 Automation Platform Integration

With CyberArk’s Dynamic Access Provider (DAP)

Table of Contents

[Red Hat Ansible Engine & Ansible project 1](#_Toc36480954)

[Lookup Plugin – conjur\_variable 1](#_Toc36480955)

[Requirements 1](#_Toc36480956)

[Parameters 2](#_Toc36480957)

[Usage Examples 2](#_Toc36480958)

[Role – conjur-host-identity 3](#_Toc36480959)

[Requirements 3](#_Toc36480960)

[Installation 3](#_Toc36480961)

[Role Variables 3](#_Toc36480962)

[Usage Examples 3](#_Toc36480963)

[Red Hat Ansible Tower 4](#_Toc36480964)

[Secrets Management System Lookup – CyberArk Conjur Secret Lookup 4](#_Toc36480965)

[Requirements 4](#_Toc36480966)

[Configuration 4](#_Toc36480967)

# Red Hat Ansible Engine & Ansible project

## Lookup Plugin – conjur\_variable

Using the lookup plugin conjur\_variable in your playbook allows you to securely retrieve secrets from Dynamic Access Provider (DAP) at task run time.

### Requirements

* Ansible v2.5 or above.
* The Ansible controller must have a [host identity from DAP](https://docs.cyberark.com/Product-Doc/OnlineHelp/AAM-DAP/Latest/en/Content/Operations/Policy/statement-ref-host.htm).
* The DAP config file (conjur.conf) and the DAP identity file (conjur.identity) should be located in /etc/ on the Ansible controller.
* Network access from Ansible controller to DAP instance.

**Note:** Environment variable delivery of the host identity values from DAP to the lookup plugin is not supported.

### Parameters

* config\_file
	+ Default: /etc/conjur.conf
	+ Variable Name: CONJUR\_CONFIG\_FILE
* identity\_file
	+ Default: /etc/conjur.identity
	+ Variable Name: CONJUR\_IDENTITY\_FILE

### Usage Examples

#### Common Usage Example

**---**

**- name: conjur\_variable Common Usage Example**

 **hosts: all**

 **tasks:**

 **- name: Retrieve secret from Dynamic Access Provider (DAP)**

 **set\_fact:**

 **conjur\_response: "{{ lookup('conjur\_variable', '/path/to/secret') }}"**

 **no\_log: yes**

 **- name: Print secret to output (Don't do this except for testing purposes)**

 **debug:**

 **msg: "{{ conjur\_response }}"**

#### Non-Default Host Identity File Paths

**---**

**- name: conjur\_variable Non-Default Host Identity File Paths**

 **hosts: all**

 **vars:**

 **CONJUR\_CONFIG\_FILE: /user/awx/conjur.conf**

 **CONJUR\_IDENTITY\_FILE: /user/awx/conjur.identity**

 **tasks:**

 **- name: Retrieve secret from Dynamic Access Provider (DAP)**

 **set\_fact:**

 **conjur\_response: "{{ lookup('conjur\_variable', '/path/to/secret') }}"**

 **no\_log: yes**

 **- name: Print secret to output (Don't do this except for testing purposes)**

 **debug:**

 **msg: "{{ conjur\_response }}"**

## Role – conjur-host-identity

Using the Role conjur-host-identity allows simple DAP host identity creation & deployment onto target nodes. This also includes deployment of [Summon](https://cyberark.github.io/summon) and [Summon-Conjur](https://github.com/cyberark/summon-conjur) provider onto the target node.

### Requirements

* Ansible 2.3 or above.
* EL 7, Ubuntu Trusty (14.04), Ubuntu Xenial (16.04)
* Network access from target node to DAP instance.

### Installation

ansible-galaxy install cyberark.conjur-host-identity

### Role Variables

* conjur\_appliance\_url\*
	+ URL of the running DAP instance.
* conjur\_account\*
	+ DAP account name.
* conjur\_host\_factory\_token\*
	+ [Host Factory](https://docs.cyberark.com/Product-Doc/OnlineHelp/AAM-DAP/Latest/en/Content/Operations/Services/host_factory.html) token for target node grant to [layer](https://docs.cyberark.com/Product-Doc/OnlineHelp/AAM-DAP/Latest/en/Content/Operations/Policy/statement-ref-layer.htm#Usage) role.
	+ This should be an environment variable on the Ansible controller.
* conjur\_host\_name\*
	+ Name of the host being granted DAP host identity.
* conjur\_ssl\_certificate\*
	+ Public SSL certificate of the DAP instance.
* conjur\_validate\_certs
	+ Boolean value to indicate if the DAP instance should validate certificates.
	+ Default: True/Yes
* summon.version
	+ Version of [Summon](https://cyberark.github.io/summon/) to install.
	+ Default: 0.6.6
* summon\_conjur.version
	+ Version of [Summon-Conjur](https://github.com/cyberark/summon-conjur) provider to install.
	+ Default: 0.5.0

### Usage Examples

#### Common Usage Example

**---**

**- name: Configure target node with DAP Host Identity & Summon**

 **hosts: servers**

 **roles:**

 **- role: cyberark.conjur-host-identity**

 **conjur\_appliance\_url: https://dap.example.com/api**

 **conjur\_account: myorg**

 **conjur\_host\_factory\_token: "{{ lookup('env', 'HFTOKEN') }}"**

 **conjur\_host\_name: "{{ inventory\_hostname }}"**

 **conjur\_ssl\_certificate: "{{ lookup('file', '/home/awx/dap.pem') }}"**

# Red Hat Ansible Tower

## Secrets Management System Lookup – CyberArk Conjur Secret Lookup

Using the native Credentials feature within Ansible Tower, a credential type is included for connecting your Tower controller to DAP in order to provide secrets to jobs at run time.

### Requirements

* Ansible Tower v3.5 or above.
* Network access from Ansible Tower node to DAP instance.

### Configuration

#### Creating the CyberArk Conjur Secret Lookup

1. In the Credentials tab, add a new Credential.
2. Fill out the Name, Description, Organization.

3. Select the “CyberArk Conjur Secret Lookup” credential type.

4. Under Type Details, complete the following text fields:
	1. Conjur URL
		1. URL to DAP instance.
	2. API Key
		1. Ansible Tower’s host identity API Key created in DAP.
		2. After the Credential is saved, this value will be encrypted and will not be allowed to be viewed by anyone again.
	3. Account
		1. DAP account name.
	4. Username
		1. Ansible Tower’s host identity created in DAP.
		2. Must be prefixed with host/.
	5. Public Key Certificate
		1. The public SSL certificate contents for DAP.



#### Testing the CyberArk Conjur Secret Lookup

1. While viewing the Credential for CyberArk Conjur Secret Lookup in Ansible Tower, click the blue Test button next to Cancel and Save.

2. In the Secret Identifier text field, add the path to the secret variable as it is in DAP. Optionally, you can include a version of the secret to return. Default is to return the latest version.

3. Click Run and you will receive a toast alert as to the Success or Failure of the test.


#### Creating a new credential retrieved from CyberArk DAP

1. Create a new credential that you would like to be retrieved from CyberArk DAP.

2. Click the  icon to the left of the text field which should contain the value of the DAP secret variable.
3. Select the name for your CyberArk Conjur Secret Lookup and click Next.

4. In the Secret Identifier text field, add the path to the secret variable as it is in DAP. Optionally, you can include a version of the secret to return. Default is to return the latest version.

5. Before saving, click the blue Test button to make sure the secret can be retrieved properly.

6. If the test passed, click Save.