what's the purpose docker container module in ansible

In Ansible, the Docker container module allows you to manage Docker containers on a host. You can use this module to perform various actions on Docker containers, such as:

* Starting and stopping containers
* Installing and removing containers
* Checking the status of containers
* Pushing and pulling images from a registry

By using the Docker container module, you can automate the management of Docker containers in your infrastructure. For example, you can use Ansible to start and stop Docker containers on a schedule, or to ensure that certain containers are always running on specific hosts.

The Docker container module is just one of many built-in modules available in Ansible. You can use these modules to automate a wide range of tasks, including configuring and managing servers, deploying applications, and more.

can you write what can we do with it, that we couldn't do before

Using the Docker container module in Ansible allows you to automate the management of Docker containers in your infrastructure. With this module, you can perform actions on Docker containers that you might normally do manually, such as starting and stopping containers, installing and removing containers, and checking the status of containers.

Here are some specific examples of tasks you can automate with the Docker container module:

* Start and stop containers on a schedule, such as stopping all containers during off-hours to save resources.
* Ensure that certain containers are always running on specific hosts.
* Install and remove containers based on changes in your infrastructure, such as adding a new container when a new application is deployed.
* Check the status of containers to ensure that they are running as expected, and take corrective action if necessary.
* Push and pull images from a registry, to make it easy to deploy new containers or update existing ones.

Overall, the Docker container module in ansible provides a convenient and flexible way to automate the management of Docker containers in your infrastructure. By using this module, you can save time and effort, and ensure that your containers are running consistently and reliably.

Why is it important that we know the module and know how to use it? What need does it answer?

The Docker container module in ansible is an important tool for automating the management of Docker containers in your infrastructure. By using this module, you can streamline the process of managing containers, saving time and effort, and ensuring that your containers are running consistently and reliably.

There are several reasons why it is important to know how to use the Docker container module:

1. Efficiency: By automating the management of Docker containers, you can save time and effort, and focus on other tasks.
2. Reliability: Automating the management of containers can help ensure that they are always running as expected, which can improve the reliability of your infrastructure.
3. Scalability: As your infrastructure grows and changes, the Docker container module can help you scale up or down as needed, by installing or removing containers as needed.
4. Consistency: By automating the management of Docker containers, you can ensure that they are consistently configured and running across your infrastructure, which can help improve the overall stability and reliability of your systems.

Overall, knowing how to use the Docker container module in ansible is an important skill for anyone responsible for managing Docker containers in their infrastructure. By using this module, you can streamline the process of managing containers, and ensure that your infrastructure is running smoothly and reliably.

can you give more background information about the module

The Docker container module is a built-in module in ansible that allows you to manage Docker containers on a host. This module is implemented using the Docker API, and provides a convenient way to perform various actions on Docker containers from within ansible.

You can use the Docker container module to start and stop containers, install and remove containers, and check the status of containers. You can also use this module to push and pull images from a registry, which can be useful when deploying new containers or updating existing ones.

To use the Docker container module in ansible, you will need to have Docker installed on the host where the module is being run. You will also need to ensure that the host has access to the Docker API, which is usually exposed on a Unix socket or over a network interface.

In addition to the Docker container module, ansible also provides a number of other built-in modules for managing various aspects of your infrastructure. These modules can be used to automate a wide range of tasks, such as configuring and managing servers, deploying applications, and more.

write me some information about docker container module in ansible

In Ansible, the Docker container module allows you to manage Docker containers. It allows you to perform a variety of tasks related to Docker containers, including starting, stopping, and managing containers.

Some of the specific actions that you can perform with the Docker container module include:

* Starting a new container
* Stopping a running container
* Restarting a container
* Removing a container
* Inspecting a container

To use the Docker container module in Ansible, you will need to have the Docker Python module installed on the host machine. You will also need to have access to the Docker daemon, either by running the Docker daemon on the host machine or by connecting to a remote Docker daemon.

To start a new container using the Docker container module, you can use the **docker\_container** module and specify the **name** of the container, the **image** to use for the container, and any additional **command** or **arguments** to pass to the container when it is started.

Here is an example of how to start a new Docker container using the Docker container module in ansible:

Copy code

- name: Start a new Docker container docker\_container: name: my\_container image: nginx:latest command: nginx -g "daemon off;"

This example will start a new Docker container with the name "my\_container" using the latest version of the nginx image and running the command **nginx -g "daemon off;"**.

You can also use the Docker container module to stop, restart, or remove a container. For example, to stop a running container, you can use the **state** parameter and set it to **stopped**:

Copy code

- name: Stop a running Docker container docker\_container: name: my\_container state: stopped

This will stop the running container with the name "my\_container".

Overall, the Docker container module in ansible provides a convenient way to manage Docker containers as part of your infrastructure automation tasks.

can you give another 5 examples

Sure! Here are five more examples of using the Docker container module in Ansible:

1. Restart a running Docker container:

Copy code

- name: Restart a running Docker container docker\_container: name: my\_container state: restarted

This will restart the running container with the name "my\_container".

1. Remove a stopped Docker container:

Copy code

- name: Remove a stopped Docker container docker\_container: name: my\_container state: absent

This will remove the stopped container with the name "my\_container".

1. Inspect a Docker container:

Copy code

- name: Inspect a Docker container docker\_container\_info: name: my\_container register: container\_info

This will retrieve information about the container with the name "my\_container" and store it in the **container\_info** variable.

1. Start a Docker container with port mapping:

Copy code

- name: Start a Docker container with port mapping docker\_container: name: my\_container image: nginx:latest ports: - "80:80" - "443:443"

This will start a new Docker container with the name "my\_container" using the latest version of the nginx image and mapping ports 80 and 443 on the host machine to the same ports on the container.

1. Start a Docker container with volume mapping:

Copy code

- name: Start a Docker container with volume mapping docker\_container: name: my\_container image: nginx:latest volumes: - "/var/www/html:/usr/share/nginx/html"

This will start a new Docker container with the name "my\_container" using the latest version of the nginx image and mapping the host machine's **/var/www/html** directory to the **/usr/share/nginx/html** directory on the container.