Readinet OS

The Readinet OS project encompasses the following projects:

- The Readinet Web Application (both local client and cloud server)
- The Operating System installed on the RockPi/Local Client
- Devops playbooks and scripts tasks related to the above items

The documentation inside this file is split by the user role (end users and developers)

End Users

Cloud Server

The readinet web application runs two separate instances of the cloud remote management server.

- Production - prod.readinet.org
- Staging - dev.readinet.org

There are three default accounts root, admin, and user. All of the users have email domain's of `mbari.net`. The default password is `R3AdiN#Tdone-litmus-maturing`

RockPi

Local Access

To use readinet, you will need to install a version of Readinet OS on your RockPi device.

1. Download the latest .img file from the `mbari-readinet-images` S3 bucket
2. Install the image to an SD Card
    - For a graphical experience, use Balena Etcher
    - For a text-based experience, use the `dd` command
3. Insert the SD Card into the RockPi and power the device on
    - The device can take up to 3 minutes to complete its initialization routine
4. After initialization completes, there will be a wifi network that appears starting with `readinet-mgmt`. (Each device will have a different suffix for this wifi address)
5. Connect to the `readinet-mgmt` network with a default password of `readinet`
6. You can access the web interface of the device at https://10.42.0.1

Connect to the cloud

In order to connect your device to the remote cloud management service, you will need to perform some additional steps.

1. Connect to the readinet-mgmt wifi address using the steps above
2. SSH into the device with `root@10.42.0.1`. The password is `readinet`
3. Initiate a vpn connection by issuing `tailscale up --login-server https://vpn.prod.readinet.org` (or `https://vpn.dev.readinet.org` for the development server)
4. The node key must then be accepted on the server. The command must be issues by a sysadmin on the remote server is `headscale nodes register --user readinet --`

`key nodekey:THE_NODEKEY_PRESENTED_BY_THE_CLIENT`

5. VPN connectivity can be verified by issuing `tailscale status` or by pinging the server through the VPN with `ping server.readinet.local`

There are two separate VPNs: Dev (vpn.dev.readinet.org) and Prod (vpn.prod.readinet.org). You can only be connected to one at a time.

Configure the device

1. SSH into the device using `ssh root@10.42.0.1`. The default password is `readinet`
2. The main configuration for the web interface is found in `/opt/web/.env`. You can edit the file by issuing `vim /opt/web/.env` or `nano /opt/web/.env`
3. After making changes, you must reboot the interface by issuing `systemctl restart readinet-web`

The configuration options are as follows:

- `SAMPLER_TTY`: The device location of the sampler on the linux box. Should start with `/dev`
- `SAMPLER_TIMEOUT_SECONDS`: The number of seconds before a sampler operation would be cancelled. Note: this value can be overridden in code for some sampler operations
- `THIS_DEVICE_ID`: The unique identifier for the device. This should match `/etc/machine-id`. Changing this value is not recommended.
- `APP_VERSION`: The version of the application displayed in the lower left hand corner of the interface. Generated through the OS build process.
- `MQTT_BROKER_URL`: The location of the MQTT Broker. In normal scenarios, it should be `server.readinet.local`.
- `ROLE`: The role of the system. Must be either `device` or `server`. There should only be one `server` in the deployment.
- `SECRET_KEY`: This is a web-server specific item to deal with cross site forgery. Do not change this value
- `FLASK_ENV`: This is signifier for what environment the application is running in. It should be `prod` unless otherwise specified.
- `SQLALCHEMY_DATABASE_URI`: A sqlite URI that points to the database in use
- `REMEMBER_COOKIE_HTTPONLY`: A web-server specific item
- `SESSION_COOKIE_SAMESITE`: A web-server specific item
- `RUN_BACKGROUND_THREADS`: Background threads provide MQTT-syncing and device status reporting. This should be `true` unless debugging.
- `APP_VERSION`: A text string that will get displayed as the current version. This should be automatically templated from the OS build.

Hardware

Core Package (Need one for every system)

- https://shop.allnetchina.cn/collections/frontpage/products/rock-pi-s?variant=29158785974374
- https://shop.allnetchina.cn/collections/frontpage/products/rock-pi-s-case
- https://shop.allnetchina.cn/collections/rock-pi-4-cases/products/dual-band-antenna-kit-for-rock-pi-s-case
- https://www.amazon.com/dp/B06XWMQ81P
- https://www.amazon.com/dp/B07TYQRXTK

Additional Debug Tools (Not needed, but helpful)

- https://shop.allnetchina.cn/products/usb-ttl-debug-cable-for-rock-pi-4

- https://shop.allnetchina.cn/collections/measurement-and-testing/products/usb-pd-power-meter-with-usb-type-c-type-b-and-type-a-ports-rgb-and-bluetooth

Developers

Local Development

1. Install `podman` and `podman-compose`
2. Run `podman-compose up`
3. Run `podman exec -it readinet_server ./bootstrap.sh`
4. Run `podman exec -it readinet_device1 ./bootstrap.sh`
5. Run `podman exec -it readinet_device2 ./bootstrap.sh`
6. If you want to start a "simulated sampler" on a device, you must run `podman exec -it readinet_deviceNNN /opt/web/device_simulator/simulate.sh`
7. Visit one of the following web addresses:
    - server: http://localhost:5000
    - device1: http://localhost:5001
    - device2: http://localhost:5002
8. If you want to root around the containers, `podman exec -it -e RUN_BACKGROUND_THREADS='' readinet_deviceNNN bash`

Devops/Sysadmin Info

Terraform controls the cloud resources, while ansible controls the software and configuration on the operating system.

To connect to a server, SSH using an SSH key must be used (passwords aren't accepted). The `admin` user should be used for ssh access, and the authorized keys can be found in `/home/admin/.ssh/authorized_keys`. A bootstrap of the first authorized users are made in the user_data declaration of the terraform.

To run ansible, the following commands are useful:

```
ansible-galaxy install -r requirements.yml
ansible-playbook -i hosts server.yml -l prod
ansible-playbook -i hosts server.yml -l dev
```

On the VPN/Headscale side, a single generic `readinet` user was created. However, this single user setup isn't required and any VPN user and identity management can be instituted.