



# Ansible Navigator

By: Er. Vikas Nehra (M. Tech, B. Tech), Experience: 15 + Years

---

## Session - 26 Agenda:

### 1. Ansible Navigator

---

#### Ansible Navigator:

ansible-navigator is a command-based tool for creating, reviewing, and troubleshooting Ansible content, including inventories, playbooks, and collections.

Ansible Navigator (ansible-navigator) is a seriously powerful text user interface (TUI) tool that offers new and even better ways to interact with your Ansible environment and content, a tool which both owes a debt to the lasting legacies of ansible-playbook and its related programs, while also delivering much-needed new features (and feature consolidations) that have been leaving the traditional Ansible CLI toolbox a bit lacking in recent times: especially as the Red Hat Ansible Automation Platform (AAP) codebase and ecosystem continue to evolve and improve rapidly for AAP customers.

Both Ansible Navigator and ansible-playbook are fundamentally CLI tools that you can use to interact with Ansible, but they actually differ from each other in their approach and features.

#### Comparison Between ansible-playbook & Ansible Navigator:

##### 1. Main usage comparison:

ansible-playbook tool is used primarily for executing Ansible playbooks.

Ansible Navigator is considered a more comprehensive tool designed to explore, manage, and execute Ansible content, including playbooks, roles, and collections, while also allowing you to use the exact same execution environment (EE) images you do in AAP.

##### 2. UI comparison:

ansible-playbook tool has a traditional CLI interface, where you run playbooks by passing flags and specifying playbook files.

Ansible Navigator features a (default) interactive mode, offering a text user interface (TUI) that allows you to navigate through your Ansible content and discover available actions. It also allows you to run your playbooks in non-interactive (aka stdout) mode.

##### 3. Execution Environment support comparison:

ansible-playbook tool has to run playbooks via some type of local Python environment (e.g., using your system Python or a Python virtualenv).

Ansible Navigator supports execution environments, which are containerized environments that isolate Ansible execution. This ensures a consistent environment for running playbooks and reduces the risk of conflicts between dependencies.

##### 4. Extensibility:

ansible-playbook tool focuses on executing playbooks, and you sometimes end up having to use other ansible-\* tools to perform other pretty common related tasks.

Ansible Navigator provides additional features, such as browsing documentation, visualizing inventory, and managing collections, all of which makes it a more versatile tool for working with Ansible content.

Ansible Navigator was really created to address some limitations of the traditional Ansible CLI tools, and it provides you with a more efficient way to work with all things Ansible:

#### Features of Ansible Navigator:

1. Improves the user experience by providing a more interactive look and feel.
2. Enhances the average discoverability of Ansible content and available actions.
3. Simplifies the usage of execution environments, allowing you to work with containerized Ansible instances. This point is highly relevant for AAP customers, as there is no equivalent functionality available to you with ansible-playbook.
4. To go a bit deeper on discoverability, this means being able to effortlessly explore and find Ansible



# Ansible Navigator

By: Er. Vikas Nehra (M. Tech, B. Tech), Experience: 15 + Years

content, actions, and information, all within the Ansible Navigator TUI.

## Some examples in practice:

1. *Exploring playbooks and roles:* You can browse through your playbooks and roles, making it easier to find specific automation tasks or examine the structure of your Ansible content.
2. *Browsing collections:* Ansible Navigator allows you to explore Ansible collections. You can navigate through the collections, view the included content, and see ways to use them in your own automations.
3. *Viewing documentation:* Ansible Navigator enables you to view documentation for modules, roles, and plugins, all within in the TUI.
4. *Visualizing inventory:* You can easily visualize your inventory in Ansible Navigator, which provides an organized view of hosts and groups. You'll find that this makes it easier to discover the structure of the inventory and understand how hosts and groups relate to each other.
5. *Listing available actions:* When using the (default) interactive mode, Ansible Navigator displays a list of available actions that you can take, such as running a playbook, examining a role, or exploring a collection. Inarguably, this feature enhances increases your awareness of "the art of the possible" for the particular Ansible content you are working with, by providing an at-your-fingertips overview of possible actions and guiding you in your interactions with custom code especially.

## Steps To Install, Configure & Use Ansible Navigator:

First of all, we have to install ansible-core, python-pip and podman packages in our ansible server in order to use ansible navigator.

**\$ sudo dnf install -y ansible-core python-pip podman**

Now, we can execute pip3 install command to install the ansible navigator.

**\$ pip3 install ansible-navigator --user**

Ansible navigator requires container environment in order to work. We can list the available container images by running podman images command.

**\$ podman images**

Now, execute ansible-navigator command to start the ansible navigator. First time it will download the ansible navigator container image and its execution environment.

**\$ ansible-navigator**

Again, execute the podman images command to see the ansible navigator container image in your machine.

**\$ podman images**

Additionally, you can also execute the ansible navigator images command to see the ansible navigator container image.

**\$ ansible-navigator images**

Your ansible navigator is now ready for the use.

## Working With Ansible Navigator:

To see the version of ansible navigator, execute the below command:

**\$ ansible-navigator --version**

To get the help on ansible navigator, execute the below command:

**\$ ansible-navigator --help**

Let's create an ansible playbook to perform the ping test with the managed nodes. This playbook will be used to learn the fundamentals of ansible navigator.

**\$ vim ping.yml**

# Ansible Navigator

By: Er. Vikas Nehra (M. Tech, B. Tech), Experience: 15 + Years

```
---
- name: Ping Test Playbook
  hosts: all
  tasks:
    - name: Testing ping
      ansible.builtin.ping:
        register: result
    - name: Print ping result
      ansible.builtin.debug:
        msg: "{{ result }}"
...
```

Check the playbook for the syntax errors.

**\$ ansible-playbook ping.yml --syntax-check**

Execute the playbook.

**\$ ansible-playbook ping.yml**

Create an ansible navigator inventory file.

**\$ vim inventory**

**[Ubuntu]**

**node2**

**[RHEL]**

**node1**

**node3**

**[dev]**

**node1**

**node2**

Create an ansible navigator configuration file.

**\$ vim ansible-navigator.yml**

```
---
ansible-navigator:
  ansible:
    inventory:
      entries:
        - inventory
  execution-environment:
    container-engine: podman
    enabled: false
    image: quay.io/ansible/ansible-navigator-demo-ee:latest
  logging:
    level: debug
  playbook-artifact:
    enable: true
    save-as: playbook-artifacts/{playbook_name}-artifact-{time_stamp}.json
# mode: stdout
...
```

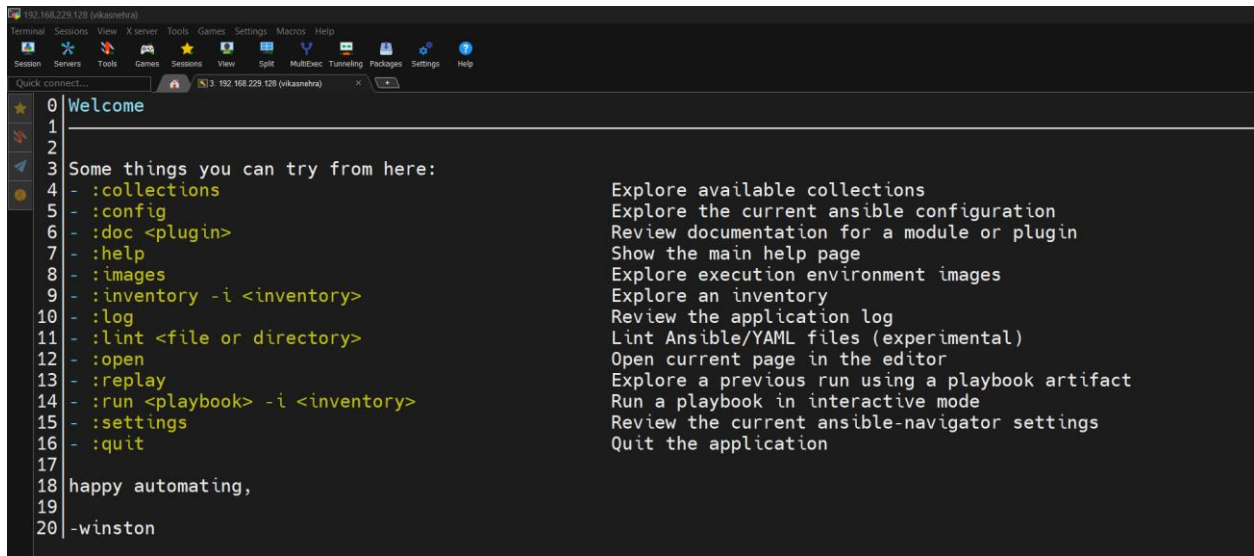
## Common Ansible-navigator commands:

*ansible-navigator*: Launches Ansible-navigator in interactive mode, providing a text user interface (TUI) to explore your Ansible content. (and *ansible-navigator run ping.yml* executes that playbook specifically.)

# Ansible Navigator

By: Er. Vikas Nehra (M. Tech, B. Tech), Experience: 15 + Years

**\$ ansible-navigator**



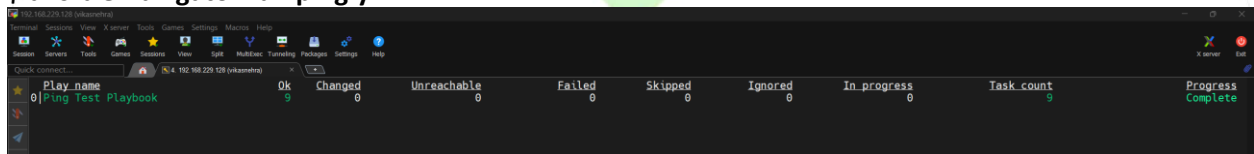
```

0 Welcome
1
2
3 Some things you can try from here:
4 -:collections      Explore available collections
5 -:config           Explore the current ansible configuration
6 -:doc <plugin>     Review documentation for a module or plugin
7 -:help            Show the main help page
8 -:images          Explore execution environment images
9 -:inventory -i <inventory> Explore an inventory
10 -:log             Review the application log
11 -:lint <file or directory> Lint Ansible/YAML files (experimental)
12 -:open           Open current page in the editor
13 -:replay         Explore a previous run using a playbook artifact
14 -:run <playbook> -i <inventory> Run a playbook in interactive mode
15 -:settings       Review the current ansible-navigator settings
16 -:quit          Quit the application
17
18 happy automating,
19
20 -winston
  
```

Here ansible navigator is in TUI mode and it waiting for the inputs from the user. To open any of these items just mention there :item\_name and press enter (e.g. to open collections write there :collections). In the list displayed on the next page just press the number against the item in the list to open it and see the details. If you want to go back and to exit the ansible-navigator TUI press the esc key on keyboard.

*ansible-navigator run ping.yml*: executes that playbook specifically.

**\$ ansible-navigator run ping.yml**



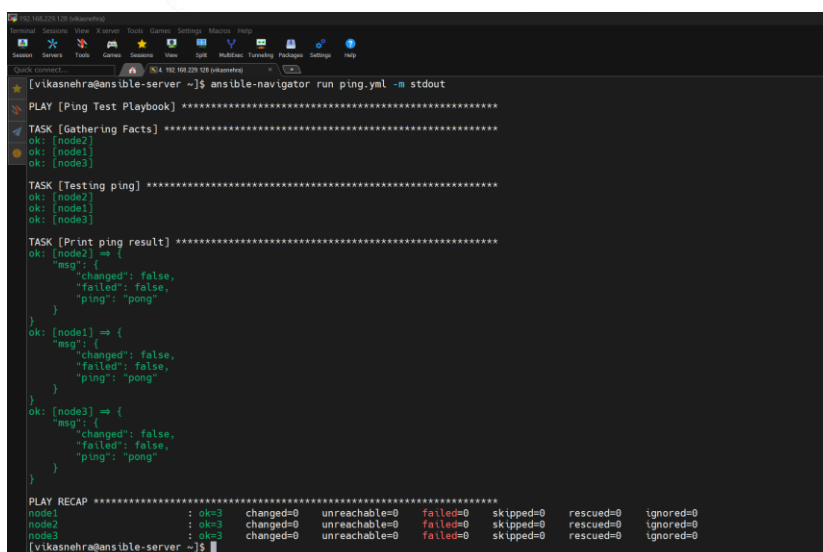
```

Play name      Ok  Changed  Unreachable  Failed  Skipped  Ignored  In progress  Task count  Progress
0 Ping Test Playbook  0    0         0             0        0         0          0          Complete
  
```

It will not show you the output by default, however you can press 0 there to see the execution process tasks statuses and the results. On the next page just press the number against the item in the list to open it and see the details. If you want to go back and to exit the ansible-navigator TUI press the esc key on keyboard.

*ansible-navigator run ping.yml -m stdout*: Executes the specified playbook in non-interactive (stdout) mode, displaying the output directly in the terminal more like ansible-playbook would.

**\$ ansible-navigator run ping.yml -m stdout**



```

[vikasnehra@ansible-server ~]$ ansible-navigator run ping.yml -m stdout

PLAY [Ping Test Playbook] *****

TASK [Gathering Facts] *****
ok: [node2]
ok: [node1]
ok: [node3]

TASK [Testing ping] *****
ok: [node2]
ok: [node1]
ok: [node3]

TASK [Print ping result] *****
ok: [node2] => {
  "msg": {
    "changed": false,
    "failed": false,
    "ping": "pong"
  }
}
ok: [node1] => {
  "msg": {
    "changed": false,
    "failed": false,
    "ping": "pong"
  }
}
ok: [node3] => {
  "msg": {
    "changed": false,
    "failed": false,
    "ping": "pong"
  }
}

PLAY RECAP *****
node1 : ok=3  changed=0  unreachable=0  failed=0  skipped=0  rescued=0  ignored=0
node2 : ok=3  changed=0  unreachable=0  failed=0  skipped=0  rescued=0  ignored=0
node3 : ok=3  changed=0  unreachable=0  failed=0  skipped=0  rescued=0  ignored=0
[vikasnehra@ansible-server ~]$
  
```



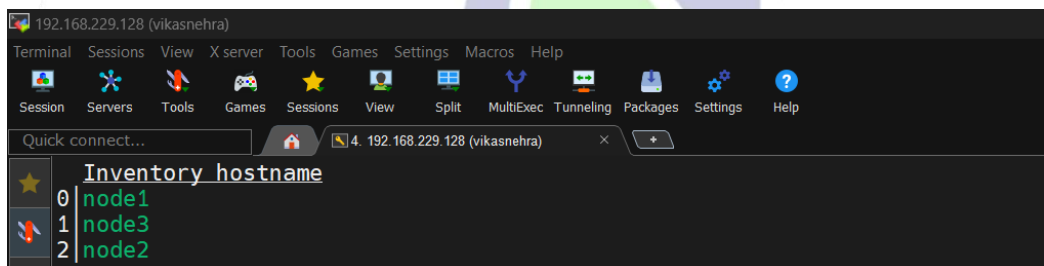
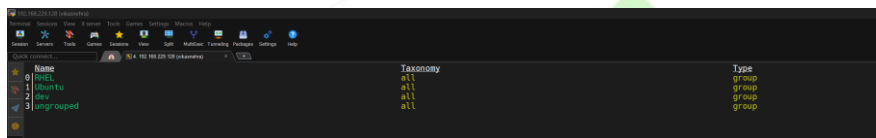
# Ansible Navigator

By: Er. Vikas Nehra (M. Tech, B. Tech), Experience: 15 + Years

It will show the execution like ansible-playbook command.

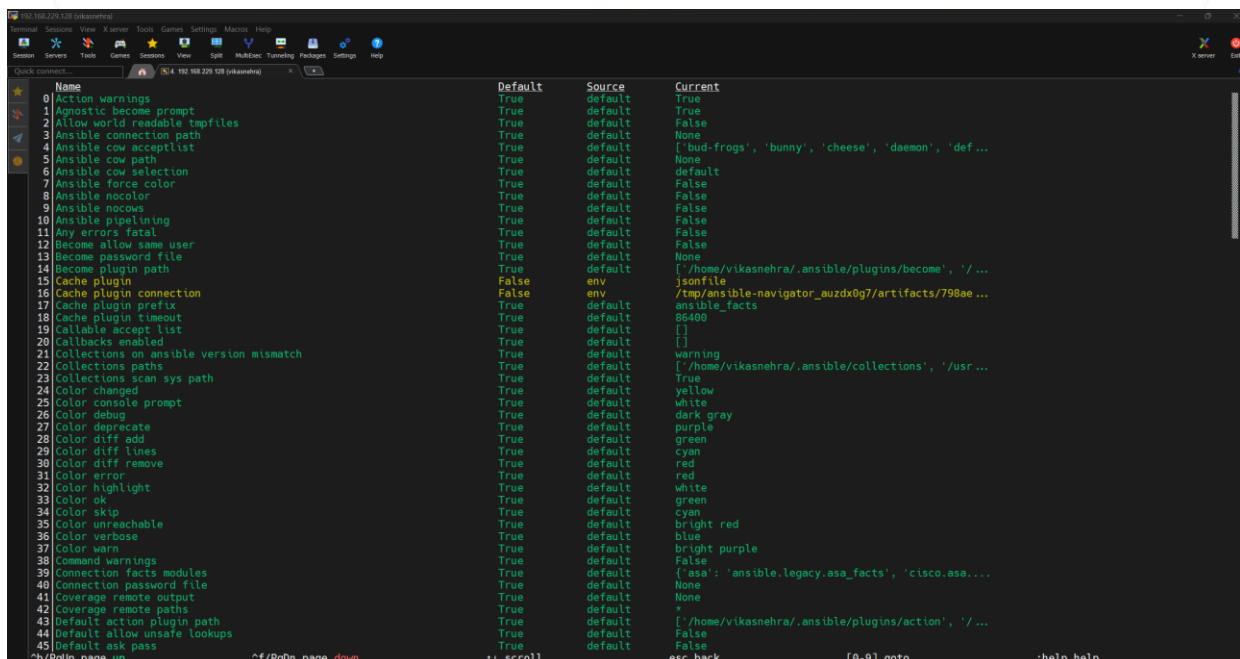
*ansible-navigator inventory -i inventory:* Visualizes the specified inventory file, showing an organized view of hosts and groups.

**\$ ansible-navigator inventory -i /etc/ansible/hosts**



*ansible-navigator config:* Displays the current Ansible configuration settings, including values from environment variables, configuration files, and defaults.

**\$ ansible-navigator config**



# Ansible Navigator

By: Er. Vikas Nehra (M. Tech, B. Tech), Experience: 15 + Years

**ansible-navigator doc ansible.builtin.ping:** Shows the documentation for the specified module, including a description, available parameters, and usage examples.

**\$ ansible-navigator doc ansible.builtin.ping**

```

Name: ansible.builtin.ping (module)
1 doc:
2 attributes:
3   check_mode:
4     description: Can run in check_mode and return changed status prediction without
5       modifying target
6     support: full
7   diff_mode:
8     description: Will return details on what has changed (or possibly needs changing
9       in check mode), when in diff mode
10    support: none
11  platform:
12    description: Target OS/families that can be operated against
13    platform: posix
14    support: N/A
15  author:
16    - Ansible Core Team
17    - Michael DeHaan
18  collection: ansible.builtin
19  description:
20    - A trivial test module, this module always returns C(pong) on successful contact.
21    - It does not make sense in playbooks, but it is useful from C(/usr/bin/ansible)
22    - to verify the ability to login and that a usable Python is configured.
23    - This is NOT ICMP ping, this is just a trivial test module that requires Python
24    - on the remote-node.
25    - For Windows targets, use the M(ansible.windows.win_ping) module instead.
26    - For Network targets, use the M(ansible.netcommon.net_ping) module instead.
27  filename: /usr/lib/python3.9/site-packages/ansible/modules/ping.py
28  has_action: false
29  module: ping
30  options:
31    data:
32      default: pong
33      description:
34        - Data to return for the C(ping) return value.
35        - If this parameter is set to C(crash), the module will cause an exception.
36      type: str
37  seealso:
38    - module: ansible.netcommon.net_ping
39    - module: ansible.windows.win_ping
40  short_description: Try to connect to host, verify a usable python and return C(pong)
41    on success
42  version_added: historical
43  version_added_collection: ansible.builtin
44  examples: |-
45    # Test we can login to 'webservers' and execute python with json lib.
46  /b/PopUp page up      /f/Pgdn page down      + scroll      esc back      :help help
  
```

**ansible-navigator images:** Lists the available container images for execution environments, along with their details and status.

**\$ ansible-navigator images**

Image	Tag	Execution environment	Created	Size
creator-ee	vs.18.0	True	4 weeks ago	747 MB

**cat ~/ansible-navigator.log:** Displays the Ansible Navigator log, which contains information about the tool's operation, including any errors or warnings.

**\$ cat ~/ansible-navigator.log**

```

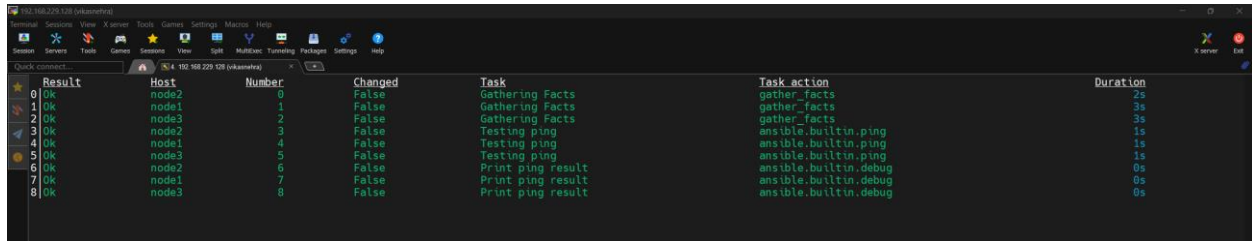
2023-06-19T09:27:33.007769+00:00 DEBUG 'ansible_navigator.actions.images_update_args' Running with exec_command as '/bin/bash' (str/Defaults)
2023-06-19T09:27:33.007816+00:00 DEBUG 'ansible_navigator.actions.images_update_args' Running with exec_shell as 'True' (bool/Defaults)
2023-06-19T09:27:33.007862+00:00 DEBUG 'ansible_navigator.actions.images_update_args' Running with execution_environment as 'False' (bool/Defaults)
2023-06-19T09:27:33.007909+00:00 DEBUG 'ansible_navigator.actions.images_update_args' Running with execution_environment_image as 'quay.io/ansible/ansible-navigator-demo-ee:latest' (str/Settings file)
2023-06-19T09:27:33.007960+00:00 DEBUG 'ansible_navigator.actions.images_update_args' Running with execution_environment_volume_mounts as 'Not set' (Constants/Not set)
2023-06-19T09:27:33.008017+00:00 DEBUG 'ansible_navigator.actions.images_update_args' Running with format as 'yaml' (str/Defaults)
2023-06-19T09:27:33.008065+00:00 DEBUG 'ansible_navigator.actions.images_update_args' Running with help_builder as 'False' (bool/Defaults)
2023-06-19T09:27:33.008117+00:00 DEBUG 'ansible_navigator.actions.images_update_args' Running with help_config as 'False' (bool/Defaults)
2023-06-19T09:27:33.008164+00:00 DEBUG 'ansible_navigator.actions.images_update_args' Running with help_doc as 'False' (bool/Defaults)
2023-06-19T09:27:33.008211+00:00 DEBUG 'ansible_navigator.actions.images_update_args' Running with help_inventory as 'False' (bool/Defaults)
2023-06-19T09:27:33.008257+00:00 DEBUG 'ansible_navigator.actions.images_update_args' Running with help_playbook as 'False' (bool/Defaults)
2023-06-19T09:27:33.008308+00:00 DEBUG 'ansible_navigator.actions.images_update_args' Running with images_details as '['everything']' (list/Defaults)
2023-06-19T09:27:33.008356+00:00 DEBUG 'ansible_navigator.actions.images_update_args' Running with inventory as '['/etc/ansible/hosts']' (list/Settings file)
2023-06-19T09:27:33.008402+00:00 DEBUG 'ansible_navigator.actions.images_update_args' Running with inventory_column as 'Not set' (Constants/Not set)
2023-06-19T09:27:33.008449+00:00 DEBUG 'ansible_navigator.actions.images_update_args' Running with lint_config as 'Not set' (Constants/Not set)
2023-06-19T09:27:33.008497+00:00 DEBUG 'ansible_navigator.actions.images_update_args' Running with lintables as 'Not set' (Constants/Not set)
2023-06-19T09:27:33.008622+00:00 DEBUG 'ansible_navigator.actions.images_update_args' Running with log_append as 'True' (bool/Defaults)
2023-06-19T09:27:33.008732+00:00 DEBUG 'ansible_navigator.actions.images_update_args' Running with log_file as '/home/vikasnehra/ansible-navigator.log' (str/Defaults)
2023-06-19T09:27:33.008779+00:00 DEBUG 'ansible_navigator.actions.images_update_args' Running with log_level as 'debug' (str/Settings file)
2023-06-19T09:27:33.008825+00:00 DEBUG 'ansible_navigator.actions.images_update_args' Running with mode as 'interactive' (str/Defaults)
2023-06-19T09:27:33.008872+00:00 DEBUG 'ansible_navigator.actions.images_update_args' Running with osc4 as 'True' (bool/Defaults)
2023-06-19T09:27:33.008918+00:00 DEBUG 'ansible_navigator.actions.images_update_args' Running with pass_environment_variable as 'Not set' (Constants/Not set)
2023-06-19T09:27:33.008965+00:00 DEBUG 'ansible_navigator.actions.images_update_args' Running with playbook as 'Not set' (Constants/Not set)
2023-06-19T09:27:33.009012+00:00 DEBUG 'ansible_navigator.actions.images_update_args' Running with playbook_artifact_enable as 'True' (bool/Settings file)
2023-06-19T09:27:33.009059+00:00 DEBUG 'ansible_navigator.actions.images_update_args' Running with playbook_artifact_replay as 'Not set' (Constants/Not set)
2023-06-19T09:27:33.009117+00:00 DEBUG 'ansible_navigator.actions.images_update_args' Running with playbook_artifact_save_as as 'playbook-artifacts/(playbook_name)-artifact-(time_stamp).json' (str/Settings file)
2023-06-19T09:27:33.009164+00:00 DEBUG 'ansible_navigator.actions.images_update_args' Running with plugin_name as 'Not set' (Constants/Not set)
2023-06-19T09:27:33.009211+00:00 DEBUG 'ansible_navigator.actions.images_update_args' Running with plugin_type as 'module' (str/Defaults)
2023-06-19T09:27:33.009258+00:00 DEBUG 'ansible_navigator.actions.images_update_args' Running with pull_arguments as 'Not set' (Constants/Not set)
2023-06-19T09:27:33.009304+00:00 DEBUG 'ansible_navigator.actions.images_update_args' Running with pull_policy as 'tag' (str/Defaults)
2023-06-19T09:27:33.009351+00:00 DEBUG 'ansible_navigator.actions.images_update_args' Running with settings_effective as 'False' (bool/Defaults)
2023-06-19T09:27:33.009444+00:00 DEBUG 'ansible_navigator.actions.images_update_args' Running with settings_sample as 'False' (bool/Defaults)
2023-06-19T09:27:33.009490+00:00 DEBUG 'ansible_navigator.actions.images_update_args' Running with settings_schema as 'json' (str/Defaults)
2023-06-19T09:27:33.009536+00:00 DEBUG 'ansible_navigator.actions.images_update_args' Running with settings_sources as 'False' (bool/Defaults)
2023-06-19T09:27:33.009583+00:00 DEBUG 'ansible_navigator.actions.images_update_args' Running with time_zone as 'UTC' (str/Defaults)
2023-06-19T09:27:33.009629+00:00 DEBUG 'ansible_navigator.actions.images_update_args' Running with workdir as '/home/vikasnehra' (str/Defaults)
2023-06-19T09:27:33.009676+00:00 DEBUG 'ansible_navigator.ui.framework.curses.window._init' self._ui_config: UIConfig(color=True, colors_initialized=True, grammar_dir=PosixPath('/home/vikasnehra/.local/lib/python3.9/site-packages/ansible_navigator/data/themes/dark_vs.json'), osc4=True, terminal_colors_path=PosixPath('/home/vikasnehra/.local/lib/python3.9/site-packages/ansible_navigator/data/themes/terminal_colors.json'), theme_path=PosixPath('/home/vikasnehra/.local/lib/python3.9/site-packages/ansible_navigator/data/themes/dark_vs.json'))
2023-06-19T09:27:33.020090+00:00 DEBUG 'ansible_navigator.ui.framework.curses.window._init' self._ui_config: UIConfig(color=True, colors_initialized=True, grammar_dir=PosixPath('/home/vikasnehra/.local/lib/python3.9/site-packages/ansible_navigator/data/themes/dark_vs.json'), osc4=True, terminal_colors_path=PosixPath('/home/vikasnehra/.local/lib/python3.9/site-packages/ansible_navigator/data/themes/terminal_colors.json'), theme_path=PosixPath('/home/vikasnehra/.local/lib/python3.9/site-packages/ansible_navigator/data/themes/dark_vs.json'))
2023-06-19T09:27:53.041897+00:00 DEBUG 'ansible_navigator.actions.back.run' Return to images, at last step
[vikasnehra@ansible-server ~]$
  
```

# Ansible Navigator

By: Er. Vikas Nehra (M. Tech, B. Tech), Experience: 15 + Years

`ansible-navigator replay playbook-artifacts/ping-artifact-2023-06-19T09\ :19\ :59.307330+00\ :00.json:`  
“Replays” a previous ansible-navigator “run”.

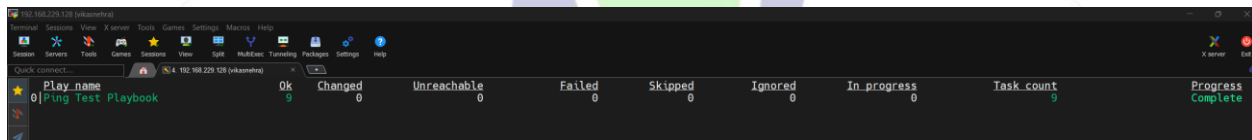
`$ ansible-navigator replay playbook-artifacts/ping-artifact-2023-06-19T09\ :19\ :59.307330+00\ :00.json`



Result	Host	Number	Changed	Task	Task action	Duration
0 OK	node2	0	False	Gathering Facts	gather_facts	2s
1 OK	node1	1	False	Gathering Facts	gather_facts	3s
2 OK	node3	2	False	Gathering Facts	gather_facts	3s
3 OK	node2	3	False	Testing ping	ansible.builtin.ping	1s
4 OK	node1	4	False	Testing ping	ansible.builtin.ping	1s
5 OK	node3	5	False	Testing ping	ansible.builtin.ping	1s
6 OK	node2	6	False	Print ping result	ansible.builtin.debug	0s
7 OK	node1	7	False	Print ping result	ansible.builtin.debug	0s
8 OK	node3	8	False	Print ping result	ansible.builtin.debug	0s

`ansible-navigator run ping.yml -i /etc/ansible/hosts --eei ansible-navigator-demo-ee:` Specifies both an inventory and execution environment in a “run”.

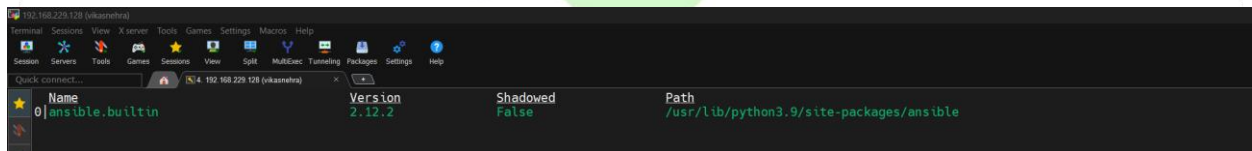
`$ ansible-navigator run ping.yml -i /etc/ansible/hosts --eei ansible-navigator-demo-ee`



Play name	Ok	Changed	Unreachable	Failed	Skipped	Ignored	In progress	Task count	Progress
0 Ping Test Playbook	9	0	0	0	0	0	0	9	Complete

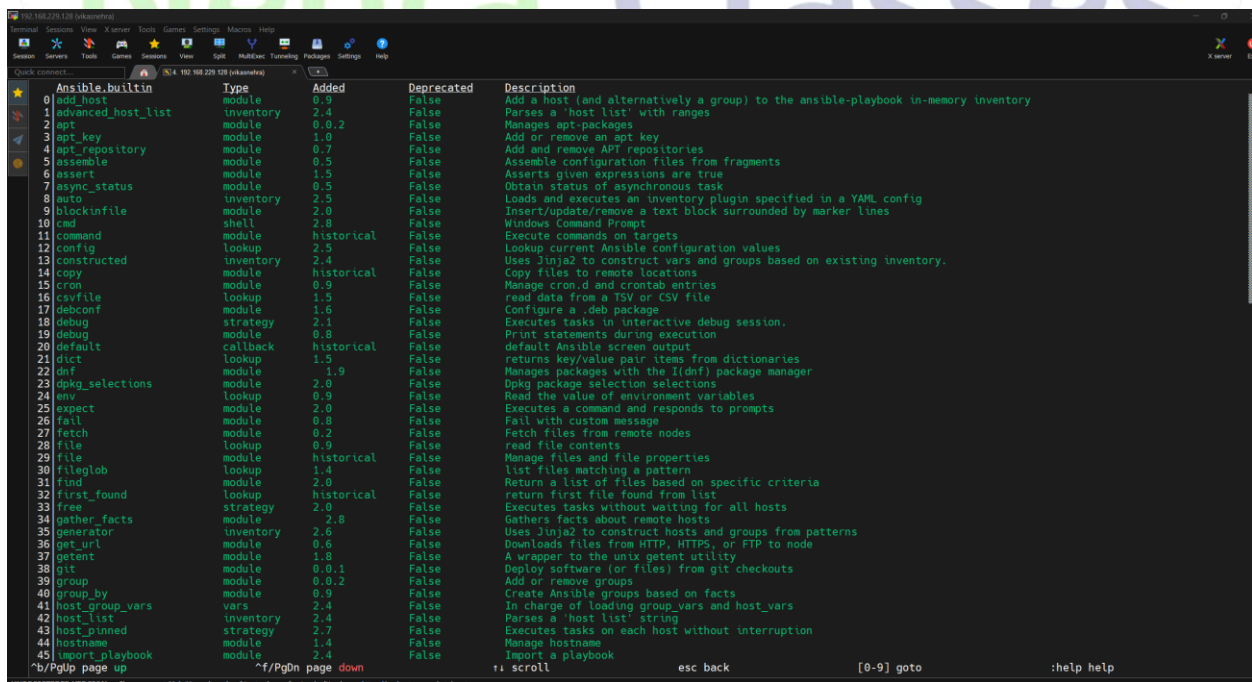
`ansible-navigator collections:` Lists all installed Ansible collections, including their name, version, and installation path.

`$ ansible-navigator collections`



Name	Version	Shadowed	Path
0 ansible.builtin	2.12.2	False	/usr/lib/python3.9/site-packages/ansible

Press 0 to see the all builtin collections list.



Ansible.builtin	Type	Added	Deprecated	Description
0 add_host	module	0.9	False	Add a host (and alternatively a group) to the ansible-playbook in-memory inventory
1 advanced_host_list	inventory	2.4	False	Parses a 'host list' with ranges
2 apt	module	0.0.2	False	Manages apt-packages
3 apt_key	module	1.0	False	Add or remove an apt key
4 apt_repository	module	0.7	False	Add and remove APT repositories
5 assemble	module	0.5	False	Assemble configuration files from fragments
6 assert	module	1.5	False	Asserts given expressions are true
7 async_status	module	0.5	False	Obtain status of asynchronous task
8 auto	inventory	2.5	False	Loads and executes an inventory plugin specified in a YAML config
9 blockinfile	module	2.0	False	Insert/update/remove a text block surrounded by marker lines
10 cmd	shell	2.8	False	Windows Command Prompt
11 command	module	historical	False	Execute commands on targets
12 config	lookup	2.5	False	Lookup current Ansible configuration values
13 constructed	inventory	2.4	False	Uses Jinja2 to construct vars and groups based on existing inventory.
14 copy	module	historical	False	Copy files to remote locations
15 cron	module	0.9	False	Manage cron.d and crontab entries
16 csvfile	lookup	1.5	False	read data from a TSV or CSV file
17 debconf	module	1.6	False	Configure a .deb package
18 debug	strategy	2.1	False	Executes tasks in interactive debug session.
19 debug	module	0.8	False	Print statements during execution
20 default	callback	historical	False	default Ansible screen output
21 dict	lookup	1.5	False	returns key/value pair items from dictionaries
22 dnf	module	1.9	False	Manages packages with the I(dnf) package manager
23 dpkg_selections	module	2.0	False	Dpkg package selection selections
24 env	lookup	0.9	False	Read the value of environment variables
25 expect	module	2.0	False	Executes a command and responds to prompts
26 fail	module	0.8	False	Fail with custom message
27 fetch	module	0.2	False	Fetch files from remote nodes
28 file	module	0.9	False	read file contents
29 file	module	historical	False	Manage files and file properties
30 fileglob	lookup	1.4	False	list files matching a pattern
31 find	module	2.0	False	Return a list of files based on specific criteria
32 first_found	lookup	historical	False	return first file found from list
33 free	strategy	2.0	False	Executes tasks without waiting for all hosts
34 gather_facts	module	2.8	False	Gathers facts about remote hosts
35 generator	inventory	2.6	False	Uses Jinja2 to construct hosts and groups from patterns
36 get_url	module	0.6	False	Downloads files from HTTP, HTTPS, or FTP to node
37 getent	module	1.8	False	A wrapper to the unix getent utility
38 git	module	0.0.1	False	Deploy software (or files) from git checkouts
39 group	module	0.0.2	False	Add or remove groups
40 group_by	module	0.9	False	Create Ansible groups based on facts
41 host_group_vars	inventory	2.4	False	In charge of loading group_vars and host_vars
42 host_list	inventory	2.4	False	Parses a 'host list' string
43 host_pinned	strategy	2.7	False	Executes tasks on each host without interruption
44 hostname	module	1.4	False	Manage hostname
45 import_playbook	module	2.4	False	import a playbook

To open any item from this list just press the same number key on your keyboard.

Thank You