# Advanced: MQTT Modules in Redundant Ignition Environment

## Prerequisites

- Knowledge of Ignition and Module installation process: Cirrus Link Module Installation

## Summary

The Inductive Automation platform and MQTT modules can be resilient to failures when configured to use redundancy. Redundant Ignition systems can be set up and configured to act as failover backups for primary/master Ignition instances. This tutorial will provide step by step instructions for installing a set of Ignition systems with redundancy on the host/primary Ignition instance as well as redundancy on the MQTT enabled edge nodes. For this tutorial we will show how to set up a total of six Ignition systems. These will be:

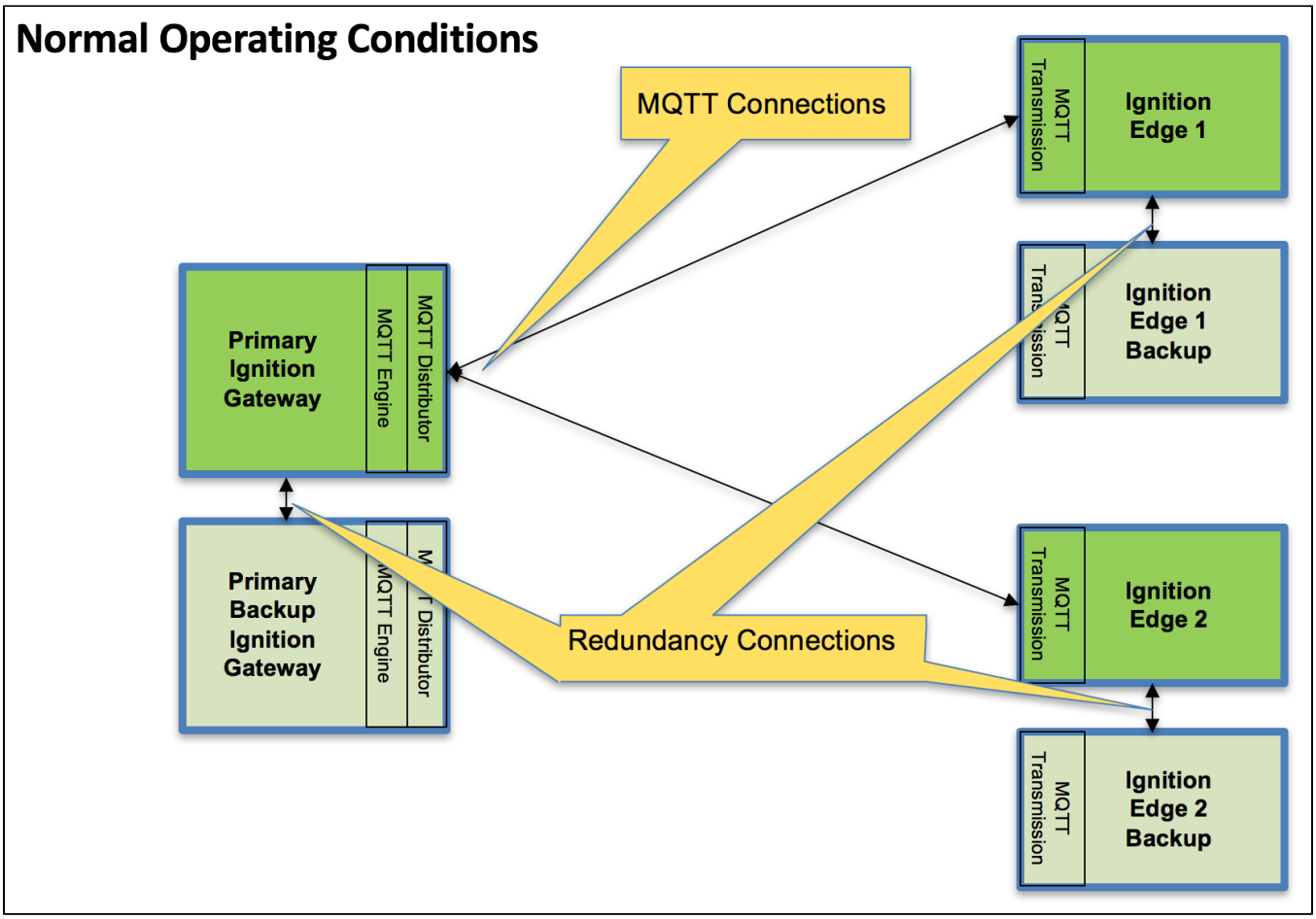
- **Ignition Primary**
  - An Ignition system running as master with MQTT Distributor and MQTT Engine installed. This is what remote Ignition systems will send data to in normal operation.
- **Ignition Primary Backup**
  - An Ignition system running as backup with MQTT Distributor and MQTT Engine installed. This is what remote Ignition systems will send data to when Ignition Primary fails.
- **Ignition Edge 1**
  - An Ignition system running as master with MQTT Transmission installed. This will send data to Ignition Primary in normal operation. If Ignition Primary is in a failed state, this will send data to Ignition Primary Backup.
- **Ignition Edge 1 Backup**
  - An Ignition system running as backup with MQTT Transmission installed. This will send data to Ignition Primary in the event that Ignition Edge 1 fails. If Ignition Primary is in a failed state and Ignition Edge 1 is in a failed state, this will send data to Ignition Primary Backup.
- **Ignition Edge 2**
  - An Ignition system running as master with MQTT Transmission installed. This will send data to Ignition Primary in normal operation. If Ignition Primary is in a failed state, this will send data to Ignition Primary Backup.
- **Ignition Edge 2 Backup**
  - An Ignition system running as backup with MQTT Transmission installed. This will send data to Ignition Primary in the event that Ignition Edge 2 fails. If Ignition Primary is in a failed state and Ignition Edge 2 is in a failed state, this will send data to Ignition Primary Backup.

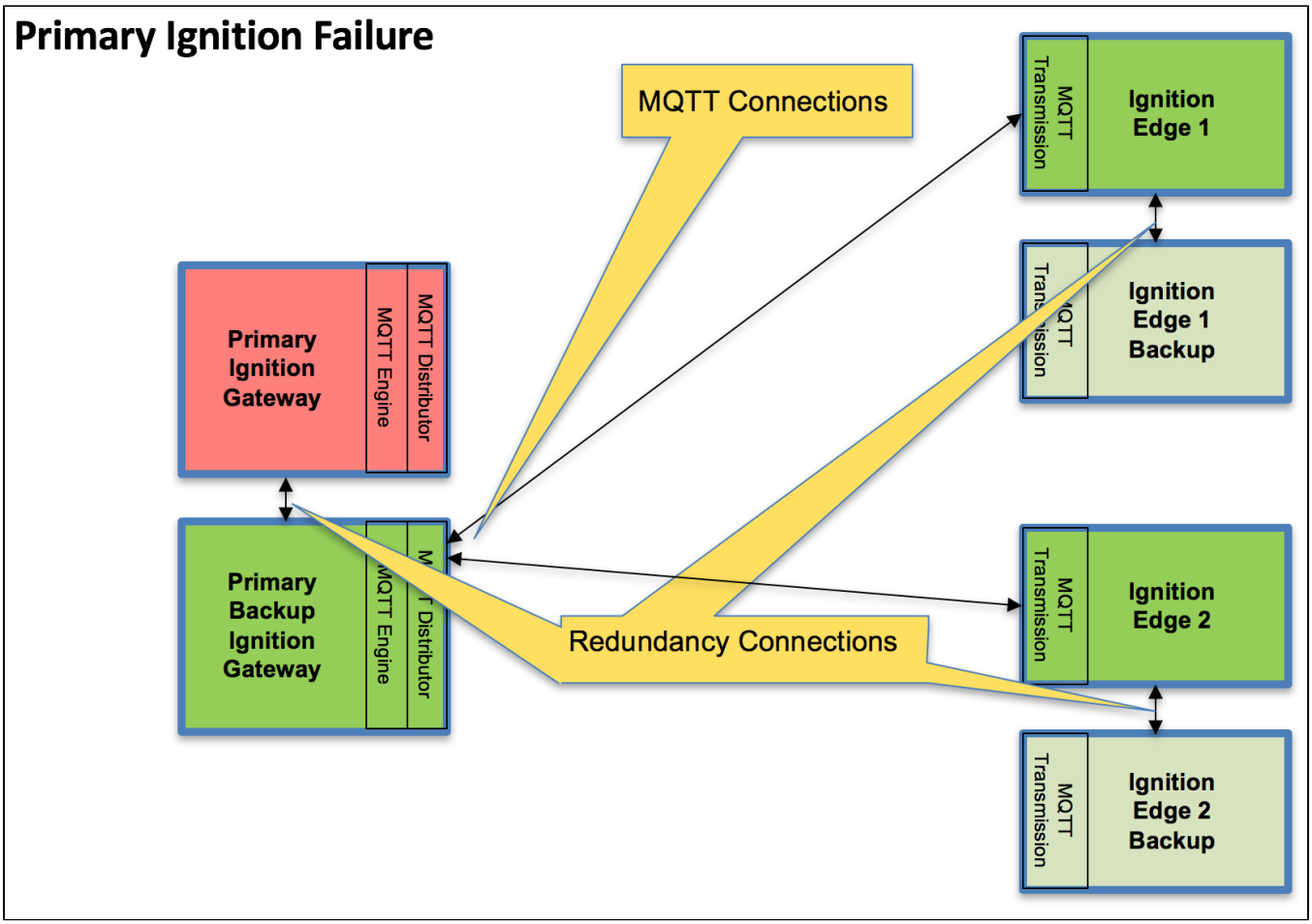
Additional Edge Nodes could be added to this infrastructure. It is also important to note that the Ignition Edge Nodes with MQTT Transmission could also be instances of Ignition Edge MQTT depending on your requirements (https://inductiveautomation.com/whats-new-ignition-edge). There are additional considerations when setting up a real world system using redundancy. These topics are not covered in this tutorial but should be taken into consideration.

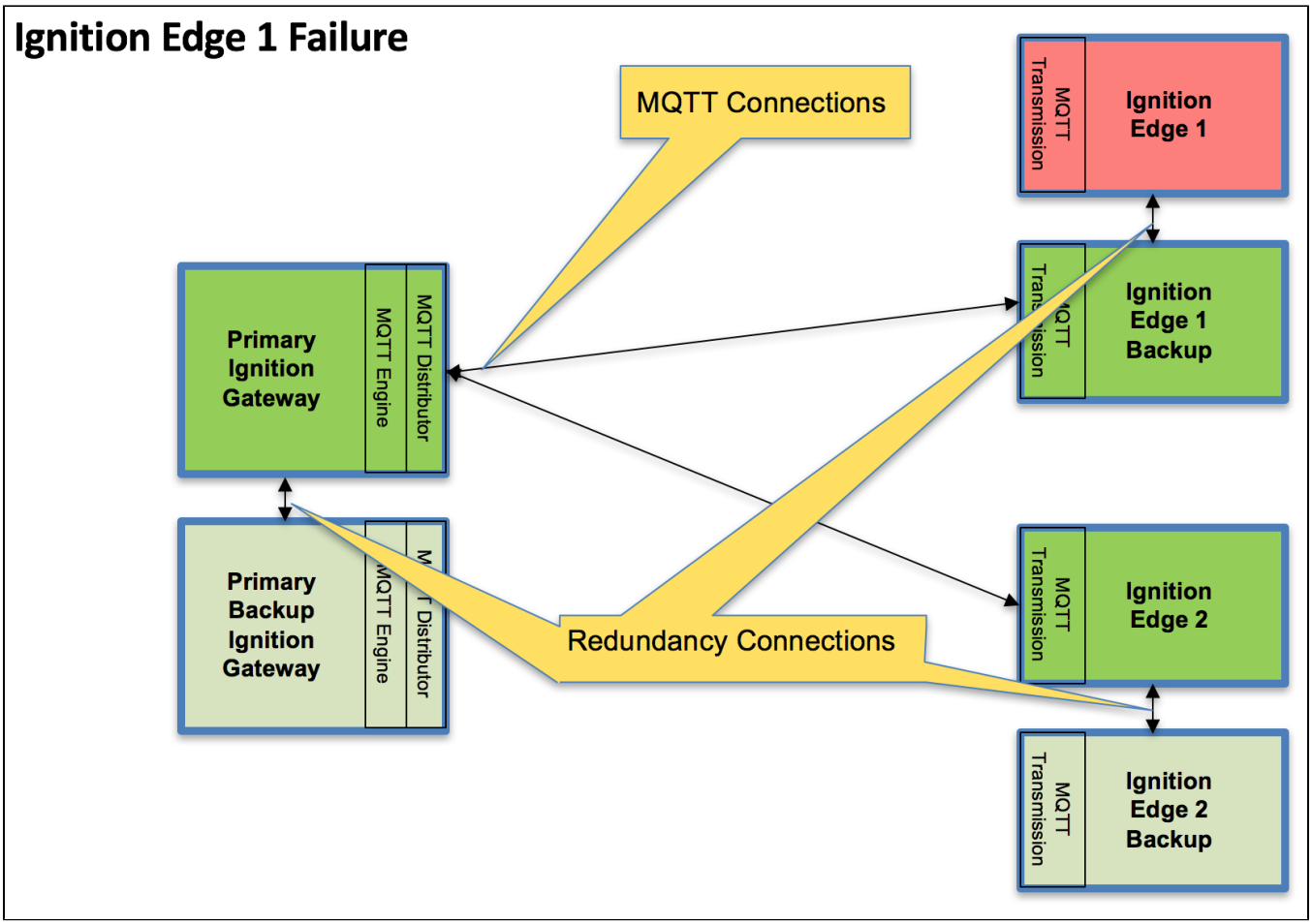
- Network paths
  - It may make sense to have edge nodes support multiple network paths to the MQTT servers. For example, Ethernet, cellular, and satellite could all exist as supported network paths on a single Edge gateway. This will help ensure additional reliability by supporting failover of networks.
- Primary Ignition and Primary Ignition backup placement
  - This tutorial was created by modeling this exact environment using Amazon AWS EC2 instances in the cloud. Reliability could be improved by putting Ignition Primary and Ignition Primary Backup in different AWS availability zones or even different AWS regions. This would allow the primary Ignition with MQTT Distributor and MQTT Engine to continue to operate even in the case of AWS failures. In the case of on premises installations of Ignition these could be placed in different physical locations and/or on secondary networks.
- MQTT Servers
  - Additional MQTT Servers can be added so MQTT connections from remote edge nodes remain established. Additional Chariot MQTT Servers can be used to make the system more robust (https://cirrus-link.com/mqtt-broker-iiot-mqtt-servers/).
- History enablement in MQTT Transmission
  - MQTT Transmission supports caching of data in the case that it can not establish a connection to any of the configured MQTT Servers. Once a connection is reestablished, it will begin reporting and flush the stored historical values to prevent data loss in catastrophic failures.

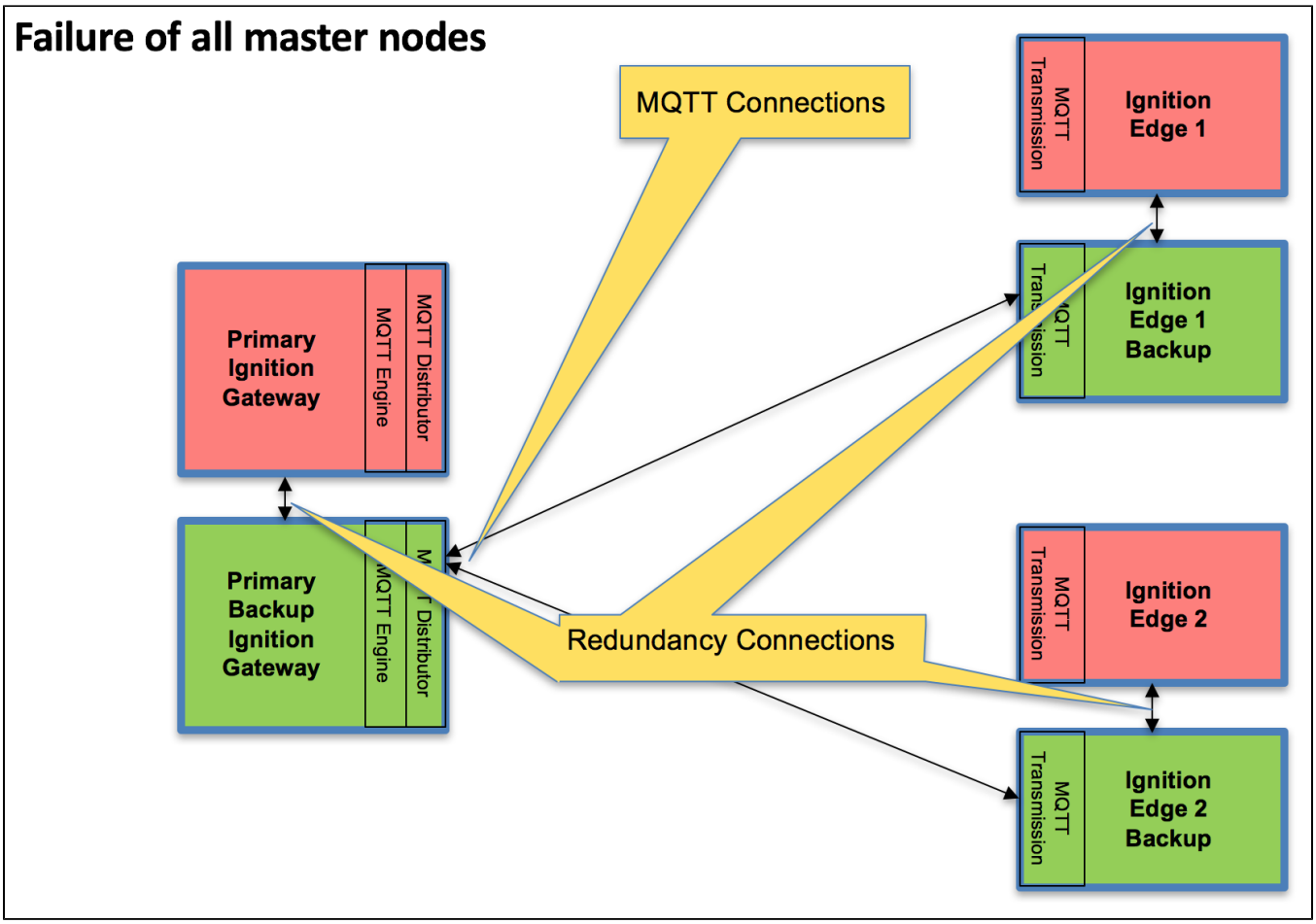
Upon completion of this tutorial you will have a functional system with redundancy/failover support for both remote edge nodes as well as the primary Ignition system that the remote edge nodes are reporting to.

## Architecture


Normal Operating Conditions
MQTT Connections
Primary Ignition Gateway
MQTT Distributor
MQTT Engine
Primary Backup Ignition Gateway
MQTT Distributor
MQTT Engine
Redundancy Connections
Ignition Edge 1
MQTT Transmission
Ignition Edge 1 Backup
MQTT Transmission
Ignition Edge 2
MQTT Transmission
Ignition Edge 2 Backup
MQTT Transmission


Primary Ignition Failure
MQTT Connections
Primary Ignition Gateway
MQTT Engine
MQTT Distributor
Primary Backup Ignition Gateway
MQTT Engine
Distributor
Redundancy Connections
MQTT Transmission
Ignition Edge 1
MQTT Transmission
Ignition Edge 1 Backup
MQTT Transmission
Ignition Edge 2
MQTT Transmission
Ignition Edge 2 Backup


Ignition Edge 1 Failure
MQTT Connections
Primary Ignition Gateway
MQTT Distributor
MQTT Engine
Primary Backup Ignition Gateway
MQTT Distributor
MQTT Engine
Redundancy Connections
Ignition Edge 1
MQTT Transmission
Ignition Edge 1 Backup
MQTT Transmission
Ignition Edge 2
MQTT Transmission
Ignition Edge 2 Backup
MQTT Transmission




## Tutorial

### Step 1: Download and Install Ignition on Six Systems

Ignition is an Industrial Application Platform that can be used to create SCADA and HMI solutions. A fully functional Ignition system can be downloaded and run in trial mode.

Go to the Inductive Automation download page and download the desired Ignition installer for Windows, Linux or MacOS;
https://inductiveautomation.com/downloads/ignition

Once the Ignition installer has been downloaded, follow the instructions provided by Inductive Automation to install and startup Ignition.

(Note: For this test infrastructure, MQTT Distributor will be installed as an Ignition module on both the Primary Ignition Gateway as well as the Primary Ignition Gateway Backup. Make sure to either turn off firewalls or at a minimum allow inbound connections to TCP/IP port 1883, as remote MQTT Clients will need to be able to establish a TCP/IP socket connection to these ports).

### Step 2: Download and Install the Cirrus Link MQTT Modules

Go to the Inductive Automation download page again and scroll down to the Third Party modules section. Find the Cirrus Link modules section and download the MQTT Distributor, MQTT Engine, MQTT Transmission modules.
https://inductiveautomation.com/downloads/ignition. For each of the Ignition instance, install the following MQTT Modules.

- **Ignition Primary**
  - MQTT Distributor and MQTT Engine
- **Ignition Primary Backup**
  - MQTT Distributor and MQTT Engine
- **Ignition Edge 1**
  - MQTT Transmission
- **Ignition Edge 1 Backup**
  - MQTT Transmission
- **Ignition Edge 2**
  - MQTT Transmission
- **Ignition Edge 2 Backup**
  - MQTT Transmission

## Step 3: Configure the MQTT Modules

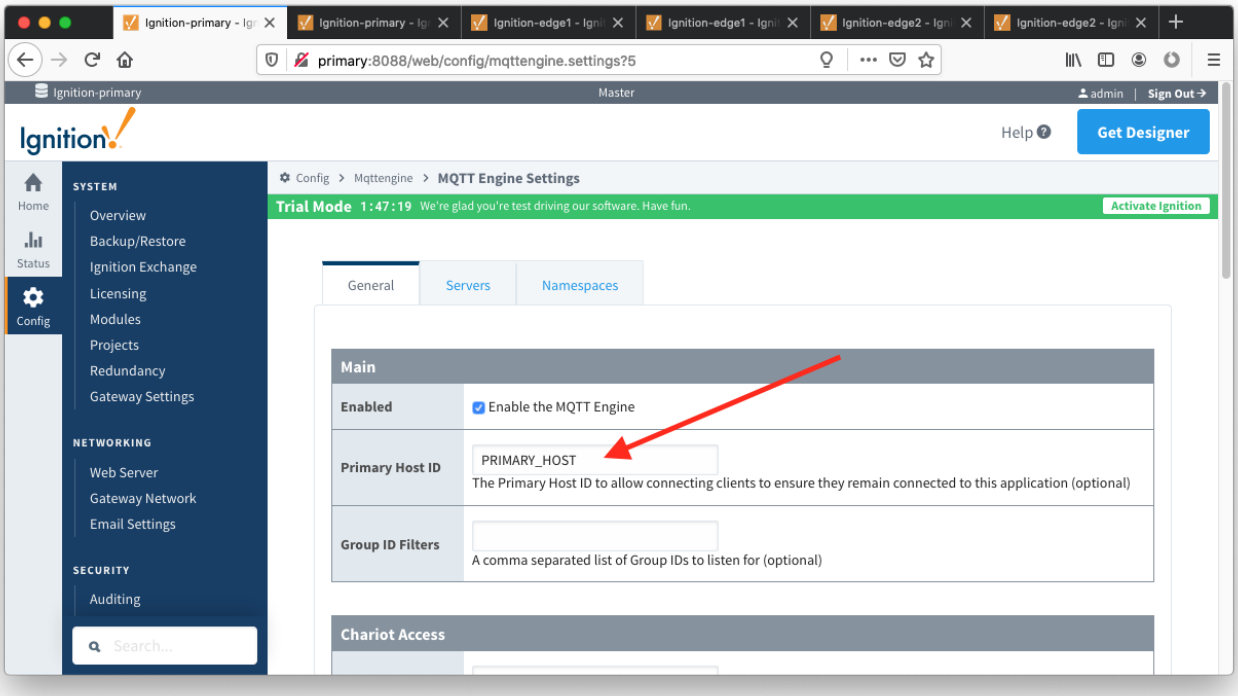
Once Ignition is installed, the MQTT Modules are installed, and everything is running we can configure the systems. Since we are going to have a backup for each master system, we only need to do most of the configuration for the master systems. Later, we can sync the configurations to the backups automatically. We'll start by configuring the modules and configure the redundancy settings in the next step.

### Ignition Primary - MQTT Distributor

- No modifications to the default parameters are required. However, it is important to make sure the Operation System allows inbound connections on port 1883 and there are no firewalls blocking inbound connections on this port from the remote edge nodes.

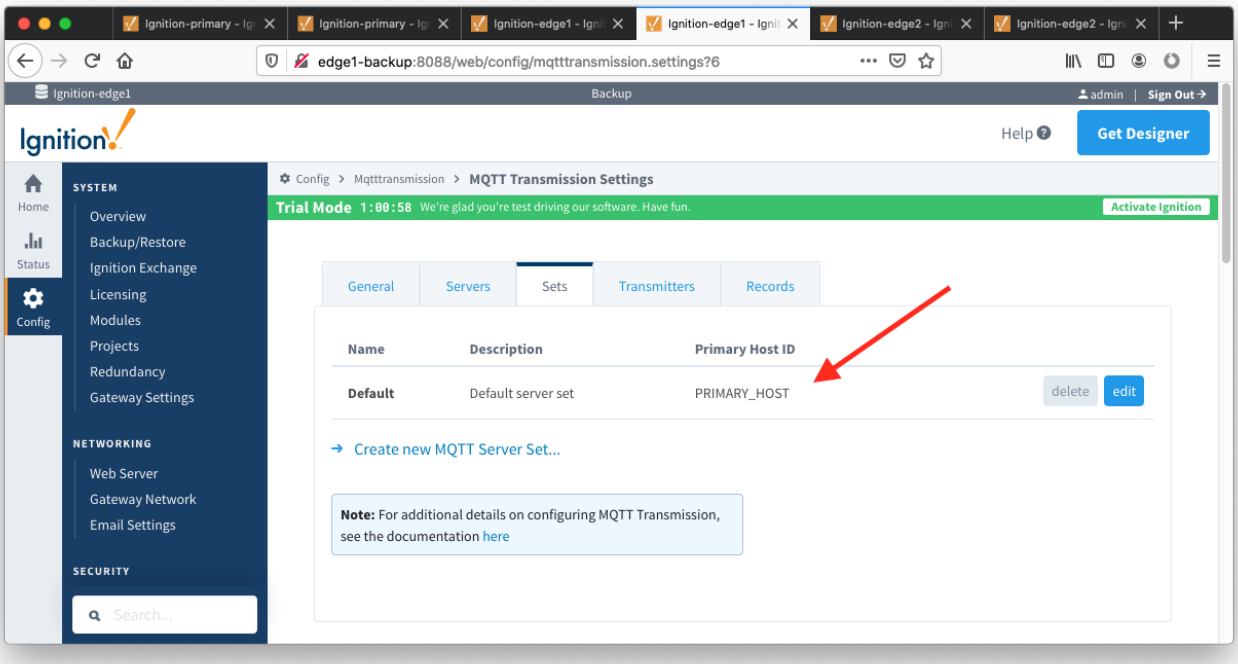
### Ignition Primary - MQTT Engine

- The only change from defaults is to set a Primary Host ID. MQTT uses Quality of Service (QOS) levels to ensure messages get delivered. However, this only ensures delivery between a single MQTT client and the MQTT server. In other words, it doesn't ensure delivery from one MQTT client to another MQTT client. Sparkplug introduces the notion of a Primary Host ID which is used to ensure client to client communications. The only requirement is that it match exactly on both the MQTT Engine and MQTT Transmission configurations. For this tutorial the Primary Host ID will be "PRIMARY_HOST"

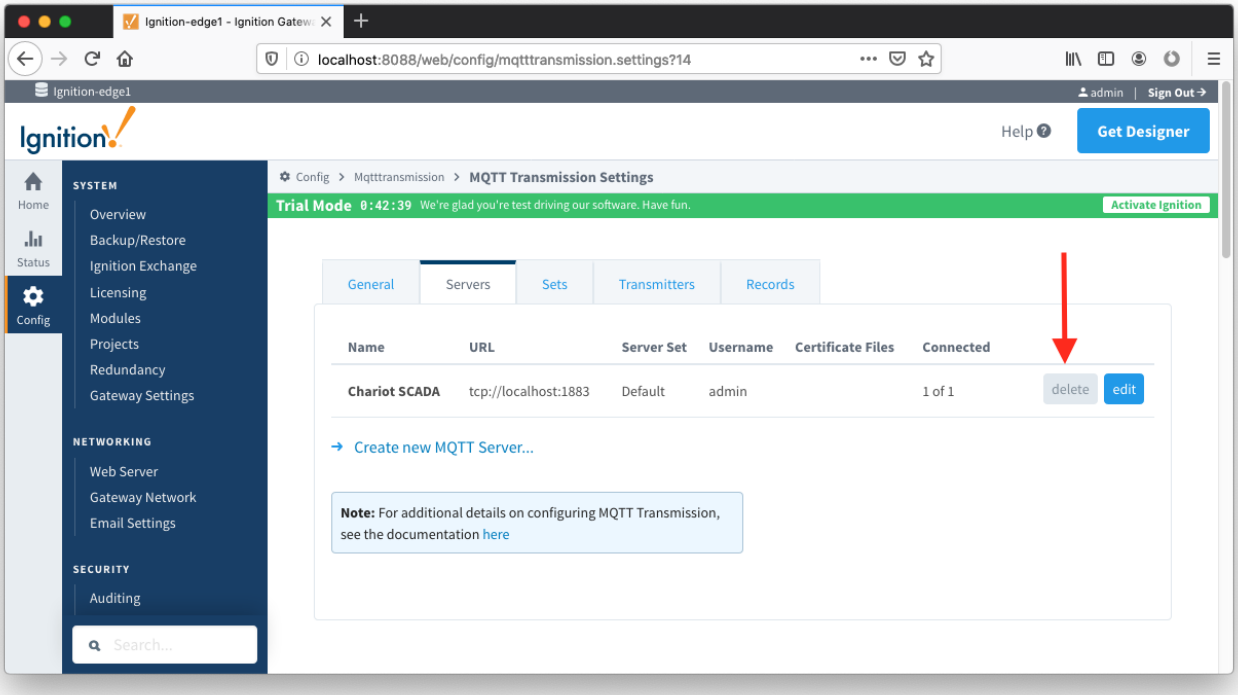



### Ignition Edge 1 and Ignition Edge 2 - MQTT Transmission (Configure the same on both Ignition instances)

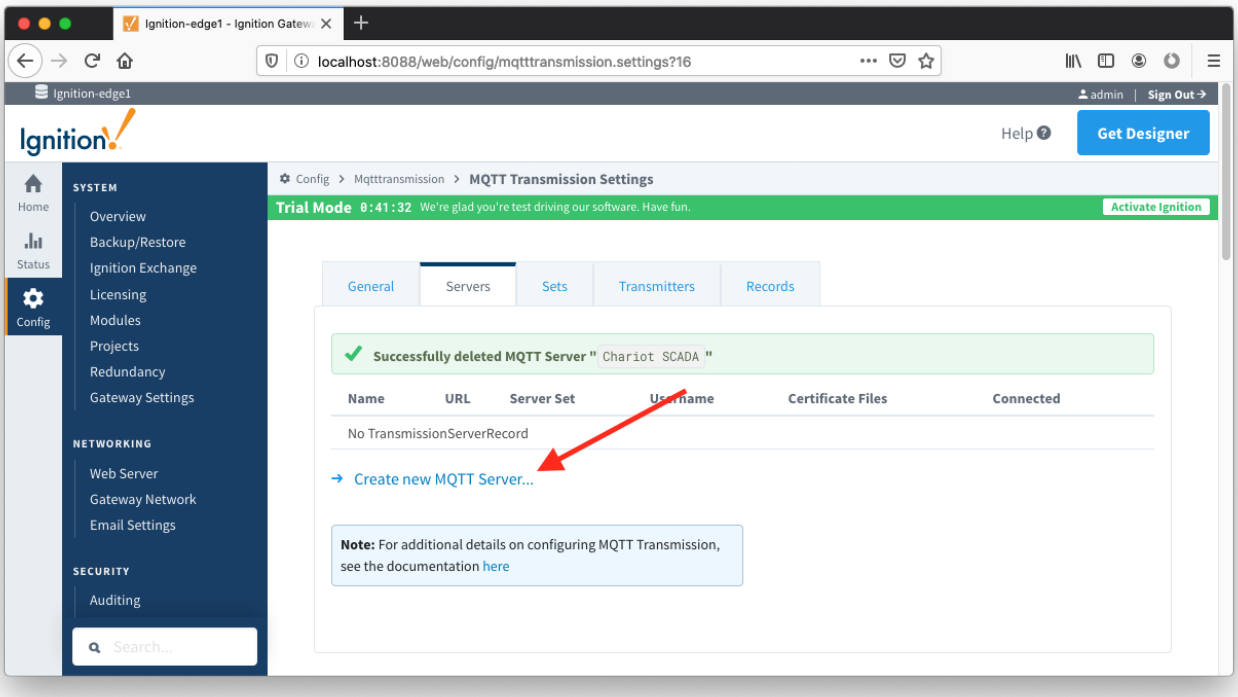
- As with the MQTT Engine configuration, the Primary Host ID must be configured on the "Default" Set.



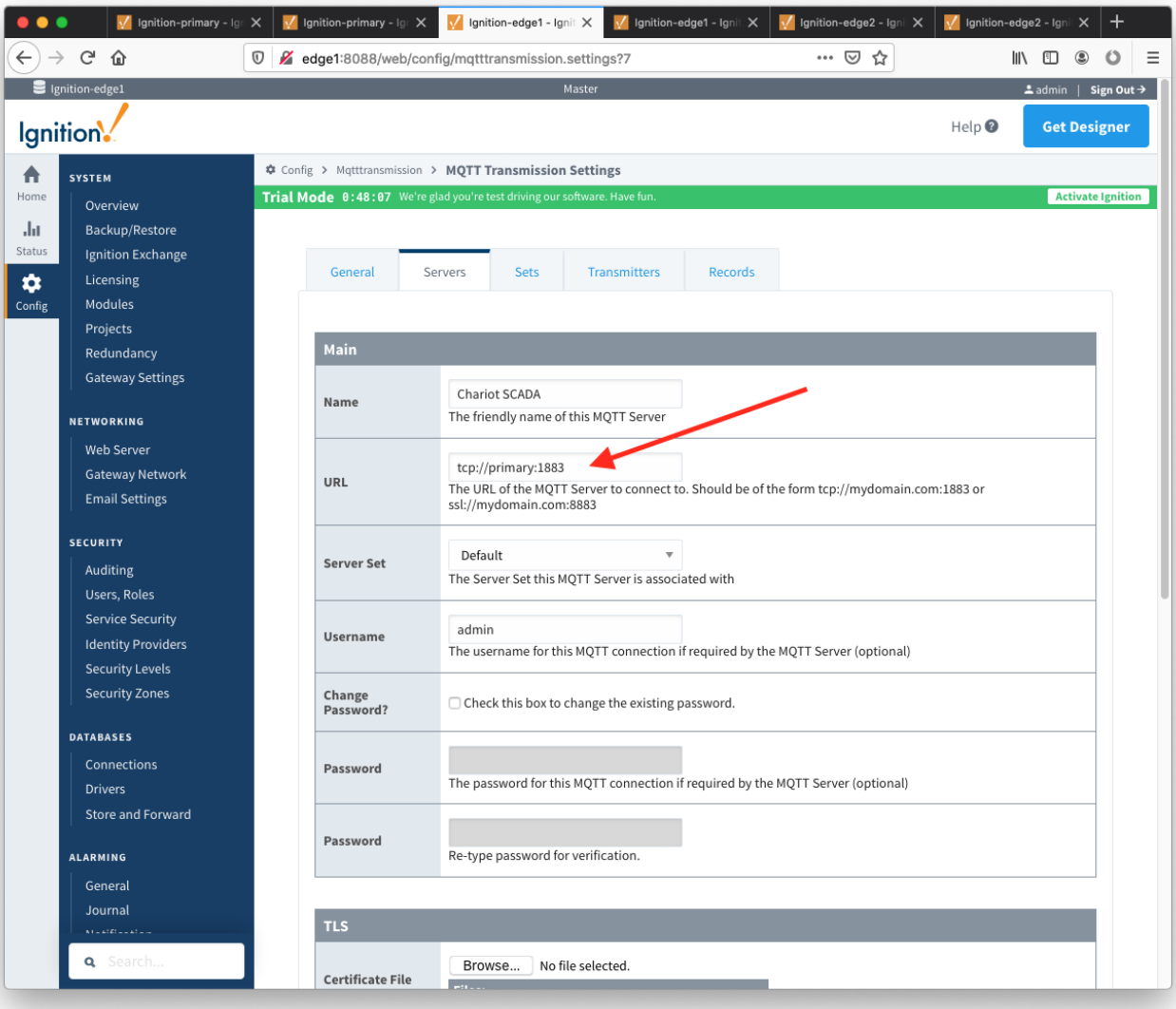

- Delete the existing "Chariot SCADA" Servers setting.



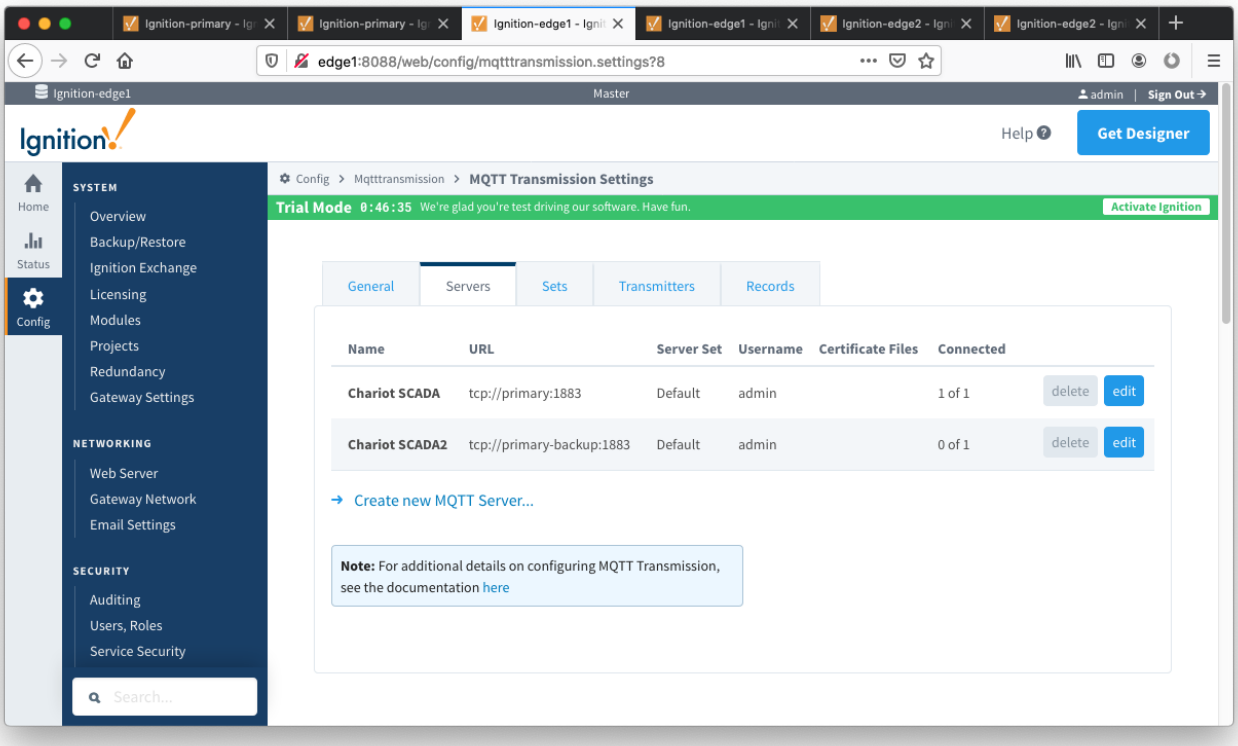

- Create a new MQTT Server configuration by clicking the link below.




- Configure MQTT Transmission to point to the Primary Ignition. Configure as shown below making sure to change the URL to reflect your network settings. For example, if you Ignition Primary is at 192.168.1.100 the MQTT Server URL would be: tcp://192.168.1.100:1883. After setting the parameters as shown below. Click the 'Save Changes' button at the bottom.




- Repeat the process of creating a MQTT Server but instead point it to the Ignition Primary Backup MQTT Server. These are the parameters to use:
  - Name: primary-backup
  - URL: tcp://primary-backup:1883
    - Change 'primary-backup' in the URL to reflect the network address of the Ignition Primary Backup server.
  - Server Type: MQTT Distributor
  - Server Set: Default
  - Username: admin
  - Password: changeme
- When complete, verify both MQTT Servers appear in the list as shown below.




- Finally, make sure to set up the same MQTT Transmission configuration in the Ignition Edge 2 instance.

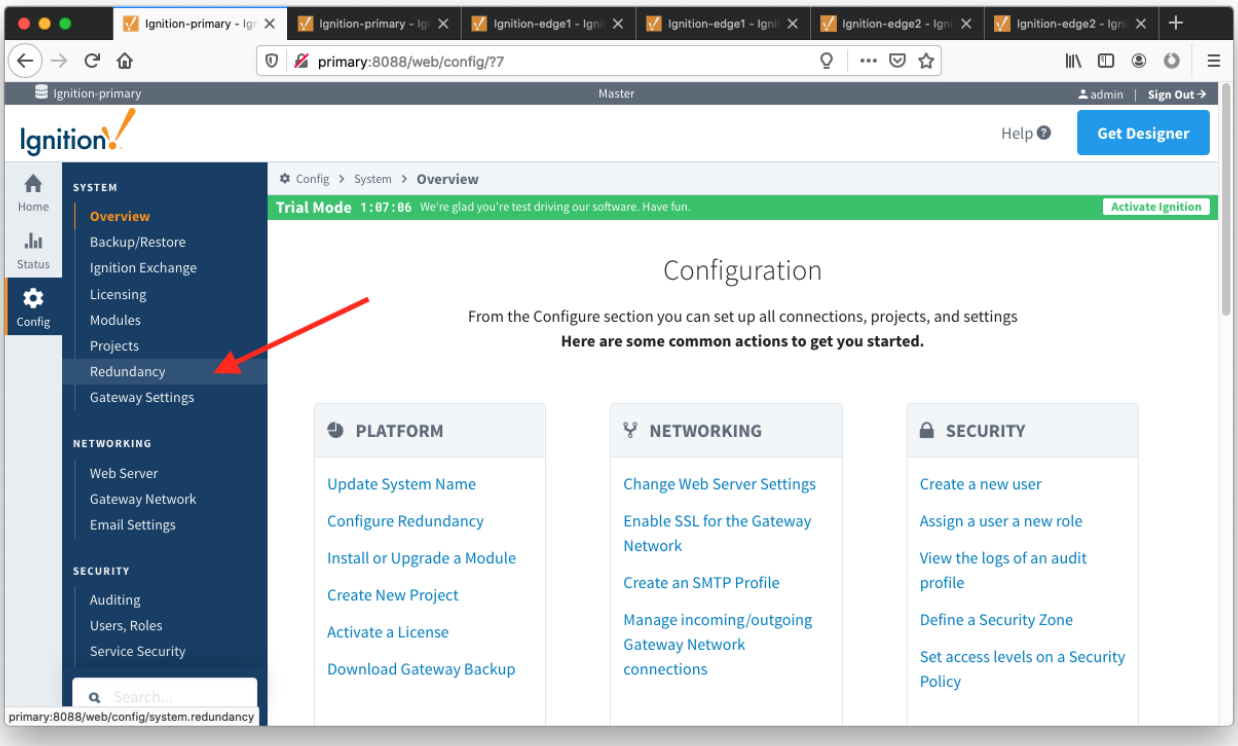
## Step 4: Configure Redundancy

The following configuration shows all of the redundancy settings that were used in setting this environment up using Amazon's AWS EC2 instances (virtual machines). The configuration will vary based on your network configuration. Additional Ignition redundancy resources can be found here Ignition Setting Up Redundancy

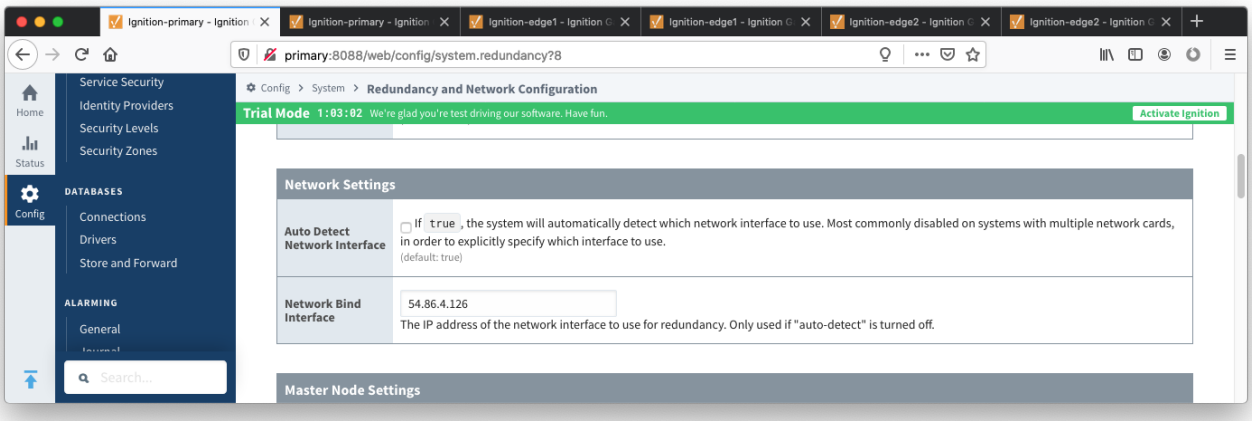
> ⚠ The Ignition Standby Activity Level, which is how the node operates when it is not the active node, has no impact on the MQTT modules. For the purposes of the tutorial we have set the Standby Activity to 'Warm'

### Ignition Primary

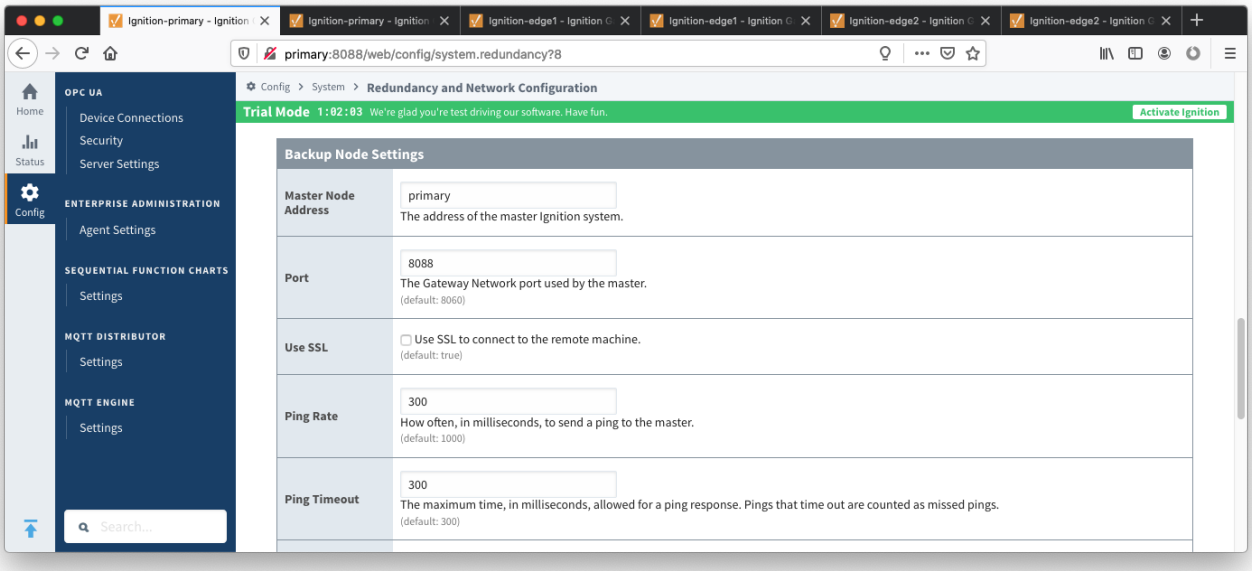
- Select Redundancy on the left navigation bar. Then set the Mode to 'Master' and set the Standby Activity left to 'Warm' as shown below.




- Set up the Redundancy Network Settings. The settings here are specific to your network setup. **On many LAN configurations none of these changes are required**. What is shown below was the configuration for setting up all of these components in Amazon's AWS EC2 instances. The changes were:
    - Uncheck 'Auto-detect network interface'
    - Set the 'Network Bind Interface' to the public IP address of the Ignition Primary EC2 instance. On a LAN this would be the primary network interface address of the Ignition Primary machine.



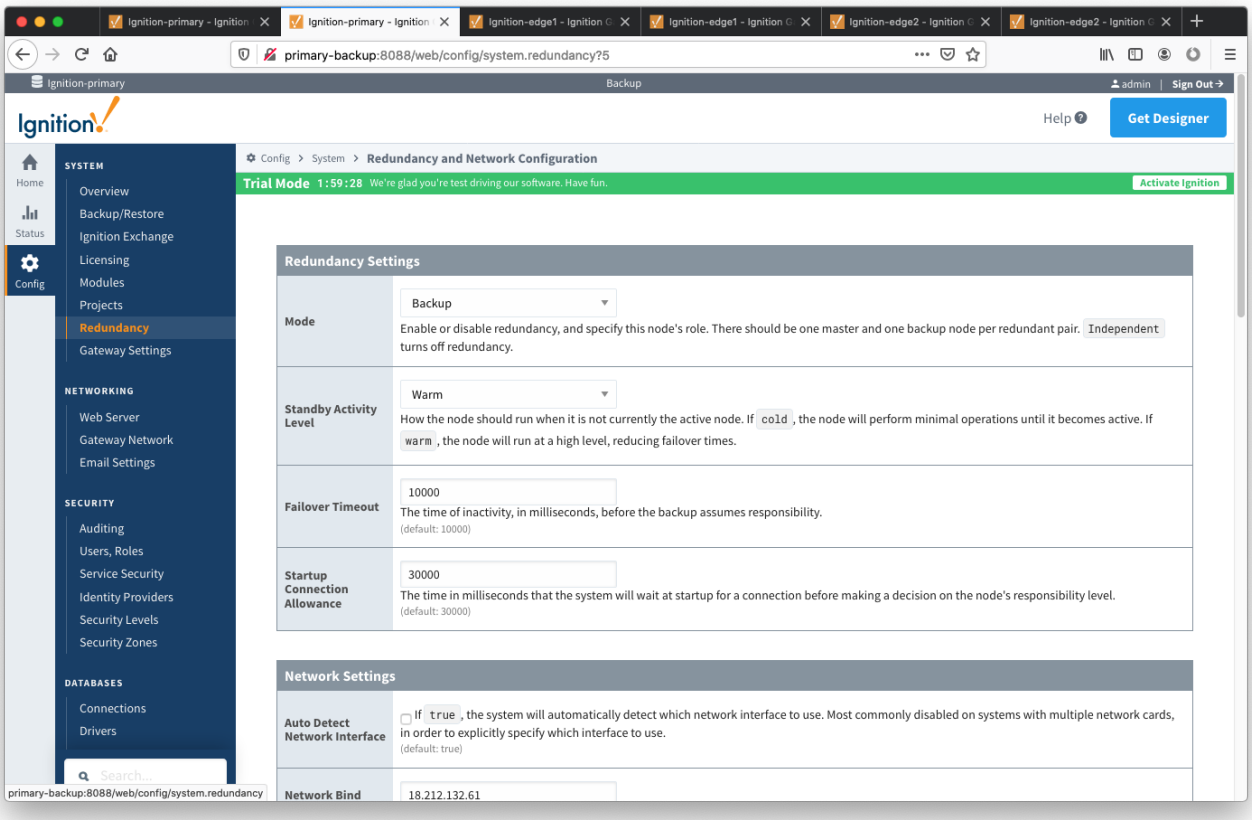

- Set the Master Node Address. Note in the configuration below a hostname is being used. This should be the primary network interface address of the Ignition Primary Gateway.




- Finally, click the 'Save Changes' button.

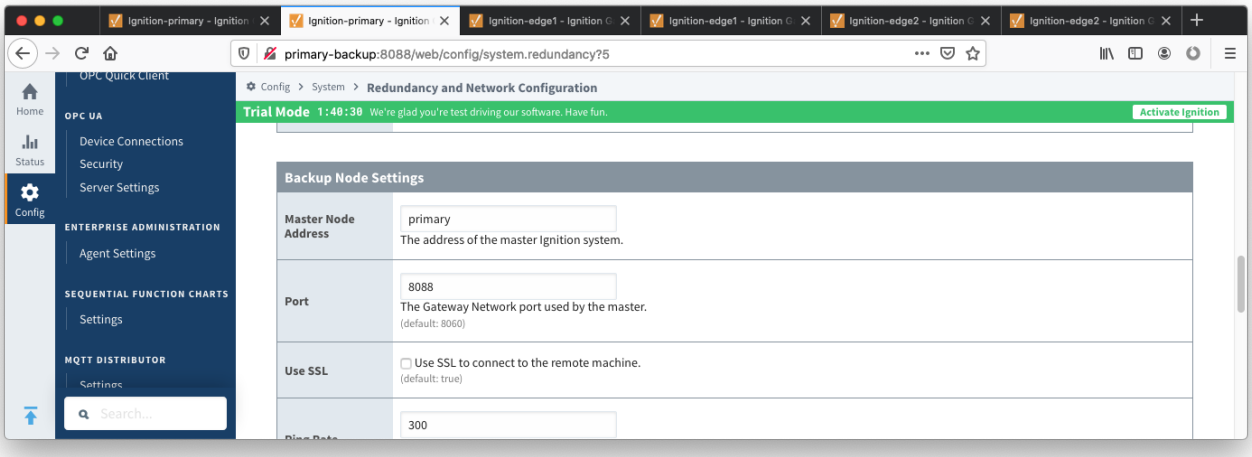
## Ignition Primary Backup

- Select Redundancy on the left navigation bar. Then set the Mode to 'Backup' and set the Standby Activity left to 'Warm' as shown below.




- Set up the Redundancy Network Settings. The settings here are specific to your network setup. **On many LAN configurations none of these changes are required**. What is shown below was the configuration for setting up all of these components in Amazon's AWS EC2 instances. The changes were:
    - Uncheck 'Auto-detect network interface'

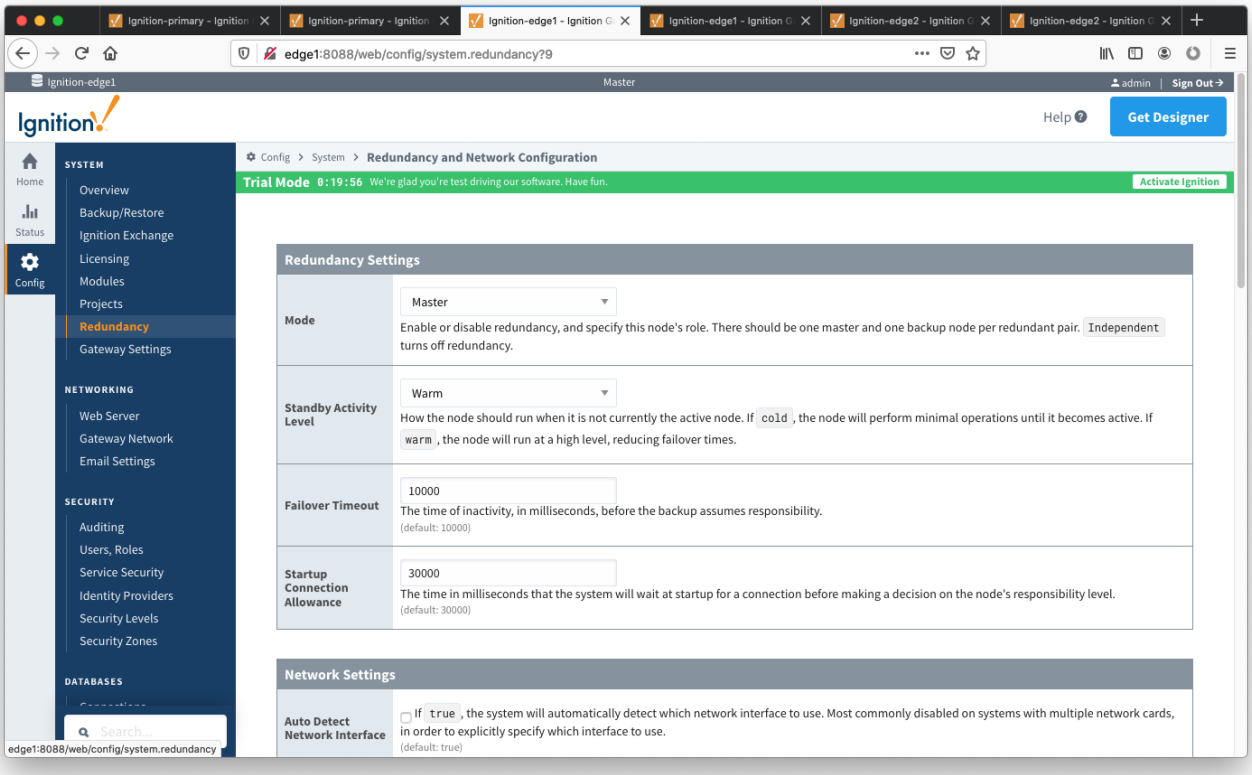
    - Set the 'Network Bind Interface' to the public IP address of the Ignition Primary Backup EC2 instance.  On a LAN this would be the primary network interface address of the Ignition Primary Backup machine.
- Set the Master Node Address.  Note in the configuration below a hostname is being used.  This should be the primary network interface address of the Ignition Primary Gateway.



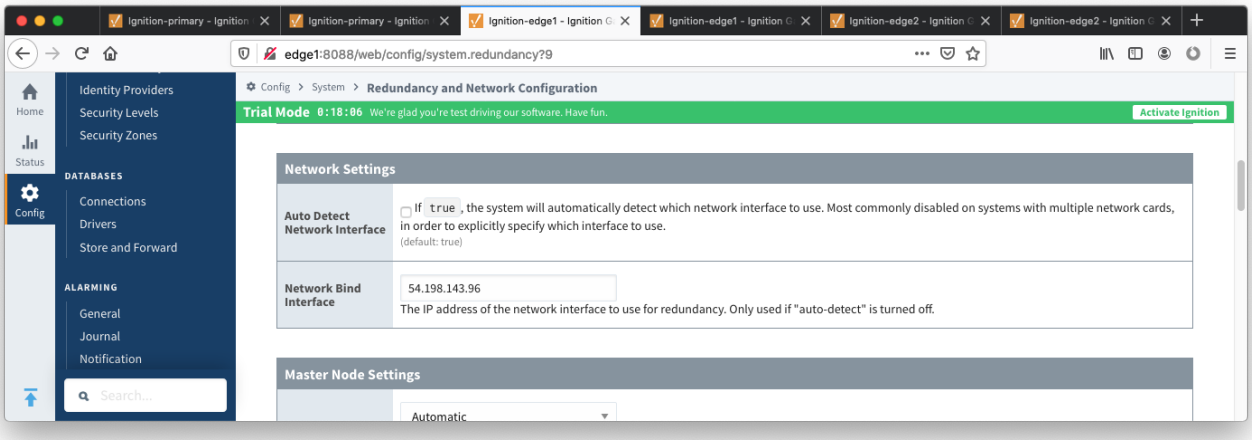

- Finally, click the 'Save Changes' button.

## Ignition Edge 1

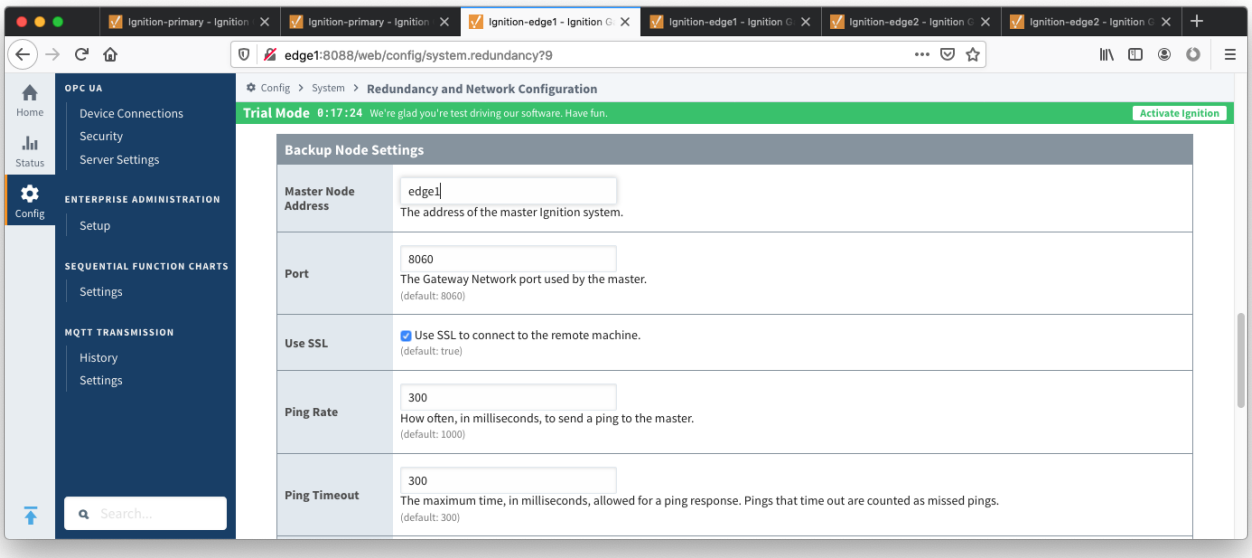
- Select Redundancy on the left navigation bar. Then set the Mode to 'Master' and set the Standby Activity left to 'Warm' as shown below.




- Set up the Redundancy Network Settings.  The settings here are specific to your network setup.  **On many LAN configurations none of these changes are required**.  What is shown below was the configuration for setting up all of these components in Amazon's AWS EC2 instances.  The changes were:
    - Uncheck 'Auto-detect network interface'
    - Set the 'Network Bind Interface' to the public IP address of the Ignition Edge 1 EC2 instance.  On a LAN this would be the primary network interface address of the Ignition Edge 1 machine.



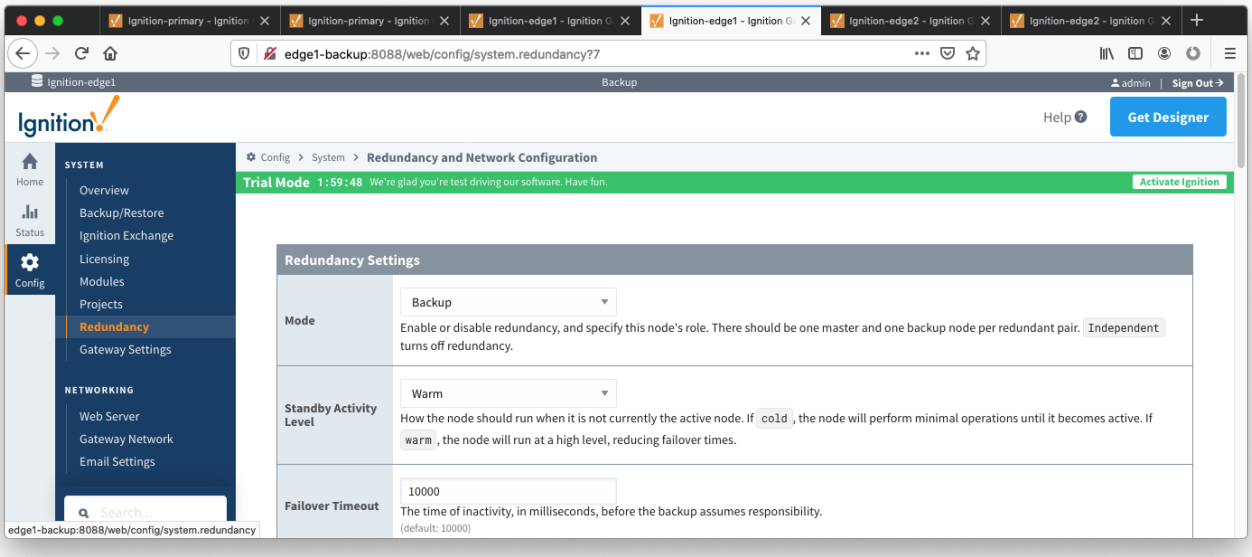

- Set the Master Node Address.  Note in the configuration below a hostname is being used.  This should be the primary network interface address of the Ignition Edge 1 Gateway.



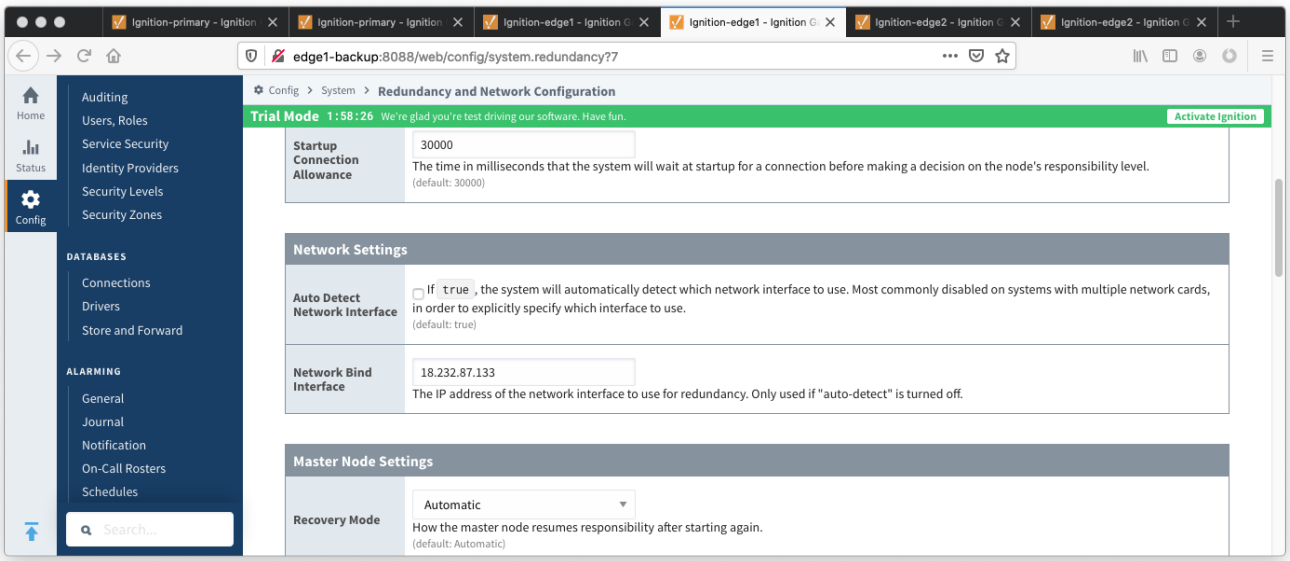

- Finally, click the 'Save Changes' button.

## Ignition Edge 1 Backup

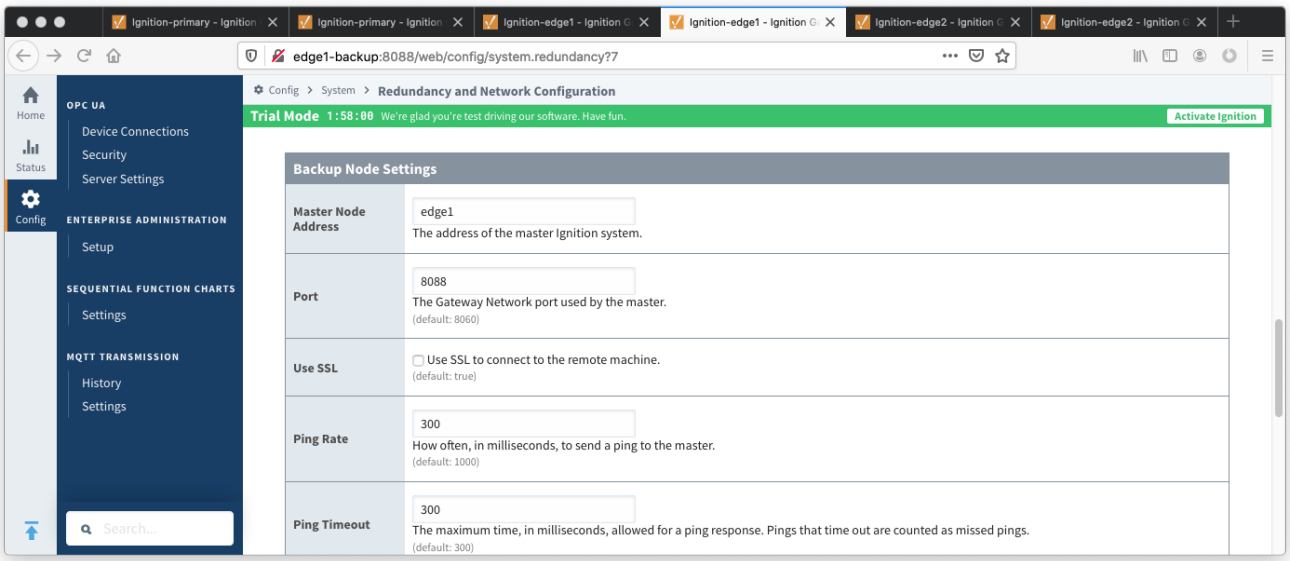
- Select Redundancy on the left navigation bar. Then set the Mode to 'Backup' and set the Standby Activity left to 'Warm' as shown below.




- Set up the Redundancy Network Settings. The settings here are specific to your network setup. **On many LAN configurations none of these changes are required**. What is shown below was the configuration for setting up all of these components in Amazon's AWS EC2 instances. The changes were:
  - Uncheck 'Auto-detect network interface'
  - Set the 'Network Bind Interface' to the public IP address of the Ignition Edge 1 Backup EC2 instance. On a LAN this would be the primary network interface address of the Ignition Edge 1 Backup machine.



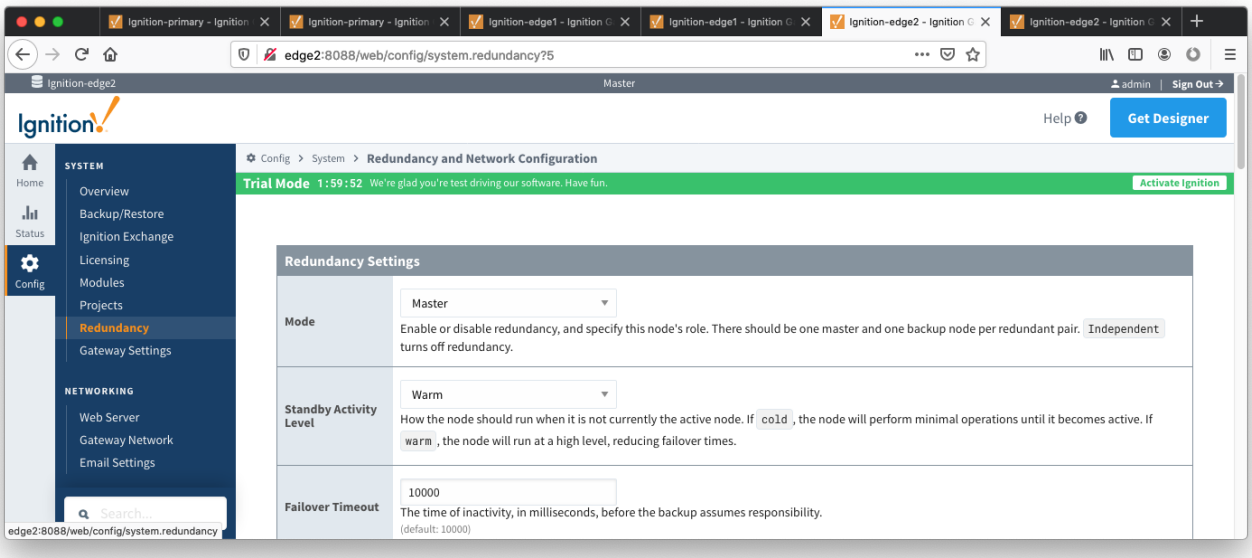

- Set the Master Node Address. Note in the configuration below a hostname is being used. This should be the primary network interface address of the Ignition Edge 1 Gateway.



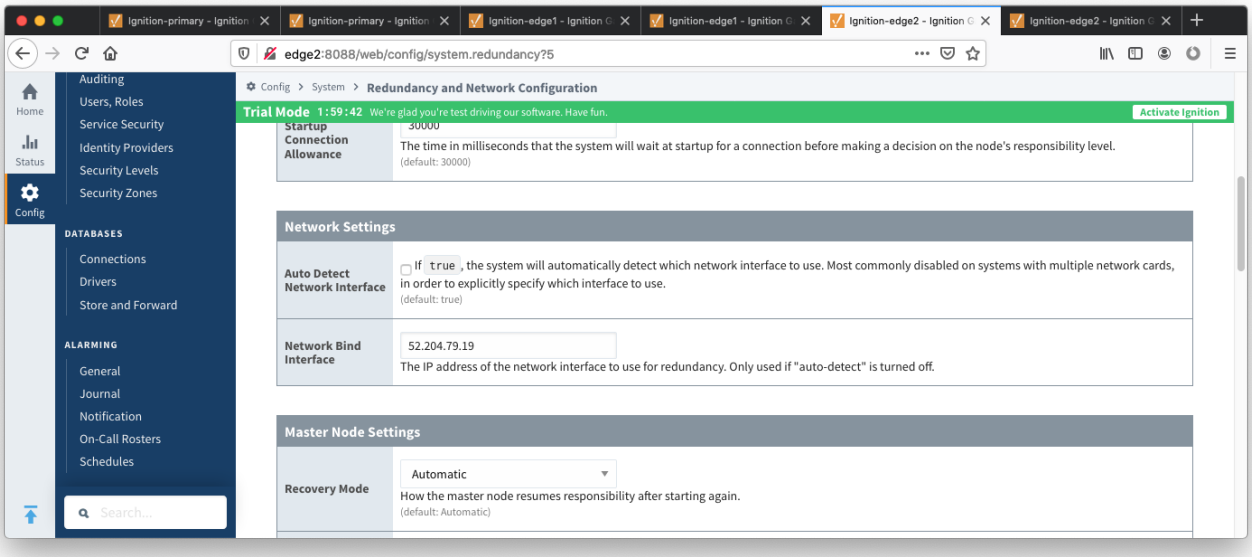

- Finally, click the 'Save Changes' button.

## Ignition Edge 2

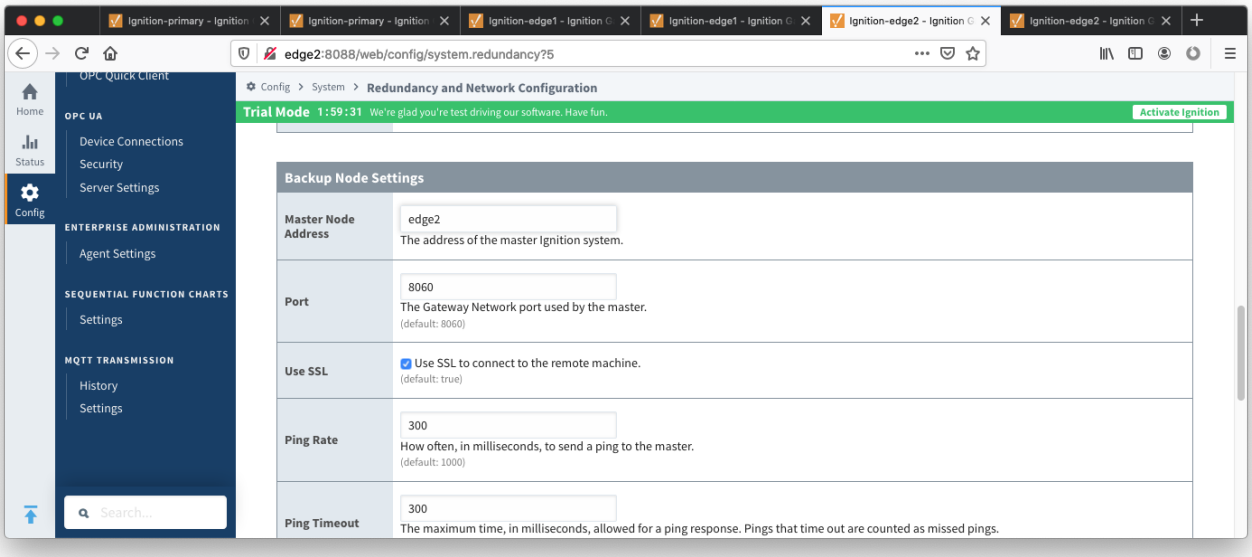
- Select Redundancy on the left navigation bar. Then set the Mode to 'Master' and set the Standby Activity left to 'Warm' as shown below.




- Set up the Redundancy Network Settings. The settings here are specific to your network setup. **On many LAN configurations none of these changes are required**. What is shown below was the configuration for setting up all of these components in Amazon's AWS EC2 instances. The changes were:
  - Uncheck 'Auto-detect network interface'
  - Set the 'Network Bind Interface' to the public IP address of the Ignition Edge 2 EC2 instance. On a LAN this would be the primary network interface address of the Ignition Edge 2 machine.



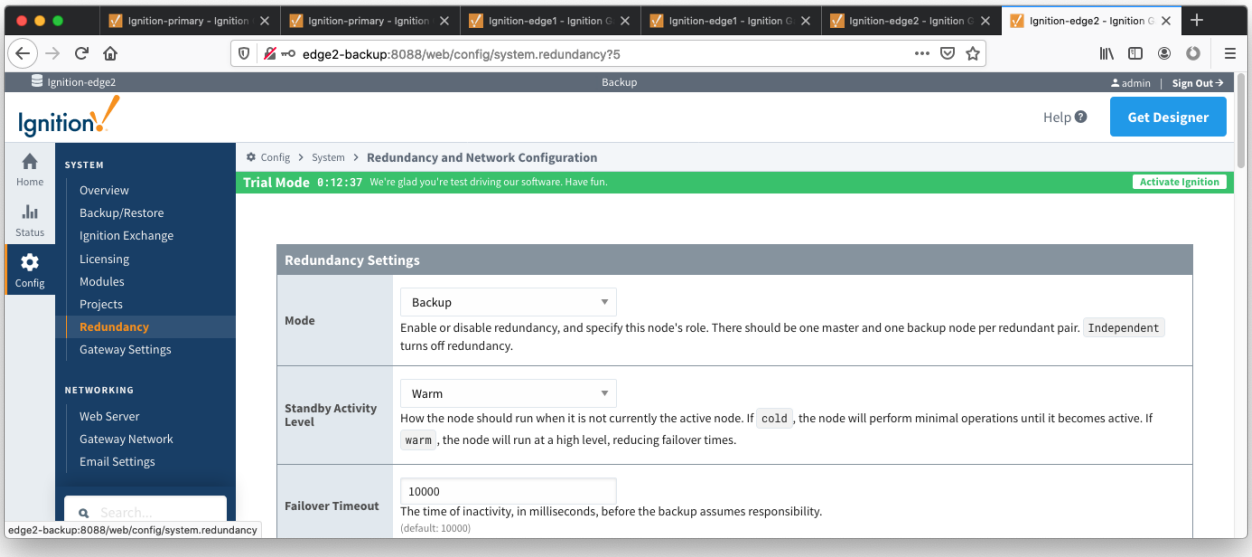

- Set the Master Node Address.  Note in the configuration below a hostname is being used.  This should be the primary network interface address of the Ignition Edge 2 Gateway.



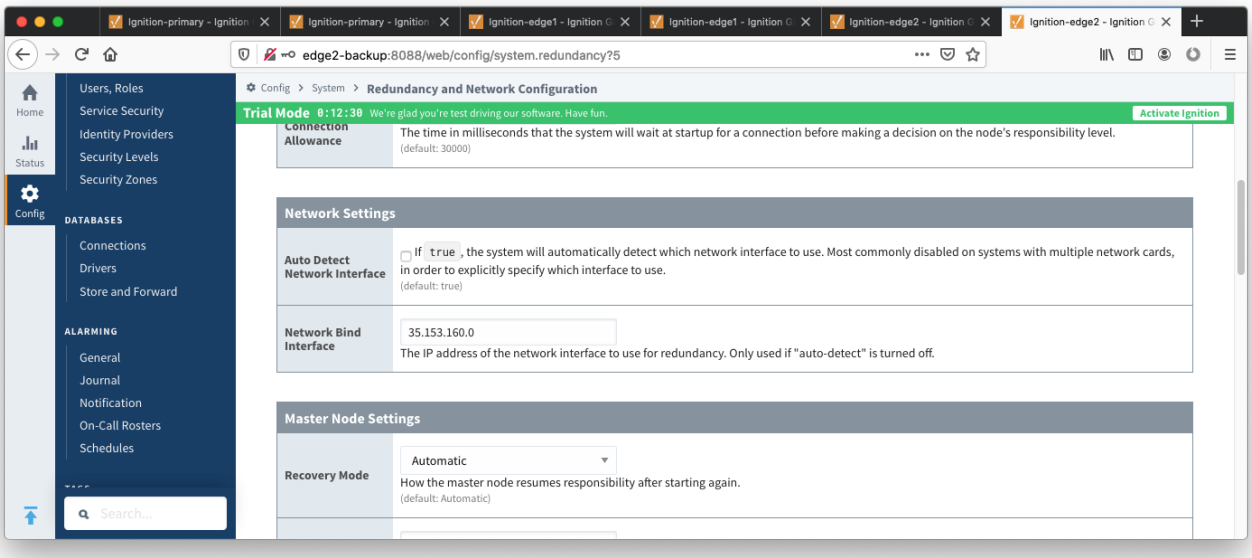

- Finally, click the 'Save Changes' button.

## Ignition Edge 2 Backup

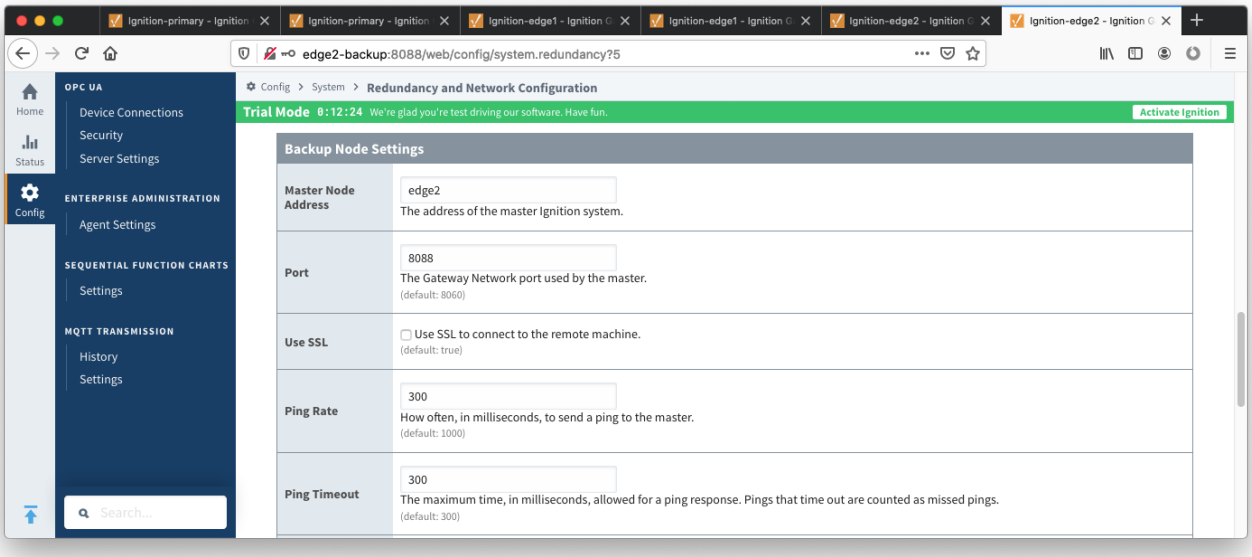
- Select Redundancy on the left navigation bar. Then set the Mode to 'Backup' and set the Standby Activity left to 'Warm' as shown below.




- Set up the Redundancy Network Settings. The settings here are specific to your network setup. **On many LAN configurations none of these changes are required**. What is shown below was the configuration for setting up all of these components in Amazon's AWS EC2 instances. The changes were:
  - Uncheck 'Auto-detect network interface'
  - Set the 'Network Bind Interface' to the public IP address of the Ignition Edge 2 Backup EC2 instance. On a LAN this would be the primary network interface address of the Ignition Edge 2 Backup machine.




- Set the Master Node Address. Note in the configuration below a hostname is being used. This should be the primary network interface address of the Ignition Edge 2 Gateway.

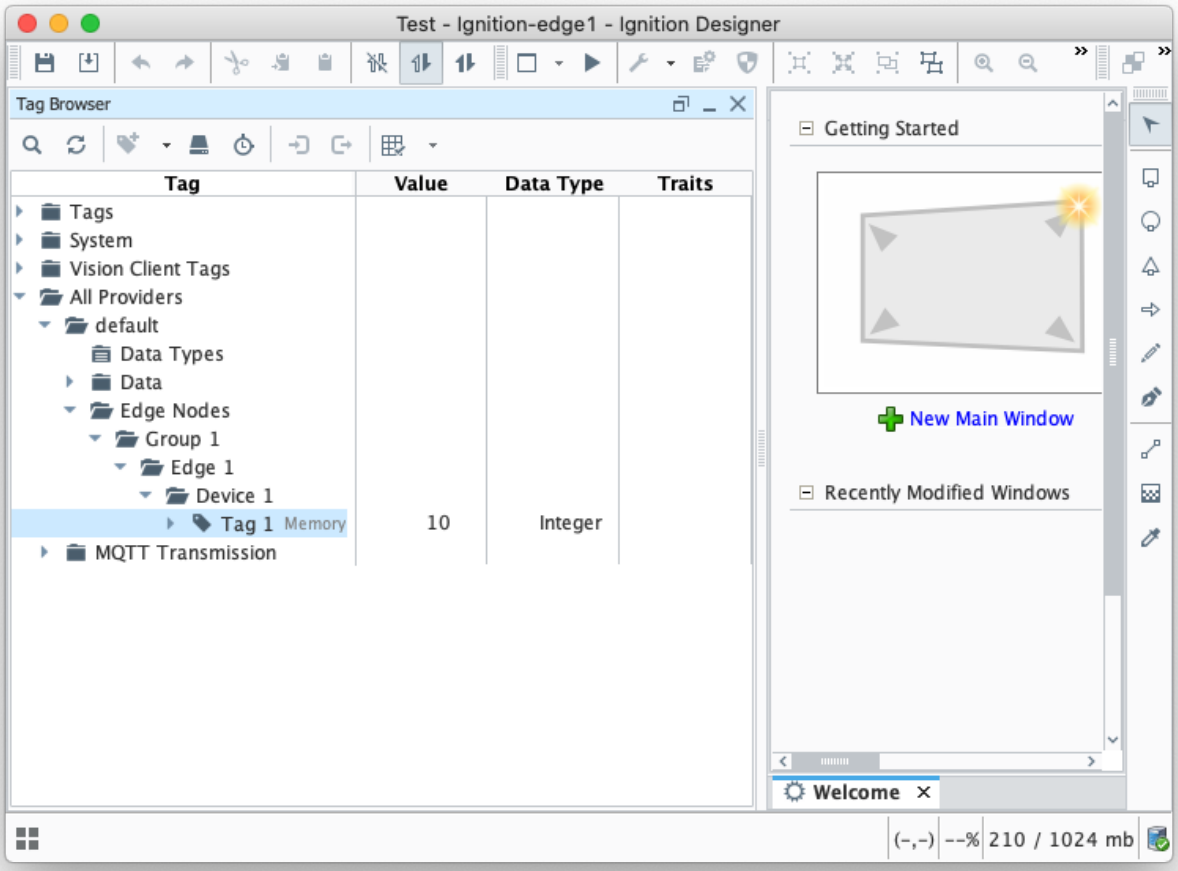



- Finally, click the 'Save Changes' button.

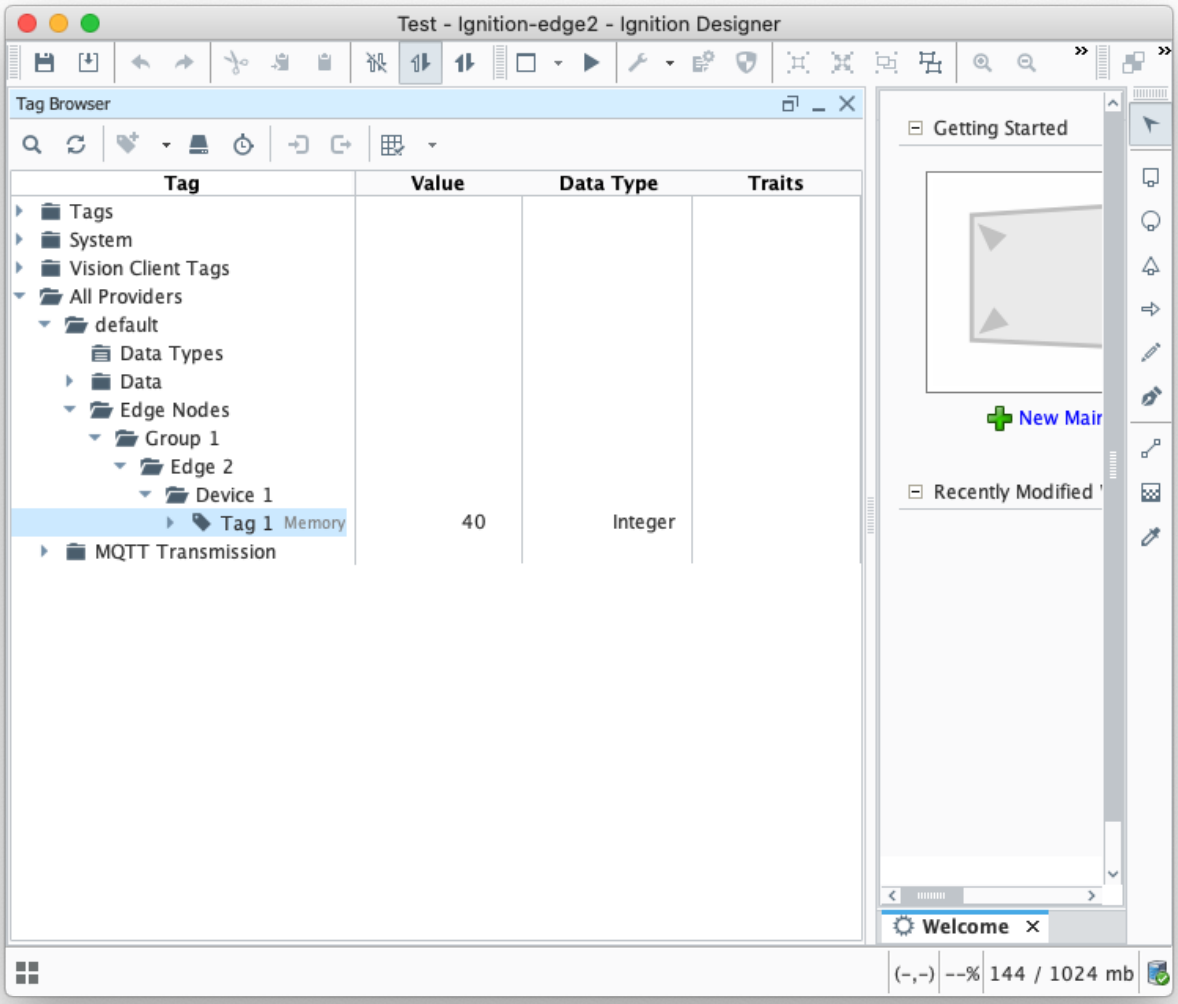
## Step 5: Create some tags in Edge 1 and Edge 2

In this step we will use Ignition Designer in the Ignition Edge 1 and Edge 2 instances to create some tags. These tags will be used by MQTT Transmission and the Default Transmitter to push data to MQTT Distributor and MQTT Engine in the Ignition Primary instances.

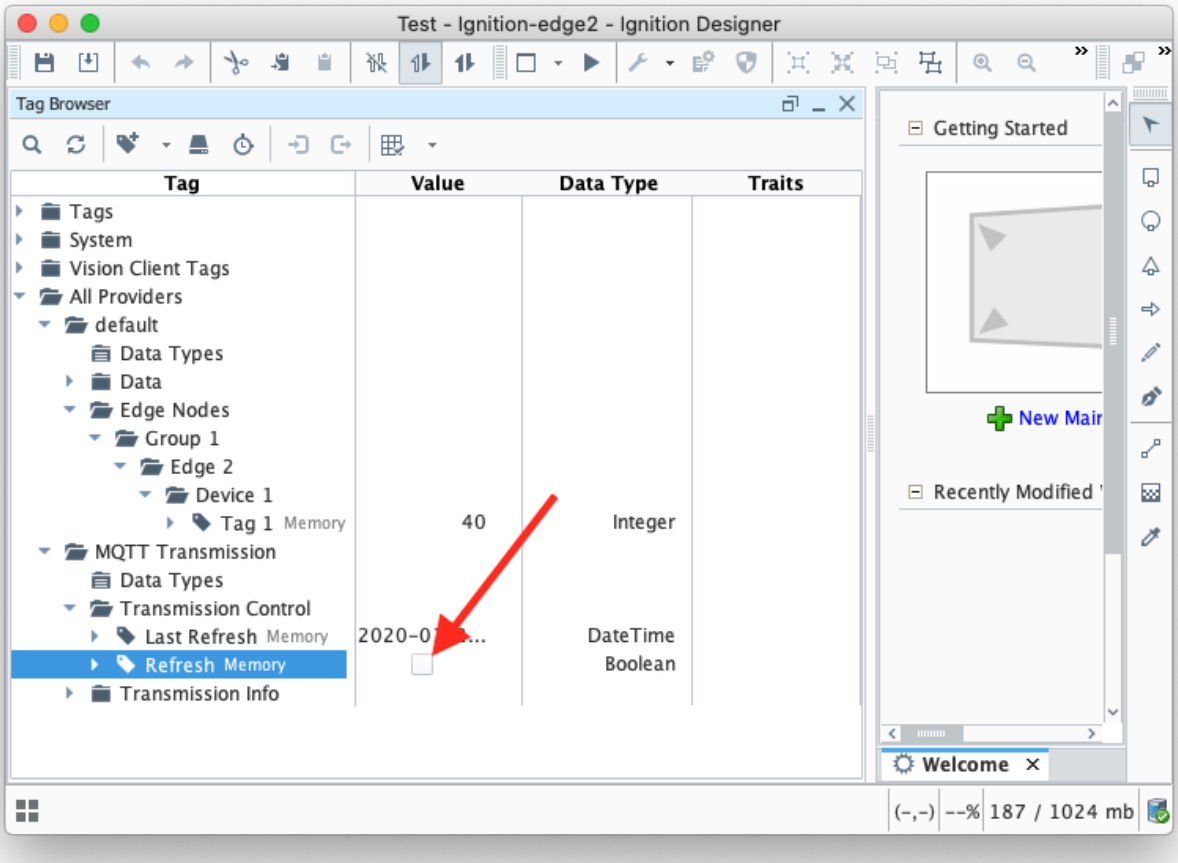
- Using Ignition Designer on Edge 1, do the following:
    - Create a new project called 'Test'.
    - In the Tag Browser, confirm there is a folder called 'Edge Nodes'. If there is not, confirm MQTT Transmission is installed.
    - In the 'Edge Nodes' folder, create a folder called 'Group 1'.
    - In the 'Group 1' folder, create a folder called 'Edge 1'.
    - In the 'Edge 1' folder, create a folder called 'Device 1'.
    - In the 'Device 1' folder, create a Tag called 'Tag 1'.
    - At the end, you should see something similar to what is shown below.



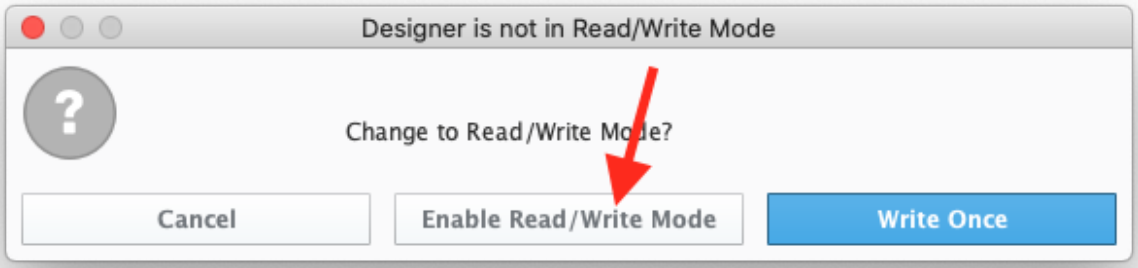

- Using Ignition Designer on Edge 2, do the following:
  - Create a new project called 'Test'.
  - In the Tag Browser, confirm there is a folder called 'Edge Nodes'.  If there is not, confirm MQTT Transmission is installed.
  - In the 'Edge Nodes' folder, create a folder called 'Group 1'.
  - In the 'Group 1' folder, create a folder called 'Edge 2'.
  - In the 'Edge 2' folder, create a folder called 'Device 1'.
  - In the 'Device 1' folder, create a Tag called 'Tag 1'.
  - At the end, you should see something similar to what is shown below




- Finally, refresh the Transmission runtime.  This is done by clicking the 'MQTT Transmission/Transmission Control/Refresh' Boolean tag.



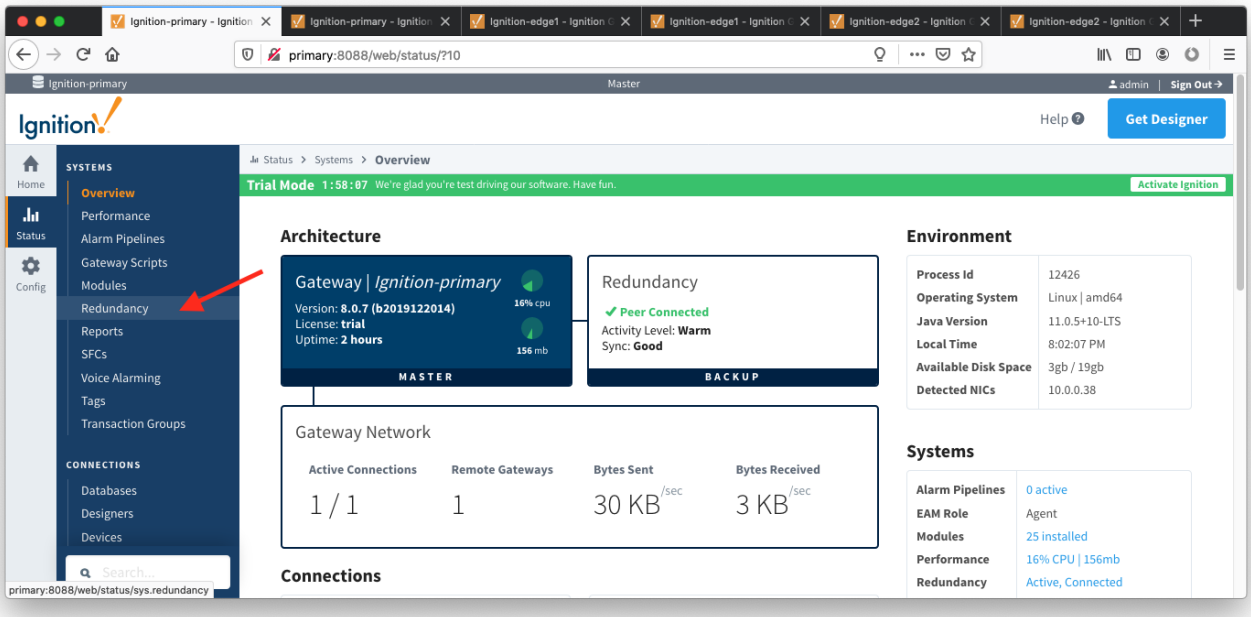

- After clicking the Boolean tag you may need to 'Enable



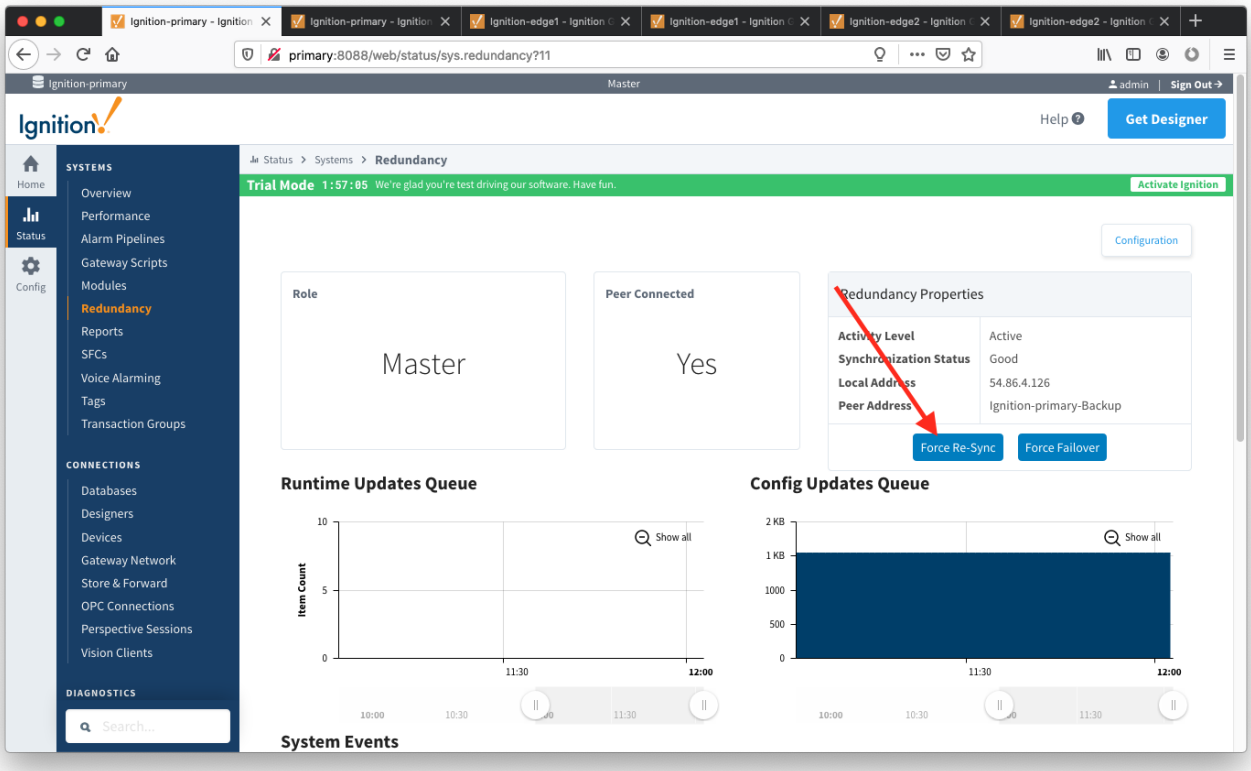

## Step 6: Force Sync of Backup Instances

The next step to to force the backup Ignition instances to receive their configurations from the master Ignition instances.  This will happen automatically eventually but for expediency we're going to force the action.

- Repeat the following steps for the following Ignition instances: Primary, Edge 1, and Edge 2
    - Browse to the Status menu and then click in the Redundancy box as shown below.



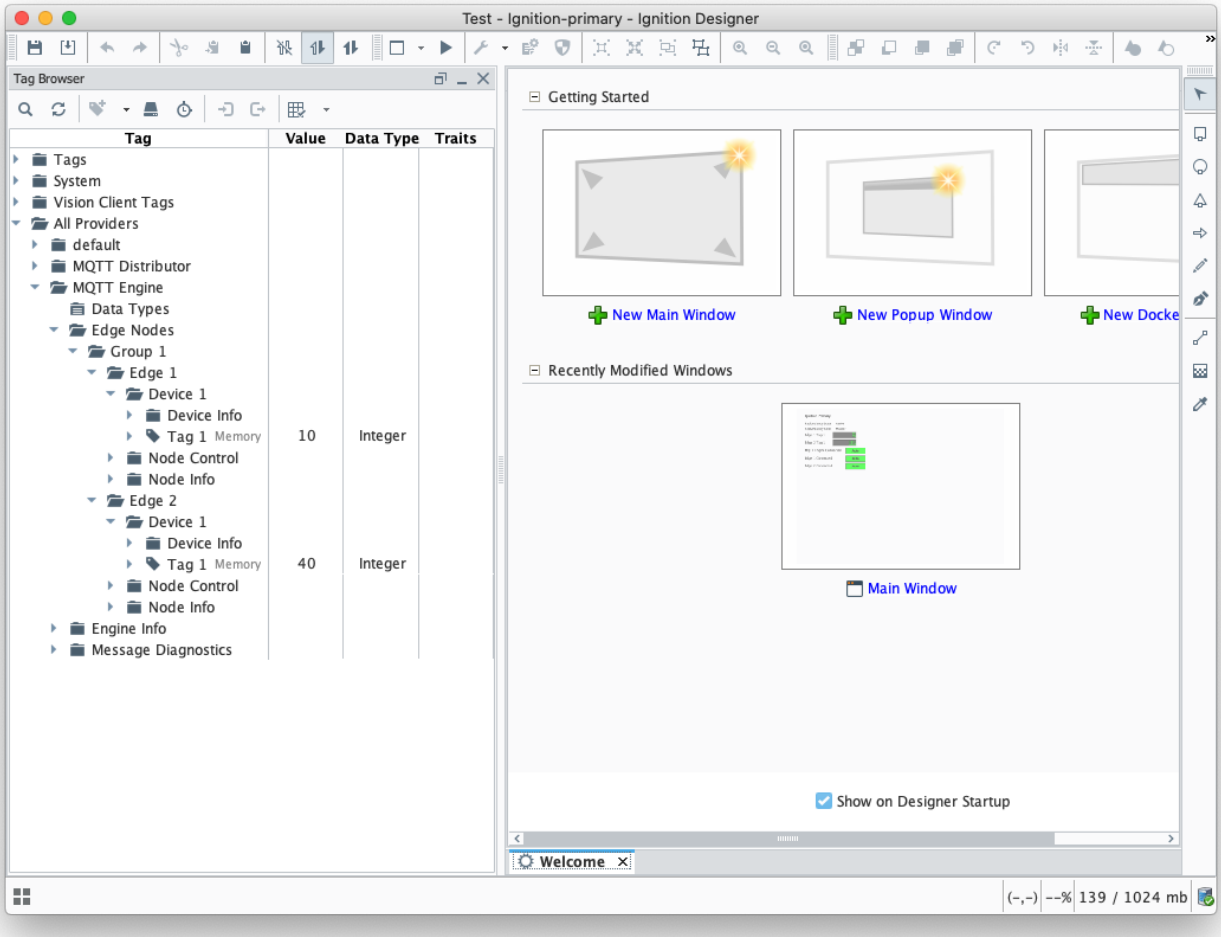

- Force the Re-sync via the button below. This will duplicate the master configuration to the backup Ignition instance it is connected to.




- Repeat the above Re-sync steps on the Edge 1 and Edge 2 Ignition instances.

## Step 7: Verify MQTT Engine is getting data from the MQTT Transmission Edge 1 and Edge 2

Open Ignition Designer on the Ignition Primary instance. Expand the MQTT Engine tag tree and validate the following tags are present. If they are present and not stale, they are properly connected.

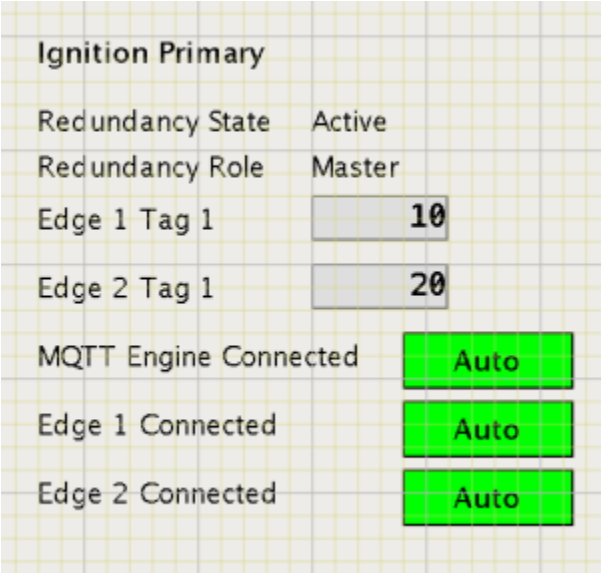



## Step 8: Test the Redundancy

In order to test the redundancy, we need to make a few simple dashboards. It is important to note that this can not be tested with Ignition Designer alone. Designer can not be opened from an Ignition backup instance since projects get replicated to the backup instances. So, to show everything working, we'll make some very rudimentary dashboards.

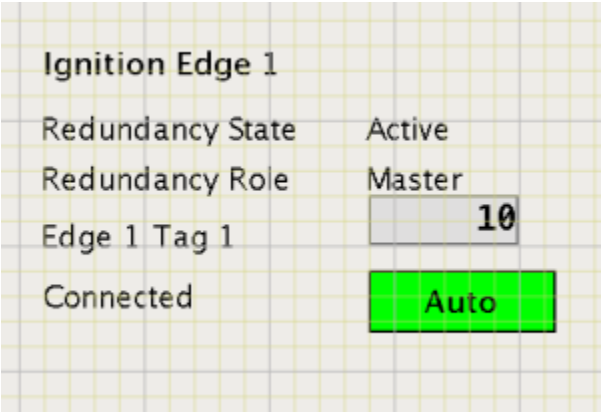
### Ignition Primary

- Create the following widgets.
  - Label - Ignition Primary
  - Label - Redundancy State Label with Tag Path of "[MQTT Engine]/Engine Info/Redundancy State"
  - Label - Redundancy Role Label with Tag Path of "[MQTT Engine]/Engine Info/Redundancy Role"
  - Label - Edge 1 Tag 1 Label with Tag Path of "[MQTT Engine]/Edge Nodes/Group 1/Edge 1/Device 1/Tag 1"
  - Label - Edge 2 Tag 1 Label with Tag Path of "[MQTT Engine]/Edge Nodes/Group 1/Edge 2/Device 1/Tag 1"
  - Label - MQTT Engine Connected Multi-State Indicator with Tag Path of "[MQTT Engine]/Engine Info/MQTT Clients/Chariot SCADA /Online"
  - Label - Edge 1 Connected Multi-State Indicator with Tag Path of "[MQTT Engine]/Edge Nodes/Group 1/Edge 1/Node Info/Online"
  - Label - Edge 2 Connected Multi-State Indicator with Tag Path of "[MQTT Engine]/Edge Nodes/Group 1/Edge 2/Node Info/Online"
- When complete - it should look similar to what is shown below



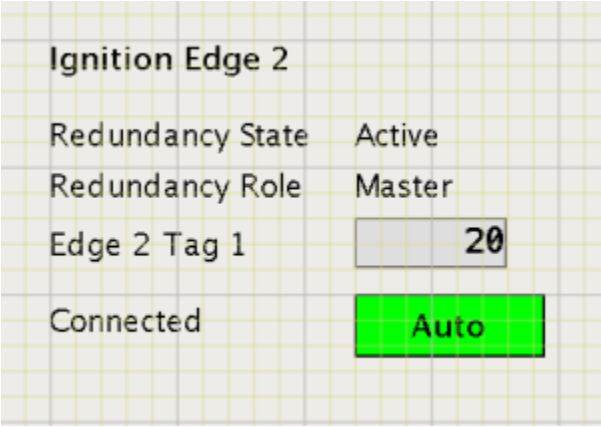

### Ignition Edge 1

- Create the following widgets
  - Label - Ignition Edge 1
  - Label - Redundancy State  Label with Tag Path of "[MQTT Transmission]/Transmission Info/Redundancy State"
  - Label - Redundancy Role  Label with Tag Path of "[MQTT Transmission]/Transmission Info/Redundancy Role"
  - Label - Edge 1 Tag 1  Label with Tag Path of "[default]/Edge Nodes/Group 1/Edge 1/Device 1/Tag 1"
  - Label - Connected  Multi-State Indicator with Tag Path of "[MQTT Transmission]/Transmission Info/MQTT Clients/Group 1-Edge 1 /Online"

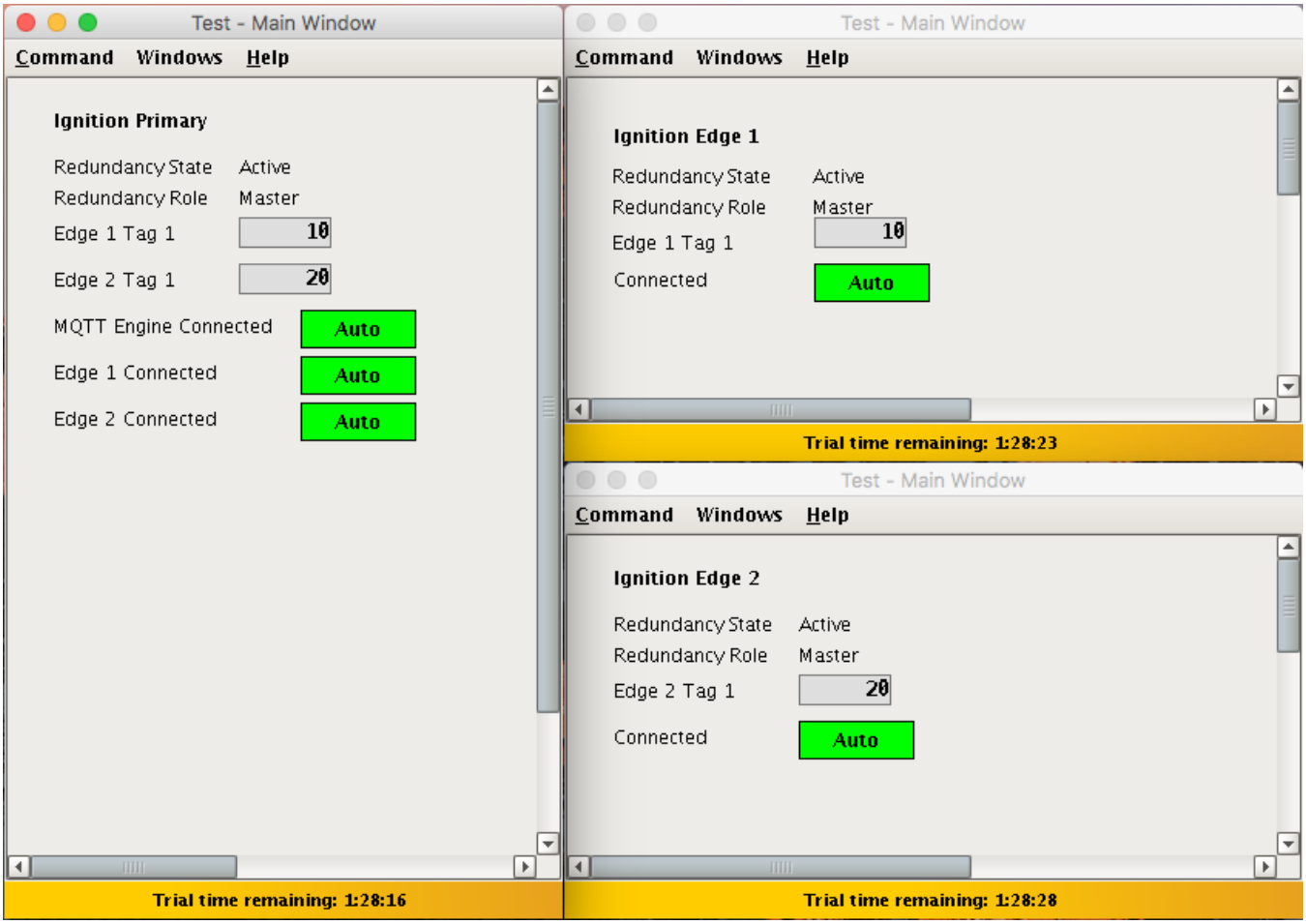



### Ignition Edge 2

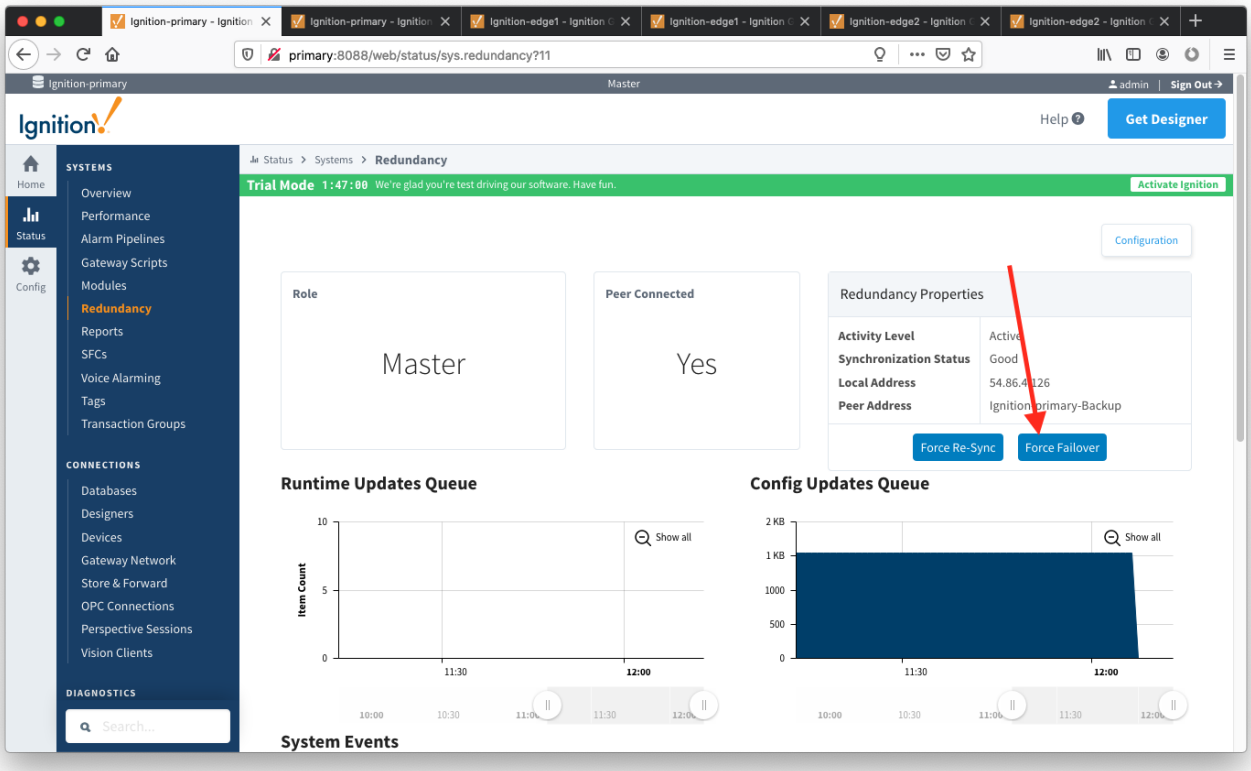
- Create the following widgets
  - Label - Ignition Edge 2
  - Label - Redundancy State  Label with Tag Path of "[MQTT Transmission]/Transmission Info/Redundancy State"
  - Label - Redundancy Role  Label with Tag Path of "[MQTT Transmission]/Transmission Info/Redundancy Role"
  - Label - Edge 2 Tag 1  Label with Tag Path of "[default]/Edge Nodes/Group 1/Edge 2/Device 1/Tag 1"
  - Label - Connected  Multi-State Indicator with Tag Path of "[MQTT Transmission]/Transmission Info/MQTT Clients/Group 1-Edge 2 /Online"




- Once all three dashboards have been created, save and publish them and close the Ignition Designer windows.
- Now open each of the Ignition client 'Test' projects. With everything running you should see three windows similar to the following.



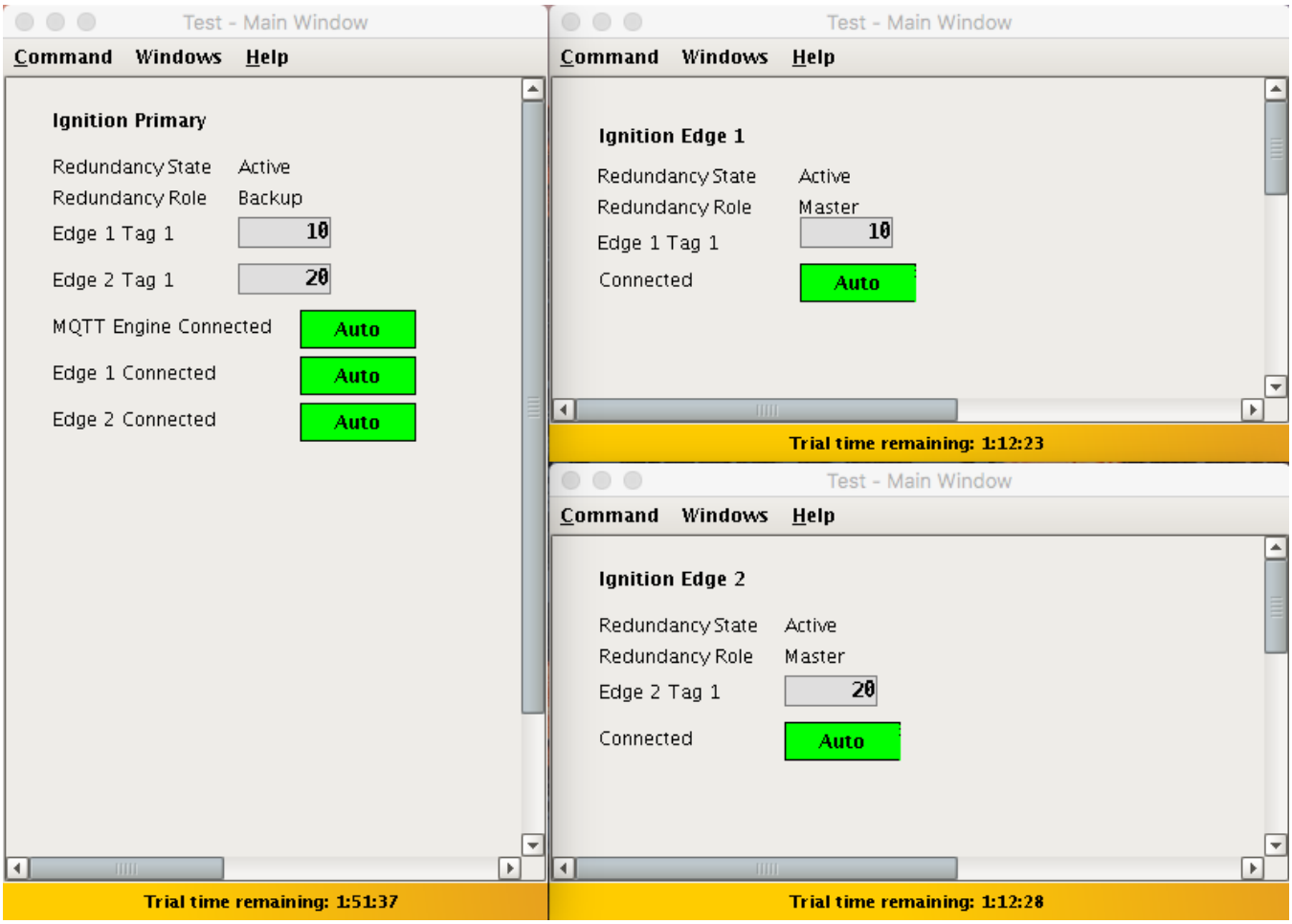

- At this point we can begin failing Ignition instances. From the Status Redundancy page we can use the 'Force Failover' button as shown below. Of course stopping the actual Ignition instance is another option.




## Primary Ignition Failure

Failing the Ignition Primary instance will cause the following.

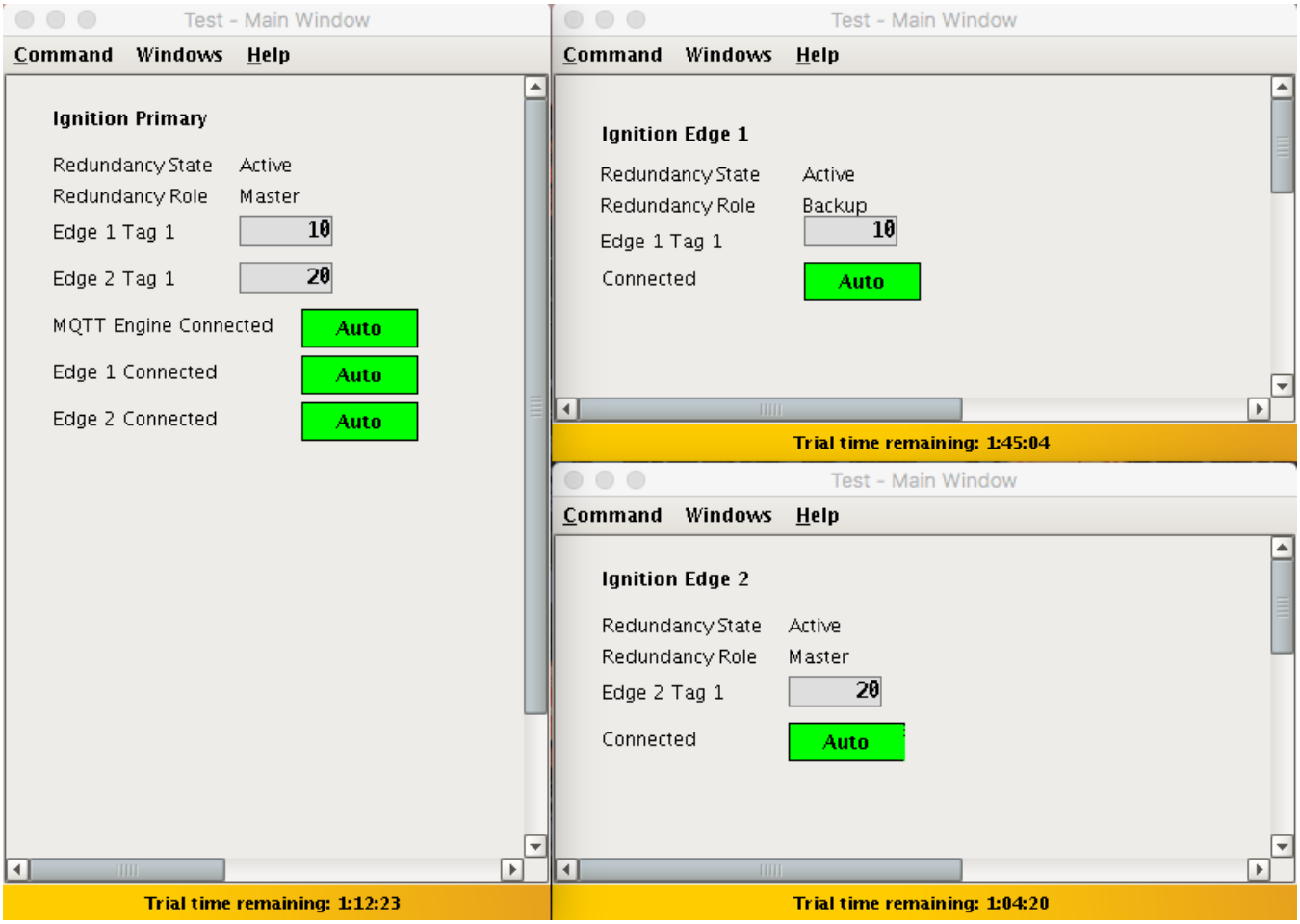
- Ignition Primary will go down and be unreachable
    - This results in all MQTT connections being lost.
- Ignition Primary Backup will come up and take the place of Ignition Primary
- MQTT Engine will reconnect on Ignition Primary
- The MQTT Transmission instances will reconnect to the new MQTT Server (MQTT Distributor) running in Ignition Primary Backup
- This is all shown in the screenshot below of the Ignition projects.  Note all connections are valid and the 'Redundancy Role' of Ignition Primary is now Backup.




### Edge 1 Failure

Failing the Ignition Edge 1 instance will cause the following.

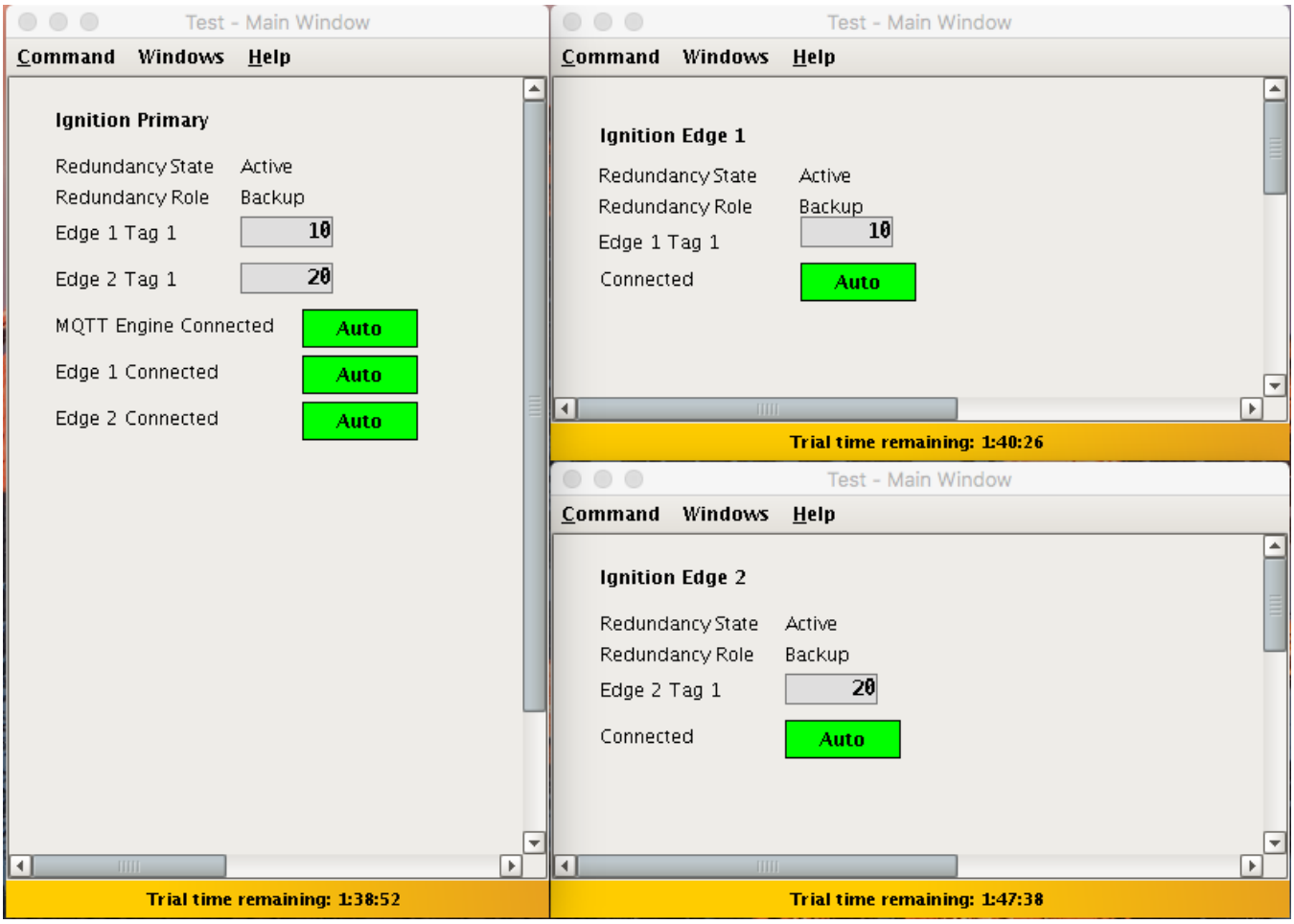
- Ignition Edge 1 will go down and be unreachable
    - This results in the MQTT Connection between Ignition Edge 1 and Ignition Primary being lost.
- Ignition Edge 1 Backup will come up and take the place of Ignition Edge 1.
- The MQTT Transmission instance on Ignition Edge 1 Backup will connect to the MQTT Server (MQTT Distributor) running in Ignition Primary
- This is all shown in the screenshot below of the Ignition projects.  Note all connections are valid and the 'Redundancy Role' of Ignition Edge 1 is now Backup.




### Failure of all Master Nodes

Failing all master Ignition instances (Primary, Edge 1, and Edge 2) will cause the following.

- Ignition Primary, Edge 1, and Edge 2 will all go down and be unreachable
    - This results in all MQTT connections being lost
- Ignition Primary Backup, Edge 1 Backup, and Edge 2 Backup will all come up and start their MQTT services.
- The new MQTT Transmission instances on Ignition Edge 1 Backup and Ignition Edge 2 Backup will connect to the new MQTT Server (MQTT Distributor) running in Ignition Primary Backup
- This is all shown in the screenshot below of the Ignition projects.  Note all connections are valid and the 'Redundancy Role' of all instances is now Backup.




To summarize, this tutorial shows how Ignition and the MQTT Modules can be used to create a resilient infrastructure that is able to withstand failures of machines and network connections within the architecture. As noted earlier, this tutorial shows the basic requirements of configuring failover support with Ignition and the MQTT Modules. This can be further improved with additional advanced concepts. Feel free to contact sales@cirrus-link.com for more information.