

# post install for servers

```
apt install -y curl && \
curl -s
"http://wiki.csgalileo.org/doku.php/tips:ubuntu:locale?do=export_code&codeblock=1" | /bin/bash
```

## script

```
apt remove -y unattended-upgrades

locale-gen it_IT.UTF-8
locale-gen en_US.UTF-8
update-locale LANG=it_IT.UTF-8 LC_MESSAGES=POSIX
timedatectl set-timezone Europe/Rome

grep -q menu-complete /etc/inputrc
if [ $? = 1 ]; then
cat >> /etc/inputrc <<EOF
"\M-s": menu-complete
"\e[A": history-search-backward
"\e[B": history-search-forward
"\M-o": "\C-p\C-a\M-f "

EOF
fi

[ -f /etc/tmux.conf ] || curl -s
"http://wiki.csgalileo.org/doku.php/tips:tmux?do=export_code&codeblock=4" > /etc/tmux.conf

cat > /etc/vim/vimrc.local <<EOF
setlocal ts=4 sts=4 sw=4 expandtab
autocmd FileType yaml setlocal ts=2 sts=2 sw=2 expandtab
set paste
set mouse=r
set showcmd      " Show (partial) command in status line.
set showmatch    " Show matching brackets.
set ignorecase   " Do case insensitive matching
set smartcase    " Do smart case matching
set incsearch    " Incremental search
set autowrite    " Automatically save before commands like :next and
:make
set hidden       " Hide buffers when they are abandoned

if has("syntax")
    syntax on
endif
```

```
set background=dark

if has("autocmd")
  au BufReadPost * if line("\") > 1 && line("\") <= line("$") | exe
"normal! g\" | endif
endif

if has("autocmd")
  filetype plugin indent on
endif

EOF

cat > /root/.aliases <<EOF
alias l="ls -alh --color"
alias ll="ls -alh --color"
alias lle="ls -alh --color | head -n30"
alias s="ssh -l root"
alias sud="sudo su"
alias llast="ls -alt | head"
alias upgrade="sudo apt-get update; sudo apt-get -y -V upgrade"
EOF

mkdir -p /root/.ssh
cat > /root/.ssh/authorized_keys <<EOF
ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAQAC9U1ZrT3Yy1xkLAc1iy6SgUXd1zIQYjE6WQ2Ww2nc
ZEjLoic0qUikaHjoaN2Ci1Fx0EEfU9Kbt0aglyMsK724n2qnrNHUUNq4yGsPjA8eAHOPDu
jGpnwShUTagREC8G5lmDRJoq6+23w0kPGyTy03WJ90V+oNMoarkkT9ycmMbrmtWIowTcse8
6l69uHPmUrpD5H0Z2y73syvLA94uqo/d3ToKV0p0SDXyJzFyq4oUx7JHLT6SqS3AWutmskM
xfB9hZVs1DB94yaHuk0msb0cdgg3s8w0zemhF9V6+G1dTivLCECIzXJe9h02lsCkD3HvceM
OT+McPSrullLIYU5h5 scipio@flinx
ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAQADJ/3Z7CfayHNUvNCirfWZz2uQ/h5om7xLU4ntShoC
g35EZ+F30hBZDlWZs0cpi3/lr8TBUfXCe0WERR8rmybQsr5mzJ4L8UZseCk6fyiDCnd0y6l
BXBAewCV0XE3E2D0mZ7HWdx8cUfklUHdP07QEU3/6e067/uXC3LTqK9moFeUllIYXTb9uD0
kUoqafNiYc+n9a0cQeV00kurtPSC74ozmaBfPFey0ua5NrdmPcT6h10QwRyLsXM0q8yYKyI
bkTBD5QtyQFVQtPD3z6kBdkdNR3vqkQzn5yJbiNXrFGMAWSPZrktFuo2zncHjw/IIJar04b
N1X09FmsE9na/hgWH venom@darkstar
EOF

curl -s
"http://wiki.csgalileo.org/doku.php/tips:ubuntu:locale?do=export_code&c
odeblock=2" > /root/.bashrc

update-alternatives --set editor /usr/bin/vim.tiny

[ -d /etc/cloud ] && echo "network: {config: disabled}" >
/etc/cloud/cloud.cfg.d/99-disable-network-config.cfg
```

```
[ -d /etc/cloud ] && perl -pi -e "s|disable_root: .*|disable_root: false|" /etc/cloud/cloud.cfg
```

## /root/.bashrc

```
# If not running interactively, don't do anything
[ -z "$PS1" ] && return

HISTCONTROL=ignoredups:ignorespace
shopt -s histappend
HISTSIZE=5000
HISTFILESIZE=10000
shopt -s checkwinsize
[ -x /usr/bin/lesspipe ] && eval "$(SHELL=/bin/sh lesspipe)"
if [ -z "$debian_chroot" ] && [ -r /etc/debian_chroot ]; then
    debian_chroot=$(cat /etc/debian_chroot)
fi

case "$TERM" in
    xterm-color) color_prompt=yes;;
esac

PS1='${debian_chroot:+($debian_chroot)}\[\033[01;32m\]\u@\h\[\033[00m\]
:\[\033[01;34m\]\w\[\033[00m\]\$ '
unset color_prompt force_color_prompt

# enable color support of ls and also add handy aliases
if [ -x /usr/bin/dircolors ]; then
    test -r ~/.dircolors && eval "$(dircolors -b ~/.dircolors)" || eval
"$(dircolors -b)"
    alias ls='ls --color=auto'
    alias grep='grep --color=auto'
    alias fgrep='fgrep --color=auto'
    alias egrep='egrep --color=auto'
fi

[ -f ~/.aliases ] && . ~/.aliases
[ -f ~/.bashrc.local ] && . ~/.bashrc.local

if [ -f /etc/bash_completion ] && ! shopt -oq posix; then
    . /etc/bash_completion
fi

# run TMUX
[[ $- != *i* ]] && return

if which tmux >/dev/null 2>&1; then
    if [[ -z "$TMUX" ]] ;then
        ID="`tmux ls | grep -vml attached | cut -d: -f1`" # get the id
of a deattached session
```

```
    if [[ -z "$ID" ]] ;then # if not available create a new one
        tmux new-session
    else
        tmux attach-session -t "$ID" # if available attach to it
    fi
fi
### NOTHING BELOW ###
```

From:  
<https://wiki.csgalileo.org/> - **Galileo Labs**

Permanent link:  
<https://wiki.csgalileo.org/tips/ubuntu/locale?rev=1486462741>

Last update: **2017/02/07 11:19**

