install copilot cli

```
#!/bin/bash
#valid on july 4 2026
# Colors for output
RED='\033[0;31m'
GREEN='\033[0;32m'
YELLOW='\033[1;33m'
NC='\033[0m' # No Color

# ──────────────────────────────────────────────────
# Detection helpers
# ──────────────────────────────────────────────────

CLI_EXISTS=false
CONFIG_EXISTS=false

detect_existing() {
    if [ -f /nzk/bin/copilot ]; then
        CLI_EXISTS=true
        echo -e "${GREEN}✓ Detected:${NC} copilot CLI at /nzk/bin/copilot"
    fi
    if [ -f /nzk/special/.deepseek_config ]; then
        CONFIG_EXISTS=true
        echo -e "${GREEN}✓ Detected:${NC} config files in /nzk/special/"
    fi
    if ! $CLI_EXISTS && ! $CONFIG_EXISTS; then
        echo -e "${YELLOW}□□  No existing installation detected.${NC}"
    fi
    echo ""
}

# ──────────────────────────────────────────────────
# Functions
# ──────────────────────────────────────────────────

install_copilot() {
```

```
    if $CLI_EXISTS; then
        echo -e "${YELLOW}⚠  Copilot CLI is already installed at /nzk/bin/copilot.${NC}"
        read -p "Reinstall? (y/N): " reinstall
        if [[ ! "$reinstall" =~ ^[Yy] ]]; then
            echo -e "${GREEN}  Skipping copilot install.${NC}"
            return 0
        fi
    fi

    echo -e "${YELLOW}Installing GitHub Copilot CLI...${NC}"
    curl -fsSL https://gh.io/copilot-install | bash

    echo -e "${YELLOW}Creating /nzk/bin directory...${NC}"
    mkdir -p /nzk/bin

    echo -e "${YELLOW}Moving copilot to /nzk/bin...${NC}"
    if [ -f ~/.local/bin/copilot ]; then
        mv ~/.local/bin/copilot /nzk/bin/copilot
    elif [ -f /usr/local/bin/copilot ]; then
        mv /usr/local/bin/copilot /nzk/bin/copilot
    else
        echo -e "${RED} Copilot binary not found after installation!${NC}"
        echo -e "${YELLOW}   Please manually move it to /nzk/bin/copilot${NC}"
        return 1
    fi
    chmod +x /nzk/bin/copilot
    echo -e "${GREEN} Copilot installed at /nzk/bin/copilot${NC}"
}

prompt_for_key() {
    local label="$1"
    local key_var
    echo -e "${YELLOW}Enter your API key${label}:${NC}"
    read -s key_var
    echo ""
    echo "$key_var"
}

# ── Choose provider and set env vars ──
choose_provider() {
```

```
echo ""
echo "Select a provider:"
echo "  1) DeepSeek    (deepseek-v4-pro / deepseek-v4-flash)"
echo "  2) OpenAI      (gpt-4o, o3, etc.)"
echo "  3) Anthropic   (claude-sonnet-4-20250514, etc.)"
echo "  4) Google      (gemini-2.5-pro, etc.)"
echo "  5) Custom      (enter your own)"
echo ""
read -p "Enter your choice [1-5]: " prov_choice

case $prov_choice in
    1)
        PROVIDER_TYPE="anthropic"
        PROVIDER_BASE_URL="https://api.deepseek.com/anthropic"
        PROVIDER_LABEL="DeepSeek"
        PROVIDER_DEFAULT_MODEL="deepseek-v4-pro"
        PROVIDER_FLASH_MODEL="deepseek-v4-flash"
        SUPPORTS_DUAL=true
        ;;
    2)
        PROVIDER_TYPE="openai"
        PROVIDER_BASE_URL="https://api.openai.com/v1"
        PROVIDER_LABEL="OpenAI"
        PROVIDER_DEFAULT_MODEL="gpt-4o"
        SUPPORTS_DUAL=false
        ;;
    3)
        PROVIDER_TYPE="anthropic"
        PROVIDER_BASE_URL="https://api.anthropic.com/v1"
        PROVIDER_LABEL="Anthropic"
        PROVIDER_DEFAULT_MODEL="claude-sonnet-4-20250514"
        SUPPORTS_DUAL=false
        ;;
    4)
        PROVIDER_TYPE="google"
        PROVIDER_BASE_URL="https://generativelanguage.googleapis.com/v1/openai"
        PROVIDER_LABEL="Google"
        PROVIDER_DEFAULT_MODEL="gemini-2.5-pro"
        SUPPORTS_DUAL=false
        ;;
```

```
        5)
            PROVIDER_LABEL="Custom"
            SUPPORTS_DUAL=false
            read -p "Enter provider type (e.g. anthropic, openai): " PROVIDER_TYPE
            read -p "Enter base URL (e.g. https://api.example.com/v1): " PROVIDER_BASE_URL
            read -p "Enter default model name: " PROVIDER_DEFAULT_MODEL
            ;;
        *)
            echo -e "${RED}  Invalid choice. Defaulting to DeepSeek.${NC}"
            PROVIDER_TYPE="anthropic"
            PROVIDER_BASE_URL="https://api.deepseek.com/anthropic"
            PROVIDER_LABEL="DeepSeek"
            PROVIDER_DEFAULT_MODEL="deepseek-v4-pro"
            PROVIDER_FLASH_MODEL="deepseek-v4-flash"
            SUPPORTS_DUAL=true
            ;;
    esac

    # For non-DeepSeek, ask if they want a secondary model too
    if [ "$SUPPORTS_DUAL" != "true" ]; then
        echo ""
        echo -e "${YELLOW}  Do you want to set up a second model with a different key?
(y/N):${NC}"
        read -r want_second
        if [[ "$want_second" =~ ^[Yy] ]]; then
            SUPPORTS_DUAL=true
            read -p "Enter second model name (e.g. gpt-4o-mini, claude-haiku): "
PROVIDER_FLASH_MODEL
        fi
    fi

    echo -e "${GREEN}  Provider set to ${PROVIDER_LABEL} (${PROVIDER_TYPE})${NC}"
}

# ── Update API keys (and optionally provider) inside existing config files ──
update_keys_only() {
    local pro_config="/nzk/special/.deepseek_config"
    local flash_config="/nzk/special/.deepseek_config_flash"

    if [ ! -f "$pro_config" ] && [ ! -f "$flash_config" ]; then
```

```
        echo -e "${RED}□ No existing config files found. Running full setup instead.${NC}"
        setup_api_keys
        return
    fi

    # Ask if they want to change provider too
    echo -e "${YELLOW}□ Change provider (DeepSeek/OpenAI/Anthropic/etc.)? (y/N):${NC}"
    read -r change_prov
    if [[ "$change_prov" =~ ^[Yy] ]]; then
        choose_provider
    else
        # Keep existing — try to read from current config files
        if [ -f "$pro_config" ]; then
            PROVIDER_TYPE=$(grep "^export COPILOT_PROVIDER_TYPE=" "$pro_config" | sed
's/^export COPILOT_PROVIDER_TYPE=//')
            PROVIDER_BASE_URL=$(grep "^export COPILOT_PROVIDER_BASE_URL=" "$pro_config" | sed
's/^export COPILOT_PROVIDER_BASE_URL=//')
            PROVIDER_DEFAULT_MODEL=$(grep "^export COPILOT_MODEL=" "$pro_config" | sed
's/^export COPILOT_MODEL=//')
            PROVIDER_LABEL="${PROVIDER_TYPE}"
        fi
        if [ -f "$flash_config" ]; then
            PROVIDER_FLASH_MODEL=$(grep "^export COPILOT_MODEL=" "$flash_config" | sed
's/^export COPILOT_MODEL=//')
        fi
    fi

    echo -e "${YELLOW}□□Enter your new API key for ${PROVIDER_DEFAULT_MODEL:-
${PROVIDER_LABEL}}:${NC}"
    read -s PRO_KEY
    echo ""
    if [ -z "$PRO_KEY" ]; then
        echo -e "${RED}□ No key entered. Aborting.${NC}"
        return 1
    fi

    # If we have a flash/second model, ask about key
    if [ -n "${PROVIDER_FLASH_MODEL:-}" ]; then
        echo -e "${YELLOW}□ Use the same key for ${PROVIDER_FLASH_MODEL}? (Y/n):${NC}"
        read -r use_same
```

```
        if [[ "$use_same" =~ ^[Nn] ]]; then
            echo -e "${YELLOW}□□Enter your API key for ${PROVIDER_FLASH_MODEL}:${NC}"
            read -s FLASH_KEY
            echo ""
            if [ -z "$FLASH_KEY" ]; then
                echo -e "${YELLOW}⚠□  No key entered, using the same key for both.${NC}"
                FLASH_KEY="$PRO_KEY"
            fi
        else
            FLASH_KEY="$PRO_KEY"
        fi
    else
        FLASH_KEY="$PRO_KEY"
    fi

    echo -e "${YELLOW}□□Updating config files...${NC}"

    # Helper: set or replace a variable in a config file
    set_config_var() {
        local file="$1" var="$2" val="$3"
        if grep -q "^export ${var}=" "$file"; then
            sed -i "s|^export ${var}=.*|export ${var}=${val}|" "$file"
        else
            echo "export ${var}=${val}" >> "$file"
        fi
    }

    # Update/regenerate pro config
    cat > "$pro_config" << EOF
# Copilot Configuration
# Generated: $(date)
# Provider: ${PROVIDER_LABEL}

export COPILOT_PROVIDER_TYPE=${PROVIDER_TYPE}
export COPILOT_PROVIDER_BASE_URL=${PROVIDER_BASE_URL}
export COPILOT_PROVIDER_API_KEY=${PRO_KEY}
export COPILOT_MODEL=${PROVIDER_DEFAULT_MODEL}
EOF
    echo -e "${GREEN}  ✓ Updated ${pro_config}${NC}"
```

```
    # Update/regenerate flash/secondary config
    if [ -n "${PROVIDER_FLASH_MODEL:-}" ]; then
        cat > "$flash_config" << EOF
# Copilot Configuration — secondary model
# Generated: $(date)
# Provider: ${PROVIDER_LABEL}

export COPILOT_PROVIDER_TYPE=${PROVIDER_TYPE}
export COPILOT_PROVIDER_BASE_URL=${PROVIDER_BASE_URL}
export COPILOT_PROVIDER_API_KEY=${FLASH_KEY}
export COPILOT_MODEL=${PROVIDER_FLASH_MODEL}
EOF
        echo -e "${GREEN}  ✓ Updated ${flash_config}${NC}"
    fi

    # Update aliases to reflect current provider label
    cat > /nzk/special/.deepseek_aliases << ALIASEOF
# Copilot aliases — ${PROVIDER_LABEL}
alias copilot-default='source /nzk/special/.deepseek_config && copilot'
alias copilot-alt='source /nzk/special/.deepseek_config_flash && copilot'
ALIASEOF
    echo -e "${GREEN}  ✓ Updated aliases${NC}"

    chmod 600 /nzk/special/.deepseek_config 2>/dev/null
    chmod 600 /nzk/special/.deepseek_config_flash 2>/dev/null
    chmod 600 /nzk/special/.deepseek_aliases 2>/dev/null

    echo -e "${GREEN} Keys and config updated.${NC}"
}

setup_api_keys() {
    mkdir -p /nzk/special

    # If config already exists, ask what to do
    if $CONFIG_EXISTS; then
        echo ""
        echo -e "${YELLOW}⚠  Config files already exist in /nzk/special/${NC}"
        echo "  1) Overwrite all config files (fresh setup)"
        echo "  2) Update API keys only (keep everything else)"
        echo "  3) Skip — don't touch config files"
```

```
    echo ""
    read -p "Enter your choice [1-3]: " cfg_choice
    case $cfg_choice in
        1)
            : # fall through to full key setup below
            ;;
        2)
            update_keys_only
            return $?
            ;;
        *)
            echo -e "${GREEN}□□  Skipping config file setup.${NC}"
            return 0
            ;;
    esac
fi

# Choose provider
choose_provider

# Prompt for primary key
echo ""
PRO_KEY=$(prompt_for_key " for ${PROVIDER_DEFAULT_MODEL}")
if [ -z "$PRO_KEY" ]; then
    echo -e "${RED}□ No key entered. Aborting.${NC}"
    return 1
fi

# Ask about second model key
if [ "$SUPPORTS_DUAL" = "true" ] && [ -n "${PROVIDER_FLASH_MODEL:-}" ]; then
    echo -e "${YELLOW}□ Use the same API key for ${PROVIDER_FLASH_MODEL}? (Y/n):${NC}"
    read -r use_same
    if [[ "$use_same" =~ ^[Nn] ]]; then
        FLASH_KEY=$(prompt_for_key " for ${PROVIDER_FLASH_MODEL}")
        if [ -z "$FLASH_KEY" ]; then
            echo -e "${YELLOW}⚠□  No key entered, using the same key for both.${NC}"
            FLASH_KEY="$PRO_KEY"
        fi
    else
        FLASH_KEY="$PRO_KEY"
```

```
        fi
    else
        FLASH_KEY="$PRO_KEY"
    fi

    echo -e "${YELLOW}□□Creating configuration files...${NC}"

    # ── Main config ──
    cat > /nzk/special/.deepseek_config << EOF
# Copilot Configuration — ${PROVIDER_LABEL}
# Generated: $(date)

export COPILOT_PROVIDER_TYPE=${PROVIDER_TYPE}
export COPILOT_PROVIDER_BASE_URL=${PROVIDER_BASE_URL}
export COPILOT_PROVIDER_API_KEY=${PRO_KEY}
export COPILOT_MODEL=${PROVIDER_DEFAULT_MODEL}
EOF

    # ── Secondary model config ──
    if [ -n "${PROVIDER_FLASH_MODEL:-}" ]; then
        cat > /nzk/special/.deepseek_config_flash << EOF
# Copilot Configuration — ${PROVIDER_LABEL} (secondary)
# Generated: $(date)

export COPILOT_PROVIDER_TYPE=${PROVIDER_TYPE}
export COPILOT_PROVIDER_BASE_URL=${PROVIDER_BASE_URL}
export COPILOT_PROVIDER_API_KEY=${FLASH_KEY}
export COPILOT_MODEL=${PROVIDER_FLASH_MODEL}
EOF
    fi

    # ── Aliases ──
    cat > /nzk/special/.deepseek_aliases << ALIASEOF
# Copilot aliases — ${PROVIDER_LABEL}
alias copilot-default='source /nzk/special/.deepseek_config && copilot'
alias copilot-alt='source /nzk/special/.deepseek_config_flash && copilot'
ALIASEOF

    chmod 600 /nzk/special/.deepseek_config
    chmod 600 /nzk/special/.deepseek_config_flash 2>/dev/null
```

```
    chmod 600 /nzk/special/.deepseek_aliases

    echo -e "${GREEN}✅ API keys and config files created.${NC}"
}

setup_shell_config() {
    echo -e "${YELLOW}⚙️  Adding to shell config...${NC}"
    SHELL_CONFIG=""
    if [ -f ~/.zshrc ]; then
        SHELL_CONFIG=~/.zshrc
    elif [ -f ~/.bashrc ]; then
        SHELL_CONFIG=~/.bashrc
    else
        echo -e "${RED}❌ No .zshrc or .bashrc found!${NC}"
        return 1
    fi

    cp "${SHELL_CONFIG}" "${SHELL_CONFIG}.backup"

    if ! grep -q "/nzk/special/.deepseek_config" "${SHELL_CONFIG}"; then
        cat >> "${SHELL_CONFIG}" << 'EOF'

# DeepSeek Copilot Configuration
export PATH="/nzk/bin:/nzk/appimages:$PATH"
source /nzk/special/.deepseek_config
source /nzk/special/.deepseek_aliases
EOF
        echo -e "${GREEN}✅ Added to ${SHELL_CONFIG}${NC}"
    else
        echo -e "${YELLOW}⚠️  Already configured in ${SHELL_CONFIG}${NC}"
    fi
}

create_readme() {
    cat > /nzk/special/README.md << EOF
# Copilot Configuration — ${PROVIDER_LABEL:-Current Provider}

## Models
| Alias              | Model                              | Config File                         |
|--------------------|------------------------------------|-------------------------------------|
```

```
| \`copilot-default\` | ${PROVIDER_DEFAULT_MODEL:-default} |
\`.deepseek_config\`                 |
| \`copilot-alt\`     | ${PROVIDER_FLASH_MODEL:-N/A}       |
\`.deepseek_config_flash\`           |

## Usage
- \`copilot\`         — Run with default model
- \`copilot-default\` — Explicitly use default model
- \`copilot-alt\`     — Use secondary model (if configured)

## Switch models manually
\`\`\`bash
source /nzk/special/.deepseek_config        # default model
source /nzk/special/.deepseek_config_flash  # secondary model
\`\`\`

## Environment Variables
- \`COPILOT_PROVIDER_TYPE\`: ${PROVIDER_TYPE:-anthropic}
- \`COPILOT_PROVIDER_BASE_URL\`: ${PROVIDER_BASE_URL:-...}
- \`COPILOT_PROVIDER_API_KEY\`: Your API key
- \`COPILOT_MODEL\`: Model name

## Files
- \`.deepseek_config\`        — Default model config + API key
- \`.deepseek_config_flash\`  — Secondary model config + API key
- \`.deepseek_aliases\`       — Convenience aliases
EOF
    echo -e "${GREEN}□ README created.${NC}"
}

print_complete() {
    echo -e "${GREEN}========================================${NC}"
    echo -e "${GREEN}□ All done!${NC}"
    echo -e "${GREEN}========================================${NC}"
    echo -e "${YELLOW}□□Next steps:${NC}"
    echo -e "1. Source your shell config: ${GREEN}source ${SHELL_CONFIG}${NC}"
    echo -e "   OR open a new terminal"
    echo -e ""
    echo -e "2. Test: ${GREEN}copilot-default${NC} or ${GREEN}copilot-alt${NC}"
    echo -e ""
```

```
    echo -e "${YELLOW}□□Files:${NC}"
    echo -e "   - /nzk/bin/copilot"
    echo -e "   - /nzk/special/.deepseek_config"
    echo -e "   - /nzk/special/.deepseek_config_flash"
    echo -e "   - /nzk/special/.deepseek_aliases"
    echo -e "   - /nzk/special/README.md"
    echo -e ""
    echo -e "${RED}□□API keys are stored in /nzk/special/.deepseek_config*${NC}"
    echo -e "${RED}   Make sure to keep these files secure!${NC}"
}

# ─────────────────────────────────────────────
# Main Menu
# ─────────────────────────────────────────────

echo -e "${NC}"
echo -e "${GREEN}╔═══════════════════════════════════════════╗${NC}"
echo -e "${GREEN}║     GitHub Copilot CLI Configurator       ║${NC}"
echo -e "${GREEN}╚═══════════════════════════════════════════╝${NC}"
echo ""

# Detect existing installation
detect_existing

echo "What do you want to do?"
echo "  1) Install copilot CLI only"
echo "  2) Full API key & provider setup (DeepSeek, OpenAI, Anthropic, etc.)"
echo "  3) Both — install CLI + full setup"
echo "  4) Update keys / provider in existing config only"
echo ""

read -p "Enter your choice [1-4]: " choice

case $choice in
    1)
        install_copilot
        ;;
    2)
        setup_api_keys
        setup_shell_config
```

```
        create_readme
        print_complete
        ;;
    3)
        install_copilot
        setup_api_keys
        setup_shell_config
        create_readme
        print_complete
        ;;
    4)
        update_keys_only
        echo -e "${GREEN}□ Key update complete.${NC}"
        ;;
    *)
        echo -e "${RED}□ Invalid choice. Please run again and select 1-4.${NC}"
        exit 1
        ;;
esac
```

Revision #1
Created 2026-07-04 17:50:03 UTC by naruzkurai
Updated 2026-07-04 17:51:01 UTC by naruzkurai