

Organize and Share Information

- Objective
- Introduction
- Infrastructure
- Apps
- Organize Information
- Share Information
- References

Objective

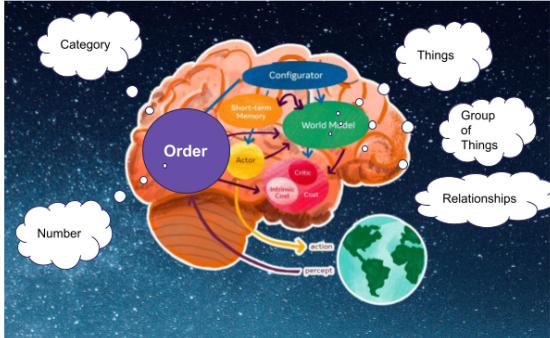
Organize and Share Information in the current workflow.

Introduction

In order to achieve the infrastructure instance is revised. The main approach is based on *David Allen Getting Things Done (GTD)*¹, and the general of the introduction are based on *The Order of Things*⁸, *A Beginner's Guide to Constructing the Universe*⁹ and *The Ending of Time*⁰.

Order of Things

Perception of the Order of its Surroundings



"Almost everything you can think of that represents an underlying order"



DAITAN | ©Daitan2020 - Private & Confidential - v4.1 3

The Ending of Time

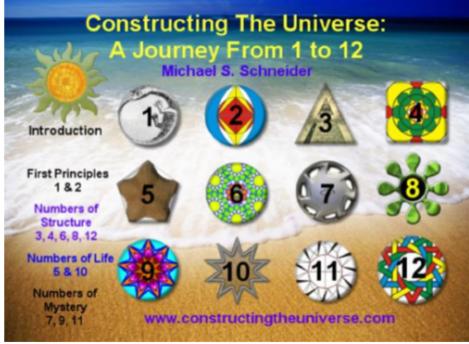
Life is a Movement in Relationship



"All life is a movement in relationship living thing on earth which is not relate something or other" —J. Krishnamurti

Constructing the Universe

The Mathematical Archetypes of Nature Art and Science



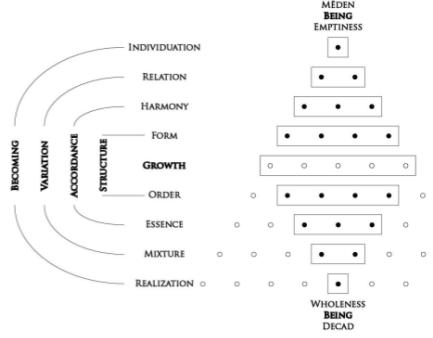
"The number is the within of all things" —Pythagoras



DAITAN | ©Daitan2020 - Private & Confidential - v4.1 5

Constructing the Universe

A voyage from 1 to 10



Infrastructure

Linux Desktop

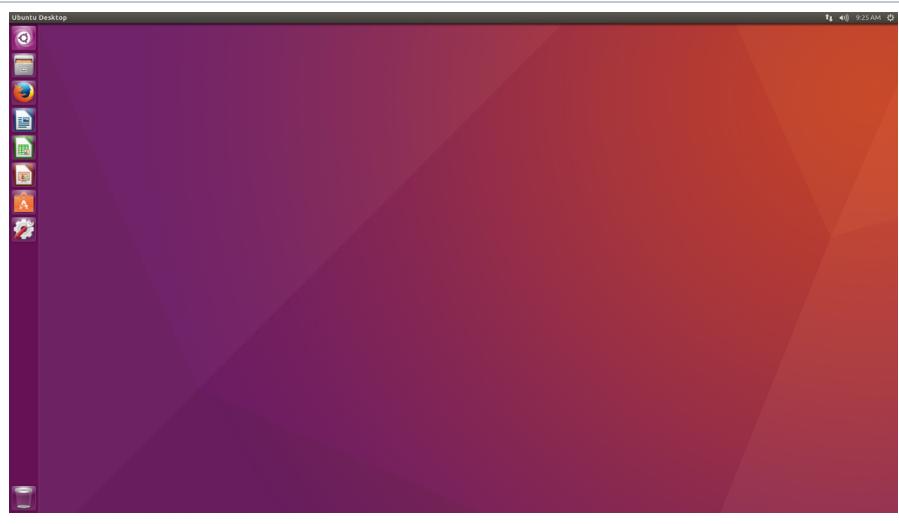
Desktop

It is the bundle of components that provide you common graphical user interface (GUI) elements such as icons, toolbars, wallpapers, and desktop widgets.

Thanks to the desktop environment, you can use Linux graphically using your mouse and keyboard like you do in other operating systems like Windows and macOS.

There are several Desktop environments and these [Desktop environments](#) determines what your Linux system looks like and how you interact with it.

Here the [Unity Desktop](#) will be chosen.



Tip

Install [Unity Desktop](#) and it will improve Linux appearance and your interaction with it!

Install Unity Desktop e Compiz

```
sudo apt-get install compiz  
sudo apt install ubuntu-unity-desktop  
sudo dpkg-reconfigure lightdm  
sudo reboot
```

Compiz Configuration

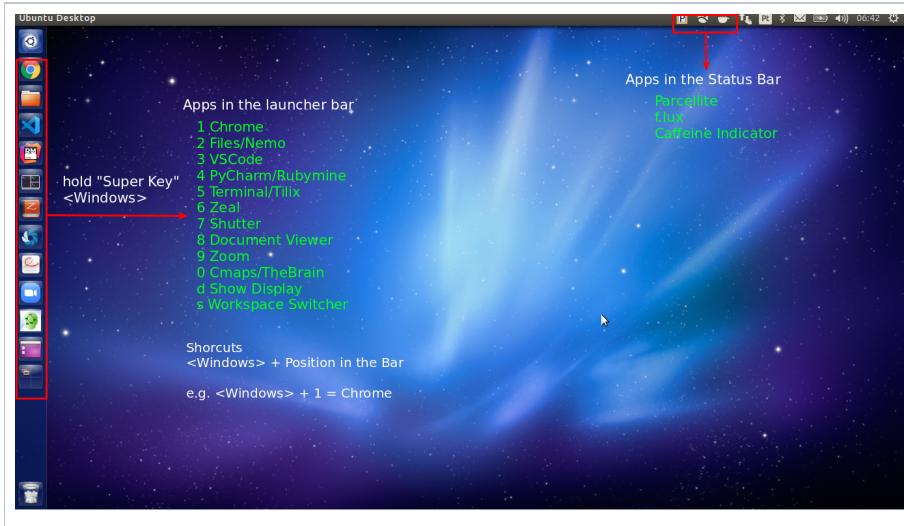
[Compiz](#) usually comes with Ubuntu, so install from Ubuntu Software "CompizConfig Settings Manager".

In order to config set keymaps for - in [Windows Manager](#) -> Grid -> Bindings. Check [Cycle Through Multiple Sizes](#) in [Resize Actions](#) tab. Click in [Windows Manager](#) and in [Place Windows](#) unticked [Workarounds](#) and in [Placement Mode](#) select [Placeacross all outputs](#). The [Placement Mode Smart](#) is the key for remembering your desired position!

Launcher Bar

Launcher Bar

A **Launcher** is an object that performs a specific action when you open it. You can find launchers in the panels, in the panel menubar, and on the desktop. A launcher is represented by an icon in all of these locations. You might use a launcher to do any of the following: [Start a particular application](#).



Tip

Here some **Linux Apps** are chosen as a suggestion but they have to be the most important and frequently accessed.

Organize the **Apps** in the **Launcher Bar** in order to have easy access to start or activate them. e.g. by the corresponding shortcut e.g. "Super Key" + "App position" in it!

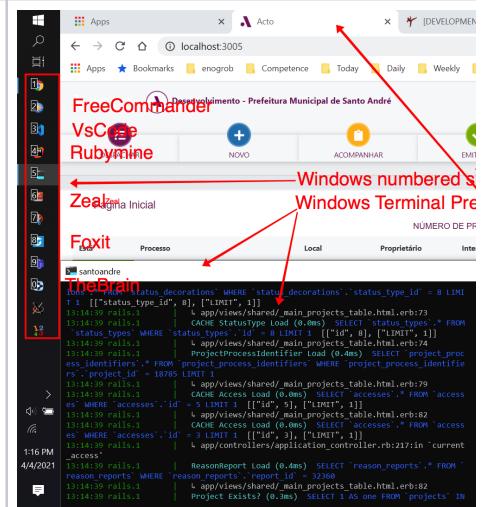

Ubuntu
★★★★★


openSUSE Leap 42
★★★★★


SUSE Linux Enterprise Server 12
★★★★★


Debian GNU/Linux
★★★★★


Kali Linux
★★★★★



Tip

Here some **Windows Apps** are chosen as a suggestion but they have to be the most important and frequently accessed.

Organize the **Apps** in the **Launcher Bar** in order to have easy access to start or activate them. e.g. by the corresponding shortcut e.g. "Super Key" + "App position" in it!

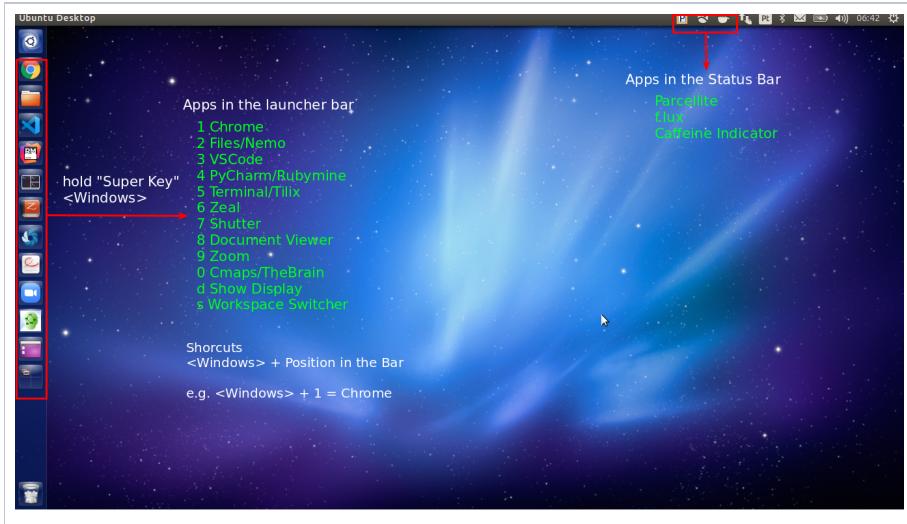
See for further info: [WSL: Operational Systems as Apps](#)

<https://github.com/enogrob/research-configuring-laptop-with-windows-subsystem-linux-wsl-2>

Menu Bar

Menu Bar

In the **Menu Bar** is usually may have some apps also to be easily accessed in order to perform handy actions such as the block the screen saver e.g. **Caffeine**, auto-copy on selection e.g. **Parcellite** or even to provide a confortable light to the eyes e.g. **f.lux**.



Tip

Caffeine prevents the desktop from becoming idle when an application is running full-screen. A desktop indicator **Caffeine indicator** supplies a manual toggle, and the command 'caffeinate' can be used to prevent idleness for the duration of any command.

Parcellite auto-copy text to the clipboard on selection.

f.lux it makes the color of your computer's display adapt to the time of day, warm at night and like sunlight during the day.

Caffeine, **Caffeine Indicator**, **Parcellite** and **f.lux** can be installed from the [Ubuntu Software](#) app.

Apps

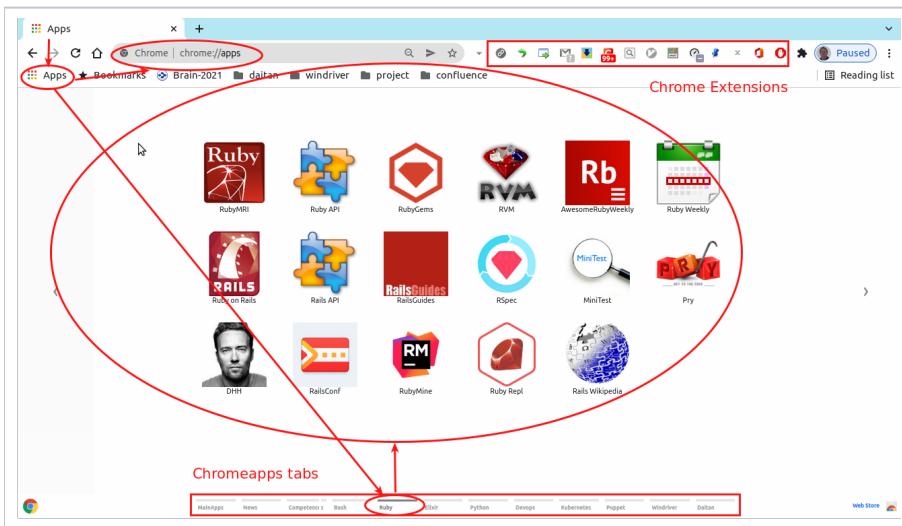
Apps

In the **Launcher Bar** are place the **Apps**, usually chosen by its importance and use frequency. Here some are chosen as instance.

1-Chrome

Chrome

Chrome can help you stay productive and get more out of your browser.



Install Chrome

```
pushd /tmp  
wget https://dl.google.com/linux/direct/google-chrome-stable_current_amd64.deb  
sudo dpkg -i --force-depends google-chrome-stable_current_amd64.deb  
sudo apt-get install -f  
echo "alias google-chrome='google-chrome --password-store=basic' " >> $HOME/.bashrc  
popd
```



Extensions

[AdBlock](#) - Block ads and pop-ups on YouTube, Facebook, Twitch, and your favorite websites.
[Auto Copy](#) - Automatically copy selected text to the clipboard. Has many configurable options for controlling the behavior.
[Click&Clean](#) - Deletes typed URLs, Cache, Cookies, your Download and Browsing History.
[Chrono Download Manager](#) - Manage all your downloads inside Chrome.
[Close Tab](#) - A stationary button that closes the current tab. Clean up tabs with less effort.
[Copy as Markdown](#) - Copy Link or Image as Markdown code.
[DevDocs](#) - This extension allows you to quickly open a tab and get straight into [DevDocs.io](#).
[Gerrit Monitor](#) - Monitor your CLs on Gerrit.
[Google Dictionary \(by Google\)](#) - View definitions easily as you browse the web.
[Google Mail Checker](#) - Displays the number of unread messages in your Google Mail inbox.
[Ignore X-Frame headers](#) - Drops X-Frame-Options and Content-Security-Policy HTTP response headers, allowing all pages to be iframed.
[Linkclump](#) - Lets you open, copy or bookmark multiple links at the same time.
[Link saver for OS X](#) - Saves current page as .webloc file.
[Magic Actions](#) - Will turn your computer into a Home Theater.
[Office](#) - View, edit, and create Office documents in your browser.
[Open-as-Popup](#) - It adds a button to the toolbar, a context menu entry and a keyboard shortcut that allows you to move a tab to a new popup window.
[Pin Tab](#) - A browser action to pin a tab.
[Print Friendly & PDF](#) - Make any web page Print Friendly & PDF.
[Recent History](#) - This extension displays your recent history, recently closed tabs, most visited pages and recent bookmarks in a one click pop-up.
[Select and convert to markdown](#) - Copy select text and convert to Markdown code.
[Speedtest by Ookla](#) - Take a Speedtest directly from your toolbar to quickly test your internet performance without interruption.
[TabJump](#) - Intelligent Tab Navigator.
[TheBrain - BrainBox](#) - Instantly capture URLs in BrainBox for use later with TheBrain.
[Vimium](#) - Vimium provides keyboard shortcuts for navigation and control in the spirit of Vim.
[Vim what?](#) - Vim cheatsheets.
[Where Am I?](#) - VPN Checker - Displays an icon of a flag of the country you appear to be coming from.

For further extensions check [Chrome Web Store](#).



Chrome Apps

[Bash](#) - Chrome apps for bash.
[Competence](#) - Chrome apps for learning, resources and knowledge management.
[DevOps](#) - Chrome apps for DevOps.
[Elixir](#) - Chrome apps for Elixir programming language.
[Kubernetes](#) - Chrome apps for Kubernetes.
[MainApps](#) - Chrome apps mostly Google.
[Puppet](#) - Chrome apps for Puppet configuration management.
[Python](#) - Chrome apps for Python.
[Ruby](#) - Chrome apps for Ruby programming language.
[Webtools](#) - Chrome for online webtools.

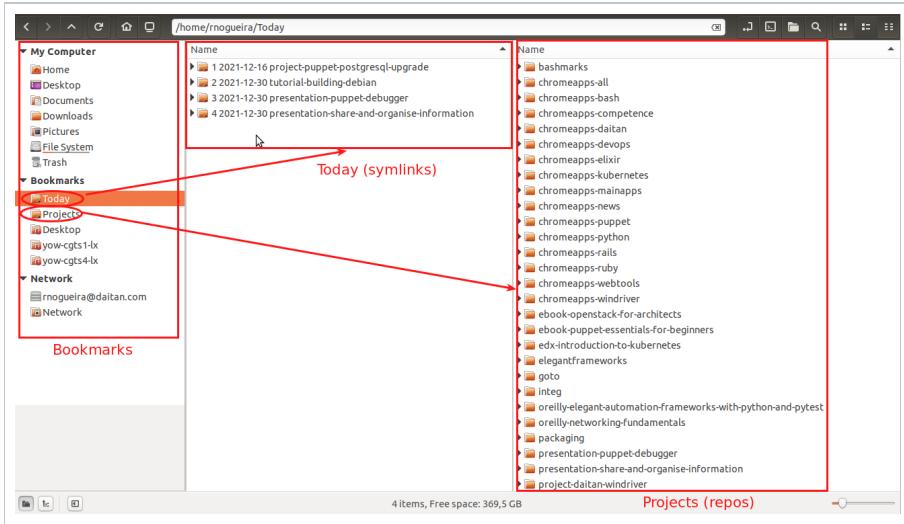
For futher info about Chrome Apps check [Create and publish custom Chrome apps & extensions](#).

2-Files/Nemo



Files/Nemo

Nemo has better features than nautilus e.g. the [Dual pane](#) which allows to implement the [GTD Technique](#) and also will easier to perform folder/file actions.



Install Nemo

```
sudo add-apt-repository ppa:embrosyn/cinnamon
sudo apt install nemo
xdg-mime default nemo.desktop inode/directory application/x-gnome-saved-search
gsettings set org.gnome.desktop.background show-desktop-icons false
```



Tip

Nemo allows to implement Getting Things Done technique(GTD) 1

Today, Someday, Scheduled, Logbook made of symlinks from **Projects** folder.

Projects folder - usually will contains repos.

Inbox place to collect things, when ready, they grouped in a folder and placed in Projects.

For further about GTD implementation see [Organize Information](#).



Tip

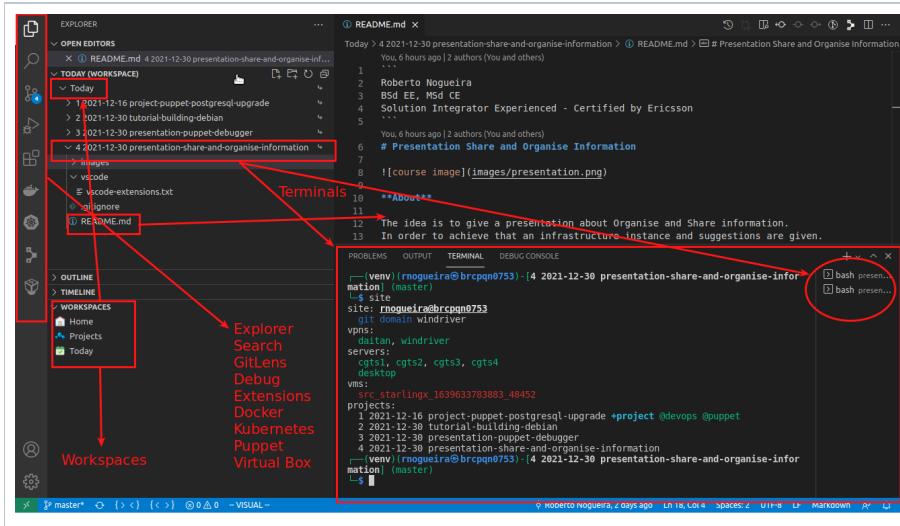
Double Commander is an other option a **free** cross platform open source file manager with two panels side by side. It is inspired by Total Commander and features some new ideas.

3-VSCode



VSCode

VSCode is a source-code editor made by Microsoft for Windows, Linux and macOS.^[9] Features include support for debugging, syntax highlighting, intelligent code completion, snippets, code refactoring, and embedded Git. Users can change the theme, keyboard shortcuts, preferences, and install extensions that add additional functionality.



Install VSCode

```

pushd /tmp
wget https://code.visualstudio.com/sha/download?build=stable&os=linux-deb-x64
sudo dpkg -i --force-depends ./code*.deb
sudo apt-get install -f
popd

```



Extensions

acherkashin.virtualbox-extension - The extension aim is to decrease your switching between different windows and allow you to control your Virtual Machines right from the Visual Studio Code activity bar.

adrianwilczynski.open-folder-in-explorer - Open selected folder in explorer (or new instance) from context menu.

alon-320.integrated-terminal-filedir - Add command to open an integrated terminal at the current file's directory.

bbenoist.vagrant - Vagrant support for Visual Studio Code.

bung87.rails - Ruby on Rails support for Visual Studio Code.

bung87.vscode-gemfile - provide hover link in Gemfile refers to online site.

CTC.vscode-tree-extension - This VSCode Extension adds syntax highlighting support for .tree files.

cweijan.vscode-office - This extension supports previewing common office file formats.

deerawan.vscode-dash - [Dash](#) documentation integration for [Visual Studio Code](#).

donjayamanne.githistory - Git History, Search and More (including git log).

eamodio.gitlens - GitLens supercharges the Git capabilities built into Visual Studio Code.

FerrierBenjamin.fold-unfold-all-icon - Just add a fold and a unfold button in the status bar (at the bottom left).

Hridoy.rails-snippets - This extension for Visual Studio Code adds snippets for Ruby on rails.

mhutchie.git-graph - View a Git Graph of your repository, and easily perform Git actions from the graph.

mindaro-dev.file-downloader - This extension surfaces an API usable by other extensions that downloads a file whose URL is provided.

ms-azuretools.vscode-docker - Docker for Visual Studio Code.

ms-kubernetes-tools.vscode-kubernetes-tools - Visual Studio Code Kubernetes Tools.

ms-python.python - Python extension for Visual Studio Code.

ms-python.vscode-pylance - Fast, feature-rich language support for Python.

ms-toolsai.jupyter - Jupyter Extension for Visual Studio Code.

ms-toolsai.jupyter-keymap - Jupyter Keymaps Extension for Visual Studio Code.

ms-toolsai.jupyter-renderers - Renderers for Jupyter Notebooks in Visual Studio Code.

ms-vscode-remote.remote-containers - Visual Studio Code Remote - Containers.

ms-vscode-remote.remote-ssh - Visual Studio Code Remote - SSH.

ms-vscode-remote.remote-ssh-edit - The Remote - SSH extension lets you use any remote machine with a SSH server as your development environment.

ms-vscode-remote.remote-wsl - this extension lets you use VS Code on Windows to build Linux applications that run on the Windows Subsystem for Linux (WSL).

ms-vscode-remote.vscode-remote-extensionpack - extension pack allows you to open any folder in a container, on a remote machine, or in the Windows Subsystem for Linux (WSL) .

ninoseki.vscode-gem-lens - A VS Code extension to show the latest version of a gem in Gemfile and gemspec.

puppet.puppet-vscode - Puppet Visual Studio Code Extension.

rebornix.ruby - This extension provides enhanced Ruby language and debugging support for Visual Studio Code.

redhat.vscode-yaml - This extension provides enhanced Ruby language and debugging support for Visual Studio Code.

sandcastle.vscode-open - Opens files using the OS's default program for the file type.

shanehofstetter.rails-open-partial - Open partial view files (*.haml, *.erb) with go to definition.

sianglim.slim - Slim language support for Visual Studio Code.

simspsencer.vscode-autohide - Causes the side bar to be hidden whenever the user clicks into the text editor.

tomoki1207.pdf - Display pdf in VSCode.

tomsaunders.vscode-workspace-explorer - Workspace Explorer provides a convenient UI to quickly switch your workspace or open a workspace in a new window.

Tyriar.sort-lines - Sort lines of text in Visual Studio Code.

vscode-icons-team.vscode-icons - Bring icons to your [Visual Studio Code](#).

vscodevim.vim - VSCodeVim is a Vim emulator for [Visual Studio Code](#).

wingrunr21.vscode-ruby - This extension provides improved syntax highlighting, language configuration, and snippets to Ruby and ERB files within Visual Studio Code.

For further extensions check [VSCode marketplace](#).

Install VSCode Extensions

```
pushd $HOME/Projects  
git clone git@github.com:enogrob/presentation-share-and-organise-information.git  
cd vscode  
cat vscode-extensions.txt | xargs -L 1 echo code --install-extension  
popd
```

List VSCode Extensions

```
code --list-extensions  
:  
vscodevim.vim  
wingrunr21.vscode-ruby
```

4-PyCharm/Rubymine

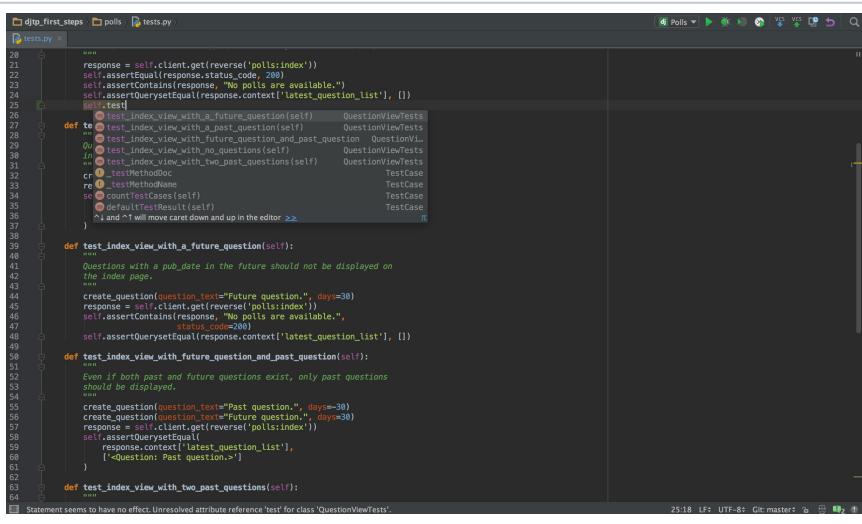


PyCharm

PyCharm is an integrated development environment (IDE) used in computer programming, specifically for the [Python](#) language.

Install PyCharm

```
pushd /tmp
wget https://download.jetbrains.com/python/pycharm-community-2021.3.1.tar.gz?_ga=2.249990982.1922373967.1641087326-2050041162.1632025130&_gac=1.251389234.1640950090.
EAIAIaQobChMI9ryS9_WN9QIVNYWVAh0cuAp6EAEYASAAEgLeUPD_BwE&_gl=1*gga9rf*_ga*MjA1MDA0MTE2Mi4xNjMyMDI1MTMw*_ga_V0XZL7
QHEB*MTY0MTA4NzMyNi44LjEuMTY0MTA4NzQzNC41NQ..
sudo tar xvzf pycharm-*.tar.gz -C /opt
cd /opt/pycharm-*/bin
./pycharm.sh
popd
```



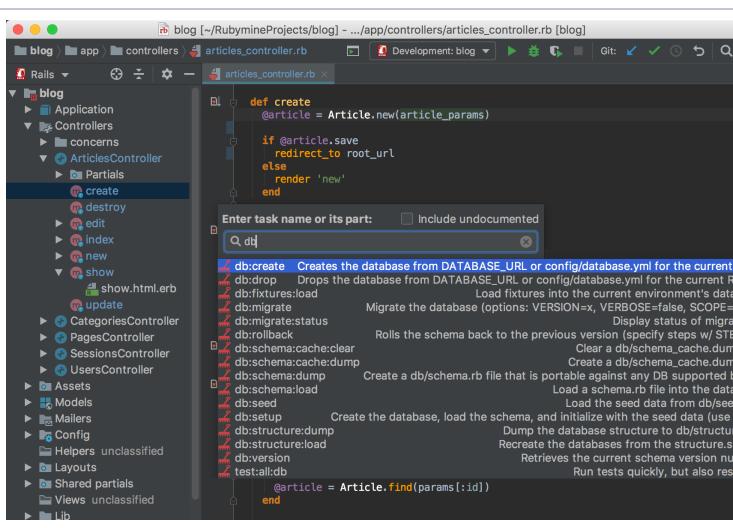
```
def test_index_view_with_a_future_question(self):
    """
    Questions with a pub_date in the future should not be displayed on
    the index page
    """
    create_question(text="Future question.", days=30)
    response = self.client.get(reverse('polls:index'))
    self.assertEqual(response['Content-Type'], 'application/json')
    self.assertContains(response, "No polls are available.", status_code=200)
    self.assertQuerysetEqual(response.context['latest_question_list'], [])

def test_index_view_with_a_past_question(self):
    """
    Even if both past and future questions exist, only past questions
    should be displayed.
    """
    create_question(text="Past question.", days=-30)
    create_question(text="Future question.", days=30)
    response = self.client.get(reverse('polls:index'))
    self.assertEqual(response['Content-Type'], 'application/json')
    self.assertContains(response, "No polls are available.", status_code=200)
    self.assertQuerysetEqual(response.context['latest_question_list'], [])

def test_index_view_with_no_questions(self):
    """
    An empty list should be displayed for no questions.
    """
    response = self.client.get(reverse('polls:index'))
    self.assertEqual(response['Content-Type'], 'application/json')
    self.assertContains(response, "No polls are available.", status_code=200)
    self.assertQuerysetEqual(response.context['latest_question_list'], [])
```

RubyMine

RubyMine is an integrated development environment (IDE) used in computer programming, specifically for the Ruby language.



```
def create
  @article = Article.new(article_params)

  if @article.save
    redirect_to root_url
  else
    render 'new'
  end
end
```

Install Rubymine

```
pushd /tmp
wget https://download.jetbrains.com/ruby/RubyMine-2021.3.1.tar.gz?
_g1=1*xm84jp*_ga*MjA1MDA0MTE2Mi4xNjMyMDI1MTMw*_ga_VOXZL7QHEB*MTY0MTA4NzMyNi44LjEuMTY0MTA4NzQzNC41NQ..&_ga=2.
158108538.1922373967.1641087326-2050041162.1632025130&_gac=1.215624227.1640950090.
EAIAIQobChM19ryS9_WN9QIVNYWVAh0cuAp6EAEYASAAEgLeUPD_BwE
sudo tar xvzf RubyMine-*.tar.gz -C /opt
cd /opt/RubyMine-*/bin
./rubymine.sh&
popd
```

Extensions

[Diff / Patch File Support](#) - Syntax highlighting for .diff files and .patch files.

[Elixir](#) - Elixir support for JetBrains IDEs.

[IdeaVim](#) - Vim emulation plugin for IntelliJ Platform-based IDEs.

[Node.js](#) - Support for [Node.js](#) projects.

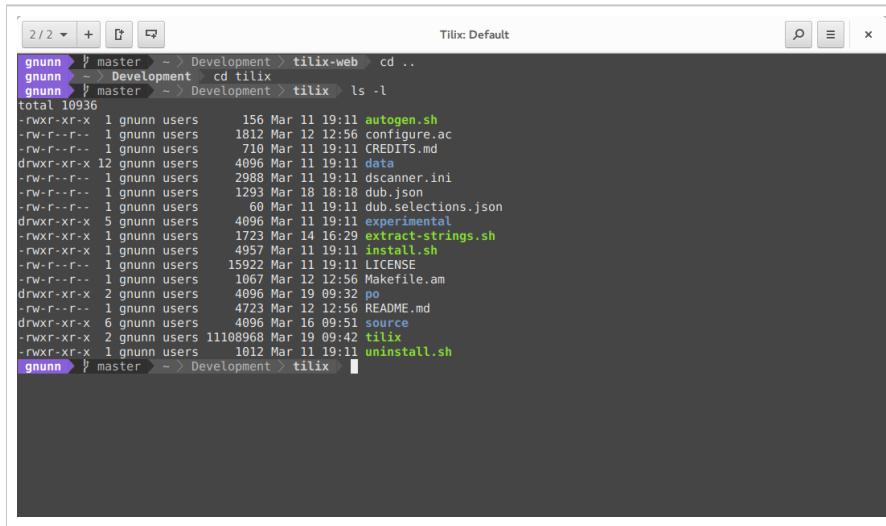
[Vagrant](#) - [Vagrant](#) support Feature.

For further extensions check [JetBrains Marketplace](#).

5-Terminal/Tilix

Terminal/Tilix

[Tilix](#) is an advanced GTK3 tiling terminal emulator that follows the Gnome Human Interface Guidelines.



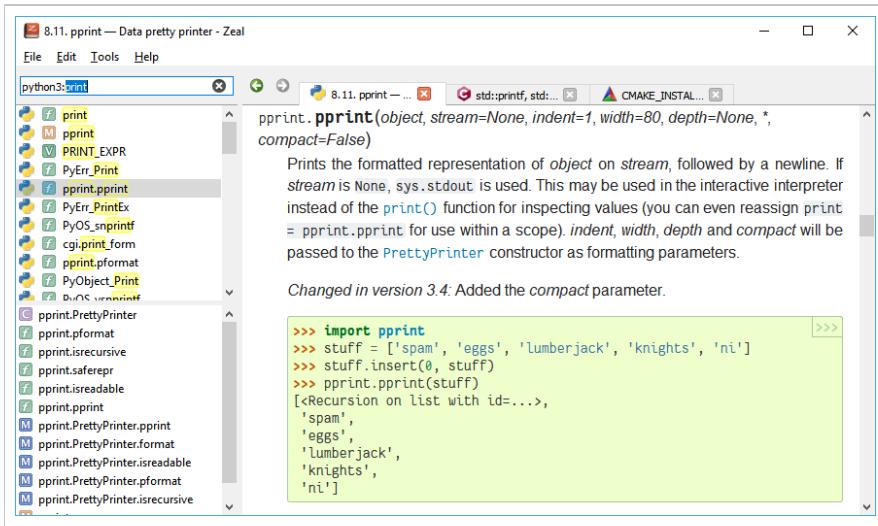
Install Tilix

```
sudo add-apt-repository ppa:webupd8team/terminix
sudo apt-get update
sudo apt install tilix
sudo ln -sf /etc/profile.d/vte-2.91.sh /etc/profile.d/vte.sh
echo "[[ $TERMINIX_ID || $VTE_VERSION ]] && source /etc/profile.d/vte.sh" >> ~/.bashrc
source ~/.bashrc
```

6-Zeal

Zeal

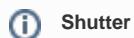
[Zeal](#) is an offline documentation browser for software developers.



Install Zeal

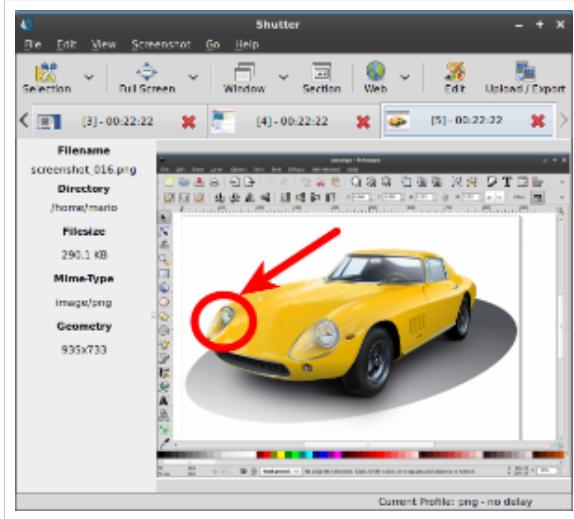
```
sudo apt-get install zeal
```

7-Shutter



Shutter

Shutter the feature-rich screenshot tool



Install Shutter

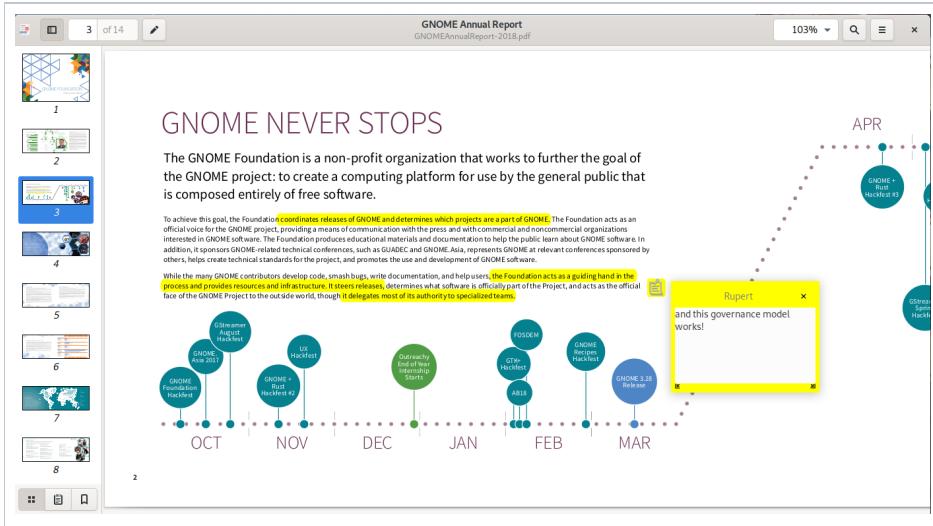
```
sudo apt-get install shutter
```

8-Document Viewer



Document Viewer

Evince (Document Viewer) is a free and open source document viewer supporting many document file formats including PDF, PostScript, DjVu, TIFF, XPS and DVI.



Install Document Viewer

```
sudo apt-get install evince
```

Ubuntu has evince in main and in the default install. [Evince packages in the Ubuntu repositories.](#)



Tip

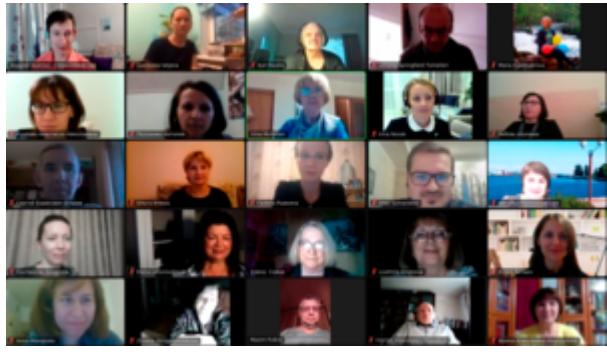
[PdfShuffler](#) is a tool for rearranging and modifying PDF files. This could be also interesting to [download](#) and install.

9-Zoom



Zoom

[Zoom](#) is a proprietary video teleconferencing software program developed by [Zoom Video Communications](#).



Install Zoom

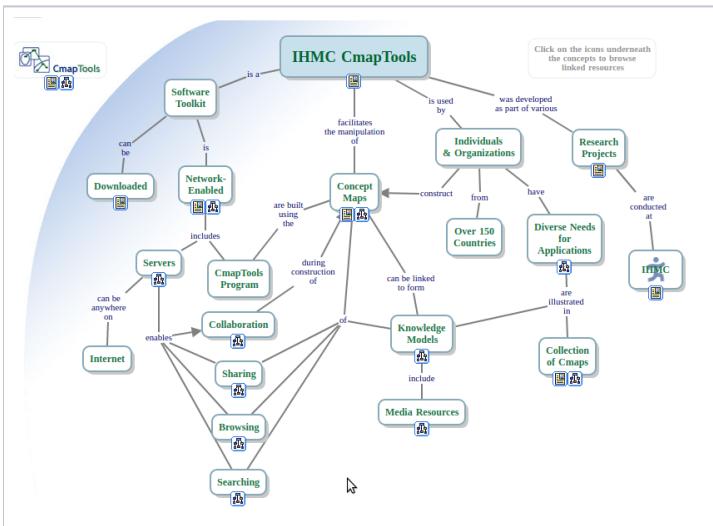
```
pushd /tmp
wget https://zoom.us/client/latest/zoom_amd64.deb
sudo dpkg -i --force-dependencies ./zoom_amd64.deb
sudo apt-get install -f
popd
```

0-Cmaps/TheBrain

CmapTools

CmapTools is a [concept mapping](#) software developed by the [Florida Institute for Human and Machine Cognition \(IHMC\)](#).^[1] It allows users to easily create graphical nodes representing concepts, and to connect nodes using lines and linking words to form a [network of interrelated propositions](#) that represent knowledge of a topic.

For further info see [Documentation Cmaps](#).

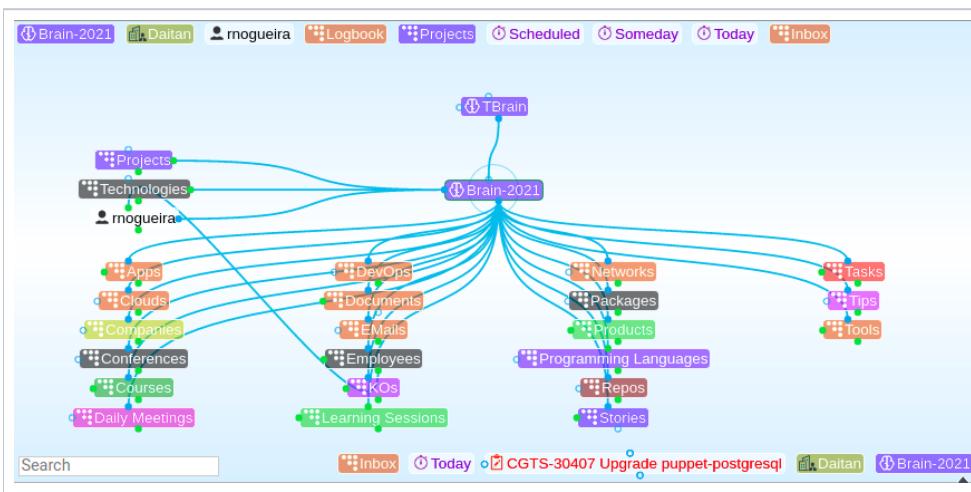


 The **CmapTools** is free as Desktop or even in [Cmap Cloud](#). [Signup](#) and then it is possible to create also in [Cmap Cloud](#) from your account.

TheBrain

The Brain is a [mind mapping](#) and [personal knowledge base](#) software from [TheBrain Technologies](#).^[1] It uses a dynamic graphical interface that maps hierarchical and network relationships.

For further info see [The Brain User Guide](#).



Tip

The Brain has a free Desktop version. [Signup](#) and then it is possible to create Brains from your account.

[d>Show Desktop](#)

i Show Desktop

Show Desktop toggle the Desktop.

✓ Tip

In order to add **Show Desktop** in the Launcher Bar see [How to add a "Show desktop" icon to the launcher?](#)

s-Workspace Switcher

i Workspace Switcher

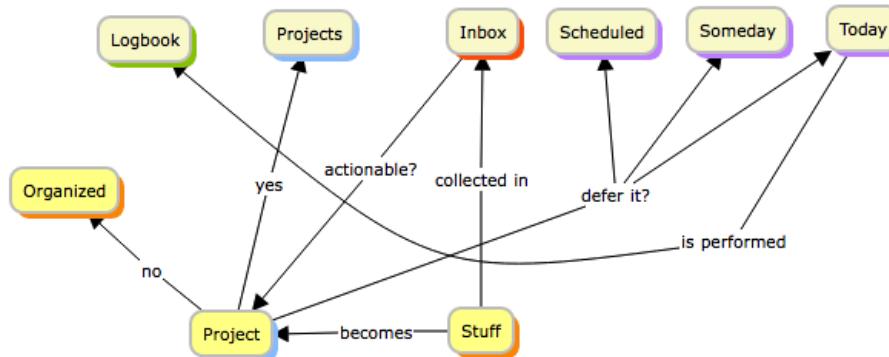
Workspace Switcher switch among Desktop workspaces.

✓ Tip

In order to add **Workspace Switcher** in the Launcher Bar see [How to add/restore workspace on the launcher?](#)

Organize Information

GDT



i GDT Implementation

Getting Things Done (GDT) can be implemented with the help of a **Today** bash script, [Files/Nemo](#) and [The Brain](#).

Today Installation

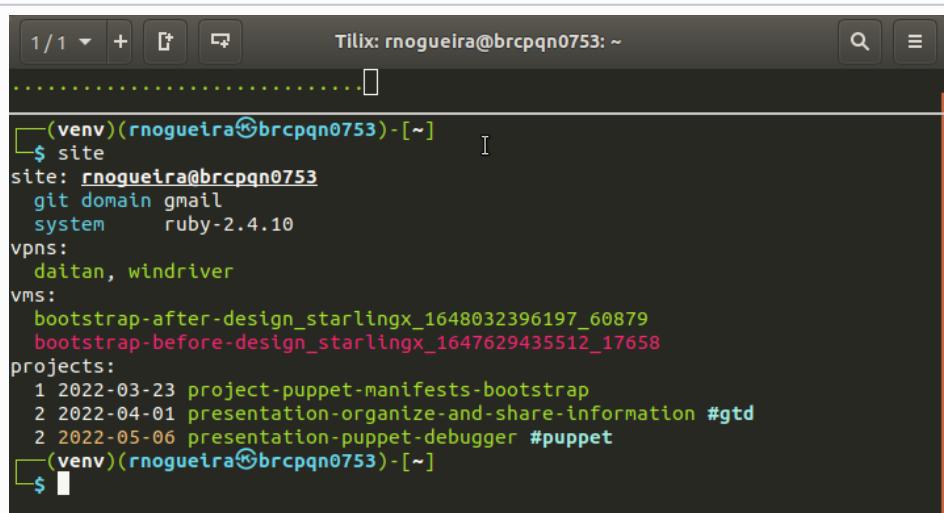
```
pushd /tmp
git clone git@github.com:enogrob/project-today-manager.git
source ./project-today-manager/today
mv project-today-manager ~/Projects
echo "test -f ~/Projects/project-today-manager/today && source ~/Projects/project-today-manager/today" >> ~/.bashrc
source ~/.bashrc
popd
```



Today is still under development!



Site Manager is an example of use of **GTD** together with **Tilix** in a CLI daily activities. It shows the `vpn`, `servers` and `vms` current status, besides the further info about `projects` (which follows GTD approach) and `screen sessions`.



The screenshot shows a terminal window titled "Tilix: rnogueira@brcpqn0753: ~". The command `$ site` is run, displaying the following information:

- site:** `rnogueira@brcpqn0753`
 - git domain gmail
 - system ruby-2.4.10
- vpns:**
 - daitan, windriver
- vms:**
 - bootstrap-after-design_starlingx_1648032396197_60879
 - bootstrap-before-design_starlingx_1647629435512_17658
- projects:**
 - 1 2022-03-23 project-puppet-manifests-bootstrap
 - 2 2022-04-01 presentation-organize-and-share-information #gtd
 - 2 2022-05-06 presentation-puppet-debugger #puppet

The terminal ends with `$` and a cursor.

For further info see [Site Manager](#).

Stuff

The Stuff is collected in **Inbox**.

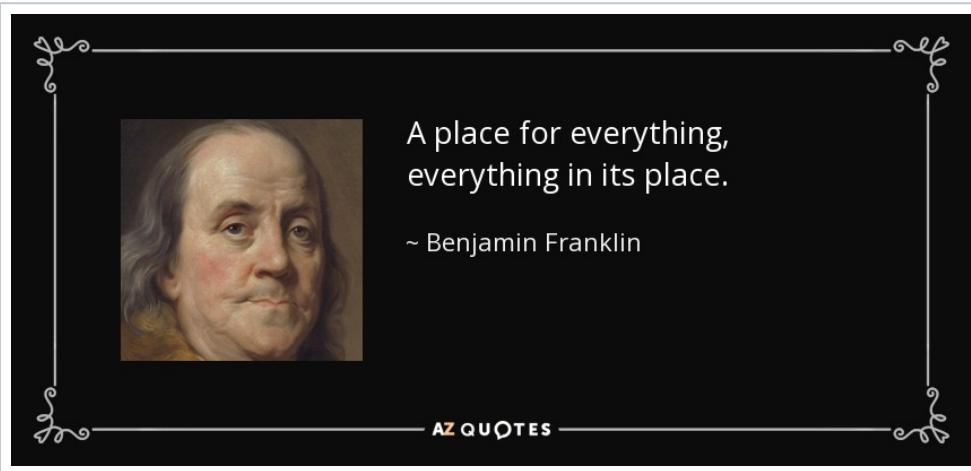
Process

The **Stuff** is revised and processed accordingly.

Projects If it actionable and is going to be a Project, a Project folder is created with Stuff as a content and It is placed in it. Otherwise it will be Organized.

Organize

If the Input is not actionable a place shall be found for it .



Perform it

If the Input is actionable and was placed as Project, so it still has to be placed in Time.

Today If the Project is going to be currently executed a symlink of it is placed in it.

Scheduled If the Project is going to be currently executed a symlink of it is placed in it.

Someday If there no execution date defined for the Project, a symlink of it is placed in it.

Log it

Logbook if the Project has been performed, the Today symlink(prefixed with performed Date) is moved to it.

Aliases and Navigation

Aliases and Navigation

Alias and navigate to directories with tab completion in Linux. See below for installation locally or even inside the Build Servers or Debian Pod.
To be able to use that locally the following definitions has to be included in `~/ .bashrc` :

```
#Debian Build Environment
export TOOL_HOME=~/DebianBuild
export WORKSPACE_HOME=~/DebianBuildWorkspace
export PROJECT=stx-deb-bld-1
export MY_REPO_ROOT_DIR=/localdisk/$USER/localdisk/designer/$USER/$PROJECT
export MY_BUILD_DIR=/localdisk/$USER/localdisk/loadbuild/$USER/$PROJECT

test -f $TOOL_HOME/tools/import-stx && pushd $TOOL_HOME/tools && source $TOOL_HOME/tools/import-stx &&
popd
```

GoTo Installation

```
! test -L ~/Projects && mkdir -p ~/Things/Projects && ln -sf ~/Things/Projects ~/Projects
cd ~/Projects
git clone --branch where-ami-in-goto git@github.com:enogrob/goto.git
echo "test -f ~/Projects/goto/goto.sh && source ~/Projects/goto/goto.sh" >> ~/.bashrc
source ~/.bashrc
```

Initial Setup (Debian - Local filesystem)

```
source ~/.bashrc

test -f ${HOME}/.config/goto && rm -f ${HOME}/.config/goto
goto -r aptly           ${WORKSPACE_HOME}/aptly
goto -r builds           ${WORKSPACE_HOME}
goto -r config           /localdisk/designer/$USER/stx-deb-bld-1/localdisk/designer/$USER/$PROJECT/cgcs-root/stx
/config
goto -r config-files    /localdisk/designer/$USER/stx-deb-bld-1/localdisk/designer/$USER/$PROJECT/cgcs-root/stx
/config-files
goto -r downloads        ${HOME}/Downloads
goto -r home              ${HOME}
goto -r iso               /localdisk/designer/rnogueira/$PROJECT/localdisk/deploy/
goto -r integ             /localdisk/designer/$USER/stx-deb-bld-1/localdisk/designer/$USER/$PROJECT/cgcs-root/stx
/integ
goto -r metal             /localdisk/designer/$USER/stx-deb-bld-1/localdisk/designer/$USER/$PROJECT/cgcs-root/stx
/metal
goto -r mirrors           /localdisk/designer/$USER/$PROJECT/mirrors
goto -r modules           /localdisk/designer/$USER/stx-deb-bld-1/localdisk/designer/$USER/$PROJECT/cgcs-root/stx
/integ/config/puppet-modules
goto -r projects          ${HOME}/Projects
goto -r stx-puppet         /localdisk/designer/$USER/stx-deb-bld-1/localdisk/designer/$USER/$PROJECT/cgcs-root/stx
/stx-puppet
goto -r stx-tools          /localdisk/designer/$USER/stx-deb-bld-1/localdisk/designer/$USER/$PROJECT/stx-tools
goto -r tmp                /tmp
goto -r today              ${HOME}/Today
goto -r tools              $TOOL_HOME/tools
clear
goto -l
```

Initial Setup (Debian - Builder Pod)

```
test -f $HOME/.config/goto && rm -f $HOME/.config/goto
test -z "$MY_BUILD_DIR" && export MY_BUILD_DIR=/localdisk/loadbuild/$USER/$PROJECT
echo "export MY_BUILD_DIR=/localdisk/loadbuild/$USER/$PROJECT" > ~/.bashrc

goto -r assemblies      $MY_BUILD_DIR/std
goto -r binaries        $MY_BUILD_DIR/binaries
goto -r builds          $MY_BUILD_DIR
goto -r clean_dir       /tmp/clean_dir
goto -r config          $MY_REPO_ROOT_DIR/cgcs-root/stx/integ/config
goto -r config-files   $MY_REPO_ROOT_DIR/cgcs-root/stx/config-files
goto -r home            $HOME
goto -r integ           $MY_REPO_ROOT_DIR/cgcs-root/stx/integ
goto -r iso              /localdisk/deploy
goto -r logs             /localdisk/log
goto -r metal            $MY_REPO_ROOT_DIR/cgcs-root/stx/metal
goto -r modules          $MY_REPO_ROOT_DIR/cgcs-root/stx/integ/config/puppet-modules
goto -r projects         $HOME/Projects
goto -r openstack        $MY_REPO_ROOT_DIR/cgcs-root/stx/integ/config/puppet-modules/openstack
goto -r repos            $MY_REPO_ROOT_DIR
goto -r stx-puppet      $MY_REPO_ROOT_DIR/cgcs-root/stx/stx-puppet
goto -r stx-tools        $MY_REPO_ROOT_DIR/stx-tools
goto -r tmp               /tmp
clear
goto -l
```

Initial Setup (Centos)

```
source ~/env.wrcp_dev

alias tree1='tree -L 1 .
'
test -f $HOME/.config/goto && rm -f $HOME/.config/goto

goto -r projects      $HOME/Projects
goto -r today         $HOME/Today

goto -r assemblies    $MY_BUILD_DIR/std
goto -r builds        $MY_BUILD_DIR
goto -r config         $MY_REPO_ROOT_DIR/cgcs-root/stx/integ/config
goto -r downloads     $MY_REPO_ROOT_DIR/cgcs-root/stx/downloads
goto -r home          $HOME
goto -r integ          $MY_REPO_ROOT_DIR/cgcs-root/stx/integ
goto -r iso            $HOME/loadbuilds/$PROJECT/export
goto -r licences      /folk/cgts/lab/licenses
goto -r mirrors        /import/mirrors/starlingx
goto -r modules        $MY_REPO_ROOT_DIR/cgcs-root/stx/integ/config/puppet-modules
goto -r openstack      $MY_REPO_ROOT_DIR/cgcs-root/stx/integ/config/puppet-modules/openstack
goto -r repos          $MY_REPO_ROOT_DIR
goto -r tools          $MY_REPO_ROOT_DIR/stx-tools
goto -r tmp            /tmp
clear
goto -l
```

Share Information

Nowadays you can share media easily using urls. Here it is suggested two tools for [How to/Procedures](#) and [Methods of Procedure](#)(Mops or Cheatsheets)

How to/Procedures

[Gist](#) is a simple way to share snippets and pastes with others.

Roberto Nogueira
enogrob
Grounded in Telecommunications and Software Engineering but the main concern is with life itself as a whole and unconditioned.
A 18 followers · 2 following
ZotWorks

KO::How to install Puppet Debugger in Ubuntu

Puppet is a configuration management tool. The user describes system resources and their state, either using a Ruby DSL or Puppet's declarative language. This system information is stored in files called **manifests** file. Puppet discovers the system information via a utility called **Facter**, and compiles the manifests into a system-specific **catalog** containing resources and resource dependency, which are applied against the target systems. Any actions taken by Puppet are then reported.

We can configure systems with Puppet either in a client-server architecture, using the **Puppet agent** and **Puppet master** applications, or in a stand-alone architecture, using the **Puppet apply** application.

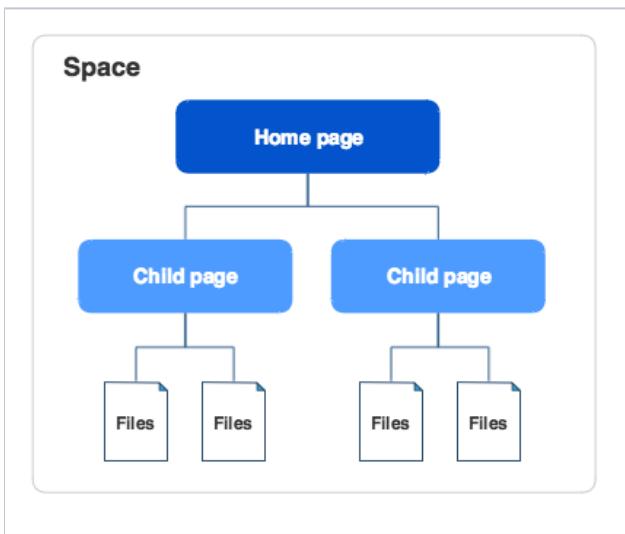
Question
How to install Puppet Debugger in Ubuntu

Tip

If you have an git account you have everything in place just access it by "<https://gist.github.com/<github id>>" or by the menu profile. So what are you waiting for?

Mops/CheatSheets

[Confluence](#) is a web-based corporate wiki (collaboration software) developed by Atlassian.



Tip

[Confluence](#) in it is possible to create Space for a Knowledge, Team, Documentation, Software project or even a Personal one. Start one!

References

1. [Getting Things Done: The Art of Stress-Free Productivity](#)
2. [Learning, Creating, and Using Knowledge: Concept Maps as Facilitative Tools in Schools and Corporations](#)
3. [Documentation Cmaps](#)
4. [The Brain User Guide](#)
5. [David Allen on GTD and TheBrain](#)
6. [Organize and Share Information](#)
7. [List of concept- and mind-mapping software](#)
8. [The Order of Things](#)
9. [A Beginner's Guide to Constructing the Universe](#)
0. [The Ending of Time](#)

