

0 chezmoi 优雅的文档设计

Quick start

Concepts

Roughly speaking, chezmoi stores the desired state of your dotfiles in the directory `~/.local/share/chezmoi`. When you run `chezmoi apply`, chezmoi calculates the desired contents for each of your dotfiles and then makes the minimum changes required to make your dotfiles match your desired state. chezmoi's concepts are [described more accurately in the reference manual](#).

Start using chezmoi on your current machine

Assuming that you have already [installed chezmoi](#), initialize chezmoi with:

```
$ chezmoi init
```

SHELL

This will create a new git local repository in `~/.local/share/chezmoi` where chezmoi will store its source state. By default, chezmoi only modifies files in the working copy.

Manage your first file with chezmoi:

```
$ chezmoi add ~/.bashrc
```

SHELL



This will copy `~/.bashrc` to `~/.local/share/chezmoi/dot_bashrc`.

Edit the source state:

```
$ chezmoi edit ~/.bashrc
```

SHELL

This will open `~/.local/share/chezmoi/dot_bashrc` in your `$EDITOR`. Make some changes and save the file.

 **Hint** 

You don't have to use `chezmoi edit` to edit your dotfiles. See [this FAQ entry](#) for more details.

See what changes chezmoi would make:

```
$ chezmoi diff
```

SHELL

Apply the changes:

```
$ chezmoi -v apply
```

SHELL

All chezmoi commands accept the `-v` (verbose) flag to print out exactly what changes they will make to the file system, and the `-n` (dry run) flag to not make any actual changes. The combination `-n -v` is very useful if you want to see exactly what changes would be made.

Next, open a shell in the source directory, to commit your changes:

```
$ chezmoi cd
$ git add .
$ git commit -m "Initial commit"
```

SHELL

[Create a new repository on GitHub](#) called `dotfiles` and then push your repo:

```
$ git remote add origin https://github.com/$GITHUB_USERNAME/dotfiles.git
$ git branch -M main
$ git push -u origin main
```

SHELL

Hint

chezmoi can be configured to automatically add, commit, and push changes to your repo.

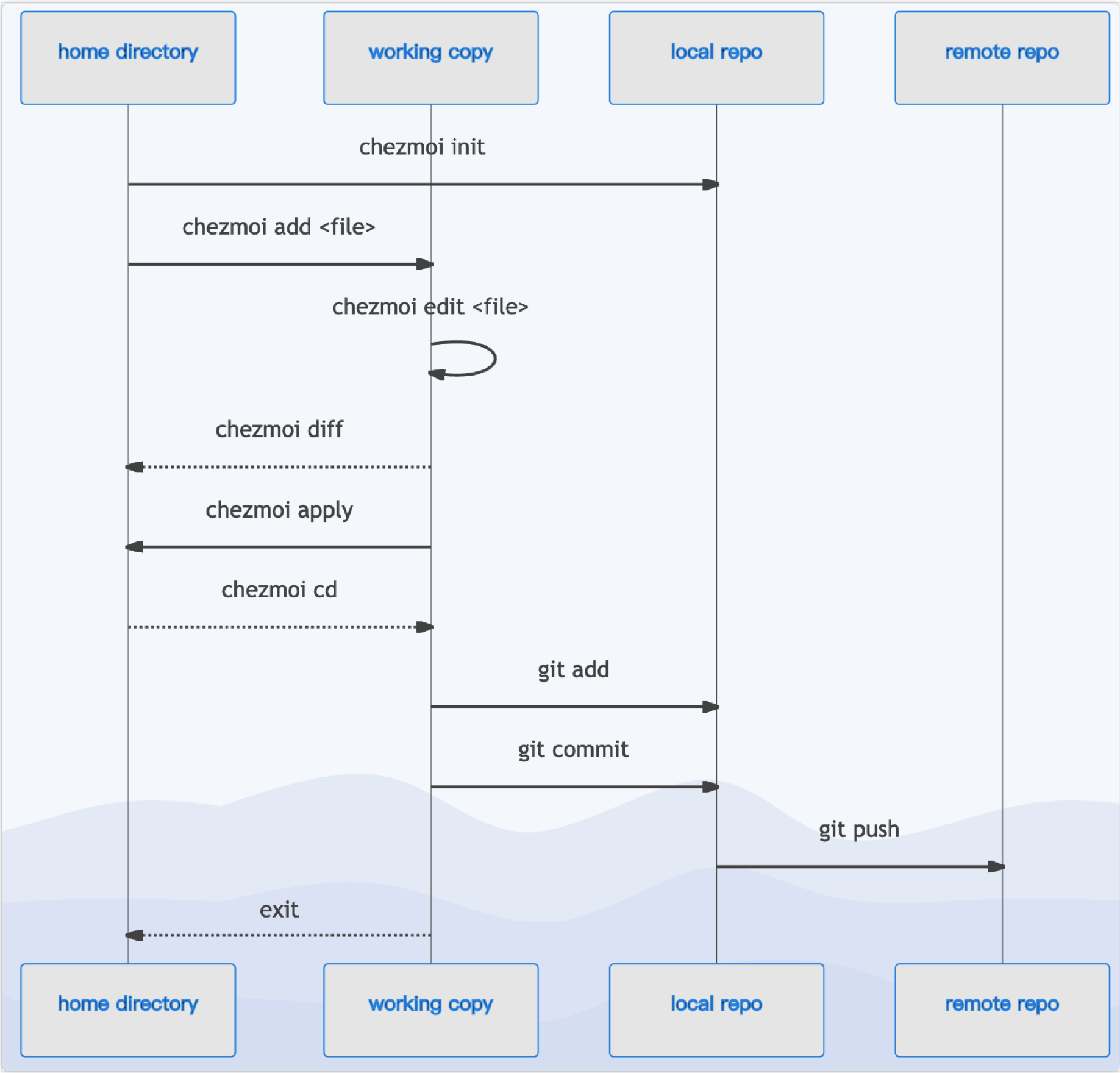
chezmoi can also be used with [GitLab](#), or [BitBucket](#), [Source Hut](#), or any other git hosting service.

Finally, exit the shell in the source directory to return to where you were:

```
$ exit
```

SHELL

These commands are summarized in this sequence diagram:



Using chezmoi across multiple machines

On a second machine, initialize chezmoi with your dotfiles repo:

```
$ chezmoi init https://github.com/$GITHUB_USERNAME/dotfiles.git
```

SHELL

This will check out the repo and any submodules and optionally create a chezmoi config file for you.

Check what changes that chezmoi will make to your home directory by running:

```
$ chezmoi diff
```

SHELL

If you are happy with the changes that chezmoi will make then run:

```
$ chezmoi apply -v
```

SHELL

If you are not happy with the changes to a file then either edit it with:

```
$ chezmoi edit $FILE
```

SHELL

Or, invoke a merge tool (by default `vimdiff`) to merge changes between the current contents of the file, the file in your working copy, and the computed contents of the file:

```
$ chezmoi merge $FILE
```

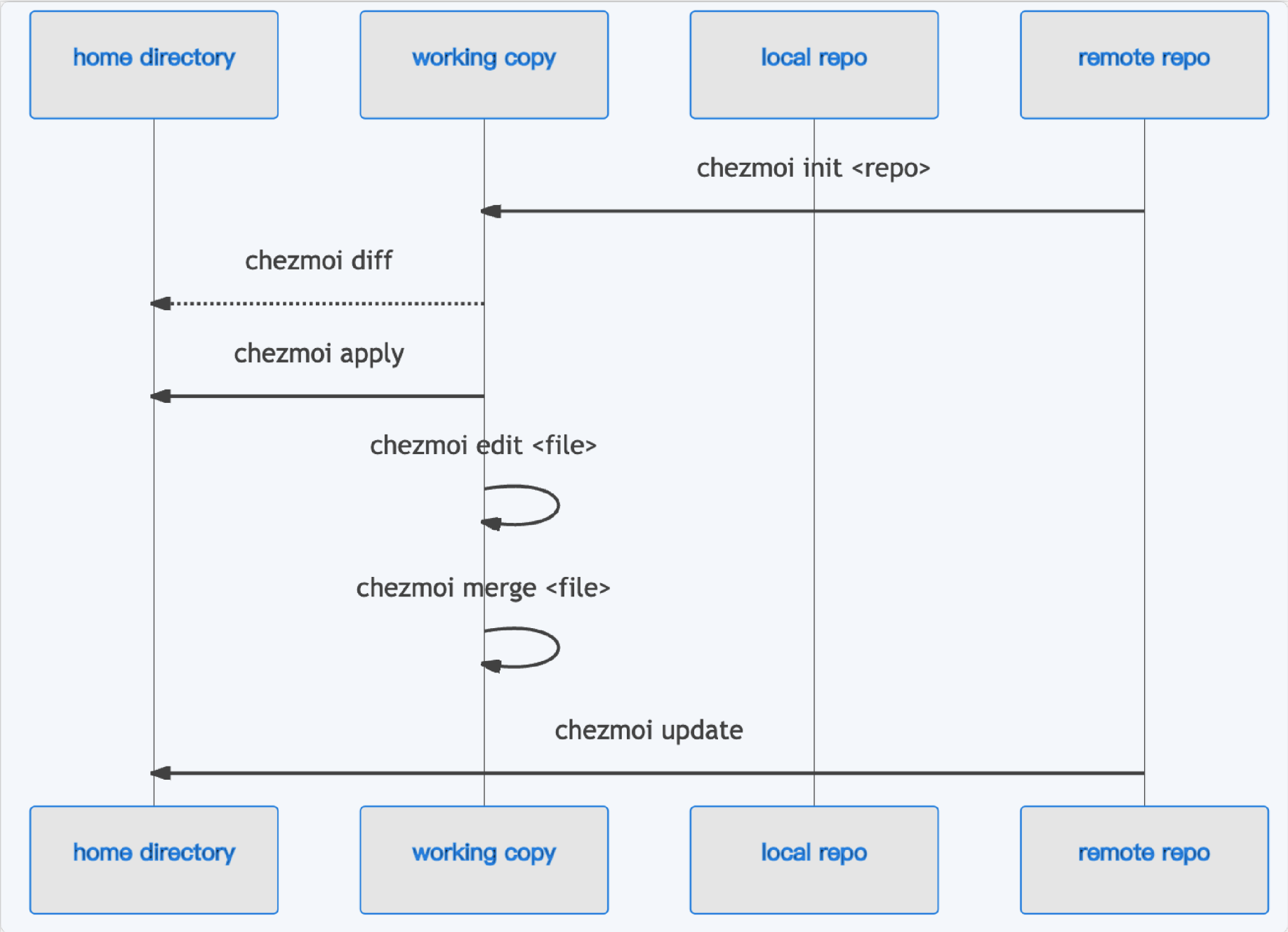
SHELL

On any machine, you can pull and apply the latest changes from your repo with:

```
$ chezmoi update -v
```

SHELL

These commands are summarized in this sequence diagram:



Set up a new machine with a single command

You can install your dotfiles on new machine with a single command:

```
$ chezmoi init --apply https://github.com/$GITHUB_USERNAME/dotfiles.git
```

SHELL

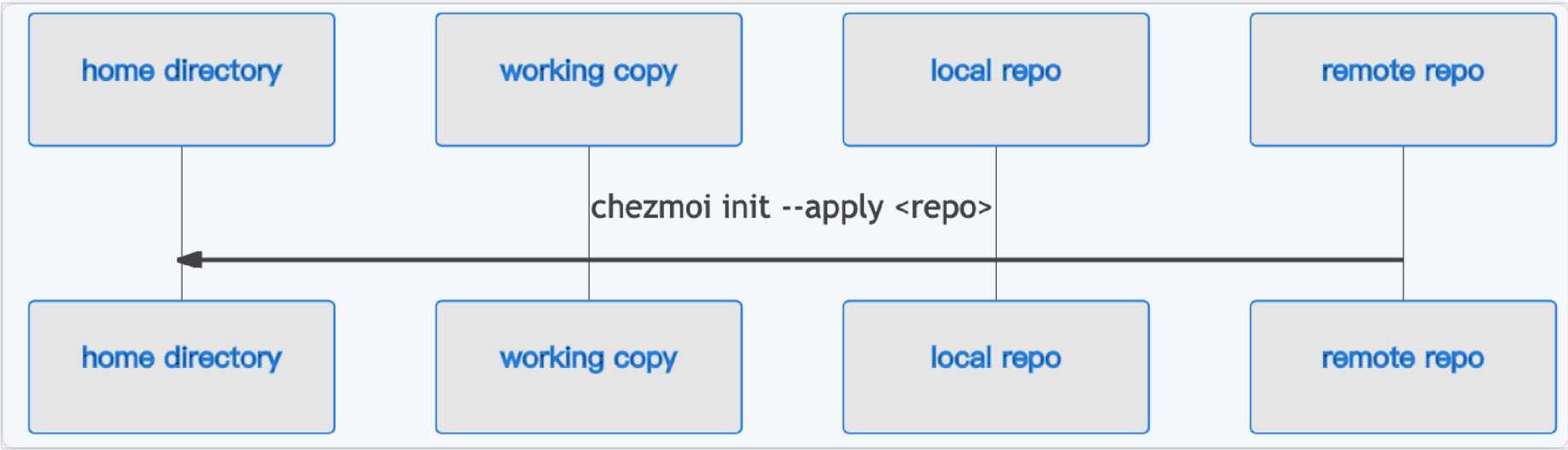
If you use GitHub and your dotfiles repo is called `dotfiles` then this can be shortened to:

```
$ chezmoi init --apply https://github.com/$GITHUB_USERNAME/dotfiles
```

```
$ chezmoi init --apply $GITHUB_USERNAME
```

SHELL

This command is summarized in this sequence diagram:



Next steps

For a full list of commands run:

```
$ chezmoi help
```

SHELL

chezmoi has much more functionality. Good starting points are reading [articles about chezmoi](#) adding more dotfiles, and using templates to manage files that vary from machine to machine and retrieve secrets from your password manager. Read the [user guide](#) to explore.