* 3 minutes

Playbooks are the language of Ansible’s configurations, deployments, and orchestrations.

You use them to manage configurations of and deployments to remote machines.

Playbooks are structured with YAML (a data serialization language) and support variables.

Playbooks are declarative and include detailed information about the number of machines to configure at a time.

## YML structure

YAML is based on the structure of key-value pairs. In the following example, the key is name, and the value is namevalue:

Name: namevalue

In the YAML syntax, a child key-value pair is placed on a new and indented line below its parent key.

Each sibling key-value pair occurs on a new line at the same indentation level as it.

Parent:  
 Children:  
 First-sibling: value01  
 Second-sibling: value02

The specific number of spaces used for indentation isn’t defined. You can indent each level by as many spaces as you want.

However, the number of spaces used for indentations at each level must be uniform throughout the file.

When there’s an indentation in a YAML file, the indented key-value pair is the value of its parent key.

## Playbook components

The following list is of some of the playbook components:

* Name. The name of the playbook. It can be any name you wish.
* Hosts. Lists where the configuration is applied or machines being targeted. Hosts can be a list of one or more groups or host patterns, separated by colons. It can also contain groups such as web servers or databases, providing that you’ve defined these groups in your inventory.
* Connection. Specifies the connection type.
* remote\_user. Specifies the user that will be connected to for completing the tasks.
* var. Allows you to define the variables that can be used throughout your playbook.
* gather\_facts. Determines whether to gather node data or not. The value can be yes or no.
* Tasks. Indicates the start of the modules where the actual configuration is defined.

## Running a playbook

You run a playbook using the following command:

Ansible-playbook < playbook name >

You can also check the syntax of a playbook using the following command.

ansible-playbook --syntax-check

The syntax check command runs a playbook through the parser.

It’s to verify it has included items. For example, files and roles, and that the playbook has no syntax errors.

You can also use the --verbose command.

* To see a list of hosts that would be affected by running a playbook, run the command:

Ansible-playbook playbook.yml --list-hosts

## Sample playbook

The following code is a sample playbook that will create a Linux virtual machine in Azure:

- name: Create Azure VM.  
 hosts: localhost  
 connection: local  
 vars:  
 resource\_group: ansible\_rg5  
 location: westus  
 tasks:  
 - name: Create resource group.  
 azure\_rm\_resourcegroup:  
 name: "{{ resource\_group }}"  
 location: "{{ location }}"  
 - name: Create virtual network.  
 azure\_rm\_virtualnetwork:  
 resource\_group: myResourceGroup  
 name: myVnet  
 address\_prefixes: "10.0.0.0/16"  
 - name: Add subnet.  
 azure\_rm\_subnet:  
 resource\_group: myResourceGroup  
 name: mySubnet  
 address\_prefix: "10.0.1.0/24"  
 virtual\_network: myVnet  
 - name: Create public IP address.  
 azure\_rm\_publicipaddress:  
 resource\_group: myResourceGroup  
 allocation\_method: Static  
 name: myPublicIP  
 register: output\_ip\_address  
 - name: Dump public IP for VM, which will be created.  
 debug:  
 msg: "The public IP is {{ output\_ip\_address.state.ip\_address }}."  
 - name: Create Network Security Group that allows SSH.  
 azure\_rm\_securitygroup:  
 resource\_group: myResourceGroup  
 name: myNetworkSecurityGroup  
 rules:  
 - name: SSH  
 protocol: Tcp  
 destination\_port\_range: 22  
 access: Allow  
 priority: 1001  
 direction: Inbound  
 - name: Create virtual network interface card.  
 azure\_rm\_networkinterface:  
 resource\_group: myResourceGroup  
 name: myNIC  
 virtual\_network: myVnet  
 subnet: mySubnet  
 public\_ip\_name: myPublicIP  
 security\_group: myNetworkSecurityGroup  
 - name: Create VM.  
 azure\_rm\_virtualmachine:  
 resource\_group: myResourceGroup  
 name: myVM  
 vm\_size: Standard\_DS1\_v2  
 admin\_username: azureuser  
 ssh\_password\_enabled: false  
 ssh\_public\_keys:  
 - path: /home/azureuser/.ssh/authorized\_keys  
 key\_data: <your-key-data>  
 network\_interfaces: myNIC  
 image:  
 offer: CentOS  
 publisher: OpenLogic  
 sku: "7.5"  
 version: latest

## Next unit: Exercise - Run Ansible in Azure Cloud Shell

[Continue](https://docs.microsoft.com/en-us/learn/modules/implement-ansible/8-exercise-run-ansible-azure-cloud-shell/)

Need help? See our [troubleshooting guide](https://docs.microsoft.com/en-us/learn/support/troubleshooting?uid=learn.wwl.implement-ansible.examine-playbook-structure&documentId=30a0ada1-91ba-a531-69aa-29ce8f2abea7&versionIndependentDocumentId=2e12daa8-8ec4-b568-d8d9-7e80640830e0&contentPath=%2FMicrosoftDocs%2Flearn-pr%2Fblob%2Flive%2Flearn-pr%2Fwwl-azure%2Fimplement-ansible%2F7-examine-playbook-structure.yml&url=https%3A%2F%2Fdocs.microsoft.com%2Fen-us%2Flearn%2Fmodules%2Fimplement-ansible%2F7-examine-playbook-structure&author=lumac) or provide specific feedback by [reporting an issue](https://docs.microsoft.com/en-us/learn/support/troubleshooting?uid=learn.wwl.implement-ansible.examine-playbook-structure&documentId=30a0ada1-91ba-a531-69aa-29ce8f2abea7&versionIndependentDocumentId=2e12daa8-8ec4-b568-d8d9-7e80640830e0&contentPath=%2FMicrosoftDocs%2Flearn-pr%2Fblob%2Flive%2Flearn-pr%2Fwwl-azure%2Fimplement-ansible%2F7-examine-playbook-structure.yml&url=https%3A%2F%2Fdocs.microsoft.com%2Fen-us%2Flearn%2Fmodules%2Fimplement-ansible%2F7-examine-playbook-structure&author=lumac#report-feedback).