

# ICN Bootable ISO

## Bootstrap flow

1. Install ubuntu 18.04 OS with our customized ISO. This is no-attended installation, no user-input. As simple as plugin the USB and press power-on button.
2. Edit config file and run script/ansible to deploy all kinds of services.

## What's inside the ISO

We use mkisofs to build the ISO from a directory name **preseed\_ubuntu**

```
sudo mkisofs -q -disable-deep-relocation -rational-rock -cache-inodes -joliet -full-iso9660-  
filenames -no-emul-boot -boot-load-size 4 -boot-info-table -e  
ltorito-boot isolinux/isolinux.bin -eltorito-catalog isolinux/boot.cat -o ubuntu-pre.iso  
preseed_ubuntu
```

We use debian-installer preseed to achieve no-attended installation.

```
$ cat preseed_ubuntu/isolinux/isolinux.cfg  
default install  
label install  
menu label ^Install Ubuntu Server  
kernel /install/vmlinuz  
append debian-installer/locale=en_US console-setup/ask_detect=false keyboard-configuration/layoutcode=us file=/cdrom/preseed  
/ubuntu-server.seed vga=788 initrd=/install/initrd.gz quiet ---
```

As the infra-local-controller is supposed not having Internet access, we have to prepare everything into the ISO. Every service should have a preparation script which prepares every files needed for setting up the service. For example, we need to have a preparation script for Kubernetes service to download kubelet package and kube-api image. All the preparation scripts are called before **mkisofs**.

The host reboots after ISO installation. Once reboot, the first thing we have to do is setting up kubernetes. Because we are supposed to run services on the Kubernetes, i.e. apt repo, docker registry, etc.

We setup private docker registry with image **registry:2**. For apt repository, we use the [debify image](#)