

RED HAT FORUM 2018 ZURICH



EVERYTHING AS CODE

A JOURNEY INTO IT AUTOMATION AND STANDARDIZATION
RAPHAEL PINSON | INFRASTRUCTURE DEVELOPER **CAMPTOCAMP**
PETER MUMENTHALER | SOLUTION ARCHITECT **RED HAT**

camptocamp

INNOVATIVE SOLUTIONS
BY OPEN SOURCE EXPERTS



INNOVATION SOLUTIONS BY OPEN SOURCE EXPERTS

AN OPEN SOURCE SPECIALIST

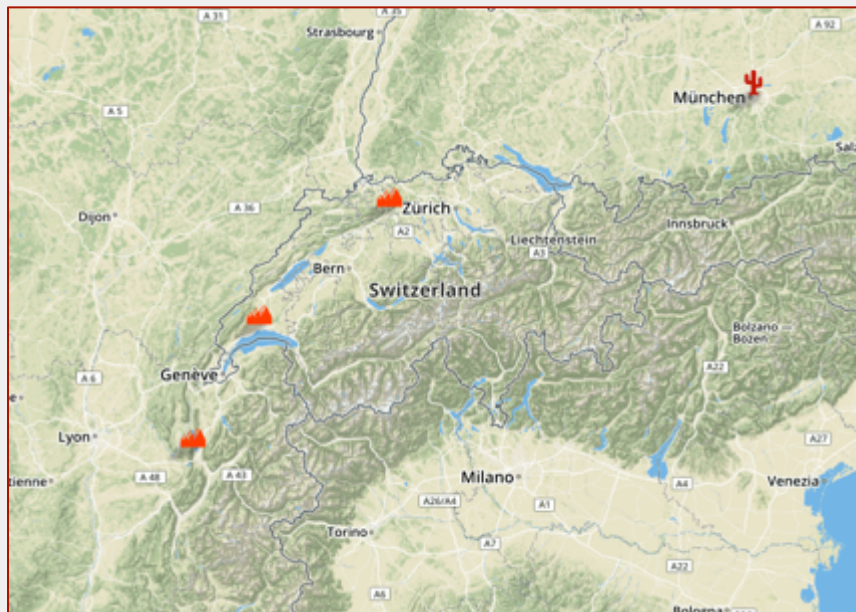
- Geographic Information Systems (GIS)
- Business Management (ERP)
- Server Management (IT Automation and Orchestration)



INNOVATION SOLUTIONS BY OPEN SOURCE EXPERTS

PRESENT IN THREE COUNTRIES

- Switzerland: Lausanne, Olten
- France: Chambéry
- Germany: Munich

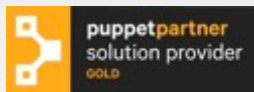




INNOVATION SOLUTIONS BY OPEN SOURCE EXPERTS

INFRASTRUCTURE DEPARTMENT

- Involved in Open Source communities (Puppet, Terraform, Rancher)
- 13 Systems Administrators
- Manages ~800 servers
- Partnerships



PREMIER

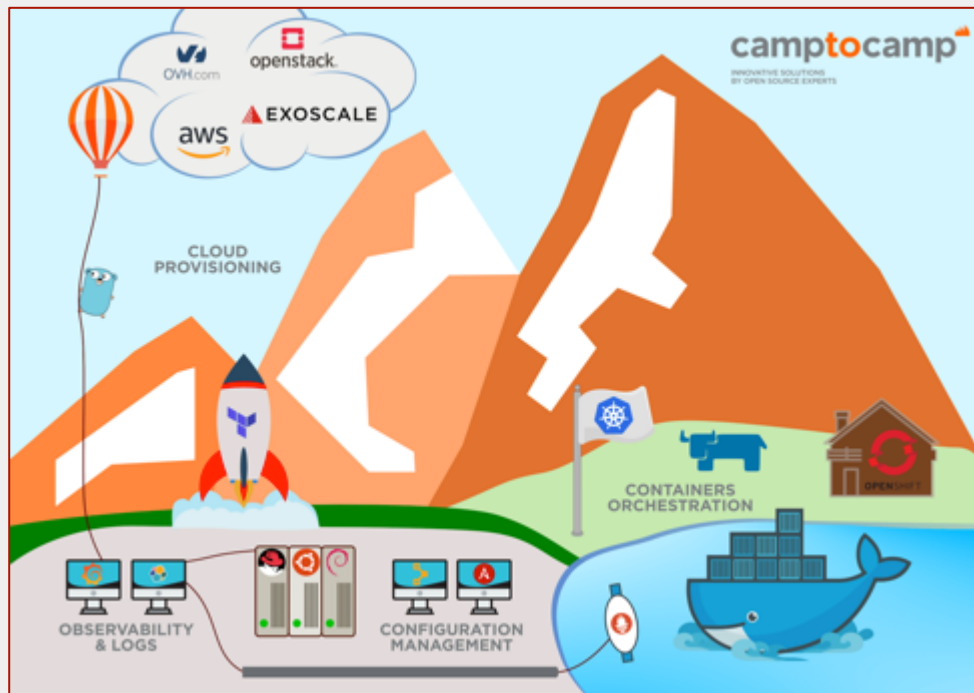
BUSINESS PARTNER

+ SWITZERLAND

GERMANY

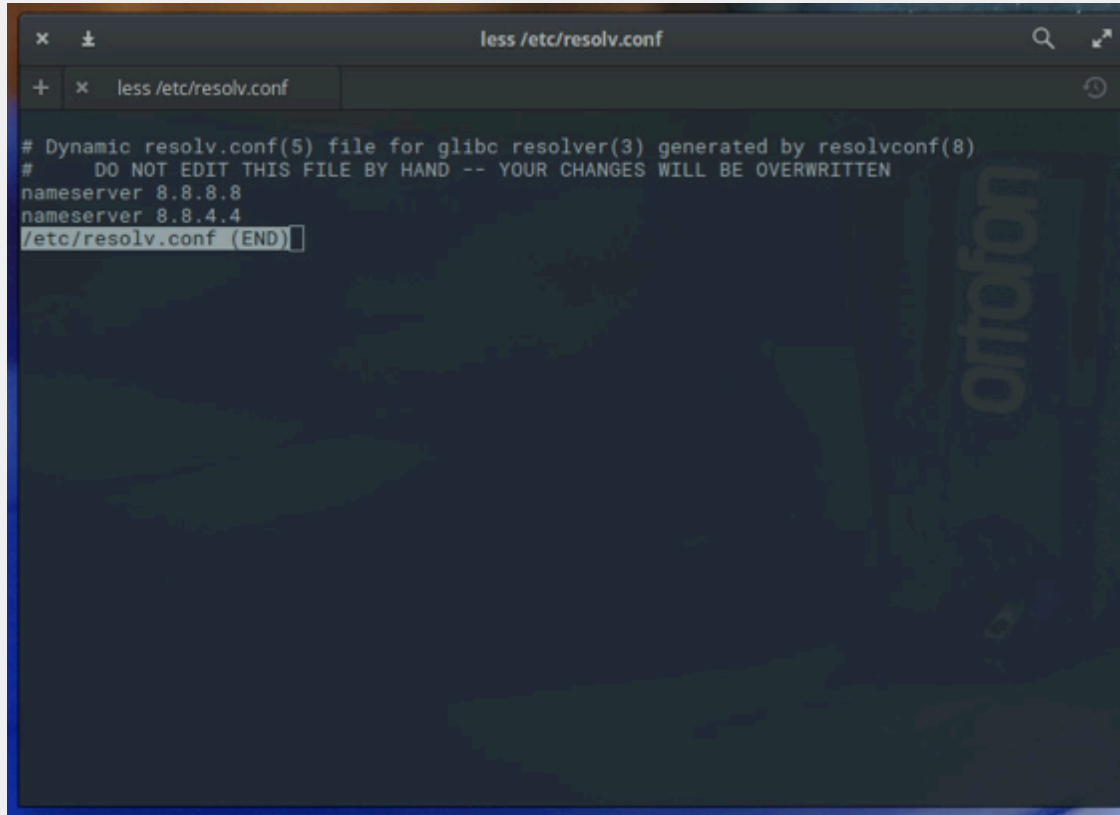


INFRASTRUCTURE TECHNOLOGY STACK



WHY AUTOMATE?

Question #1

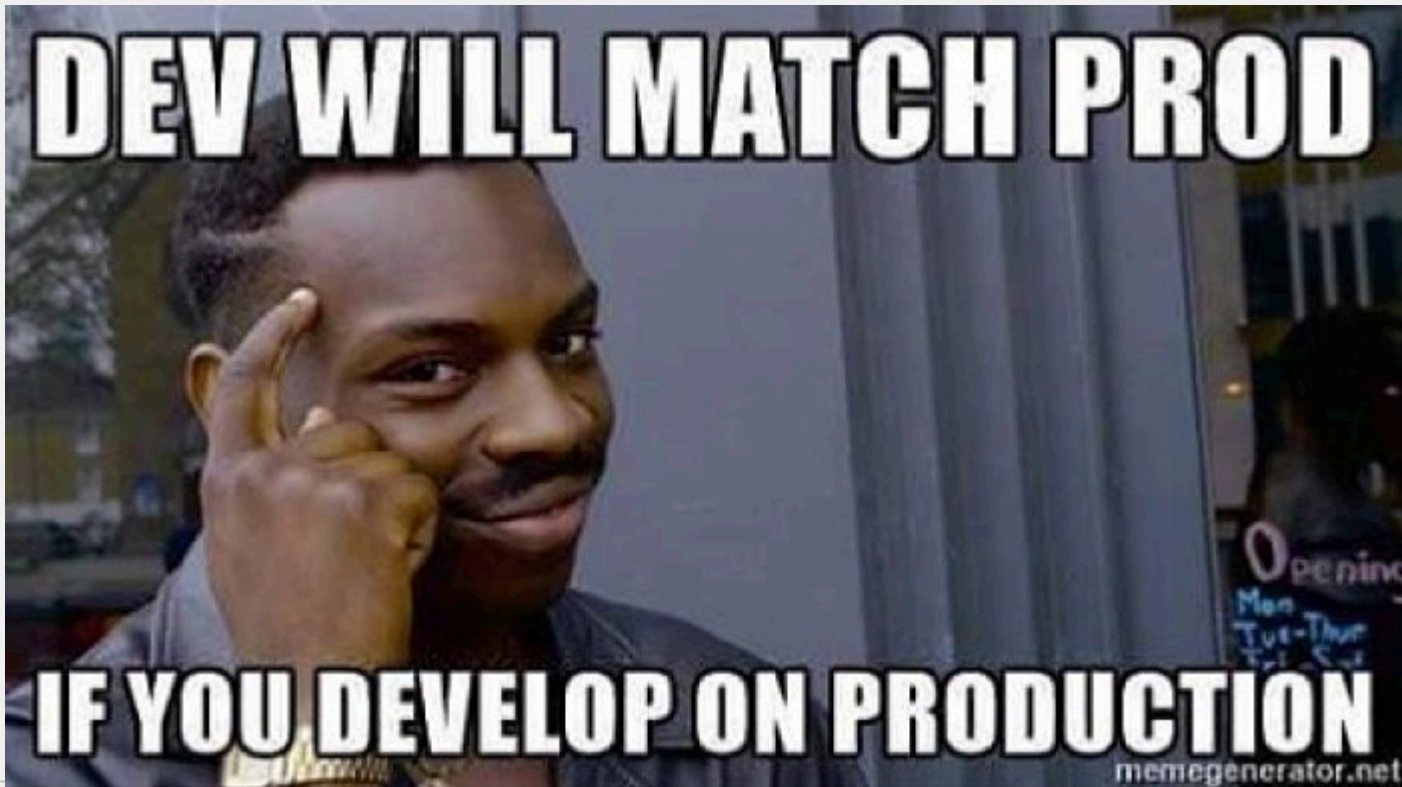


A screenshot of a terminal window with a dark background. The window title is "less /etc/resolv.conf". The terminal shows the following text:

```
# Dynamic resolv.conf(5) file for glibc resolver(3) generated by resolvconf(8)
#     DO NOT EDIT THIS FILE BY HAND -- YOUR CHANGES WILL BE OVERWRITTEN
nameserver 8.8.8.8
nameserver 8.8.4.4
/etc/resolv.conf (END)
```

The cursor is positioned at the end of the last line, after "(END)".

Question #2



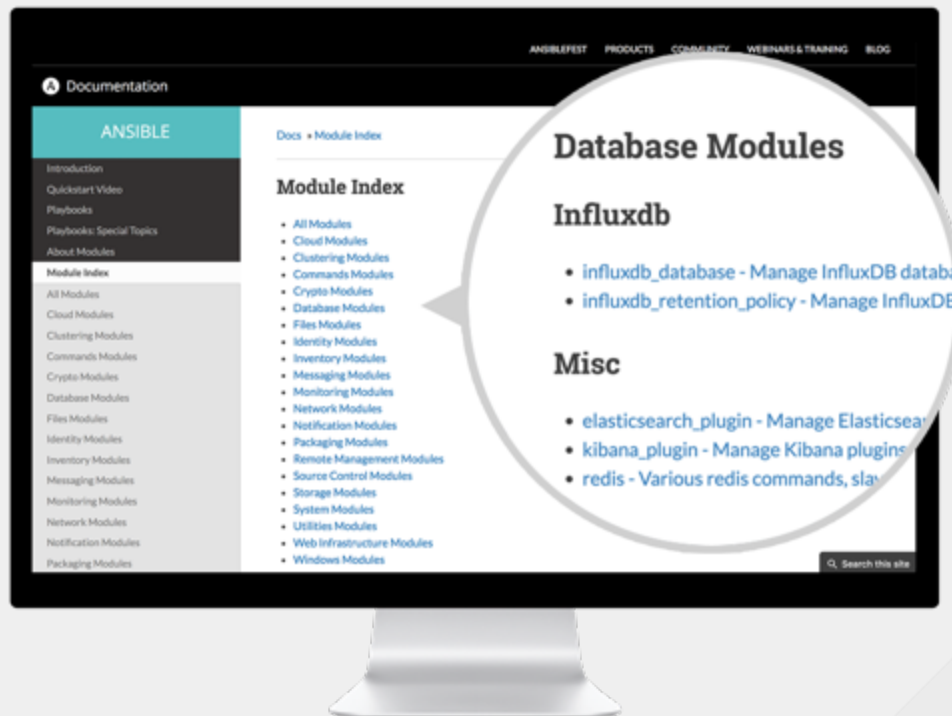
Define once...

```
---
- name: install and start apache
  hosts: web
  become: yes
  vars:
    http_port: 80

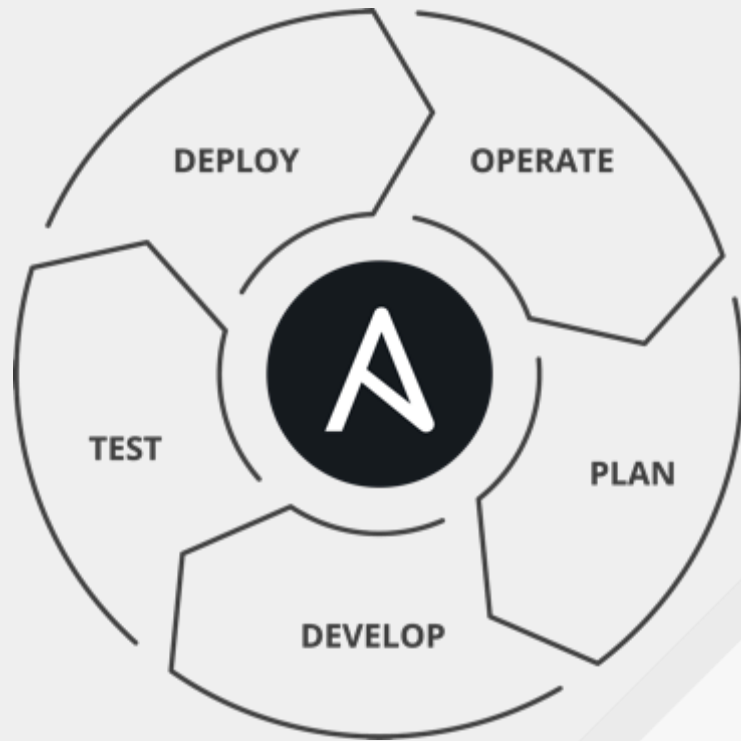
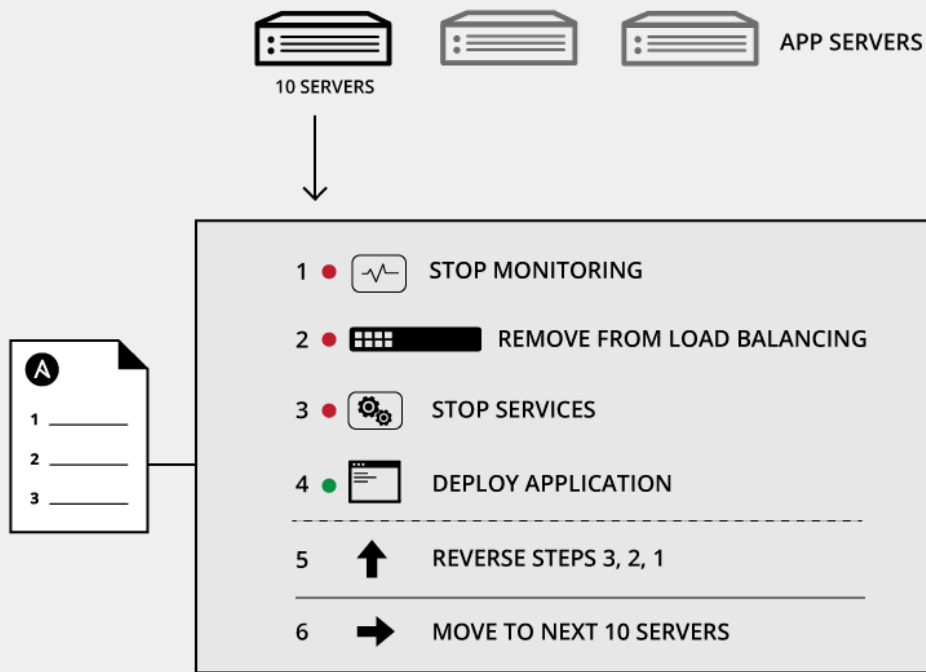
tasks:
- name: httpd package is present
  yum:
    name: httpd
    state: latest

- name: latest index.html file is present
  copy:
    src: files/index.html
    dest: /var/www/html/

- name: httpd is started
  service:
    name: httpd
    state: started
```



Apply x times...



With Automation everything is...

- Documented / Always up2date
- Abstraction of infrastructure
- Knowledge of the state of your IT
- State is always enforced
- Auditable / Traceable changes
- Compliance
- Reproducible
- Reporting
- Easy to integrated into changemangement process - standard(s) procedures
- Fast and close to production
- Less errors because standardized environment and unit tests
- IT staff spends their time on valuable things that engage their abilities



BUSINESS



DEV



QA/SECURITY



IT OPERATIONS

PROVISIONING AS CODE

CLOUD INFRASTRUCTURE AUTOMATION

TOOLS: OPENSTACK & TERRAFORM

- Provision networks, security groups, etc. from code
- Provision VMs and orchestrate their deployment from code
- Support for multiple cloud providers and third party tools

BENEFITS

- Full control of all cloud resources
- Reproducibility of environments
- Easy decommissioning



PROVISIONING AS CODE

CASE STUDY: PROVISION A DEVELOPMENT ENVIRONMENT

OBJECTIVE

Deploy a Development Environment in a fully automated way

IMPLEMENTATION

Terraform is used to:

- provision networks and nodes, integrate with Puppet/Satellite
- set up Github/Gitlab groups and rights
- set up observability with Grafana dashboard

BENEFITS

- Fully automated, extremely fast deployment of environments
- All resources managed in one location



CONFIGURATION MANAGEMENT

MAINTAINING MACHINE STATE

TOOLS: TWO APPROACHES

- Agent approach (Cfgmgmt): Cfengine, Puppet, Chef
- Agentless approach (Deploy): Ansible, Salt

BENEFITS

- No manual management of nodes
- Reproducibility of nodes
- Inventory
- More confidence in environment and deployments



CONFIGURATION MANAGEMENT

CASE STUDY: MANAGE SERVER FLEET WITH PUPPET

OBJECTIVE

Manage configuration for hundreds of servers

IMPLEMENTATION

A Puppet Server deployed as part of Red Hat Satellite/Foreman

BENEFITS

- Very fast setup of new nodes
- Dynamic, up-to-date inventory
- Nodes are kept up-to-date with their target configuration
- Centralized, automated patch management
- Possibility to use Content Views for Configuration QA



DEPLOYMENT AS CODE

DEPLOYING COMPLEX SOFTWARE

TOOLS: ANSIBLE

- Orchestration of multi-node deployments with dependencies

BENEFITS

- Reproducibility of deployments
- More confidence in deployments



ANSIBLE

DEPLOYMENT AS CODE

CASE STUDY: OPENSIFT DEPLOYMENT

OBJECTIVE

Deploy an OpenShift cluster, with various node roles and inter-nodes dependencies.

IMPLEMENTATION

Red Hat provides Ansible playbooks to deploy an OpenShift cluster.

BENEFITS

- Fully automated OpenShift deployment
- Much easier/faster to deploy new nodes in the cluster



ANSIBLE



OPENSIFT

CONTAINER ORCHESTRATION

CONTAINERIZED APPLICATION DEPLOYMENT AS CODE

TOOLS: OPENSIFT & HELM

- Gather all Kubernetes objects in one package

BENEFITS

- Distribute applications as catalog entries
- Fully customizable packages



OPENSIFT



CONTAINER ORCHESTRATION

CASE STUDY: PUPPET STACK DEPLOYMENT WITH HELM

OBJECTIVE

Deploy a Dockerized Puppet Stack on OpenShift

IMPLEMENTATION

Helm charts for the Puppet Stack components (Puppetserver, PuppetDB, Puppetboard, R10k)

BENEFITS

- Easily deploy a full Puppet stack on OpenShift
- Parameters and secrets passed to templates



OPENSIFT



CONTINUOUS INTEGRATION

CI/CD PIPELINES AS CODE

TOOLS: JENKINS

- Continuous Integration platform
- Integrates with multiple Version Control Systems
- Allows to build pipelines
- CI/CD as Code with Jenkinsfile
- Extensible with Groovy code

BENEFITS

- Flexible way to program CI/CD tasks
- Portable to other Jenkins instances
- Pipeline approach



Jenkins

CONTINUOUS INTEGRATION

CASE STUDY: CI/CD WITH OPENSIFT

OBJECTIVE

Automatically deploy a GIS multi-tiered application from code in OpenShift

IMPLEMENTATION

Use ImageStreams with OpenShift builds (using Jenkins) and Jenkinsfile, along with triggers for deployment

BENEFITS

Fully automated pipeline workflow for developers, with automated checks and triggers.



OBSERVABILITY AS CODE

LOGS, METRICS & MONITORING

TOOLS: EFK, PROMETHEUS, GRAFANA

- Gather logs & metrics
- Provide code for data processing and dashboards

BENEFITS

- Prometheus time series provide a flexible query language
- JSON dashboards provide portability



OBSERVABILITY AS CODE

CASE STUDY: MONITOR PUPPET RESOURCES AND PERFORMANCE

OBJECTIVE

Gather metrics performance metrics for the Puppet stack, both on the server and agents.

IMPLEMENTATION

- The Puppet prometheus_reporter module provides a report plugin to collect agent data
- JMX for internal Puppet Enterprise server metrics, gathered in Prometheus

BENEFITS

Easily deployed dashboard with tuned metrics and monitoring



BACKUPS AS CODE

AUTOMATE BACKUPS OF CONTAINER VOLUMES

TOOLS: KUBERNETES AND BIVAC

- Automatic backups of PVCs
- Tune backup parameters with Labels

BENEFITS

- Allow users of container platforms to choose which Data Volumes to backup
- Allow users to tune how backups are performed



kubernetes

BACKUPS AS CODE

CASE STUDY: BACKUP DATA VOLUMES IN OPENSIFT

OBJECTIVE

Provide an API for developers to backup their Data Volumes

IMPLEMENTATION

The Bivac stack is deployed on Openshift projects to detect and backup Physical Volume Claims automatically.

BENEFITS

Backup container data where it is stored and let users tune how to perform the backup.



kubernetes

CONCLUSION

EVERYTHING AS CODE

TOOLS

- Many open-source tools to automate:
 - Systems Administration
 - Software Testing and Deployment

BENEFITS

- Increased flexibility
- Increased collaboration between teams
- DevOps everything



kubernetes