan sättas på många olika sätt
- Avsett att köras med **pull** i client-server-arkitektur
- Men kan även köras med **push** med eller utan server

# Nackdelar

- Kokbokshanteringen är ömtålig
- Attributordningen är hemsk
- Serverbegreppet är gammalmodigt
- Pull-metoden kräver övervakning
- Servern är besvärlig

# Fördelar

- Kan hantera **hela** infrastrukturen
  - automatisering
  - kontroll
- Pull-metoden ger parallella körningar
- Kraftfullt språk
- Finns för många OS
- Kan hantera publika moln

# Övningar

- Gör steg 4 och 5 i [Try Chef Infra](#)
  - städa inte upp förrän senare
- Skriv en enkel egen kokbok som t.ex.
  - skapar en fil som ägs av `myuser:mygroup` och har filskyddet `-r--r--r--`
  - skapar en fil med innehållet "`filen ligger på <hostname>`" där `<hostname>` är `web1`, `web2` eller `lb` beroende på nod
  - uppgraderar alla paket på en nod
- Skaffa en chef-server
  - <https://manage.chef.io/signup>
  - gratis upp till 5 noder och 2 användare
- <https://learn.chef.io>

# Tack för ordet!

