

Automated Hardware Provisioning in the Real-world
http://spkane.github.io/velocityny_2014-automating_hardware





Sean P. Kane - [@spkane](#)
New Relic - Site Engineering

Automated Hardware Provisioning in the Real-world

or

"How to Build Your Cloud Out of Metal"

- 0 TO 60 -

- 00: Small hardware orders with manual server installs
- 60: Racks worth of hardware and automated installs

What we need to build

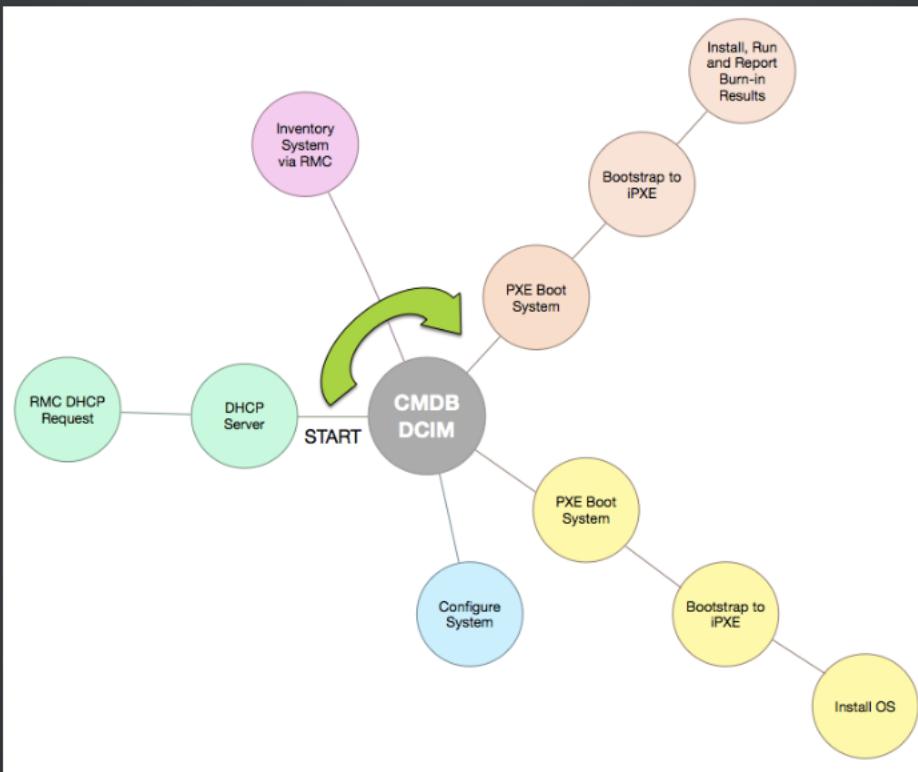


Image: [@spkane](#)

DESIGN FOR AGILITY

Just Enough Process

What's the bare minimum we need to:

- Detect new hardware in a rack
- Automatically inventory and test it
- Automatically install and configure it

**IT'S ALL GREEK
TO ME**

Learning to Talk Like a Native

IPMI

Intelligent Power Management Interface

- Power Management
- Sensor Query & Tuning
- System Inventory & Logs
- SNMP Traps

BMC

Baseboard Management Controller

- The brain behind IPMI
- The BMC is the minimum management component

RMC

Remote Management Cards

DRACs, and iLOs, and RSAs, Oh my!

- Always On Management
- SSH & Web Interfaces
- Deep system capabilities
- KVM over IP

IPMI breakout

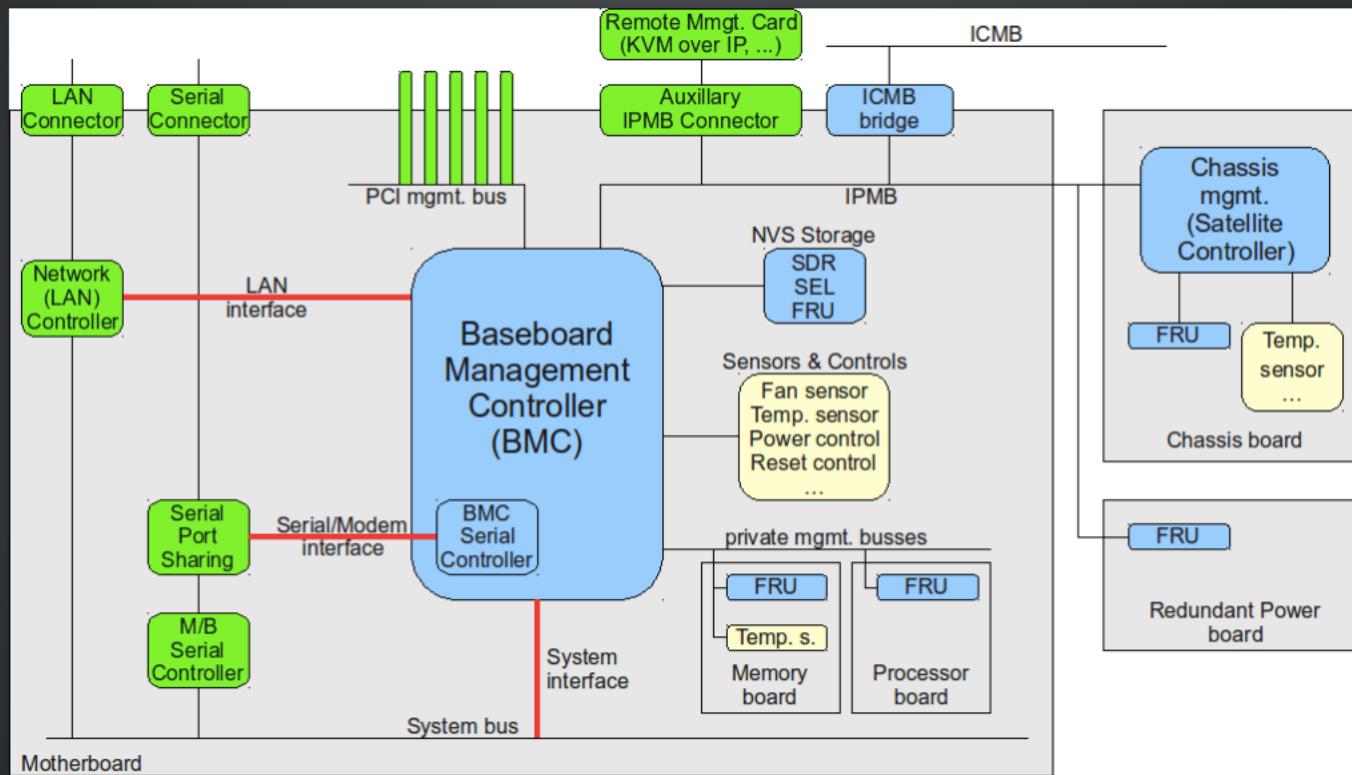


Image: [Thomas-Krenn AG](#)

CMDB
Configuration Management
Database

DCIM
Datacenter Information
Management

ORDER & RACK THE HARDWARE

- RMCs should be set to DHCP
- They can also use DDNS

-

DHCP - Dynamic Host Configuration Protocol

DDNS - Dynamic Domain Name Service

BOOTSTRAPPING THE SYSTEM

Waking up the RMC

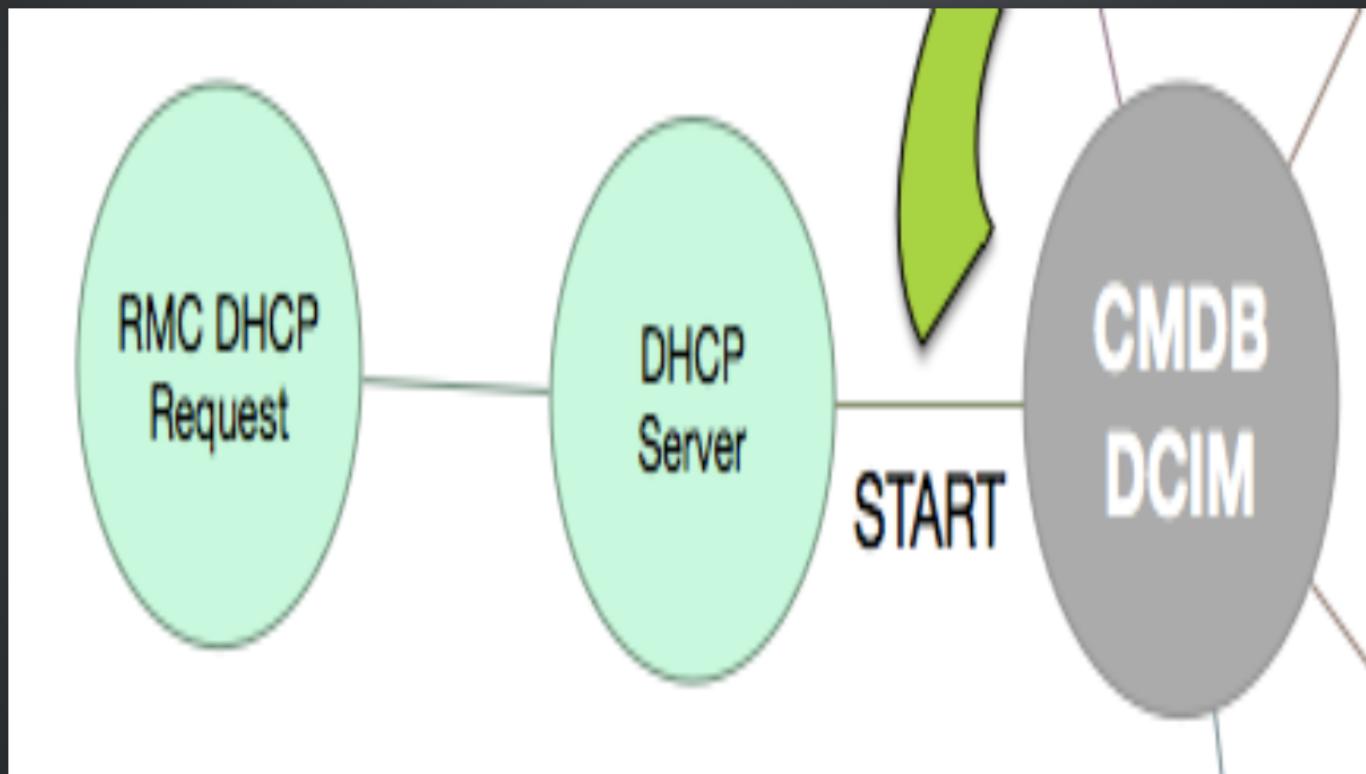


Image: [@spkane](#)

DHCP TRIGGER

The magical bootstrap

```
on commit {
    set clientip = binary-to-ascii(10, 8, ".", lease
    set clientmac = binary-to-ascii(16, 8, ":", sub
    set leasetime = binary-to-ascii(10,32,"",encode
    set vendorclass = pick-first-value(option vendor
    set ddns-hostname = pick-first-value(
        option fqdn.hostname,
        option host-name,
        concat("dhcp-", binary-to-ascii(10, 8, "-"
        suffix(leased-address, 2))));)
    execute("/usr/local/bin/dhcp-event.sh", "commit
}
```

- Note: The execute script is blocking. It should return immediately!

TRIGGER SCRIPT

```
#!/bin/bash

API_USER_AUTH="something_secret"
API_HOST="http://cmdb.example.com"
API_BASE_URL="/api/v1"
API_DISCOVER_URL="/hosts/discover"
CURL_BIN="/usr/bin/curl"
URL="${API_HOST}${API_BASE_URL}${API_DISCOVER_URL}"

if [ $1 == "commit" ]; then
    ${CURL_BIN} --max-time 3 \
    --silent \
    -H 'Content-Type: application/json' \
    -X POST \
    --user ${API_USER_AUTH} \
    --data "{ \"host\": { \"ip_address\": \"${2}\", \
    ${URL} } > /dev/null
else
    exit 0
fi
```

Detect and Test

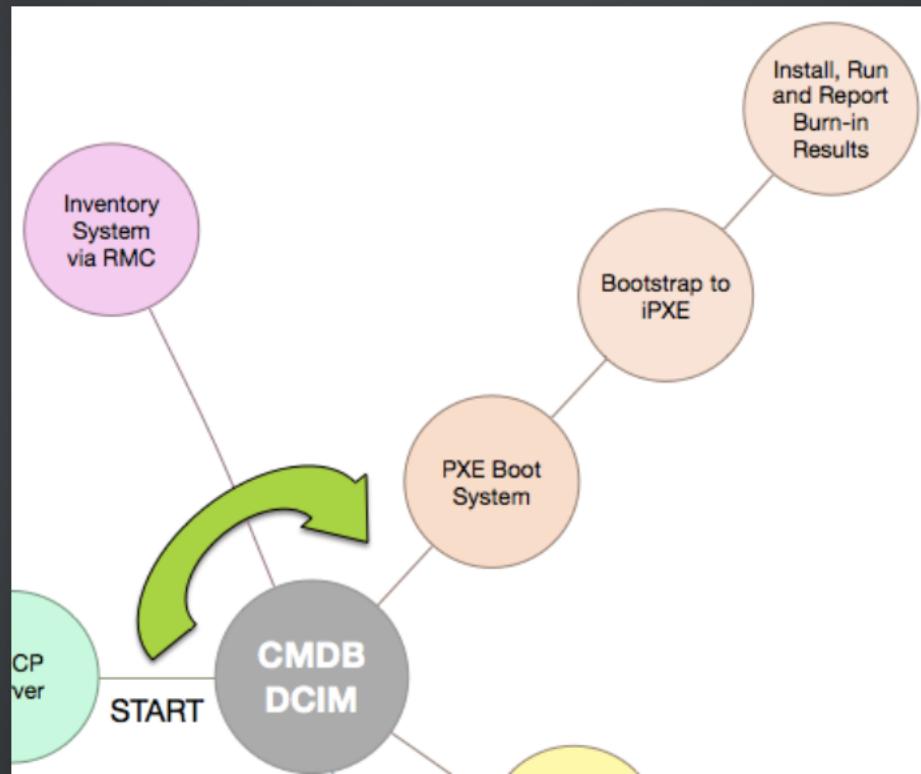


Image: [@spkane](#)

INTERROGATE THE RMC

- ID system via Serial Number
- Determine hardware details

REGISTER THE SYSTEM

- Create a detailed entry in the CMDB/DCIM

BOOT THE SYSTEM

PXE & iPXE

Preboot eXecution Environment

CHAINLOADING IPXE FROM PXE

Give me the Network Bootloader

```
host dhcp-host10 {
    hardware ethernet FF:CA:3A:6E:66:00;
    fixed-address 10.20.250.42;
    option host-name "dhcp-host.example.com";
    option routers 10.20.250.1;
    if exists user-class and option user-class = "gP"
        filename "http://10.20.250.10:80/ipxe/system";
    } else if exists user-class and option user-class
        filename "http://10.20.250.10:80/ipxe/system";
    } else {
        filename "undionly.kpxe";
    }
}
```

BOOTING IPXE

```
#!ipxe

:retry_dhcp
dhcp && isset http://boot.ipxe.org/demo/boot.php || c
echo Booting from http://boot.ipxe.org/demo/boot.php
chain http://boot.ipxe.org/demo/boot.php
```

BOOTING THE SYSTEM

Give me the Kernel & Ramdisk

```
#!ipxe  
  
kernel vmlinuz-3.16.0-rc4 bootfile=http://boot.ipxe.c  
initrd initrd.img  
boot
```

LIVE BOOT OS VIA IPXE

- Firmware updates
- Hardware Burn-in

BURNIN TOOLS

- bonnie++: <http://www.coker.com.au/bonnie++/>
- memtester: <http://pyropus.ca/software/memtester/>
- Redhat Memtest:
<http://people.redhat.com/~dledford/memtest>
- Phoronix Test Suite (PTS): <http://www.phoronix-test-suite.com/>

REGISTER THE RESULTS

- Update host record in CMDB/DCIM

Installing Systems

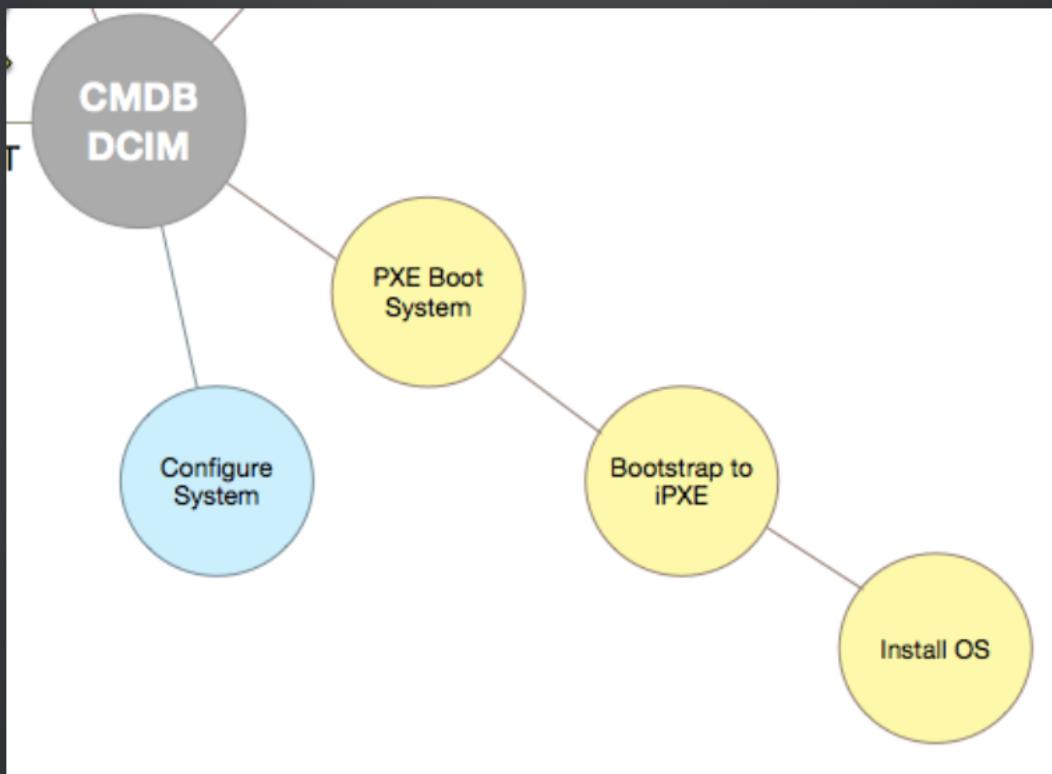


Image: [@spkane](#)

OS INSTALLERS

Give me a script

Kickstart or Preseed

- Kickstart example: <http://goo.gl/z1fBwY>
- Preseed example: <http://goo.gl/QWXvqR>

PACKAGE REPOSITORIES

Give me the media

-
Yum, APT, etc.

OS CONFIGURATION

Give me the configuration

-

Cloud-Init, Puppet, Chef, etc.

- Cloud-init examples: <http://goo.gl/ogQUwN>
- Puppet examples: <http://goo.gl/Tk77SH>
- Chef examples: <http://goo.gl/wA5nDH>

REGISTER THE RESULTS (AGAIN)

- Update host record in CMDB/DCIM

Success!!



Image: [Torkild Retvedt](#)

OUR NEXT STEPS

MICRO-SERVICES EVERYWHERE

Small REST APIs
for workflow construction

- Inventory (over-time)
- Health
- Remote configuration
- Etc.

THE FUTURE...

REST APIs in Hardware

- Software should not rely on SSH
- Humans should not use XML
- Nothing should implement a custom protocol

Open Compute Project

<http://www.opencompute.org/>

ANY QUESTIONS?

Interested in a great job?

Talk to me

or visit

<http://newrelic.com/about/careers>

Sean P. Kane

@spkane