Red Hat
Ansible
Automation

Ansible Best Practices, Part 2

Develop Automation in Teams

Red Hat

Red Hat
Ansible
Automation

IBM Hackathon, Part 2

Develop Automation in Teams

Red Hat


CODE MUST BE ORGANIZED
USE GIT!

Invented 2005 by Linus Torvalds for kernel.org

- Distributed Version Control
- Easy to use on all platforms
- Snapshot based
- Support for many branches
- Handle large projects

Supports different Server Backends

- Github.com
- Gitlab.com
- Self hosted (Gitea, Googs, Gitlab)

Reads:
https://opensource.com/downloads/cheat-sheet-git

Pro Git: https://git-scm.com/book/en/v2

Reference: https://git-scm.com/docs

Github Docs: https://guides.github.com/

Gitlab Basics: https://docs.gitlab.com/ee/gitlab-basics/

Git Local

```
yum/dnf install git
or (Debian/Ubuntu)
apt install git
```

```
cd folder
git init
git add -A
git commit -m "Initial commit"
```

Reads:

https://opensource.com/downloads/cheat-sheet-git

Pro Git: https://git-scm.com/book/en/v2

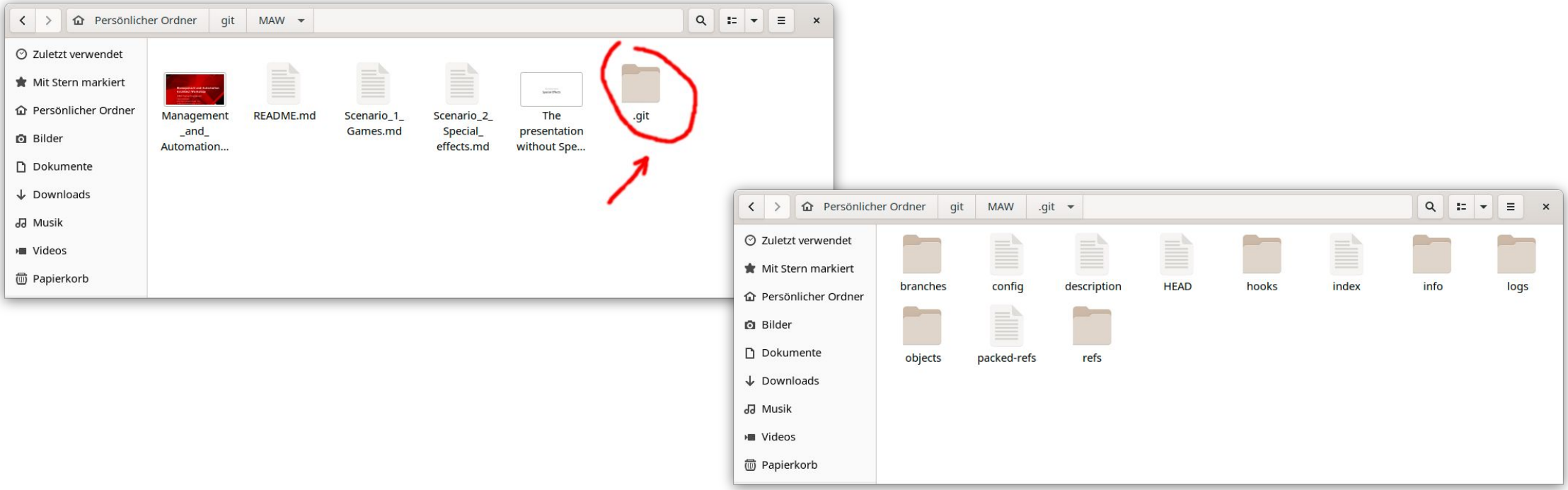
Reference: https://git-scm.com/docs

Github Docs: https://guides.github.com/

Gitlab Basics: https://docs.gitlab.com/ee/gitlab-basics/

Git is a content-addressable filesystem.

At the core of Git is a simple key-value data store. You can insert any kind of content into a Git repository, for which Git will hand you back a unique key you can use later to retrieve that content.



Red Hat

Git Remote

```
git remote add origin git@<SERVER URL>:<USER>/<repo>.git
git push -u origin master
```

After Initial Push

On Change:

`git add <changed file>` or **`git add -A`**

Commit Change:

`git commit -m “Comment for this Commit”`

And send to remote

`git push`

Reads:

https://opensource.com/downloads/cheat-sheet-git

Pro Git: https://git-scm.com/book/en/v2

Reference: https://git-scm.com/docs

Github Docs: https://guides.github.com/

Gitlab Basics: https://docs.gitlab.com/ee/gitlab-basics/

1 commit 1 file changed

Commits on Oct 27, 2014

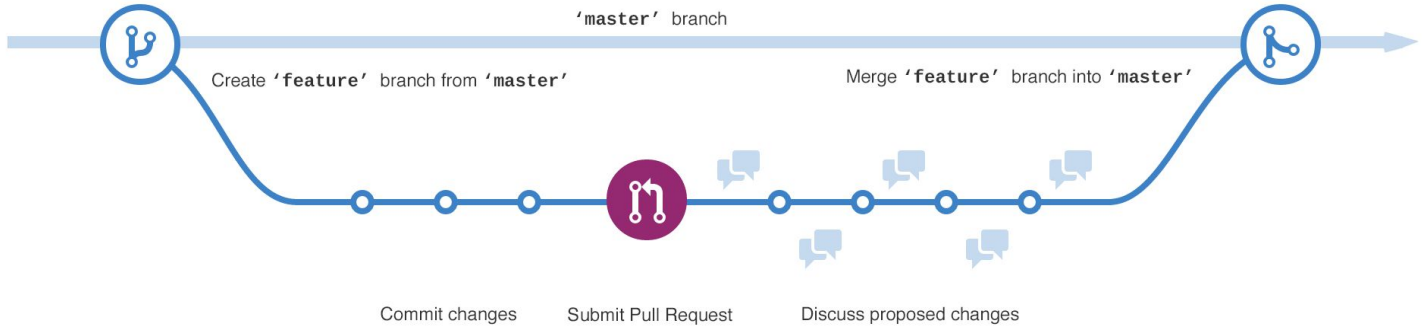
hubot Finish README ···

Showing 1 changed file with 1 addition and 1 deletion.

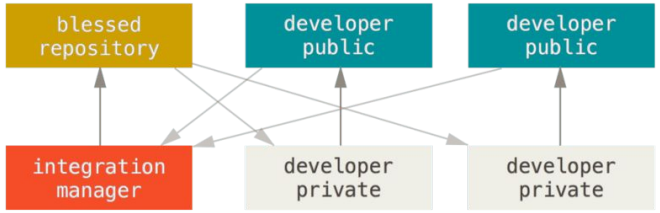
2 README.md

```
...  ...  @@ -1,4 +1,4 @@
1    1     hello-world
2    2     ===========
3    3
4         -Just another repository
     4    +Hubot here, I like Node.js and Coffee
          them far superior to Earth tacos.
```

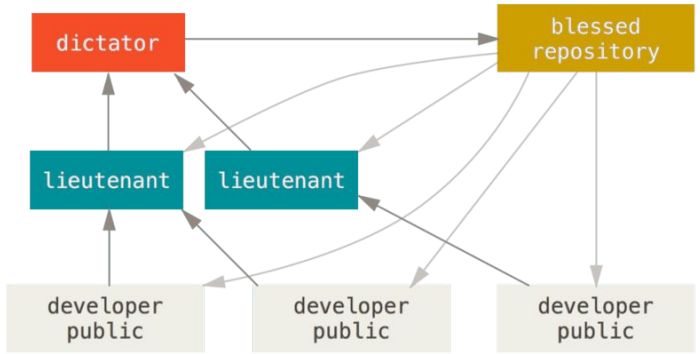
Snapshots show changes between commits

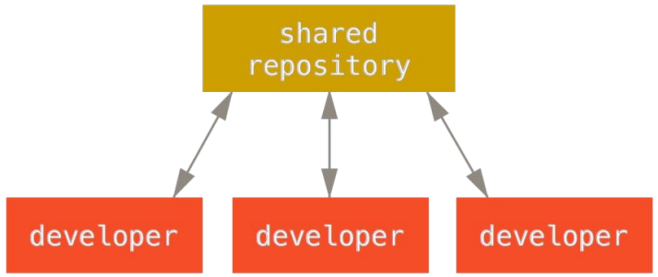


Branches allow to split development trees from the main code repository and merge changes back



Hierarchical Workflows for distributed development in Git





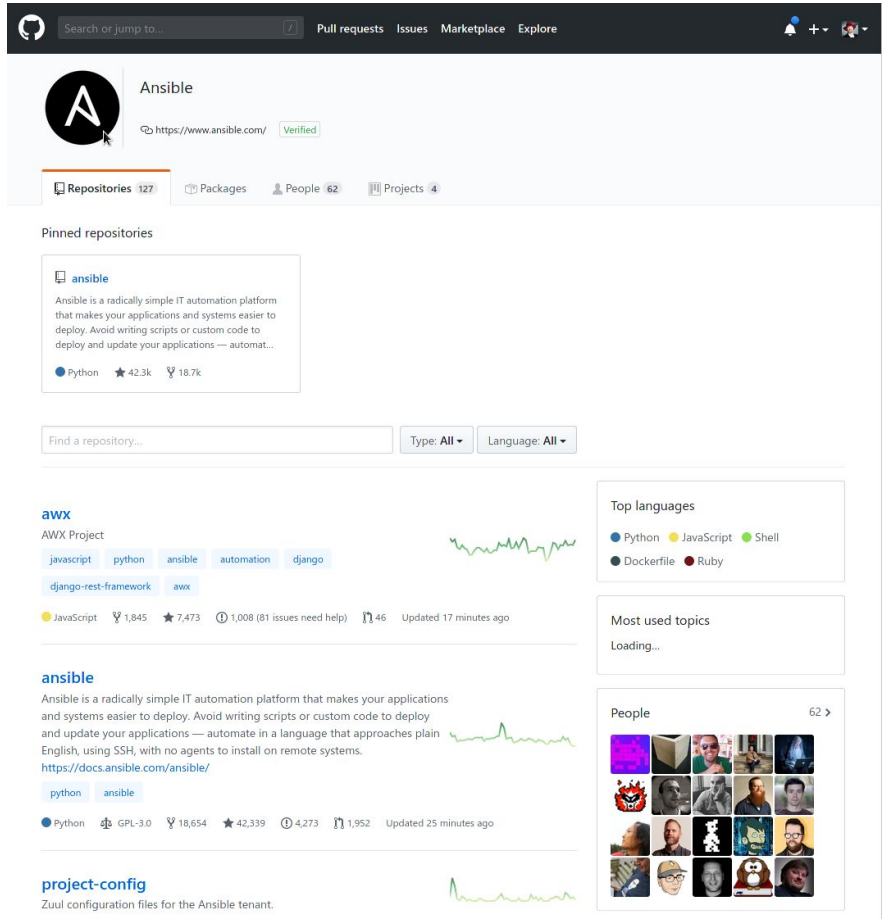
Source:
https://guides.github.com/activities/hello-world/
https://git-scm.com/book/en/v2

Red Hat

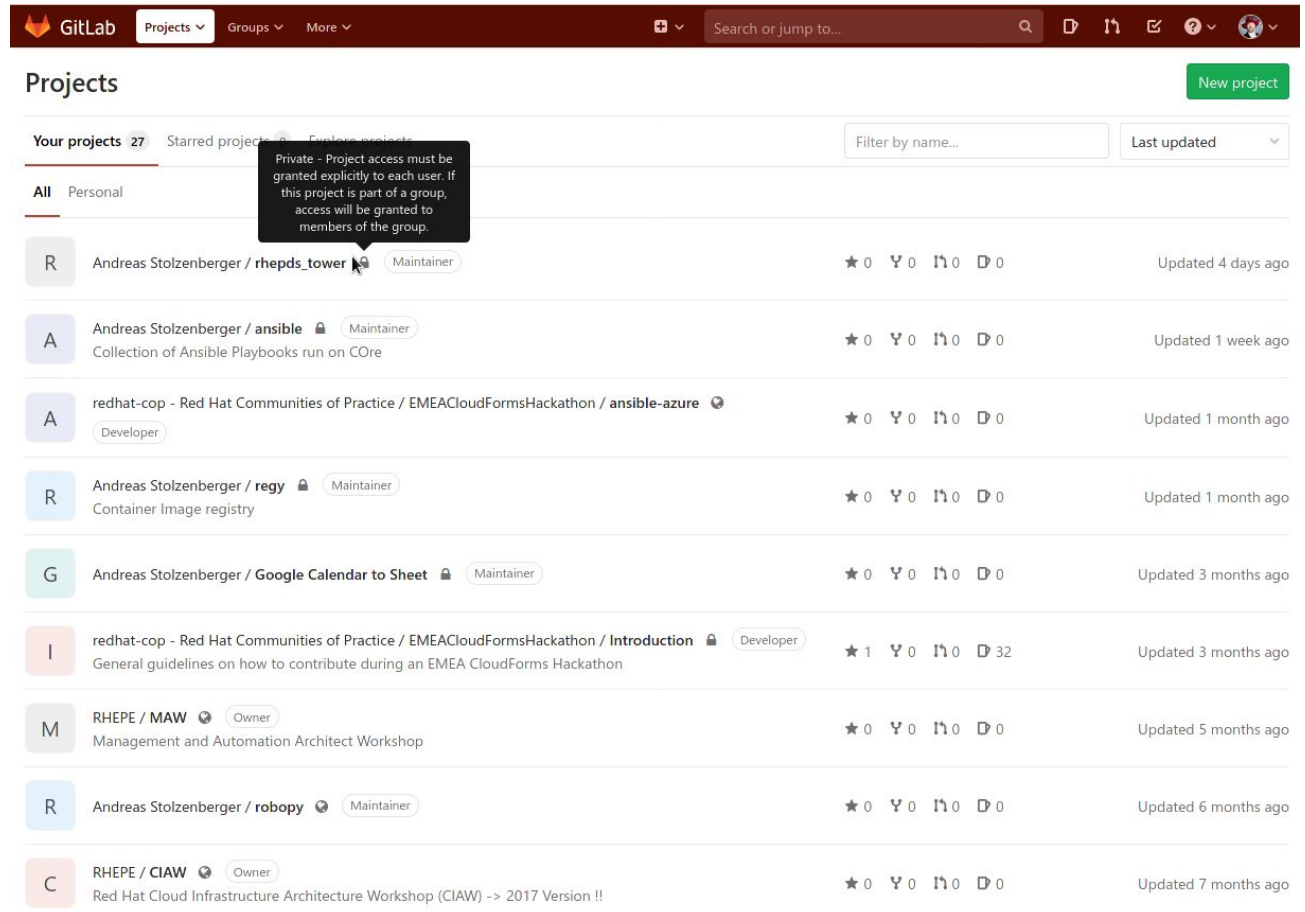
1. git commit

2. git push

3. leave building



Github.com: Most popular for Open Source Projects, owned by Microsoft



Gitlab.com: Less popular, more features in free tier, Code available for on-prem Installation.

Red Hat

Gogs

- Written in Go
- Simple, easy and fast
- Runs Linux, Linux ARM, Windows

Gitea

- Forked from Googs
- Written in Go
- Simple, easy and fast
- Runs Linux, Linux ARM, Windows
- Better Maintained

Bonobo

- Written in .NET
- Integrates in IIS
- Windows only

Local Datacenter
Staging Environments
Airgap (Export / Import)

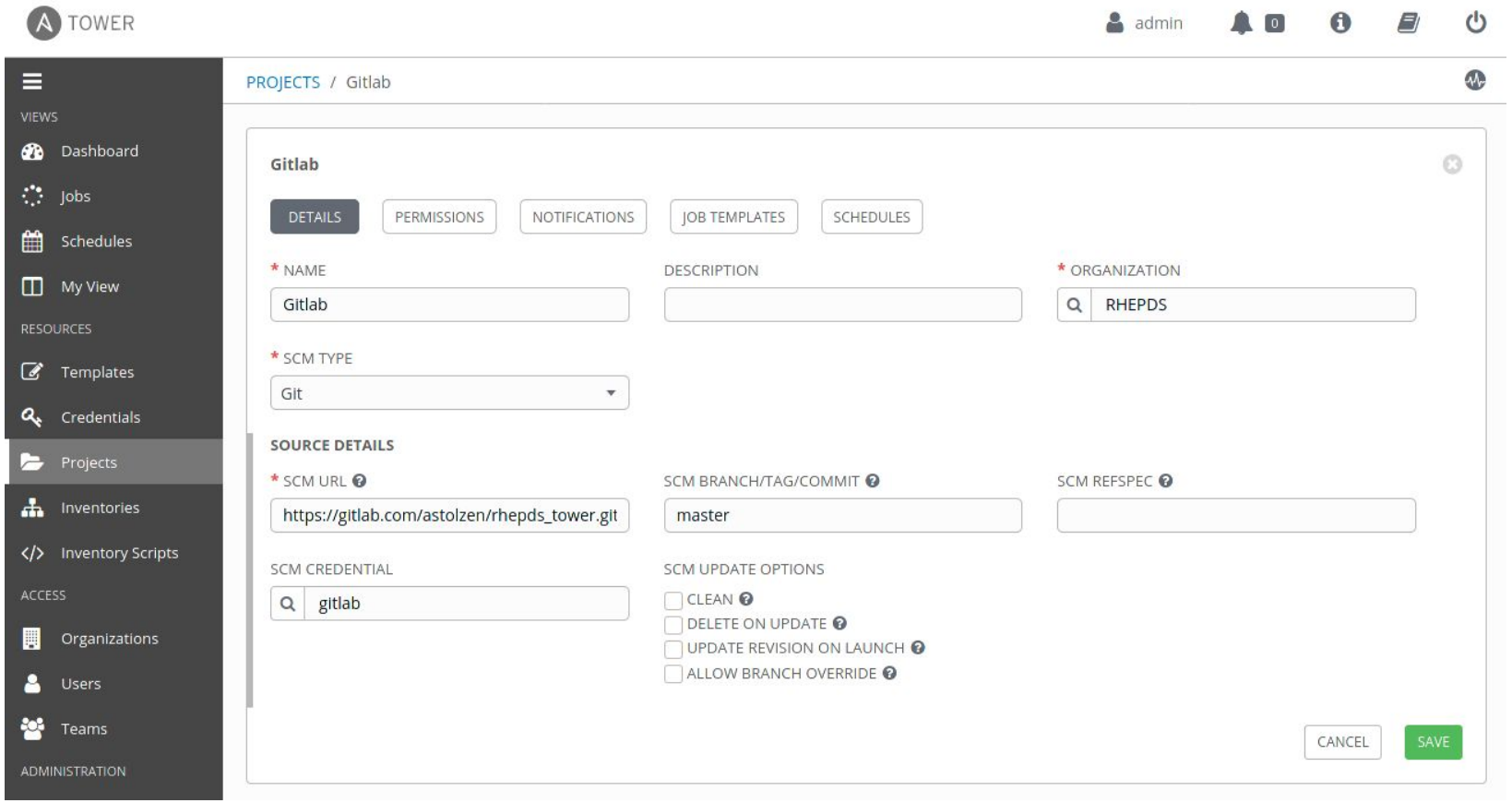
Start with one Git repository - but when it grows, use multiple!

At the beginning: put everything in one Git repository

In the long term:

- One Git repository per role
- Dedicated repositories for completely separated teams / tasks
- Integration via roles/requirements.yml

Tower requires SVNs for Playbooks



- Every Project is based on one Repository
- Templates are based on Playbooks from Projects

Red Hat

Tower can import a repository multiple times with different branches

- Use feature or staging branches in your Git
- Import them all separately, address them separately
- Useful for testing of new features but also to move changes through stages

Roles and Requirements

requirements.yml

```
# from GitHub
- src: https://github.com/bennojoy/nginx

# from GitHub, overriding the name and specifying a specific tag
- name: nginx_role
  src: https://github.com/bennojoy/nginx
  version: master

# from GitLab or other git-based scm, using git+ssh
- src: git@gitlab.company.com:mygroup/ansible-base.git
    scm: git
```

- Specific Roles as a part of the Main Repository
- Generic Roles in separate Repositories
- Import via roles/requirements.yml
- Project refresh will update dependencies

Tower automatically imports Roles during Project update

- Do not copy roles into your playbook repository, just create a roles/requirements.yml
- Tower will automatically import the roles during Project installation
- Mix roles from various sources
- Fix version in roles/requirements.yml to have auditable environment!

Collections from AutomationHub and Galaxy

requirements.yml

```
collections:
  - name: junipernetworks.junos
    source: https://galaxy.ansible.com

  - name: f5networks.f5_modules
    source: https://cloud.redhat.com/api/automation-hub/
```

- Modules, Plug-Ins and Roles
- Import via collections/requirements.yml
- Project refresh will update dependencies
- Works together with other Role Sources

Mixed Dependencies and ressources

- Use AutomationHub for secure and supported Collections
- Publish your generic Roles on Galaxy
- Open Source your free Playbooks on Github/Gitlab
- Keep confidential content and variable declarations in local Repositories

Thank you

Red Hat is the world's leading provider of enterprise open source software solutions. Award-winning support, training, and consulting services make Red Hat a trusted adviser to the Fortune 500.

linkedin.com/company/red-hat

youtube.com/user/RedHatVideos

facebook.com/redhatinc

twitter.com/RedHat

Red Hat