Ansible for z/os

Owned by Ketevan

Jul 18, 2024 • 2 min read • 1 person viewed

This is a record of my own experience with ansible and zxplore. Do not rely on this alone: search online, and come to this guide if you are stuck.

Installation

1. Ansible is designed to run on LINUX Systems, so for windows you have to install windows

- sudo apt update
- sudo apt upgrade

From this point on, everything is done on the ubuntu terminal.

2. Install Ansible (you need python and pip already)

3. Install ansible IBM Z collection

ansible-galaxy collection install ibm.ibm_zos_core

Pay attention to which versions are compatible.

Version	Controller	Managed Node	G
1.10.x	• ansible-core >=2.15.x • Ansible >=8.0.x • AAP >=2.4	• z/OS V2R4 - V2Rx • z/OS shell • IBM Open Enterprise SDK for Python • IBM Z Open Automation Utilities >=1.3.0	2
1.9.x	• ansible-core >=2.14 • Ansible >=7.0.x • AAP >=2.3	• z/OS V2R4 - V2Rx • z/OS shell • IBM Open Enterprise SDK for Python • IBM Z Open Automation Utilities 1.2.5 - 1.2.x	0

4. Install IBM Open Enterprise SDK for Python.

Get the pax file and unpax it.

5. You might need to use a public key.

Inventory File

C directory for it.

I

r

name_the_host :

ansible_host: 204.90.115.200

Ansible_user: your_user_name

How to get the python path?

Manually establish a connection to zxplore (ssh Zxxxxx@204.90.115.200)

Once you have connected type the command

Echo $path

From the long path you will get, you need to guess which path is python:

/z/z50837 > echo $PATH

/VERSYSB/usr/lpp/IBM/cvg/go/bin:/usr/lpp/IBM/cyp/v3r9/pyz/bin:/usr/lpp/IBM/zoautil/bin:/usr/lpp/db2c10/jdbc/bin:/bin:/z/bin:/global/bin:.:/usr/lpp/java/J17.0_64/bin:/usr/lpp/zLiberty/V8R5/wlp/bin:/z/z50837/bin:/z/rocket/tools/bin:/usr/lpp/ported/bin::/usr/lpp/perl/bin:/usr/lpp/IBM/cnj/v18r0/IBM/node-latest-os390-s390x/bin:/usr/lpp/IBM/cnj/v18r0/njsc/bin:/global/node/bin:/usr/lpp/mqm/V9R2M0/web/bin:/usr/lpp/IBM/cobol/igyv6r4/bin

/z/z50837 > ls /usr/lpp/IBM/cyp/v3r9/pyz/bin

This was the path for me, and should look similar to others too.

In the inventory file copy the path and add /python3 at the end, because without it, the system thinks the path you provide is an executable file.

In general, not all hosts need the path to be specified.(For Zxplore you will probably need to specify)

M ost 'address' and the user name.

1. Execute command

ansible <group_name_from inventory_file> -m ping -i <inventory.yml> -vvv

-vvv is to get a long message, and is not necessary, but could be helpful to identify a problem

2. During the process if you get command timeout errors, it's probably a different issue.
   Increase value of command timeout : EXPORT ANSIBLE_PERSISTENT_COMMAND_TIMEOUT=

   default is 30 second, make a significant increase: I set to 240

   And add -vvv to commands to see a long message.

Ping with a playbook

We will use the inventory file for this and create another .yml file, which is a playbook.

In a playbook you need to give a name to the play (program) , specify hosts , name the tasks and specify the tasks:

-(space)name: pingi

hosts: z_host (make sure this host is defined in inventory file)

tasks:

-(space)name: Pinging host z_host

ping:

register: result

Syntax matters: so pay attention to dashes, spaces and indentation.

N ible-playbook command.

a

P [ping]

****

```
*******************************

TASK [Gathering Facts]
```

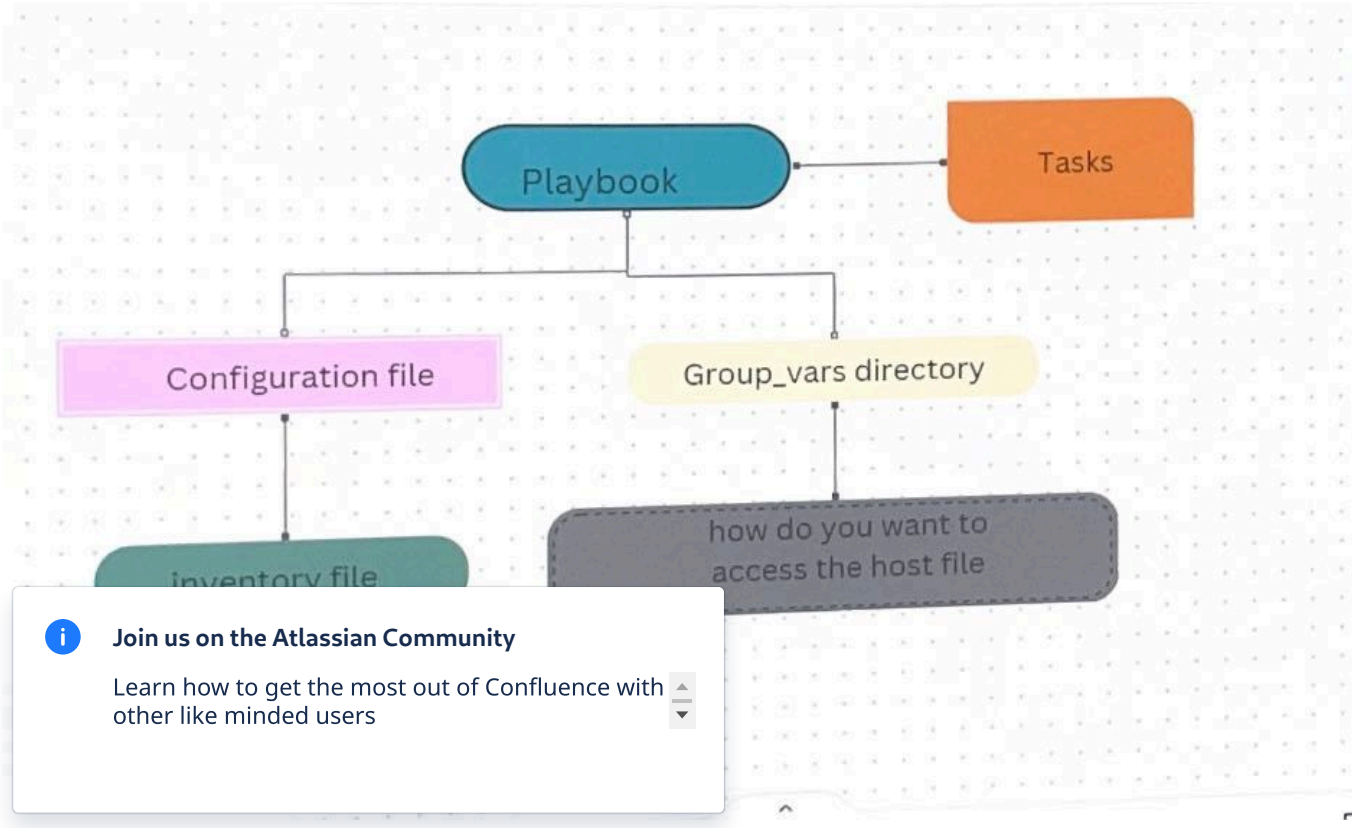


```
ok: [z_host]

TASK [Pinging host z_host]
*********************************************************************************
*****************
ok: [z_host]

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

Execute a real task



variables and of course the **playbook**

Configuration File

Ansible comes with a default configuration file, but that probably won't do. It is good idea to create a configuration file within your working directory.

I am pretty sure if you name the file ansible.cfg withing the working directory it will be used as configuration file. Use **ansible --version** to check what configuration file you are using.

For the simplest configuration file you need a section header and to specify the inventory file:

```
[defaults]
inventory = ./inventory.yml <--- path to inventory file
```

This is configuration file from the IBM extended ansible course, but the simple version above also works.

```
[defaults]
host_key_checking = False
forks = 25
inventory = ./inventory
retry_files_enabled = False

[ssh_connection]
pipelining = True
```

Inventory file

Inventory file is very simple: just make sure it is named exactly how you named in configuration f

a ay:

[

204.90.115.200

why is this inventory file different from the old one? i don't know.

but this was works, the old one does not

Environment variables

This goes by a naming convention that can not be altered, or at least requires a lot of work.

create a directory withing the working directory named group_vars. **In it create a file named after the host and add .yml of course.** Here we define all the environment variables and username and password.

ansible_user: Z50837 # USER used for deploy
ansible_password: QUA31WOW # USER used for deploy

zoau_home: /usr/lpp/IBM/zoautil # Path for your ZOAU directory
python_path: /usr/lpp/IBM/cyp/v3r9/pyz # Path for your Python
ansible_python_interpreter: "{{python_path}}/bin/python3"

Most probably you can just directly copy the file.

playbook

Of course you need a playbook to actually execute:

- hosts: zxp
  gather_facts: no
  environment:
  ZOAU_ROOT: "{{zoau_home}}"
  ZOAU_HOME: "{{zoau_home}}"

/usr/lib:."

collections:

```
    - ibm.ibm_zos_core
  tasks:
[I-7] Internship Ketev...  /  Ansible for z/...
    zos_copy:
      src: /home/keti/ansible-zos/somefile
      dest: /z/z50837/ansb2-complete
~                                                                    ~
```

Here we are using on of predefined modules (Modules — Red Hat Ansible Certified Content for IBM Z documentation) and copying file from the local source to uss.

Execute

Use command

anisble-playbook <playbook_file>

Full example of a simple task

Copying a local file to uss using pre-defined module- zos_copy.

ansible.cfg

[defaults]
inventory = ./inventory.yml

inventorty.yml

[zxp]
204.90.115.200

playbook.yml


```
  hosts: zxp
  gather_facts: no
  environment:
```

```
    ZOAU_ROOT: "{{zoau_home}}"
    ZOAU_HOME: "{{zoau_home}}"
    PATH: "{{zoau_home}}/bin:{{python_path}}/bin:/bin: "
```



```
    _BPXK_AUTOCVT: "ON"

collections:
  - ibm.ibm_zos_core

tasks:
  - name: copy to USS
    zos_copy:
      src: /home/keti/ansible-zos/somefile
      dest: /z/z50837/ansb2-complete
```

Directory > **Group_vars > zxp.yml**

ansible_user: Z50837 # USER used for deploy
ansible_password: QUA31WOW # USER used for deploy

zoau_home: /usr/lpp/IBM/zoautil # Path for your ZOAU directory

#here you might need to change the version of the python you are using
python_path: /usr/lpp/IBM/cyp/v3r9/pyz # Path for your Python

ansible_python_interpreter: "{{python_path}}/bin/python3"

command

ansible-playbook <playbook.yml>

Again we are using a pre-defined module.

This playbook submits a JCL stored in PDS member on zxplore. JCL is written for a very simple COBOL code that just displays a message.


```
tasks:
  - name: pingi
    ping:
    register: result
  - name: submiteasyjcl
    zos_job_submit:
      src: Z50837.JCL(LOVEJCL)
      location: DATA_SET
    register: result
```

At this point you might have a problem is you have not installed the compatible versions. If nothing works, go back to the installation steps. I needed to downgrade my ansible core collection. You might have a same problem.