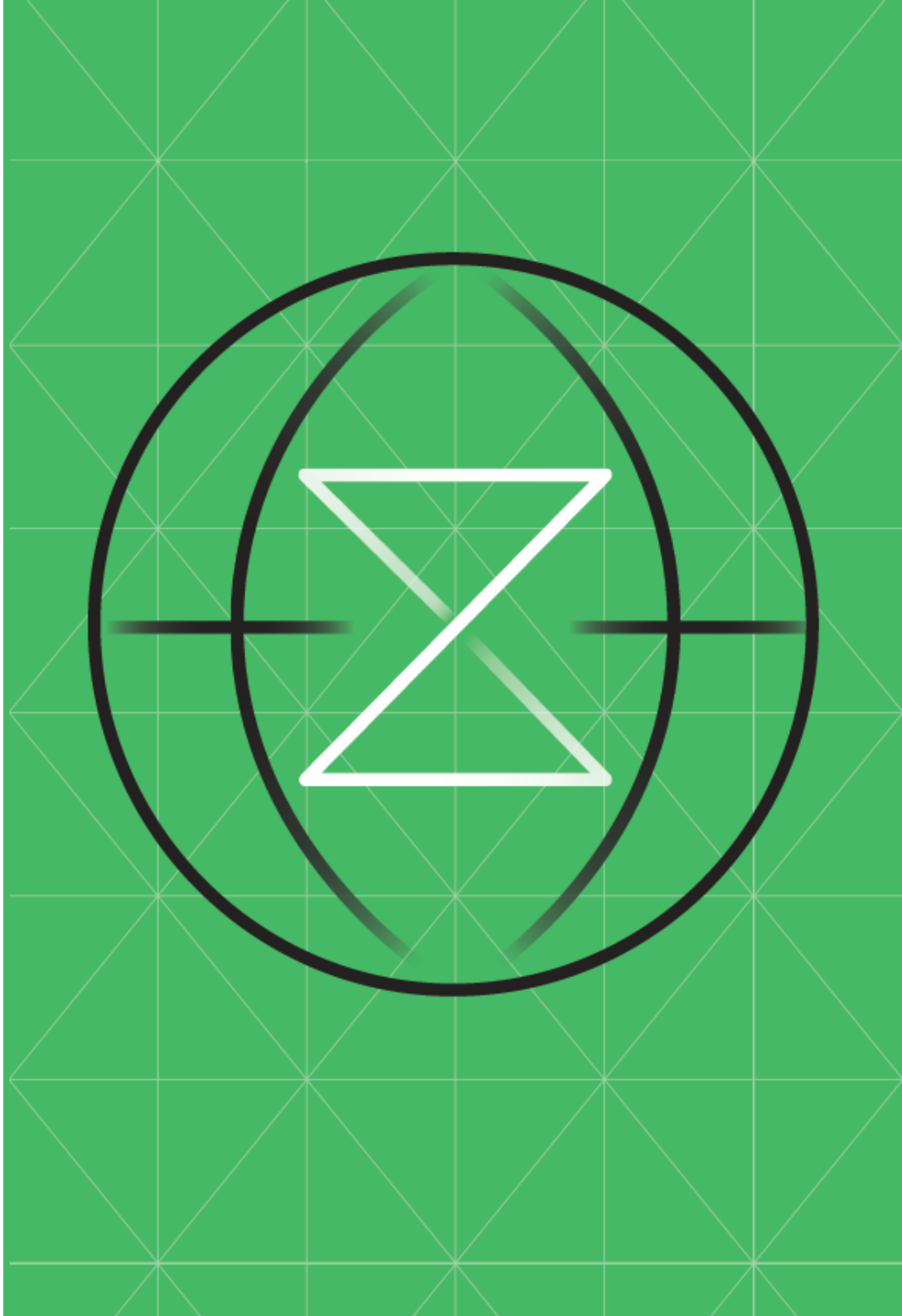
Extended

230301-1114

ANSB1

Grab an Ansible Upgrade



Z Xplore

MANAGING SYSTEMS WITH ANSIBLE

Elevate your Mainframe Experience with Ansible.

Ansible is becoming very popular as a common systems management tool across a wide variety of platforms and operating environments.

Ansible is primarily used for automating repetitive administration tasks, to help ensure they are executed correctly and consistently.

“Playbooks” define the tasks to be performed, and platform-specific modules provide the implementations needed for tasks to execute.

This challenge will provide an introduction to Ansible, and the zOS platform modules.

ANSB1|230301-1114

The Challenge

Now we get to play around in Ansible, a tool that brings it all together so you can focus on what you want to get done. This is “next-level” stuff, and you’re about to get some hands-on experience bringing it to life on this system.

Before You Begin

Save these Ansible challenges for last.

To really get the most out of Ansible, it helps if you know what’s available to you EVERYWHERE on the platform.

Investment

Steps	Duration
6	30 minutes

ANSB1|230301-1114

1 LOAD THE ANSIBLE IMAGE

In the Docker challenge, you set up the Docker extension for VSCode, and connected to a basic Ubuntu container.

You can now start working with Ansible by launching the command:

```
docker run -it icr.io/ibmzxplore/ansible:zxp202108
```

Again, you will see docker automatically pull a copy of the image into your docker runtime, and start the container.

ibmuser@Avalon ~ % `docker run -it ibmzxplore/ansible:zxp202108`

```
Unable to find image 'ibmzxplore/ansible:zxp202108' locally
zxp202108: Pulling from ibmzxplore/ansible
2d473b07cdd5: Pull complete
882e9ac78b3d: Pull complete
01427b2d9ab7: Pull complete
ed7124d72584: Pull complete
51d475619f90: Pull complete
28d349b793e6: Pull complete
4f4fb700ef54: Pull complete
7250dc8a2ea6: Pull complete
b96fe367ab79: Pull complete
23e2aacbd155: Pull complete
b8b4868e71ff: Pull complete
53a2ed1f84ed: Pull complete
41c6dda7dbad: Pull complete
Digest: sha256:1230391f272cf29d52c5c9c59a04859ac307541790d6f078ce47b003cb8fe9ef
Status: Downloaded newer image for ibmzxplore/ansible:zxp202108
[root@48a80eca25ab ~]#
```

Note that the command prompt (**[root@48a80eca25ab ~]#**) indicates that your terminal is connected to the running container.

If you are using a Macbook M1 or M2 laptop, you may see a message like this:

> WARNING: The requested image's platform (linux/amd64) does not match the detected host platform (linux/arm64/v8) and no specific platform was requested

Do not worry, it will still work!

Note this container image comes from the IBM registry (icr.io), instead of the usual/default Docker registry.

If you experience difficulties with downloads from icr.io, such as:

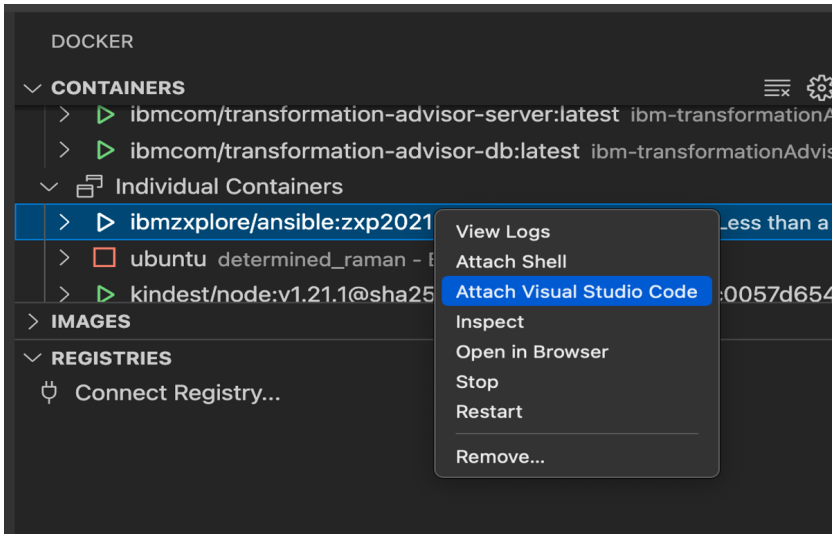
> docker: error pulling image configuration: download failed after attempts=1: denied: You have exceeded your pull traffic quota for the current month. Review your pull traffic quota and pricing plan. For more information, see https://ibm.biz/BdPdFA.

try this instead:

```
docker run -it ibmzxplore/ansible:zxp202108
```

2 ATTACH VSCODE CONSOLE

In VSCode, find the icon on the left side for “Remote Explorer”, and then locate the Ansible container that you just pulled down.

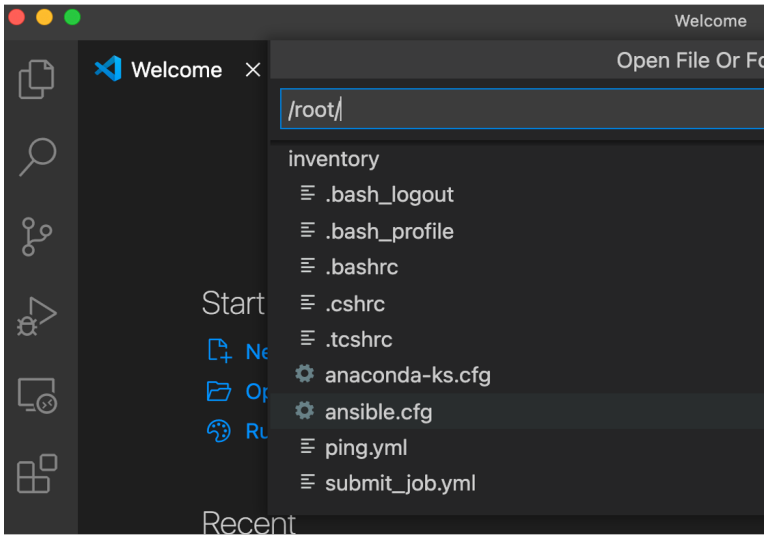


Right-click on it and select “Attach to Visual Studio Code”. This will launch a new VSCode window, which gives you a view into what is happening within this active container image.

Remember: This is the view into the container, *not* your computer’s regular filesystem.

Notice the green box at the bottom of VSCode showing you what the view is, and also that the Zowe icon is absent.

3 OPEN /ROOT/



The most relevant files you see are:

anaconda-ks.cfg
    This is a “kickstart” file made by anaconda, used to describe the environment.
ansible.cfg
    A configuration file for how you like Ansible to run. It can also be used to set file paths, polling intervals, and even the colors you want in Ansible messages.
ping.yml
    This, along with *submit-job* is a playbook. Playbooks describe the tasks you want done
inventory > hosts
    A list of known systems, which Ansible may connect to.
group_vars > zxp.yml
    Information about how you want to interact with specific systems

4 CHECK ZXP.YML

Take a look at the *group_vars/zxp.yml* file – it has been customized already to meet the configuration of the system; specifically, the **zoau_home**, **python_path**, and **ansible_python_interpreter** values.

```
---
ansible_port: 22                                # SSH Port
ansible_user: youruser                          # USER used for deploy
ansible_password: yourpassword                  # USER used for deploy
# ansible_ssh_pipelining: True                  # needed for encoding on ssh connections
# System Related Variables
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"
_TAG_REDIR_ERR: txt
_TAG_REDIR_IN: txt
_TAG_REDIR_OUT: txt
tgt_tmp_dir: "/tmp"
ctl_tmp_dir: "/tmp"
```

All that is left is for you to change **ansible_user** and **ansible_password** to match your IBM Z system login.

Make those changes, save the file, and get ready to launch a command.

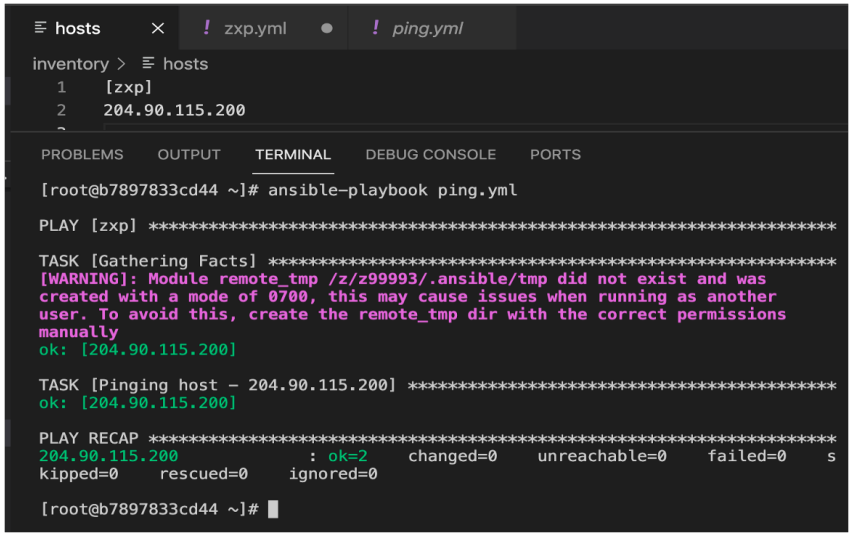
Note that this is just the configuration for a specific system, so you do not enter its IP address - that happens in the *inventory > hosts* file.

5 THE PING OF TRUTH

You have made it through a long sequence of setup steps; here is the moment you have been preparing for.

Open a terminal (from within this container environment) and enter the command:

```
ansible-playbook ping.yml
```



If all goes well, you will see a success message like the one above.

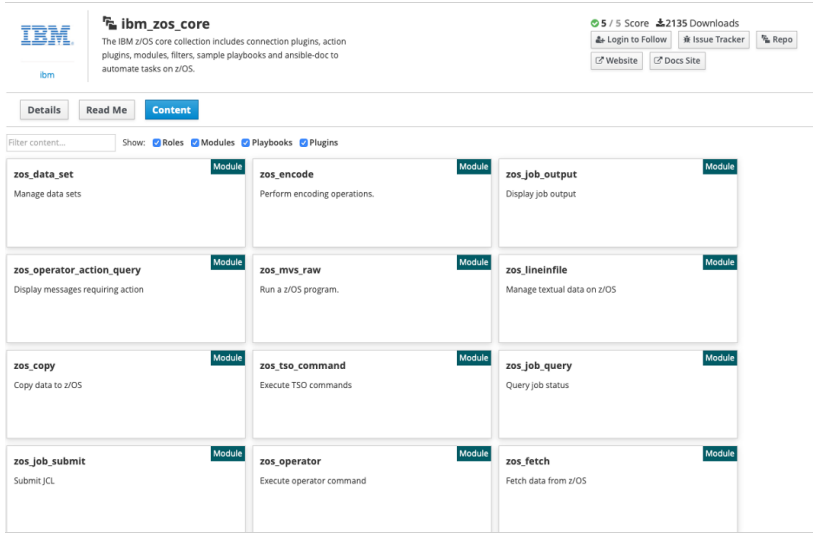
The pink output just means this is the first time Ansible has run, and it has now created temp directories.

It’s all good! Your first Ansible playbook run.

6 MARK IT COMPLETE

You can mark your challenge complete by submitting the **CHKAANS1** job in **ZXP.PUBLIC.JCL**

For more information, visit the *zos_operator* module page on the z/OS Core Collection reference to find the syntax for this module.



Refer to the **submit_job.yml** file as a reference, and feel free to borrow the “collections” and “environment” section to get your playbook working correctly, as that one is calling z/OS-specific tasks.

ping.yml is a generic playbook that works without those parts.

Nice job - let’s recap	Next up …
We’re so happy we can barely contain our excitement. We took the scenic route in getting here, and we hope you saw enough along the way to inspire you to keep going. Now we’ve got Ansible set up on your computer (or wherever you installed it), and have verified that we can call commands from a playbook. You also learned that there is a difference between the core commands built into Ansible (like Ping) and the ones you get once you start referencing the z/OS-specific ones. There is much more to see from here, so let’s keep moving.	You have set it all up, now get ready to use it. Proceed to **ANSB2** and you can stretch your skills a little more.