Automating Linux Installation with Anaconda and Kickstart

A comprehensive guide to streamlining and standardising your Linux deployment process across multiple systems

What is Anaconda?

Anaconda is the sophisticated installation framework behind some of the world's most widely deployed Linux distributions.

Default Installer

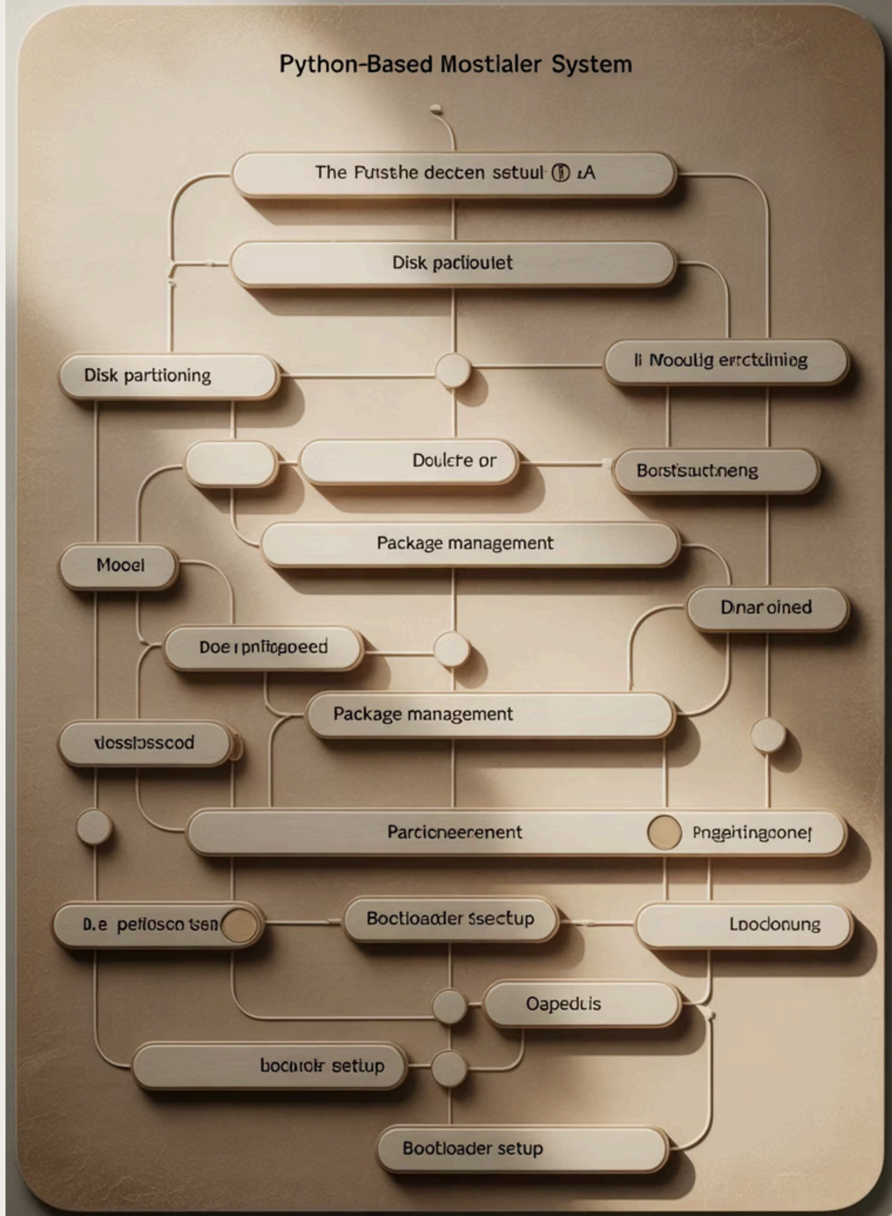
Powers installation for Fedora, Red Hat Enterprise Linux (RHEL), CentOS, and their derivatives

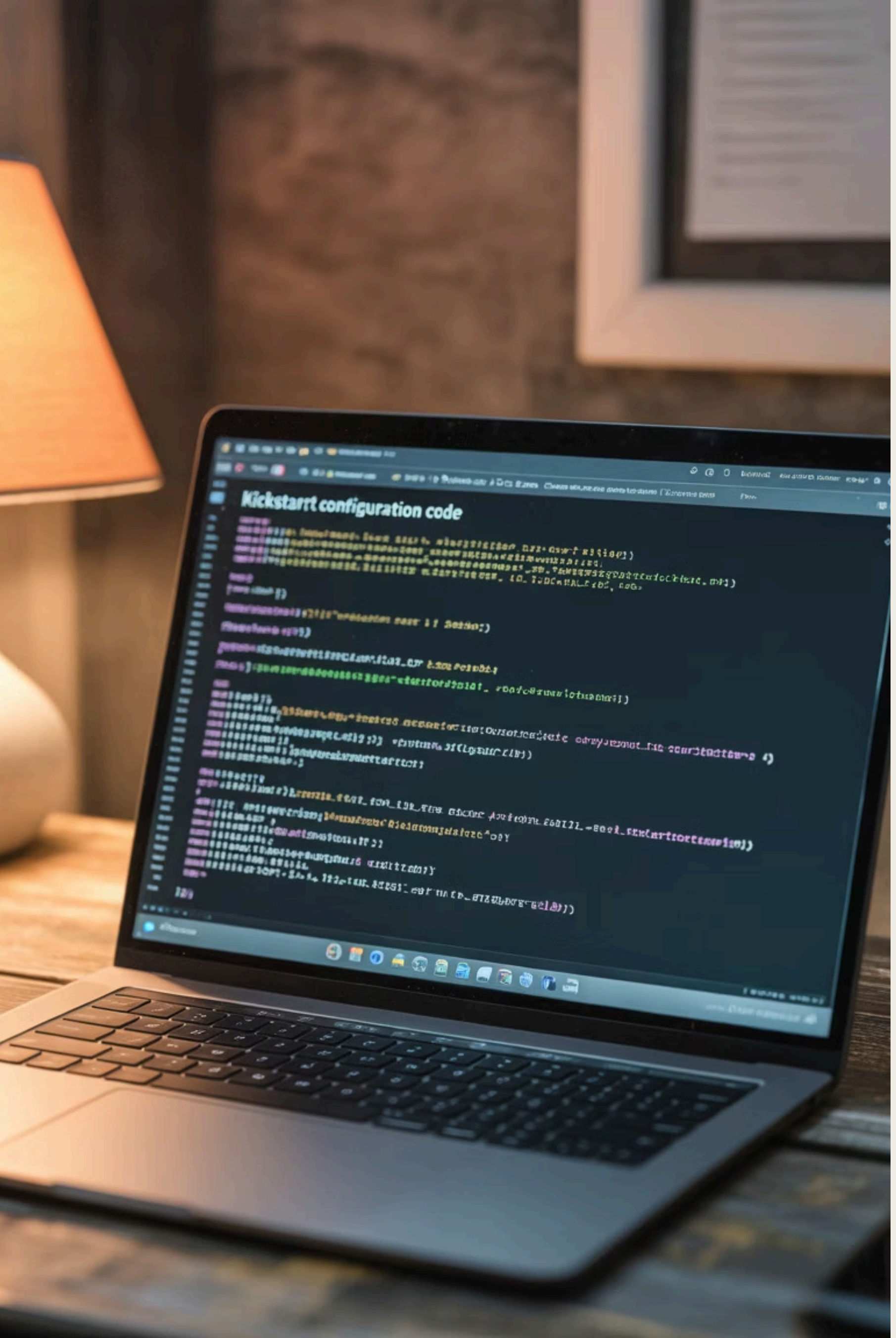
Modular Architecture

Python-based system handling disk setup, package installation, bootloader configuration, and comprehensive system setup

Multiple Interfaces

Supports graphical UI, text mode (TUI), and fully automated Kickstart installations for enterprise deployments





Kickstart Installation: The Basics

Kickstart transforms the installation process from an interactive experience to a fully automated deployment.

1 Automation Foundation

Kickstart files are plain text files containing predefined answers to all installation questions, eliminating manual intervention

2 Enterprise Scalability

Enables consistent, repeatable installations across dozens, hundreds or thousands of machines with identical configurations

3 Structured Format

Contains distinct sections for commands, package selection lists, and pre/post-installation scripts that execute at specific points

Creating a Kickstart File

The most straightforward approach to building a Kickstart file is to leverage an existing installation as your template.

Perform a Manual Installation

Complete a standard installation with your desired configuration choices

Locate Generated File

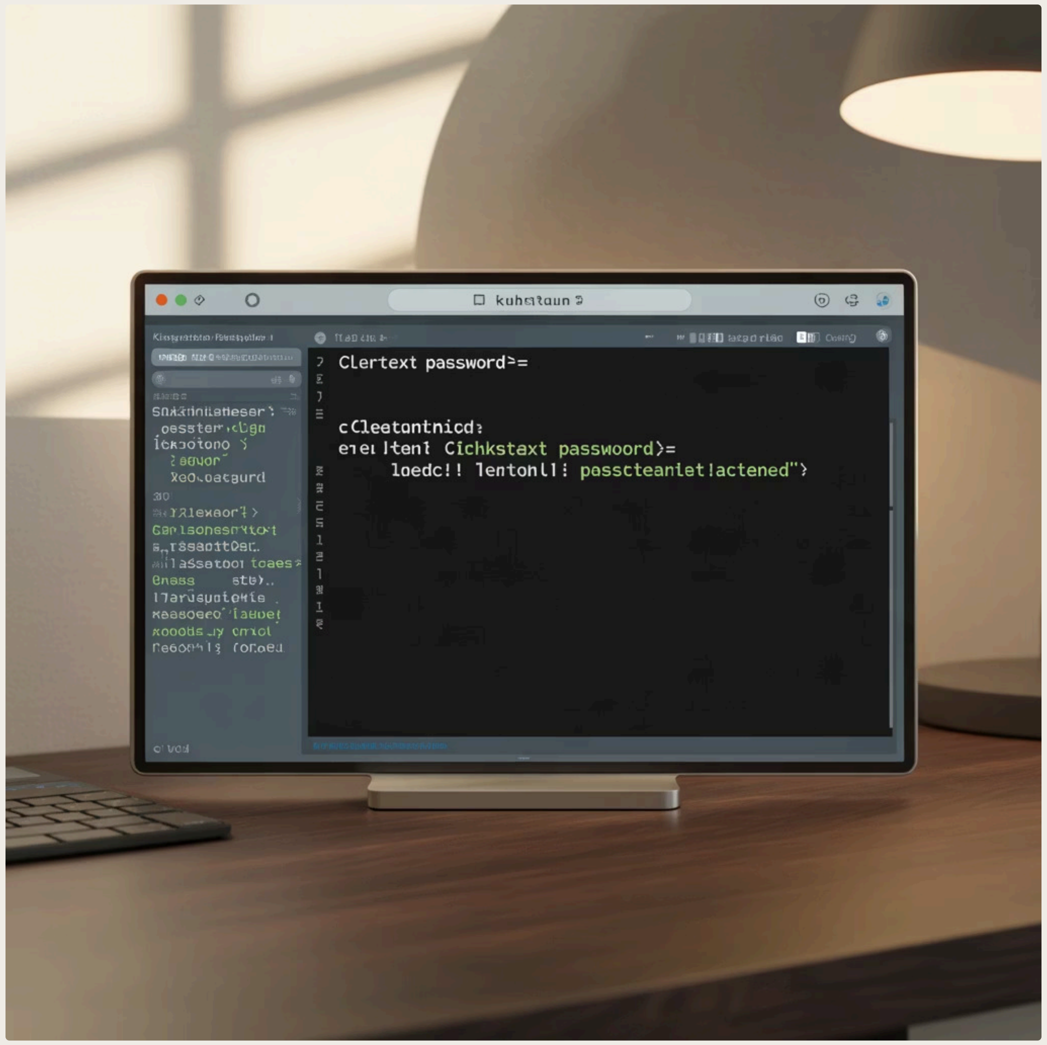
Anaconda automatically saves your choices as `/root/anaconda-ks.cfg` on the newly installed system

Customise for Automation

Edit this file to refine partitioning, networking, user creation, and other settings for your deployment needs

Editing Tips

- Use any standard text editor to modify the file
- Comments start with # for documentation
- Sections must appear in the correct order
- Most interactive prompts can be disabled with appropriate settings



Making Kickstart Available & Starting Installation

For successful automated installation, the system needs access to your Kickstart file during boot.

Prepare Kickstart File

Place your completed Kickstart file in an accessible location:

- USB drive or other removable media
- Local hard drive partition
- Network server (HTTP, FTP, NFS)

Configure Boot Parameters

Boot the installer with the `inst.ks=` kernel option pointing to your file location

Installation Begins

Anaconda reads the Kickstart file and automatically proceeds with installation using your predefined settings

```
Example boot command: linux inst.ks=http://myserver.example.com/kickstart/rhel8-base.cfg
```

Key Kickstart Sections & Syntax Highlights

Command Section

```
# Installation source
url --url=http://mirror.example.com/rhel/8/BaseOS/x86_64/os/

# Root password (hashed)
rootpw --iscrypted $6$qEuJ.../4.k$MGvH4Nt3eJU/

# Network configuration
network --bootproto=dhcp --device=eth0 --activate

# Partitioning
clearpart --all --initlabel
part /boot --fstype=xfs --size=1024
part pv.01 --size=40000
volgroup vg_root pv.01
logvol / --vgname=vg_root --size=35000 --name=lv_root
```

Package & Script Sections

```
# Package selection
%packages
@core
@base
chromy
wget
vim
%end

# Post-installation script
%post
echo "Installation complete!" > /root/install.log
systemctl enable httpd
firewall-cmd --permanent --add-service=http
%end
```

Always end each section with `%end` to properly delimit script sections

Validating Kickstart Files

Prevent deployment failures by validating your Kickstart files before using them in production environments.

1

Install Validation Tool

The ksvalidator utility is part of the pykickstart package

```
sudo dnf install pykickstart
```

2

Run Validation Check

Execute the command against your Kickstart file

```
ksvalidator /path/to/kickstart.ks
```

3

Review & Fix Issues

Address any syntax errors or incompatible options reported by the validator

No output typically means the file passed validation successfully

Anaconda Boot Options for Kickstart

Anaconda supports numerous boot options to customise the installation source and process. These options are specified at the boot prompt or in bootloader configuration.

inst.repo=

Specifies the installation source location. Can be a CDROM, HTTP server, NFS share, or local ISO file

```
inst.repo=http://mirror.example.com/rhel/8/os/x86_64/
```

inst.ks=

Points to the location of your Kickstart file. Supports multiple protocols and local paths

```
inst.ks=nfs:192.168.0.100:/exports/kickstart/ks.cfg
```

inst.addrepo=

Adds additional repositories during installation for accessing extra packages

```
inst.addrepo=updates,http://updates.example.com/rhel8
```

ip=

Configures the network interface for installations requiring network access

```
ip=192.168.0.10::192.168.0.1:255.255.255.0:myhost:ens3:none
```

Real-World Example: Sample Kickstart Snippet

```
# RHEL 8 Server Kickstart Template
# Basic system configuration
keyboard --vckeymap=uk --xlayouts='gb'
lang en_GB.UTF-8
timezone Europe/London --utc

# Installation source
url --url="https://mirror.example.com/rhel/8/BaseOS/x86_64/os/"

# Disk partitioning
clearpart --all --initlabel
part /boot --fstype=xfs --size=1024
part /boot/efi --fstype=efi --size=600
part pv.01 --size=40000 --grow
volgroup vg_root pv.01
logvol / --vgname=vg_root --size=10000 --name=lv_root
logvol /var --vgname=vg_root --size=5000 --name=lv_var
logvol swap --vgname=vg_root --size=4000 --name=lv_swap
```

```
# User configuration
rootpw --iscrypted $6$4xM...UO1
user --name=sysadmin --groups=wheel --iscrypted --
password=$6$RT...UZ/

# Network configuration
network --bootproto=static --device=ens3 --gateway=192.168.1.1 --
ip=192.168.1.200 --nameserver=192.168.1.10 --
netmask=255.255.255.0 --noipv6 --activate --
hostname=server01.example.com

# Package selection
%packages
@core
@base
vim
tmux
git
ansible
%end

# Post-installation script
%post
echo "Installation completed at $(date)" > /root/install-
complete.log
%end
```

This example demonstrates a production-ready Kickstart file for deploying RHEL 8 servers with static networking, LVM partitioning, and essential administration tools. It can be used as a starting template for your own deployments.