Foreman and Ansible

Pulling together

Adam Růžička <aruzicka@redhat.com>, aruzicka:matrix.org and discourse

whoami

- aruzicka on most platforms
  - adamruzicka on github because someone beat me to the short variant
- Working on Foreman for the last ~10 years
- Mostly remote execution and the tasking system

Agenda

- History
- How pull mode works
- The idea
- Demo
- More sophisticated approach
- Yet another demo
- What would, could be made to and would not work
- Q & A

History

- Foreman 1.8 - Ansible
- Foreman 1.9 - SSH
- Foreman 1.16 - Ansible as a remote execution provider
- Foreman 1.22 - ansible-runner
- Foreman 3.1 - Pull mode

How pull mode works - Foreman side

- From user's point of view, everything should work the same way it did with push
- Everything should just work™
- Pull mode is only applicable Script provider jobs
- To be extra clear - Ansible always uses SSH
    - Or whatever ansible_connection variable is set to

How pull mode works - Proxy side

- When configured for pull-mqtt mode
- Proxy runs a MQTT broker
- When proxy receives a job from foreman, it:
    - Stores it
    - Notifies the target host over MQTT
    - Waits for the host to pick the job up
    - Receives updates from the host
    - Eventually finishes

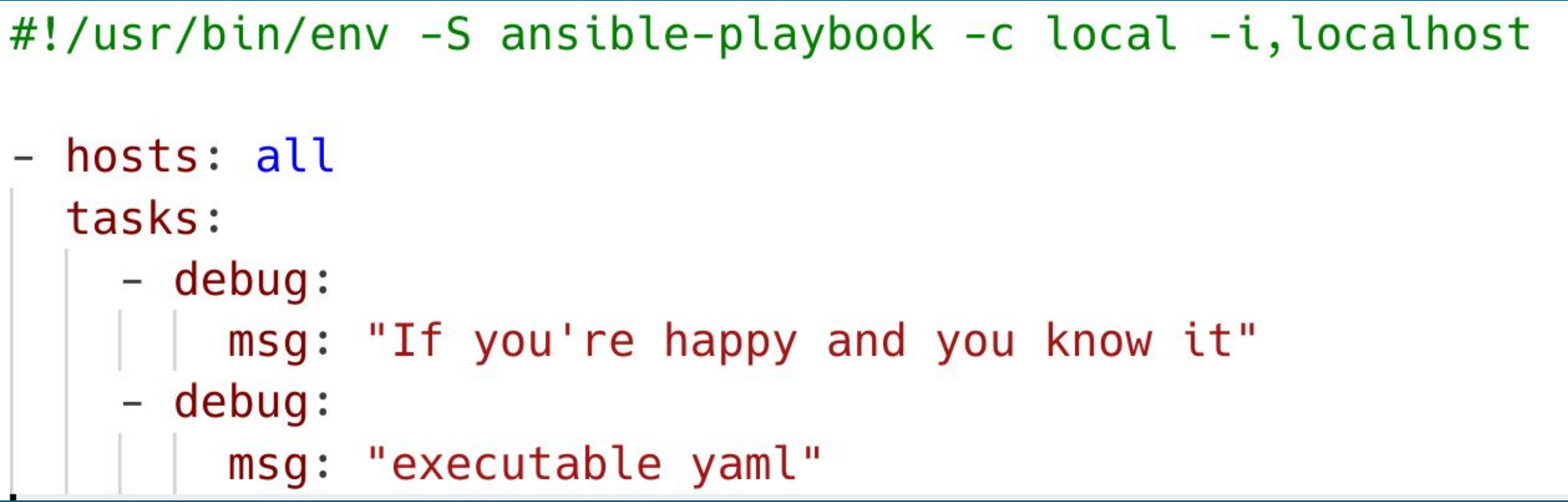
How pull mode works - Client side

- Clients run yggdrasil
- Yggdrasil runs our own worker
- Yggdrasil connects to MQTT and listens for messages
- Upon receiving a message, yggdrasil downloads the thing to run over HTTPS and passes it to the worker
- Upon receiving a message, the worker:
  - Writes the script to a file, sets the executable bit on it
  - Runs the script
  - Sends updates to the smart proxy

The idea

- We have a client that can execute anything, as long as the host has an interpreter for it.
- Isn't ansible "just" an interpreter?

The idea - executable YAML



```
#!/usr/bin/env -S ansible-playbook -c local -i,localhost

- hosts: all
  tasks:
    - debug:
        msg: "If you're happy and you know it"
    - debug:
        msg: "executable yaml"
```

DEMO

More sophisticated approach

- Don't force to use users to use script provider with ansible shebang
- Have smart proxy wrap the playbook into a script that would execute the playbook locally
- Pass the wrapped playbook to the client we already have

Yet another DEMO

What would work?

- "The essentials"
  - Running, cancelling, live output, effective user changing
- What way we deal we try to match ansible output to hosts is a little bit clunky, in this scenario it would be much better

What would definitely not work?

- Orchestration
    - With push, we run one ansible-runner per group of hosts
        - Ansible can be used to orchestrate within the group
        - But it is sort of unpredictable which hosts will end up grouped together
    - With pull, this wouldn't be possible as each host would be running it's own instance of ansible
        - Just like we had before introducing ansible-runner
        - On the flip side, it would "appear faster" as slow hosts would not block others in their group

What could be made to work?

- Roles, collections and dependencies in general
  - Right now the smart proxy has these on its filesystem
  - We'd have to have some sort of mechanism to get the relevant things to the remote hosts
- Config report callback
  - With push, a callback plugin is configured on the smart proxy and ansible calls to Foreman directly
  - With pull, the callback would have to be configured
    - And the client would have to have direct visibility to Foreman
      - Or the smart proxy would have to act as a real proxy
    - Also, certificates

My questions to you

- Is pull mode for ansible something that you would even want?
- Is solving the roles use case essential to you or is "ansible as a shell replacement" good enough?
- Is report callback something you need?

Q & A

Thank you

Adam Růžička <aruzicka@redhat.com>, aruzicka:matrix.org and discourse