

Remote management - Ansible as an example

Kosti Korhonen

All the copied output (O1, O2, O3, O4) with analysis

```
provision@faa9c948bdd2:~/ansible01$ ansible-playbook gitplaybook.yaml

PLAY [webservers] *****

TASK [Gathering Facts] *****
ok: [ansi01]

TASK [Install git] *****
changed: [ansi01]

TASK [Query uptime] *****
changed: [ansi01]

TASK [debug] *****
ok: [ansi01] => {
  "msg": [
    " 13:43:56 up  6:04,  1 user,  load average: 0.49, 0.21, 0.13"
  ]
}

PLAY RECAP *****
ansi01                : ok=4    changed=2    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```

O1: First run

I would assume the uptime-command's output is mostly fine. Nothing really to compare it to yet as this is the first run. The uptime being so high seems a bit odd to me. If I read it correctly it is saying it has been up for 6h and 4mins. With further investigation, all my containers seem to have the same uptime. It seems like the displayed uptime is the uptime of the host and the kernel does not track individual containers' uptime.

```
provision@faa9c948bdd2:~/ansible01$ ansible-playbook gitplaybook.yaml

PLAY [webservers] *****

TASK [Gathering Facts] *****
ok: [ansi01]

TASK [Install git] *****
ok: [ansi01]

TASK [Query uptime] *****
changed: [ansi01]

TASK [debug] *****
ok: [ansi01] => {
  "msg": [
    " 13:44:35 up  6:04,  1 user,  load average: 0.48, 0.23, 0.14"
  ]
}

PLAY RECAP *****
ansi01                : ok=4    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```

O2: Second run

Uptime's output seems okay. I had not gone over one minute before the last run so the uptime is the same. Current time changes, as it should, and load times seem reasonable to me. Also, something to notice from the playbook run is that the git install task went from changed to ok as it did not need to install the git in this run for this node.

```

provision@faa9c948bdd2:~/ansible01$ ansible-playbook gitplaybook.yaml

PLAY [webserver] *****

TASK [Gathering Facts] *****
The authenticity of host '172.17.0.4 (172.17.0.4)' can't be established.
ECDSA key fingerprint is SHA256:fxFHYWI0onfBY3fidkRVw+y8IITCorditWSIKNeXGoA.
Are you sure you want to continue connecting (yes/no/[fingerprint])? ok: [ansi01]
yes
ok: [ansi02]

TASK [Install git] *****
ok: [ansi01]
changed: [ansi02]

TASK [Query uptime] *****
changed: [ansi01]
changed: [ansi02]

TASK [debug] *****
ok: [ansi01] => {
  "msg": [
    " 13:46:41 up  6:06,  1 user,  load average: 0.63, 0.29, 0.16"
  ]
}
ok: [ansi02] => {
  "msg": [
    " 13:46:41 up  6:06,  1 user,  load average: 0.63, 0.29, 0.16"
  ]
}

PLAY RECAP *****
ansi01      : ok=4    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
ansi02      : ok=4    changed=2    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

```

O3: Third run

Can't figure anything else to add to previous notices. Now as we get the second node in we get double the outputs.

```

provision@faa9c948bdd2:~/ansible01$ ansible-playbook gitplaybook.yaml

PLAY [webserver] *****

TASK [Gathering Facts] *****
ok: [ansi02]
ok: [ansi01]

TASK [Install git] *****
ok: [ansi02]
ok: [ansi01]

TASK [Query uptime] *****
changed: [ansi01]
changed: [ansi02]

TASK [debug] *****
ok: [ansi01] => {
  "msg": [
    " 13:47:24 up  6:07,  1 user,  load average: 0.83, 0.36, 0.20"
  ]
}
ok: [ansi02] => {
  "msg": [
    " 13:47:24 up  6:07,  1 user,  load average: 0.83, 0.36, 0.20"
  ]
}

PLAY RECAP *****
ansi01      : ok=4    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
ansi02      : ok=4    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

```

O4: Fourth run

Nothing to add. All seems as expected taking the earlier notes into consideration.

What was easy and what was difficult

I would say the hardest thing was to get familiar with completely new system (Ansible). I bet it helped to have some experience on the Docker before this. Instructions were okay an internet seemed to have a lot of information available for this. Most of my time went to try and setup and debug my own environment so that the actual task could then be done.

The writing of the configurations and playbooks seemed easily understandable and the library for different actions seemed quite large. I learned a lot of useful insight on the tool and will most probably be using it in future in CI/CD related work.