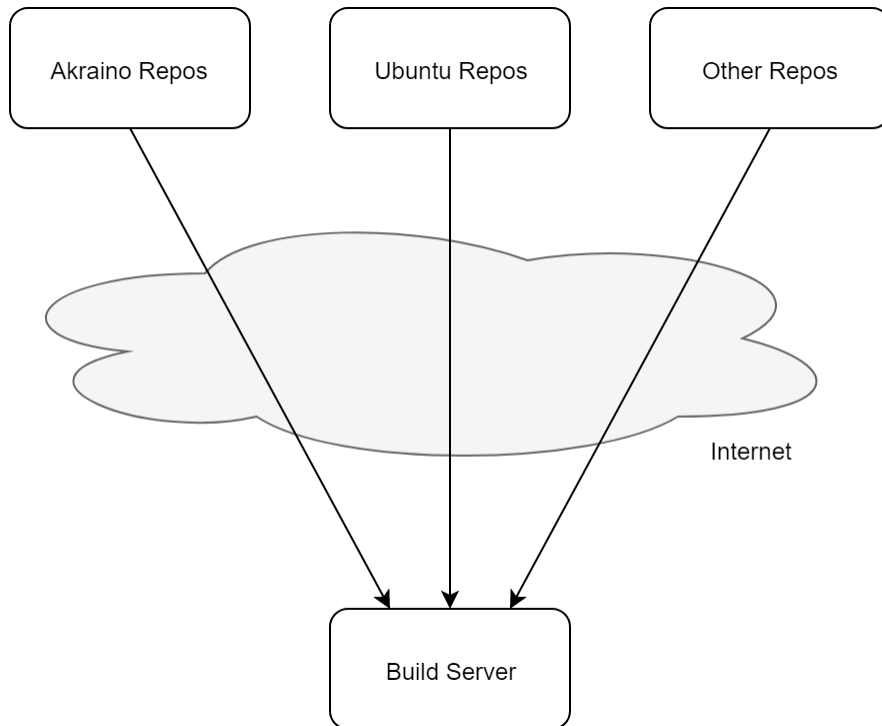


Build Server Configuration

- [Introduction](#)
- [Preflight requirements](#)
 - [Networking](#)
 - [Software](#)
- [Preflight checks](#)
- [Installing the Build Server packages](#)

Introduction



The 'Build Server' is only used to build and deploy a Regional Controller (RC). Therefore it can be considered ephemeral to the deployment and operation of the other Network Cloud RCs and Unicycle/Rover edge pod infrastructure.

Typically the build server would be created in a VM but could be a physical bare metal server.

The installation procedure to create the build server primarily involves cloning the set of repositories and packages implementing the Network Cloud Family blueprints in the R1 release.

Preflight requirements

Networking

The build server must have connectivity to the internet to be able to clone the necessary repos and packages.

The build server must have IP connectivity to any RCs it is to deploy. In addition, the RCs being build will need access to a DHCP server. There are several options for the DHCP requirement:

- If the Build Server is on the same broadcast domain, the Build Server can act as the DHCP server for the RC.
- If the Build Server is not on the same broadcast domain, the network can be setup with dhcp relays to forward DHCP requests to the Build Server.
- A pre-existing DHCP server can be used (either local to the RC or using dhcp relays)

Software

Before starting to clone upstream repos and packages the build server must have the following pre-installed

- Ubuntu Release 16.04
- Latest version of the following apt packages:
 - docker (used to run dhcp and web containers)
 - python (used for redfish api calls to bare metal server)
 - python-requests (used for redfish api calls to bare metal server)
 - python-pip (used to install hpe redfish tools)
 - sshpass (used to copy keys to new server)
 - xorriso (used to extract Ubuntu files to web server)
 - make (used to build custom ipxe efi file used during bare metal server boot)
 - gcc (used to build custom ipxe efi file used during bare metal server boot)

Preflight checks

Ensure Ubuntu Release 16.04 (specifically) and Docker version is 1.13.1 or newer:

```
root@build_server:/# lsb_release -rs
16.04
root@build_server:/# docker --version
Docker version 1.13.1, build 092cba3
```

Ensure required packages are installed including python, python-requests, python-pip, sshpass, xorriso, make, and gcc are installed. Install any missing packages with apt-get install -y <package name>

```
root@build_server:/# apt list python python-requests python-pip sshpass xorriso make gcc
Listing... Done
gcc/xenial,now 4:5.3.1-1ubuntu1 amd64 [installed]
make/xenial,now 4.1-6 amd64 [installed,automatic]
python/xenial-updates,now 2.7.12-1~16.04 amd64 [installed]
python-pip/xenial-updates,xenial-updates,now 8.1.1-2ubuntu0.4 all [installed]
python-requests/xenial-updates,xenial-updates,now 2.9.1-3ubuntu0.1 all [installed]
sshpass/xenial,now 1.05-1 amd64 [installed] xorriso/xenial,now 1.4.2-4ubuntu1 amd64 [installed]
```

Installing the Build Server packages

If you haven't done so already, elevate yourself to root:

```
user@build_server:/# sudo -i
```

Clone the Redfish Bootstrapping Script repository for use as part of Akraino tools:

```
## Download the latest redfish artifacts from LF Nexus

root@build_server:/# mkdir -p /opt/akraino/redfish

root@build_server:/# NEXUS_URL=https://nexus.akraino.org

root@build_server:/# curl -L "$NEXUS_URL/service/local/artifact/maven/redirect?r=snapshots&g=org.akraino.
redfish&a=redfish&v=0.0.2-SNAPSHOT&e=tgz" | tar -xozv -C /opt/akraino/redfish
```

Clone the Akraino Regional Controller repository:

```
## Download the latest Regional_controller artifacts from LF Nexus ##

root@build_server:/# mkdir -p /opt/akraino/region

root@build_server:/# NEXUS_URL=https://nexus.akraino.org

root@build_server:/# curl -L "$NEXUS_URL/service/local/artifact/maven/redirect?r=snapshots&g=org.akraino.
regional_controller&a=regional_controller&v=0.0.2-SNAPSHOT&e=tgz" | tar -xozv -C /opt/akraino/region
```

At this point there will be two new directories where the cloned NC artifacts have been created.

```
root@build_server:/# ls /opt/akraino/

redfish  region
```

This completes the installation of the build server and it can now be used to deploy one or more Regional Controllers.