

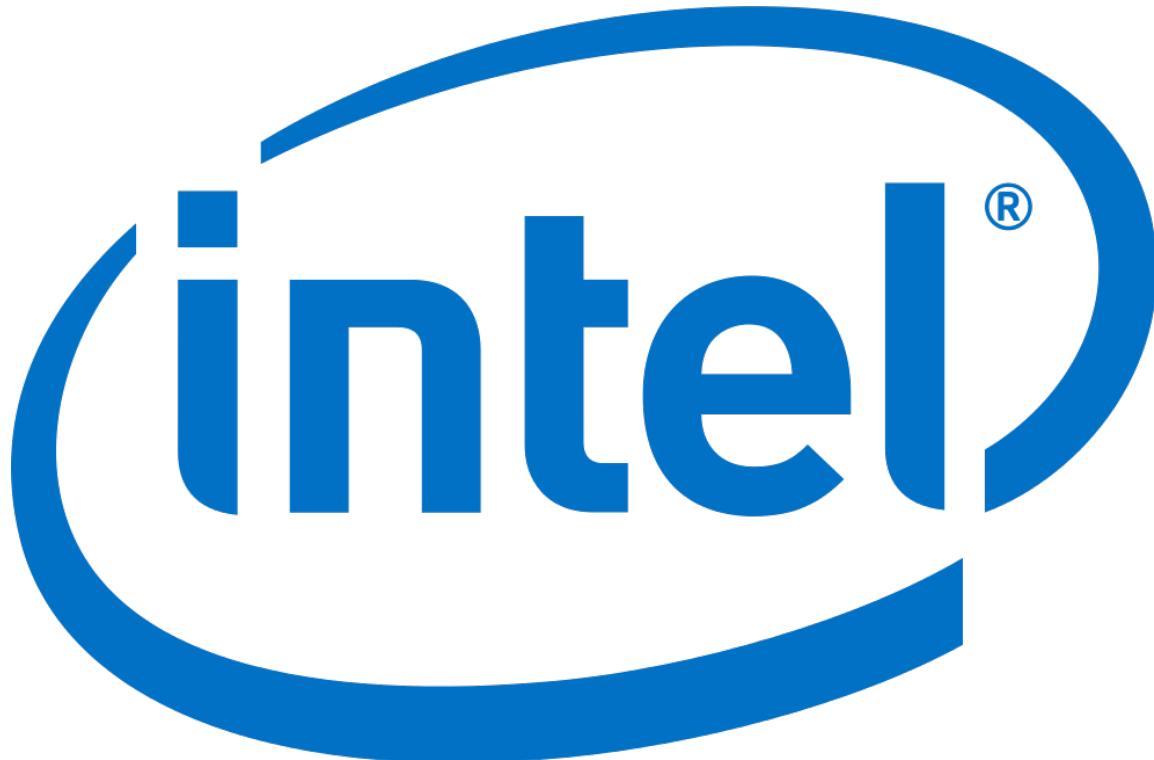
Intel IoT Gateway / Arduino 101 with AWS IoT

Hands-On Workshop

Authors:

Martin Kronberg | martin.kronberg@intel.com | [@martinkronberg](https://twitter.com/martinkronberg)

Priyanka Bagade | priyanka.bagade@intel.com | [@bagadepriyanka](https://twitter.com/bagadepriyanka)



[Table of Contents](#)

Topics covered: Intel IoT Gateway, Grove sensors, Node-RED, and AWS Internet of Things Platform

In this lab, we will unbox and set up an Intel IoT Gateway and the Arduino 101 board (with a Grove Starter kit)

a. Using Node-RED, running on the Intel NUC Gateway... ADD DESCRIPTION OF FINAL WORKSHOP MATERIAL

Getting Started with Grove IoT Commercial Developer Kit

Step 1 - Unboxing Grove IoT Commercial Developer Kit and Arduino 101

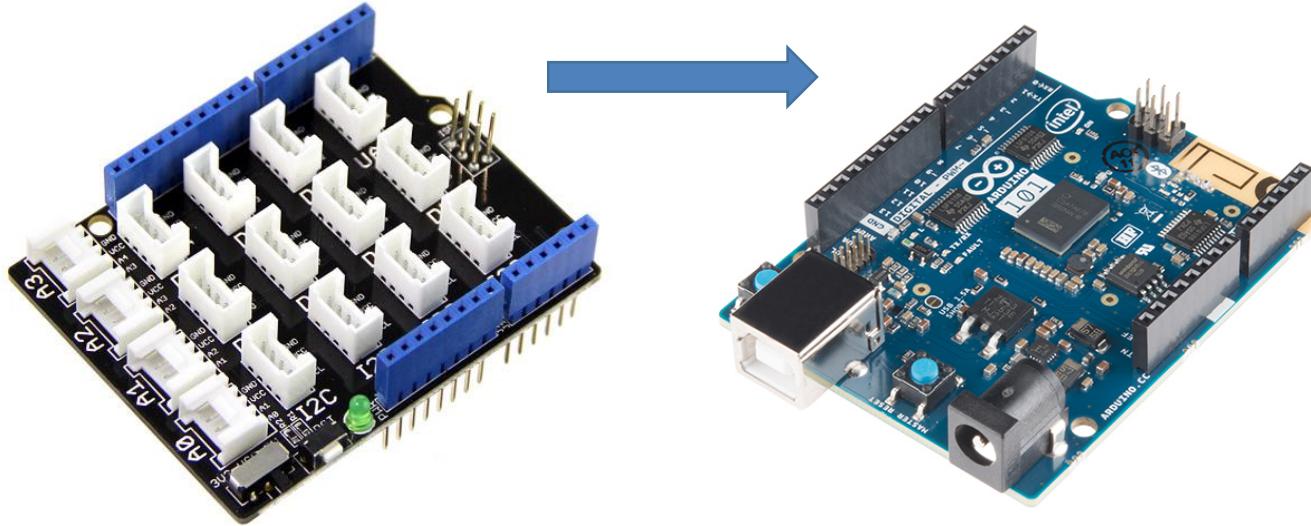


Arduino 101 is not included in Grove IoT Commercial Development Kit and will be given separately.

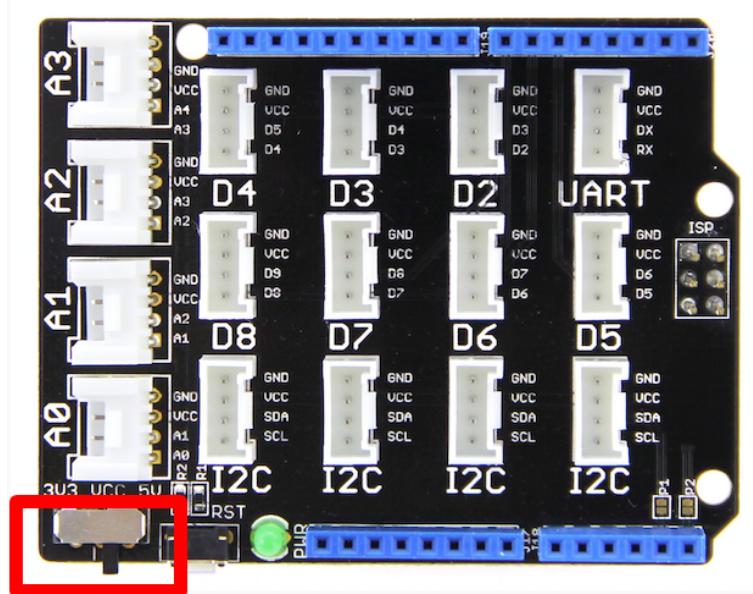


Step 2 – Connect Base shield to Arduino 101

Base shield will be included in the Grove starter kit. Attach the Grove base shield to the Arduino 101.

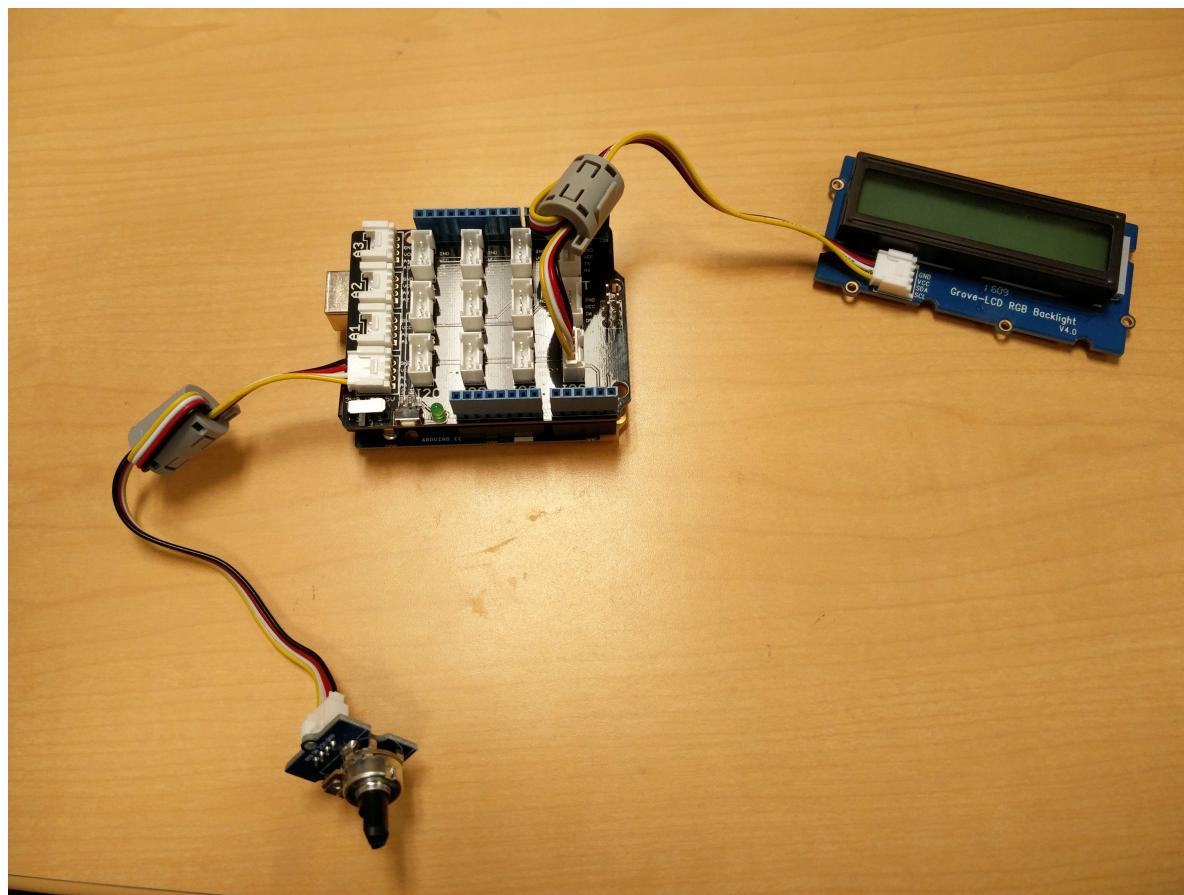
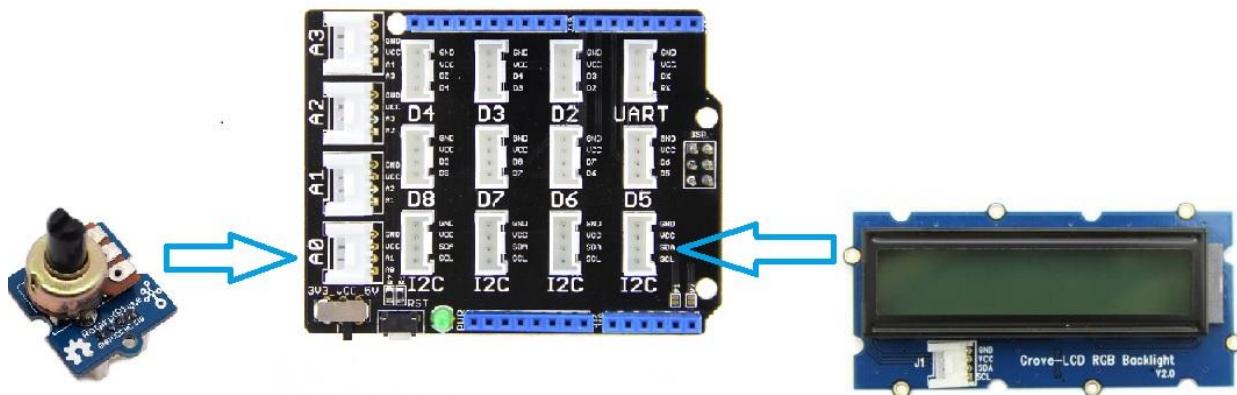


Step 3 – Move Base Shield switch towards 5 Volt



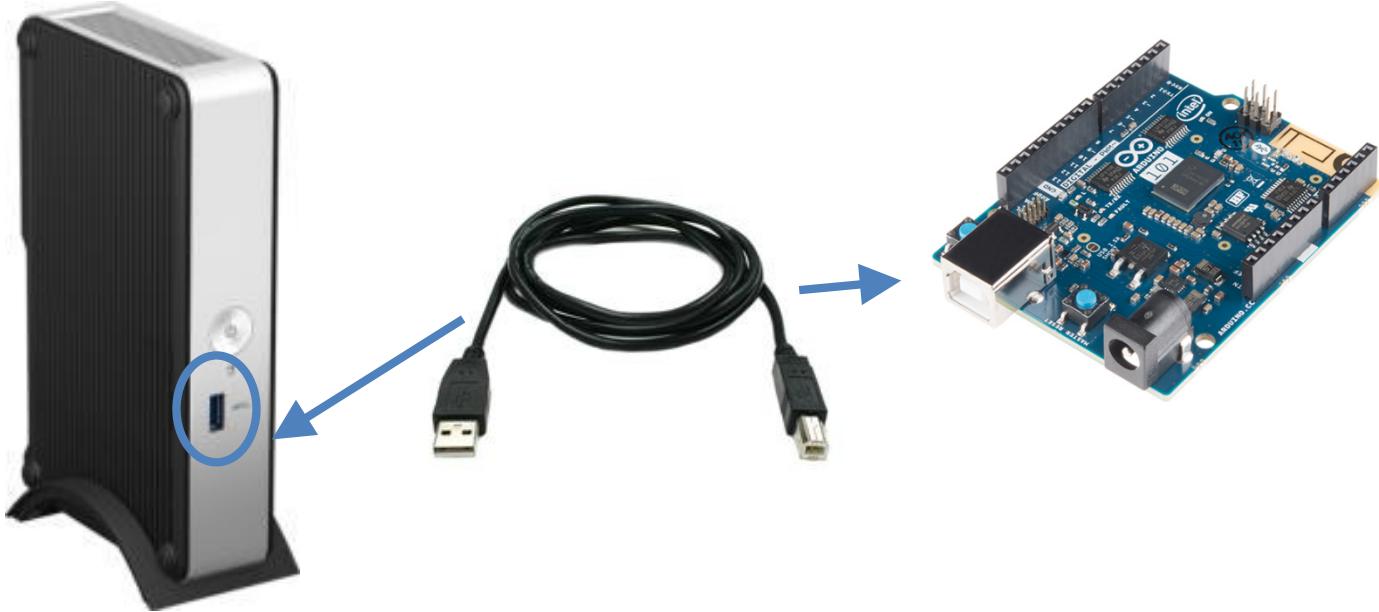
Step 4 – Connect LCD and Rotary Angle to Arduino 101

Connect LCD to any I2C socket and rotary angle to A0 socket with a grove connector cable as shown below, the cables are in the green box.

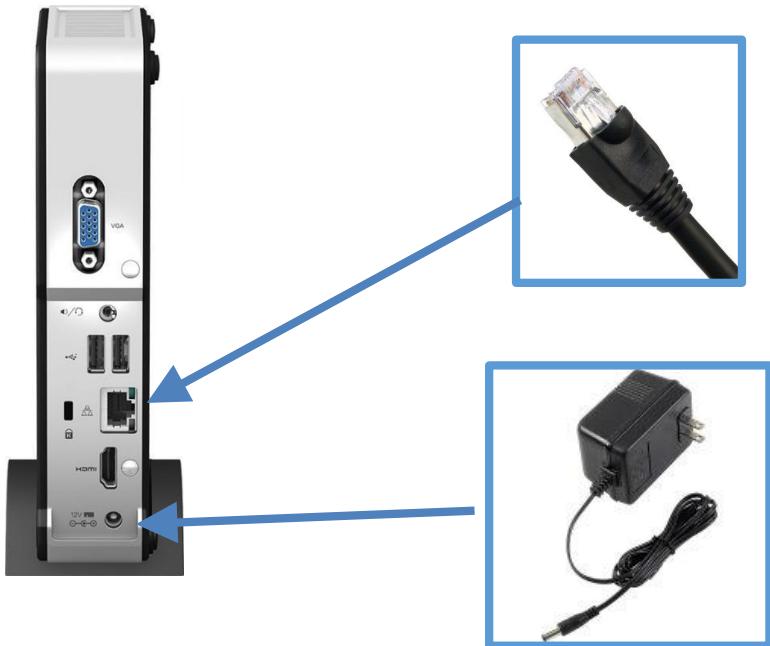


Step 5 – Connect Arduino 101 and Intel IoT Gateway via USB A-B Cable

Now connect Arduino 101 with Intel IoT Gateway via USB A-B cable.



Step 6 – Connect Intel IoT gateway with Ethernet cable and Power adapter



Step 7 – Press Power button to start Gateway

Press power button to start gateway. After around 2 minutes, it will display the IP address of Gateway on the LCD. **Note down the IP address.**



Step 8 – Open your browser and go to IP address to access the IoT Developer Hub

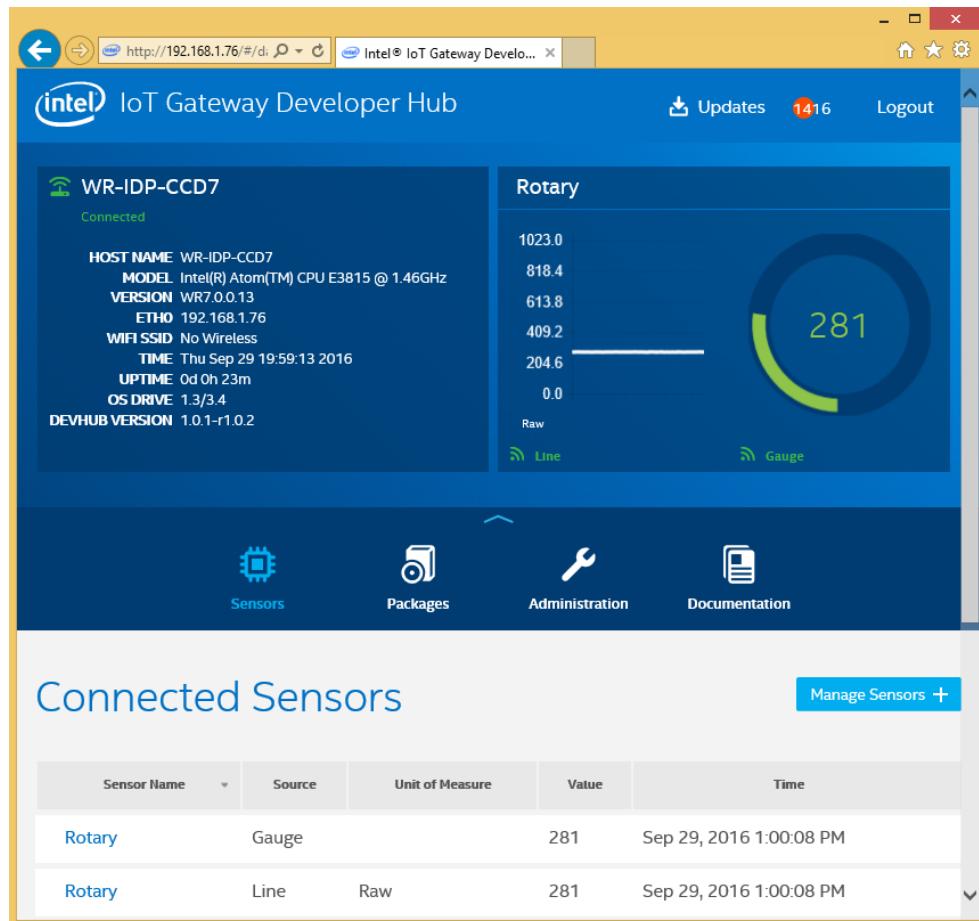
- Login to the Intel® IoT Gateway Developer Hub using root for both the username and the password.
- Use the Intel® IoT Gateway Developer Hub to explore and extend the basic capabilities of the Gateway.

Intel NUC Developer Hub Overview

The Developer Hub is a front end interface for the Gateway. It has 5 main functions:

- Display sensor data and basic Gateway information on a configurable dashboard
- Access the Node-RED development environment
- Add repositories, install and upgrade packages
- Administer the Gateway – update OS, configure network setting
- Access documentation

Dashboard



Dashboard with Rotary Angle Sensor Displayed

By default, the Gateway is programmed to display the value of the rotary angle sensor (knob) on the dashboard. Twist the knob to see the gauge move.

Package Manager

WR-IDP-CCD7 Rotary 402 Raw Line 402 Gauge

Sensors Packages Administration Documentation

PLEASE DON'T UPDATE AT EVENT!

Install Updates 1428 Add Repo + Add Packages +

Package Name	Category	Launch Capability	Update	Running	Auto Run	Activity State
acl	libs					
acpid	base					
alsa-conf	libs/multimedia					

View of Packages screen

On the package manager interface, you will most likely see an option to “Install Updates”.

Do Not Install Updates!

This process can take a significant amount of time and will use up a lot of bandwidth.

Administration

WR-IDP-CCD7
Connected

HOST NAME WR-IDP-CCD7
MODEL Intel(R) Atom(TM) CPU E3815 @ 1.46GHz
VERSION WR7.0.0.13
ETH0 192.168.1.76
WIFI SSID No Wireless
TIME Thu Sep 29 19:59:13 2016
UPTIME 0d 0h 30m
OS DRIVE 1.3/3.4
DEVHUB VERSION 1.0.1-r1.0.2

Rotary

1023.0
818.4
613.8
409.2
204.6
0.0

414

Line Gauge

Sensors Packages Administration Documentation

PLEASE DON'T UPDATE AT EVENT!

Operating System

Intel® IoT Gateway Software Suite
WindRiver® Linux 7, 3.14 Kernel

Restart OS

To restart the system and apply all installed updates, select **Restart OS**.

Save OS Image

Save your OS image to a USB Flash Drive. This will save your image with additional security policies applied. For production deployments, an upgrade to Pro is recommended.

[Upgrade to Pro](#)

Change Password

Select **Change Password** to edit the

Quick Tools

Node-RED APP CLOUD >_

Launch Launch Launch Launch

Connection Details

Your connection on DevHub is currently not secured (<http://>). Enabling security (<https://>) will ensure your activity on DevHub is secure.

Enable

Configure your proxy to manage your

Configure

In the Administration interface you will see options to Update the OS, Upgrade to Pro, as well as others.
Please do not select any of these options!

[Restore to Factory OS](#)

To erase the drive and reinstall the OS from the embedded backup volume, select **Restore to Factory OS**. Be sure to save your work. For a secure restore of the OS, see the [Security Guide](#).

Quick Tools



Node-RED

[Launch](#)



APP CLOUD

[Launch](#)



[Launch](#)



[Launch](#)

Connection Details

Your connection on DevHub is currently not secured (<http://>). Enabling security (<https://>) will ensure your activity on DevHub is secure.

[Enable](#)

Configure your proxy to manage your internet connection default access methods. This will ensure you are properly connected to the internet and able to use all DevHub features.

[Configure](#)

Further down on the page you will see links to some quick tools. Feel free to explore these, however:
Please do not alter any settings in the configuration tool (two gears)!

Documentation

Tutorials

Setup a sensor in Node-Red

How to setup an IoT Sensor using Node-Red

[View Video](#)

[View the tutorial](#)

Connect sensor output to cloud in Node-Red

Building upon 1st tutorial, learn how to push Sensor data into Cloud Node-Red then display the result within Developer Hub charts.

[View Video](#)

[View the tutorial](#)

Learn to use Wind River® Helix™ App Cloud

Create a Hello World app using a cloud development environment to build IoT apps and deploy onto the gateway.

[View Video](#)

[View the tutorial](#)

Save and Deploy OS Image

Learn how to save gateway OS and apps for deployment onto another gateway.

The Documentation tab will give you links to some of the documentation for using the Gateway.

Connect to your Gateway with a terminal client

In order to perform advanced configuration of the Gateway either a monitor and keyboard, or a Secure Shell (SSH) connection is required. On OSX and Linux there are default programs that can do this - Screen and SSH respectively. However, on Windows no default exists, however Putty is light weight and easy to install and can be used to SSH into the Gateway.

For Windows Users:

- Visit the [PuTTY download page](#).
- Under the "For Windows on Intel x86" heading, click on the "putty.exe" link to download the latest release version to your computer.



- Double-click putty.exe on your computer to launch PuTTY.
- Enter IP address of the Gateway
- You can login with the username **root** and the password **root**.

For Mac and Linux Users:

- Open a Terminal
- Type `$ ssh root@<<IP Address>>`. Replace <<IP Address>> with the IP address of your gateway.
- Enter **root** as password

Starting Node-RED for Blinky LED

- Go to Development Hub by entering the IP Address assigned to your Gateway (Eg. <http://192.168.1.1>) in your browser.
- To load the Node-RED interface, click **Administration** on the navigation ribbon. Click **Launch** under the Node-RED icon.

The screenshot shows the Intel IoT Gateway Developer Hub interface. At the top, there is a blue header bar with the Intel logo, the text "IoT Gateway Developer Hub", and a "Logout" button. A red circle highlights the "Administration" button in the navigation ribbon. Below the header, there are several cards: one for "Operating System" (WindRiver Linux 7, 3.14 Kernel) with "Install OS Updates" and "Upgrade to Pro" buttons; another for "Quick Tools" with icons for "Node-RED", "APP CLOUD", "Gears", and "More". The "Node-RED" icon is also highlighted with a red circle and has a "Launch" button below it.

- Node-Red is a visual tool for writing the Internet of Things (IoT).
- Enter username and password as **root** and **root** respectively.



WR-IDP-CCD7

Rotary 412 Raw Line 412 Gauge



Sensors



Packages



Administration



Documentation

Node-RED

Deploy ▾

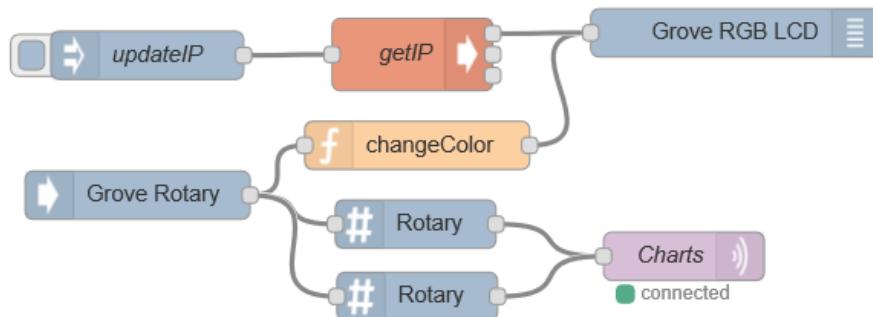


filter nodes

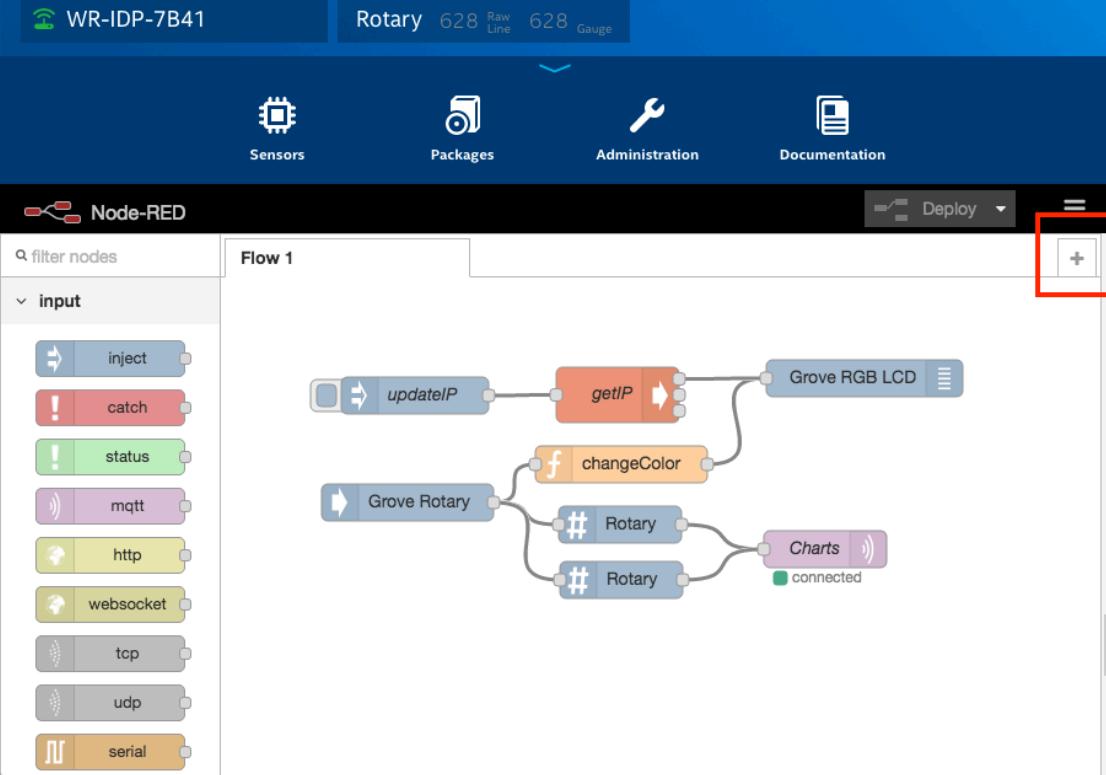
Flow 1

Flow 2

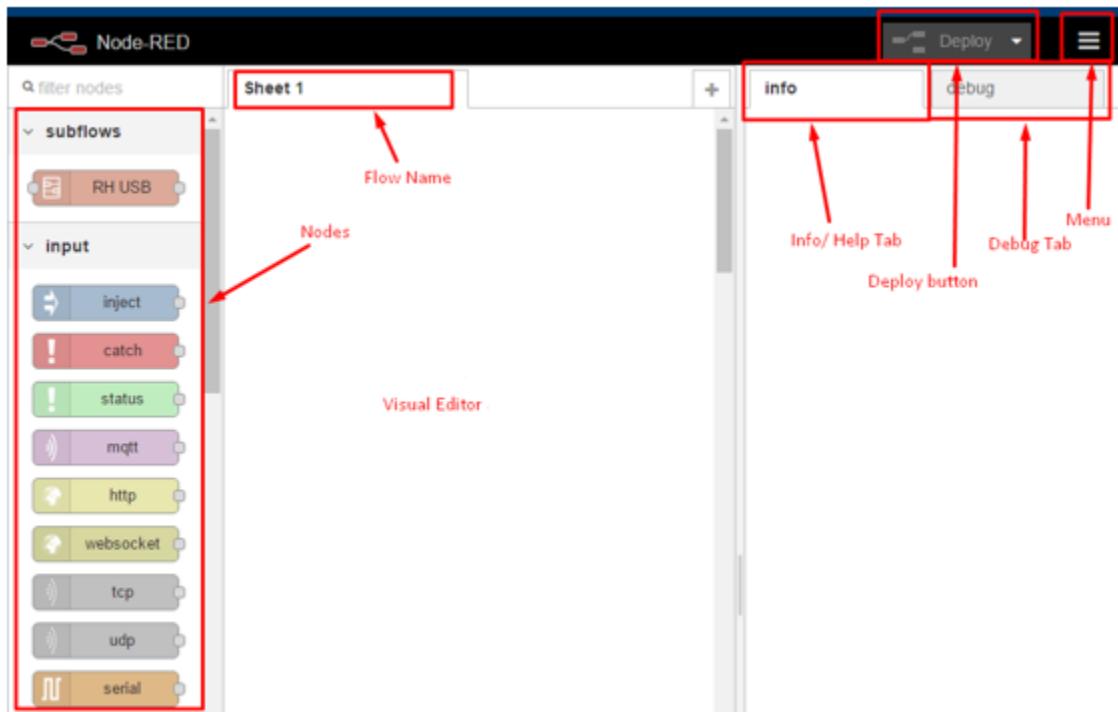
Flow 3



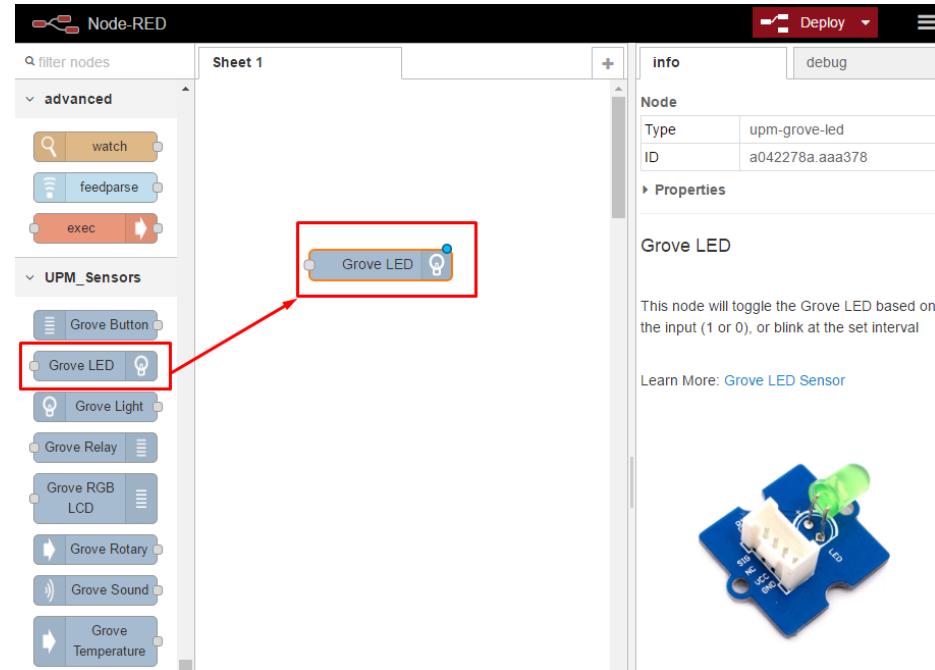
You will see the default application that sends the IP address to the LCD screen and the rotary angle sensor data to the chart on the dashboard. Leave this flow as is for now.



Create a new flow to program the rest of the exercises. Click the + button in the upper right corner of the Node-RED interface.

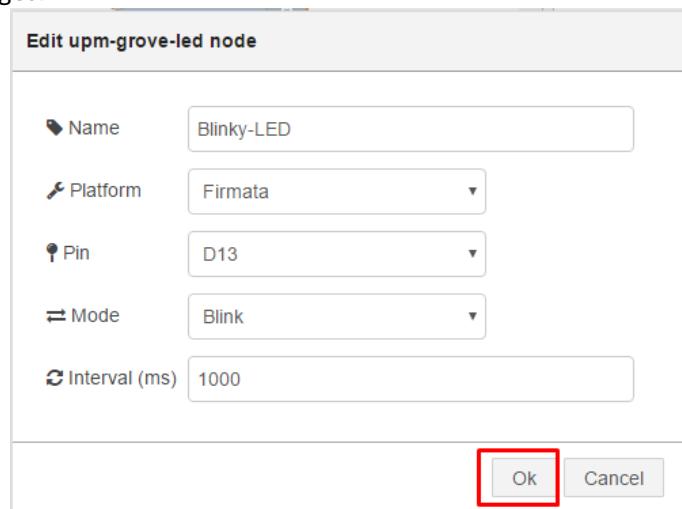


- Deploy Blinky LED Node. Go to Nodes, at the bottom of page there will be group on Nodes called UPM Sensors. Drag Grove LED from Nodes to Visual Editor



- Double Click on Grove LED node on visual Editor. It will open a dialog box to edit properties of node.
 - Name – Blinky LED
 - Platform – Firmata
 - Pin – D13
 - Mode- Blink
 - Interval(ms) – 1000

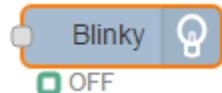
Then Click OK to save changes.



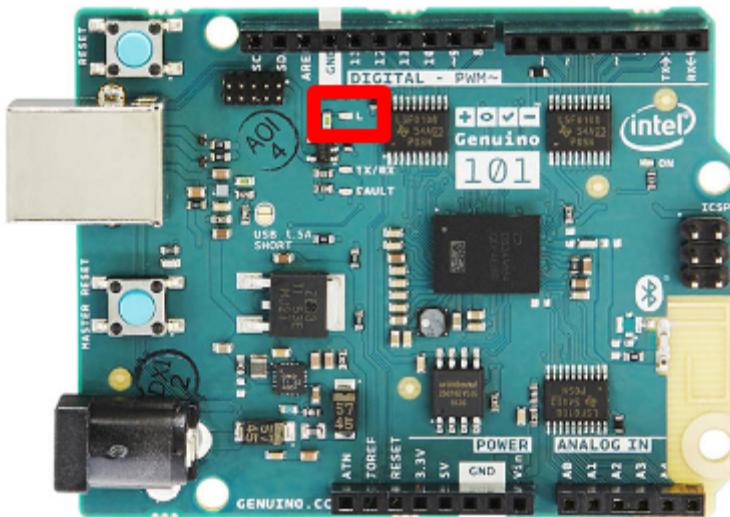
- Click the Deploy button on the top of menu bar to deploy the Node-RED flow.

The screenshot shows the Node-RED interface with a single node named "Blinky-LED" selected. The "Deploy" button at the top right is highlighted with a red box. The node properties show the name is "Blinky-LED", type is "upm-grove-led", and ID is "a042278a.aaa378". The node description states: "This node will toggle the Grove LED based on the input (1 or 0), or blink at the set interval".

- The Arduino 101 LED on board LED will start blinking



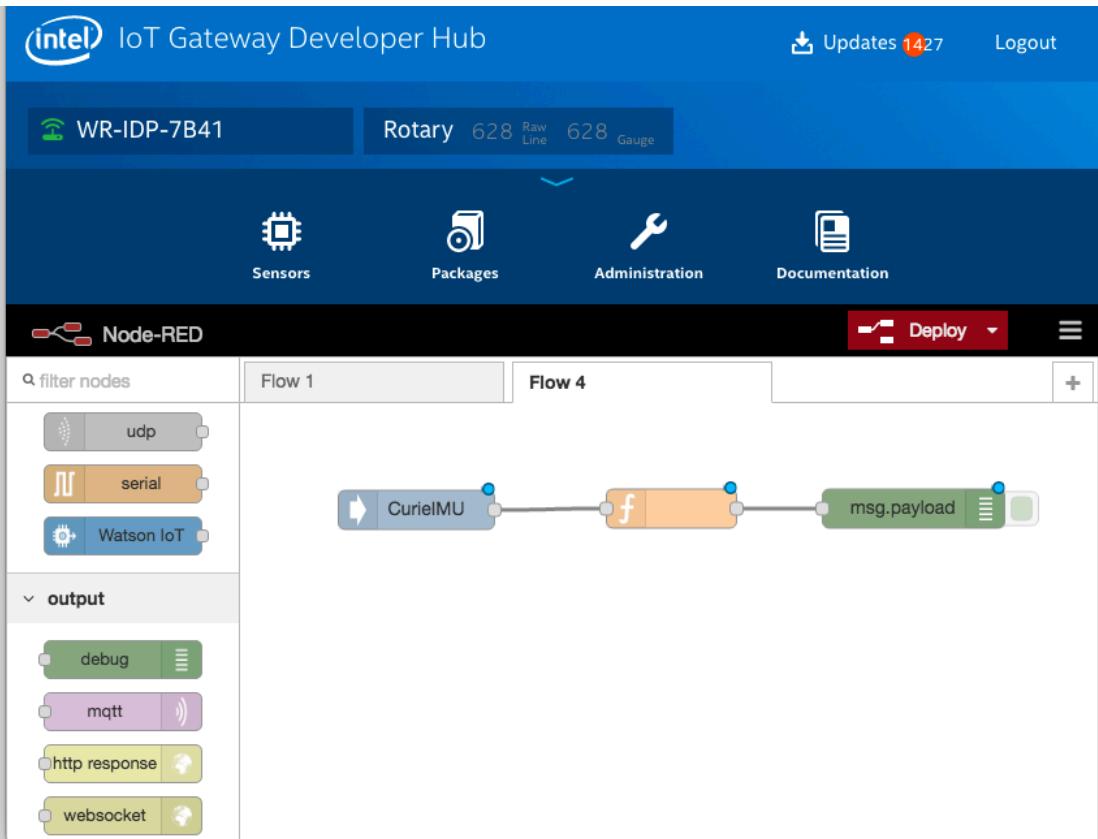
Note: With the Grove Base Shield attached to the Arduino, the LED will be difficult to see. You will need to peek under the shield to see the LED on the Arduino 101 blinking on and off.



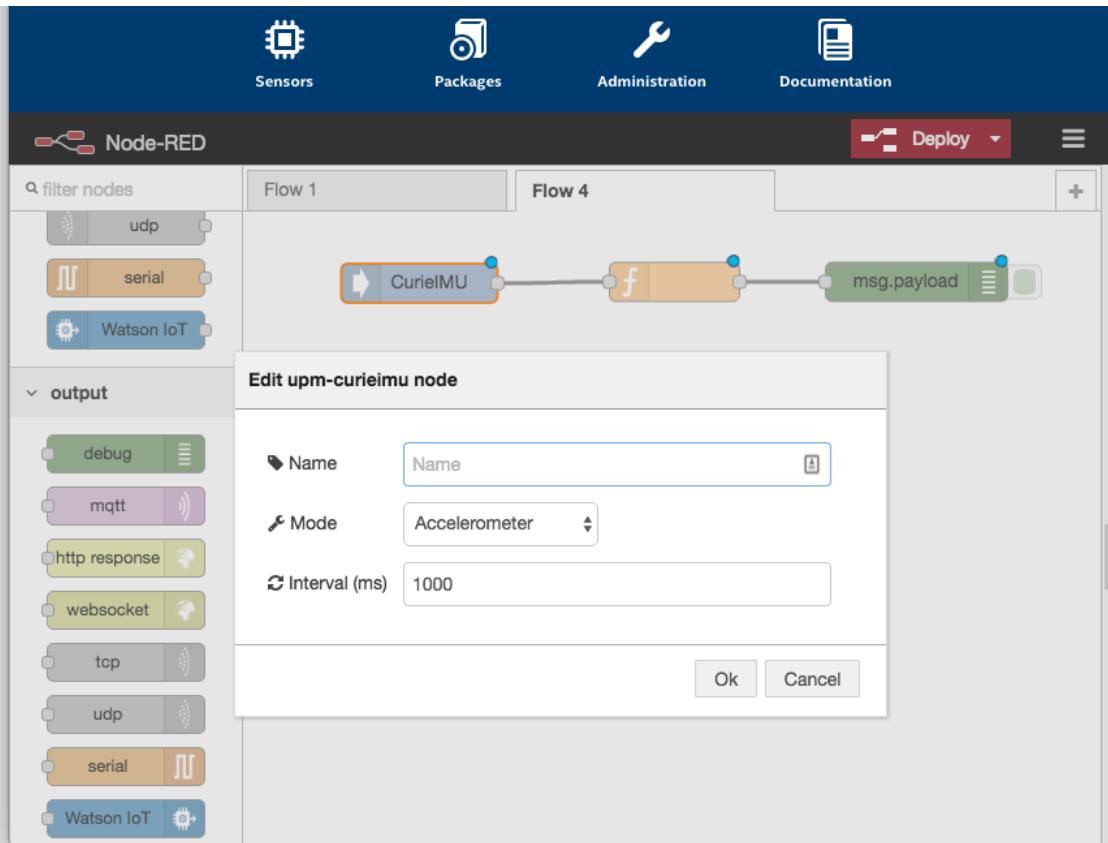
Note: When connecting the Intel IoT Gateway to an Arduino 101, the gateway uses a subplatform called Firmata to communicate GPIO requests to the Arduino 101. Firmata is a generic protocol for communicating with microcontrollers from software on a host computer. It is intended to work with any host computer software package. Basically, this firmware establishes a protocol for talking to the Arduino from the host software.

Adding Sensors and Processing to the Gateway

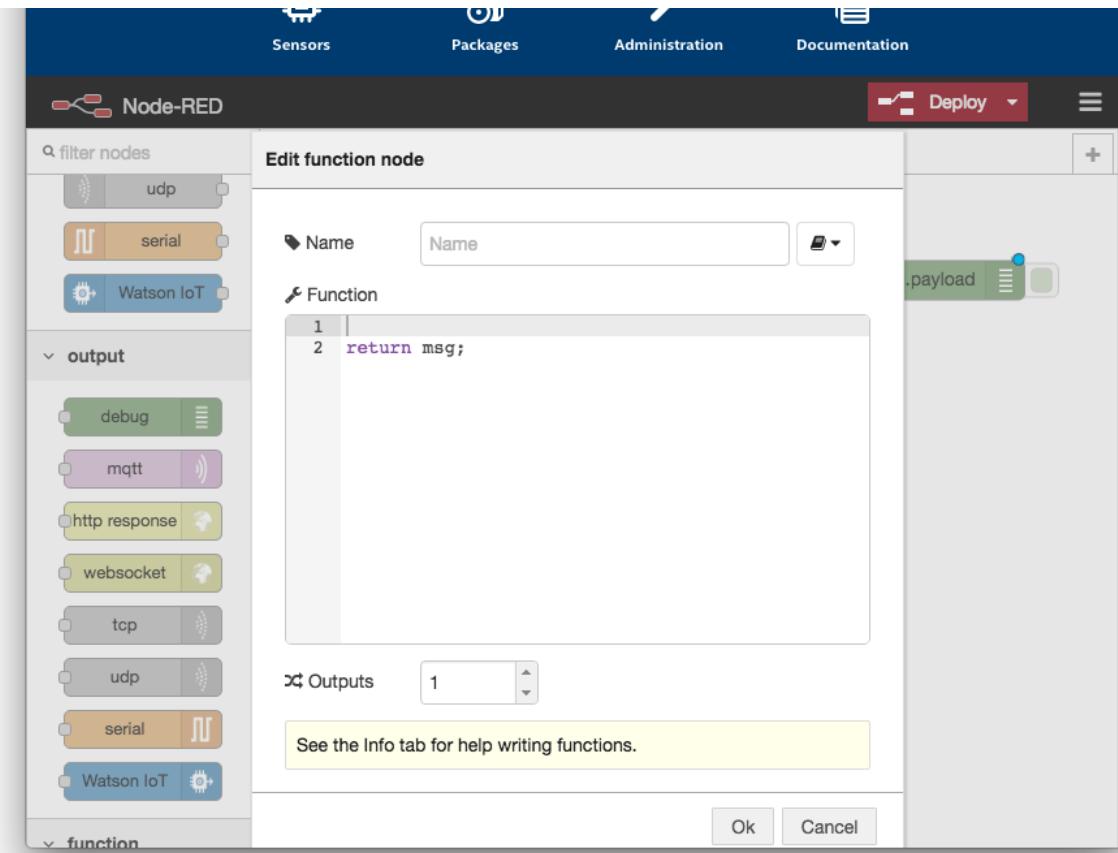
Now that you have a basic understanding of how to interact with the Gateway let's move on to adding some more sensor data and processing to the gateway.



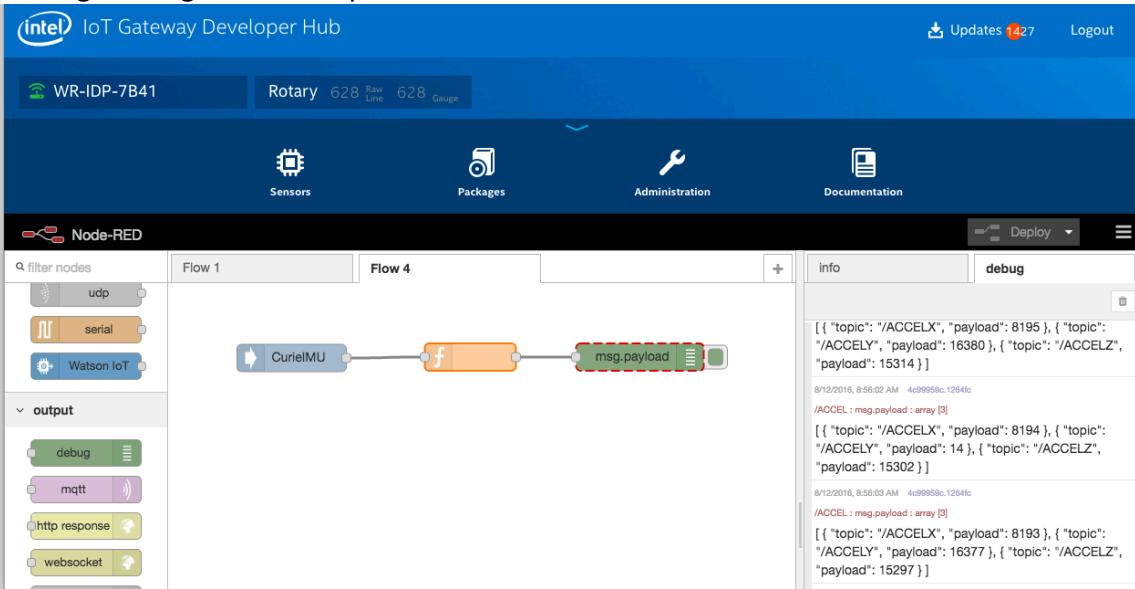
- From **UPM Sensors** add **CurielMU** node
- From **function** add a **function** node
- From **output** add a **debug** node
- Wire the nodes together



If you double click the **CurieIMU** node you can bring up the configuration interface. Here you can name the node, change the mode from Accelerometer to Gyroscope and set the intervals at which the node will send data. Leave these as defaults for now.



If you double click the function node you will bring up the function interface. Here you can enter JavaScript code that will be executed when the node gets input. By default, it only has “return msg;” which will simply pass the incoming message to the output. Let’s leave this for now.



Hit deploy to run the flow. Open the debug tab to see the output from the function node.

Note: you may have to drag from the right to open the debug tab.

As you can see the CurielMU reads out the accelerometer data once a second and passes it to the debug output.

If we are sending data up to a cloud service, we probably want to limit it to significant data. Let's add a filter to the function node so that it only passes the accelerometer data to the debug tab if the values between reading exceed some critical value – i.e. the accelerometer is bumped.

The screenshot shows the Intel IoT Gateway Developer Hub interface. At the top, there are tabs for 'WR-IDP-7B41', 'Rotary 628 Raw Line 628 Gauge', and a notifications badge for 'Updates 1427'. Below the header are navigation links for 'Sensors', 'Packages', 'Administration', and 'Documentation'. The main area is titled 'Node-RED' and shows a 'Flow 1' with a single node: an 'output' node connected to a 'function' node. The 'function' node has the name 'Send above critical delta' and contains the following JavaScript code:

```
i 1 var D_ID = "gateway_01"
i 2 var crit = 5000;
i 3
i 4 var output = {}// Create output object
i 5
i 6 context.array = context.array || [[0,0,0],[0,0,
i 7
i 8
i 9 context.X = msg.payload[0] // read in X axis va
i 10 context.Y = msg.payload[1] // read in Y axis va
i 11 context.Z = msg.payload[2] // read in Z axis va
i 12
i 13
i 14 if (Math.abs(context.array[0] - context.X.payload)
i 15     Math.abs(context.array[1] - context.Y.payload)
i 16     Math.abs(context.array[2] - context.Z.payload) > crit) {
i 17     output = {topic: "/ACCELZ", payload: context.array}
i 18 }
i 19
i 20
i 21
i 22
i 23
i 24
i 25
i 26
i 27
i 28
i 29
i 30
i 31
i 32
i 33
i 34
i 35
i 36
i 37
i 38
i 39
i 40
i 41
i 42
i 43
i 44
i 45
i 46
i 47
i 48
i 49
i 50
i 51
i 52
i 53
i 54
i 55
i 56
i 57
i 58
i 59
i 60
i 61
i 62
i 63
i 64
i 65
i 66
i 67
i 68
i 69
i 70
i 71
i 72
i 73
i 74
i 75
i 76
i 77
i 78
i 79
i 80
i 81
i 82
i 83
i 84
i 85
i 86
i 87
i 88
i 89
i 90
i 91
i 92
i 93
i 94
i 95
i 96
i 97
i 98
i 99
i 100
i 101
i 102
i 103
i 104
i 105
i 106
i 107
i 108
i 109
i 110
i 111
i 112
i 113
i 114
i 115
i 116
i 117
i 118
i 119
i 120
i 121
i 122
i 123
i 124
i 125
i 126
i 127
i 128
i 129
i 130
i 131
i 132
i 133
i 134
i 135
i 136
i 137
i 138
i 139
i 140
i 141
i 142
i 143
i 144
i 145
i 146
i 147
i 148
i 149
i 150
i 151
i 152
i 153
i 154
i 155
i 156
i 157
i 158
i 159
i 160
i 161
i 162
i 163
i 164
i 165
i 166
i 167
i 168
i 169
i 170
i 171
i 172
i 173
i 174
i 175
i 176
i 177
i 178
i 179
i 180
i 181
i 182
i 183
i 184
i 185
i 186
i 187
i 188
i 189
i 190
i 191
i 192
i 193
i 194
i 195
i 196
i 197
i 198
i 199
i 200
i 201
i 202
i 203
i 204
i 205
i 206
i 207
i 208
i 209
i 210
i 211
i 212
i 213
i 214
i 215
i 216
i 217
i 218
i 219
i 220
i 221
i 222
i 223
i 224
i 225
i 226
i 227
i 228
i 229
i 230
i 231
i 232
i 233
i 234
i 235
i 236
i 237
i 238
i 239
i 240
i 241
i 242
i 243
i 244
i 245
i 246
i 247
i 248
i 249
i 250
i 251
i 252
i 253
i 254
i 255
i 256
i 257
i 258
i 259
i 260
i 261
i 262
i 263
i 264
i 265
i 266
i 267
i 268
i 269
i 270
i 271
i 272
i 273
i 274
i 275
i 276
i 277
i 278
i 279
i 280
i 281
i 282
i 283
i 284
i 285
i 286
i 287
i 288
i 289
i 290
i 291
i 292
i 293
i 294
i 295
i 296
i 297
i 298
i 299
i 300
i 301
i 302
i 303
i 304
i 305
i 306
i 307
i 308
i 309
i 310
i 311
i 312
i 313
i 314
i 315
i 316
i 317
i 318
i 319
i 320
i 321
i 322
i 323
i 324
i 325
i 326
i 327
i 328
i 329
i 330
i 331
i 332
i 333
i 334
i 335
i 336
i 337
i 338
i 339
i 340
i 341
i 342
i 343
i 344
i 345
i 346
i 347
i 348
i 349
i 350
i 351
i 352
i 353
i 354
i 355
i 356
i 357
i 358
i 359
i 360
i 361
i 362
i 363
i 364
i 365
i 366
i 367
i 368
i 369
i 370
i 371
i 372
i 373
i 374
i 375
i 376
i 377
i 378
i 379
i 380
i 381
i 382
i 383
i 384
i 385
i 386
i 387
i 388
i 389
i 390
i 391
i 392
i 393
i 394
i 395
i 396
i 397
i 398
i 399
i 400
i 401
i 402
i 403
i 404
i 405
i 406
i 407
i 408
i 409
i 410
i 411
i 412
i 413
i 414
i 415
i 416
i 417
i 418
i 419
i 420
i 421
i 422
i 423
i 424
i 425
i 426
i 427
i 428
i 429
i 430
i 431
i 432
i 433
i 434
i 435
i 436
i 437
i 438
i 439
i 440
i 441
i 442
i 443
i 444
i 445
i 446
i 447
i 448
i 449
i 450
i 451
i 452
i 453
i 454
i 455
i 456
i 457
i 458
i 459
i 460
i 461
i 462
i 463
i 464
i 465
i 466
i 467
i 468
i 469
i 470
i 471
i 472
i 473
i 474
i 475
i 476
i 477
i 478
i 479
i 480
i 481
i 482
i 483
i 484
i 485
i 486
i 487
i 488
i 489
i 490
i 491
i 492
i 493
i 494
i 495
i 496
i 497
i 498
i 499
i 500
i 501
i 502
i 503
i 504
i 505
i 506
i 507
i 508
i 509
i 510
i 511
i 512
i 513
i 514
i 515
i 516
i 517
i 518
i 519
i 520
i 521
i 522
i 523
i 524
i 525
i 526
i 527
i 528
i 529
i 530
i 531
i 532
i 533
i 534
i 535
i 536
i 537
i 538
i 539
i 540
i 541
i 542
i 543
i 544
i 545
i 546
i 547
i 548
i 549
i 550
i 551
i 552
i 553
i 554
i 555
i 556
i 557
i 558
i 559
i 560
i 561
i 562
i 563
i 564
i 565
i 566
i 567
i 568
i 569
i 570
i 571
i 572
i 573
i 574
i 575
i 576
i 577
i 578
i 579
i 580
i 581
i 582
i 583
i 584
i 585
i 586
i 587
i 588
i 589
i 590
i 591
i 592
i 593
i 594
i 595
i 596
i 597
i 598
i 599
i 600
i 601
i 602
i 603
i 604
i 605
i 606
i 607
i 608
i 609
i 610
i 611
i 612
i 613
i 614
i 615
i 616
i 617
i 618
i 619
i 620
i 621
i 622
i 623
i 624
i 625
i 626
i 627
i 628
i 629
i 630
i 631
i 632
i 633
i 634
i 635
i 636
i 637
i 638
i 639
i 640
i 641
i 642
i 643
i 644
i 645
i 646
i 647
i 648
i 649
i 650
i 651
i 652
i 653
i 654
i 655
i 656
i 657
i 658
i 659
i 660
i 661
i 662
i 663
i 664
i 665
i 666
i 667
i 668
i 669
i 670
i 671
i 672
i 673
i 674
i 675
i 676
i 677
i 678
i 679
i 680
i 681
i 682
i 683
i 684
i 685
i 686
i 687
i 688
i 689
i 690
i 691
i 692
i 693
i 694
i 695
i 696
i 697
i 698
i 699
i 700
i 701
i 702
i 703
i 704
i 705
i 706
i 707
i 708
i 709
i 710
i 711
i 712
i 713
i 714
i 715
i 716
i 717
i 718
i 719
i 720
i 721
i 722
i 723
i 724
i 725
i 726
i 727
i 728
i 729
i 730
i 731
i 732
i 733
i 734
i 735
i 736
i 737
i 738
i 739
i 740
i 741
i 742
i 743
i 744
i 745
i 746
i 747
i 748
i 749
i 750
i 751
i 752
i 753
i 754
i 755
i 756
i 757
i 758
i 759
i 760
i 761
i 762
i 763
i 764
i 765
i 766
i 767
i 768
i 769
i 770
i 771
i 772
i 773
i 774
i 775
i 776
i 777
i 778
i 779
i 780
i 781
i 782
i 783
i 784
i 785
i 786
i 787
i 788
i 789
i 790
i 791
i 792
i 793
i 794
i 795
i 796
i 797
i 798
i 799
i 800
i 801
i 802
i 803
i 804
i 805
i 806
i 807
i 808
i 809
i 810
i 811
i 812
i 813
i 814
i 815
i 816
i 817
i 818
i 819
i 820
i 821
i 822
i 823
i 824
i 825
i 826
i 827
i 828
i 829
i 830
i 831
i 832
i 833
i 834
i 835
i 836
i 837
i 838
i 839
i 840
i 841
i 842
i 843
i 844
i 845
i 846
i 847
i 848
i 849
i 850
i 851
i 852
i 853
i 854
i 855
i 856
i 857
i 858
i 859
i 860
i 861
i 862
i 863
i 864
i 865
i 866
i 867
i 868
i 869
i 870
i 871
i 872
i 873
i 874
i 875
i 876
i 877
i 878
i 879
i 880
i 881
i 882
i 883
i 884
i 885
i 886
i 887
i 888
i 889
i 890
i 891
i 892
i 893
i 894
i 895
i 896
i 897
i 898
i 899
i 900
i 901
i 902
i 903
i 904
i 905
i 906
i 907
i 908
i 909
i 910
i 911
i 912
i 913
i 914
i 915
i 916
i 917
i 918
i 919
i 920
i 921
i 922
i 923
i 924
i 925
i 926
i 927
i 928
i 929
i 930
i 931
i 932
i 933
i 934
i 935
i 936
i 937
i 938
i 939
i 940
i 941
i 942
i 943
i 944
i 945
i 946
i 947
i 948
i 949
i 950
i 951
i 952
i 953
i 954
i 955
i 956
i 957
i 958
i 959
i 960
i 961
i 962
i 963
i 964
i 965
i 966
i 967
i 968
i 969
i 970
i 971
i 972
i 973
i 974
i 975
i 976
i 977
i 978
i 979
i 980
i 981
i 982
i 983
i 984
i 985
i 986
i 987
i 988
i 989
i 990
i 991
i 992
i 993
i 994
i 995
i 996
i 997
i 998
i 999
i 1000
i 1001
i 1002
i 1003
i 1004
i 1005
i 1006
i 1007
i 1008
i 1009
i 1010
i 1011
i 1012
i 1013
i 1014
i 1015
i 1016
i 1017
i 1018
i 1019
i 1020
i 1021
i 1022
i 1023
i 1024
i 1025
i 1026
i 1027
i 1028
i 1029
i 1030
i 1031
i 1032
i 1033
i 1034
i 1035
i 1036
i 1037
i 1038
i 1039
i 1040
i 1041
i 1042
i 1043
i 1044
i 1045
i 1046
i 1047
i 1048
i 1049
i 1050
i 1051
i 1052
i 1053
i 1054
i 1055
i 1056
i 1057
i 1058
i 1059
i 1060
i 1061
i 1062
i 1063
i 1064
i 1065
i 1066
i 1067
i 1068
i 1069
i 1070
i 1071
i 1072
i 1073
i 1074
i 1075
i 1076
i 1077
i 1078
i 1079
i 1080
i 1081
i 1082
i 1083
i 1084
i 1085
i 1086
i 1087
i 1088
i 1089
i 1090
i 1091
i 1092
i 1093
i 1094
i 1095
i 1096
i 1097
i 1098
i 1099
i 1100
i 1101
i 1102
i 1103
i 1104
i 1105
i 1106
i 1107
i 1108
i 1109
i 1110
i 1111
i 1112
i 1113
i 1114
i 1115
i 1116
i 1117
i 1118
i 1119
i 1120
i 1121
i 1122
i 1123
i 1124
i 1125
i 1126
i 1127
i 1128
i 1129
i 1130
i 1131
i 1132
i 1133
i 1134
i 1135
i 1136
i 1137
i 1138
i 1139
i 1140
i 1141
i 1142
i 1143
i 1144
i 1145
i 1146
i 1147
i 1148
i 1149
i 1150
i 1151
i 1152
i 1153
i 1154
i 1155
i 1156
i 1157
i 1158
i 1159
i 1160
i 1161
i 1162
i 1163
i 1164
i 1165
i 1166
i 1167
i 1168
i 1169
i 1170
i 1171
i 1172
i 1173
i 1174
i 1175
i 1176
i 1177
i 1178
i 1179
i 1180
i 1181
i 1182
i 1183
i 1184
i 1185
i 1186
i 1187
i 1188
i 1189
i 1190
i 1191
i 1192
i 1193
i 1194
i 1195
i 1196
i 1197
i 1198
i 1199
i 1200
i 1201
i 1202
i 1203
i 1204
i 1205
i 1206
i 1207
i 1208
i 1209
i 1210
i 1211
i 1212
i 1213
i 1214
i 1215
i 1216
i 1217
i 1218
i 1219
i 1220
i 1221
i 1222
i 1223
i 1224
i 1225
i 1226
i 1227
i 1228
i 1229
i 1230
i 1231
i 1232
i 1233
i 1234
i 1235
i 1236
i 1237
i 1238
i 1239
i 1240
i 1241
i 1242
i 1243
i 1244
i 1245
i 1246
i 1247
i 1248
i 1249
i 1250
i 1251
i 1252
i 1253
i 1254
i 1255
i 1256
i 1257
i 1258
i 1259
i 1260
i 1261
i 1262
i 1263
i 1264
i 1265
i 1266
i 1267
i 1268
i 1269
i 1270
i 1271
i 1272
i 1273
i 1274
i 1275
i 1276
i 1277
i 1278
i 1279
i 1280
i 1281
i 1282
i 1283
i 1284
i 1285
i 1286
i 1287
i 1288
i 1289
i 1290
i 1291
i 1292
i 1293
i 1294
i 1295
i 1296
i 1297
i 1298
i 1299
i 1300
i 1301
i 1302
i 1303
i 1304
i 1305
i 1306
i 1307
i 1308
i 1309
i 1310
i 1311
i 1312
i 1313
i 1314
i 1315
i 1316
i 1317
i 1318
i 1319
i 1320
i 1321
i 1322
i 1323
i 1324
i 1325
i 1326
i 1327
i 1328
i 1329
i 1330
i 1331
i 1332
i 1333
i 1334
i 1335
i 1336
i 1337
i 1338
i 1339
i 1340
i 1341
i 1342
i 1343
i 1344
i 1345
i 1346
i 1347
i 1348
i 1349
i 1350
i 1351
i 1352
i 1353
i 1354
i 1355
i 1356
i 1357
i 1358
i 1359
i 1360
i 1361
i 1362
i 1363
i 1364
i 1365
i 1366
i 1367
i 1368
i 1369
i 1370
i 1371
i 1372
i 1373
i 1374
i 1375
i 1376
i 1377
i 1378
i 1379
i 1380
i 1381
i 1382
i 1383
i 1384
i 1385
i 1386
i 1387
i 1388
i 1389
i 1390
i 1391
i 1392
i 1393
i 1394
i 1395
i 1396
i 1397
i 1398
i 1399
i 1400
i 1401
i 1402
i 1403
i 1404
i 1405
i 1406
i 1407
i 1408
i 1409
i 1410
i 1411
i 1412
i 1413
i 1414
i 1415
i 1416
i 1417
i 1418
i 1419
i 1420
i 1421
i 1422
i 1423
i 1424
i 1425
i 1426
i 1427
i 1428
i 1429
i 1430
i 1431
i 1432
i 1433
i 1434
i 1435
i 1436
i 1437
i 1438
i 1439
i 1440
i 1441
i 1442
i 1443
i 1444
i 1445
i 1446
i 1447
i 1448
i 1449
i 1450
i 1451
i 1452
i 1453
i 1454
i 1455
i 1456
i 1457
i 1458
i 1459
i 1460
i 1461
i 1462
i 1463
i 1464
i 1465
i 1466
i 1467
i 1468
i 1469
i 1470
i 1471
i 1472
i 1473
i 1474
i 1475
i 1476
i 1477
i 1478
i 1479
i 1480
i 1481
i 1482
i 1483
i 1484
i 1485
i 1486
i 1487
i 1488
i 1489
i 1490
i 1491
i 1492
i 1493
i 1494
i 1495
i 1496
i 1497
i 1498
i 1499
i 1500
i 1501
i 1502
i 1503
i 1504
i 1505
i 1506
i 1507
i 1508
i 1509
i 15010
i 15011
i 15012
i 15013
i 15014
i 15015
i 15016
i 15017
i 15018
i 15019
i 15020
i 15021
i 15022
i 15023
i 15024
i 15025
i 15026
i 15027
i 15028
i 15029
i 15030
i 15031
i 15032
i 15033
i 15034
i 15035
i 15036
i 15037
i 15038
i 15039
i 15040
i 15041
i 15042
i 15043
i 15044
i 15045
i 15046
i 15047
i 15048
i 15049
i 15050
i 15051
i 15052
i 15053
i 15054
i 15055
i 15056
i 15057
i 15058
i 15059
i 15060
i 15061
i 15062
i 15063
i 15064
i 15065
i 15066
i 15067
i 15068
i 15069
i 15070
i 15071
i 15072
i 15073
i 15074
i 15075
i 15076
i 15077
i 15078
i 15079
i 15080
i 15081
i 15082
i 15083
i 15084
i 15085
i 15086
i 15087
i 15088
i 15089
i 15090
i 15091
i 15092
i 15093
i 15094
i 15095
i 15096
i 15097
i 15098
i 15099
i 150100
i 150101
i 150102
i 150103
i 150104
i 150105
i 150106
i 150107
i 150108
i 150109
i 150110
i 150111
i 150112
i 150113
i 150114
i 150115
i 150116
i 150117
i 150118
i 150119
i 150120
i 150121
i 150122
i 150123
i 150124
i 150125
i 150126
i 150127
i 150128
i 150129
i 150130
i 150131
i 150132
i 150133
i 150134
i 150135
i 150136
i 150137
i 150138
i 150139
i 150140
i 150141
i 150142
i 150143
i 150144
i 150145
i 150146
i 150147
i 150148
i 150149
i 150150
i 150151
i 150152
i 150153
i 150154
i 150155
i 150156
i 150157
i 150158
i 150159
i 150160
i 150161
i 150162
i 150163
i 150164
i 150165
i 150166
i 150167
i 150168
i 150169
i 150170
i 150171
i 150172
i 150173
i 150174
i 150175
i 150176
i 150177
i 150178
i 150179
i 
```

```

var D_ID = "gateway_01"
var crit = 5000;

var output = {}
context.array = context.array || [0,0,0]

context.X = msg.payload[0]
context.Y = msg.payload[1]
context.Z = msg.payload[2]

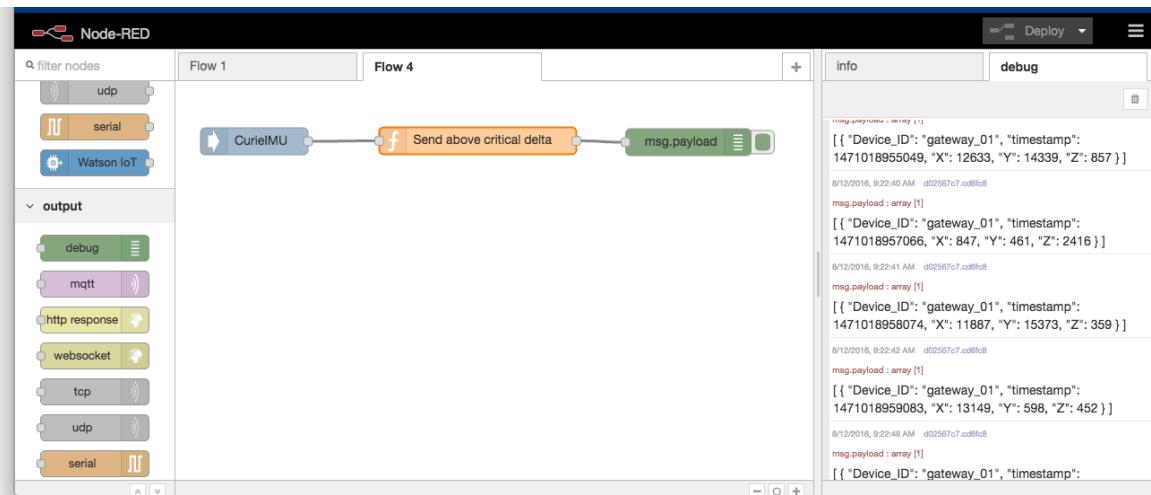
if (Math.abs(context.array[0] - context.X.payload) > crit ||
    Math.abs(context.array[1] - context.Y.payload) > crit ||
    Math.abs(context.array[2] - context.Z.payload) > crit)
{
    output.payload = [
        {
            d: {
                "Device_ID": D_ID,
                "timestamp" : Date.now(),
                "X" : context.X.payload,
                "Y" : context.Y.payload,
                "Z" : context.Z.payload
            }
        }
    ]
    context.array = [context.X.payload, context.Y.payload, context.Z.payload]
    return output;
}
else {
    context.array = [context.X.payload, context.Y.payload, context.Z.payload]
}

```

}

Note: The context object is specific to Node-RED. Any variables inside the node are lost every time the code is run (whenever the node gets a message input). In order to preserve the variables across iterations the context object is used. There are also contexts for the entire flow and even across flows. Think of them as global variables. For more info see: <http://nodered.org/docs/creating-nodes/context>

The code does a few things. First it parses the incoming message into X, Y, and Z accelerometer values. Then it compares the incoming values against the last measured values which are stored in a an array ([0,0,0] at the start). If the delta between values is above the critical value, 5000 in our case, it sends the accelerometer data to the output. The code then overwrites the array with the incoming values to prepare for the next iteration. The code also adds a Device Id and timestamp to the output.



Hit deploy to run the code. The debug output should only display the accelerometer data when the Arduino 101 board registers movement.

Share Node-RED flows across workspaces

You can export and import entire flows (nodes and connections) out of and into your workspace.

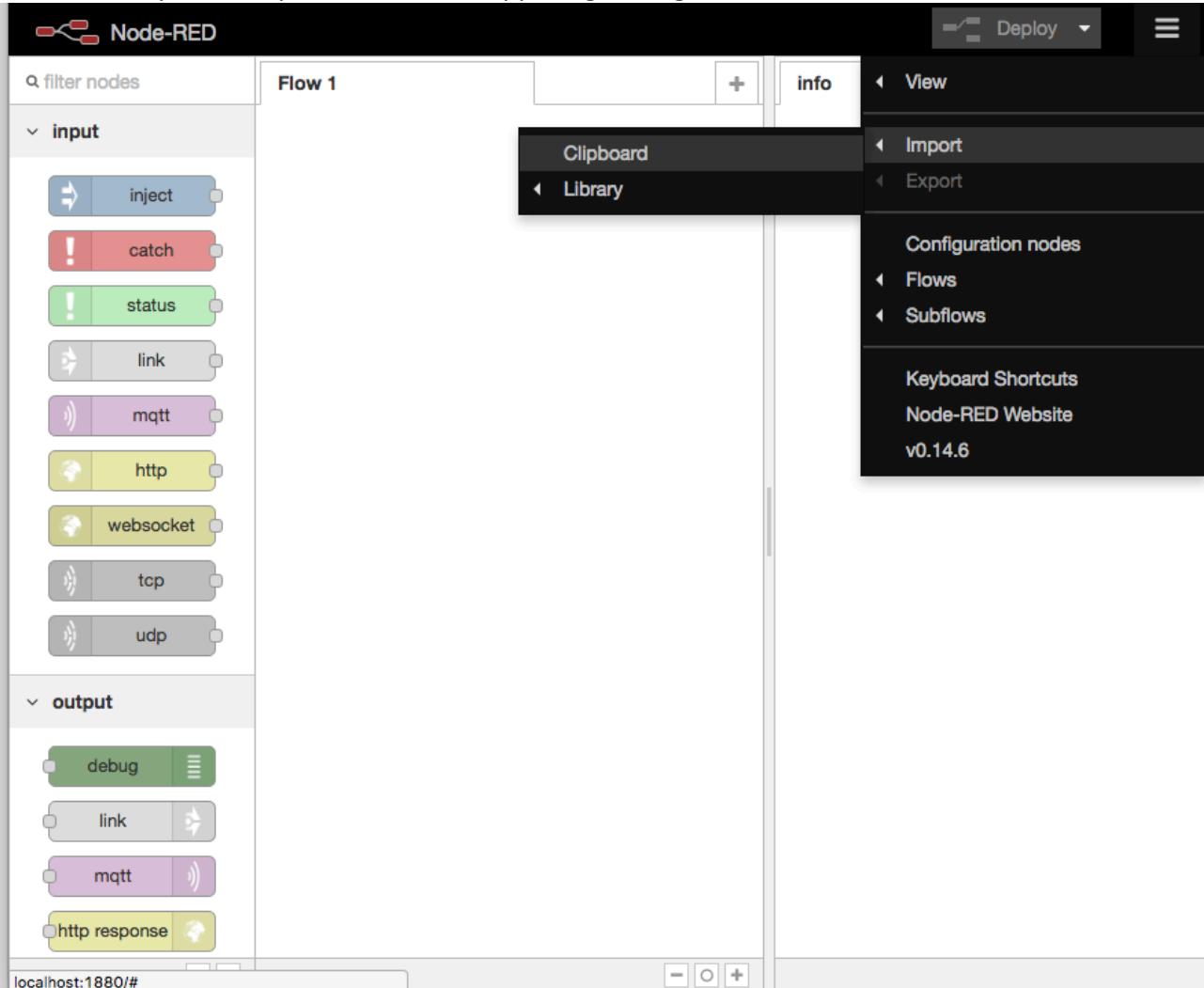
Importing Flows:

To import the above flow go to:

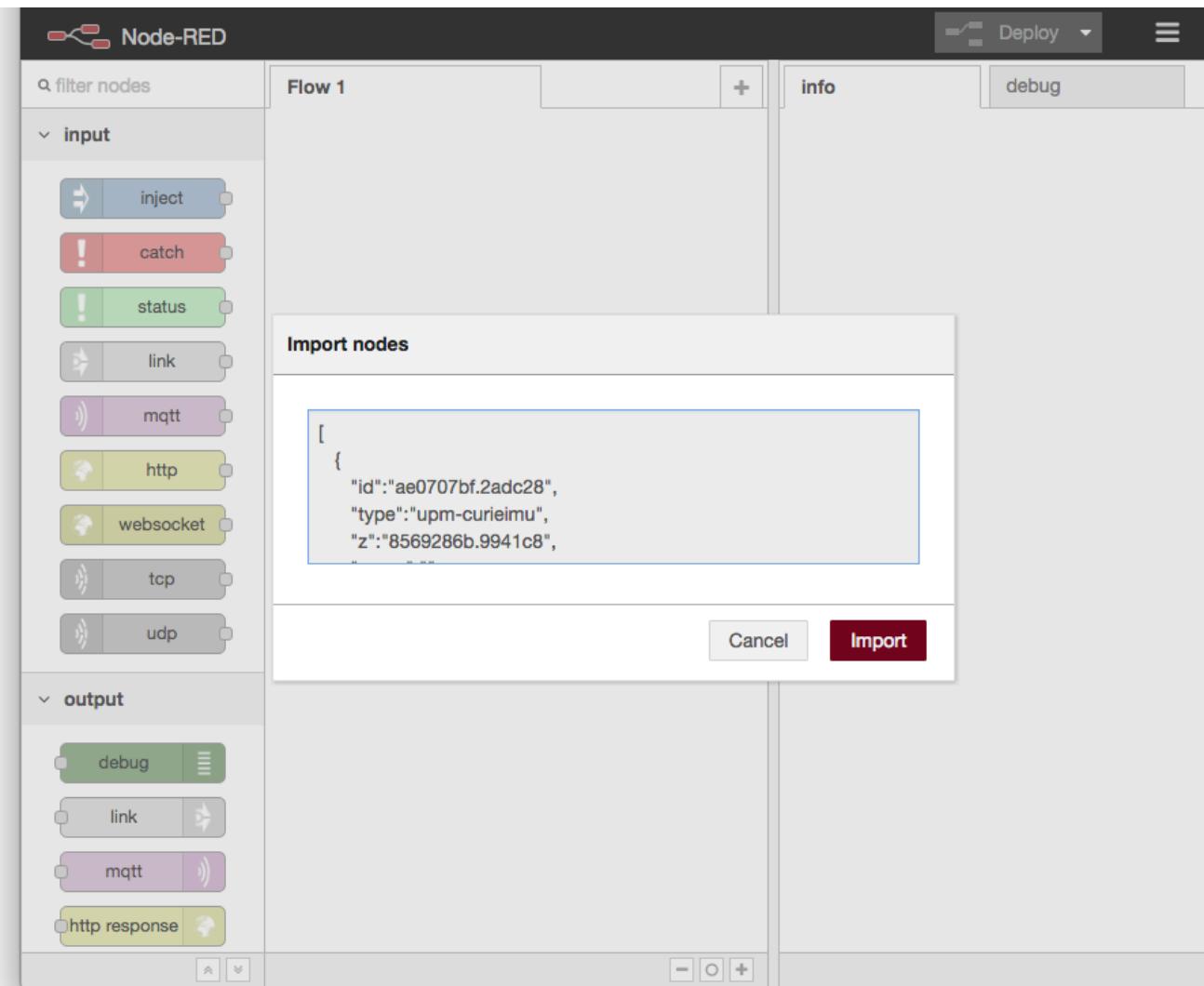
<https://github.com/martinkronberg/Intel-AWS-IoT-Workshop/blob/master/read-accel.json>

Copy the JSON code to your clipboard (ctrl-c)

Navigate to the Import -> Clipboard from the upper right burger:



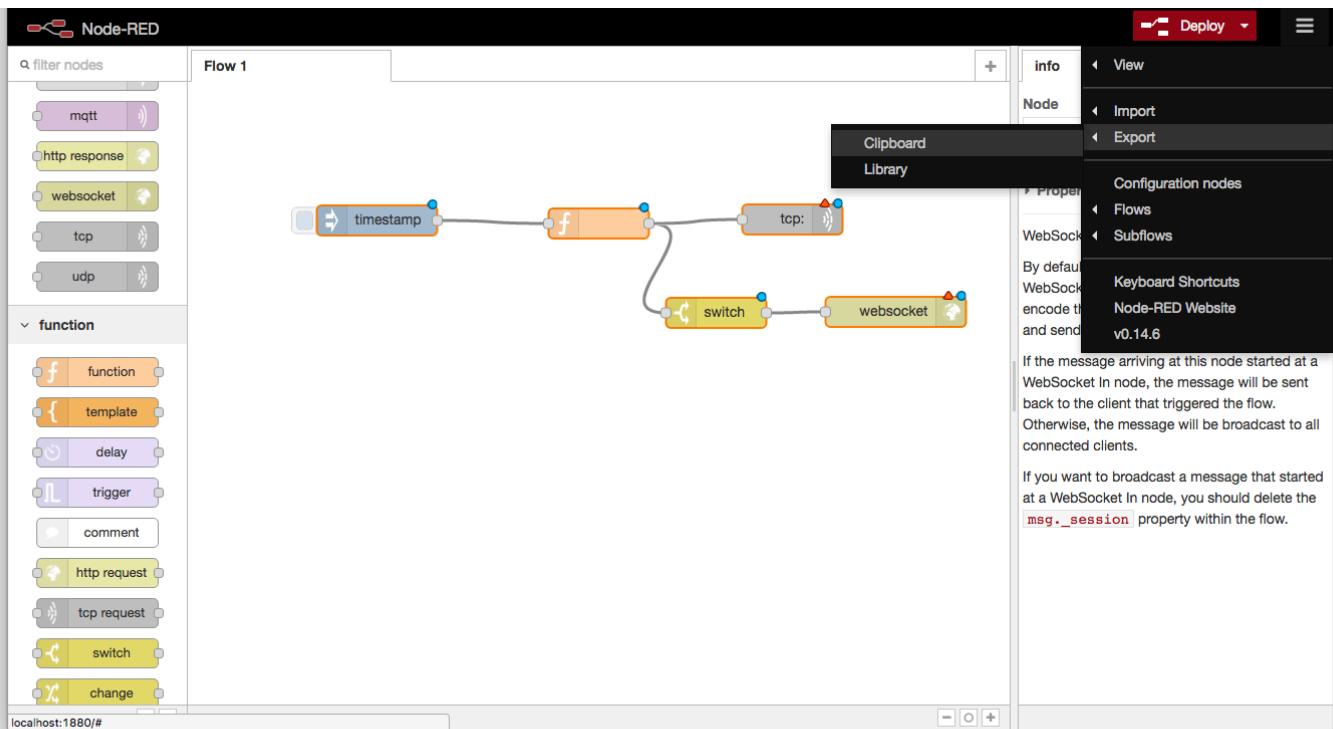
Paste the JSON from your clipboard (ctrl-v) and clip import:



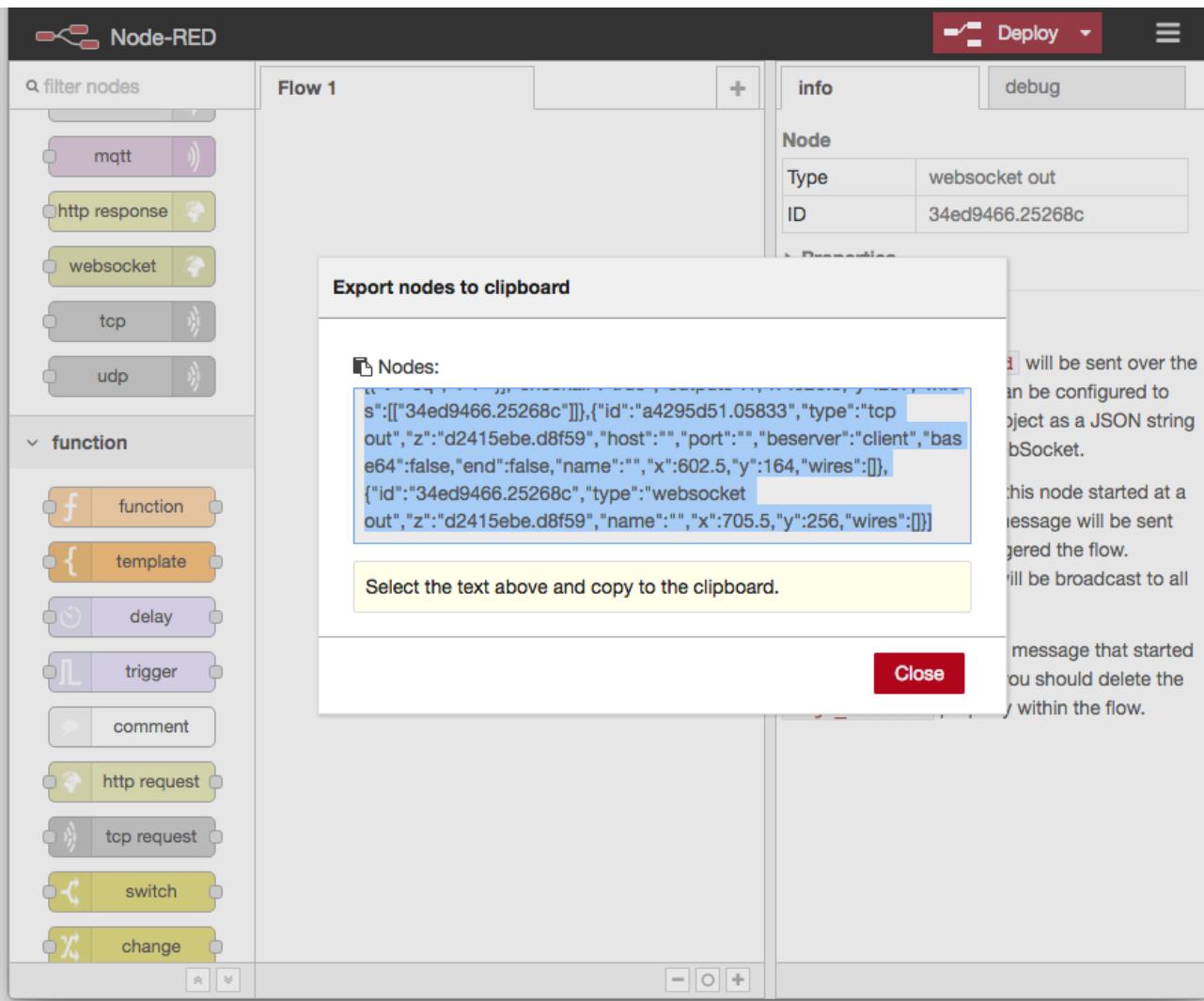
You will now have the entire flow inside your workspace. Follow the same procedure for importing anymore code you come across in this workshop.

Exporting Flows

In order to export a flow select the nodes you wish to export, navigate to Export -> Clipboard from the burger



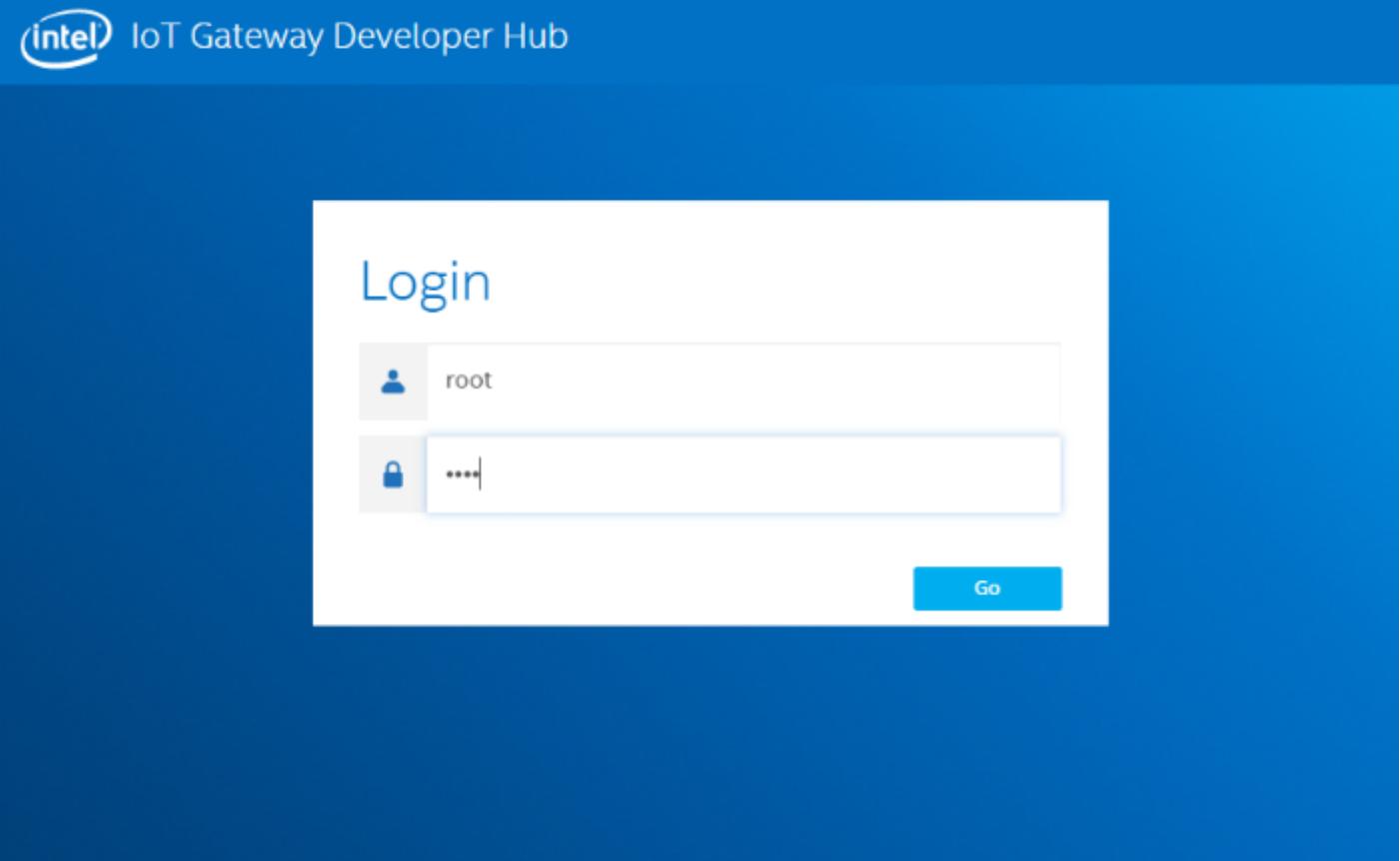
Copy the text in the window that opens. You can now paste this somewhere and share it with others. I would recommend running it through a json beautifier like: <https://jsonformatter.curiousconcept.com/> first if you want it to be readable.



Add AWS IoT Nodes and OS Monitoring

1. Access the console on your gateway using either a monitor or keyboard connected directly, or SSH (recommended).
2. Add the GPG key for the cloud repository using the following command:

```
rpm --import http://iotdk.intel.com/misc/iot_pub.key
```
3. On your development device (e.g., laptop) open a web browser and load the IoT Gateway Developer Hub interface by entering the IP address of your gateway in the address bar.
Tip: You can find your gateway's IP address using the ifconfig command.
4. Login to the IoT Gateway Developer Hub interface using your credentials. The default login and password are both **root**.



5. Add the **IoT Cloud** repository.

The screenshot shows the Intel IoT Gateway Developer Hub dashboard. At the top, there's a header with the Intel logo, 'Updates 1428', and a 'Logout' link. Below the header, there are four main navigation icons: 'Sensors' (radio tower icon), 'Packages' (blue square icon with a white gear), 'Administration' (wrench icon), and 'Documentation' (book icon). The 'Packages' icon is circled in orange. The main content area is titled 'Installed Packages'. It features a table with columns: Package Name, Category, Launch Capability, Update, Running, Auto Run, and Activity State. Two packages are listed: 'acl' (Category: libs) and 'acpid' (Category: base). Above the table, there are three buttons: 'Install Updates' (with a download icon and '1428' notifications), 'Add Repo +' (which is also circled in orange), and 'Add Packages +'.

Package Name	Category	Launch Capability	Update	Running	Auto Run	Activity State
acl	libs					
acpid	base					

6. Go to the **Packages** section and click the **Add Repo +** button.

The screenshot shows the 'Manage Repositories' interface. On the left, under 'Existing Repositories', there is a list with one item: 'Intel_Repository'. To its right is a blue 'Delete' button. On the right side of the screen, there is a form for 'Add New Repository'. This form includes fields for 'Name' (containing 'IoT_cloud') and 'URL' (containing 'http://iotdk.intel.com/repos/iot-cloud/wrlir'). Below these fields is an 'Authentication' section with 'Username' and 'Password' fields. At the bottom of the form is a blue 'Add Repository' button. A large orange rectangle highlights the 'Add New Repository' form. A smaller orange rectangle highlights the 'Update Repositories' button at the bottom left. A green message at the bottom states: 'The repository IoT_cloud_connector was successfully deleted.'

7. Populate the fields with the following information and click **Add Repository**:

Name: IoT_Cloud

URL: <http://iotdk.intel.com/repos/iot-cloud/wrlinux7/rcpl13>

8. Finally, click the **Update Repositories** button to update the package list.

Adding AWS* support to your Intel® IoT Gateway

The screenshot shows the Intel IoT Gateway Developer Hub interface. At the top, there's a header bar with the Intel logo, the title "IoT Gateway Developer Hub", a notification for "Updates 1428", and a "Logout" button. Below the header, a navigation bar includes links for "WR-IDP-CCD7", "Rotary 403 Raw Line 403 Gauge", "Sensors", "Packages" (which is highlighted with an orange box), "Administration", and "Documentation". The main content area is titled "Installed Packages". It features a table with columns: Package Name, Category, Launch Capability, Update, Running, Auto Run, and Activity State. Two packages are listed: "acl" (Category: libs) and "acpid" (Category: base). Above the table are buttons for "Install Updates" (with a count of 1428), "Add Repo +", and "Add Packages +", with the "Add Packages +" button also highlighted with an orange box.

1. Click the **Add Packages +** button to bring up the list of packages you can install.

The screenshot shows a modal dialog titled "Add New Packages". It contains a search bar with the text "cloud-aws" and a message instructing users to "Select Install to load new packages and applications to your Packages page.". Below the search bar is a table with columns: Package Name, Category, and Vertical. One entry is visible: "packagegroup-cloud-aws" (Category: base). To the right of this entry is a blue "Install" button, which is highlighted with an orange box.

Search for **cloud-aws** using the search box at the top of the package window. Click the **Install** button next to the **packagegroup-cloud-aws** entry.

2. Back in your ssh connection, run the following command to restart the Node-RED environment on the NUC. This is necessary because the package that we just installed updated the resources available to Node-RED so it needs to be re-initialized:

```
$systemctl restart node-red-experience
```

What Next?

To explore some advanced projects that you can build with the gateway make sure you have the Intel XDK IoT Edition up and running and check out:

Smart Home Path to Product:

<https://software.intel.com/en-us/articles/iot-path-to-product-how-to-build-the-smart-home-prototype>

Smart Vending Machine Path to Product:

<https://software.intel.com/en-us/articles/iot-path-to-product-how-to-build-an-intelligent-vending-machine>

Connected Transportation Path to Product:

<https://software.intel.com/en-us/articles/iot-path-to-product-how-to-build-a-connected-transportation-solution>

A collection of How-To IoT Projects:

<https://software.intel.com/en-us/blogs/2015/11/04/announcing-18-new-how-to-intel-iot-code-samples>