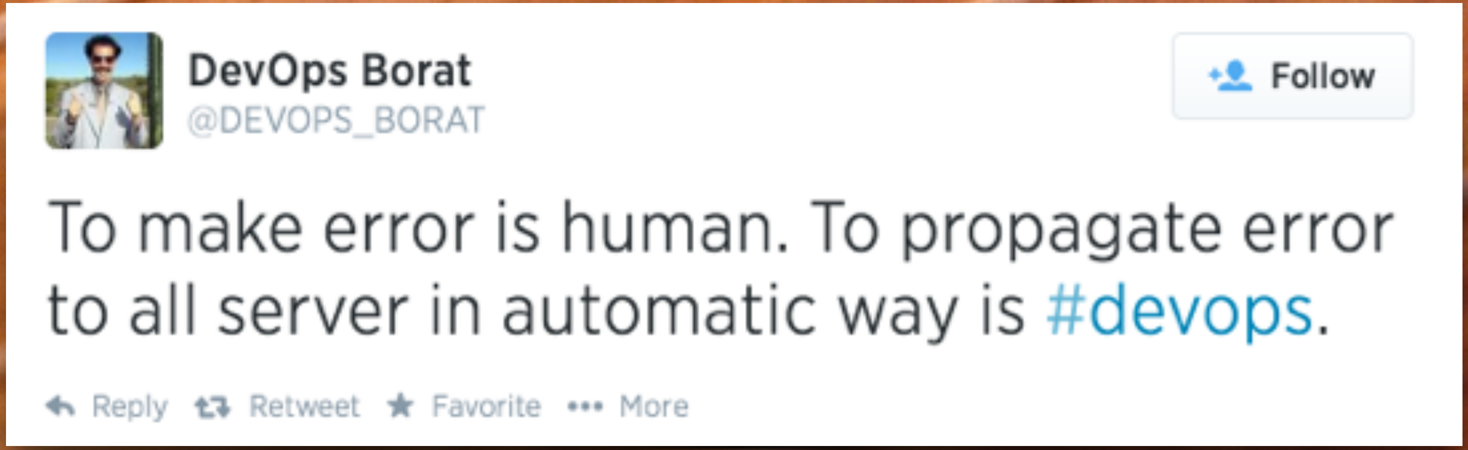
codecentric

Multi-Stage Ansible

Victor Volle
Ansible Meetup München 2016

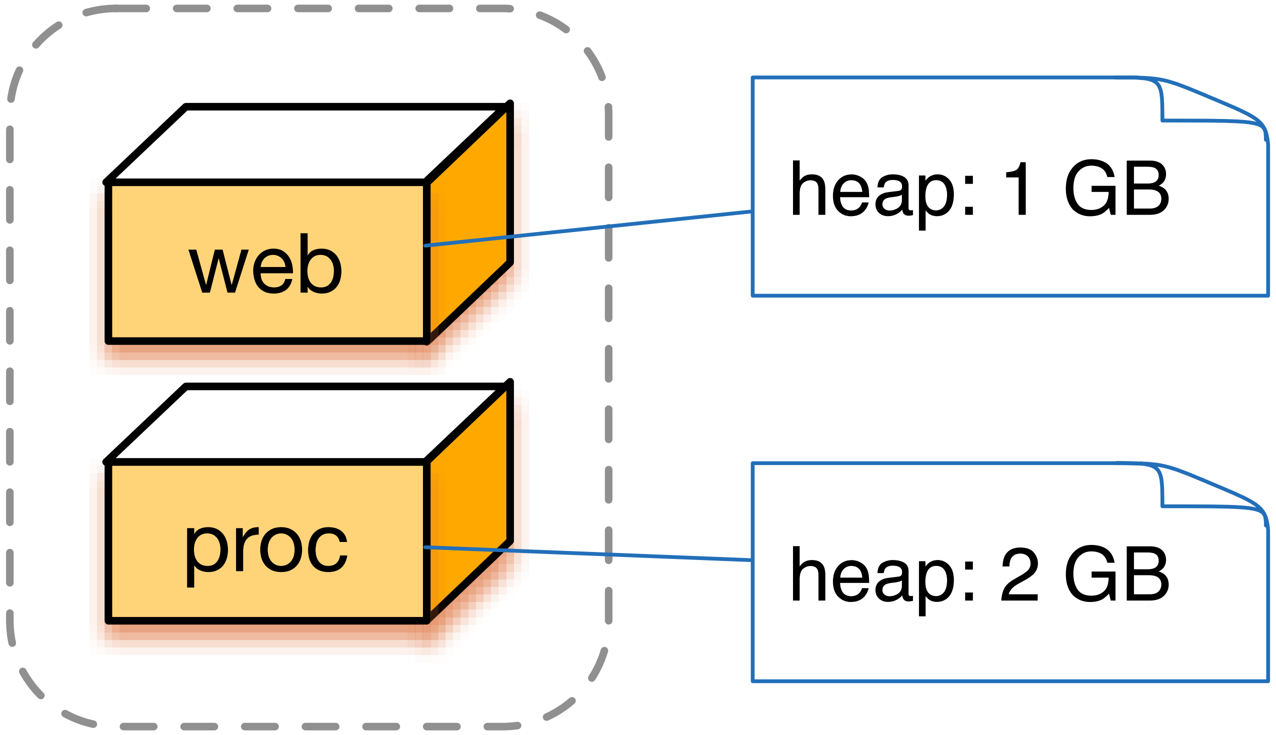
From 10 servers to 1000


DevOps Borat
@DEVOPS_BORAT

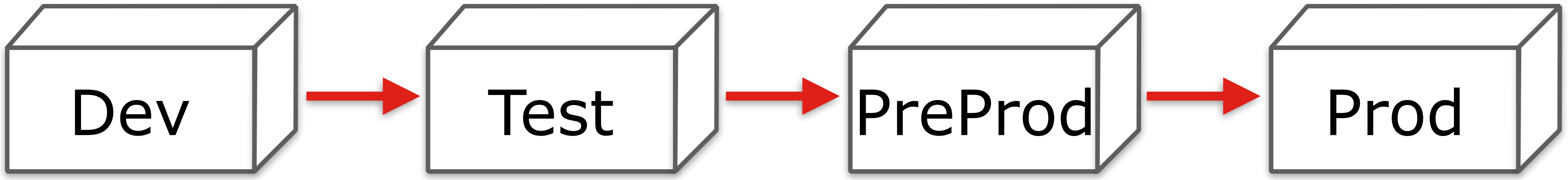
To make error is human. To propagate error to all server in automatic way is #devops.

https://www.flickr.com/photos/mariachily/4110399647

“configuration”



“configuration”



web	**mango01** heap=2GB	**mango02** heap=2GB	**mango03-06** heap=4GB	**mango07-10** heap=4GB
proc	**dattel01** heap=2GB	**dattel02** heap=4GB	**dattel02, dattel03** heap=8GB	**dattel07, dattel08** heap=8GB

multi-stage: possible solutions

1. group variables
2. multiple inventories
3. exploiting variable precedence
4. using “children”
5. creating your own vars plugin
6. talk to Brian Coca

multi-stage: possible solutions

1. **group variables**
2. multiple inventories
3. exploiting variable precedence
4. using “children”
5. creating your own vars plugin
6. talk to Brian Coca

group variables (inventory)

```
# Inventory
[web]
mango[01-10]
[proc]
dattel[01-08]


[dev]
mango01
dattel01
...
[prod]
mango[07-10]
dattel[07-08]
```

```
...
[web:vars]
heap=2G
[proc:vars]
heap=2G


[dev:vars]
heap=2G
```

group variables

it is not possible to set different values for "heap" for **web** and **proc**

we need a "namespace"

group variables (inventory)

```
# Inventory
[web]
mango[01-10]
[proc]
dattel[01-08]


[dev]
mango01
dattel01
...
[prod]
mango[07-10]
dattel[07-08]
```

```
...
[web:vars]
webheap=2G
[proc:vars]
procheap=2G


[dev:vars]
webheap=2G
procheap=4G
...
[prod:vars]
webheap=4G
procheap=8G
```

	dev	prod
procheap	4G	2G

groups are alphabetically sorted

"proc" < "prod"

Default Workspace structure

group variables

```
workspace/
├── files/
├── group_vars/
├── host_vars/
├── roles/
├── templates/
├── ansible.cfg
├── ansible_inventory
├── java.yml
…
```

groups are still sorted alphabetically

Variables

- Avoid defining the variable “x” in 47 places and then ask the question “which x gets used”. Why? Because that’s not Ansible’s Zen philosophy of doing things.

  There is only one Empire State Building. One Mona Lisa, etc. Figure out where to define a variable, and don’t make it complicated.

multi-stage: possible solutions

1. group variables
2. **multiple inventories**
3. exploiting variable precedence
4. using “children”
5. creating your own vars plugin
6. talk to Brian Coca

inventories per environment

```
# dev_inventory

[web]
mango01

[proc]
dattel01

[dev:children]
web
proc


[dev:vars]
webheap=2G
procheap=4G
```

…

```
# prod_inventory

[web]
mango02

[proc]
dattel02

[prod:children]
web
proc


[prod:vars]
webheap=4G
procheap=8G
```

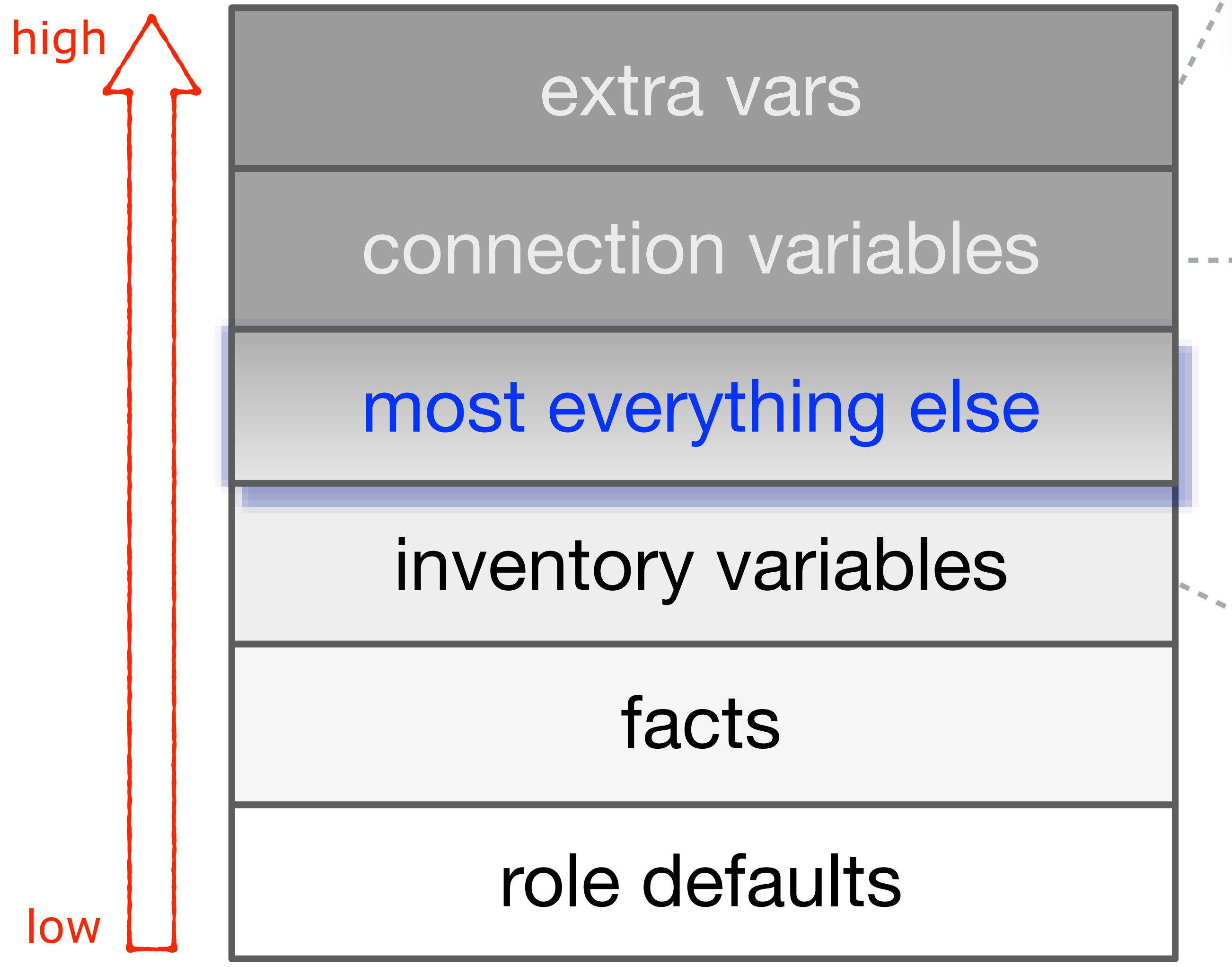
- **"good enough" in most cases! Use it as long as possible**
- *(you should place the variables in the group_vars folder though)*
- you cannot invoke a playbook on *all* hosts — which you probably do not want anyway
- there is some duplication in the inventories, but not too much
- **get's very unwieldy when you add another dimension like 'location'**

see e.g. : http://rosstuck.com/multistage-environments-with-ansible/

multi-stage: possible solutions

1. group variables
2. multiple inventories
3. **exploiting variable precedence**
4. using “children”
5. creating your own vars plugin
6. talk to Brian Coca

Variable Precedence

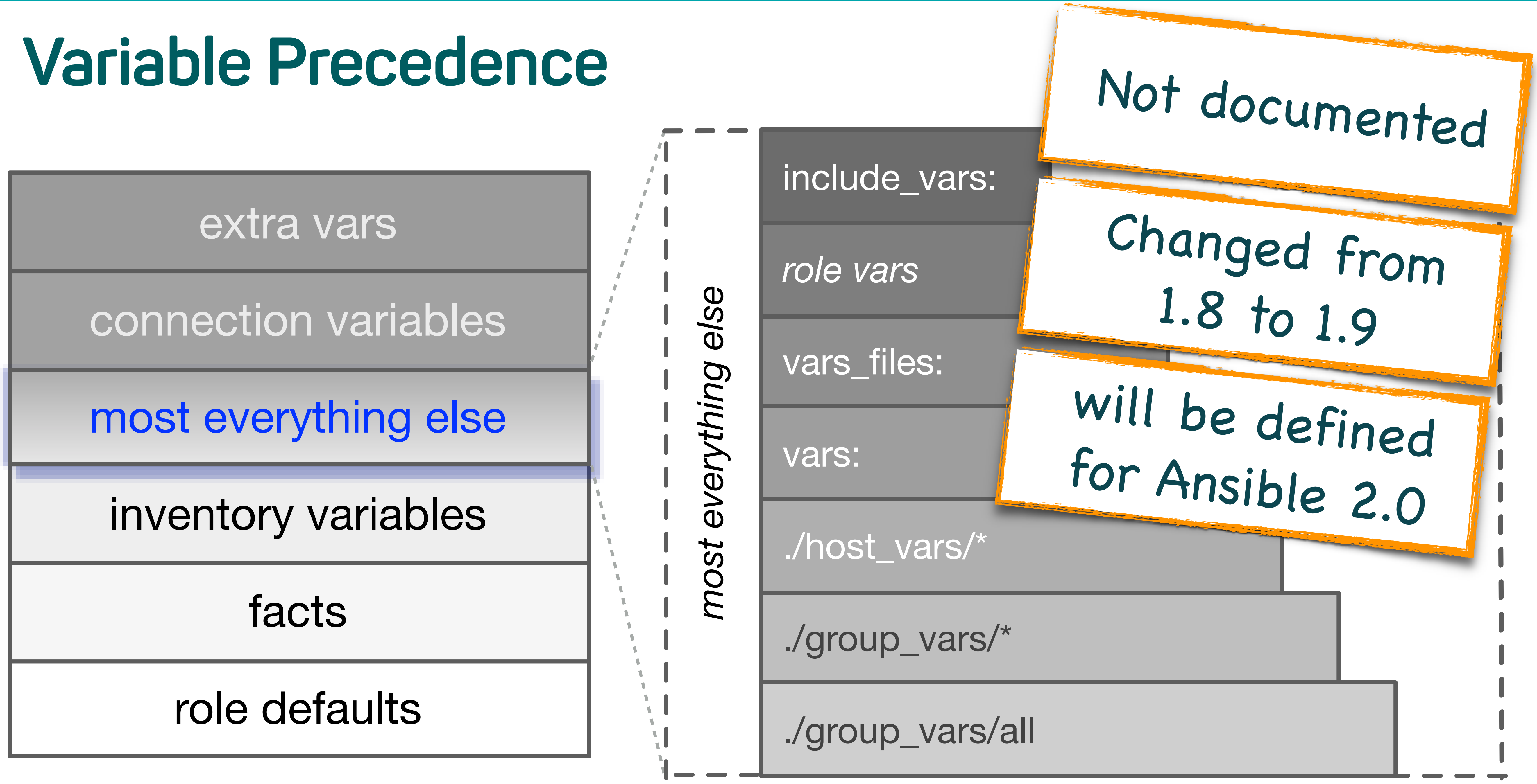


```
❯ ansible-playbook -e “heap=16G" ...
```

```
# Inventory
mango01 ansible_ssh_host=10.0.1.16
```

```
# Inventory
...
mango10 heap=8G
[web:vars]
heap=4G
```

Variable Precedence



codecentric

variable precedence

Use *include_vars* task or *vars_files*

(That’s what we are currently doing)

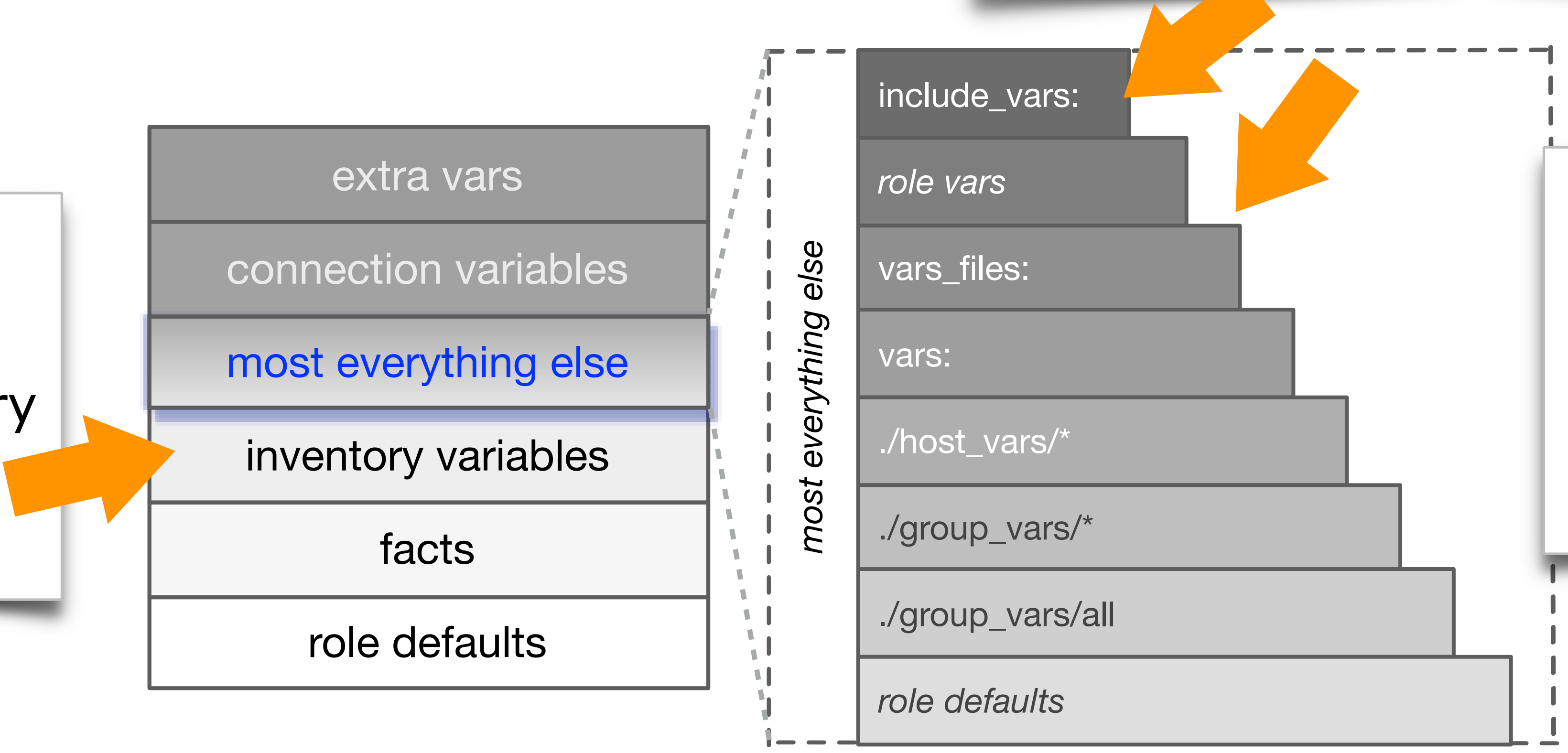
Problem: you have to add that to every Playbook

Inventory per Environment

Problem: inventory variables have a low priority

Write your own vars plugin

(That’s what I am currently planning)



Variables per stage/envrionment

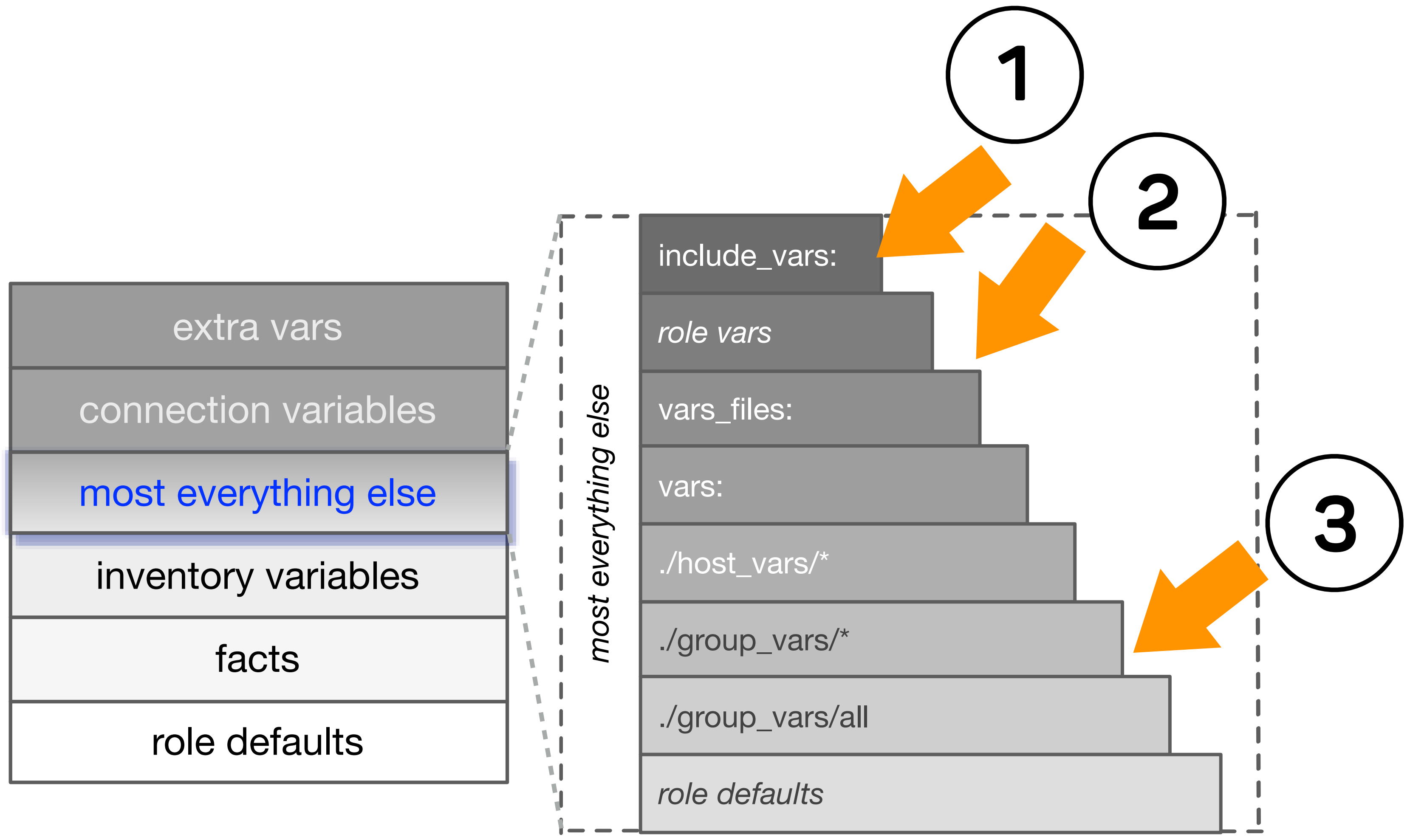
```
# varprecedence.yml
---
- name: Check Var precedence
  hosts: "{{ lookup('env','STAGE') }}:&web"

  vars_files:
    - "{{ lookup('env','STAGE') }}_vars.yml"

  tasks:
    - include_vars: include_vars{{ lookup('env','STAGE') }}.yml
    ...
```

servers, which are in group "web" and in group "*{{STAGE}}*"

(1) load variables from environment/stage specific file

(2) load variables from environment/stage specific file

“exploiting variable precedence”

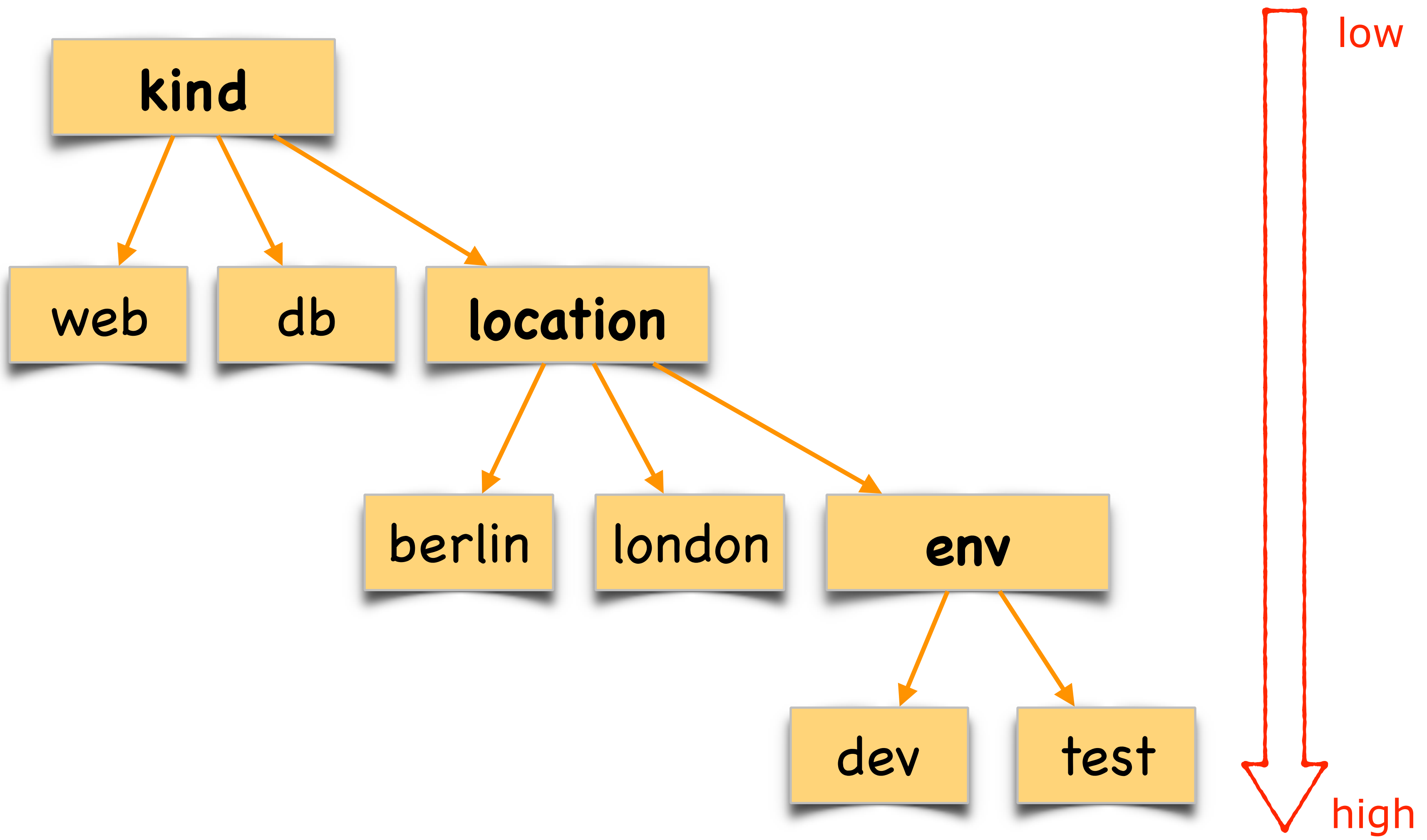


multi-stage: possible solutions

1. group variables
2. multiple inventories
3. exploiting variable precedence
4. **using “children”**
5. creating your own vars plugin
6. talk to Brian Coca

children have higher priority than their parent

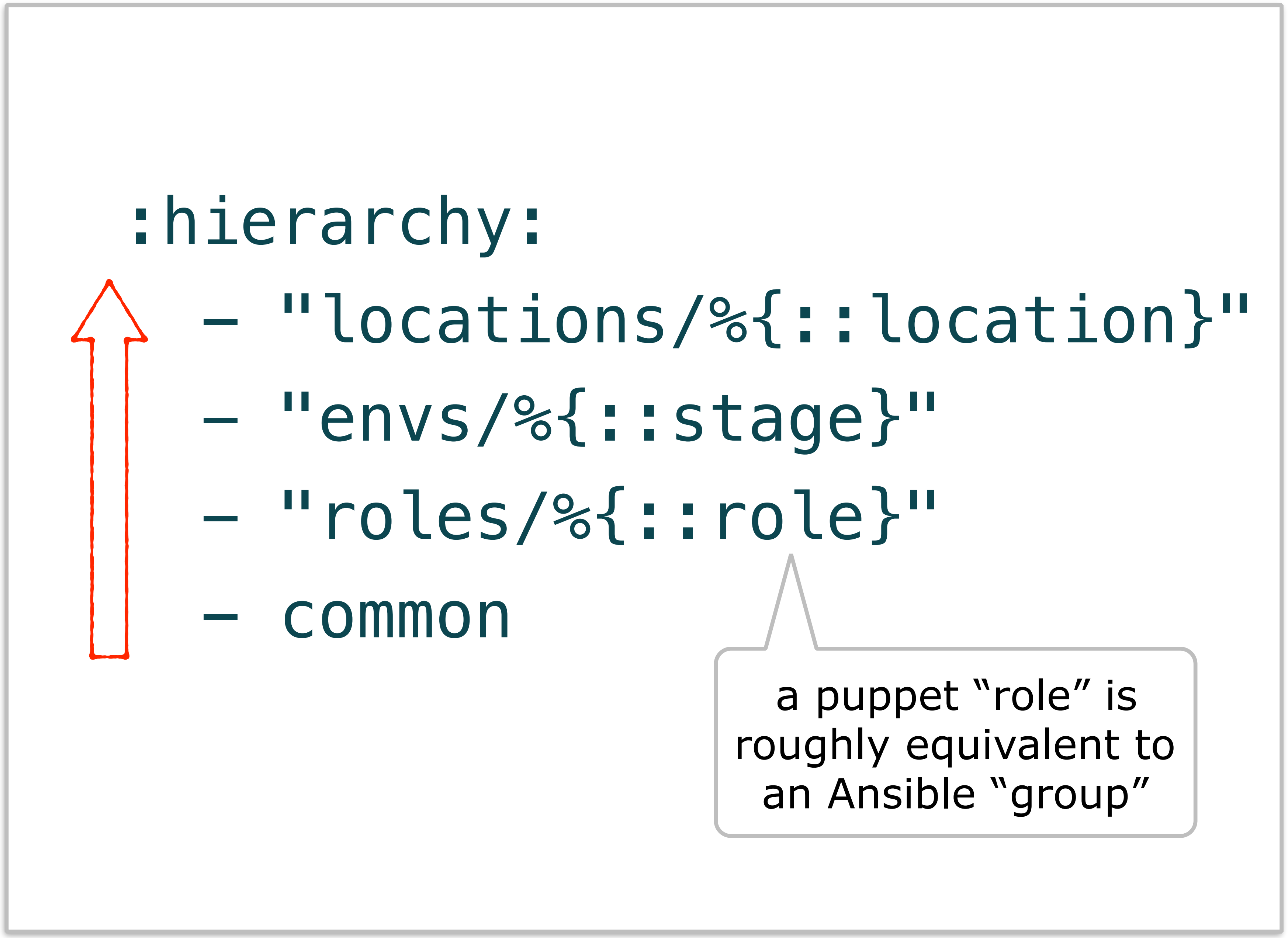
```
[env:children]
dev
test

[location:children]
berlin
env
london

[kind:children]
db
location
web
```


$$\begin{tikzcd}
\text{kind} \arrow[d] \arrow[dr] \arrow[drr] & & & & \\
\text{web} & \text{db} & \text{location} \arrow[d] \arrow[dr] \arrow[drr] & & \\
& & \text{berlin} & \text{london} & \text{env} \arrow[d] \arrow[dr] & \\
& & & & \text{dev} & \text{test}
\end{tikzcd}$$

multi-stage: possible solutions

1. group variables
2. multiple inventories
3. exploiting variable precedence
4. using “children”
5. **creating your own vars plugin**
6. talk to Brian Coca

Puppet Hiera: a simple Hierarchical Database



```
:hierarchy:
  - "locations/%{::location}"
  - "envs/%{::stage}"
  - "roles/%{::role}"
  - common
```

a puppet “role” is roughly equivalent to an Ansible “group”

Puppet Hiera: a simple Hierarchical Database

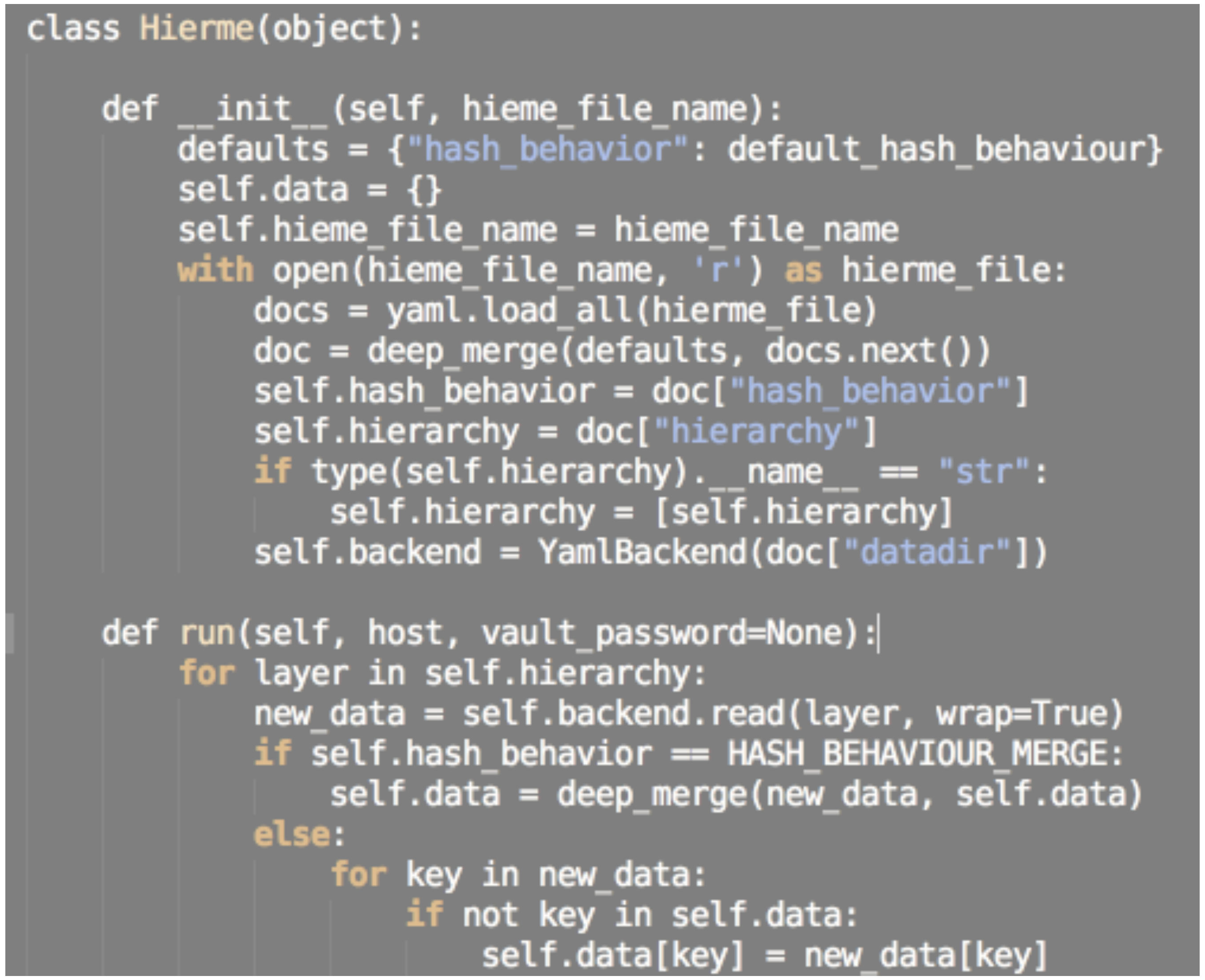
```
:hierarchy:
  - "locations/%{::location}"
  - "envs/%{::stage}"
  - "roles/%{::role}"
  - common
```

```
├── locations/
|    ├── ams.yaml
|    ├── asia.yaml
|    ├── berlin.yaml
|    ├── praha.yaml
|    └── timb.yaml
├── envs/
|    ├── dev.yaml
|    ├── preprod.yaml
|    ├── prod.yaml
|    └── test.yaml
└── roles/
     ├── proc.yaml
     └── web.yaml
```

Ansible: vars_plugin

https://github.com/kontrafiktion/ansible-plugins

nothing to see there, yet

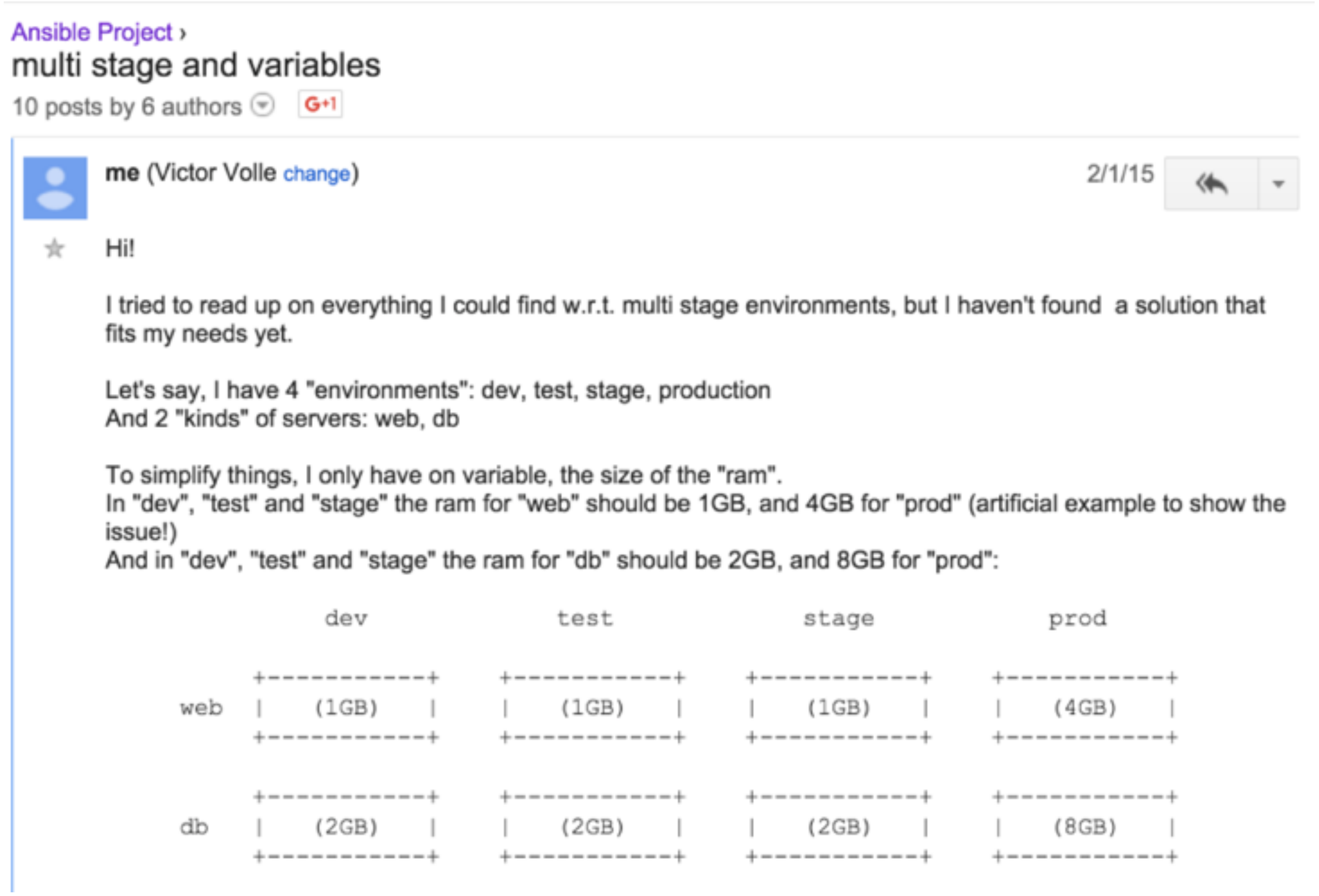


```
class Hierme(object):

    def __init__(self, hieme_file_name):
        defaults = {"hash_behavior": default_hash_behaviour}
        self.data = {}
        self.hieme_file_name = hieme_file_name
        with open(hieme_file_name, 'r') as hierme_file:
            docs = yaml.load_all(hierme_file)
            doc = deep_merge(defaults, docs.next())
            self.hash_behavior = doc["hash_behavior"]
            self.hierarchy = doc["hierarchy"]
            if type(self.hierarchy).__name__ == "str":
                self.hierarchy = [self.hierarchy]
            self.backend = YamlBackend(doc["datadir"])

    def run(self, host, vault_password=None):
        for layer in self.hierarchy:
            new_data = self.backend.read(layer, wrap=True)
            if self.hash_behavior == HASH_BEHAVIOUR_MERGE:
                self.data = deep_merge(new_data, self.data)
            else:
                for key in new_data:
                    if not key in self.data:
                        self.data[key] = new_data[key]
```

multi-stage: possible solutions

1. group variables
2. multiple inventories
3. exploiting variable precedence
4. using “children”
5. creating your own vars plugin
6. **talk to Brian Coca**

Februar 2015: Mailing list



Ansible Project ›

multi stage and variables

10 posts by 6 authors G+1

me (Victor Volle change) 2/1/15

Hi!

I tried to read up on everything I could find w.r.t. multi stage environments, but I haven't found a solution that fits my needs yet.

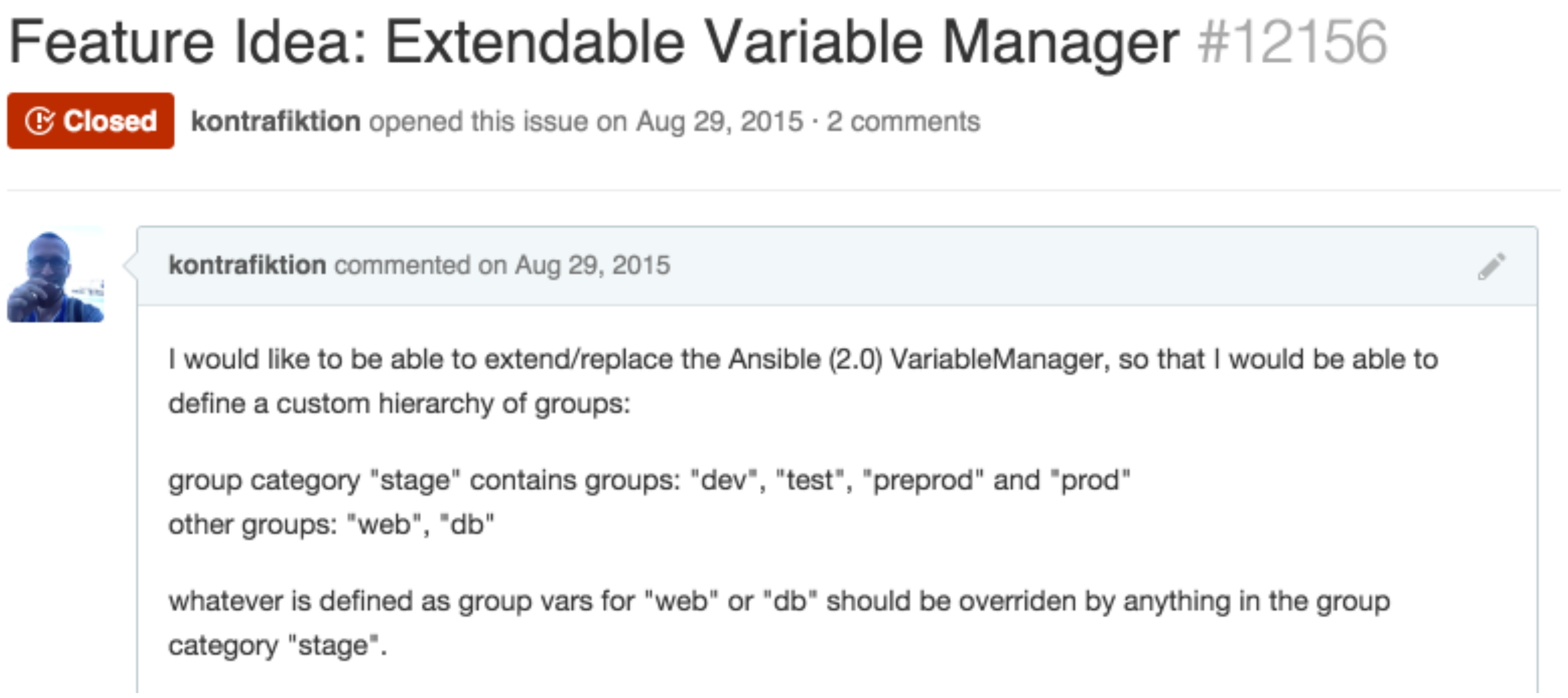
Let's say, I have 4 "environments": dev, test, stage, production
And 2 "kinds" of servers: web, db

To simplify things, I only have on variable, the size of the "ram".
In "dev", "test" and "stage" the ram for "web" should be 1GB, and 4GB for "prod" (artificial example to show the issue!)
And in "dev", "test" and "stage" the ram for "db" should be 2GB, and 8GB for "prod":

```
              dev              test             stage             prod

         +-----------+    +-----------+    +-----------+    +-----------+
  web    |   (1GB)   |    |   (1GB)   |    |   (1GB)   |    |   (4GB)   |
         +-----------+    +-----------+    +-----------+    +-----------+

         +-----------+    +-----------+    +-----------+    +-----------+
  db     |   (2GB)   |    |   (2GB)   |    |   (2GB)   |    |   (8GB)   |
         +-----------+    +-----------+    +-----------+    +-----------+
```

codecentric

August 2015: Issue 12156



Feature Idea: Extendable Variable Manager #12156

Closed **kontrafiktion** opened this issue on Aug 29, 2015 · 2 comments

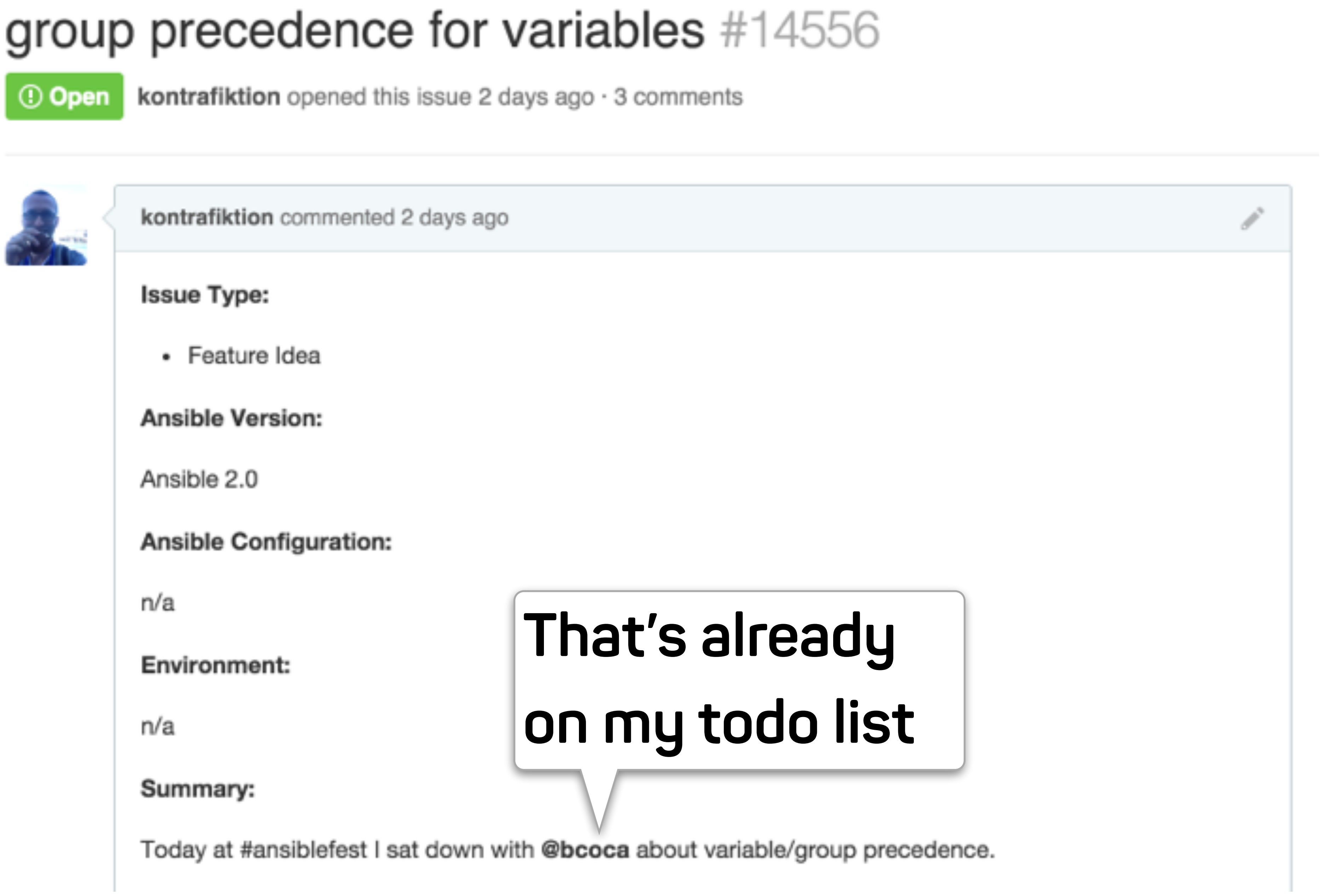
kontrafiktion commented on Aug 29, 2015

I would like to be able to extend/replace the Ansible (2.0) VariableManager, so that I would be able to define a custom hierarchy of groups:

group category "stage" contains groups: "dev", "test", "preprod" and "prod"
other groups: "web", "db"

whatever is defined as group vars for "web" or "db" should be overriden by anything in the group category "stage".

February 2016: #ansiblefest



group precedence for variables #14556

Open kontrafiktion opened this issue 2 days ago · 3 comments

kontrafiktion commented 2 days ago

Issue Type:

- Feature Idea

Ansible Version:

Ansible 2.0

Ansible Configuration:

n/a

Environment:

n/a

Summary:

Today at #ansiblefest I sat down with **@bcoca** about variable/group precedence.

That's already on my todo list

Who me?

Dr. Victor Volle

IT-Something

"Senior IT Consultant"	2015-	codecentric
"Architect"	2008-	Senacor Technologies AG
"Chief Architect JEE"	2004-	ING DiBa
Developer, Architect, Head of …	1996-	develop group, Erlangen