Requirements:

In order to establish the connection with Redfish Server, the Redfish Server needs to be running on the configured (see Configuration below) IP Address and Port.

Configuration:

Before configuring variables using RedfishConfig.efi, it is important to update the uefi image. Otherwise unforeseen errors might be seen when running the RedfishConfig.efi application.
To update the uefi image, copy the default.bfb provided over to the Bluefield ARM OS and run the following:
/opt/mellanox/scripts/bfrec --bootctl ./default.bfb
After running the above command look for a confirmation message saying that platform configuration is successful. Now reboot the machine and continue on to boot to UEFI Shell for below configuration.

It is required to configure several variables to establish communication between UEFI Redfish Client and Redfish Service (Web Server). To aid in this process, a RedfishConfig.efi application is provided. Place RedfishConfig.efi on the ESP using OS. This UEFI application will set the following variables:
HostIpAddress HostIpMask HostMacAddress RedfishServiceIpAddress RedfishServiceIpMask RedfishServiceIpPort

In order to use RedfishConfig.efi, it is necessary to boot to the EFI internal shell:
UEFI menu->Boot Manager->Efi Internal Shell

Once booted to UEFI Shell, select FS0 (fs0:) and list (ls) files to check whether the application is present. Now run the application passing the variables mentioned above in respective order.

NOTE: For IPv6, use respective address format

Example: RedfishConfig.efi 192.168.101.13 255.255.255.0 001ACAD94A88 192.168.101.1 255.255.255.0 5000

Where:
192.168.101.13 - **HostIpAddress (Ip address to be used on the host interface)**
255.255.255.0 - **HostIpMask (Subnet mask to be used on the host interface)**
001ACAD94A88 - **HostMacAddress (Mac address of the interface to be used)**
192.168.101.1 - **RedfishServiceIpAddress (Redfish Server Ip Address)**
255.255.255.0 - **RedfishServiceIpMask (Redfish Server Subnet mask)**
5000 - **RedfishServiceIpPort (Port where Redfish Server is listening)**

On a successful configuration, a confirmation message will be seen on the console.

NOTE: If using IPv6, an additional step is required to configure the IP interface from the UEFI menu -> Device Manager -> Network Device List -> Interface (base on MAC) -> IPv6 Configuration -> Advanced Configuration

Set the IPV6 Address of Host Interface here (it should be the same as the one used above in RedfishConfig.efi). Save and Commit the changes. Save and Exit.

After this Restart/Reset the system.

Expectation/Output:

For IPv6:
On reboot after successful configuration, the Redfish UEFI Client will perform a GET on /redfish/v1/, this can be captured if some kind of logging is enabled on the server side or using an application like tcpdump.

For IPv4:
On reboot after successful configuration, the Redfish UEFI Client will perform a PATCH Request on /redfish/v1/Systems/System-1 with SKU of the Bluefield System. In order for SKU to be sent over a PATCH HTTP Request. The MFG partition of the system needs to be written with system related information. Please see the Notes section below in order to write to the MFG partition.

Notes:

1. Tested using OOB interface on Bluefield platform.
2. If OOB MAC is not already set to a fixed value, it needs to be fixed so it doesn't randomize with every reboot. See below for instructions on how to provision the OOB to a fixed MAC address.

Provision OOB to a fixed MAC:

OOB MAC can be found on the Bluefield card tag. Boot into Bluefield image OS and then perform the following steps:

1. **cd /sys/bus/platform/drivers/mlx-bootctl**
2. **echo XX:XX:XX:XX:XX:XX > oob_mac**
3. **echo 1 > mfg_lock**

Provisioning the MFG partition:

Before resetting the EEPROM MFG/pushing BFB:

1. Gather the following info: OOB MAC, OPN, SKU, MODL, SN and UUID

OOB MAC can be read from the EEPROM by running: “bfcfg -d”. It can also be read from ifconfig or ip from the BF2 ARM.

The VPD info can be gathered using “mlxvpd -d <mst/pci device>”, either from the PCIe host or from the BF2 ARM. (The target device for mlxvpd can either be the PCIe bdf of the HCA or the path to its MST device under /dev/mst/)

2. After collecting the info, initiate a reboot of the BF2 ARM and stop at UEFI by pressing the ESC key. In the UEFI Menu, go to Device Manager -> System Configuration -> Reset MFG Info and hit ENTER. This will clear the MFG partition of the EEPROM. Please note that the EFI variables need not be cleared.

After resetting the EEPROM MFG:

1. After booting to Linux on BF2, create /etc/bf.cfg config file with the following fields populated:

   MFG_OOB_MAC=<oob_mac>

   MFG_OPN=<opn>

   MFG_SKU=<sku>

   MFG_MODL=<model_number>

   MFG_SN=<serial_number>

   MFG_UUID=<uuid>

Please note that the tool expects ALL the above fields to be populated in bf.cfg. If any of these are missing, the info will not be written to the EEPROM.

2. Once bf.cfg is populated, run “bfcfg”. The bfcfg tool will write all the above fields to the EEPROM MFG. You may check the logfile /tmp/bfcfg.log to see if the write to the EEPROM was successful. On success, the MFG partition will be locked and further writes to it will not be allowed. “bfcfg -d” will now print the updated EEPROM MFG variables.