**Ansible Variables With Arrays**

Just like in programming languages where arrays are used to store a collection of items of the same data type, in Ansible, Arrays are used to define variables with multiple values.

Arrays are defined using the syntax shown.

vars:

arrayname:

- item1

- item2

- item3

- item4

Suppose you have a list of student names to be printed to stdout. Instead of defining them as individual variables, define an array with the student names as the values.

vars:

students:

- Alice

- Mark

- Peter

Here is the complete playbook file that prints out the student names contained in the array.

---

- hosts: all

vars:

students:

- Alice

- Mark

- Peter

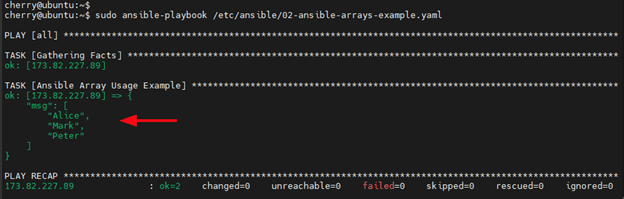
tasks:

- name: Ansible Array Usage Example

debug:

msg: "{{ students }}"

On runtime, the values are printed out as shown.



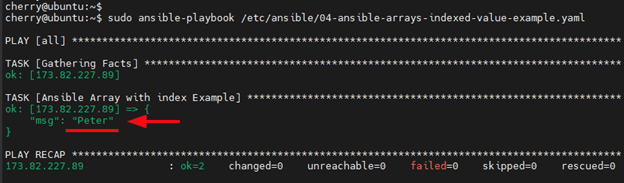
In addition, you can also access individual items from an array using index values ( starting from 0 ). When the playbook is modified as shown, it prints out the third value in the array which is "Peter".

tasks:

- name: Ansible Array Usage Example

debug:

msg: "{{ students[2] }}"



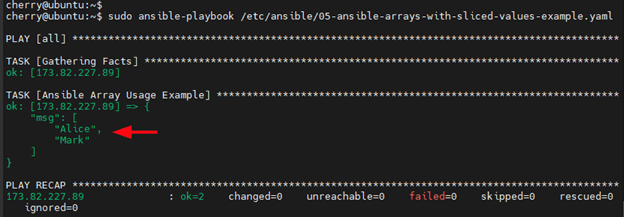
Just like in Python, you can also slice a range of elements in an array. The lower bound is inclusive, while the upper bound is exclusive. The playbook prints out the values "Alice" and "Mark" when you slice the array like this:

tasks:

- name: Ansible Array Usage Example

debug:

msg: "{{ students[0:2] }}"



[**#**](https://www.cherryservers.com/blog/how-to-use-variables-in-ansible-playbooks#ansible-variables-with-dictionaries)**Ansible Variables With Dictionaries**

A dictionary is an unordered collection of mutable items where each item is represented as a key-value pair. In a key-value pair, each key is mapped to its associated value, and a colon ( : ) is used to separate the key from its corresponding value.

Here is a dictionary’s syntax:

vars:

arrayname:

dictionary\_1:

key1: value1

key2: value2

dictionary\_2:

key1: value1

key2: value2

Let us add attributes to the values to thestudents array. The following playbook adds 3 key-value pairs to each of the dictionary elements ( Alice, Mark, and Peter ).

---

- hosts: all

vars:

students:

Alice:

gender: female

age: 21

city: Boston

Mark:

gender: male

age: 23

city: Dallas

Peter:

gender: male

age: 26

city: Miami

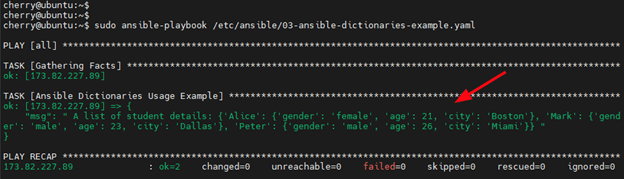
tasks:

- name: Ansible Dictionaries Usage Example

debug:

msg: " A list of student details: {{ students }}"

During playbook runtime, all the dictionaries and their values are printed out.



Just like arrays, you can also access individual elements in a dictionary variable. There are two ways of going about this. You can use either the **dot notation** or **bracket notation**.

The dot notation takes the format: variable.value

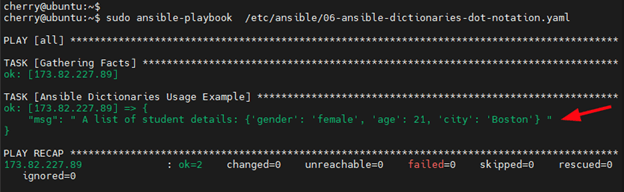
For example, to print out the student details for Alice edit the playbook as follows.

tasks:

- name: Ansible Dictionaries Usage Example

debug:

msg: " A list of student details: {{ students.Alice }} "



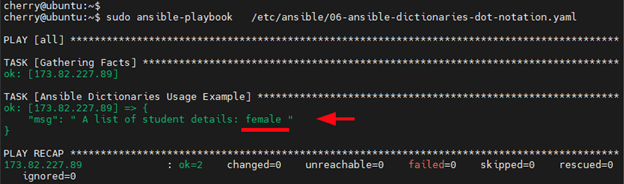
In addition, you can further narrow down and print out the value of a specific key. For example, you can print out the gender that Alice belongs to by referencing the variable as follows.

tasks:

- name: Ansible Dictionaries Usage Example

debug:

msg: " A list of student details: {{ students.Alice.gender }} "



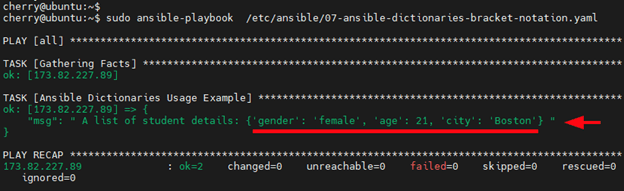
The bracket notation takes the following syntax: variable['value'].The following code snippet prints out the details of the student called Alice.

tasks:

- name: Ansible Dictionaries Usage Example

debug:

msg: " A list of student details: {{ students['Alice'] }} "



## Define Ansible Variables at Playbook Runtime

Variables can also be defined when executing a playbook by passing the variables on the command line using the --extra-varsor -e argument. The variable is enclosed in a single-quoted string inside a pair of single curly braces.

Variables passed during playbook runtime take the highest precedence and override the variables defined in the playbook. To better illustrate this, let us take a simple playbook that captures and prints out the value of a variable.

---

- hosts: all

vars:

car: "Corvette"

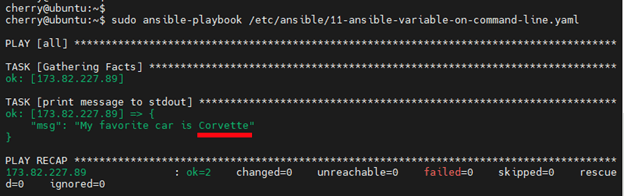
task:

- name: print message to stdout

debug:

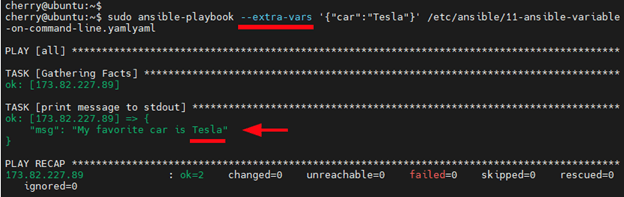
msg: "My favorite car is {{ car }}"

Upon runtime, the string My favorite car is and the value of the variable name Corvetteis printed.



To override the value of the variable defined in the playbook with a different value, for example, Tesla" invoke the --extra-vars argument followed by the key-value pair as shown.

$ sudo ansible-playbook --extra-vars '{"car":"Tesla"}' /etc/ansible/11-ansible-variable-on-command-line.yaml



This time around, the value Corvette is replaced by Tesla.