This paper provides a method to produce a custom GRR server ISO. The ISO is used in conjunction with ESXi and Ansible for offline (no Internet access) deployments. The primary goal is to create a preconfigured ISO that bypasses all Ubuntu 18.04 installation steps requiring manual input. The ISO will also have the GRR server Debian package, and all APT packages installed required for the GRR installation and configuration process and packages needed for Ansible automation. This process uses a Ubuntu 18.04 desktop and Cubic ISO customization application to customize the GRR ISO.

After installing a Ubuntu 18.04 desktop, use apt to update the apt repository and install the required cubic files.

```
$ sudo apt-add-repository ppa:cubic-wizard/release
$ sudo apt update
$ sudo apt install cubic
$ sudo apt install ssh
$ sudo apt-get install system-config-kickstart
$ sudo apt install vim
```

After downloading the appropriate Ubuntu 18.04 ISO, Run cubic.

```
$ cubic
```

The Cubic application will open. Select a directory where you want to save the customized ISO (Figure 1). Click 'Next.'

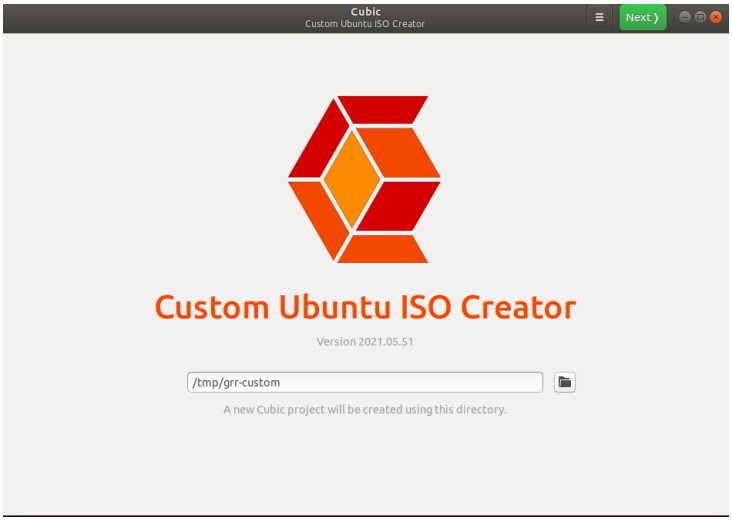

Figure 1

Next, we want to select our standard Ubuntu ISO to customize. In Figure 2, we see the screen for choosing the filename of the Ubuntu ISO under 'Original Disk…'. Once we do this, the remainder of the fields will automatically populate. Under 'Custom Disk…', change the custom ISO to a new name. Here, I just add 'grr' to the custom ISO name. Select 'Next' once complete.

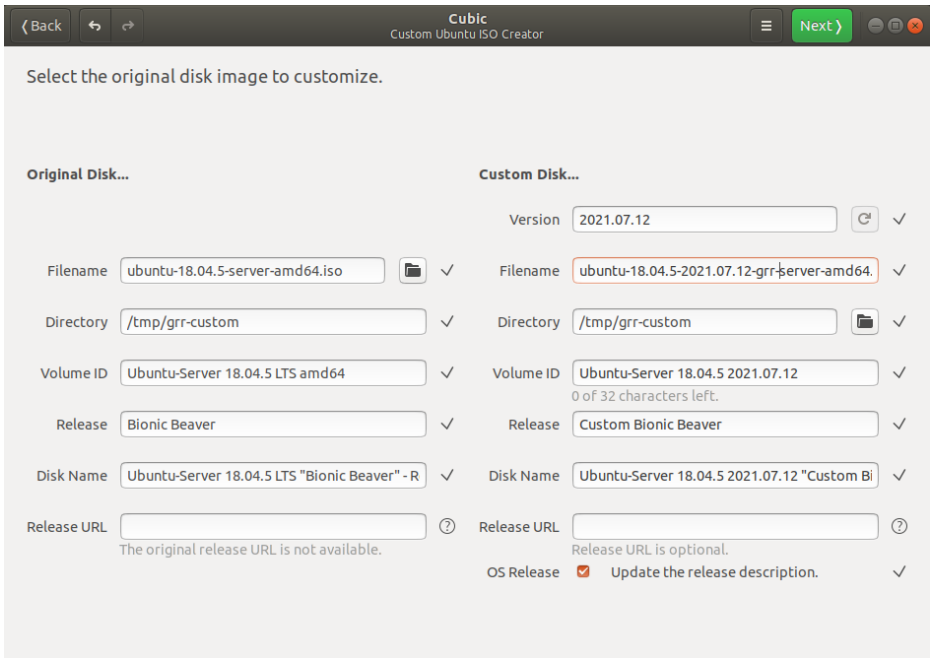

Figure 2

Cubic will automatically process the ISO for further customization. Click 'Next'. In Figure 3, we are brought to a virtual environment where we can add files and apt packages to the custom ISO. We will want to add the GRR server package and packages to support Ansible and the NGINX SSL proxy.

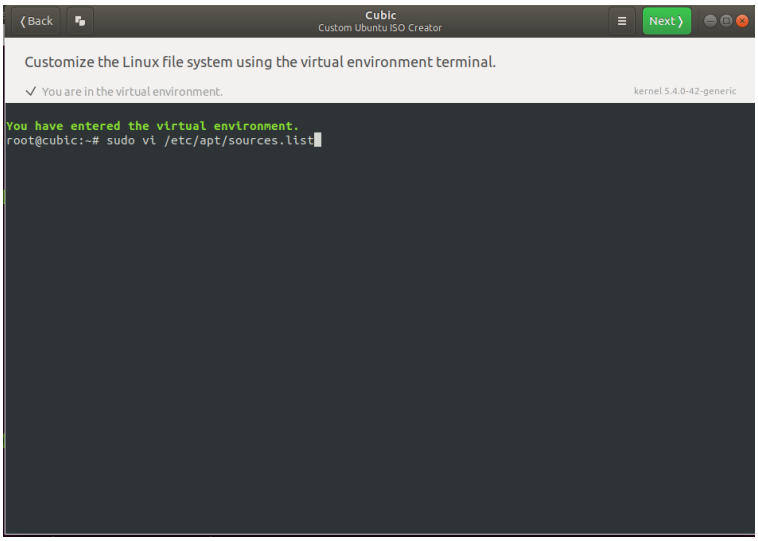

Figure 3

The first thing we want to do at the virtual environment command-line is repairing the APT repository sources list.

```
# vi /etc/apt/sources.list
```

We want to change the file. Delete everything in the file and add the following.

```
deb http://us.archive.ubuntu.com/ubuntu bionic main restricted
deb http://us.archive.ubuntu.com/ubuntu bionic-updates main restricted
deb http://us.archive.ubuntu.com/ubuntu bionic universe
deb http://us.archive.ubuntu.com/ubuntu bionic-updates universe
deb http://us.archive.ubuntu.com/ubuntu bionic multiverse
deb http://us.archive.ubuntu.com/ubuntu bionic-updates multiverse
deb http://us.archive.ubuntu.com/ubuntu bionic-backports main restricted universe multiverse
deb http://us.archive.ubuntu.com/ubuntu bionic-security main restricted
deb http://us.archive.ubuntu.com/ubuntu bionic-security universe
deb http://us.archive.ubuntu.com/ubuntu bionic-security multiverse
deb http://archive.ubuntu.com/ubuntu/ bionic main
deb http://security.ubuntu.com/ubuntu/ bionic-security main
deb http://archive.ubuntu.com/ubuntu/ bionic-updates main
```

Save the file and perform an APT repository update.

```
# sudo apt update
```

Next, let's add the wget package and pull down the GRR server Debian package to install.

```
# sudo apt install wget

# wget https://storage.googleapis.com/releases.grr-response.com/grr-server_3.4.3-1_amd64.deb
```

Now we want to add packages required by GRR and Ansible.

```
# apt install mysql-server python3-dev python3-mysqldb

# apt install python3-pip python3-pexpect openssh-server vim debhelper dh-make zip rpm

# apt install nginx apache2-utils
```

Next, check the pip version, upgrade pip, and install the python module for using Ansible to generate OpenSSL certificates for Nginx.

```
# pip3 -V

# pip3 install --upgrade pip

# pip3 -V

# pip3 install pyOpenSSL
```

We want to run the grr install process just enough to download and install all of the required grr packages.

```
# apt install ./grr-server_3.4.3-1_amd64.deb
```

We will see a message stating that the MySQL service cannot be detected (Figure 4). The message will ask if you want to continue. Type 'Y' and press 'Enter'.

```
################################################################################
GRR has failed to detect a running MySQL instance on this machine.
This is ok if you plan on connecting to a remote MySQL instance.
If you aren't though, we recommend you exit this installation and install MySQL first.
FYI you can skip this check by setting DEBIAN_FRONTEND=noninteractive.
################################################################################

Would you like to proceed with GRR's installation? [Yn]:
Progress: [ 17%] [#############.......................................................................]
```

Figure 4

Once you reach the question on Fleetspeak (Figure 5), Ctl-Z out of the install process.

```
No old config file found.

Step 1: Setting Basic Configuration Parameters
We are now going to configure the server using a bunch of questions.
Use Fleetspeak (EXPERIMENTAL, next generation communication framework)? [yN]:  [N]:
Progress: [ 67%] [#######################################################################.....]
```

Figure 5

At this point, all packages required for a GRR installation should be downloaded.

After performing a Ctrl-z, the APT and Debian package processes will be in an unstable state. We want to remove the processes and remove the APT and Debian GRR package from the cache; this will not remove the downloaded GRR Debian package (.deb) from the ISO.

```
# ps aux | grep dpkg
```

Kill the Debian PIDs reported from the above command. There may be more than one PID.

```
# kill -9 <pid #>
```

Check for PIDs associated with the APT process.

```
# ps aux | grep apt
```

Kill the APT PIDs reported from the above command. There may be more than one PID.

```
# kill -9 <pid #>
```

Finally, we need to remove the GRR installation from Debian and APT. Note, this will not delete the GRR Debian package (.deb) that we downloaded earlier from the disk. Once complete, Click 'Next.'

```
# dpkg -r grr-server
# apt remove grr-server
```

Cubic will begin preparing the ISO (Figure 6). Once complete, click 'Next.'

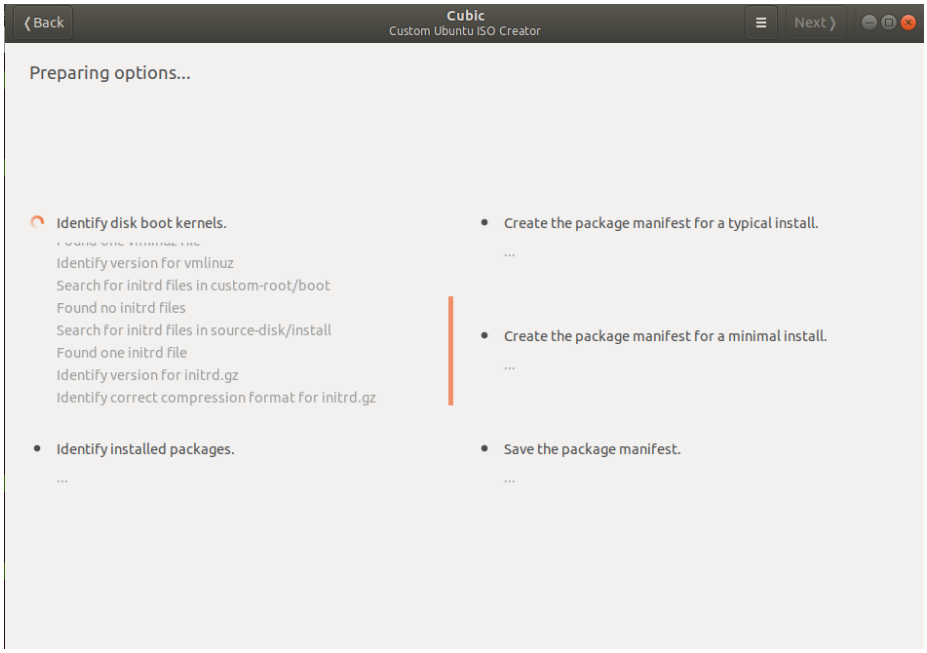

Figure 6

A screen is presented to allow us to remove packages from the ISO (Figure 7). If we need to remove packages for some reason, select them. Otherwise, click 'Next.'

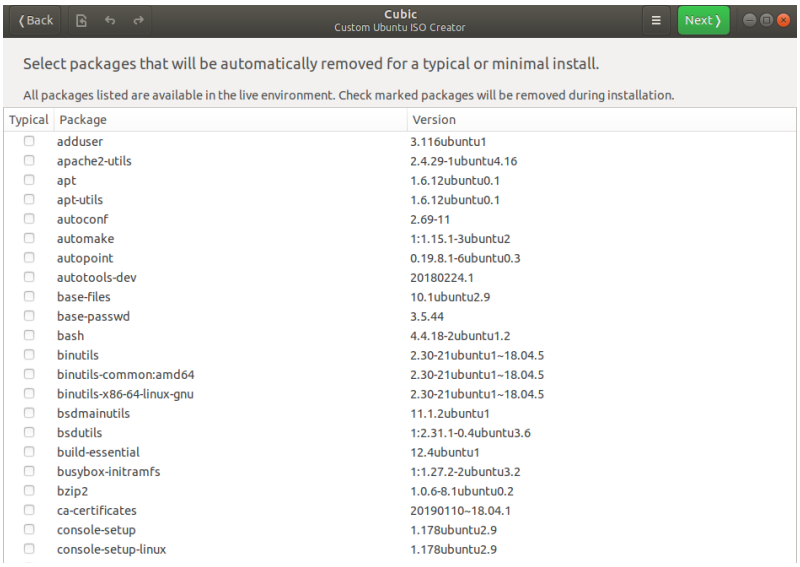

Figure 7

The next screen (Figure 8) presents us with three tabs, Kernel, Preseed, and Boot. We are only concerned with the Preseed and Boot tabs. Click the 'Preseed' tab.

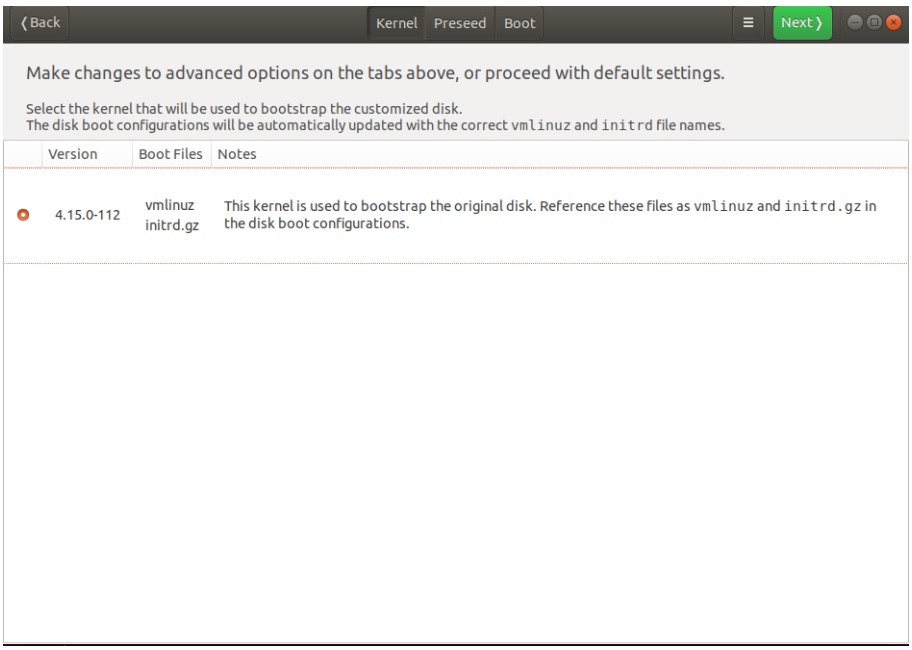

Figure 8

On the 'Preseed' tab (Figure 9), we will create a preseed file named 'auto-inst.seed'. This file will control how we install the customized ISO. Click the 'add file' icon  . Create a new file named 'auto-inst.seed.'

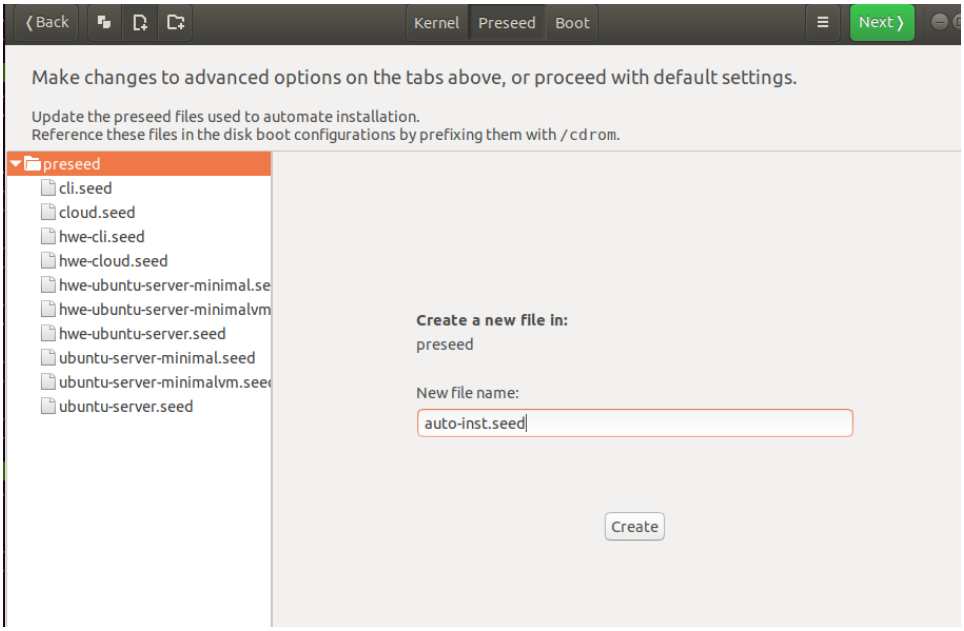

Figure 9

Looking at Figure 10, we can see the seed file. Below Figure 10 is the configuration posted into the seed file. Note that the server name, username, and passwords are fictitious.

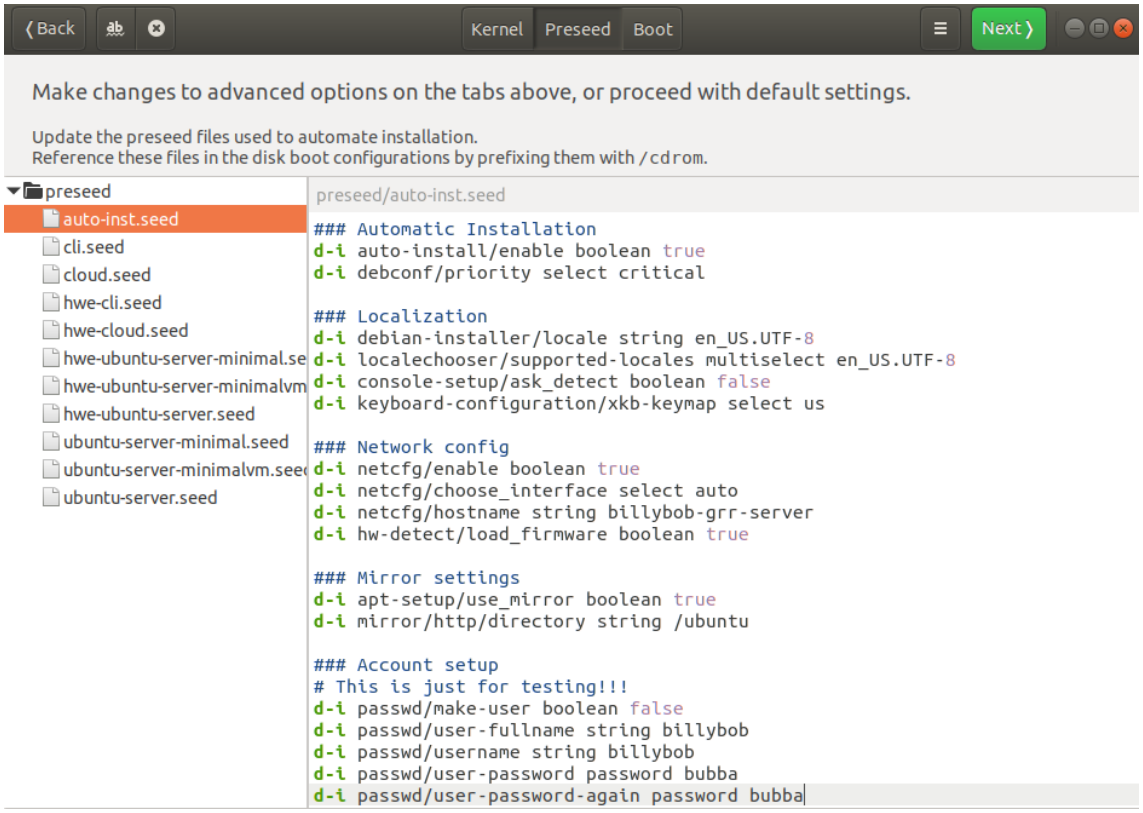

Figure 10

Auto-inst.seed file

Automatic Installation
d-i auto-install/enable boolean true
d-i debconf/priority select critical

Localization
d-i debian-installer/locale string en_US.UTF-8
d-i localechooser/supported-locales multiselect en_US.UTF-8
d-i console-setup/ask_detect boolean false
d-i keyboard-configuration/xkb-keymap select us

Network config
d-i netcfg/enable boolean true
d-i netcfg/choose_interface select auto
d-i netcfg/hostname string **billybob-grr-server**
d-i hw-detect/load_firmware boolean true

Mirror settings
d-i apt-setup/use_mirror boolean true

```
d-i mirror/http/directory string /ubuntu

### Account setup
# This is just for testing!!!
d-i passwd/make-user boolean false
d-i passwd/user-fullname string billybob
d-i passwd/username string billybob
d-i passwd/user-password password bubba
d-i passwd/user-password-again password bubba
d-i user-setup/allow-password-weak boolean true

# Set to true if you want to encrypt the first user's home directory.
d-i user-setup/encrypt-home boolean false

### Clock and time zone setup
d-i clock-setup/utc boolean true

# You may set this to any valid setting for $TZ; see the contents of
# /usr/share/zoneinfo/ for valid values.
d-i time/zone string America/New_York

# Controls whether to use NTP to set the clock during the install
d-i clock-setup/ntp boolean false

### Partitioning
# !!!DANGER don't use this without knowing what you are doing!!!
# comment out this block it you want the installer to ask about the
# partitioning, which is much safer!

# The following will partition disk /dev/sda with an EFI partition, a root partition
# and a swap file. AND WONT ASK TO CONFIRM ANYTHING i.e. it will overwrite existing partitions
d-i preseed/early_command string umount /media || true
d-i partman/unmount_active boolean true
d-i partman-auto/disk string /dev/sda
d-i partman-auto/method string regular
d-i partman-auto/choose_recipe select atomic
d-i partman-partitioning/confirm_write_new_label boolean true
d-i partman/choose_partition select finish
d-i partman/confirm boolean true
d-i partman/confirm_nooverwrite boolean true

# The kernel image (meta) package to be installed;
d-i base-installer/kernel/image string linux-generic
#d-i base-installer/kernel/altmeta string hwe-18.04

### Package selection
d-i tasksel/first multiselect none
```

d-i pkgsel/language-packs multiselect en
d-i pkgsel/update-policy select none

Apt setup
You can choose to install restricted and universe software, or to install
software from the backports repository.
d-i apt-setup/main boolean true
d-i apt-setup/multiverse boolean true
d-i apt-setup/restricted boolean true
d-i apt-setup/universe boolean true
d-i apt-setup/backports boolean true
d-i apt-setup/services-select multiselect security
d-i apt-setup/security_host string **us.archive.ubuntu.com**
d-i apt-setup/security_path string **/ubuntu**

Verbose output and no boot splash screen.
d-i debian-installer/quiet boolean false
d-i debian-installer/splash boolean false

d-i cdrom-detect/eject boolean true

Avoid that last message about the install being complete.
This will just finish and reboot
d-i finish-install/reboot_in_progress note

Next, click the 'Boot' tab. The screen (Figure 11) presents Multiple files. We are only concerned with changing the grub.cfg and txt.cfg files.

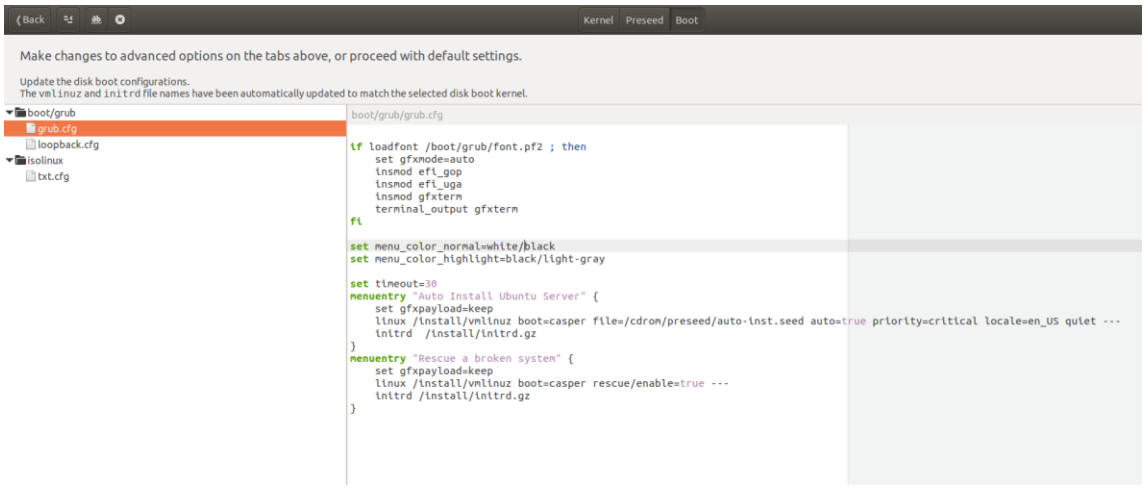

Figure 11

Below is the grub.cfg file we will replace with the current one. The configuration file below has been updated to remove any pausing in the installation process. It also points to the auto-inst.seed file for the installation configuration. Delete everything in the current grub.cfg file and add the following.

Grub.cfg file

```
if loadfont /boot/grub/font.pf2 ; then
	set gfxmode=auto
	insmod efi_gop
	insmod efi_uga
	insmod gfxterm
	terminal_output gfxterm
fi

set menu_color_normal=white/black
set menu_color_highlight=black/light-gray

set timeout=30
menuentry "Auto Install Ubuntu Server" {
	set gfxpayload=keep
	linux /install/vmlinuz boot=casper file=/cdrom/preseed/auto-inst.seed auto=true
priority=critical locale=en_US quiet ---
	initrd  /install/initrd.gz
}
menuentry "Rescue a broken system" {
	set gfxpayload=keep
	linux /install/vmlinuz boot=casper rescue/enable=true ---
	initrd /install/initrd.gz
}
```

Click on the txt.cfg file (Figure 12). We are going to change the isolinux txt.cfg file to point to the new auto-inst.seed file for installation configuration instructions.

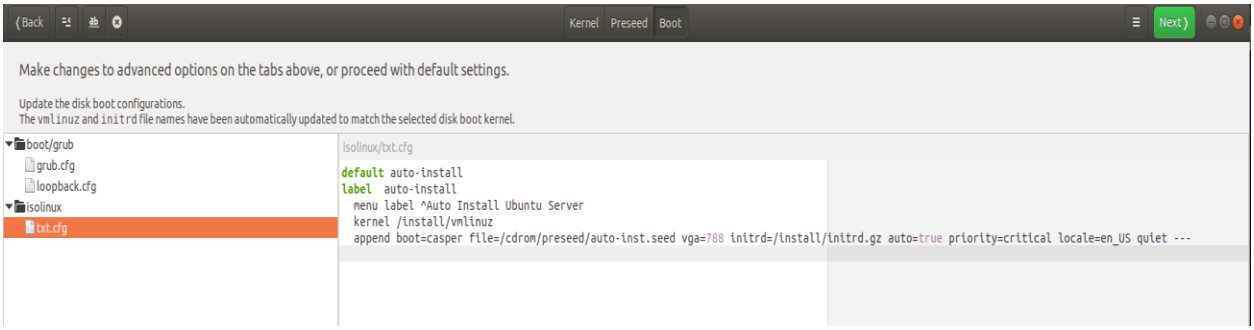

Figure 12

Delete the current txt.cfg configuration and add the following to it. Once complete, click 'Next.'

txt.cfg file

```
default auto-install
label  auto-install
  menu label ^Auto Install Ubuntu Server
  kernel /install/vmlinuz
  append boot=casper file=/cdrom/preseed/auto-inst.seed vga=788 initrd=/install/initrd.gz auto=true
priority=critical locale=en_US quiet ---
```

A screen with the Linux compression selection is presented (Figure 13). Click' Generate.'

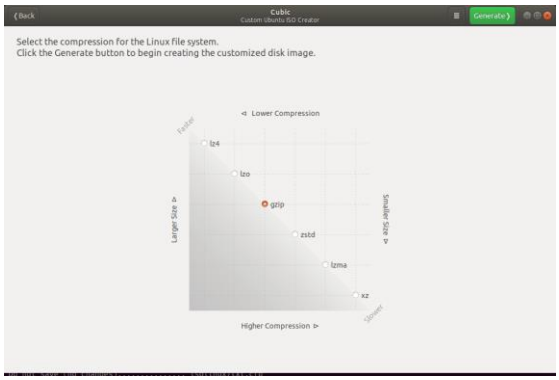

Figure 13

Finally, a screen displaying the ISO generation process is presented (Figure 14). Once complete, click 'Finish.' This completes the GRR custom ISO process.

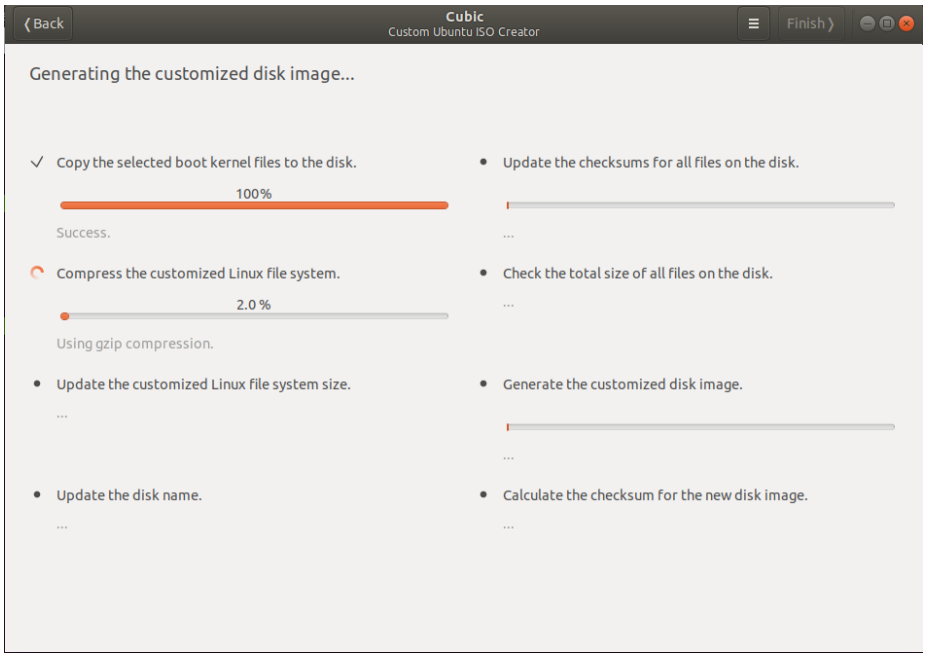

Figure 14