



MORPHEUS

Using Morpheus to Make Ansible Better

Integration Overview with
Ansible and Ansible Tower

Last Updated: March 2020

Copyright © 2020 Morpheus Data, LLC. All Rights Reserved
All third-party product and company names are property of their
respective holders and use does not imply any specific endorsement

INTRODUCTION

Both Morpheus and Ansible provide a lot of benefits to increase efficiency and agility via automation across hybrid or multi-cloud environments. When it comes to automation there are usually many ways to achieve the same end-state and the same is true here. In this document, we will highlight some of the gaps in Ansible deployments and cover significant benefits to be gained by leveraging the Morpheus - Ansible integration to orchestrate playbook runs and enable self-service provisioning. This paper will also cover how Morpheus can help lower TCO by eliminating the need for more expensive add-ons such as Ansible Tower. Before we jump in let's spend a minute on Configuration Management benefits and challenges.

A QUICK PRIMER ON CONFIGURATION MANAGEMENT (CM)

Configuration management is a process for ensuring consistency among systems by documenting, tracking interdependencies, and remediating when configuration drift is detected. CM is all about state management. CM practices include keeping track of artifacts, revision control and the establishment of baselines. If something goes wrong, you can determine what was changed and by who.

For organizations embracing DevOps and Continuous Integration / Continuous Delivery (CI/CD) this is critical because they can track configuration state changes in development and then enforce an identical state of dependencies through test and production environments. When coupled with self-service provisioning, organizations are able to quickly tear down and refresh the entire release pipeline at any time because everything is stored and managed as code within the CM tool.

Popular CM tools include Ansible, Chef, Salt, and Puppet. While there are some differences in how these tools are architected, they all generally fall into the same category. The tools also differ in approach in their domain-specific languages (DSL) and approach to automation scripting. In this document we will be focusing on Ansible with Morpheus vs Ansible Tower. Ansible has gained a lot of traction in the industry in the recent years because of its simplicity in getting started. The gaps described below apply to Ansible and the other CM tools alike.

GAPS IN CONFIGURATION MANAGEMENT TOOLS

GRAPHICAL USER INTERFACE (GUI or lack thereof) - Many of these tools provide a GUI as a paid 'add-on' (i.e. Ansible Tower). However, most of their GUIs provide only basic reporting and administration while the core configuration state of a machine still has to be managed via a code-base and a domain-specific language (DSL). This forces Operations teams to write configurations like a developer, by pushing code to Source Code Management (SCM) tools like Git, SVN, etc... Most organizations need both GUI and Command Line Interface / API access to meet their administrative and business needs.

ROLE-BASED ACCESS CONTROL (RBAC) - Control and identity management is vital to properly scope and govern the activities of self-service users, especially in large Enterprises or Service Provider deployments. While some RBAC is available in paid versions of the popular CM tools, they typically do not provide anywhere near the granularity and multi-tenancy capabilities that Morpheus provides. This is because CM tools added RBAC as an afterthought whereas it's a core design within the Morpheus unified orchestration platform. Additionally, CM-oriented RBAC only touches a piece of the governance surface area. RBAC done right should go beyond CM to include infrastructure and cloud access, policy management, approvals, etc.

THIRD-PARTY INTEGRATIONS - Most CM tools have manifests, recipes, playbooks or other libraries available to manipulate infrastructure as part of establishing a baseline state. The challenge is that it's rare to find an off-the-shelf template that allows IT to start automating their entire infrastructure out-of-the-box. The result is it can take a very long time for organizations to see the results of CM deployments.

FRAGILITY VS AGILITY - Organizations can spend months trying to develop the operational expertise to knit together CM scripts for a specific workflow. After all that work, they are left with a brittle and fragile structure that cannot quickly adapt to changes in personnel, platform strategy, or cloud consumption choices. Additionally, your point of automation can become your point of lock-in, but even fully implemented CM tools by themselves only address a fraction of cloud management and application lifecycle challenges.

HOW MORPHEUS AND ANSIBLE ARE BETTER TOGETHER

The most secure way to run and manage Ansible.

Morpheus Agent Command Bus to shut down ingress

Ansible requires access to port 22 (SSH on Linux), and port 5985 (WINRM for Windows) to function. In many environments these ports are blocked due to security policies, essentially rendering Ansible unusable.

Morpheus is uniquely able to get around this problem and make Ansible usable again in highly secure environments. How? The Morpheus Command Bus feature leverages the Morpheus agent which uses port 443 so Ansible playbooks can still be executed. You can find more details on this at the following location:

https://docs.morpheusdata.com/en/latest/integration_guides/Automation/ansible.html - using-ansible-over-the-morpheus-agent-command-bus

NEW ANSIBLE INTEGRATION
✕

NAME

ENABLED

ANSIBLE GIT URL

KEY PAIR

PLAYBOOKS PATH

ROLES PATH

GROUP VARIABLES PATH

HOST VARIABLES PATH

USE ANSIBLE GALAXY

ENABLE VERBOSE LOGGING

USE MORPHEUS AGENT COMMAND BUS

[SAVE CHANGES](#)

Morpheus Cypher Service for secure secret management

Morpheus has a built-in service called Cypher. It's a secure Key/Value store. Cypher allows the storage of secret data (like credentials or variable sets) in a highly encrypted way for future retrieval. Ansible Tower has a similar feature (secrets), but open source Ansible doesn't. When leveraging Morpheus and Ansible to execute playbooks (at any stage: provisioning, day 2, or ad hoc), these variables can be seamlessly passed to the playbook(s), so variables only have to be maintained and managed in one place. Playbooks run faster in Morpheus because of the quick access to the Morpheus collected facts of the instances/hosts which can be used as Morpheus variables in playbooks.

Morpheus Governance and Role-based Access Control (RBAC)

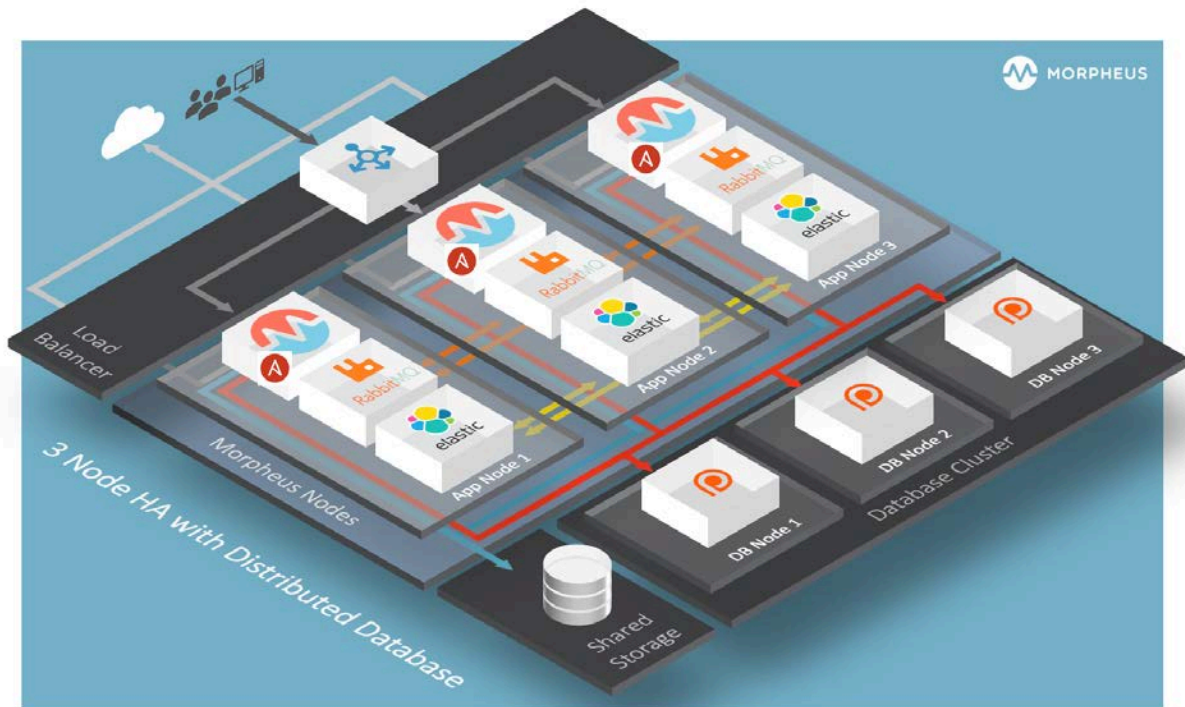
As noted earlier, CM tools have limitations when it comes to governance and RBAC. If you have access to an Ansible server, you can run any playbook. By integrating Ansible with Morpheus you can take advantage of the fine-grained governance and RBAC available in Morpheus to extend those capabilities when providing access to Ansible playbooks. These can be consumed during the

provisioning and post provisioning phases, or during ongoing day-2 operations, based on the RBAC entitlements granted to the consumer. Another point to discuss here is that Morpheus Governance and Control goes well beyond Ansible Playbooks. Because it's a Multi-Cloud Provisioning Orchestration platform Morpheus takes that governance to Private and Public Clouds, as well as any additional systems integrated with it.

Improve Ansible Resilience and Availability.

Morpheus High Availability (HA) and what it means to Ansible

When installing Morpheus there are a number of installation types. One of them is to install Morpheus in an HA configuration (3 nodes). Because the Ansible binaries get installed on the Morpheus HA appliances they inherently take advantage of this capability making Ansible HA as well. Think of it this way: Morpheus HA automatically translates to Ansible HA (which is something that Ansible does not have). Below is a diagram of the 3 Node HA implementation of Morpheus. Ansible would be installed on the 3 application nodes: App Node 1, App Node 2, and App Node 3.



Track history of all your Ansible runs.

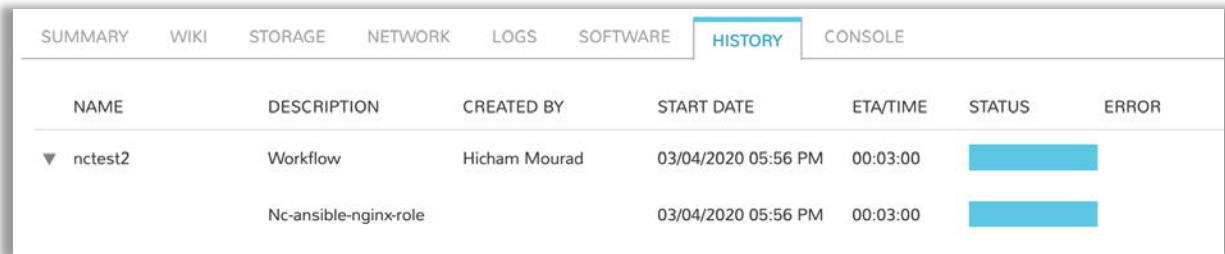
Morpheus Live Output

From Morpheus, whether you run an Ansible playbook during provisioning, post provisioning or even at any time during the lifecycle of the instance, the output of the playbook run is available in the history of that resource. Easily accessible to

the resource owner (and whoever else that may have access). This data can also be exported if desired by the user.

Morpheus Audit History

Morpheus tracks and audits all activities on the Morpheus managed instances. This includes any Ansible playbooks that are run on any individual instances or application instances. This is all made available in the HISTORY tab of the resource. For auditing and logging Morpheus uses the Common Event Format (CEF) which is an open logging standard.



NAME	DESCRIPTION	CREATED BY	START DATE	ETA/TIME	STATUS	ERROR
▼ nctest2	Workflow	Hicham Mourad	03/04/2020 05:56 PM	00:03:00		
	Nc-ansible-nginx-role		03/04/2020 05:56 PM	00:03:00		

Then drilling into the details, we will see more detailed execution information



HISTORY DETAIL [Close]

nctest2

Description: nc-ansible-install-httpd
Start Date: 03/04/2020 11:38 AM
Created By: Nick Celebic
Duration: 00:00:16
Status: complete

PROCESS OUTPUT

```
PLAY [This sets up an httpd webservice] *****
TASK [Gathering Facts] *****
ok: [nctest2.localdomain]

TASK [Install apache packages] *****
changed: [nctest2.localdomain]

TASK [ensure httpd is running] *****
changed: [nctest2.localdomain]

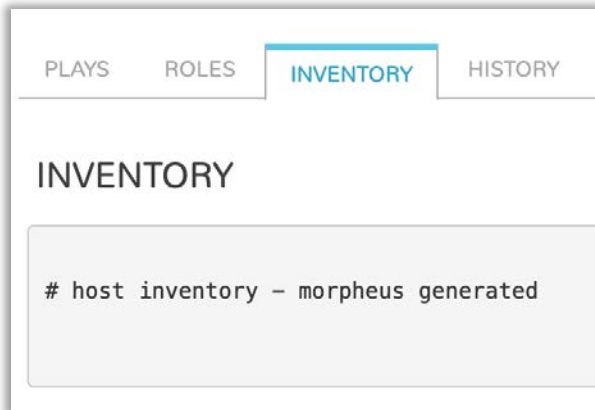
PLAY RECAP *****
nctest2.localdomain : ok=3  changed=2  unreachable=0  failed=0  skipped=0  rescued=0  ignored=0
```

Simplify Ansible Management and Lower Costs

Morpheus' Dynamic Inventory Management

Ansible leverages an inventory file to track the hosts that playbooks can be executed against. This file has to be manually maintained in Ansible and Ansible Tower. A huge benefit that Morpheus provides is that it will create and manage the inventory file automatically. Morpheus will also remove an entry when a

machine is removed. You will never have to touch this file again; it enables complete hands-off inventory management. Once Ansible integration is enabled on a Morpheus infrastructure group, then all provisioned instances (in any cloud) will automatically get added to the inventory file by Morpheus. (to get this same functionality with Red Hat one would need to purchase, install, and configure both Ansible Tower and CloudForms)



Leverage Open Source Ansible instead of paying for Ansible Tower

Open source Ansible is obviously free. Ansible Tower is not. By integrating open source Ansible with Morpheus organizations can take advantage of the Morpheus UI, rich set of capabilities, and broad cloud support to reduce and save costs while achieving quicker time to value. Everyone wants to do more with less! To achieve similar capabilities to Morpheus and Ansible, one would need to consider Ansible Tower and CloudForms which increase the costs.

That said, if you're already invested in Ansible Tower and are happy with it we fully support integration into Tower as well.

The rest of this document will go into the technical specifics of how to integrate and leverage Ansible from within Morpheus. You can also access the latest Ansible integration documentation at:

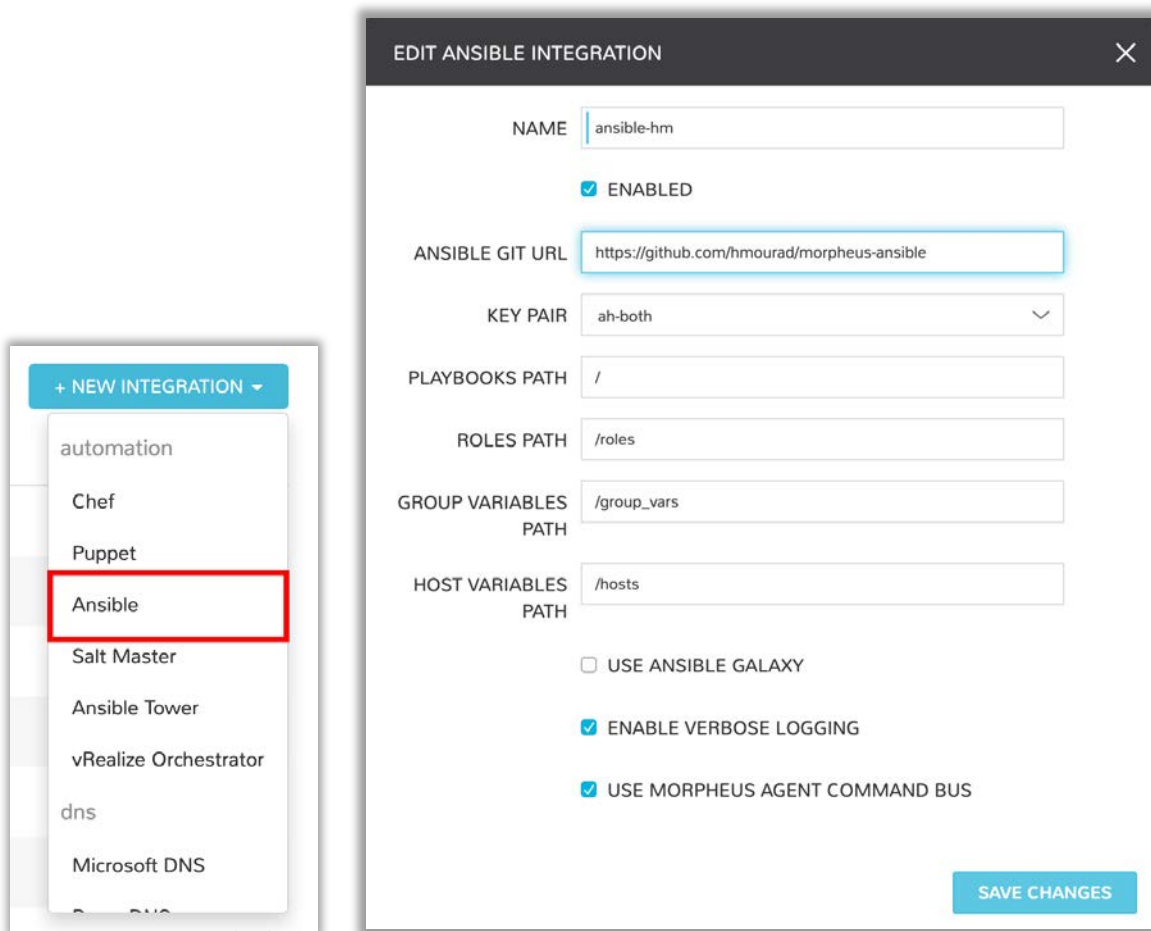
https://docs.morpheusdata.com/en/4.2.0/integration_guides/Automation/ansible.html

HOW TO CONFIGURE AND USE ANSIBLE FROM MORPHEUS

Let's show how to go ahead and implement the integration and walk through an example of how to run an Ansible playbook on a newly provisioned application.

The first step is to install Ansible on the Morpheus appliance. It should automatically be installed, but if not, you can find additional details here: https://docs.morpheusdata.com/en/latest/troubleshooting/Ansible_Troubleshooting.html

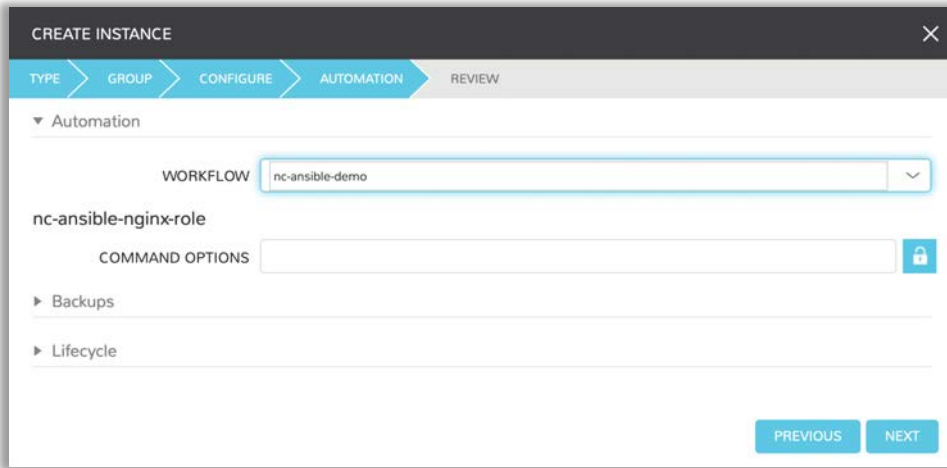
The second step is to add the Ansible integration in the Morpheus UI and point to the Git repository where the Ansible playbooks reside.



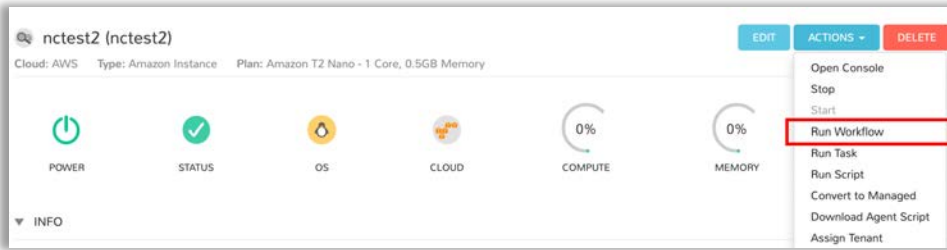
NOTE: Many Larger environments will block SSH and WINRM ports for security purposes. Ansible cannot function as it relies on those protocol ports. Morpheus can still execute the Ansible playbooks.

Now that the setup is complete, we are able to run Ansible playbooks from multiple locations in Morpheus. For example:

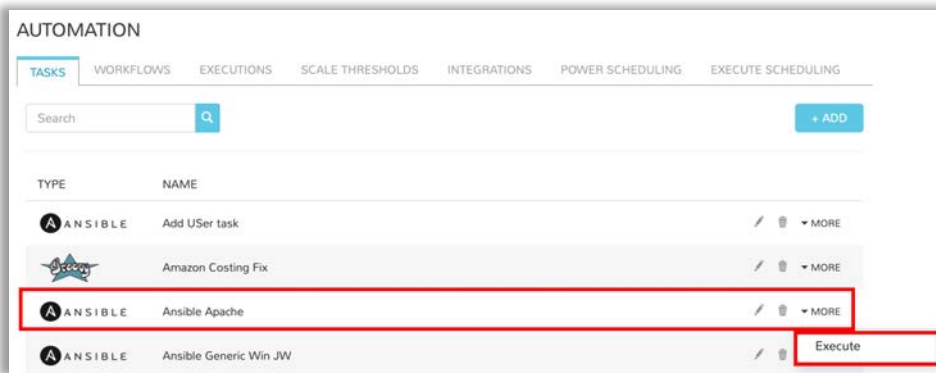
1. During the instance or application provisioning phases.



2. As a day 2 operation run a workflow that will kick off an Ansible playbook.



3. Execute an Ansible Task in Morpheus.



Regardless of how you chose to run the Ansible Playbook(s), you can track the results under the HISTORY tab on the target resource.

NAME	DESCRIPTION	CREATED BY	START DATE	ETA/TIME	STATUS	ERROR
▼ nctest2	Workflow	Hicham Mourad	03/04/2020 05:56 PM	00:03:00	Running	
	Nc-ansible-nginx-role		03/04/2020 05:56 PM	00:03:00	Running	

SUMMARY

We've learned all about the benefits of using Morpheus and Ansible together, and how Morpheus makes Ansible better, and together they provide great value to any organization. We've also highlighted how easy it is to setup this integration to take advantage of these benefits. The table below further highlights the capabilities they bring together. One final point I'd like to make is that any organization looking to procure Ansible Tower can save costs by leveraging Morpheus and open source Ansible. Be the hero on all fronts, use Morpheus and take advantage of its rich set of integrations.

Feature	Ansible	Ansible Tower	Morpheus & Ansible
Graphical User Interface	✗	✓	✓
Role-Based Access Control	✓	✓	✓
Governance Beyond Configuration Mgmt.	✗	✗	✓
High Availability	✗	✓	✓
Security: Get around blocked ports	✗	✗	✓
Audit Trail	✓	✓	✓
Logging and Analytics integration	✓	✓	✓
Approvals	✗	✗	✓

Full = ✓

Partial = ✓

None = ✗

To get started with Morpheus today, please visit www.morpheusdata.com
Request a demo of Morpheus at www.morpheusdata.com/demo/