Auto install pterodactyl

Just run this script

```bash
#!/bin/bash

set -e

##################################################################################
####################
#                                                #
# Project 'pterodactyl-installer'                                #
#                                                #
# Copyright (C) 2018 - 2024, Vilhelm Prytz, <vilhelm@prytznet.se>                #
#                                                #
#   This program is free software: you can redistribute it and/or modify         #
#   it under the terms of the GNU General Public License as published by          #
#   the Free Software Foundation, either version 3 of the License, or            #
#   (at your option) any later version.                                #
#                                                #
#   This program is distributed in the hope that it will be useful,              #
#   but WITHOUT ANY WARRANTY; without even the implied warranty of                #
#   MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the                 #
#   GNU General Public License for more details.                           #
#                                                #
#   You should have received a copy of the GNU General Public License             #
#   along with this program.  If not, see <https://www.gnu.org/licenses/>.         #
#                                                #
# https://github.com/pterodactyl-installer/pterodactyl-installer/blob/master/LICENSE #
#                                                #
# This script is not associated with the official Pterodactyl Project.            #
# https://github.com/pterodactyl-installer/pterodactyl-installer              #
#                                                #
##################################################################################
#################

export GITHUB_SOURCE="v1.1.1"
```

```bash
export SCRIPT_RELEASE="v1.1.1"
export GITHUB_BASE_URL="https://raw.githubusercontent.com/pterodactyl-installer/pterodactyl-installer"

LOG_PATH="/var/log/pterodactyl-installer.log"

# check for curl
if ! [ -x "$(command -v curl)" ]; then
  echo "* curl is required in order for this script to work."
  echo "* install using apt (Debian and derivatives) or yum/dnf (CentOS)"
  exit 1
fi

# Always remove lib.sh, before downloading it
[ -f /tmp/lib.sh ] && rm -rf /tmp/lib.sh
curl -sSL -o /tmp/lib.sh "$GITHUB_BASE_URL"/master/lib/lib.sh
# shellcheck source=lib/lib.sh
source /tmp/lib.sh

execute() {
  echo -e "\n\n* pterodactyl-installer $(date) \n\n" >>$LOG_PATH

  [[ "$1" == *"canary"* ]] && export GITHUB_SOURCE="master" && export SCRIPT_RELEASE="canary"
  update_lib_source
  run_ui "${1//_canary/}" |& tee -a $LOG_PATH

  if [[ -n $2 ]]; then
    echo -e -n "* Installation of $1 completed. Do you want to proceed to $2 installation? (y/N): "
    read -r CONFIRM
    if [[ "$CONFIRM" =~ [Yy] ]]; then
      execute "$2"
    else
      error "Installation of $2 aborted."
      exit 1
    fi
  fi
}

welcome ""

done=false
```

```
while [ "$done" == false ]; do
  options=(
    "Install the panel"
    "Install Wings"
    "Install both [0] and [1] on the same machine (wings script runs after panel)"
    # "Uninstall panel or wings\n"

    "Install panel with canary version of the script (the versions that lives in master, may be broken!)"
    "Install Wings with canary version of the script (the versions that lives in master, may be broken!)"
    "Install both [3] and [4] on the same machine (wings script runs after panel)"
    "Uninstall panel or wings with canary version of the script (the versions that lives in master, may be broken!)"
  )

  actions=(
    "panel"
    "wings"
    "panel;wings"
    # "uninstall"

    "panel_canary"
    "wings_canary"
    "panel_canary;wings_canary"
    "uninstall_canary"
  )

  output "What would you like to do?"

  for i in "${!options[@]}"; do
    output "[$i] ${options[$i]}"
  done

  echo -n "* Input 0-$((${#actions[@]} - 1)): "
  read -r action

  [ -z "$action" ] && error "Input is required" && continue

  valid_input=("$(for ((i = 0; i <= ${#actions[@]} - 1; i += 1)); do echo "${i}"; done)")
  [[ ! " ${valid_input[*]} " =~ ${action} ]] && error "Invalid option"
  [[ " ${valid_input[*]} " =~ ${action} ]] && done=true && IFS=";" read -r i1 i2 <<<"${actions[$action]}" &&
execute "$i1" "$i2"
```

```
    done


    # Remove lib.sh, so next time the script is run the, newest version is downloaded.

    rm -rf /tmp/lib.sh
```

Revision #1
Created 26 November 2024 20:43:34 by Nicolas
Updated 6 December 2024 01:01:33 by Nicolas