TrakCare deployment using Ansible

Go away or I will replace you with a simple yaml script

Carlos "casep" Sepulveda
casep@intersystems.com

3rd July, 2020

Plan

Motivation I

```
1  echo "##############################################"
2  INST=`instname $SITE $ENV $TYPE$VER`
3  #INST=$(/bin/ccontrol qlist | grep $SITE | grep $ENV | grep $VER | cut -d"^" -f1
       )
4  TRAKNS=`traknamespace $SITE $ENV`
5  TRAKPATH=`trakpath $SITE $ENV $TYPE$VER`
6  echo "Vanilla Trak $VER Install for $SITE : $ENV ($INST: $TRAKNS)"
7  # check if we need to do this
8  if [ -f ${TRAKPATH}/web/default.htm -a -f ${TRAKPATH}/db/data/CACHE.DAT ]; then
9    echo "Already appears to be web and databases installed"
10   exit 0
11 fi
12 # get cache password if needed
13 if [ -z "$CACHEPASS" ]; then
14   getpass "Cache Password" CACHEPASS 1
15 fi
16 # get Trak zip password if needed
17 if [ -z "$TRAKZIPPASS" ]; then
18   getpass "TrakCare .zip Password" TRAKZIPPASS 1
19 fi
20 # find installer
21 installer=`locatefilestd $VER_*_R*_B*.zip`
22 #installer=/trak/iscbuild/installers/T2016.1_20151117_1506_R1_ENXX_FULL_B3.zip
23
24
25
```

Motivation II

```
26 echo $installer
27 # check for target web/ directory
28 if [ ! -d ${TRAKPATH}/web ]; then
29   echo "FATAL - expecting \"${TRAKPATH}/web/\" to be created with appropriate
         permissions in advance" >&2
30   exit 1
31 fi
32 # install dependancies
33 osspecific depends
34 # check that expect is available
35 if [ ! -x /usr/bin/expect ]; then
36   echo "FATAL - can't find executable /usr/bin/expect" >&2
37   exit 1
38 fi
```

Listing 1: Extract of the old code.

Just impossible to keep the old code properly updated. A mess of different perl, bash, expect and similar scripts.

Background

Ansible?

"Ansible is a radically simple IT automation engine that automates cloud provisioning, configuration management, application deployment, intra-service orchestration, and many other IT needs" [1]
"...the simple, yet powerful IT automation engine that thousands of companies are using to drive complexity out of their environments and accelerate DevOps initiatives" [6]

Why Ansible?

"It uses no agents and no additional custom security infrastructure, so it's easy to deploy - and most importantly, it uses a very simple language (YAML, in the form of Ansible Playbooks) that allow you to describe your automation jobs in a way that approaches plain English" [1]

Simple sintax

```
1 ---
2 - name: This is a hello-world example
3   hosts: ansibleclient01
4   tasks:
5     - name: Create a file called '/tmp/testfile.txt' with the content 'hello
          world'.
6       copy:
7         content: hello world
8         dest: /tmp/testfile.txt
```

Listing 2: Example of sintax for a task.

Simple to run

```
1  $ ansible-playbook HelloWorld.yml
2
3  PLAY [This is a hello-world example] ********************************************
4
5  TASK [setup] ********************************************************************
6  ok: [ansibleclient01]
7
8  TASK [Create a file called '/tmp/testfile.txt' with the content 'hello world'.]
9  changed: [ansibleclient01]
10
11 PLAY RECAP *********************************************************************
12 ansibleclient01 : ok=2 changed=1 unreachable=0 failed=0
```

Listing 3: Example of playbook execution.

Simple to debug (maybe)

```
$ ansible-playbook helloworld.yml
ERROR! We were unable to read either as JSON nor YAML, these are the errors we
      got from each:
JSON: No JSON object could be decoded

Syntax Error while loading YAML.
  mapping values are not allowed in this context

The error appears to be in '/tmp/helloworld.yml': line 8, column 13, but may
be elsewhere in the file depending on the exact syntax problem.

The offending line appears to be:

      content: hello world
        dest: /tmp/testfile.txt
            ^ here
```

Listing 4: Example of failed execution.

Want to know more?

Intro to Playbooks[4].

Deploying TrakCare

```
$ ansible-playbook site.yml
...
TASK [web_server : Set httpd_can_network_connect flag] ************************
ok: [web02]
ok: [web01]

TASK [web_server : remove the contents of no longer required installers] ******
changed: [web02]
changed: [web01]

TASK [web_server : reboot the machine to confirm first round of changes] ******
changed: [web02]
changed: [web01]

TASK [web_server : wait for server web01 to come back online] *****************
ok: [web01 -> localhost]
ok: [web02 -> localhost]

PLAY RECAP ********************************************************************
db01   : ok=124 changed=23 unreachable=0 failed=0 skipped=35 rescued=0 ignored=0
web01  : ok=92  changed=12 unreachable=0 failed=0 skipped=18 rescued=0 ignored=0
web02  : ok=92  changed=12 unreachable=0 failed=0 skipped=18 rescued=0 ignored=0
```

Listing 5: TrakCare up and running!

Components of the the project I

```
1  |-- ansible.cfg
2  |-- db_server.yml
3  |-- group_vars
4  |   `-- all
5  |       |-- vars-db_server.yml
6  |       |-- vars-standard-defaults.yml
7  |       `-- vars-start-here.yml
8  |-- hosts
9  |-- roles
10 |   |-- db_server
11 |   |   |-- tasks
12 |   |   |   |-- configure-nfs-db.yml
13 |   |   |   |-- configure-SELinux-tc.yml
14 |   |   |   |-- deploy-trakcare.yml
15 |   |   |   |-- install-iris-silent.yml
16 |   |   |   |-- main.yml
17 |   |   |   |-- transfer-iris-installer.yml
18 |   |   |   |-- transfer-trakcare-installer.yml
19 |   |   |   `-- tuning-iris.yml
20 |   |   `-- templates
21 |   |       `-- iris.service
22 |   |-- os_common
23 |   |   |-- tasks
24 |   |   |   |-- configure-Firewall.yml
25 |   |   |   |-- configure-RedHat.yml
26 |   |   |   `-- main.yml
```

Components of the the project II

```
27 |   |   `-- templates
28 |   |       `-- rhel7.ntp.conf.j2
29 |   |-- summary
30 |   |   `-- tasks
31 |   |       `-- main.yml
32 |   `-- web_server
33 |       |-- tasks
34 |       |   |-- configure-nfs-web.yml
35 |       |   |-- deploy-webgateway.yml
36 |       |   |-- hack-webgateway.yml
37 |       |   |-- install-RedHat-Apache.yml
38 |       |   |-- main.yml
39 |       |   `-- restart_server.yml
40 |       |-- templates
41 |       |   |-- template_csp_ini.ini
42 |       |   `-- template_tc_nspace_alias.conf
43 |       `-- vars
44 |           |-- vars-mpm.yml
45 |           |-- vars-RedHat-Apache.yml
46 |           `-- vars-webgateway.yml
47 |-- site.yml
48 `-- web_server.yml
```

Listing 6: Files in the project.

ansible.cfg

Holds the definition for your **inventory**. Ansible works against multiple managed nodes or "hosts" in your infrastructure at the same time, using a list or group of lists know as inventory[2].
Also stored here, the definition for the **remote_user**. This user is the one used to access the remote servers during deployment.

group_vars I

While automation exists to make it easier to make things repeatable, all systems are not exactly alike; some may require configuration that is slightly different from others. In some instances, the observed behavior or state of one system might influence how you configure other systems.
For example, you might need to find out the IP address of a system and use it as a configuration value on another system.
Ansible uses variables to help deal with differences between systems[5].
A variable is defined in the format:
name_of_the_variable: value
And it is referenced in the script as:
name_of_the_variable

group_vars II

Example of variables:

```
1  # Iris details
2  webgateway_installer: WebGateway-2020.1.0.215.0-lnxrhx64.tar.gz
3  webgateway_version: WebGateway-2020.1.0.215.0-lnxrhx64
4  # The need of version and minor is due to the difference between the
5  # installer and the folder that contains
6  iris_version: IRISHealth-2019.1.0.510
7  # The "minor version" is required due to the difference between the name and
8  # the contents of the file with the installer
9  iris_minor_version: .4-lnxrhx64
10 iris_install_kit_filename: "{{ iris_version }}{{ iris_minor_version }}.tar.gz"
11 iris_install_kit_path: "{{ path_to_install }}/{{ iris_version }}{{
       iris_minor_version }}"
12 iris_install_csp_kit_filename: "{{ iris_version }}-csp.tar.gz"
13 trak_installer_filename: T2019_20200110_1334_R5_SYS_FULL_B506.zip
```

Listing 7: Variables for a TrakCare 2019 deployment.

hosts

Here you can find the definition for the groups of servers.
Each group will have different tasks associated to it.

```
1 [web_servers]
2 web01
3 web02
4
5 [db_servers]
6 db01
```

Listing 8: Example definition for hosts and groups.

More information[2].

site.yml

This file holds the main configuration for the site.
It includes, which roles to load and which hosts to manage.
More definition on importing playbooks[3].

roles

Roles are ways of automatically loading certain vars_files, tasks, and handlers based on a known file structure. Grouping content by roles also allows easy sharing of roles with other users.[3].

```
site.yml
webservers.yml
fooservers.yml
roles/
    common/
        tasks/
        handlers/
        files/
        templates/
        vars/
        defaults/
        meta/
    webservers/
        tasks/
        defaults/
        meta/
```

Listing 9: Role Directory Structure.

TrakCare 2019

TrakCare 2019 configuration:

```
1 iris_version: IRISHealth-2019.1.0.510
2 # The "minor version" is required due to the difference between the name and
3 # the contents of the folder within the installer
4 iris_minor_version: .4-lnxrhx64
5 trak_installer_filename: T2019_20200110_1334_R5_SYS_FULL_B506.zip
```

Listing 10: All the changes required for a 2019 deployment.

TrakCare 2020

TrakCare 2020 configuration:

```
1 iris_version: IRISHealth-2020.1.0.215
2 # The "minor version" is required due to the difference between the name and
3 # the contents of the folder within the installer
4 iris_minor_version: .0-lnxrhx64
5 trak_installer_filename: T2020_20200401_0909_R0_SYS_FULL_B14.zip
```

Listing 11: All the changes required for a 2020 deployment.

References

References I

How Ansible Works.
https://www.ansible.com/overview/how-ansible-works.

How to build your inventory.
https://docs.ansible.com/ansible/latest/user_guide/intro_inventory.html.

Import a playbook.
https://docs.ansible.com/ansible/latest/modules/import_playbook_module.html.

References II

Intro to Playbooks.
https://docs.ansible.com/ansible/latest/user_guide/playbooks_intro.html.

Using Variables.
https://docs.ansible.com/ansible/latest/user_guide/playbooks_variables.html.

Why Ansible?
https://www.ansible.com/overview/it-automation.

Built using LaTeX with 100% recycled electrons