ConfigShell

Christian ENGEL (2023-01 V0.8.0)

ConfigShell

ConfiShell is an open-source (MIT-based) project to support Linux/UNIX® interactive working environments. This work is not paid by an employer. It offers:

- Shell interactive support for bash, fish, zsh (partly)
- Linux support for proxied environments
- golang development support
- TLS & SSH key and certificate creation and analysis
- bash scripting, markdown & LaTeX support
- git encryption and general git support
- k8s support

Features

- can be installed (git clone) without password
- script included to automatically upgrade when logging in
- continuously be updated
- experience from 40 years of UNIX experience

The Installation

1st Steps - put your shoes on…

1. Use the default path if possible

   ```
   mkdir /opt/ConfigShell
   ```

2. Install ConfigShell

   ```
   git clone https://github.com/engelch/ConfigShell /opt/ConfigShell
   ```

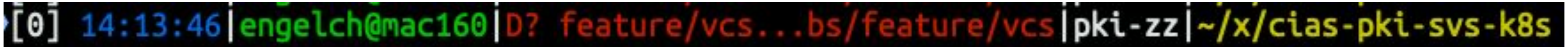
3. Activate it for you (home directory of the current user)

   ```
   /opt/ConfigShell/installDotFiles2home
   ```

4. Restart your shell

You are using ConfigShell, if

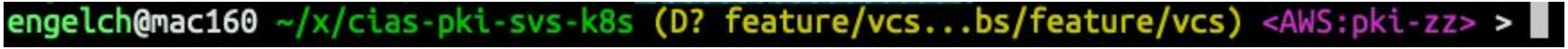
with bash, your prompt looks like # EXTRA by ConfigShell



```
[0] 14:13:46|engelch@mac160|D? feature/vcs...bs/feature/vcs|pki-zz|~/x/cias-pki-svs-k8s
```

exit-code prompt-creation user@host git-repo aws-profile CWD

Here, a possible fish prompt # EXTRA by ConfigShell



```
engelch@mac160 ~/x/cias-pki-svs-k8s (D? feature/vcs...bs/feature/vcs) <AWS:pki-zz> >
```

user@host   PWD/CWD   if in a git-repo   if AWS_PROFILE

Shell Enhancement

About

- abbreviations for often used programmes
- let's live the AKÜFI

About

- abbreviations for often used programmes
- let's live the AKÜFI (Abkürzfimmel /abbreviation madness)
- standard on all systems (@work, @home, @IoT, @srv)

bash & fish Commands 1 - Directories # EXTRA by ConfigShell

`ls`	multiple columns, colourised, showing file type
`la`	like ls above with -a option (showing dot files)
`ll`	like ls above with -l option (long listing, one file per line)
`lla`	like ls above with -la options
`lld`	like ls above with -d (show current and parent directory file)
`llad`	like ls above with -adl
`cd..`	same as cd .. (typoo saver)
`.2 … .7`	cd ../.. … cd ../../../../../../..
`mkcd <dir>`	create directory and make it the CWD (current working directory)
`brmd`	go to the parent directory and try to delete the child directory (must be mt)

BASH & FISH COMMANDS 1 - DIRECTORIES

EXTRA by ConfigShell

`ls`	multiple columns, colourised, showing file type
`la`	like ls above with -a option (showing dot files)
`ll`	like ls above with -l option (long listing, one file pe
`lla`	like ls above with -la options
`lld`	like ls above with -d (show current and parent direct
`llad`	like ls above with -adl
`cd..`	same as cd .. (typoo saver)
`.2 … .7`	cd ../.. … cd ../../../../../../..
`mkcd <dir>`	create directory and make it the CWD (current working directory)
`brmd`	go to the parent directory and try to delete the child directory (must be mt)

cd -

change to previous CWD

BASH & FISH COMMANDS 2 # EXTRA by ConfigShell

`cp / rm / mv`	all mapped to include the -i option (confirm deletion, protect overwrite)
`l`	less (Δ to some Ubuntu default installations)
`j`	jobs
`wh`	which
`rm~  rmtex`	delete backup files or TeX intermediate files
`a`	alias (show aliases - depending on the shell there might be more)
`pu`	pushd .
`po`	popd

BASH LOADING FILES # EXTRA by ConfigShell

```
bash uses ~/.bashrc.d

    *.rc   files are sourced into the current shell

    *.sh   files are executed (sub-shell)
```

When?

1. Done when starting the shell
2. Re-Done by executing rl or rlFull (bash)

Bash Loading Files - Example # EXTRA by ConfigShell

bash uses ~/.bashrc.d

```
    *.rc    files are source
    *.sh    files are execute
```

~/.bashrc.d/aws.rc

```
export AWS_PROFILE=pki-zz
```

When?

1. Done when starting the shell
2. Re-Done by executing rl or rlFull (bash)

FISH LOADING FILES

fish uses the default directory ~/.config/fish

Most files here are s-linked to /opt/ConfigShell, but:

fish sources 2 files at start/end of loading if existing:

```
    ~/.config/fish/pre.fish            # EXTRA by ConfigShell
    ˜/.config/fish/conf.d/*.fish
    ~/.config/fish/post.fish           # EXTRA by ConfigShell
```

EXTRA: ~/.config.fish/conf.d/*.sh are executed with bash

FISH VS BASH VS ZSH

comparison article on medium.com

zellwk.com

- completion in bash can be a pain and was better before (good old tcsh times,...)
- zsh scripts and add-ones (oh-my-zsh) can become complex
- +good defaults: fish
  -missing in fish: !! !$

FISH VS BASH VS ZSH

comparison article on medium.com

zellwk.com

oh-my-fish
can fix it

https://github.com/oh-my-fish/oh-my-fish

- completion in bash can be a pain and was better before (good old tcsh times,...)

- zsh scripts and add-ones (oh-my-zsh) can become complex

- +good defaults: fish
  -missing in fish: !! !$

TLS & SSH Enhancement

GIT & GEE Enhancement

Kubernetes (k8s) Enhancement

Development Enhancement

Script Development

1. Usually, stay with bash as this is the most common shell on all systems these days.

2. Start a script with shebang
   `#!/usr/bin/env bash`

3. Use shellcheck to check your shell

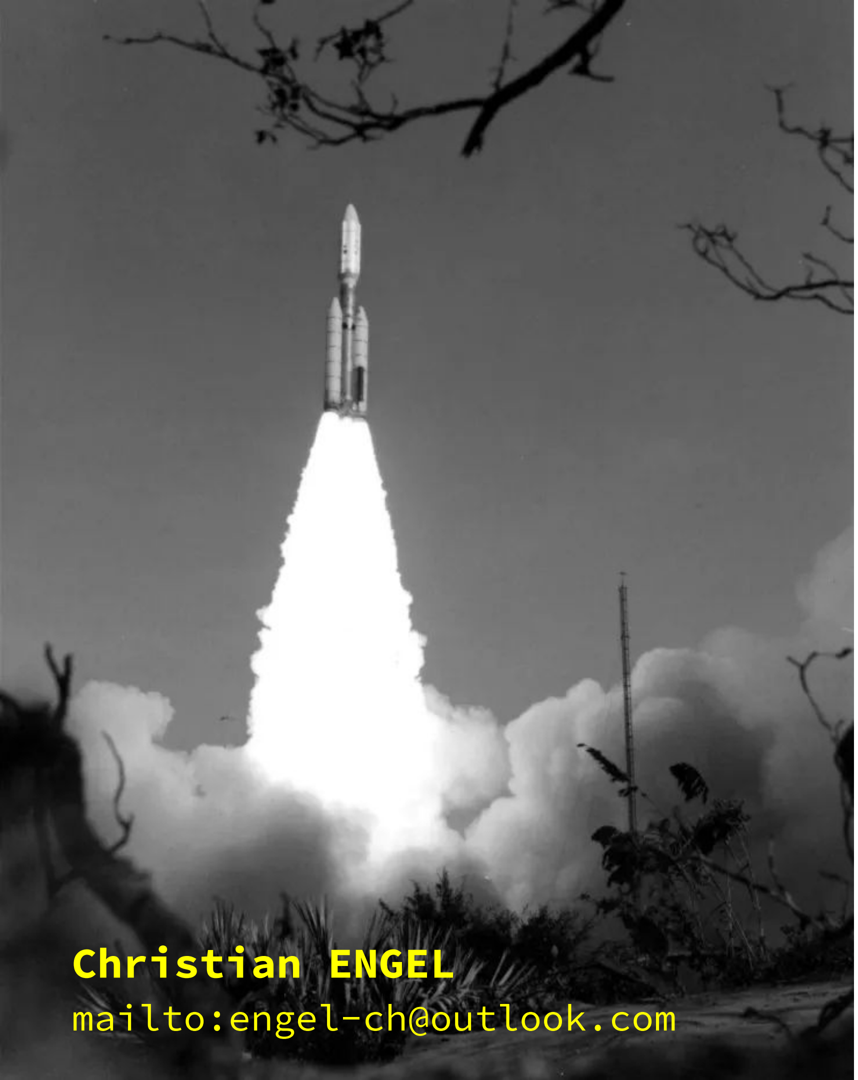
4. Easy start: copy
   `/opt/ConfigShell/Template/bash.mini.skeleton.sh`

OS-Specific Enhancement

–

SubModules

JOIN THE TEAM



Christian ENGEL

mailto:engel-ch@outlook.com

The Future of ConfigShell

Help us

Configshell is a work of enthusiastic persons, please join

Propose Ideas

Tell us the opinion about it, please help to improve it

Code with us

Even better, add functionality, streamline elements