

# IBSNOW: AWS AMI Access Instructions

## Prerequisites

Before being able to access the AMI you must have completed the installation process [here](#). Once the AMI in AWS Marketplace has been used to create an EC2 instance in AWS it is possible to access the filesystem of the EC2 instance via SSH.

## AMI Access Instructions

The only access to the IoT Bridge system is via SSH. As part of the EC2 instance provisioning process a 'keypair' is associated with the instance. This is something that is available during the keypair creation process and can not be retrieved after the initial creation so make sure to save it. To access the instance, you will need the following pieces of information.

**Username: ubuntu**

**Private key: (as created during the EC2 instance provisioning process)**

**SSH port: 22**

In addition to the above information, you must make sure the SSH port (22) is open for inbound connections in the security group you have associated with the IoT Bridge EC2 instance you have created.

There is nothing unique about connecting to a IoT Bridge instance over any other other AWS Ubuntu based instance. AWS provides detailed instructions on this process of connecting via SSH and SCP [here](#) if you are unfamiliar with it.

## Files of Interest

The following shows the files that are likely of most interest with regard to IoT Bridge functionality.

```
@ This is the core configuration file where Snowflake and MQTT Server configuration is performed - details can
be found here:
# https://docs.chariot.io/display/CLD80/IBSNOW%3A+Configuration
/opt/ibsnow/conf/ibsnow.properties

# This is the log file directory. Cirrus Link support may ask for these files to resolve configuration,
connectivity, or other
# issues.
/opt/ibsnow/log

# For support tickets and debugging purposes Cirrus Link support may ask you to make modifications to this file
on occasion
/opt/ibsnow/conf/logback.xml

# This starts the IoT Bridge service - note the service starts automatically on boot so using this script
directly
# is only generally needed for debugging
sudo systemctl start ibsnow

# This stops the service
sudo systemctl stop ibsnow
```