**Prerequisites**

The prerequisites for preparing to deploy a private LTE Magma network are listed below. It is preferred to use a MAC Laptop or PC, Windows can be used with CYGWIN. Prepare or install the following hardware and software before you begin. A wired internet connection is recommended, do NOT to use a VPN during installation as it may effect connectivity between VM’s.   
  
Install the following onto the PC/Laptop that will be deploying the Magma Dev Package:

**Development Tools**

Install the following developer tools:

1. [Docker](https://www.docker.com/)  Image Depository to deploy Magma containers
2. [Homebrew](https://brew.sh/) *only* for MacOS users, see the [pyenv installation instructions](https://github.com/pyenv/pyenv" \l "installation)
3. [VirtualBox](https://www.virtualbox.org/) Oracle Virtual Box, version 5.2 or higher
4. [Kubectl](https://kubernetes.io/docs/tasks/tools/install-kubectl/) kubernetes CLI tool allows you to run commands against Kubernetes clusters
5. [Helm](https://helm.sh/) 2.0 manages Kubernetes applications; define, install, and upgrade
6. [Terraform](https://www.terraform.io/downloads.html) manages the Orchestrator cloud deployment

Prepare the following hardware & software:

1. **An eNodeB** - the recommended and tested models are:

* BaiCells Nova 233 TDD Outdoor
* BaiCells Nova 243 TDD Outdoor
* Assorted Baicells indoor units for lab deployments
* An Antenna connector for the eNodeB

Support for other RAN hardware can be implemented inside the enodebd service on the AGW, it is recommended to start with one of the above eNodeB’s.

1. **An Access Gateway (AGW)** - the recommended requirements are: Any 64bit-X86 machine, Dual-CPU, 2GHz clock speed or faster, 2GB RAM, 128GB-256GB SSD storage which can support a Debian Linux installation. With a minimum of 2+ physical ethernet interfaces (enp1s0=SGi and enp2s0= S1),

* A USB stick with 2GB+ capacity to load a Debian Stretch ISO Used to build the AGW  
    
  *To add the Magma Access Gateway (AGW) to the new network, use the Magma Network Management System (NMS).*

1. **An AWS Account** - with the permissions to create multiple Amazon EC2 virtual servers for the Magma Orchestrator, Network Management System (NMS) and Access Gateways if they are deployed in the cloud. When using access and secret keys for an administrator account ***do not use*** the root account.

* Create Root account
* Create an Admin User account
* Obtain an <access key ID> for an administrator account
* Obtain a <secret key> for an administrator account

1. **Magma Dev Package** - clone from the [Facebook Github](https://github.com/facebookincubator/magma/) repository

* Setup a [GitHub Account](https://help.github.com/en/github/getting-started-with-github/set-up-git)
* Install Git to use command line

1. **A Registered Domain**/URL for Orchestrator endpoints
2. **CPE** such as smartphone or tablet SIM card with known key values for testing.

**Installing Homebrew**

MAC users install brew , or replace with your OS appropriate package manager, see the [pyenv installation instructions](https://github.com/pyenv/pyenv" \l "installation).

brew install pyenv

# Replace .bash\_profile with your appropriate shell RC file

# IMPORTANT: Use .bash\_profile, not .bashrc for bash

echo -e 'if command -v pyenv 1>/dev/null 2>&1; then\n eval "$(pyenv init -)"\nfi' >> ~/.bash\_profile

exec $SHELL

pyenv install 3.7.3

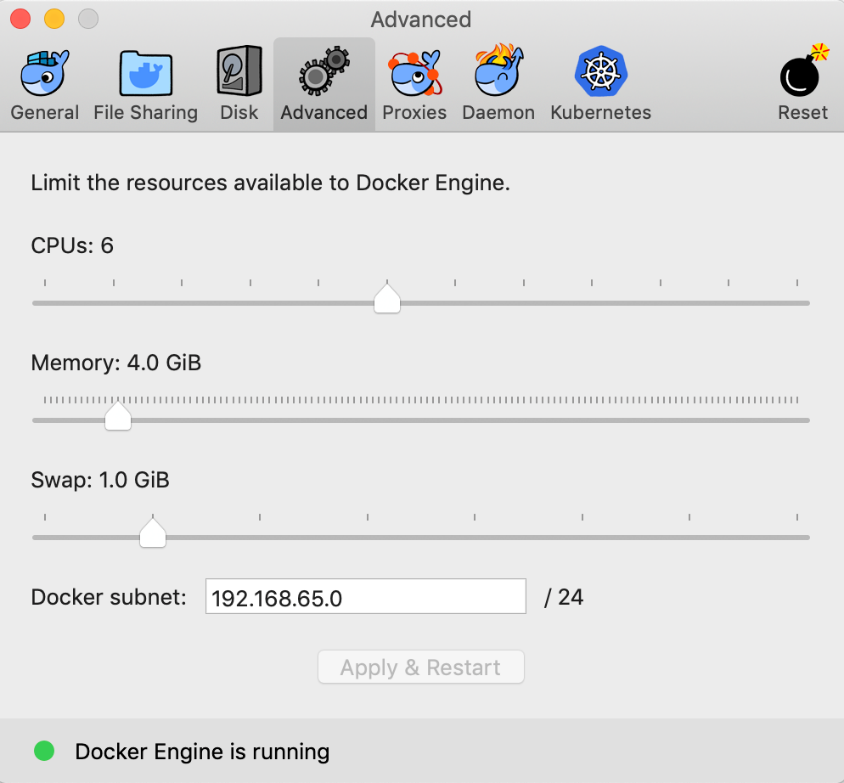
pyenv global 3.7.3

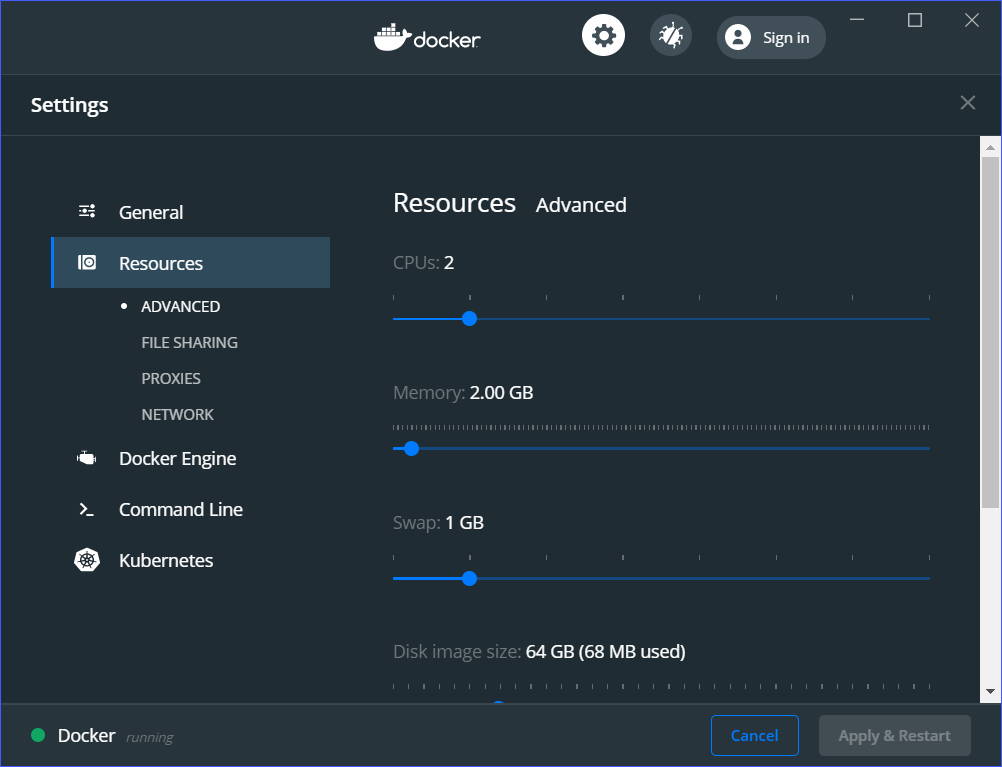
pip3 install ansible fabric3 jsonpickle requests PyYAML

**Installing Docker**Download and install Docker onto the PC/Laptop that will be deploying the Magma Dev Package.

After installation for MacOS, start Docker for Mac and increase the memory allocation for the Docker engine to at least 4GB (Preferences -> Advanced).

Windows users start Docker for Windows and increase the memory allocation for the Docker engine to at least 4GB (Docker -> Settings).





## Setup AWS-IAM-Authenticator

We support building the AGW and Orchestrator on MacOS and Linux host operating systems. Doing so on a Windows environment is possible but is not preferred. You may prefer to use a Linux virtual machine if you are on a Windows host.

AWS-IAM-Authenticator provides authentication to your Amazon Web Services (AWS) Kubernetes cluster.  After the Developer Tools have been installed from above, install the AWS tools. (*replace brew with your OS-appropriate package manager as necessary*):

brew install aws-iam-authenticator kubernetes-cli kubernetes-helm terraform

pip3 install awscli

aws configure

When prompted by aws configure to provide an <access key ID> and <secret key> for an administrator user in AWS (do not use the root user account). Skip this step if you will use something else for managing AWS credentials.

## Preparing the Production Hardware

### Access Gateways

Access Gateways (AGWs) can be deployed on to any AMD64 architecture machine which can support a Debian Linux installation. The basic system requirements for the AGW production hardware are:

1. 2+ Gigabit Ethernet NIC ports/interfaces ( etch1 and etch2 )

NICs that support Hardware Offloading include:

* + NIC number 1
  + NIC number 2
  + etc . . .

1. AMD64 dual-core processor, 2GHz clock speed or faster
2. 2GB RAM
3. 128GB-256GB SSD storage
4. A USB stick with 2GB+ capacity to load a Debian Stretch ISO
5. Peripherals (keyboard, screen) for your production AGW box for use during provisioning

Orchestrator and NMS

The Orchestrator deployment depends on the following components:

1. An AWS account
2. A Docker image repository (e.g. Docker Hub, JFrog)
3. A registered domain for Orchestrator endpoints

It is recommended to deploy the Magma Orchestrator cloud component into AWS. Our open-source Terraform scripts target an AWS deployment environment. If you are familiar with DevOps and are willing to roll your own, Orchestrator can run on any public/private cloud with a Kubernetes cluster available to use. The deployment documentation will assume an AWS deployment environment.

You will also need a Docker image repository available to publish the various Orchestrator NMS containers to. We recommend using a private repository for this.