



# Publish Ansible Roles on GitHub & Ansible Galaxy

By: Er. Vikas Nehra (M. Tech, B. Tech), Experience: 15 + Years

## Session - 23 Agenda:

1. How To Publish Ansible Roles on GitHub & Ansible Galaxy

### Ansible Galaxy:

Ansible Galaxy is Ansible's official hub for sharing Ansible content. You can jump-start your automation project by using Galaxy with great content from the Ansible community. Galaxy provides pre-packaged units of work known to Ansible as Roles.

### Ansible Roles:

Role enables the sharing and reuse of Ansible tasks. It contains Ansible playbook tasks plus all the supporting files, variables, templates, and handlers needed to run the tasks. A Role is a complete unit of automation that can be reused and shared.

### Importance of Ansible Roles:

Roles play an important part in breaking the complex Ansible Playbooks, the core component of any Ansible configuration, into easy to reuse multiple files. With Ansible Roles you can:

- Bundle automation content and make it reusable
- Group multiple sets of tasks in an effective manner
- Modify and reduce syntax errors
- Release modules independent of Ansible Core releases
- Ansible Roles are equivalent to modules in Puppet and Cookbooks in Chef.

### Publishing an Ansible Role

Go to <https://galaxy.ansible.com>

Log in to Galaxy using GitHub credentials.

Go to "My Content" tab. Click on "add content" option. There are two ways of publishing roles on ansible galaxy:

- (a) directly using browse option, you can upload ansible roles zip file in tar.gz format.
- (b) using Github. You can fetch your code repository from Github.

### Publishing Ansible Roles on GitHub:

Execute the command in the same order for your roles which you want to publish on GitHub.

Let's suppose if we want to publish the Apache roles on GitHub which we created in the previous session for the Apache web server configuration on the managed nodes.

First copy the apache.yml playbook to the ansible roles apache directory so that you can upload the playbook alongwith the roles as well.

```
$ sudo cp -r /home/vikasnehra/apache.yml /etc/ansible/roles/apache/
```

Now, go to the apache roles directory where you created the roles directory for apache.

```
$ cd /etc/ansible/roles/apache/
```

Install the git packages if not installed earlier on your ansible server.

```
$ sudo dnf install -y git
```

Verify the installation of git packages and installed git package version.



# Publish Ansible Roles on GitHub & Ansible Galaxy

By: Er. Vikas Nehra (M. Tech, B. Tech), Experience: 15 + Years

\$ git --version

You can list the available ansible roles in your ansible server using below command:

\$ ansible-galaxy list

Verify the location where you are working, its very important before you execute git command to initialize the roles.

\$ pwd

Initialize the files and directories of apache roles so that they can be published on the GitHub. Git init can be used to convert an existing, un-versioned project to a Git repository or initialize a new, empty repository.

\$ sudo git init

If you will execute the ls -lh command it will list a hidden directory (.git) under apache, which stores the metadata and object database for your project.

\$ ls -lha

Now login into your GitHub account where you want to publish the ansible roles. Create a new repository there with any name (say apache) on your GitHub account. Then copy its URL

Now, you can define the location where your code will be published on GitHub using the URL of your repository that have you copied earlier.

\$ sudo git remote add origin <https://github.com/ansible22/apache.git>

You can check the status of the files whether are they ready to be published or not using below command. In Git, the status command shows the state of the working directory, staging area, and which files are untracked (names shown in red and not staged) or tracked (shown in green and staged). The tracked files, shown in green, are found in the staging area and will be saved with the next commit.

\$ sudo git status

You can make files ready for the upload using below command. The git add command adds a change in the working directory to the staging area. It tells Git that you want to include updates to a particular file in the next commit. However, git add doesn't really affect the repository in any significant way—changes are not actually recorded until you run git commit.

\$ sudo git add .

Check the status of the files again, this time all of the files will be highlighted in the green color.

\$ sudo git status

Set a Git username which will be used to publish the code on the above-mentioned repository URL.

\$ sudo git config user.name ansible22

Set a Git user email address which will be used to publish the code on the above-mentioned repository URL.

\$ sudo git config user.email [redhatansible22@gmail.com](mailto:redhatansible22@gmail.com)

You can execute git commit command for the apache to create a snapshot of the staged changes.



# Publish Ansible Roles on GitHub & Ansible Galaxy

By: Er. Vikas Nehra (M. Tech, B. Tech), Experience: 15 + Years

The git commit command is one of the core primary functions of Git. Prior use of the git add command is required to select the changes that will be staged for the next commit. Then git commit is used to create a snapshot of the staged changes along a timeline of a Git project's history.

```
$ sudo git commit -m 'apache'
```

Now, the command `git push origin master` can be used to push any commits made locally on the `master` branch to a remote repository on `origin`. Authentication may be required if this is the first time pushing to that location.

```
$ sudo git push origin master
```

It will ask for the password and show the error because GitHub doesn't support the password authentication to push the code to the repository.

You can generate the authentication token from developer's settings in your GitHub profile. Now you can again execute the above command and provide the username and authentication token when prompted.

username:

**ansible22**

token:

**ghp\_wHjbUIQZ5FtTVwA42yAiRn8FFjGaHm28tLE7**

Your code (Ansible Role) has been published successfully at GitHub. Now you can fetch it from there to Ansible Galaxy using below steps.

1. Go to "My Content" tab. Click on "add content" of namespace where the role is pushed on GitHub.
2. Select the role repository to add to.
3. After adding the repository, you can find it in the tab "My Imports".
4. Click on the import icon, in the top right corner, to import the latest release in Ansible Galaxy.

## Best Practices to be followed while developing Ansible Roles:

- Name of git repo should be same as Role name.
- Variables names in the Roles should be unique, should not override other Role/playbook variables. To achieve this, use Role prefix for every variable in the Role.
- List the other Role dependencies in the meta/main.yml
- Search for Ansible Galaxy role if you can reuse existing Role for the requirement.
- Use Git tags as Role versions
- As Role is a reusable component, it should not contain any inventories or playbooks.
- Add test automation in the test folder. Integration with Travis CI is a simple and easy way to write CI pipeline for Ansible Roles.
- Role variables defined in 'vars' have a very high precedence - they can only be overwritten by passing them on the command line in the specific task or in a block. Therefore, all your variables should be defined in the 'defaults' unless those are constants.