



Advanced Ansible: better infrastructure

Justin W. Flory
justinwflory.com



What we'll talk about

- Quick introduction to Ansible
- *For fun*: Managing Linux workstations
- *For real*: Managing infrastructure and preserving sanity

github.com/jwflory/infrastructure
github.com/jwflory/swiss-army



What is Ansible?

SIMPLE

- Human readable automation
- No special coding skills needed
- Tasks executed in order
- Get productive quickly**

POWERFUL

- App deployment
- Configuration management
- Workflow orchestration
- Orchestrate the app lifecycle**

AGENTLESS

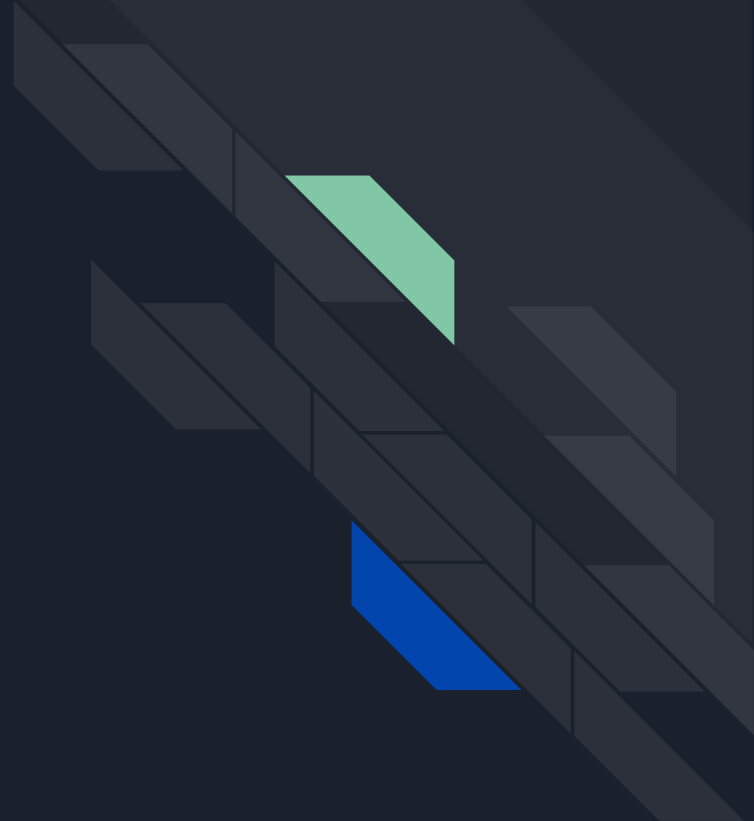
- Agentless architecture
- Uses OpenSSH and WinRM
- No agents to exploit or update
- Predictable, reliable and secure**

```
1 ---
2 - name: synchronize horizon
3   hosts: horizon
4   become: yes
5
6   roles:
7     - base/centos-7
8     - firewalld
9     - influxdb
10    - nginx
11    - php-fpm
12    - sshd
13    - yum-cron
14    - irc/limnoria
15    - sites/blog.justinwflory.com
16    - sites/crystalcraftmc.com
17    - sites/horizon.justinwflory.com
18    - sites/i.jwf.io
19    - sites/mirror.justinwflory.com
```

```
51 - name: git clone/pull RITlug/teleirc
52   git:
53     repo: "https://github.com/RITlug/teleirc.git"
54     dest: "/usr/lib64/teleirc/{{ item.value.cn }}"
55     version: "{{ item.value.version }}"
56     force: yes
57     loop: "{{ bots|dict2items }}"
58     notify:
59       - install build dependencies
60       - install runtime dependencies with yarn
61       - remove build dependencies
62       - restart teleirc
63
64 - name: "set file permissions to teleirc:teleirc in /usr/lib64/teleirc/"
65   file:
66     state: directory
67     recurse: yes
68     path: "/usr/lib64/teleirc"
69     owner: teleirc
70     group: teleirc
71
72 - name: add templated configuration
73   template:
74     src: env
75     dest: "/usr/lib64/teleirc/{{ item.value.cn }}/.env"
76     mode: 0640
77     owner: teleirc
78     group: teleirc
79     seuser: system_u
80     loop: "{{ bots|dict2items }}"
81     notify:
82       - restart teleirc
```

Ansible uses **YAML** (YAML Ain't Markup Language)

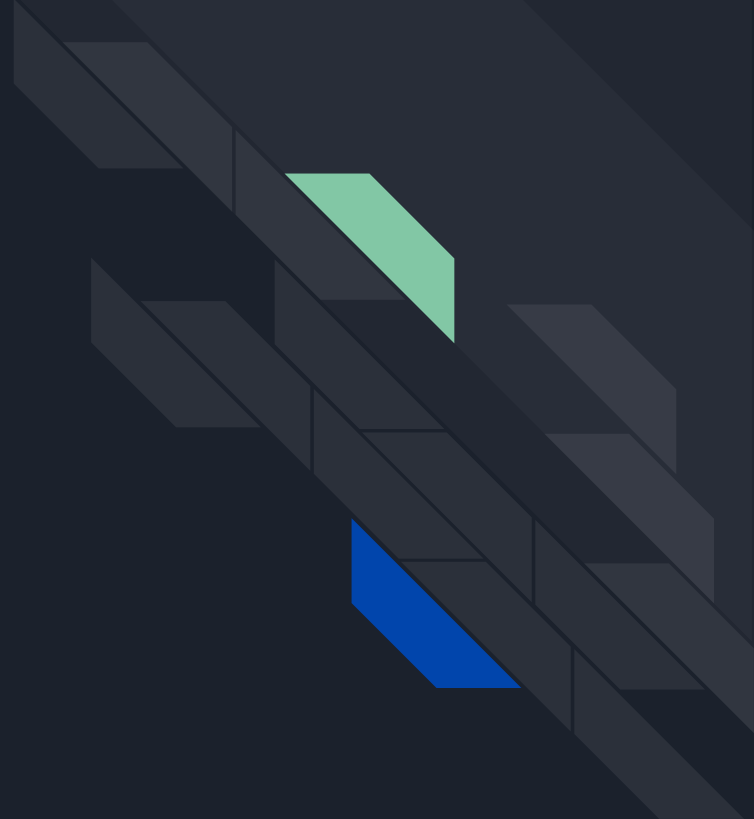
Managing Linux workstations



Introducing jwflory/swiss-army

- Ansible project to **provision and set up** Linux workstations with preferred configurations
- Divided in two parts: **system** and **app** roles
- Create custom configuration by **adding roles to playbooks**
- Why is it useful?
 - Maintain configuration between computers or across a fleet of systems
 - Easy to opt in and out based on environment (e.g. some apps not available on some OS's)
 - Sharing configurations with others is easy – share a link to GitHub

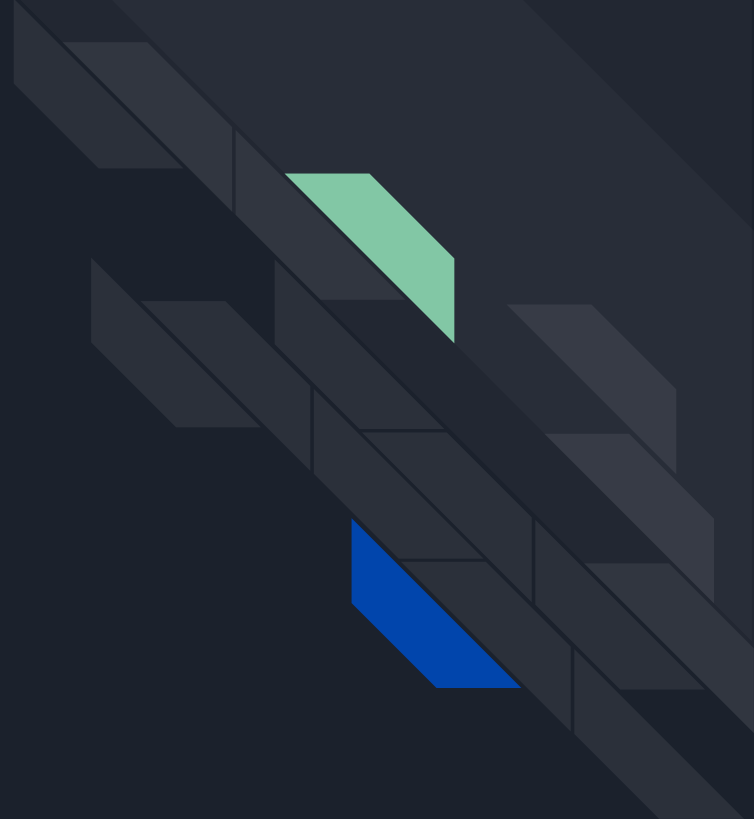
Managing infrastructure and sanity



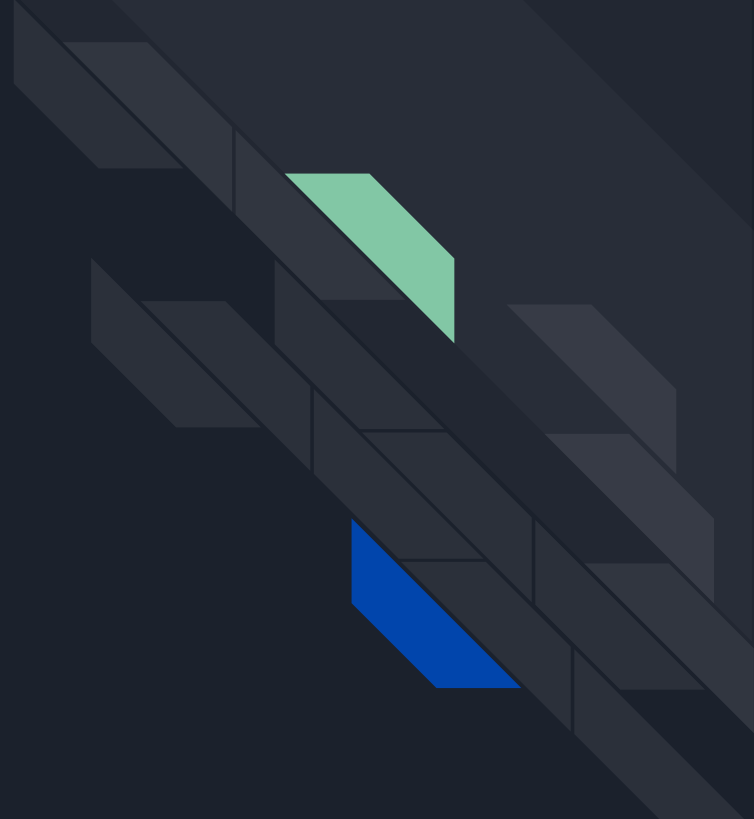
Introducing jwflory/infrastructure

- Ansible project to:
 - **Deploy applications** and services to staging or production environments
 - **Create workflow** around introducing changes
- Like swiss-army: create custom configurations by **adding roles to host-based playbook**
 - Likely better ways to do this at scale – client-server model is dated

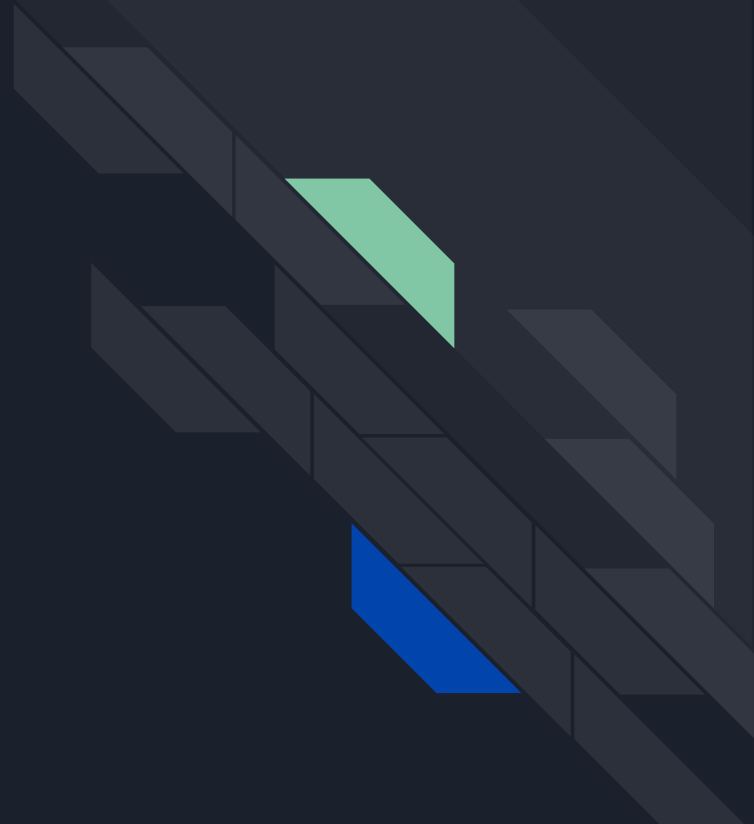
Why is it useful?



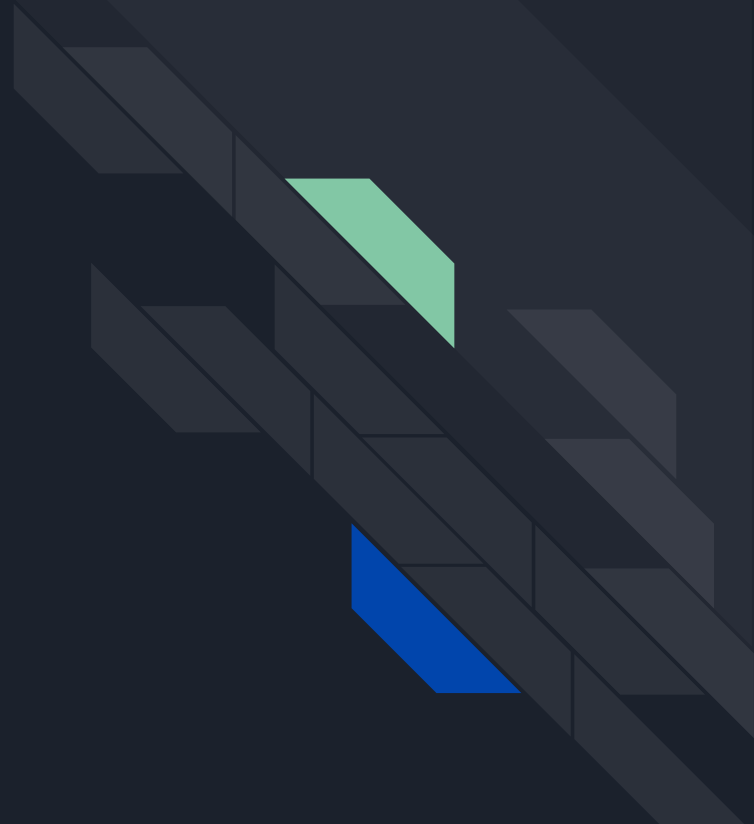
STOP: wasting time in
the terminal



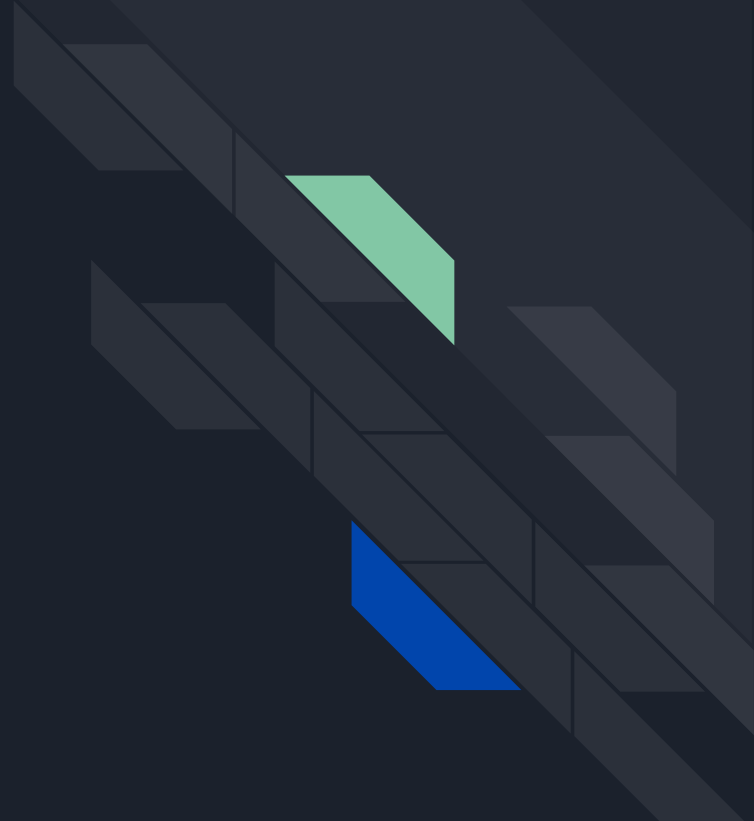
STOP: pushing to prod



START: testing changes



START: catching
problems in code review





[**LIVE**]

Because untested live demos are *always* a great idea.