Kazoo Commission User Guide

Thomas Newman

# About

## Motivation

Voice over IP (VoIP) technology replaces traditional analog phones with IP phones that communicate to the Plain Old Telephone Service (POTS) though the use of PBX (Private Branch Exchange) software. 2600hz Kazoo provides a fault-tolerant PBX that can be scaled to tens of thousands of calls on commodity hardware through the use of the Erlang programming language and the CouchDB database. Once a phone system is set up using 2600hz Kazoo, the phones must be manually provisioned, which is problematic and extremely costly not only because configuration changes require reprogramming all impacted phones in the field but also because the manual process is complicated, error-prone and requires a highly skilled technician to provision the phones. Traditional approaches, such as TFTP and HTTP provisioning, are insecure as the phone is not authenticated by the server, and the connection to the server is not encrypted, allowing phone configurations to be stolen, forged and used for nefarious purposes.

## Solution

Kazoo Commission simplifies the phone provisioning process by automatically generating a phone’s configuration simply by programming a URL containing the hostname of the Kazoo Commission server along with the phone’s manufacturer, model, associated account and MAC address into the phone. The phone retrieves a dynamically-generated configuration from the Kazoo Commission server. To prevent credentials to be intercepted or forged, the Kazoo Commission server uses HTTP over Transport Layer Security (HTTPS). To ensure a phone (or attacker) cannot retrieve the configuration of other phones, Kazoo Commission requires the phone to provide a certificate signed by a trusted root certificate authority. If the certificate is valid and trusted, Kazoo Commission will only send the configuration to the phone after verifying that the certificate’s subject field contains the requested MAC address. Kazoo Commission is provided as a Docker container to ease deployment.

# System Architecture



# System Workflow

This section describes how Kazoo Commission works:

1. The certificate for the Root Certificate Authority used for SSL on the web server is installed/trusted on the phone to be provisioned.
2. The URL to the Kazoo Commission server is installed on the phone.
3. The phone and Nginx web server running on the Kazoo Commission server perform an SSL handshake. The handshake will only succeed if the phone trusts the SSL certificate used by Nginx, and Nginx trusts the SSL certificate used by the client.
4. The phone makes a request to Nginx for its configuration. Nginx forwards this request to Kazoo Commission, which is running inside a Docker container. The phone manufacturer, phone model, account name and MAC address used to generate the phone’s configuration are included in the URL. Nginx forwards the subject from the phone’s SSL certificate.
5. Kazoo Commission verifies that the phone’s SSL subject matches the requested MAC address.
6. Kazoo Commission queries the Kazoo CouchDB database to retrieve configuration data for the phone based on the information included in the URL.
7. Kazoo Commission generates a configuration file that is returned to the phone based on the Kazoo CouchDB database query and information included in the URL.
8. The phone installs the configuration file and is now fully provisioned. The phone will periodically poll Kazoo Commission for new configuration.

# Prerequisites

The following prerequisites are required before Kazoo Commission can be installed and used:

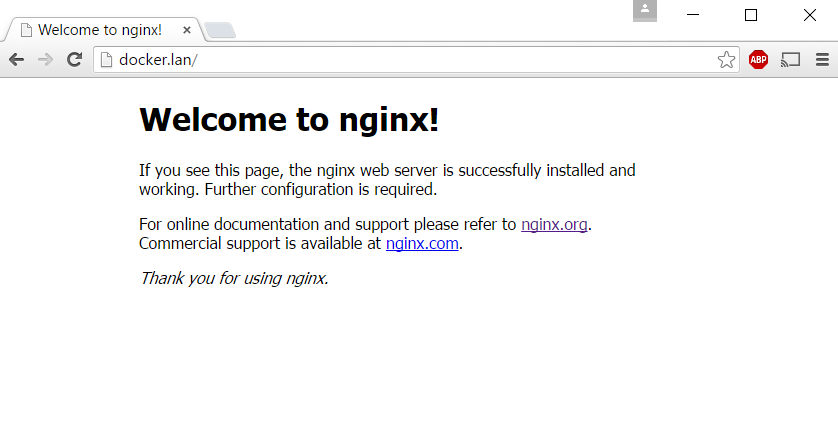
1. A 2600hz Kazoo server, which can be installed using the instructions at <https://2600hz.atlassian.net/wiki/display/Dedicated/via+RPM>
2. A new installation of Ubuntu Server, which can be obtained from <http://www.ubuntu.com/download/server>
   1. The Latest version is recommended over the LTS (Long Term Support) version

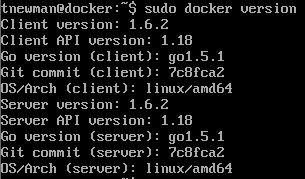
# Installation Instructions

## Install Docker and nginx

1. Log into the Ubuntu command line with a user that has root privileges (sudo)
2. Update system packages, install Nginx and install Docker with   
   **sudo apt-get update -y && \  
   sudo apt-get upgrade -y && \  
   sudo apt-get -y install nginx docker.io**
3. Run nginx and enable it on system startup with  
   **sudo systemctl enable nginx && \**

**sudo systemctl start nginx**

1. Reboot the server with  
   **sudo reboot**
2. Verify that nginx was installed successfully by visiting the server’s hostname or IP address in a web browser. A “Welcome to nginx” page will appear if nginx was installed correctly.  
   
3. Verify that Docker was installed successfully with  
   **sudo docker version**



## Install Kazoo Commission

1. Clone the Kazoo Commission GitHub repository with  
   **git clone https://github.com/tnewman/KazooCommission.git**
2. Build the Kazoo Commission Docker Container with

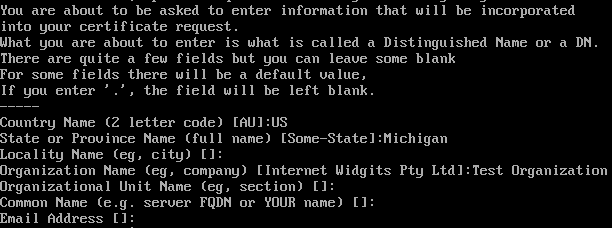
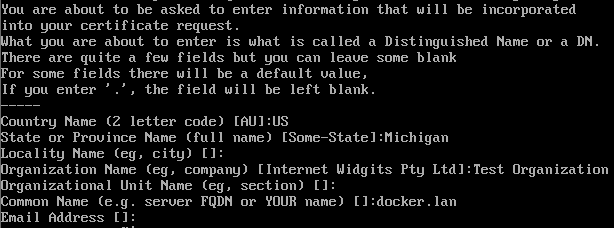
**sudo docker build -t kazoocommission KazooCommission**

## SSL Certificates

Creating certificate authorities is an advanced topic. This guide assumes that you will create a single root certificate authority and use it to sign individual certificates. If you have different requirements, skip this section, and follow the remaining sections of the guide. You should not proceed without fully understanding the security implications of managing a private root certificate authority.

1. Create a key for your root certificate authority with

**sudo openssl genrsa -aes256 -out /etc/nginx/rootCA.key 4096**

1. Make sure to remember the passphrase from step 1 as it will be needed to sign certificate requests later. This passphrase must be secure to avoid compromising the root certificate authority.
2. Create a certificate signing request for your root certificate authority with  
   **sudo openssl req -x509 -new -key /etc/nginx/rootCA.key -days 1024 -out /etc/nginx/rootCA.pem**
3. Fill in the fields when prompted. Only the country, state and organization are mandatory. These should be filled in according to your needs.  
   
4. Create a key for your certificate with  
   **sudo openssl genrsa -out /etc/nginx/kazoocommission.key 2048**
5. Create a certificate signing request for your certificate with  
   **sudo openssl req -new -key /etc/nginx/kazoocommission.key -out /etc/nginx/kazoocommission.csr**
6. Fill in the fields when prompted. The country, state and organization are mandatory and should be filled in according to your needs. The common name field should be filled in with the server’s hostname or IP address, depending on which one will be used to access the server.  
   
7. Sign the certificate signing request with the root certificate authority key with  
   **sudo openssl x509 -req -in /etc/nginx/kazoocommission.csr -CA /etc/nginx/rootCA.pem -CAkey /etc/nginx/rootCA.key -CAcreateserial -days 366 -out /etc/nginx/kazoocommission.pem**
8. Combine the certificate and root certificate into a certificate bundle with  
   **sudo bash -c "cat /etc/nginx/kazoocommission.pem /etc/nginx/rootCA.pem > /etc/nginx/kazoocommissionbundle.pem"**
9. Copy the root certificate to the nginx html folder so that clients can download it with  
   **sudo cp /etc/nginx/rootCA.pem /var/www/html/kazooCommissionCA.cer**

## Client Root CA

## Create a file at **/etc/nginx/clientca.pem** using a text editor containing the following Cisco Root CA. If you want to support other CAs for clients, they must be added to this file:

**-----BEGIN CERTIFICATE-----**

**MIIF7zCCBNegAwIBAgIRANB9jBXAuny2RGmYseqJh58wDQYJKoZIhvcNAQEFBQAw**

**gewxCzAJBgNVBAYTAlVTMRMwEQYDVQQIEwpDYWxpZm9ybmlhMREwDwYDVQQHEwhT**

**YW4gSm9zZTEdMBsGA1UEChMUQ2lzY28gU21hbGwgQnVzaW5lc3MxMzAxBgNVBAsT**

**KkNpc2NvIFNtYWxsIEJ1c2luZXNzIENlcnRpZmljYXRlIEF1dGhvcml0eTE1MDMG**

**A1UEAxMsQ2lzY28gU21hbGwgQnVzaW5lc3MgQ2xpZW50IFJvb3QgQXV0aG9yaXR5**

**IDIxKjAoBgkqhkiG9w0BCQEWG2Npc2Nvc2ItY2VydGFkbWluQGNpc2NvLmNvbTAe**

**Fw0xMzA4MDIyMjM3NDNaFw0zNTA2MjgyMjM3NDNaMIHsMQswCQYDVQQGEwJVUzET**

**MBEGA1UECBMKQ2FsaWZvcm5pYTERMA8GA1UEBxMIU2FuIEpvc2UxHTAbBgNVBAoT**

**FENpc2NvIFNtYWxsIEJ1c2luZXNzMTMwMQYDVQQLEypDaXNjbyBTbWFsbCBCdXNp**

**bmVzcyBDZXJ0aWZpY2F0ZSBBdXRob3JpdHkxNTAzBgNVBAMTLENpc2NvIFNtYWxs**

**IEJ1c2luZXNzIENsaWVudCBSb290IEF1dGhvcml0eSAyMSowKAYJKoZIhvcNAQkB**

**FhtjaXNjb3NiLWNlcnRhZG1pbkBjaXNjby5jb20wggEiMA0GCSqGSIb3DQEBAQUA**

**A4IBDwAwggEKAoIBAQC/wvg65saJIYyCoHmRc3LzdNWoTqc9ewKrayyNcYICdnr6**

**vy6M57BHFZarg49IDefnFfLtVC7NfeM2NPbrBaPVOVcuau6yCrd7pt2C6WqUAS+J**

**HVKT9OwjCK5vBAqUXZKU1jrEWGnaKy5kz3cOKYLDvn166/j00VwYd4WkXugeUfbU**

**efHhyER8Z62c95uAdB8yBXnD1WdB3xyAmhBXgJt+q+ZQJIJCBs/fNH0K6XBW3G8K**

**xRsyevDhcy4h1JJ61lOWg7ONgrx/XgPt6X5jObs3CsYyx/7bP7CKAoWDeCqHMlqx**

**gv843w1LgzGOr3jmeUaUji7DGDQ2MZC2OokeBhpnAgMBAAGjggGIMIIBhDAdBgNV**

**HQ4EFgQU+MIzZ6kS/F1DI55V035XQBpVQhAwggErBgNVHSMEggEiMIIBHoAU+MIz**

**Z6kS/F1DI55V035XQBpVQhChgfKkge8wgewxCzAJBgNVBAYTAlVTMRMwEQYDVQQI**

**EwpDYWxpZm9ybmlhMREwDwYDVQQHEwhTYW4gSm9zZTEdMBsGA1UEChMUQ2lzY28g**

**U21hbGwgQnVzaW5lc3MxMzAxBgNVBAsTKkNpc2NvIFNtYWxsIEJ1c2luZXNzIENl**

**cnRpZmljYXRlIEF1dGhvcml0eTE1MDMGA1UEAxMsQ2lzY28gU21hbGwgQnVzaW5l**

**c3MgQ2xpZW50IFJvb3QgQXV0aG9yaXR5IDIxKjAoBgkqhkiG9w0BCQEWG2Npc2Nv**

**c2ItY2VydGFkbWluQGNpc2NvLmNvbYIRANB9jBXAuny2RGmYseqJh58wDAYDVR0T**

**BAUwAwEB/zARBglghkgBhvhCAQEEBAMCAgQwEwYDVR0lBAwwCgYIKwYBBQUHAwIw**

**DQYJKoZIhvcNAQEFBQADggEBAJiVNjWYUSaSZsbbza0aqX8SLALDNihPBSDzhaKh**

**901sS2hHCm/5827658/MV6V/YNbZun/zgRbi18WDDBqEgiSaq18gXCEmJLdtA1+t**

**jhCbjCuabLygDE1cUtcAu/+5c6oXaZjKpUx5vJ5zSLG1wZDYiIn0olW7eGvokTcZ**

**Pzd9IMTqwfMX8U9JtW3+8yQ78YSY0A70JL1+54bub/99bEn6dU3Z6/h8H809wxYz**

**IziMlnJiUC1v6mgMprq7Dgj1XenA0sm+865zrmO69o006WCxbqL4y4v9AyzBkeBF**

**EuYmmIpRFm9cNiBv/ZmWOnuLsVYs3reR7Da8FFbD32L91DY=**

**-----END CERTIFICATE-----**

**-----BEGIN CERTIFICATE-----**

**MIIEyjCCBDOgAwIBAgIQRb9IwM64j3vI4W2FYlpbjzANBgkqhkiG9w0BAQQFADCB**

**4jELMAkGA1UEBhMCVVMxEzARBgNVBAgTCkNhbGlmb3JuaWExETAPBgNVBAcTCFNh**

**biBKb3NlMSAwHgYDVQQKExdTaXB1cmEgVGVjaG5vbG9neSwgSW5jLjEwMC4GA1UE**

**CxMnU2lwdXJhIFRlY2hub2xvZ3kgQ2VydGlmaWNhdGUgQXV0aG9yaXR5MTIwMAYD**

**VQQDEylTaXB1cmEgVGVjaG5vbG9neSBDbGllbnQgUm9vdCBBdXRob3JpdHkgMTEj**

**MCEGCSqGSIb3DQEJARYUd2VibWFzdGVyQHNpcHVyYS5jb20wHhcNMDQwMjA3MjIy**

**OTU3WhcNMzQwMTMwMjIyOTU3WjCB4jELMAkGA1UEBhMCVVMxEzARBgNVBAgTCkNh**

**bGlmb3JuaWExETAPBgNVBAcTCFNhbiBKb3NlMSAwHgYDVQQKExdTaXB1cmEgVGVj**

**aG5vbG9neSwgSW5jLjEwMC4GA1UECxMnU2lwdXJhIFRlY2hub2xvZ3kgQ2VydGlm**

**aWNhdGUgQXV0aG9yaXR5MTIwMAYDVQQDEylTaXB1cmEgVGVjaG5vbG9neSBDbGll**

**bnQgUm9vdCBBdXRob3JpdHkgMTEjMCEGCSqGSIb3DQEJARYUd2VibWFzdGVyQHNp**

**cHVyYS5jb20wgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAOchzps50Rgb08dQ**

**tfyMcamdclwaZIz8/aZRxrJB7i/J7BPTm0yv7BqTQ2vELgBFKdJJFNv58RvwHyi0**

**U8Bj/IW0PfXpXDvnV7+15Bn8kz/s0Oqu3qpCCi36M49Cv2m5T84SNFImP/gB0lZp**

**cJ4BxWLWE5TyBtzirz7vKyrFVaX1AgMBAAGjggF9MIIBeTAdBgNVHQ4EFgQUTIM4**

**K53G42WrGVExpck1m1EKIyEwggEgBgNVHSMEggEXMIIBE4AUTIM4K53G42WrGVEx**

**pck1m1EKIyGhgeikgeUwgeIxCzAJBgNVBAYTAlVTMRMwEQYDVQQIEwpDYWxpZm9y**

**bmlhMREwDwYDVQQHEwhTYW4gSm9zZTEgMB4GA1UEChMXU2lwdXJhIFRlY2hub2xv**

**Z3ksIEluYy4xMDAuBgNVBAsTJ1NpcHVyYSBUZWNobm9sb2d5IENlcnRpZmljYXRl**

**IEF1dGhvcml0eTEyMDAGA1UEAxMpU2lwdXJhIFRlY2hub2xvZ3kgQ2xpZW50IFJv**

**b3QgQXV0aG9yaXR5IDExIzAhBgkqhkiG9w0BCQEWFHdlYm1hc3RlckBzaXB1cmEu**

**Y29tghBFv0jAzriPe8jhbYViWluPMAwGA1UdEwQFMAMBAf8wEQYJYIZIAYb4QgEB**

**BAQDAgIEMBMGA1UdJQQMMAoGCCsGAQUFBwMCMA0GCSqGSIb3DQEBBAUAA4GBAI7q**

**kIOEuZ/XjXdl4ELN0nFYI1FBXlLfEFVOTwMZQW4C2E/4zkt+byqVsn1VssL0/zcD**

**h+GwncOyZIq788J+wo9GsJ3pK9D0sYHUWiHwCxTRCdowpm5jCYv3n7mBj7WpDDSP**

**nm1uSlCS43FmhlbK4Pk8OV/jnNLWe2U1Iglv+qDp**

**-----END CERTIFICATE-----**

## Configure nginx

1. Append the following server block to the end of **http** section of **/etc/nginx/nginx.conf** using a text editor:

**server {**

**listen 80;**

**}**

**server {**

**listen 443;**

**ssl on;**

**ssl\_certificate /etc/nginx/kazoocommissionbundle.pem;**

**ssl\_certificate\_key /etc/nginx/kazoocommission.key;**

**ssl\_verify\_client on;**

**ssl\_client\_certificate /etc/nginx/clientca.pem;**

**location / {**

**proxy\_pass http://localhost:8000;**

**proxy\_set\_header X-SSL-Subject $ssl\_client\_s\_dn;**

**}**

**}**

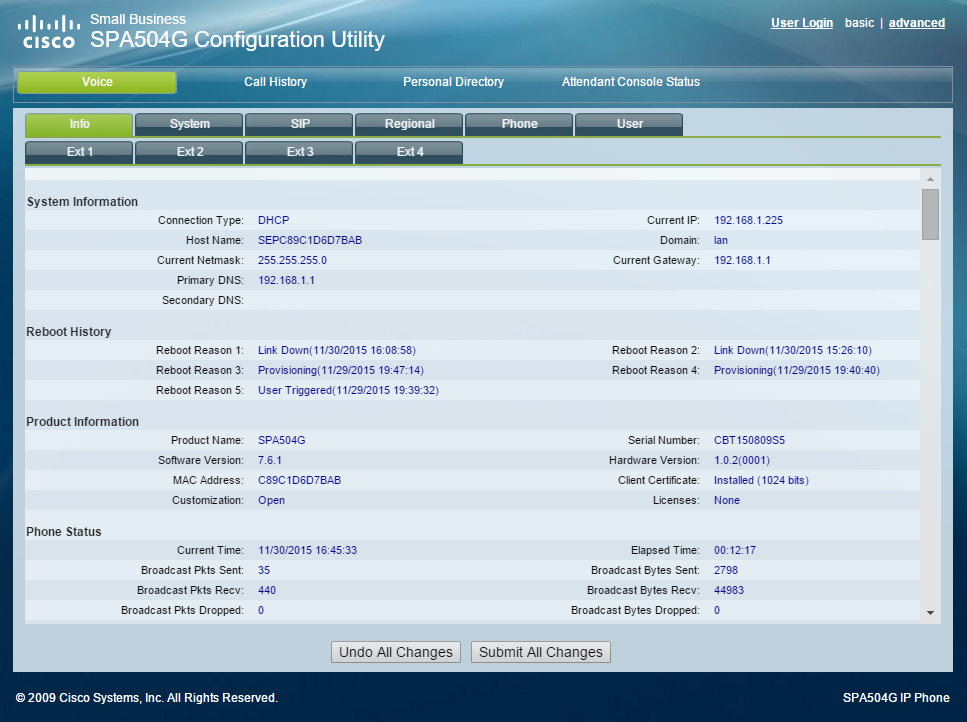
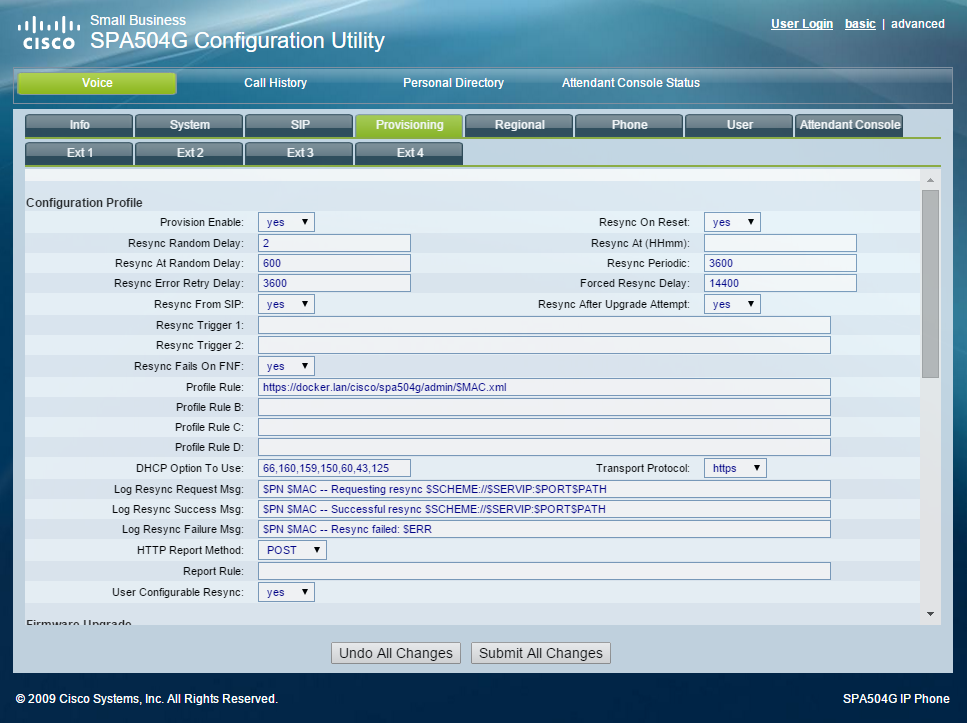
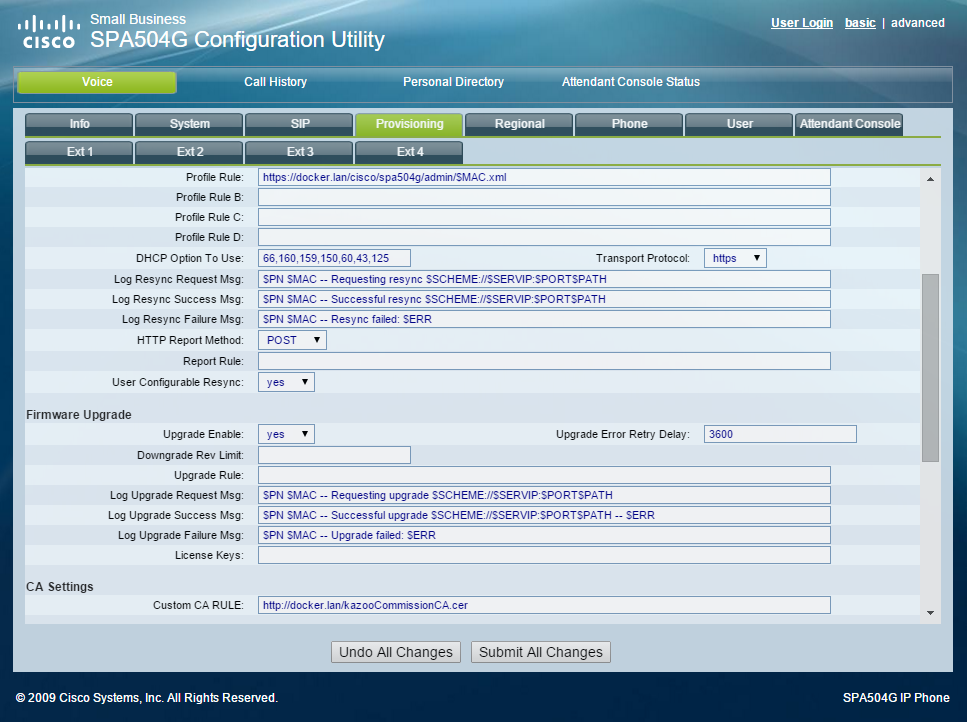
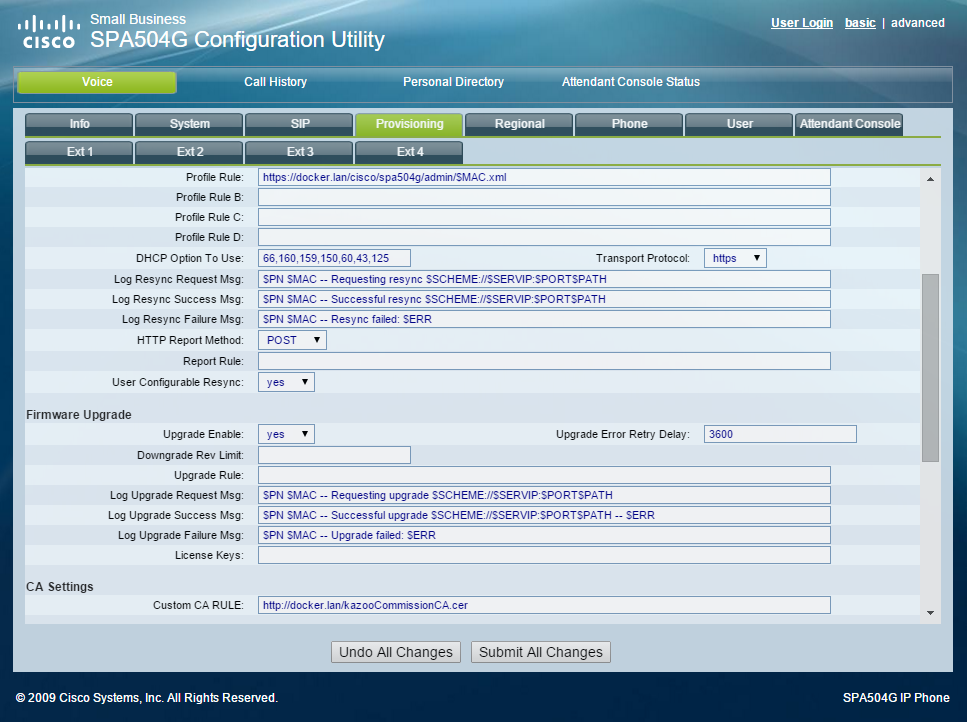
1. Restart nginx with  
   **sudo systemctl restart nginx**

## Configure Docker

1. Use the following command to run the Kazoo Commission Docker Container, making sure to set KAZOO\_COUCH\_DB\_URL to the URL of the database for Kazoo and KAZOO\_COMMISSION\_SIP\_OUTBOUND\_PROXY to the hostname of the SIP server that phones should connect to:  
   **sudo docker run --restart=always -d -e "KAZOO\_COUCH\_DB\_URL=http://fedora.lan:5984" -e "KAZOO\_COMMISSION\_SIP\_OUTBOUND\_PROXY=fedora.lan" -e "SIP\_DNS\_A\_RECORD=SIP\_DNS\_A\_RECORD" -p 8000:8000 kazoocommission**
2. Kazoo Commission can now be used to auto provision phones.

# Configuring Phones

## Cisco SPA 504G

1. Obtain the IP address from the phone using the phone’s on-screen menu.
2. Open the phone’s configuration by navigating to the phone’s IP address in your web browser.
3. Select the **Admin Login** link.  
   
4. Select the **Advanced** link.  
   
5. Select the **Provisioning** tab.  
   
6. Enter **https://kazoocommissionhost/cisco/spa504g/accountname/MAC.xml** under **Profile Rule**, substituting the hostname of the Kazoo Commission server for kazoocommissionhost, accountname for the account the phone is associated with and MAC with the phone’s lowercase MAC address.  
   
7. Enter **http://kazoocommissionhost/kazooCommissionCA.cer** under **Custom CA Rule**, substituting the hostname of the Kazoo Commission server for kazoocommissionhost.  
   
8. Click **Submit All Changes**.  
   
9. The phone will reboot and be provisioned with the configuration generated by Kazoo Commission.  
   

# Troubleshooting

## Phone Not Downloading Root CA Certificate

Verify that the phone is retrieving the Root CA Certificate from the server by examining the nginx log with **cat /var/log/nginx/access.log**.

* If there is no request from the phone, there is a connectivity problem that must be resolved.
* If there is a 404 Not Found error, ensure that the URL to the certificate is correct. The URL should be http://yourserverhostname/kazooCommissionCA.cer, where yourserverhostname is the actual hostname of the server nginx is running on.

## Phone Not Downloading Configuration

Verify that the phone is retrieving its configuration from the server by examining the nginx log with **cat /var/log/nginx/access.log**.

* If there is no request from the phone, there is a connectivity problem that must be resolved.
* If there is a 400 Bad Request error, ensure the relevant Root Certificate Authority is present in **/etc/nginx/kazoocommissionbundle.pem**.
* If there is a 404 Not Found error, ensure that the account name, MAC address, phone manufacturer and phone model are correct in the provisioning URL.
* If there is a 403 Forbidden error, ensure that the phone is requesting the configuration for its MAC address and not a different MAC address. Ensure that the phone includes the MAC address in the SSL Certificate Subject Field in the phone’s documentation.
* If there is a 500 Server error, that **KAZOO\_COUCH\_DB\_URL** represents a Kazoo Couch Database URL that can be accessed from the Kazoo Commission server.

## Phone Not Registering

* Ensure that **KAZOO\_COMMISSION\_SIP\_OUTBOUND\_PROXY is pointing to the correct Kazoo SIP server.**
* Ensure that **SIP\_DNS\_A\_RECORD** is set to use DNS A Records. Ensure that **SIP\_DNS\_A\_RECORD** is not set to use DNS SRV Records.