



Ansible on Windows Fundamentals

Preparing Hosts for Automation

Preparing Microsoft Windows Hosts for Ansible

Objective

- Enable Windows-based managed hosts to accept Ansible connections

Requirements for Windows-Based Managed Hosts

- Supported Microsoft Windows operating system versions:
 - Windows Server (2008, 2008 R2, 2012, 2012 R2, 2016, or 2019)
 - Windows 7, 8.1, or 10
- To manage a Windows-based server, Ansible must connect and run code
 - WinRM must be enabled
 - Ansible must be able to authenticate to the managed host
 - The managed host must have PowerShell 3.0 or newer and .NET Framework 4.0 or newer (Windows Server 2012 and later and Windows 8.1 and later has the right software pre-installed)

Configuring Windows for Ansible Access

- WinRM is a Microsoft implementation of the SOAP-based WS-Management standard protocol.
- It is enabled by default since Windows Server 2012.
- Ansible uses the PowerShell Remoting Protocol (PSRP) on top of WinRM to execute PowerShell commands.
- Ansible 2.8 provides an experimental feature allowing you to use SSH instead of WinRM to connect to Windows hosts.

Configuring for Development Environments

- Find the configuration script in the official Ansible repository on GitHub:
<https://github.com/ansible/ansible/blob/devel/examples/scripts/ConfigureRemotingForAnsible.ps1>
- When run as an Administrator on a managed host, the script performs the following actions:
 - Confirms PowerShell version 3 or higher is installed.
 - Runs the WinRM service and configures it to automatically start at boot.
 - Enables PowerShell Remoting and an SSL Listener
 - Set the `LocalAccountTokenFilterPolicy` registry key to 1.
 - Optionally configures Basic or CredSSP authentication.
 - Configures the Windows firewall to allow WinRM connections over both HTTP and HTTPS.

Options for the ConfigureRemotingForAnsible.ps1 Script

<code>-EnableCredSSP</code>	Enables the CredSSP authentication protocol.
<code>-DisableBasicAuth</code>	Disables Basic authentication. Recommended.
<code>-CertValidityDays</code>	Days until the HTTPS SSL certificate expires. Default is 1095 days (3 years).
<code>-ForceNewSSLCert</code>	Forces the creation of a new SSL certificate.
<code>-SubjectName</code>	Sets the Common Name of the SSL certificate.
<code>-SkipNetworkProfileCheck</code>	Enable PS Remoting without checking the network profile.
<code>-Verbose</code>	Enable verbose output for debugging.

Configuring Remoting in Production Environments

- `ConfigureRemotingForAnsible.ps1` was designed for lab/development environments
- For production, tailor the script to meet your security requirements
- You should have a way to automatically enable WinRM and remote authentication when a new Windows managed host is provisioned

Authentication

You can select from one of several methods for Ansible to authenticate to Windows managed hosts, using either a local or domain user accounts:

OPTION	LOCAL ACCOUNTS	ACTIVE DIRECTORY ACCOUNTS	CREDENTIALS DELEGATION	HTTP ENCRYPTION
Basic	Yes	No	No	No
Certificate	Yes	No	No	No
Kerberos	No	Yes	Yes	Yes
NTLM	Yes	Yes	No	Yes
CredSSP	Yes	Yes	Yes	Yes

Authentication

- For security reasons, avoid Basic and Certificate authentication
- NTLM might be enabled by default for WinRM, but it also has security weaknesses
- CredSSP is the best choice if you authenticate with local (not domain) user accounts
 - Include the **-EnableCredSSP** argument when running **ConfigureRemotingForAnsible.ps1**
 - Supports authentication for both local and domain accounts.
 - Encrypts credentials prior to transmitting them to the target host.
 - Credential delegation allows your target server to forward your credentials to a second server.

Authentication

- Kerberos is recommended if you are using a domain user account
 - The Windows-based managed hosts must be members of the domain
 - Only supports authentication with domain user accounts
 - The domain controller, Ansible control node, and managed host use encrypted credentials
 - Default Microsoft network authentication protocol
- Your control node needs to be configured with the authentication credentials for the method you pick
- That will be covered in the next presentation

Preparing Red Hat Ansible Tower for Windows Management

Objectives

- Create an inventory of managed hosts in Red Hat Ansible Tower
- Configure the inventory to use the correct credentials to authenticate to the managed hosts

Preparing Red Hat Ansible Tower to Manage Hosts

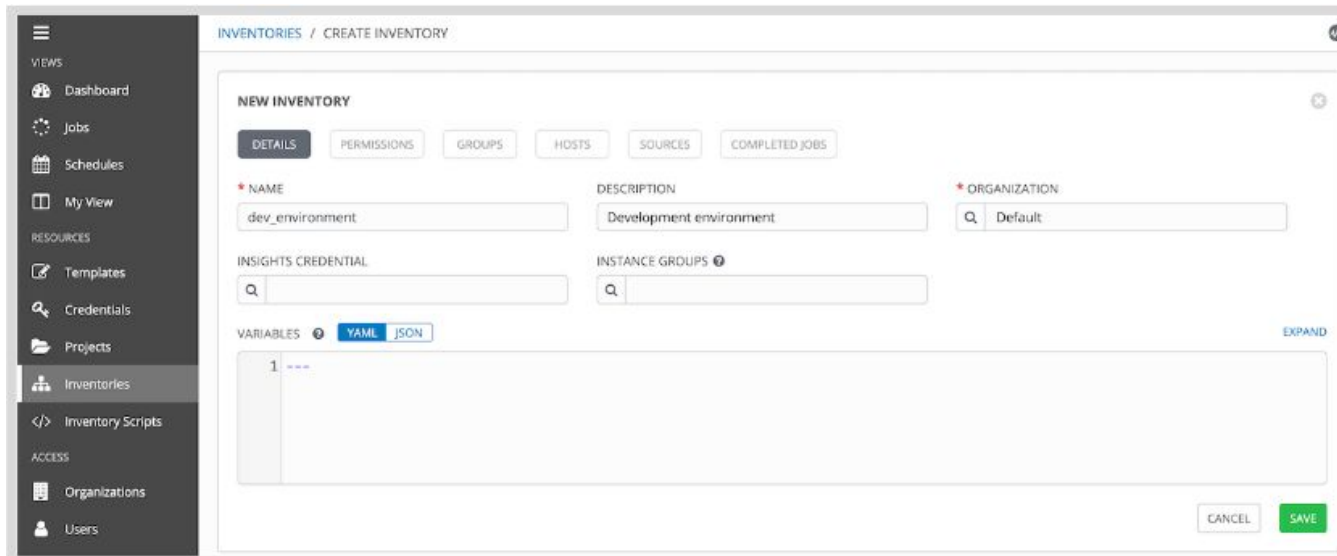
- An **inventory** is a collection of hosts and groups of hosts which are managed by Ansible
- Before you can run a playbook with Ansible Tower, you have to configure it with an inventory
- You can also set connection variables for those hosts to configure authentication settings
- More than one inventory can be configured
- Different inventories can be used for different purposes

Creating a Static Inventory in Ansible Tower

- The simplest kind of inventory to set up is a static inventory that you manually write
- Two ways to set up a static inventory:
 - Manage it from Ansible Tower's web-based user interface
 - Use an existing inventory file stored in a version control system
- This presentation will illustrate the first option.
- For more information on the second option, see <https://docs.ansible.com/ansible-tower/latest/html/administration/scm-inv-import.html>

Creating an Inventory in the Ansible Tower Web UI

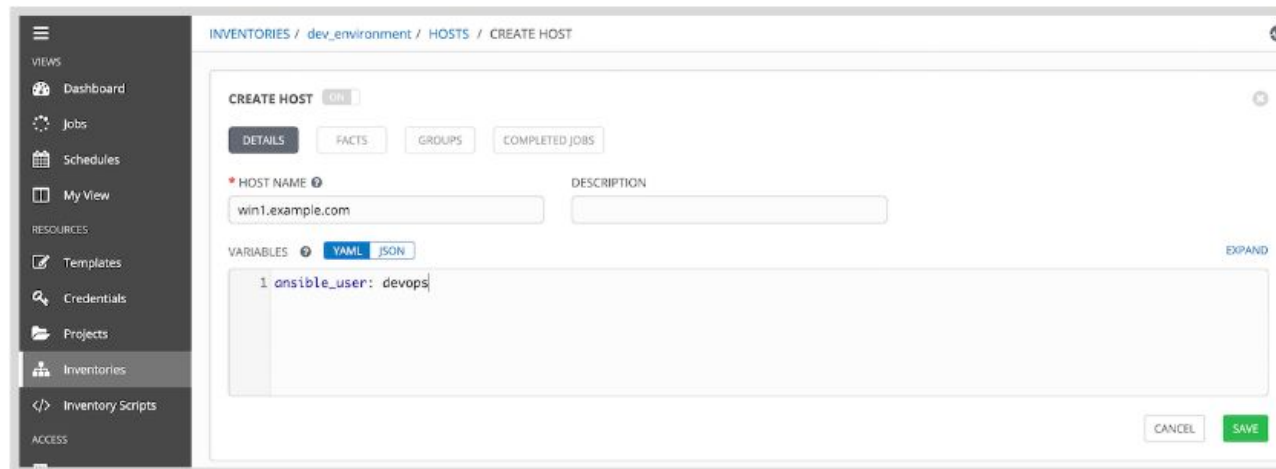
- Log into Ansible Tower (the admin user will work for this example).
- Click on **Inventories**.
- In the INVENTORIES window, click the + button.
- Enter a NAME for the inventory and its ORGANIZATION (often “Default”)



The screenshot shows the 'NEW INVENTORY' form in the Ansible Tower Web UI. The left sidebar contains navigation links: VIEWS (Dashboard, Jobs, Schedules, My View), RESOURCES (Templates, Credentials, Projects, Inventories, Inventory Scripts), and ACCESS (Organizations, Users). The main content area is titled 'INVENTORIES / CREATE INVENTORY' and features tabs for DETAILS, PERMISSIONS, GROUPS, HOSTS, SOURCES, and COMPLETED JOBS. The 'DETAILS' tab is active, showing fields for NAME (dev_environment), DESCRIPTION (Development environment), ORGANIZATION (Default), INSIGHTS CREDENTIAL, and INSTANCE GROUPS. A VARIABLES section at the bottom allows selection between YAML and JSON, with a text area for input. CANCEL and SAVE buttons are located at the bottom right.

Adding a Host to an Inventory

- In the Ansible Tower GUI, click the **Inventories** menu, then click on the name of the inventory.
 - Click the **HOSTS** button, then click on **+**. This displays the “Create a new host” tooltip.
 - In the HOST NAME field enter the hostname or IP address of the managed host.
 - In the VARIABLES text box, you can set values for variables that apply only to this host.
 - Click **SAVE**.



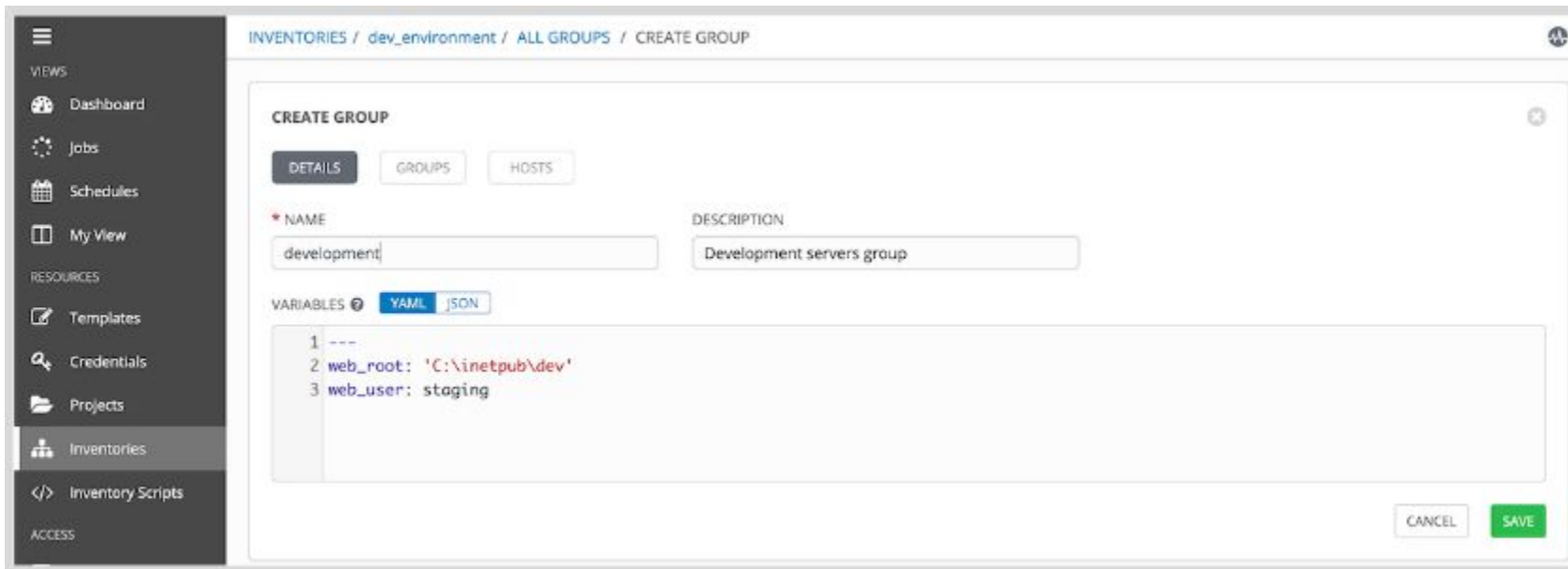
The screenshot shows the 'CREATE HOST' form in the Ansible Tower GUI. The breadcrumb navigation at the top reads 'INVENTORIES / dev_environment / HOSTS / CREATE HOST'. On the left is a dark sidebar with a menu containing 'Dashboard', 'Jobs', 'Schedules', 'My View', 'Templates', 'Credentials', 'Projects', 'Inventories' (highlighted), 'Inventory Scripts', and 'ACCESS'. The main content area has a 'CREATE HOST' toggle set to 'ON'. Below it are tabs for 'DETAILS' (selected), 'FACTS', 'GROUPS', and 'COMPLETED JOBS'. The 'DETAILS' tab contains a 'HOST NAME' field with the value 'win1.example.com' and an empty 'DESCRIPTION' field. Below these is a 'VARIABLES' section with a dropdown set to 'YAML' and a text area containing '1 ansible_user: devops'. An 'EXPAND' link is to the right of the text area. At the bottom right are 'CANCEL' and 'SAVE' buttons.

Organizing Hosts into Groups

- Groups allow you to organize hosts into a set that can be managed together
- Hosts may be in multiple groups at the same time
 - All hosts that are in a particular data center
 - All hosts that have a particular purpose
 - Dev / Test / Prod hosts can be grouped
- Groups can be nested
 - The *europa* group might include a *paris_dc* group and a *london_dc* group
- This allows you to run playbooks on particular groups
- This allows you to set a variable to a specific value for all hosts in a group

Creating a Group

- In the Ansible Tower GUI, click the **Inventories** menu, and click on the inventory to edit.
- Click the **GROUPS** button, then click on **+**. This will open the “Create a new group” tooltip.
- In the NAME field, enter the name of the group.
- Define any values for variables
- Click **SAVE**.



The screenshot shows the 'CREATE GROUP' form in the Ansible Tower GUI. The breadcrumb trail at the top reads 'INVENTORIES / dev_environment / ALL GROUPS / CREATE GROUP'. The left sidebar contains a menu with 'Inventories' selected. The form has three tabs: 'DETAILS' (active), 'GROUPS', and 'HOSTS'. The 'NAME' field contains 'development' and the 'DESCRIPTION' field contains 'Development servers group'. The 'VARIABLES' section has a dropdown set to 'YAML' and a text area containing the following YAML code:

```
1 ---
2 web_root: 'C:\inetpub\dev'
3 web_user: staging
```

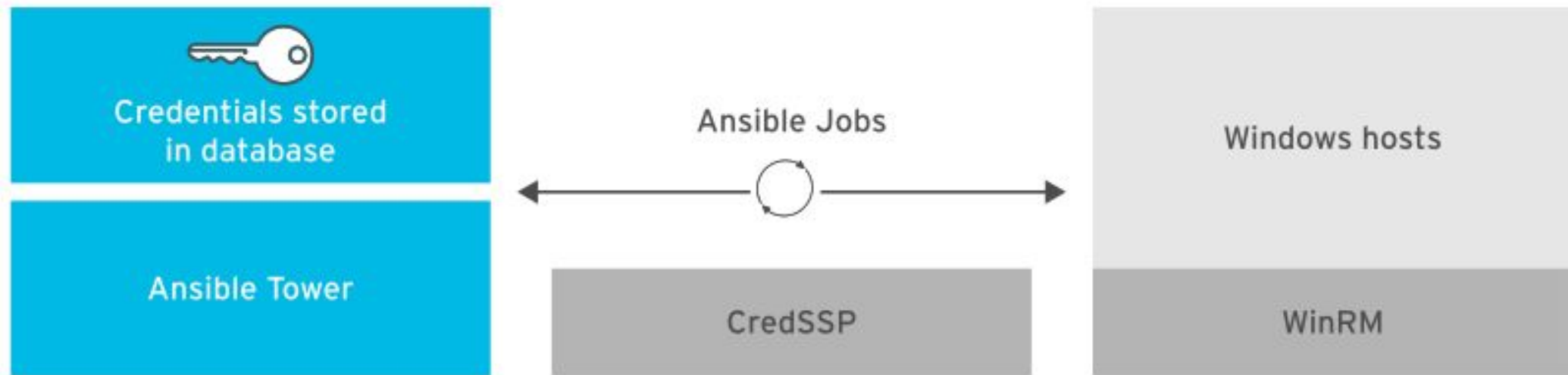
At the bottom right of the form are 'CANCEL' and 'SAVE' buttons.

Adding a New Host to a Group

- In the Ansible Tower GUI, click the **Inventories** menu, and click on the inventory to edit.
- Click the **GROUPS** button, then click on the group to edit.
- Click the **HOSTS** button, then click on **+**. This will open the “Add a host” tooltip. Select “New Host”.
- In the HOST NAME field, enter the hostname or IP address of the managed host to add.
- Define any values for variables that affect only that host (overriding any group variables).
- Click **SAVE**.

Connection Variables and Microsoft Windows Hosts

- Ansible Tower uses credentials stored in its database to connect to Windows hosts.
- The connection is used to run playbooks or ad hoc modules.
- Many of the connection settings can be configured with variables.



Configuring Connection Variables for Windows Hosts

- There are advantages to putting all Windows-based managed hosts in a group
 - Collective management of all Windows servers
 - If they all use the same credentials for authentication, the connection variables can be set on that group

Configuring Basic Authentication

- There are two main Ansible connection variables to set to enable Basic authentication:

Variable	Purpose
<code>ansible_connection</code>	Protocol to use to connect to the host (use winrm)
<code>ansible_winrm_transport</code>	The authentication method to use for WinRM (use basic)

This mechanism is simple to set up but **is not secure, and should be avoided**. It exposes the user's password on the network in a way that can be easily decoded if a secure channel is not in use.

More details are available at
https://docs.ansible.com/ansible/latest/user_guide/windows_winrm.html#basic

Configuring CredSSP Authentication

- Use this instead of Basic authentication.
- There are four main Ansible connection variables to set to enable CredSSP authentication:

Variable	Purpose
ansible_connection	Protocol to use to connect to the host (use winrm)
ansible_port	Network port to use with that protocol (use 5986)
ansible_winrm_transport	The authentication method to use for WinRM (use credssp)
ansible_winrm_server_cert_validation	Whether or not to validate the WinRM TLS certificate

More details are available at

https://docs.ansible.com/ansible/latest/user_guide/windows_winrm.html#credssp

CredSSP Authentication Example

Default inventory

DETAILS PERMISSIONS GROUPS HOSTS SOURCES COMPLETED JOBS

* NAME: Default inventory

DESCRIPTION: This is the default inventory

* ORGANIZATION: Default

INSIGHTS CREDENTIAL: [Search]

INSTANCE GROUPS: [Search]

VARIABLES ? **YAML** JSON EXPAND

```
1 ansible_connection: winrm
2 ansible_port: 5986
3 ansible_winrm_server_cert_validation: ignore
4 ansible_winrm_transport: credssp
```

- 1 Instructs Ansible Tower to use WinRM as a connection method.
- 2 Instructs Ansible to use port 5986 for TLS encryption.
- 3 Instructs Ansible to ignore the Certificate Authority Signature.
- 4 Instructs Windows Ansible to use the CredSSP transport method.

Configuring Kerberos Authentication

- The managed host needs to be joined to the domain and you need a domain user for Ansible to use
- There are two main Ansible connection variables to set to enable Kerberos authentication:

Variable	Purpose
ansible_connection	Protocol to use to connect to the host (use winrm)
ansible_winrm_transport	The authentication method to use for WinRM (use kerberos)

More details are available at
https://docs.ansible.com/ansible/latest/user_guide/windows_winrm.html#kerberos

Creating Machine Credentials

- To securely provide the `ansible_user` and `ansible_password` setting for managed hosts, create a machine credential
- Users of Ansible Tower can use these credentials, but cannot retrieve the value of the password directly
- The machine credential's password is stored in encrypted form in Ansible Tower

Creating Machine Credentials

- In the Ansible Tower web UI, click **Credentials**, then click **+** to “Create a new credential”.
 - For NAME, enter a name for your machine credential.
 - Select your ORGANIZATION (often “Default”).
 - For CREDENTIAL TYPE, select **Machine**, then click **SELECT**.
 - For USERNAME, enter the name of the Windows user you use to authenticate.
 - For PASSWORD, enter the password for that Windows user.
 - Click **SAVE**.
-
- When you set up your Ansible Playbook’s job template, it can use this credential.
 - It is simpler if you can use the same user name and password for a large number of hosts.