



Building the Internet of Things with Eclipse IoT*

Benjamin Cabé – Eclipse Foundation
@kartben

** and more!*



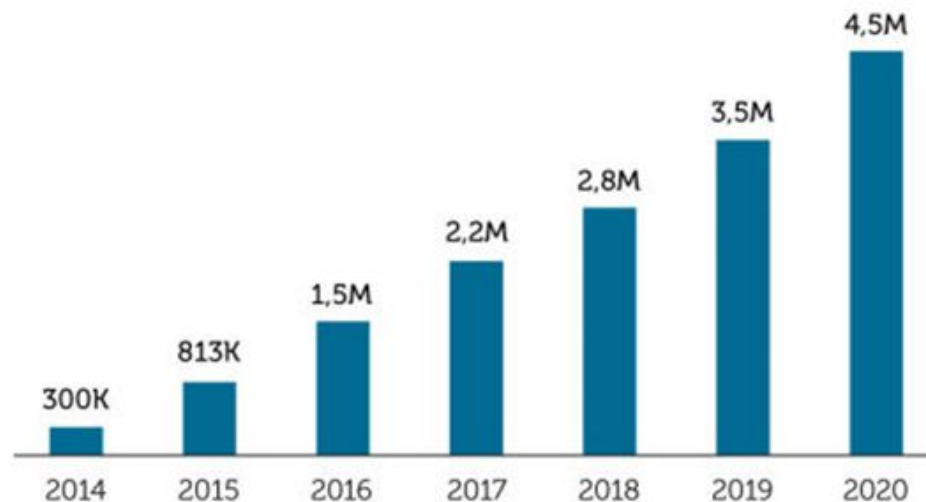








THE NUMBER OF IOT DEVELOPERS 2014-2020



Source: VisionMobile estimates, 2014



Report: IoT: Breaking Free From Internet And Things | vmob.me/IoT

©VisionMobile | June 2014 | Licensed under CC BY ND





- 19 open-source projects*
 - Lots of **Java** but also C, C++, Python, Go, .Net, ...
-
- **IoT Standards**
 - **Services & Frameworks**

* and counting!

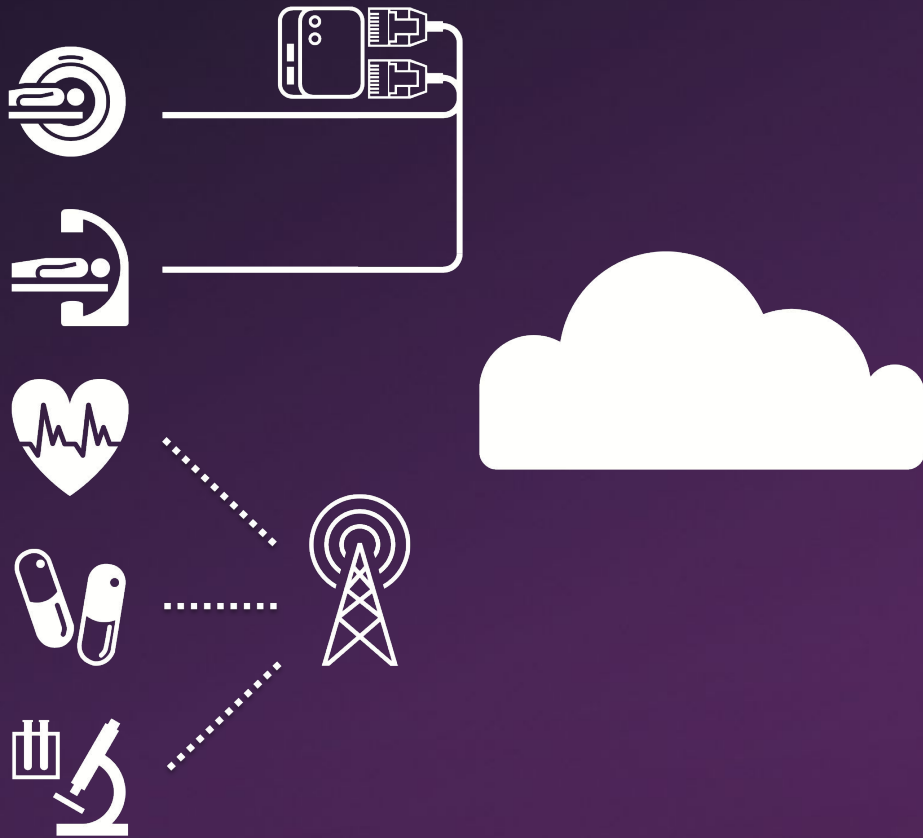
End-to-end IoT?



End-to-end IoT?



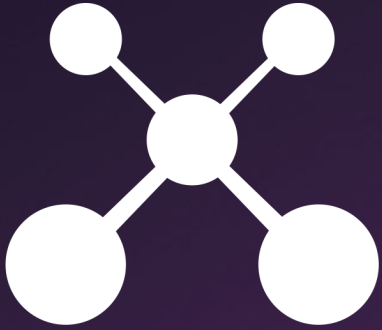
End-to-end IoT?



End-to-end IoT?

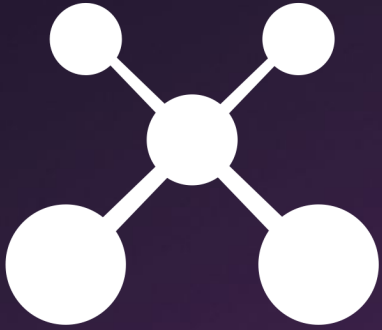


What you will learn today



CONNECT

What you will learn today

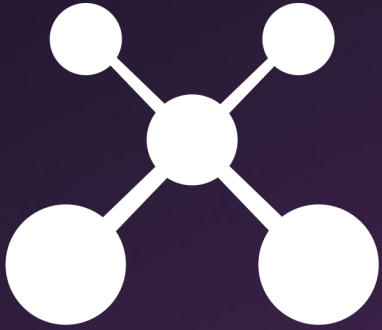


CONNECT



MANAGE

What you will learn today



CONNECT

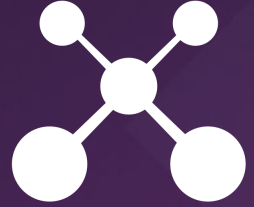


MANAGE



VISUALIZE

Connecting things to the IoT?

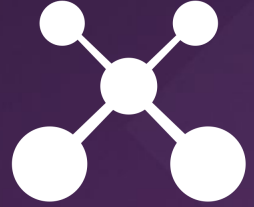


Network is often not reliable

Bandwidth == \$\$\$

Different communication patterns

Connecting things to the IoT



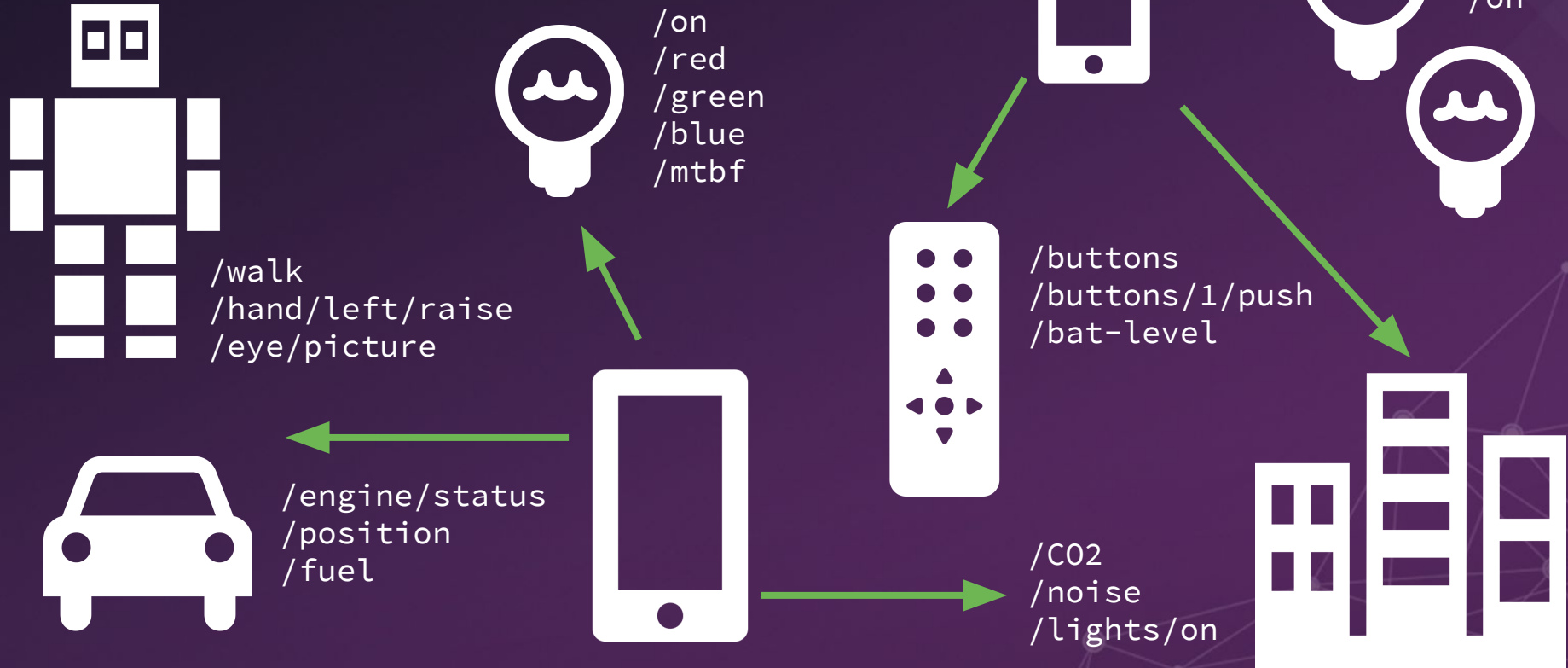
- **CoAP**

- « HTTP over UDP »
- Expose your device as a resource to the Internet of Things

- **MQTT**

- Publish/Subscribe model
- TCP-based

CoAP: The web-of-things



Eclipse Californium



- Focus on scalability and usability
- To be used in IoT cloud servers or M2M/IoT devices running Java
- Includes **DTLS** implementation (Scandium), HTTP/CoAP bridge, Plugtests, ...

<http://eclipse.org/californium>

Californium 101



CoapServer, CoapResource, CoapExchange

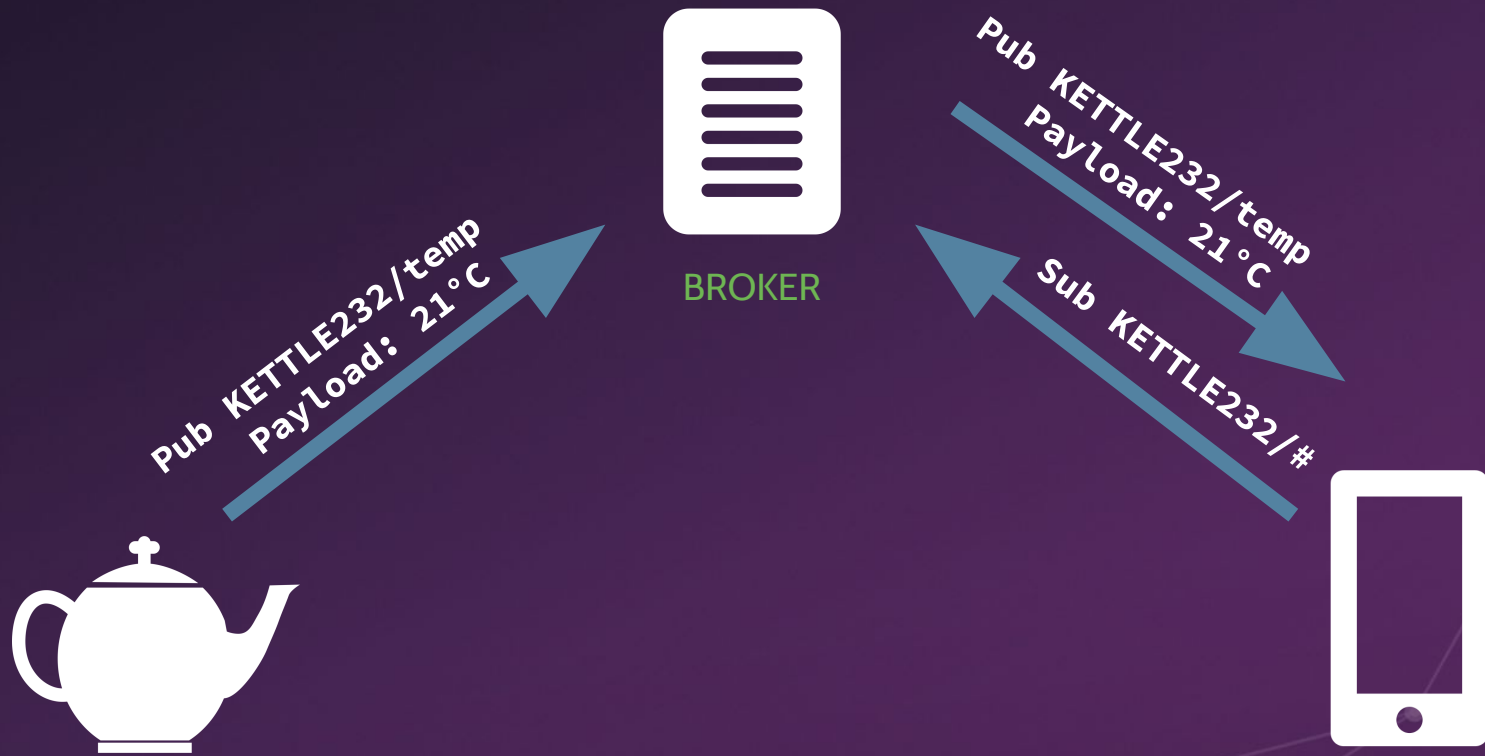
1. Implement custom resources
(extend CoapResource)
2. Add resources to the CoAP server
3. Start the server

```
import static org.eclipse.californium.core.coap.CoAP.ResponseCode.*;
```



```
public class MyResource extends CoapResource {  
    @Override  
    public void handleGET(CoapExchange exchange) {  
        exchange.respond("hello world"); // reply with 2.05 payload (text/plain)  
    }  
  
    @Override  
    public void handlePOST(CoapExchange exchange) {  
        exchange.accept(); // make it a separate response  
  
        if (exchange.getRequestOptions() ...) {  
            // do something specific to the request options  
        }  
        exchange.respond(CREATED); // reply with response code only (shortcut)  
    }  
}
```

MQTT: Publish & Subscribe



MQTT in 5 keywords

Pub-Sub

Wildcards

Quality of Service

Last Will & Testament

Retained Messages

Eclipse Paho



- Open-source MQTT clients
- Pick your language!
 - Java
 - JavaScript
 - C/C++, Objective C
 - Go, Lua, Python, .NET, WinRT, ...

<http://eclipse.org/paho>

```
MqttClient c = new MqttClient("tcp://iot.eclipse.org:1883",  
    MqttClient.generateClientId());
```



```
mqttClient.setCallback(new MqttCallback() {  
    @Override  
    public void messageArrived(String topic, MqttMessage message)  
        throws Exception {  
        // process received message  
        // ...  
    }  
});  
  
mqttClient.connect();  
mqttClient.subscribe("mygateway/#");
```

Open source MQTT brokers

- **Eclipse Mosquitto**

- C implementation
- Pretty scalable (1000 clients == 3MB RAM)

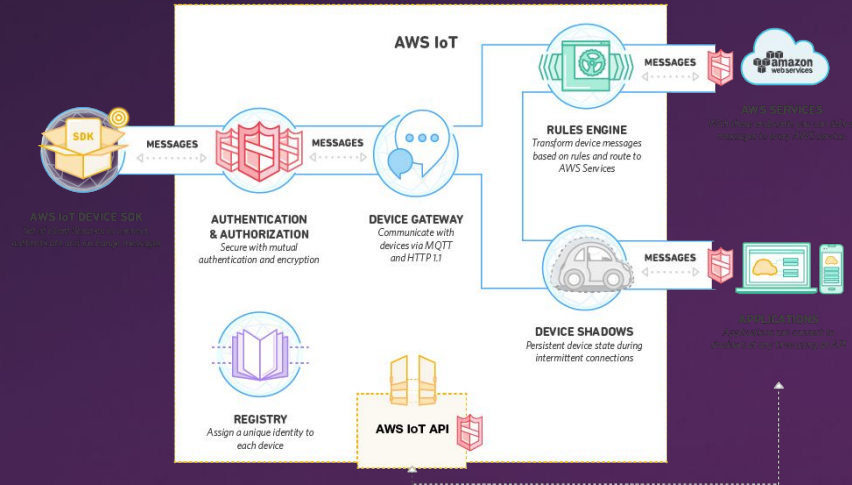
- **But also...**

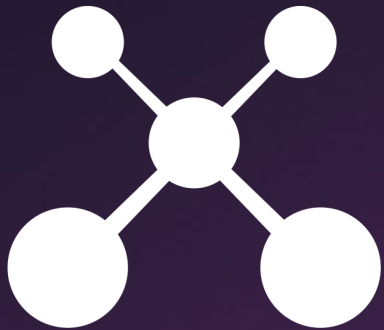
- Moquette (Java, Based on Netty and LMAX disruptor)
- VerneMQ (Erlang)
- Mosca (Node.js)

⇒ <https://github.com/mqtt/mqtt.github.io/wiki/servers>

Oh, and by the way...

Amazon just announced support for MQTT in their new AWS IoT cloud platform





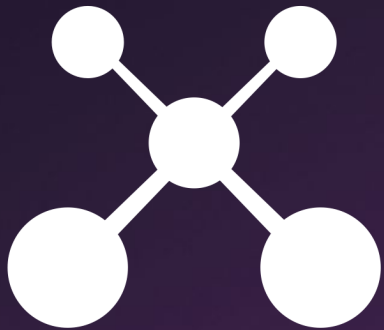
CONNECT



MANAGE



VISUALIZE



CONNECT



MANAGE



VISUALIZE

Yup, lots of aspects to manage

- **Network**

- PPP cellular connection, WiFi hotspot, Zigbee coordination, VPN, firewall ...

- offline/online mode

- **Applications**

- Remote install, start, stop, configure, ...

- Sandboxing

- **Hardware**

Gateways to the rescue!



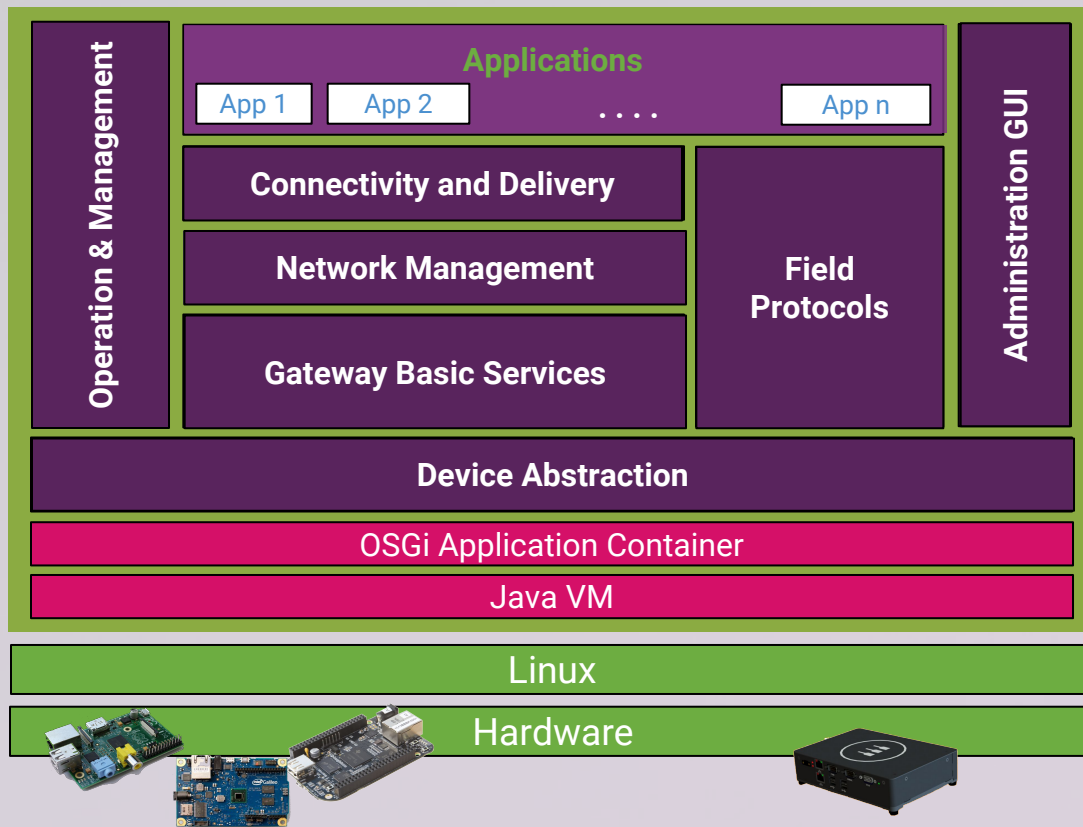
Gateway



Gateway



Eclipse Kura



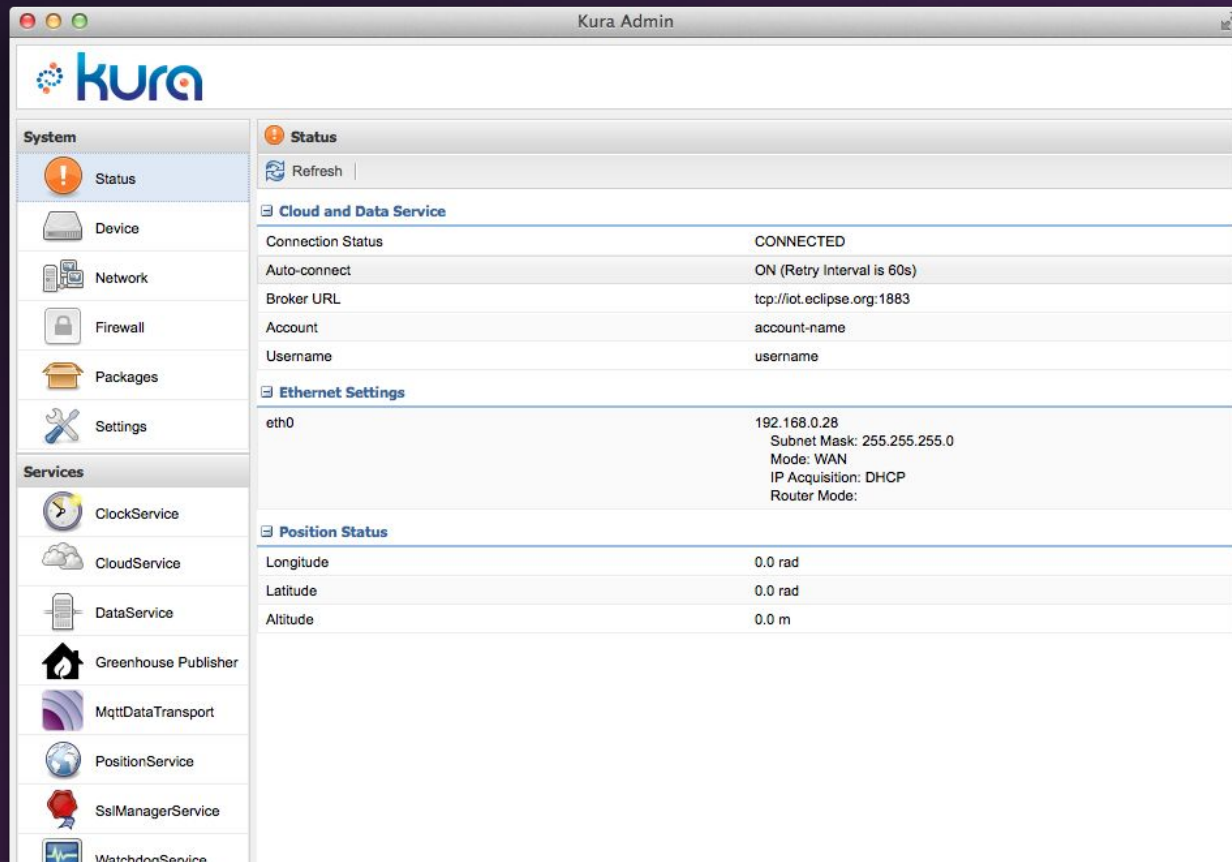
Installing Kura

```
cd ~  
  
sudo apt-get update  
  
wget https://s3.amazonaws.com/kura_downloads/raspbian/release/  
      \1.3.0/kura_1.3.0_raspberry-pi-2_installer.deb  
  
sudo dpkg -i kura_1.3.0_raspberry-pi-2_installer.deb  
  
sudo apt-get install -f  
  
sudo reboot
```

First steps with Kura

- Network management
 - Cellular Modem, WiFi
 - Firewall
 - NAT
- OSGi and system administration
- IoT server communication settings

First steps with Kura



The screenshot displays the Kura Admin web interface. The left sidebar contains a 'System' section with links to Status, Device, Network, Firewall, Packages, and Settings. Below this is a 'Services' section listing various services like ClockService, CloudService, DataService, Greenhouse Publisher, MqttDataTransport, PositionService, SslManagerService, and WatchdogService. The main content area shows the 'Status' page with a 'Refresh' button and three expandable sections: 'Cloud and Data Service', 'Ethernet Settings', and 'Position Status'.

System

- Status
- Device
- Network
- Firewall
- Packages
- Settings

Services

- ClockService
- CloudService
- DataService
- Greenhouse Publisher
- MqttDataTransport
- PositionService
- SslManagerService
- WatchdogService

Status

Refresh

Cloud and Data Service

Connection Status	CONNECTED
Auto-connect	ON (Retry Interval is 60s)
Broker URL	tcp://iot.eclipse.org:1883
Account	account-name
Username	username

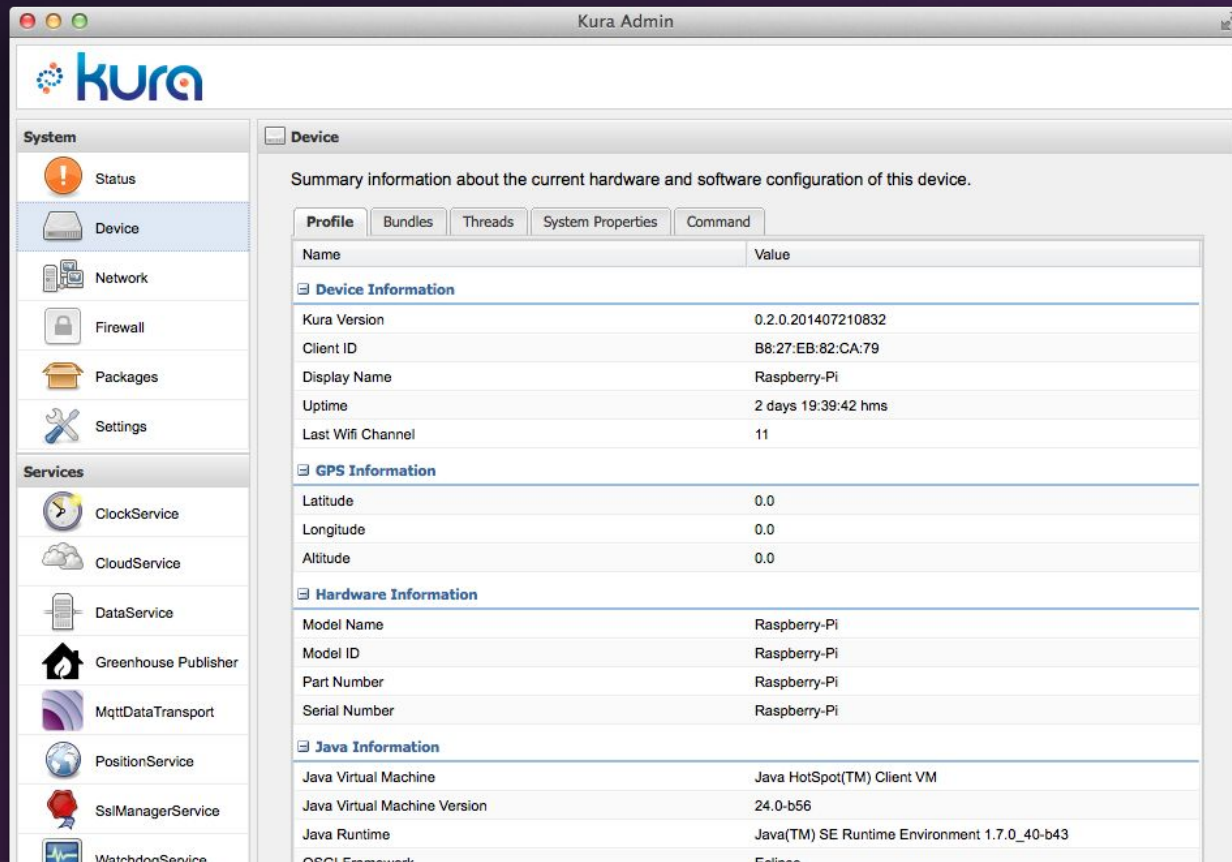
Ethernet Settings

eth0	192.168.0.28 Subnet Mask: 255.255.255.0 Mode: WAN IP Acquisition: DHCP Router Mode:
------	---

Position Status

Longitude	0.0 rad
Latitude	0.0 rad
Altitude	0.0 m

First steps with Kura



The screenshot displays the Kura Admin web interface. The left sidebar contains a 'System' section with icons for Status, Device (selected), Network, Firewall, Packages, and Settings. Below this is a 'Services' section with icons for ClockService, CloudService, DataService, Greenhouse Publisher, MqttDataTransport, PositionService, SslManagerService, and WatchdogService. The main content area is titled 'Device' and shows a summary of hardware and software configuration. It includes tabs for Profile, Bundles, Threads, System Properties, and Command. The 'Profile' tab is active, displaying a table of device information categorized into Device Information, GPS Information, Hardware Information, and Java Information.

Kura Admin

System

- Status
- Device**
- Network
- Firewall
- Packages
- Settings

Services

- ClockService
- CloudService
- DataService
- Greenhouse Publisher
- MqttDataTransport
- PositionService
- SslManagerService
- WatchdogService

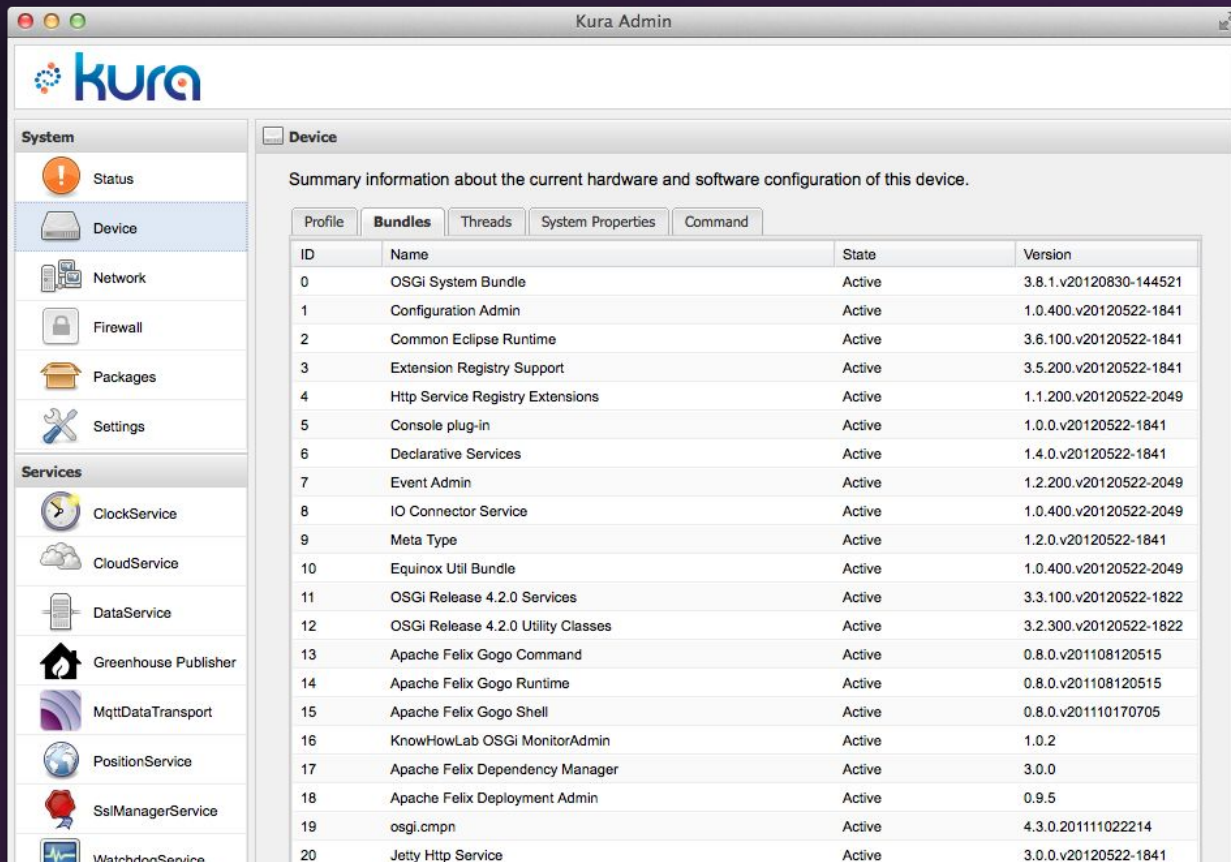
Device

Summary information about the current hardware and software configuration of this device.

Profile Bundles Threads System Properties Command

Name	Value
Device Information	
Kura Version	0.2.0.201407210832
Client ID	B8:27:EB:82:CA:79
Display Name	Raspberry-Pi
Uptime	2 days 19:39:42 hms
Last Wifi Channel	11
GPS Information	
Latitude	0.0
Longitude	0.0
Altitude	0.0
Hardware Information	
Model Name	Raspberry-Pi
Model ID	Raspberry-Pi
Part Number	Raspberry-Pi
Serial Number	Raspberry-Pi
Java Information	
Java Virtual Machine	Java HotSpot(TM) Client VM
Java Virtual Machine Version	24.0-b56
Java Runtime	Java(TM) SE Runtime Environment 1.7.0_40-b43
OS	FreeBSD

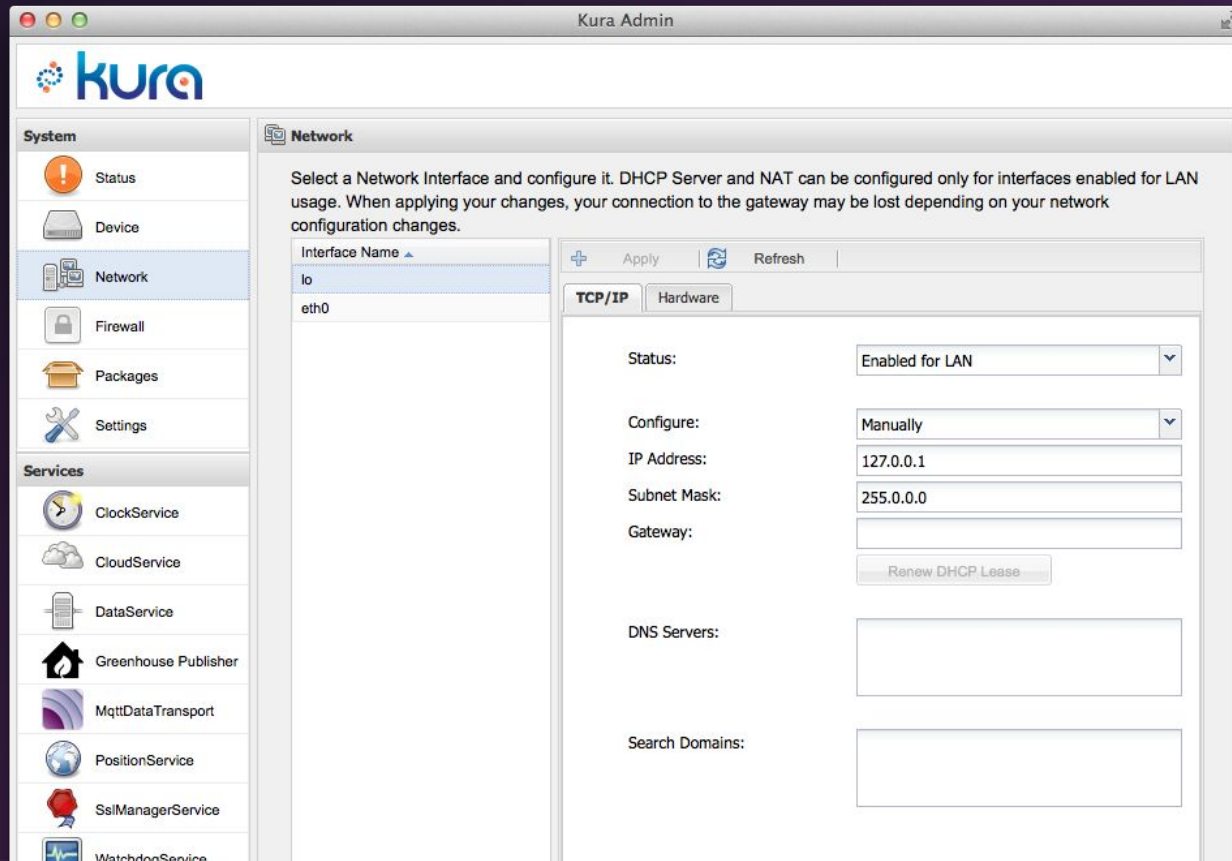
First steps with Kura



The screenshot displays the Kura Admin web interface. On the left is a sidebar with navigation links for System (Status, Device, Network, Firewall, Packages, Settings) and Services (ClockService, CloudService, DataService, Greenhouse Publisher, MqttDataTransport, PositionService, SslManagerService, WatchdogService). The main content area is titled 'Device' and shows a summary of hardware and software configuration. A tabbed interface at the top of the main area includes 'Profile', 'Bundles' (selected), 'Threads', 'System Properties', and 'Command'. The 'Bundles' tab contains a table of installed bundles.

ID	Name	State	Version
0	OSGi System Bundle	Active	3.8.1.v20120830-144521
1	Configuration Admin	Active	1.0.400.v20120522-1841
2	Common Eclipse Runtime	Active	3.6.100.v20120522-1841
3	Extension Registry Support	Active	3.5.200.v20120522-1841
4	Http Service Registry Extensions	Active	1.1.200.v20120522-2049
5	Console plug-in	Active	1.0.0.v20120522-1841
6	Declarative Services	Active	1.4.0.v20120522-1841
7	Event Admin	Active	1.2.200.v20120522-2049
8	IO Connector Service	Active	1.0.400.v20120522-2049
9	Meta Type	Active	1.2.0.v20120522-1841
10	Equinox Util Bundle	Active	1.0.400.v20120522-2049
11	OSGi Release 4.2.0 Services	Active	3.3.100.v20120522-1822
12	OSGi Release 4.2.0 Utility Classes	Active	3.2.300.v20120522-1822
13	Apache Felix Gogo Command	Active	0.8.0.v201108120515
14	Apache Felix Gogo Runtime	Active	0.8.0.v201108120515
15	Apache Felix Gogo Shell	Active	0.8.0.v201110170705
16	KnowHowLab OSGi MonitorAdmin	Active	1.0.2
17	Apache Felix Dependency Manager	Active	3.0.0
18	Apache Felix Deployment Admin	Active	0.9.5
19	osgi.cmpn	Active	4.3.0.201111022214
20	Jetty Http Service	Active	3.0.0.v20120522-1841

First steps with Kura



The screenshot displays the Kura Admin web interface. The left sidebar contains a 'System' section with links to Status, Device, Network (selected), Firewall, Packages, and Settings. Below this is a 'Services' section with links to ClockService, CloudService, DataService, Greenhouse Publisher, MqttDataTransport, PositionService, SslManagerService, and WatchdogService. The main content area is titled 'Network' and includes a warning message: 'Select a Network Interface and configure it. DHCP Server and NAT can be configured only for interfaces enabled for LAN usage. When applying your changes, your connection to the gateway may be lost depending on your network configuration changes.' A table lists network interfaces: 'lo' (selected) and 'eth0'. To the right of the table are 'Apply' and 'Refresh' buttons. Below the table are two tabs: 'TCP/IP' (active) and 'Hardware'. The 'TCP/IP' tab shows configuration fields for 'Status' (set to 'Enabled for LAN'), 'Configure' (set to 'Manually'), 'IP Address' (127.0.0.1), 'Subnet Mask' (255.0.0.0), and 'Gateway' (empty). There is a 'Renew DHCP Lease' button. At the bottom, there are fields for 'DNS Servers' and 'Search Domains'.

Kura Admin

kura

System

- Status
- Device
- Network**
- Firewall
- Packages
- Settings

Services

- ClockService
- CloudService
- DataService
- Greenhouse Publisher
- MqttDataTransport
- PositionService
- SslManagerService
- WatchdogService

Network

Select a Network Interface and configure it. DHCP Server and NAT can be configured only for interfaces enabled for LAN usage. When applying your changes, your connection to the gateway may be lost depending on your network configuration changes.

Interface Name
lo
eth0

TCP/IP | Hardware

Apply | Refresh

Status: Enabled for LAN

Configure: Manually

IP Address: 127.0.0.1

Subnet Mask: 255.0.0.0

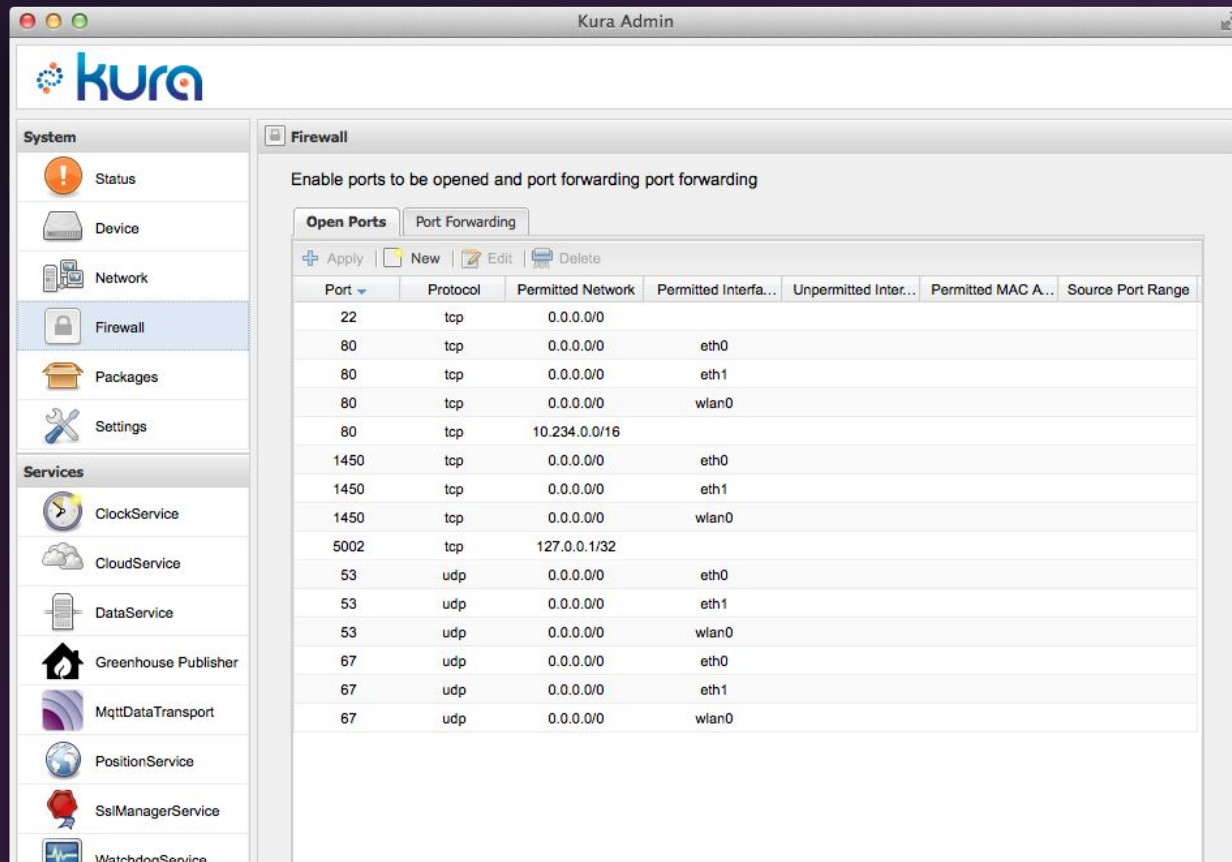
Gateway:

Renew DHCP Lease

DNS Servers:

Search Domains:

First steps with Kura



The screenshot shows the Kura Admin web interface. The left sidebar contains a 'System' section with links to Status, Device, Network, Firewall (selected), Packages, and Settings. Below this is a 'Services' section with links to ClockService, CloudService, DataService, Greenhouse Publisher, MqttDataTransport, PositionService, SslManagerService, and WatchdogService. The main content area is titled 'Firewall' and includes the instruction 'Enable ports to be opened and port forwarding port forwarding'. It features two tabs: 'Open Ports' (active) and 'Port Forwarding'. The 'Open Ports' tab contains a table with columns: Port, Protocol, Permitted Network, Permitted Interface, Unpermitted Interface, Permitted MAC Address, and Source Port Range. The table lists several open ports, including 22, 80, 1450, 5002, 53, 67, and 127.0.0.1/32, with their respective protocols and permitted networks/interfaces.

Kura Admin

kura

System

- Status
- Device
- Network
- Firewall
- Packages
- Settings

Services

- ClockService
- CloudService
- DataService
- Greenhouse Publisher
- MqttDataTransport
- PositionService
- SslManagerService
- WatchdogService

Firewall

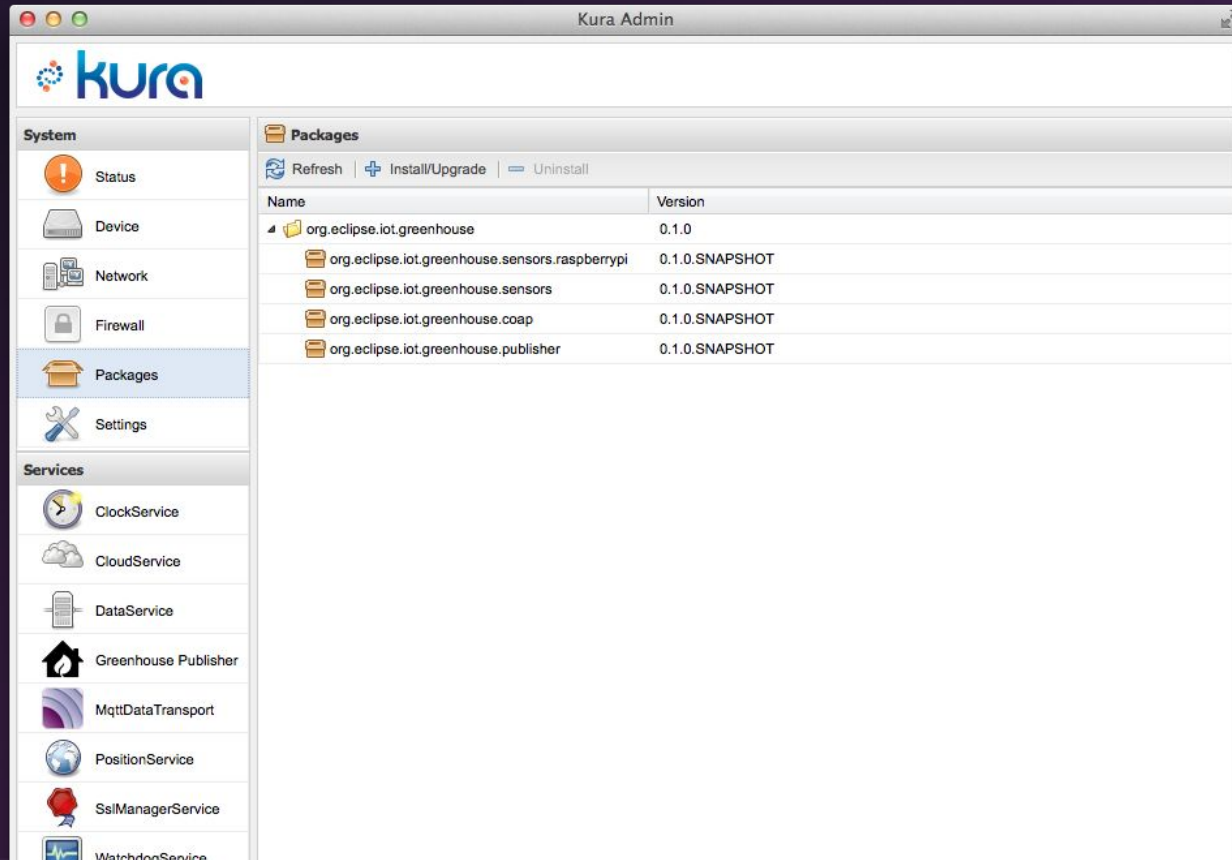
Enable ports to be opened and port forwarding port forwarding

Open Ports Port Forwarding

+ Apply New Edit Delete

Port	Protocol	Permitted Network	Permitted Interface	Unpermitted Interface	Permitted MAC Address	Source Port Range
22	tcp	0.0.0.0/0				
80	tcp	0.0.0.0/0	eth0			
80	tcp	0.0.0.0/0	eth1			
80	tcp	0.0.0.0/0	wlan0			
80	tcp	10.234.0.0/16				
1450	tcp	0.0.0.0/0	eth0			
1450	tcp	0.0.0.0/0	eth1			
1450	tcp	0.0.0.0/0	wlan0			
5002	tcp	127.0.0.1/32				
53	udp	0.0.0.0/0	eth0			
53	udp	0.0.0.0/0	eth1			
53	udp	0.0.0.0/0	wlan0			
67	udp	0.0.0.0/0	eth0			
67	udp	0.0.0.0/0	eth1			
67	udp	0.0.0.0/0	wlan0			

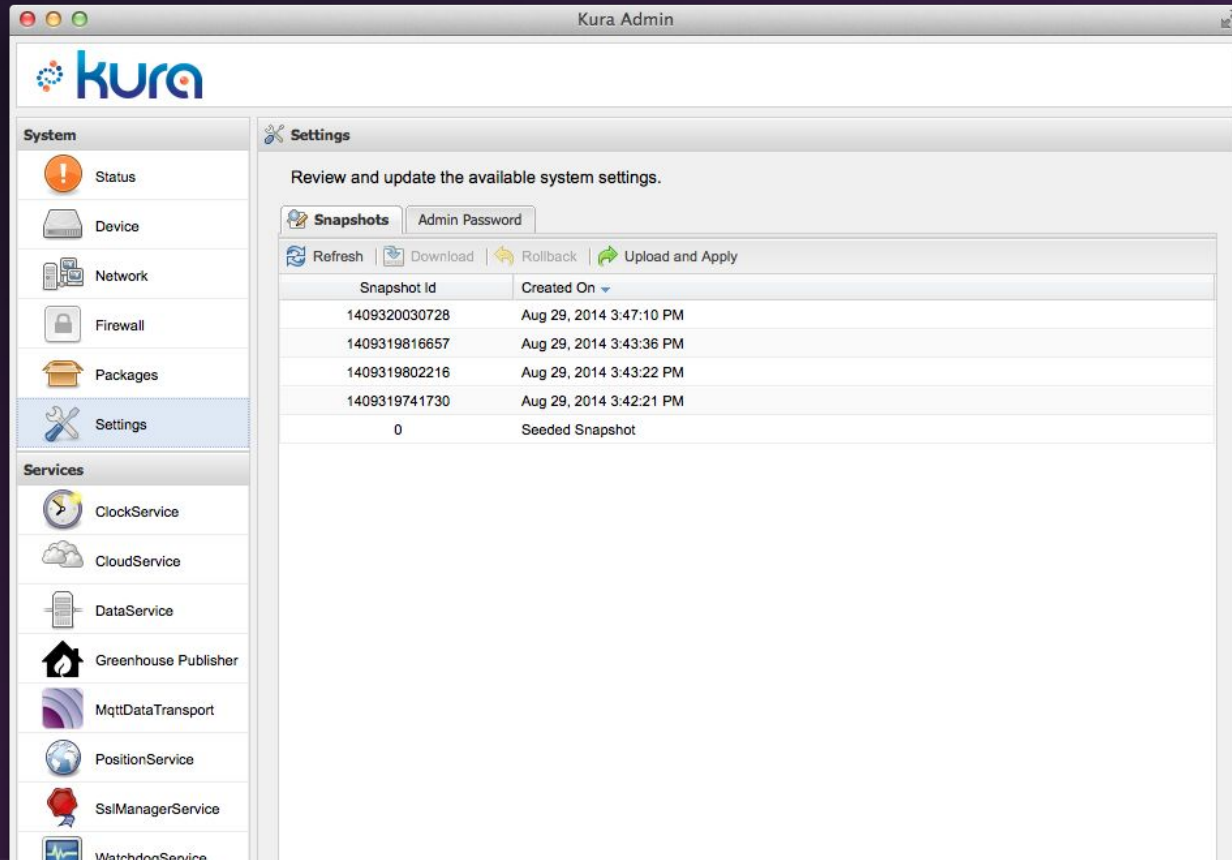
First steps with Kura



The screenshot displays the Kura Admin web interface. The top navigation bar includes the Kura logo and a title bar labeled "Kura Admin". The left sidebar contains a "System" section with links to Status, Device, Network, Firewall, Packages (highlighted), and Settings. Below this is a "Services" section with links to ClockService, CloudService, DataService, Greenhouse Publisher, MqttDataTransport, PositionService, SslManagerService, and WatchdogService. The main content area is titled "Packages" and features a table of installed packages. Above the table are buttons for "Refresh", "Install/Upgrade", and "Uninstall".

Name	Version
org.eclipse.iot.greenhouse	0.1.0
org.eclipse.iot.greenhouse.sensors.raspberrypi	0.1.0.SNAPSHOT
org.eclipse.iot.greenhouse.sensors	0.1.0.SNAPSHOT
org.eclipse.iot.greenhouse.coap	0.1.0.SNAPSHOT
org.eclipse.iot.greenhouse.publisher	0.1.0.SNAPSHOT

First steps with Kura



The screenshot displays the Kura Admin web interface. The left sidebar contains two main sections: 'System' and 'Services'. The 'System' section includes links for Status, Device, Network, Firewall, Packages, and Settings (which is currently selected). The 'Services' section includes links for ClockService, CloudService, DataService, Greenhouse Publisher, MqttDataTransport, PositionService, SslManagerService, and WatchdogService. The main content area is titled 'Settings' and contains a sub-section for 'Snapshots'. This section includes a table with columns for 'Snapshot Id' and 'Created On'. The table lists four snapshots with their respective IDs and creation times, followed by a 'Seeded Snapshot' with ID 0.

Kura Admin

System

- Status
- Device
- Network
- Firewall
- Packages
- Settings**

Services

- ClockService
- CloudService
- DataService
- Greenhouse Publisher
- MqttDataTransport
- PositionService
- SslManagerService
- WatchdogService

Settings

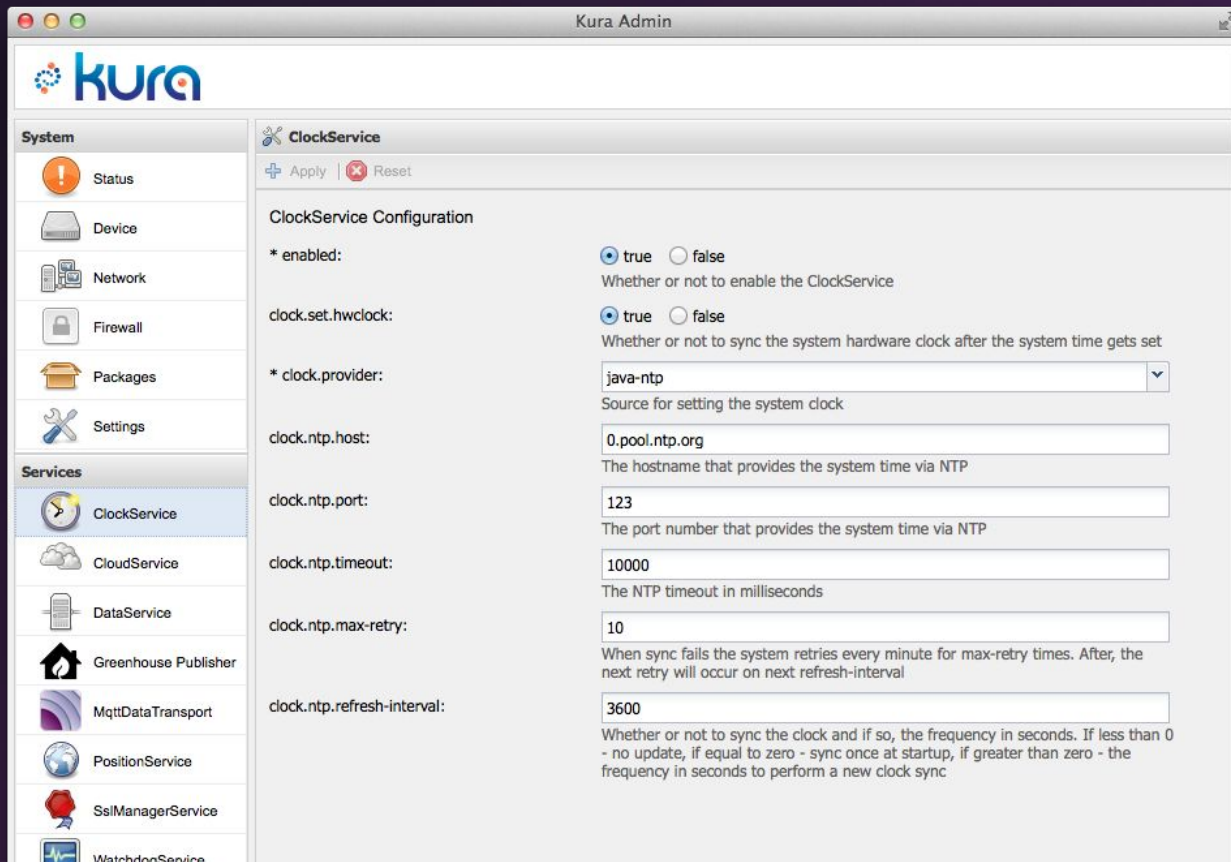
Review and update the available system settings.

Snapshots | Admin Password

Refresh | Download | Rollback | Upload and Apply

Snapshot Id	Created On
1409320030728	Aug 29, 2014 3:47:10 PM
1409319816657	Aug 29, 2014 3:43:36 PM
1409319802216	Aug 29, 2014 3:43:22 PM
1409319741730	Aug 29, 2014 3:42:21 PM
0	Seeded Snapshot

First steps with Kura

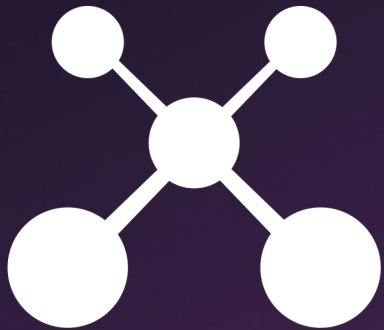


The screenshot displays the Kura Admin web interface. The top navigation bar includes the Kura logo and a 'System' menu. The left sidebar lists various system components: Status, Device, Network, Firewall, Packages, Settings, and Services. The 'Services' section is expanded, showing 'ClockService' as the selected option. The main content area is titled 'ClockService Configuration' and contains several settings:

- * enabled:** A radio button set with 'true' selected and 'false' unselected. Below it, a description states: 'Whether or not to enable the ClockService'.
- clock.set.hwclock:** A radio button set with 'true' selected and 'false' unselected. Below it, a description states: 'Whether or not to sync the system hardware clock after the system time gets set'.
- * clock.provider:** A dropdown menu with 'java-ntp' selected. Below it, a description states: 'Source for setting the system clock'.
- clock.ntp.host:** A text input field containing '0.pool.ntp.org'. Below it, a description states: 'The hostname that provides the system time via NTP'.
- clock.ntp.port:** A text input field containing '123'. Below it, a description states: 'The port number that provides the system time via NTP'.
- clock.ntp.timeout:** A text input field containing '10000'. Below it, a description states: 'The NTP timeout in milliseconds'.
- clock.ntp.max-retry:** A text input field containing '10'. Below it, a description states: 'When sync fails the system retries every minute for max-retry times. After, the next retry will occur on next refresh-interval'.
- clock.ntp.refresh-interval:** A text input field containing '3600'. Below it, a description states: 'Whether or not to sync the clock and if so, the frequency in seconds. If less than 0 - no update, if equal to zero - sync once at startup, if greater than zero - the frequency in seconds to perform a new clock sync'.

Kura API

- OSGi services that you can re-use in your own components
 - ClockService
 - DataService, CloudService
 - CryptoService (AES, base64, SHA-1)
 - PositionService (geolocation)
 - ... and many others
- And of course you can leverage a huge ecosystem of Java and OSGi libraries



CONNECT



MANAGE



VISUALIZE

End-user interaction

- JavaFX Charts
- Eclipse BIRT
- Smartphone app (e.g Android)
 - <https://www.eclipse.org/paho/clients/android>
- MQTT + WebSockets = ♥
 - <https://www.eclipse.org/paho/clients/js>

MQTT + WebSockets

```
var client = new Paho.MQTT.Client("ws://iot.eclipse.org/ws",  
                                  "client-" + new Date().getTime());  
client.onMessageArrived = function(message) {  
    // my stuff  
}  
client.connect({  
    onSuccess: function() {  
        client.subscribe("myRootTopic/#");  
    }  
});
```

DEMO!

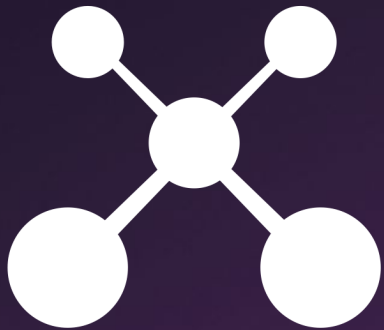
Data Analytics for IoT?



Apache Spark Streaming

- Stream processing in Java, Python & Scala
- Built-in connectors for Kafka, Twitter, ZeroMQ, Flume, Kinesis & ... MQTT!
- A nice programming model for consolidating time-series data
- Awesome combo when used with Spark MLlib!

DEMO!



CONNECT



MANAGE



VISUALIZE



Eclipse IoT is also...

Industrial IoT

- Open source implementations of IEC standards
- Eclipse SCADA, 4DIAC, Rise V2G, ...

Eclipse IoT is also...

Device Management

- **LWM2M** is an Open Mobile Alliance Standard
- Device Management on top of CoAP
- Eclipse Leshan and Wakaama are two implementations

Eclipse IoT is also...

Secured Service Discovery



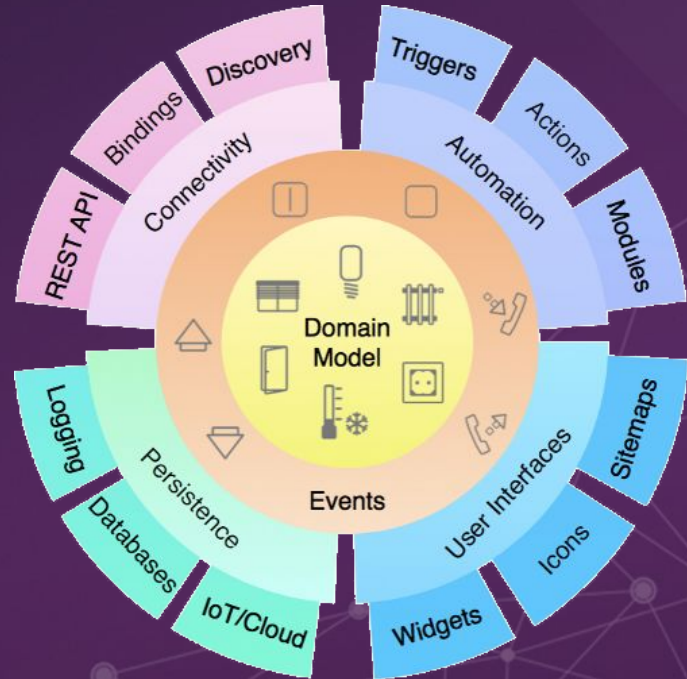
VERISIGN™

- **Eclipse Tiaki**
- Leveraging DNS-SEC and DNS-SD for retrieving a device configuration parameter, or its public key for establishing secured communications

Eclipse IoT is also...



- Flexible Framework
- Based on **Java** and **OSGi**
- Huge number of “bindings”:
KNX, Nest, Philips HUE, ...



If you had to remember only 3 things...

#1

Kura is awesome!
Go download it now!

<http://eclipse.org/kura>



If you had to remember only 3 things...

#2

Build your own greenhouse &
follow the tutorial

<http://iot.eclipse.org/java/tutorial>



If you had to remember only 3 things...

#3

Eclipse IoT is much more than
Kura and Java!

<http://iot.eclipse.org/>



Get Involved!



- Open bugs / fix bugs
- Request new features
- Write articles, tutorials
- Participate on the mailing lists
- Propose your project!

One more thing...



One more thing...



OPEN IOT **CHALLENGE** 2.0

Apply before November 23!

<http://iot.eclipse.org/open-iot-challenge>

Thank you! Questions?

benjamin@eclipse.org

@kartben

<http://blog.benjamin-cabe.com>

<http://iot.eclipse.org>