

# Devops

The art of managing infrastructure at scale



# What is Devops?

A fusion of various roles:

- Sysadmin/Operations
- Developer
- Network administrator

The role is ambiguous in nature and definition is evolving. The goal should be automate and streamline as many processes as possible.



# Why Devops is needed

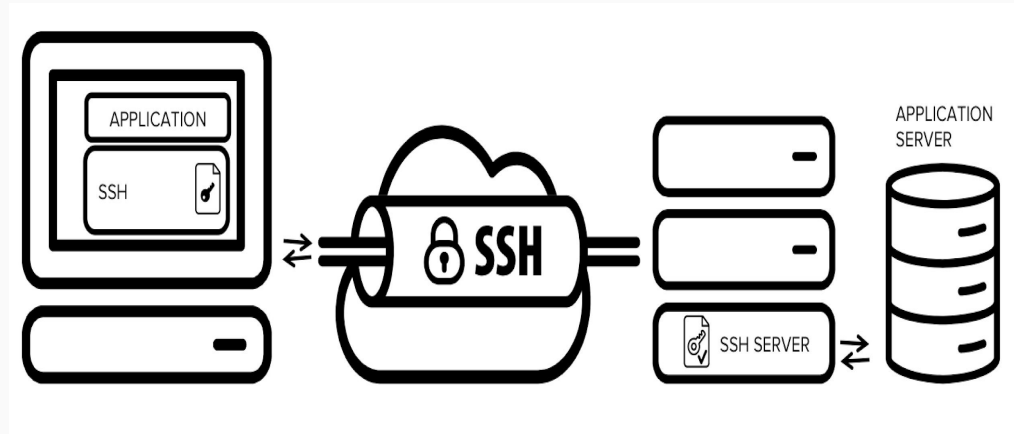
- Streamlining developer processes. More time saved for developers not having to manually deploy is time and money saved.
- Managing infrastructure manually doesn't scale up. Installing, setting up and maintaining a single server is very different from doing this for 100s of servers at once.
- Making smart infrastructure choices for both current and future infrastructure needs allowing you to seamlessly scale as per your needs.
- Automated recovery, diagnostics and reporting means peace of mind and minimising downtime and headaches that go with it.
- Manual processes are prone to failures and don't scale.

DevOps is more like a philosophical movement, not yet a precise collection of practices, descriptive or prescriptive.

Automating infrastructure

# Traditional method

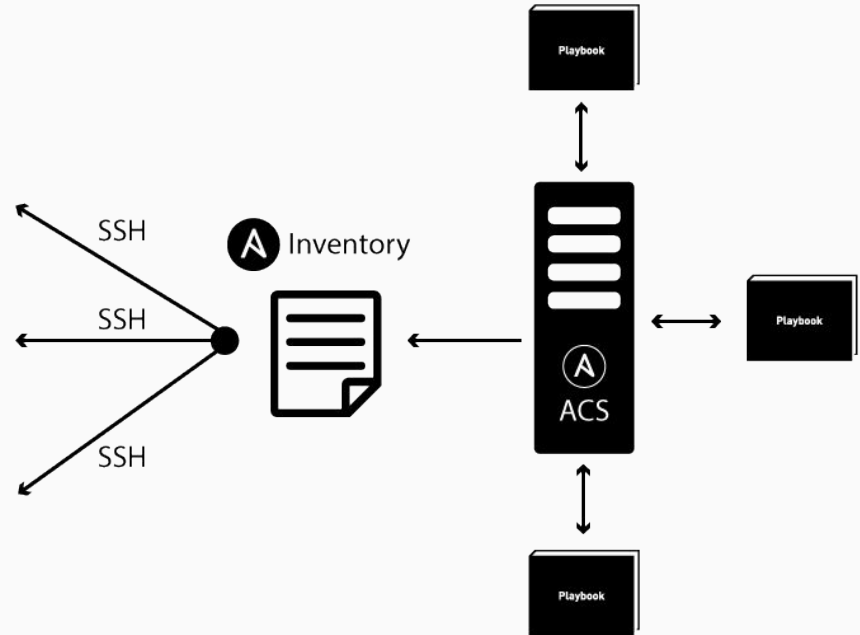
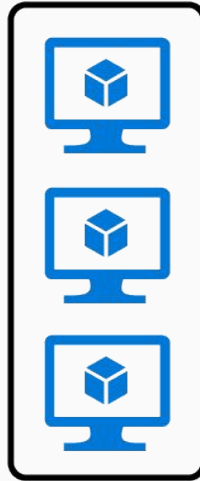
- SSHing/RDPing into each server to manually install applications or make necessary changes
- This quickly becomes a problem when you have more than a handful of servers to manage
- Manual processes are often error-prone and eat up time in repetitive processes that need to be documented



# Devops method

- Automate infrastructure right from launching a machine to setting up applications and plug it into a GUI with clickable actions
- Manage infrastructure as code. Any change in codebase can be automatically applied to all running servers
- Guarantee that changes have been made consistently

Azure VMs



# Pick your poison

There are a myriad of configuration tools which include Ansible, Chef, Salt, Terraform, Puppet, standard Bash/Python scripts



**CHEF**™



ANSIBLE



**puppet**  
labs®

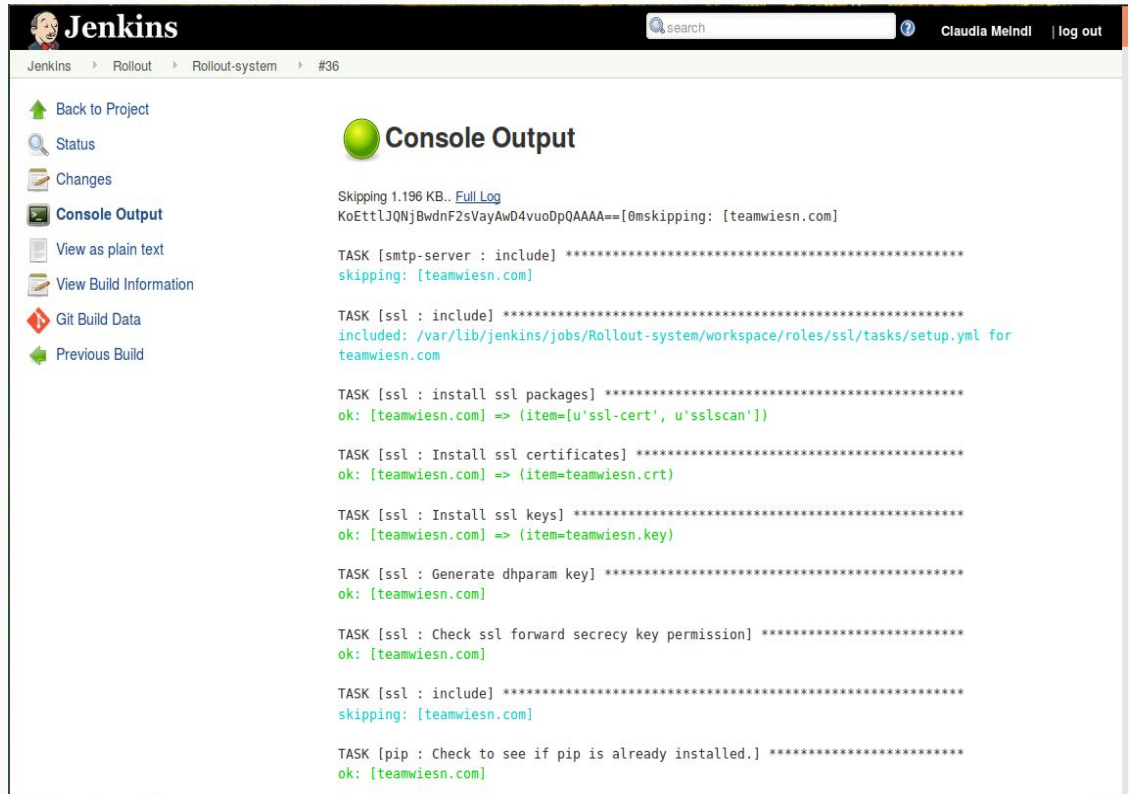


**SALTSTACK**

Build server

# Build server

- Think of this as the GUI frontend to all your automated scripts
- It also offers other functionality like pipelines, plugins, credential store etc.
- Popular open source option is Jenkins



The screenshot displays the Jenkins web interface. At the top, the Jenkins logo is on the left, and a search bar, the user name 'Claudia Melndl', and a 'log out' link are on the right. Below the header, the breadcrumb navigation shows 'Jenkins > Rollout > Rollout-system > #36'. On the left sidebar, there are several navigation links: 'Back to Project', 'Status', 'Changes', 'Console Output' (which is highlighted), 'View as plain text', 'View Build Information', 'Git Build Data', and 'Previous Build'. The main content area is titled 'Console Output' and shows the following log entries:

```
Skipping 1.196 KB.. Full Log
KoEttl3QnjBwdnF2sVayAwD4vuoDpQAAAA==[0mskipping: [teamwiesn.com]

TASK [smtp-server : include] *****
skipping: [teamwiesn.com]

TASK [ssl : include] *****
included: /var/lib/jenkins/jobs/Rollout-system/workspace/roles/ssl/tasks/setup.yml for
teamwiesn.com

TASK [ssl : install ssl packages] *****
ok: [teamwiesn.com] => (item=[u'ssl-cert', u'sslscan'])

TASK [ssl : Install ssl certificates] *****
ok: [teamwiesn.com] => (item=teamwiesn.crt)

TASK [ssl : Install ssl keys] *****
ok: [teamwiesn.com] => (item=teamwiesn.key)

TASK [ssl : Generate dhparam key] *****
ok: [teamwiesn.com]

TASK [ssl : Check ssl forward secrecy key permission] *****
ok: [teamwiesn.com]

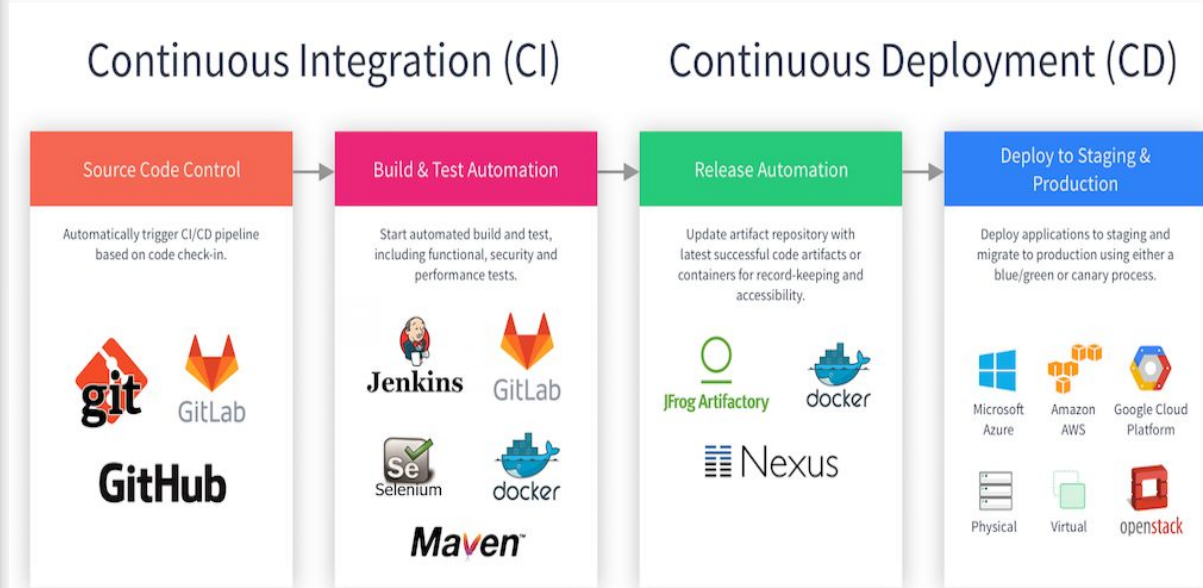
TASK [ssl : include] *****
skipping: [teamwiesn.com]

TASK [pip : Check to see if pip is already installed.] *****
ok: [teamwiesn.com]
```

Improving developer workflow

# CI/CD

- The idea is to get from code to staging/production as easy/quickly as possible
- Unit tests and user acceptance testing can be written to automate testing of software before release as well
- SCM plugins can watch for updates and build/deploy when it sees one
- Release automation would build entire application



# Leveraging “Cloud” - Why it's important

# Advantages of “Cloud”

- Easy ability to bring up/down servers with pay per hour model. Allowing you to add/remove servers as necessary & increasing flexibility of setup
- Some companies may offer you “managed” services, allowing you to not worry about maintaining them. Eg. AWS RDS
- Most Cloud providers have a CLI tool for managing resources. This would aid in automation
- Practically infinite scalability without having to over-provision from day 1

# Devops and Open Source

# Devops FOSS

- Many of the provisioning tools are either fully or partly open source
- Open source tools are very well integrated among themselves
- The ecosystem allows you to build complex products without paying for the code or support



# Devops for Network Administrators

# Network automation

- Many companies are moving network logic into software for more control eg. VMware, cloud companies like Azure, Open vSwitch
- Traditional switches/routers can be automated as well since all of them have Python and SSH enabled. Ansible has modules for the same
- Vendors like Cisco are focusing on ease of automation of network hardware

## Networking Hosts

Python code is executed locally



CONTROL NODE

local executions



NETWORKING HOSTS

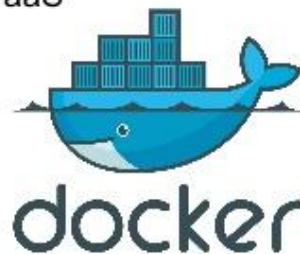
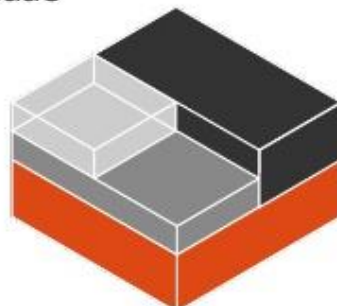
Ansible Playbooks use:  
**connection: local** and a provider  
*or*  
**connection: network\_cli**  
*or*  
**connection: netconf**

# The rise of containers

Docker, Kubernetes has taken over. LXC/LXD is also a very good option to replace virtual machines. The community tooling is excellent and allows for

## Docker vs LXD?

- VM alternative
- Multiple Process Container
- Operating System
- Online Resource Control
- IaaS
- App Encapsulation
- Single Process container
- Development
- HUB
- PaaS



“The most powerful tool we have as  
developers is automation.”



# Thanks!

Contact me:

LinkedIn: varunpriolkar

me@varunpriolkar.com

<https://varunpriolkar.com>

