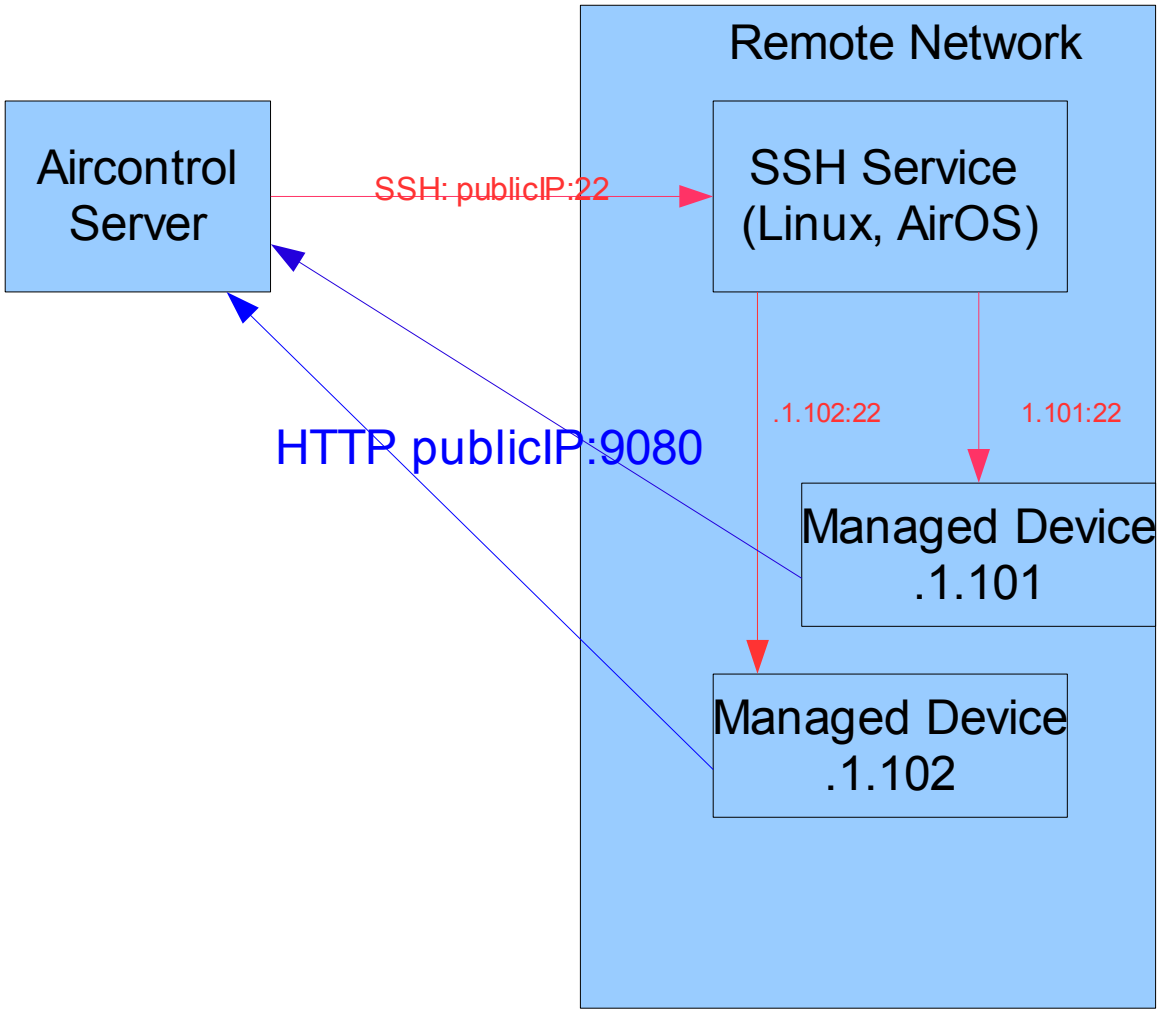
Management of Remote Networks through AirControl

This outlines the solution to manage devices in remote, non-routed networks from a central AirControl server. This is only needed when there is no route to the remote devices from the AirControl host.

How does it work?



AirControl uses SSH port forwarding ("tunneling") through the SSH host/gateway for all operation that require access via SSH to device (connect, firmware upgrade etc.). Note that access to the AirOS web interface is not supported through this mechanism, this is only for AirControl operation.

Managed device reports to AirControl server at public/routed address.

By using SSH port forwarding, the need for the AirControl user to manually forward each device SSH port though iptables or employ other mechanisms on the network adapter level, setup VPN etc. is avoided.

Requirements:

AirControl version 1.3.3 or later. See AirControl upgrade instructions below.

- Public (or routed private) address/port for AirControl server that can be accessed from the remote network. Managed devices send HTTP traffic to that port.
- Host with SSH service in the remote network: This can be any machine with SSH service that allows port forwarding. The machines SSH port needs to be accessible to AirControl (directly or through port forwarding on upstream router etc.) We have tested with Linux, pfsense and AirOS (AirOS version 3.6+ or 5.3+ required for discovery scan in remote network, 5.3-beta is available from forum).
- Devices you wish to manage through tunnel need to have SSH service enabled.

AirControl Upgrade

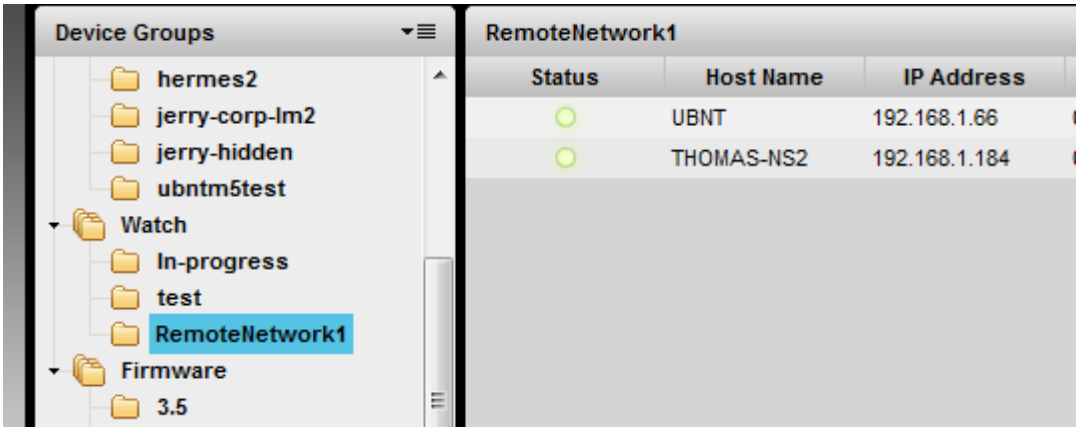
The tunneling feature is available with AirControl 1.3.3 or later. For more recent changes, you can also use the latest -dev build (upgrade only of the build number is higher than latest -beta release):

- Open the “About” dialog. Press CTRL-SHIFT-U
- Change update site URL to: http://www.ubnt.com/downloads/aircontrol/update-dev/
- Click “Check for Updates” and then update to the new -dev version shown

Setup/Configuration

Create device group for tunnel

Devices need to be grouped to associate a tunnel (it is recommended to use dynamic groups for this). First preference would be to group by private IP scheme (192.168.1.XXX -> Tunnel1, 192.168.2.XXX -> Tunnel2 etc.). If that is not possible because private sub net masks overlap between networks, use a device name prefix scheme or something similar. The point is to not have to manage groups manually once they are created when new devices are added in the remote network. Although for testing and smaller static setups it is also possible to use static groups.



Above shows group “RemoteNetwork” which will be associated to a tunnel.

Configure Tunneling Settings

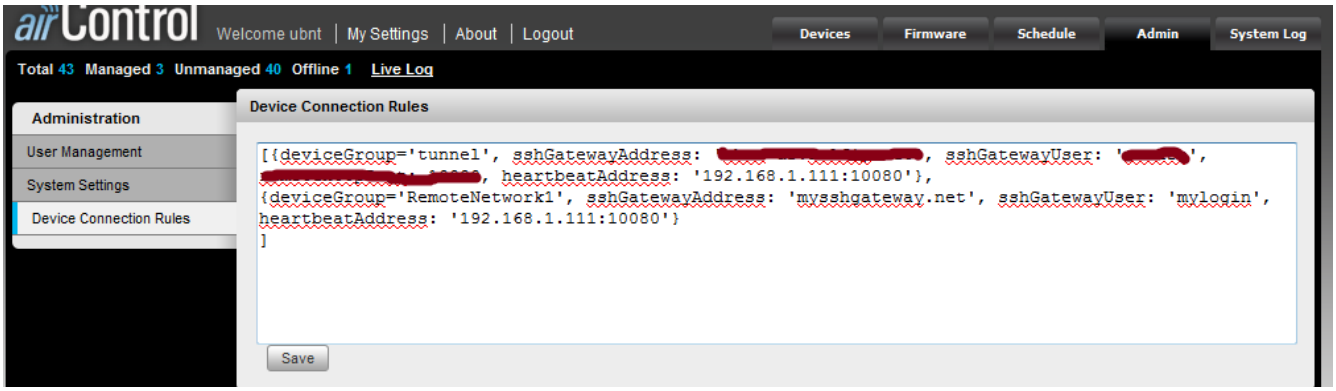
The tunnel settings will be entered under Admin->Device Connection Rules.

If this section is not enabled in the build you are using:

Open <AC_DIR>/webapps/ROOT/admin.xhtml
on line 39 enable menu link

```
<ui:fragment rendered="true">
    <li>
    <a:commandLink id="deviceGroupSettings"
```

There should be no need to restart the server, just refresh the page.



The rules are entered in JSON format (be careful when editing, there is very little validation, we will build a specific configuration UI for this in the future). All addresses and other values need to be in quotes. Example:

[{deviceGroup='RemoteNetwork0', sshGatewayAddress: 'mysshgateway1.net:222', sshGatewayUser: 'mylogin', heartbeatAddress: '192.168.1.111:9080'},
{deviceGroup='RemoteNetwork1', sshGatewayAddress: 'managed-airos-device:22', sshGatewayUser: 'mcuser', heartbeatAddress: '192.168.1.111:9080'}]

deviceGroup: The name of the previously created device group that contains the devices that you want

to manage through tunnel.

sshGatewayAddress: Address of the host in the remote network that provides SSH access. Any host that is in the remote private network and has a public address or address reachable from AirControl. It needs to have SSH service that supports key based authentication and SSH port forwarding. You should be able to SSH into that port from your AirControl machine.

heartbeatAddress: The HTTP port of your AirControl server that remote devices talk to. The AC server needs to have a public IP or external public IP forwarded to it. There is a possibility to also tunnel the reverse traffic but it is more robust to use a directly route able address for the server.

SSH authentication will require the AirControl public key on the gateway in authorized_keys. You can extract the public key from any of the already managed devices from ~mcuser/.ssh/authorized_keys

If you are using an AirOS device as SSH gateway, connect it first in AirControl through the public IP and then enter that public IP as “sshGatewayAddress“ and “mcuser” as “sshGatewayUser“ in the tunnel definition. You don't need to manually setup the SSH public key in this case.

Scan through Tunnel

Once tunneling is configured, you can scan for devices through that gateway. In the “Scan” dialog, you will find an additional drop-down to select the tunnel.