

Custom iPXE

Jem Camba - 2019-03-06 - in Infrastructure

How to install a custom Operating System or modified image.

Introduction

Packet supports passing custom iPXE scripts during provisioning, which allows you to install a custom operating system manually, or via automated kickstart.

Since we don't test any manually installed operating system, it goes without saying that you'll need to be familiar with our [S.O.S. and rescue mode](#) services in case you need to troubleshoot your server.

Step-by-Step Usage

Select the "Custom iPXE" operating system from the portal, or the custom_ipxe slug if using the API.

The screenshot shows the 'Configure your servers' form in the Packet portal. The form is titled 'Configure your servers:' and has a 'New' button in the top right corner. Below the title, there are five main sections: Hostname, Type, OS, Location, and Options. The Hostname field contains 'custom-ipxe'. The Type dropdown is set to 'Type 1 - \$0.40/hour'. The OS dropdown is set to 'Custom iPXE'. The Location dropdown is set to 'Parsippany, NJ'. The Options section has a 'MANAGE' button and a plus icon. Below these fields, there is a summary bar with a yellow background. It shows 'DEPLOYING TO' with a link to 'Provisioning Tests', 'EXPECTED COST' of '\$0.40 / hour', and a 'Deploy' button.

If you have your iPXE script hosted at a publicly accessible http(s) location, put the URL to your script in the text field, or use the ipxe_script_url API parameter. When we serve up iPXE during the boot process we will chain-load your iPXE script URL.

Should the device fail during ipxe boot, with your device set to persistently boot from iPXE you can edit your iPXE URL reboot the device to try again.

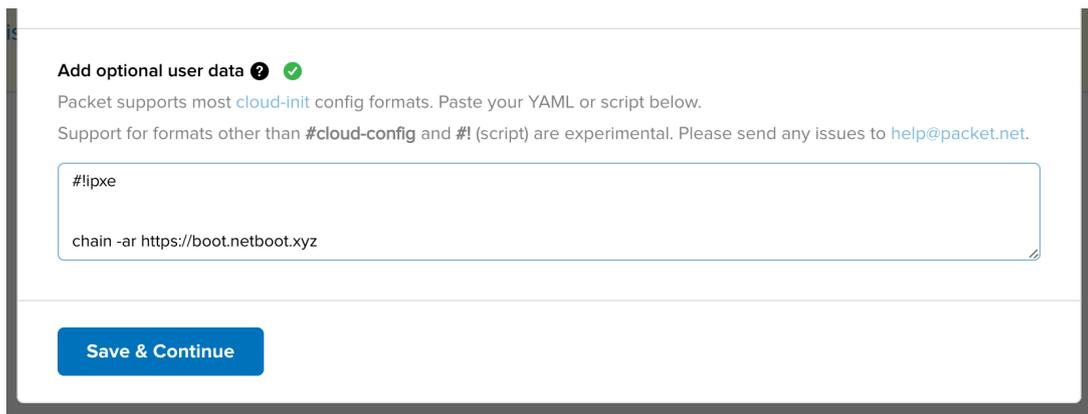
IPXE Script URL

https://boot.netboot.xyz|

Save

Cancel

Alternatively, you can pass the contents of your iPXE script directly via user-data with the `#!ipxe` hashbang. User data is also a feature in which you can edit should your provisioning with iPXE fail.



Add optional user data ? ✓

Packet supports most [cloud-init](#) config formats. Paste your YAML or script below.

Support for formats other than `#cloud-config` and `#!` (script) are experimental. Please send any issues to help@packet.net.

```
#!ipxe

chain -ar https://boot.netboot.xyz
```

Save & Continue

After serving up iPXE via DHCP, the device will be marked as active in our API and portal. Since the server is sitting on the Bootloader Options and it has no ssh access, we will have to use our [Console Access](#).

ssh device-id@sos.facility-code.packet.net

Netboot.xyz

If you're using netboot.xyz to manually install your operating system, once you connect to our S.O.S. service, you will see the following:

```
netboot.xyz

Default:
  Boot from local hdd (280)
Distributions:
  Linux Installs
  BSD Installs
  FreeDOS
  Hypervisor Installs
  Live Boot
  Security Related
  Windows
Tools:
  Utilities
  Architecture: x86_64 (64bit)
  iPXE shell
  Network card info
Signature Checks:
  netboot.xyz [ enabled: true ]
  Images [ enabled: true ]
```

For more information about each of the OSes, you can go to netboot.xyz

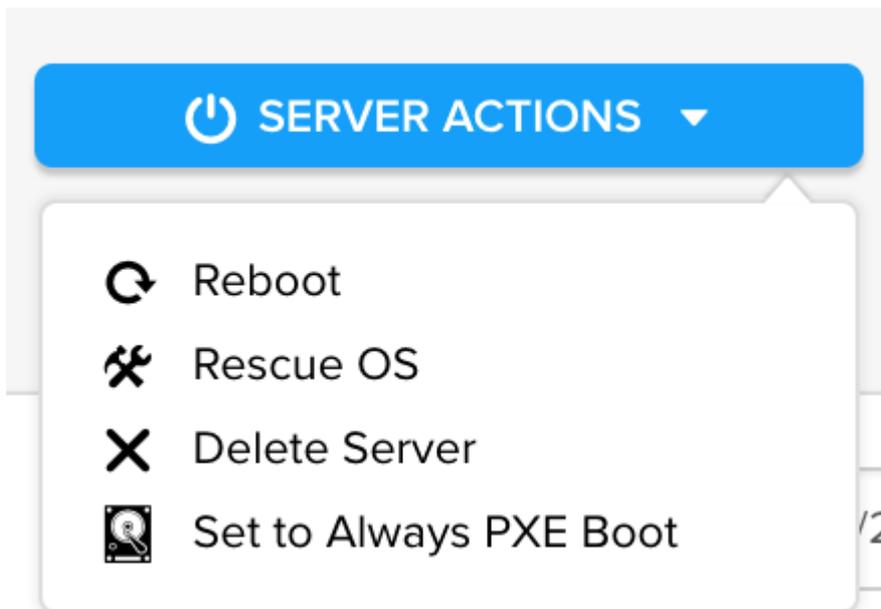
If the operating System is not listed there, you can install via ISO by selecting the iPXE shell option and enter the following;

```
kernel https://boot.netboot.xyz/memdisk iso raw
initrd http://url/to/iso
boot
```

If it fails during initramfs trying to load the CD device, update the install media to look for install media via the memdisk. More information can be found about this issue [here](#).

Persisting PXE

When provisioning the Custom iPXE operating system kicks off, we set the next boot option to PXE on first boot. By default, this PXE process only happens once on the first boot. To set your device to continuously boot to iPXE first, you can edit it under 'server actions' through the customer portal.



If true, PXE will persist as the first boot option past initial provisioning reboots. This is great for testing your iPXE provisioning script and lays the foundation for future, "always-pxe-based OS's" on Packet.

Custom iPXE Usage Notes

- If you would like to interact with your device via S.O.S. to perform a manual install and are not using netboot.xyz, our x86 servers require `console=ttyS1,115200n8`, and our aarch64 servers require `console=ttyAMA0,115200`.
- DHCP is available during a Custom iPXE device's entire life, so you can get network configured via DHCP and then setup networking statically in the OS by discovering the IP address information from our ec2-style metadata service: <https://metadata.packet.net/metadata> from the host.
- We load up our own iPXE build which chain-loads a Packet-managed `/auto.ipxe` script that will serve up either the chain-loaded iPXE script URL that you specify or the `#!ipxe` script if you've passed in your script directly via `userdata`.
- We set the variable `set ipxe_cloud_config packet` prior to any chain-loading or running your iPXE script. You can use this to perform Packet-specific iPXE commands if you want to maintain a unified iPXE script.

Please Note: If you chain-load your own iPXE build/version, you'll likely lose the `ipxe_cloud_config` variable.

Tags
iPXE