

UNCLASSIFIED



RED HAT ANSIBLE AUTOMATION CONTROLLER SECURITY TECHNICAL IMPLEMENTATION GUIDE (STIG) OVERVIEW

05 January 2026

Developed by Red Hat and DISA for the DOD

UNCLASSIFIED

Trademark Information

Names, products, and services referenced within this document may be the trade names, trademarks, or service marks of their respective owners. References to commercial vendors and their products or services are provided strictly as a convenience to our users, and do not constitute or imply endorsement by the Defense Information Systems Agency (DISA) of any nonfederal entity, event, product, service, or enterprise.

TABLE OF CONTENTS

	Page
1. INTRODUCTION.....	1
1.1 Executive Summary.....	1
1.2 Authority.....	1
1.3 Vulnerability Severity Category Code Definitions.....	1
1.4 STIG Distribution.....	2
1.5 SRG Compliance Reporting.....	2
1.6 Document Revisions.....	2
1.7 Other Considerations.....	2
1.8 Product Approval Disclaimer.....	3
2. ASSESSMENT CONSIDERATIONS.....	4
2.1 Security Assessment Information – Controller Guidance Only.....	4
2.2 Security Assessment Information – External Identity and Access Management.....	4
2.3 Security Assessment Information – External Central Logging.....	5
2.4 Security Assessment Information – Host System.....	5
2.5 Security Assessment Information – Other Systems.....	6
3. CONCEPTS AND TERMINOLOGY CONVENTIONS.....	7
3.1 Ansible Automation Platform overview.....	7
3.2 Ansible Automation Platform Components.....	7
3.3 Playbooks and Execution Environments.....	8
3.4 Module Execution & Ansible Mesh.....	9
3.5 Credentialing and Inventorying.....	10
4. GENERAL SECURITY REQUIREMENTS.....	11
4.1 Hardening of Integrated External Services.....	11
4.2 Disaster Recovery and Continuity of Operations.....	11

LIST OF TABLES

	Page
Table 1-1: Vulnerability Severity Category Code Definitions	2

LIST OF FIGURES

	Page
Figure 3-1: Ansible Automation Platform 2.2+ General Architecture.....	7
Figure 3-2: Network Automation Compared to Servers	9
Figure 3-3: Automation Mesh for Distributed Execution Environments	9

1. INTRODUCTION

1.1 Executive Summary

The Red Hat Ansible Automation Controller Security Technical Implementation Guide (STIG) is published as a tool to improve the security of Department of Defense (DOD) information systems. This document is meant for use in conjunction with other STIGs, such as the Red Hat Enterprise Linux 8.x STIG or other appropriate operating system STIGs, Postgres DB, AAA, Central Logging, Enclave, Network Infrastructure, and appropriate application SRGs and/or STIGs.

This STIG applies to the automation controller component of the Red Hat product Ansible Automation Platform version 2.2. It assumes this product is installed and configured in accordance with the documented installation instructions provided by Red Hat. This STIG also assumes delegation of certain security control implementation to external, integrated enterprise systems including a central identity and access management system, central logging management system, a database, and others as noted below.

1.2 Authority

Department of Defense Instruction (DODI) 8500.01 requires that “all IT [information technology] that receives, processes, stores, displays, or transmits DOD information will be [...] configured [...] consistent with applicable DOD cybersecurity policies, standards, and architectures.” The instruction tasks that DISA “develops and maintains control correlation identifiers (CCIs), security requirements guides (SRGs), security technical implementation guides (STIGs), and mobile code risk categories and usage guides that implement and are consistent with DOD cybersecurity policies, standards, architectures, security controls, and validation procedures, with the support of the NSA/CSS [National Security Agency/Central Security Service], using input from stakeholders, and using automation whenever possible.” This document is provided under the authority of DODI 8500.01.

Although the use of the principles and guidelines in these SRGs/STIGs provides an environment that contributes to the security requirements of DOD systems, applicable NIST SP 800-53 cybersecurity controls must be applied to all systems and architectures based on the Committee on National Security Systems (CNSS) Instruction (CNSSI) 1253.

1.3 Vulnerability Severity Category Code Definitions

Severity Category Codes (referred to as CAT) are a measure of vulnerabilities used to assess a facility or system security posture. Each security policy specified in this document is assigned a Severity Category Code of CAT I, II, or III.

Table 1-1: Vulnerability Severity Category Code Definitions

Category	DISA Category Code Guidelines
CAT I	Any vulnerability, the exploitation of which will directly and immediately result in loss of Confidentiality, Availability, or Integrity.
CAT II	Any vulnerability, the exploitation of which has a potential to result in loss of Confidentiality, Availability, or Integrity.
CAT III	Any vulnerability, the existence of which degrades measures to protect against loss of Confidentiality, Availability, or Integrity.

1.4 STIG Distribution

Parties within the DOD and federal government's computing environments can obtain the applicable STIG from the DOD Cyber Exchange website at <https://cyber.mil/>. This site contains the latest copies of STIGs, SRGs, and other related security information. Those without a Common Access Card (CAC) that has DOD Certificates can obtain the STIG from <https://public.cyber.mil/>.

1.5 SRG Compliance Reporting

All technical NIST SP 800-53 requirements were considered while developing this STIG. Requirements that are applicable and configurable will be included in the final STIG. A report marked Controlled Unclassified Information (CUI) will be available for items that did not meet requirements. This report will be available to component authorizing official (AO) personnel for risk assessment purposes by request via email to: disa.stig_spt@mail.mil.

1.6 Document Revisions

Comments or proposed revisions to this document should be sent via email to the following address: disa.stig_spt@mail.mil. DISA will coordinate all change requests with the relevant DOD organizations before inclusion in this document. Approved changes will be made in accordance with the DISA maintenance release schedule.

1.7 Other Considerations

DISA accepts no liability for the consequences of applying specific configuration settings made on the basis of the SRGs/STIGs. It must be noted that the configuration settings specified should be evaluated in a local, representative test environment before implementation in a production environment, especially within large user populations. The extensive variety of environments makes it impossible to test these configuration settings for all potential software configurations.

For some production environments, failure to test before implementation may lead to a loss of required functionality. Evaluating the risks and benefits to a system's particular circumstances and requirements is the system owner's responsibility. The evaluated risks resulting from not applying specified configuration settings must be approved by the responsible AO. Furthermore, DISA implies no warranty that the application of all specified configurations will make a system 100 percent secure.

Security guidance is provided for the DOD. While other agencies and organizations are free to use it, care must be given to ensure that all applicable security guidance is applied at both the device hardening level and the architectural level because some settings may not be configurable in environments outside the DOD architecture.

1.8 Product Approval Disclaimer

STIGs provide configurable operational security guidance for products being used by the DOD. STIGs, along with vendor confidential documentation, also provide a basis for assessing compliance with cybersecurity controls/control enhancements, which supports system assessment and authorization (A&A) under the DOD Risk Management Framework (RMF). Department of Defense AOs may request available vendor confidential documentation for a product that has a STIG for product evaluation and RMF purposes from disa.stig_spt@mail.mil. This documentation is not published for general access to protect the vendor's proprietary information.

AOs have the purview to determine product use/approval in accordance with (IAW) DOD policy and through RMF risk acceptance. Inputs into acquisition or pre-acquisition product selection include such processes as:

- National Information Assurance Partnership (NIAP) evaluation for National Security Systems (NSS) (<https://www.niap-ccevs.org/>) IAW CNSSP #11.
- National Institute of Standards and Technology (NIST) Cryptographic Module Validation Program (CMVP) (<https://csrc.nist.gov/groups/STM/cmvp/>) IAW federal/DOD mandated standards.

2. ASSESSMENT CONSIDERATIONS

The Ansible Automation Platform is an enterprise automation tool that is generally deployed in production environments. Best practices and this STIG generally implement security controls in these types of systems via a collection of integrated components forming a single system; this STIG assumes delegation of controls per this section.

2.1 Security Assessment Information – Controller Guidance Only

This STIG is not intended to provide technical guidance for all portions of the Ansible Automation Platform 2.2. This platform is composed of multiple components, many of which are not mandatory for use and may or may not be deployed. The central and primary management component—the Ansible Automation Platform automation controller—is critical to all installations and operations and is the only subject of this STIG. This component is discussed in the context of other platform components more fully described in section 3.2. Security assessors should familiarize themselves with other platform components that may affect the security posture of the overall platform. If optional platform components are used, such as Ansible Automation Platform Mesh, system owners and security assessors should follow best practices per the associated architectural description and model of the platform as outlined in section 3.

2.2 Security Assessment Information – External Identity and Access Management

This STIG delegates certain security controls to an external Identity and Access Management system; delegated controls are out of scope of this STIG. Security controls in this STIG ensure the correct integration of this external system but thereafter rely on the security capabilities of this system. Security Assessors should ensure the choice of Identity and Access Management system that Ansible Automation Platform 2.2 is integrated with provides the following capabilities:

- Identification of inactive users: Users that have not logged into the Ansible Automation Platform within a designated, organizationally defined period must be identified by the external Identity and Access Management system and organizationally defined remedial actions taken. Typical actions include removal of users from the authorized user's group or list maintained by the external Identity and Access Management system such that subsequent attempts to access the Ansible Automation Platform are denied and appropriate logs, system events, and other tracking information is generated by the Identity and Access Management system.
- PKI user identity mapping: external Identity and Access Management system must map the authenticated identity extracted from a user's PKI certification credential to the individual user or group account provided during PKI-based authentication to the Ansible Automation Platform.
- (SSH) Key management: The Ansible Automation Platform typically connects to managed nodes via SSH. Generation, distribution, protection of, decommissioning, revocation, rotation, replacement, break-glass, and other key management and life cycle functions must be carried out in a manner that protects systems under management by the Ansible platform. It is common that IAM or specialized KMS services are used for this purpose.
- Role and Privilege Changes: When device, host, user, or other roles or security relevant attributes change, the external Identity and Access Management system must remove, delete,

or render inoperable all associated and currently active authorizations to the Ansible Automation Platform.

The product allows the use of several supported providers including Active Directory, LDAP, and others. This STIG was tested and validated using an LDAP provider.

2.3 Security Assessment Information – External Central Logging

This STIG delegates certain security controls to an external Log Collector and Management system; delegated controls are out of scope of this STIG. Security controls in this STIG ensure the correct integration of this external system, but thereafter rely on the security capabilities of this system. Security Assessors should ensure the choice of Log Collector and Management system that Ansible Automation Platform 2.2 is integrated with provide the following capabilities:

- **Log Access:** The external central logging provider must provide access controls and services for appropriate users to search, locate, and recover logs, log metadata, and activity records. The automation controller produces log and audit information based on its operations, and the external log system must be configured to parse and process the log records produced by the controller for effective investigation of security exceptions, diagnostics, incident response, or other events.
- **Log retention:** The external central logging provider must retain access, activity, and audit logs per organization-defined policy. This STIG and the automation controller provide for limited host-based log retention. But host-based storage or other resource limits associated with the Ansible Automation Platform may be insufficient to maintain sufficient log or audit records that central logging systems are equipped to retain.
- **Notification of failure:** The external central logging provider must notify log administrators under organizationally defined failure conditions. This includes failure to connect to or receive log messages from the integration Ansible Automation Platform.

The product allows the use of several supported central logging providers including Splunk, Loggly, Sumologic, Elastic stack (formerly ELK stack), and others. This STIG was tested and validated using a Splunk log collector.

2.4 Security Assessment Information – Host System

This STIG delegates certain security controls to the underlying general-purpose operating system which is required to have an appropriate STIG applied to it; delegated controls are out of scope of this STIG. Security controls in this STIG ensure the correct integration of this underlying operating system, but thereafter rely on the security capabilities of this system. As noted in section 1.1, Ansible Automation Platform 2.2 must be installed according to published installation instructions, including the use of a validated underlying host operating system for all hosts. It is expected this host operating system will provide the following capabilities:

- **Managing FIPS modules:** The underlying operating system must ensure the operational state of FIPS modules including startup, self-test, and entropy tests.
- **Validation of installed packages.**

The product requires the use of Red Hat Enterprise Linux 8.x configured in accordance with the applicable STIG and configured to SELinux in enforcing mode and to use FIPS mode. This STIG was tested and validated using Red Hat Enterprise Linux 8.2 as the underlying host for all hosts.

2.5 Security Assessment Information – Other Systems

This STIG delegates certain security controls to other external systems; delegated controls are out of scope of this STIG. Security controls in this STIG ensure the correct integration of this external system, but thereafter rely on the security capabilities of these other systems. Security Assessors should ensure the Ansible Automation Platform 2.2 is integrated with systems that provide the following capabilities:

- Denial of Service (DoS) protections: Other intermediary systems must provide appropriate DoS protections for the Ansible Automation Platform. These protections are normally provided by enterprise managed internet/network connection points, CDNs, hardware and software firewalls, hardware and software load balancers, hardware and software proxies, network routers and switches, network operations centers, and similar systems. The Ansible Automation Platform's NGINX web server can partially implement these functions for each host. However, this host-level protection should not be completely relied on, as it is generally inadequate for mitigating large-scale DoS/DDoS attacks. More information about using and configuring host level protections for DoS may be found at <https://www.nginx.com/blog/mitigating-ddos-attacks-with-nginx-and-nginx-plus/>. This information includes recommendations for limiting the rate of requests and number of connections, closing slow connections, denylisting IP addresses, allowlisting IP addresses, using caching to smooth traffic spikes, blocking requests, limiting connections to backend servers, and identifying and handling reconnaissance traffic.
- PostgreSQL: The platform stores state information in a database. This STIG configures the Ansible Automation Platform automation controller to use a mutual TLS connection to its backend database. It is important to note that any integrated PostgreSQL server installations must be configured in TLS mode and the TLS ciphers for these databases must match those used by the controller and per organizational policy. The PostgreSQL server configuration is outside of the scope of this STIG. If PostgreSQL server(s) is/are configured to support the appropriate protocols and ciphers, the administrator should configure Ansible Tower to enforce that encryption from the client side for all remote databases at installation time using by setting the "pg_sslmode" variable to "verify-full" in the inventory file, in accordance with the installation documentation. To implement this capability on an existing controller system, conduct a backup, reinstallation, and restore with the proper setting configured to comply with this requirement; this configuration cannot be changed once a system has been installed. More information about backing up and restoring Ansible Automation Platform can be found in the documentation here: https://docs.ansible.com/automation-controller/latest/html/administration/backup_restore.html.

3. CONCEPTS AND TERMINOLOGY CONVENTIONS

3.1 Ansible Automation Platform overview

Ansible is an open source, command-line IT automation software application written in Python. It can configure systems, deploy software, and orchestrate advanced workflows to support application deployment, system updates, and more.

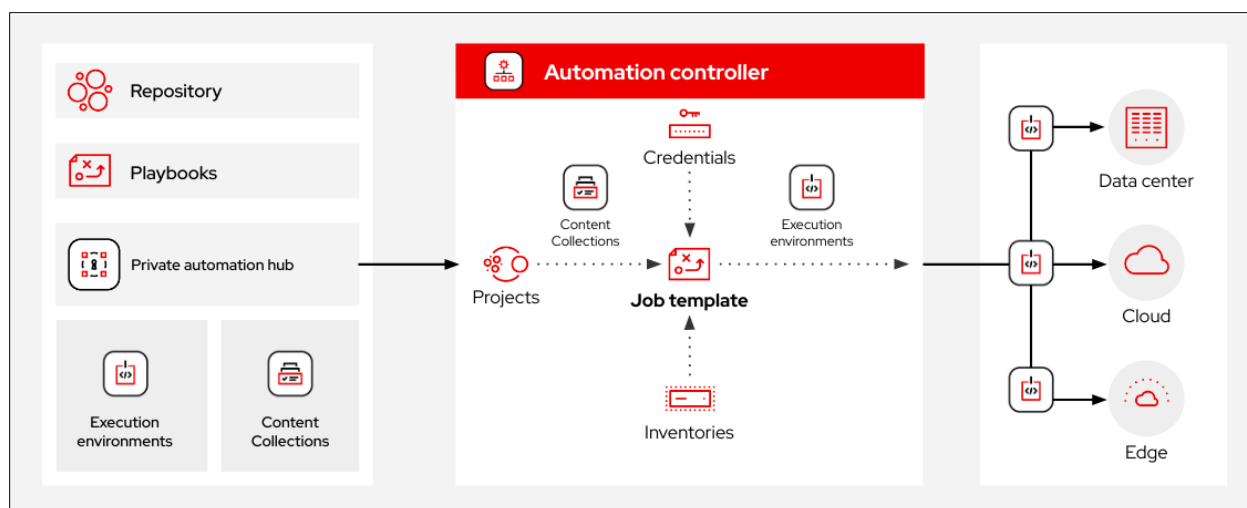
Ansible's main strengths are simplicity and ease of use. It also has a strong focus on security and reliability, featuring minimal moving parts. It uses OpenSSH for transport (with other transports and pull modes as alternatives) and uses a human-readable language that is designed for getting started quickly without extensive training.

3.2 Ansible Automation Platform Components

The Ansible Automation Platform is built from several primary components. Red Hat Ansible Automation Platform is a subscription product built on the foundations of an open-source community project Ansible with numerous enterprise features added. It combines more than a dozen upstream projects into an integrated, streamlined product. Each product component also has a specific purpose with a well-defined scope.

The automation controller is the WebUI and API for Ansible automation, which is based on the upstream project AWX. This component is bundled into the platform to manage automation. Ansible Automation Platform's automation controller (or just the controller) is the primary target component of this STIG. The controller itself consists of a web application, a web application load balancer, and connections or integrations with additional components of the Ansible Automation Platform or network accessible external components provided by the user or enterprise. Figure 3-1 shows the general architecture of the platform in the context of these other components.

Figure 3-1: Ansible Automation Platform 2.2+ General Architecture



Within the controller, the web application load balancer is provided via NGINX. While this load balancer may be used as a general-purpose tool in other contexts, this STIG requires it to service

only the controller web application. This NGINX component should not be modified from its installed configuration except per this STIG's controls and the guidance provided in this document (e.g., for host-based DoS protections).

The controller's web application is serviced via a Django Daphne ASGI/WSGI server. This server backend is a large library of Python routines to implement the controller's primary business logic. The server content, which is primarily scripts and other executable items, should not be modified except per this STIG's controls and the guidance provided in this document.

A common PostgresDB database service is shared by all controllers. This database service maintains critical operational state information. When the controllers are operating in a HA mode, this database shares state information about the controllers. For simple installations, Ansible Automation Platform may utilize a single database instance co-located with the automation controller. For production use, it is recommended that a separate but integrated database system be used, including independent HA, DR, and COOP capabilities.

3.3 Playbooks and Execution Environments

Ansible's approach to orchestration is one of finely tuned simplicity to facilitate use of existing knowledge while not having to remember special syntax or features. Playbooks are the primary artifact for encoding orchestration of multiple domains within an IT infrastructure against indicated target environments. Playbooks provide very detailed control as a set of declarative YAML instructions for tasks to execute at runtime. As playbooks imply executable processes on external systems, care must be taken to ensure this executable content is safe.

Playbooks themselves are outside the scope of this STIG, and neither automation controller nor this STIG is intended to provide protections against malicious content in the form of playbooks or other user content. System owners should provide human resource controls, test, and validation processes, CI/CD or similar processes, and other technological controls to manage and verify playbook content prior to execution.

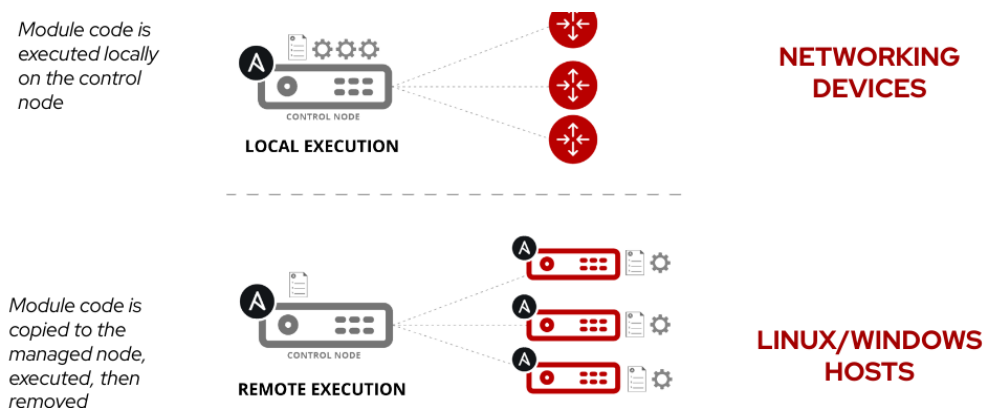
Ansible execution environments are Linux containers which provide the necessary tools and runtimes to execute ansible automation content in the form of and described by Ansible playbooks, modules, and extensions. Each execution environment is built from common, reusable based container images with necessarily dependencies including Python virtual environments, Python libraries, and other runtimes. The automation controller orchestrates the creation, deployment, monitoring, and termination of individual execution environments.

Ansible execution environments themselves are outside the scope of this STIG, and neither automation controller nor this STIG is intended to provide protections against malicious content in the form of execution environments outside the controller. System owners should provide human resource controls, test and validation processes, CI/CD or similar processes, and other technological controls to manage and verify execution environments prior to use.

3.4 Module Execution & Ansible Mesh

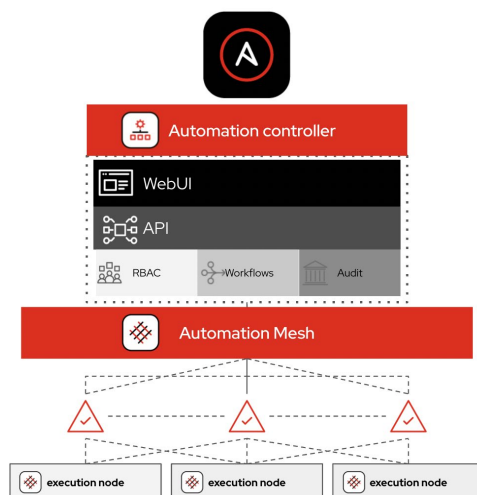
The Ansible Automation Platform has the concept of a control node and a managed node. The control node is where Ansible is executed from, for example where a user runs the ansible-playbook command. Managed nodes are the devices being automated, for example, a Linux host or Microsoft Windows server.

Figure 3-2: Network Automation Compared to Servers



For automating Linux and Windows, Ansible works by connecting to managed nodes and pushing out small programs, called "Ansible modules," to them. These programs are written to be resource models of the desired state of the system. Ansible then executes these modules (over SSH by default) and removes them when finished. These modules are designed to be idempotent when possible, so that they only make changes to a system when necessary.

Figure 3-3: Automation Mesh for Distributed Execution Environments



For automating network devices and other IT appliances where modules cannot be executed, Ansible will run on the control node. Since Ansible is agentless, it can still communicate with devices without requiring an application or service to be installed on the managed node. To

understand more about how network automation is different, refer to the Ansible documentation. To increase execution capacity for devices without the ability to run modules, Ansible Automation Platform can spread automation jobs out across execution nodes using a technology called Automation Mesh. This technology creates a secure overlay network across dedicated nodes that provide execution environments decoupled from the controller(s). To understand more about how network automation is different, refer to the Ansible documentation.

3.5 Credentialing and Inventorying

For Ansible to execute, it needs an inventory of the targeted managed nodes subject to its automation and credentials to allow the execution environments to authenticate and connect to those managed nodes.

Community Ansible is decentralized—meaning it relies on a user’s existing OS credentials to control access to remote machines. And if needed, Ansible can easily connect with Kerberos, Lightweight Directory Access Protocol (LDAP), and other centralized authentication management systems. Credentials such as usernames and passwords may be stored as variables for Ansible, encrypted and stored with Ansible Vault, or by storing credentials in an inventory file.

Red Hat Ansible Automation Platform can act as a centralized authentication as well as integrate with industry standard tools like CyberArk AIM, Conjur, HashiCorp Vault, and Microsoft Azure Key Vault. Automation controller hashes local automation controller user passwords with the PBKDF2 algorithm using a SHA256 hash. Users who authenticate via external account mechanisms (LDAP, SAML, OAuth, and others) do not have any password or secret stored. For more information, refer to [Ansible Automation Platform’s secret handling and connection security documentation](#).

4. GENERAL SECURITY REQUIREMENTS

4.1 Hardening of Integrated External Services

Ansible Automation Platform must use several external services, as noted in section 2.1. The deployment and management of these services is outside the scope of this STIG. All services that are used by Ansible Automation Platform should be hardened to an appropriate level via a well-defined risk management framework (RMF), including the use of associated STIGs as applicable. Failure to do so may allow compromise of data, managed systems, user operations, or other impacts via lateral attacks from these integrated systems.

4.2 Disaster Recovery and Continuity of Operations

All critical ICT systems should provide robust disaster recovery (DR) and continuity of operations (COOP) plans as appropriate. Ansible Automation Platform 2.2 controller provides internal mechanisms to enable high availability but more complex operations for DR and COOP are outside the scope of this STIG. These operations normally require significant planning and people and processes outside a single product. More information about architectural options and best practices for configuring and operating the Ansible Automation Platform in these scenarios can be found at <https://github.com/redhat-cop/automate-tower-ha-dr>.

More information about backing up and restoring Ansible Automation Platform can be found in the documentation here: https://docs.ansible.com/automation-controller/latest/html/administration/backup_restore.html.