

# WISE-DeviceOn

## IoT Device Operation Management

User Manual  
English v-1.0.14



# Revision History

Date	Version	Author	Reviewer	Description
2019-12-16	1.0.5	Sephiroth.Wang		First version released Section 1, 2, 3, 4.4, 6, 7
2019-12-23	1.0.5	Alex.Shao	Sephiroth.Wang	Section 4.1, 4.2, 4.3, 4.5
2019-12-27	1.0.5	Terry.Lu	Sephiroth.Wang	Section 5.1
2020-01-08	1.0.5	Iris.Pan	Sephiroth.Wang	Section 5.2
2020-01-10	1.0.5	Wunhuei.Liou	Sephiroth.Wang	Section 5.3
2020-02-03	1.0.6	Sephiroth.Wang		Section 6.7, 6.8, 6.9
2020-02-10	1.0.6	MingWei.Tasi	Sephiroth.Wang	Section 4.6
2020-02-12	1.0.6	Sephiroth.Wang		Section 6.10
2020-03-11	1.0.6	Scott.Chang	Sephiroth.Wang	Section 6.3
2020-03-25	1.0.7	Sephiroth.Wang		Add Section 6.8, and move 6.8-> 6.9, 6.9->6.10 Add Section 7.2
2020-04-01	1.0.8	Sephiroth.Wang		Change wallpaper on first page Add new log mechanism on Section 6.6 Add Section 4.7 for device provisioning.
2020-04-06	1.0.9	Alex.Shao	Sephiroth.Wang	Section 2.1.4
2020-04-10	1.0.10	Daniel.Hung	Sephiroth.Wang	Add plugin development on RISC (Section 5.1.4)
2020-04-21	1.0.11	Listzt.Kao	Sephiroth.Wang	Add plugin development on Android (Section 5.1.5) Modify WISE-DeviceOn -> WISE- DeviceOn
2020-09-02	1.0.12	Sephiroth.Wang		Review all document and update new user interface for v-4.2.3
2020-09-09	1.0.13	Sephiroth.Wang		Add Telegram, Teams, Slack Notification Service (4.3.5 ~ 4.3.7)
2020-09-22	1.0.14	Sephiroth.Wang		Add Linux Agent Onboarding and Re-adjust catalog.

# Table of Contents

<b>Revision History .....</b>	<b>1</b>
<b>Table of Contents .....</b>	<b>2</b>
<b>1. Introduction.....</b>	<b>6</b>
1.1 <b>Feature Highlights .....</b>	<b>8</b>
1.2 <b>DeviceOn Server Versions .....</b>	<b>12</b>
1.2.1      Standalone, VM (Cloud).....	12
1.2.2      Azure PaaS, Kubernetes (Cloud) .....	14
1.2.3      Data Service Server for Private Cloud.....	14
1.2.4      WISE-PaaS/EnSaaS (Cloud) .....	15
1.3 <b>DeviceOn Agent Versions .....</b>	<b>16</b>
1.3.1      WISE-Agent (Client).....	17
1.4 <b>Security .....</b>	<b>17</b>
1.4.1      Role-Based Access Control (RBAC) .....	17
1.4.2      SSL Encryption .....	17
1.4.3      Security Scan.....	18
<b>2. Getting Started.....</b>	<b>20</b>
2.1 <b>DeviceOn Cloud Installation.....</b>	<b>20</b>
2.1.1      Setup Standalone Version (On-premise).....	20
2.1.2      Setup Standalone Version for Ubuntu Linux (On-premise).....	28
2.1.3      Redeem AKS Version from WISE-PaaS Marketplace .....	30
2.2 <b>DeviceOn Client Installation.....</b>	<b>34</b>
2.2.1      Setup Device Onboarding (Windows) .....	34
2.2.2      Setup Device Onboarding (Linux) .....	44
<b>3. DeviceOn User Interface &amp; Functions .....</b>	<b>46</b>
3.1 <b>DeviceOn Server (Standalone).....</b>	<b>46</b>

3.1.1	Standalone Server Control .....	46
3.1.2	Background Watchdog Service .....	50
<b>3.2</b>	<b>DeviceOn WISE-Agent .....</b>	<b>50</b>
3.2.1	WISE-Agent Connection .....	50
3.2.2	WISE-Agent Services.....	55
<b>3.3</b>	<b>DeviceOn User Interface .....</b>	<b>55</b>
3.3.1	DeviceOn Overview .....	59
3.3.2	Device Management .....	64
3.3.3	Account Management .....	94
3.3.4	Event Logs.....	96
3.3.5	OTA (Remote Provisioning) .....	98
3.3.6	System Configuration.....	111
3.3.7	Dashboard .....	121
<b>4.</b>	<b>Hands-On LABs .....</b>	<b>124</b>
4.1	<b>How to Create a Real-time Action into Overview .....</b>	<b>125</b>
4.1.1	Prerequisite.....	125
4.1.2	Step-by-Step.....	125
4.2	<b>How to Remote Software Provisioning via OTA .....</b>	<b>128</b>
4.2.1	Prerequisite.....	128
4.2.2	Step-by-Step.....	129
4.3	<b>How to Set a Device Threshold and Event Notify Services .....</b>	<b>134</b>
4.3.1	Prerequisite.....	134
4.3.2	Steps to Set Event Notification Service – Email .....	134
4.3.3	Steps to Set Event Notification Service – LINE .....	136
4.3.4	Steps to Set Event Notification Service – WeChat .....	139
4.3.5	Steps to Set Event Notification Service – Telegram .....	143
4.3.6	Steps to Set Event Notification Service – Microsoft Teams .....	146

4.3.7	Steps to Set Event Notification Service – Slack .....	149
4.3.8	Steps to Set Thresholds to a Device .....	152
<b>4.4</b>	<b>How to Visualize Device Data via Grafana Dashboard .....</b>	<b>155</b>
4.4.1	Prerequisite.....	155
4.4.2	Step-by-Step.....	155
<b>4.5</b>	<b>How to Enable/Disable Windows Lockdown Features .....</b>	<b>159</b>
4.5.1	Prerequisite.....	159
4.5.2	Step-by-Step.....	159
<b>4.6</b>	<b>How to Manage DeviceOn on AKS .....</b>	<b>161</b>
4.6.1	Prerequisite.....	161
4.6.2	Steps to Upgrade DeviceOn.....	161
4.6.3	Step to Monitor Container Healthy and Status .....	162
4.6.4	Steps to Expose Database/RabbitMQ to Access.....	165
4.6.5	Steps to Deploy DeviceOn to AKS by Manual .....	167
<b>4.7</b>	<b>How to Batch Provision to Your Devices.....</b>	<b>170</b>
4.7.1	Prerequisite.....	171
4.7.2	Steps to Local Provisioning .....	171
4.7.1	Troubleshooting .....	175
<b>5.</b>	<b>DeviceOn Development Guide .....</b>	<b>176</b>
<b>5.1</b>	<b>WISE-Agent Plugin Development .....</b>	<b>176</b>
5.1.1	WISE-Agent Architecture .....	177
5.1.2	Prerequisite.....	178
5.1.3	Develop a Plugin on Windows Environment.....	178
5.1.4	Develop a Plugin on Linux Environment.....	183
5.1.5	Develop a Plugin on Android Environment .....	186
<b>5.2</b>	<b>DeviceOn UI Plugin Development.....</b>	<b>187</b>
5.2.1	Prerequisite.....	187

5.2.2	Environment Setup .....	187
5.2.3	Develop a Sample Add-in .....	188
5.2.4	Develop an Add-in to Access DeviceOn API .....	192
<b>6.</b>	<b>FAQ .....</b>	<b>195</b>
6.1	Why Some of Devices Cannot Power On.....	195
6.2	Why Cannot Remote Control via KVM (Remote Desktop) .....	199
6.3	Why Cannot Screenshot and Always Show Device “No Login” .....	200
6.4	Why the WISE-Agent Cannot Install? With error code 12007?.....	205
6.5	Why the WISE-Agent Cannot Download from Device Onboarding?.....	206
6.6	Why the Acronis and McAfee failed to install? .....	207
6.7	Why Your SMTP Server Cannot Send a Mail?.....	208
6.8	How the Device Data Flow and Debug from Edge to Cloud .....	208
6.9	How to Enable/Disable plugins on WISE-Agent.....	209
6.10	How to Enable and Adjust WISE-Agent Log Levels .....	210
6.11	How to Change DeviceOn Server Address (Standalone).....	213
6.12	How to Migrate/Transfer EdgeSense Database to DeviceOn (WISE-PaaS/EnSaaS) .....	214
6.13	How Does DeviceOn Interact with AI and Machine Learning .....	219
6.14	How to Enable Data Retention on DeviceOn .....	220
6.15	How to Enable HTTPS on DeviceOn Web Service .....	221
<b>7.</b>	<b>Reference .....</b>	<b>227</b>
7.1	User Permission .....	227
7.2	Retrieve My Azure Account Information .....	231
7.2.1	Method 1 – Create & Get Information on Azure Portal .....	231
7.2.2	Method 2 – Create via Azure CLI (Command-line Tool) .....	236

## 1. Introduction

A surge in market demand for Industrial IoT products has rapidly increased the number of connected devices that are currently deployed and managed across different locations. It is essential to effectively manage, monitor, and control thousands of connected devices while ensuring uninterrupted service. Devices must work properly and securely after they have been deployed - without requiring frequent visits from service technicians. Customers require secure access to their devices in order to detect, troubleshoot, and undertake time-critical actions.



With Advantech's WISE-DeviceOn, users can swiftly utilize onboard devices, efficiently monitor device health status, and securely send software and firmware updates over-the-air (OTA) on-site and remotely at scale.

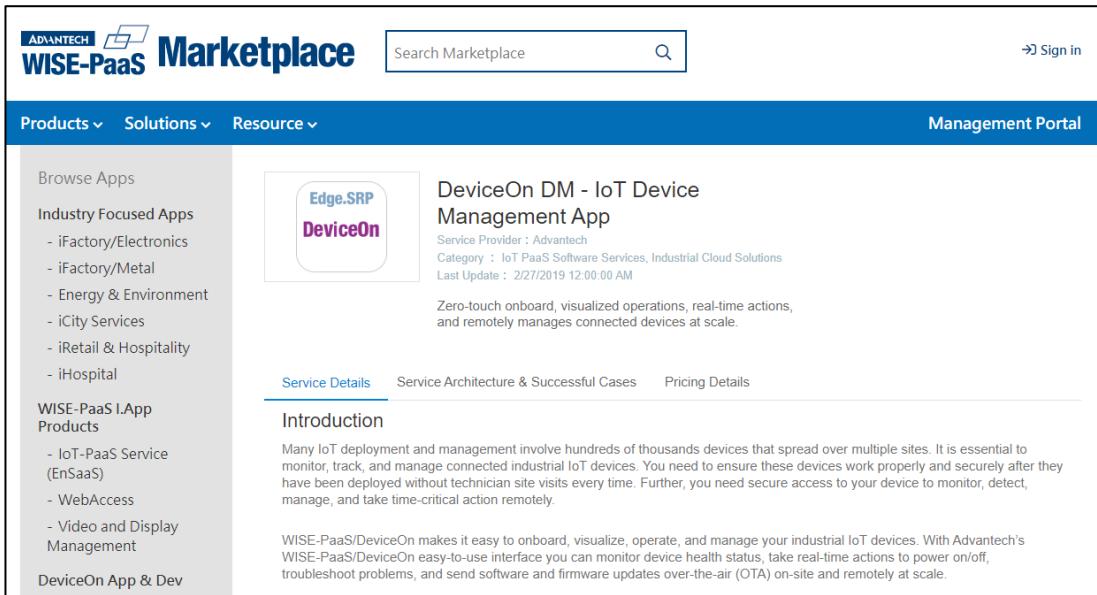
Advantech's brand-new designed IoT device operations and management App solution gives users a transformational plug-and-play experience. Beginning with onboarding devices, [WISE-DeviceOn](#)'s zero-touch IoT tech seamlessly registers Advantech hardware systems with identity security and field site settings. A fast and simple setup helps provide instant intelligent edge onboarding, data acquisition, and status visualization at the device operations center. Power on/off,

troubleshooting, and mission-critical actions are available at the tap of a button for quick and easy access. OTA software updates itself securely by sending software patch, firmware, software, and configuration updates through batch provisioning. The App is designed to ensure maximum efficiency in IoT device operations and management.



Power up your IoT devices with this hardware and software integrated solution. Get the most out of the WISE-DeviceOn's features with predictive device maintenance like IPC HDD lifecycle prediction, analytics-based dashboard and automated event alerts. In bringing artificial intelligence to your IoT needs, Advantech delivers improved risk management, faster daily operations, and better device performance while improving business value and intelligence through the extraction of big data.

WISE-DeviceOn is compatible with all Advantech hardware systems and works on popular platforms and services like the WISE-PaaS public/private cloud, Microsoft Azure, VM on-premise, and Kubernetes. [Get your WISE-DeviceOn version on the WISE-PaaS Marketplace](#) and kick-start your new and improved device operations and management experience.



The screenshot shows the WISE-PaaS Marketplace interface. At the top, there's a search bar labeled "Search Marketplace" with a magnifying glass icon and a "Sign in" button. Below the header, a navigation bar includes "Products", "Solutions", "Resource", and "Management Portal". The main content area displays a product listing for "DeviceOn DM - IoT Device Management App". On the left, a sidebar lists categories like "Browse Apps", "Industry Focused Apps" (with sub-options for iFactory/Electronics, iFactory/Metal, Energy & Environment, iCity Services, iRetail & Hospitality, iHospital), "WISE-PaaS I.App Products" (with sub-options for IoT-PaaS Service (EnSaaS), WebAccess, Video and Display Management), and "DeviceOn App & Dev". The main listing for "DeviceOn DM - IoT Device Management App" includes a thumbnail image of the Edge.SRP logo, the app name, service provider (Advantech), category (IoT PaaS Software Services, Industrial Cloud Solutions), last update (2/27/2019 12:00:00 AM), and a brief description: "Zero-touch onboard, visualized operations, real-time actions, and remotely manages connected devices at scale." Below the description are links for "Service Details", "Service Architecture & Successful Cases", and "Pricing Details".

## 1.1 Feature Highlights

- **Device Connectivity & Monitoring**

With more and more IoT devices in the field and the need for remote management and monitoring of those devices, the most important thing is how to achieve secure and fast onboarding to WISE-DeviceOn. There are two mechanisms provided, one is **Zero-touch**, where the user does not need to configure any of their devices. Just power-on the devices and they will connect to DeviceOn automatically. However, there is the limitation that the device's network must have the ability to directly connect to the public cloud. The second mechanism is called "**One-time configuration, automated onboarding**". Based on this mechanism, the user only sets up one device to connect to the cloud and uses this device to search and bring others to the cloud. Furthermore, this scenario supports public/private provisioning if there is no public cloud connection due to environmental limitations.

DeviceOn supports general real-time monitoring of device health that includes hard disk, CPU, memory, network load and provides various alerting mechanisms. Additional proprietary sensors such as CO<sub>2</sub>, battery monitoring or various proprietary protocols can be supported through design-in services.

- **Bulk Management & Maintenance**

For management and real-time control of a group of devices, DeviceOn offers a default overview with one-click actions, such as "One-Click Power On", "One-Click Protection", "One-Click Recovery", "One-Click Turn off backlight" and so on. Operators do not need to spend lots

of effort to setup devices one by one, but can simply “One-Click” maintain their field devices.

The following actions are supported by DeviceOn:

- Power Saving
  - Power On/Off, Reboot
  - \*Backlight On/Off
- Security
  - Protection On/Off
  - System Backup/Recovery
  - \*\*USB Lock/Unlock
    - Block USB drives and removable disks** (*Not supported on “Administrator” user*)
  - \*\*Keyboard Lock/Unlock
    - Block function key, such as “ALT”, “CTRL”, and windows key.**
  - \*\*Touch Gesture Lock/Unlock (*supported with capacitive touch panel only*)
  - \*\*Touch Lock/Unlock
- System
  - Screenshot
  - Audio Mute/Unmute
  - \*Watchdog Enable/Disable (*Default reset time is 60s*)
    - Reboots the system if it becomes unresponsive, to avoid hanging at “BSOD” (Blue Screen of Death) or similar situations**
  - \*\*Notification Block/Unblock
    - Disable windows notification from applications and other sources**
  - \*\*UWF Enable/Disable
    - Helps to protect your drives by intercepting and redirecting any writes to the drive (app installation, settings changes, saved data) to a virtual overlay**

Above actions prefixed with ‘\*’ require the respective Advantech SUSI Driver and actions prefixed with ‘\*\*’ require following operating systems:

- **Windows 10 Enterprise LTSC 2019 (LTSC)**
- **Windows 10 Enterprise 2016 LTSB (LTSB)**
- **Device Remote Control**
  - **Device Diagnostics**
    - Provides remote control mechanism, such as KVM (Remote Keyboard-Video-Mouse) for real-time remote desktop access to the devices. The screenshot functionality allows to capture the device’s current screen output for potential troubleshooting. Another feature is access to Windows or Linux shells, for example in order to quickly retrieve

network status via ipconfig/ifconfig, netstat to dump socket/TCP/UDP information, without having to use the full graphical user interface.

- **OTA (Over the Air)**

OTA supports an open framework, which can easily integrate 3rd party storage, such as FTP and cloud solutions (Azure Blob, AWS S3, AliYun, Openstack Swift). It does not only support remote update and deployment, but supports automatic update from server side as well as scheduled updates that get triggered from the agent side. Scheduling helps to avoid peak network traffic times and allows implementation of download and deployment schemes that reduce potential impact to a minimum.

The framework supports upgrade package backups as well as rollback to the previous version when required.

Scripting support (shell/batch) allows to implement flexible update mechanisms.

- **Power Management**

Sets the power on/off schedule for remotely located devices; the schedule can be set on a daily, weekly, monthly, or yearly basis. Supports Agent mode enable powering on across networks.

- **Protection Management**

DeviceOn system protection is powered by McAfee, providing white list protection against unauthorized application execution, and also sending warnings of any unauthorized activities.

- **Backup & Recovery**

DeviceOn system recovery is powered by Acronis, providing hot backup and scheduled backup, and also one-click recovery.

- **Simplified Operation & Support**

In general, the utmost goal of system integrators or IoT device operation managers is meeting service level KPIs without having to spend huge efforts or daily maintenance. Once hardware fails, it results in a serious increase in operation cost. DeviceOn provides rule-based management and implements HDD failure prediction. If a managed device shows any anomaly on a specific component or sensor, DeviceOn can send alert messages through **email** or **SMS**,

or can optionally integrate with social media services such as **LINE**, **WeChat**. The DeviceOn overview shows overall status, upcoming schedule, top 5 potential risk devices as well as device location at a glance.

There is a summary for these feature highlights on different operation system and hardware requirement.

	DeviceOn Feature Highlight	Windows 7, 8, 10	Windows 10 LTSC, LTSB	Ubuntu 16.04 x64	Linux on RISC (Yocto)	Android on RISC
<b>Standard Offering</b>	Role-Based Access Control	●	●	●	●	●
	Device Zero-touch Onboarding	●	●	●	●	●
	Device & Device Group Management	●	●	●	●	●
	Device Threshold Detection (Rule-based Engine)	●	●	●	●	●
	Notification & Alert Service (Mail, SMS, LINE, WeChat, WhatsApp, Telegram, Teams, Slack)	●	●	●	●	●
	Device Real-time & Historical Data Monitoring	●	●	●	●	●
	OTA, Software, Firmware Provisioning	●	●	●	●	●
	Power Control, Terminal, Screenshot, Remote Desktop	●	●	●	○	○
	Backup/Recovery, Protection	●	●	●		
	Device Data with Zero-Downtime	●	●	●	●	●
	Operation Management (Batch Control & Statistical Analysis)	●	●	●	●	●
	Audio Volume Control	●	●			
	1-Click to Data Visualization	●	●	●	●	●
	Statics System Report	●	●	●	●	●
	Intel AMT Remote Control and Management	●	●	●		
	Device Map (Open street, Google, Baidu)	●	●	●	●	●

<b>Advantech Hardware Support</b>	Hardware Watchdog Monitoring	●	●	●		
	Hardware GPIO Control & Customized	●	●	●	●	●
	Brightness & Backlight Control	●	●	●	●	●
	Hardware Sensor Monitoring	●	●	●	○	○
	BIOS Update	●	●	●		
	Advantech Industrial SQ Flash/RAM	●	●			
	Remote Management & Monitoring		●			
<b>Windows 10 Lockdown Features</b>	USB Drive Block		●			
	Keyboard Lock & Filter		●			
	Touch Screen & Gesture Lock		●			
	Windows Notification Block		●			
	UWF Protection		●			

## 1.2 DeviceOn Server Versions

DeviceOn is based on a microservice design, each component is stateless and supports multiple instances for scale up. This results in heavily simplified deployment to WISE-PaaS (Cloud Foundry), Azure PaaS, standalone virtual machines or Kubernetes. Both public cloud and private cloud (on-premise) deployments are supported. This chapter provides an introduction and provides a summary of requirements for those scenarios. The container version of DeviceOn starts from version number **v-1.1.x** (WISE-PaaS/Azure Kubernetes), while the standalone version starts from **v-4.2.x**. The standalone version comprises of IoTHub, database (PostgreSQL and MongoDB), Dashboard (Grafana), Webservices (Tomcat) and DeviceOn core applications.

### 1.2.1 Standalone, VM (Cloud)

The standalone version provides all packages of the DeviceOn software in one installer package, including RabbitMQ as a message broker, MongoDB, PostgreSQL as databases, Grafana for visualization, Tomcat for web services, and a watchdog service that protects DeviceOn core components from crashing or becoming unresponsive.

This section specifies the minimum hardware requirements for DeviceOn Cloud (Standalone) and the operating systems on which DeviceOn is supported. In general, the better the hardware configuration

of your computer, the better your experience with DeviceOn will be. To achieve a more satisfying experience with DeviceOn, particularly in terms of the client software, it is highly recommended that your system be substantially better than the minimum requirements specified in the following sections. This is particularly true if running server software locally on the same system as the client software.

Attention to the following areas can make a significant improvement to your overall user experience and enjoyment of the software:

- Memory - the more RAM your computer has, the better.
- CPU speed - the faster, the better.
- Hard Drive - the larger, the better.

General Operation Systems and Recommendations:

- ✓ **Windows Server 2008 R2 64-bit ([KB2999226 Required](#))**
- ✓ **Windows Server 2012 R2 Standard 64-bit ([KB2919442](#), [KB2919355](#), [KB2999226 Required](#))**
- ✓ **Windows Server 2012 R2 Datacenter 64-bit ([KB2999226 Required](#))**
- ✓ **Windows Server 2016/2019 64-bits**

#### *Reserve Port for DeviceOn Server Used*

Name & Description		Inbound Port
<b>1</b>	DeviceOn HTTP, HTTPS Web Services	80, 443 [Depends on Installation]
<b>2</b>	DeviceOn Dashboard (Grafana)	3000 [Depends on Installation]
<b>3</b>	Message Broker (RabbitMQ) MQTT, MQTTS	1883, 8883
<b>4</b>	Message Broker (RabbitMQ) AMQP, AMQPs	5671, 5672
<b>5</b>	Message Broker (RabbitMQ) Management Console	15672
<b>6</b>	Repeater for Remote Desktop	5501
<b>7</b>	Websockify for Remote Desktop	6083 ~ 6183
<b>8</b>	Database for MongoDB	27017
<b>9</b>	Database for PostgreSQL	5432

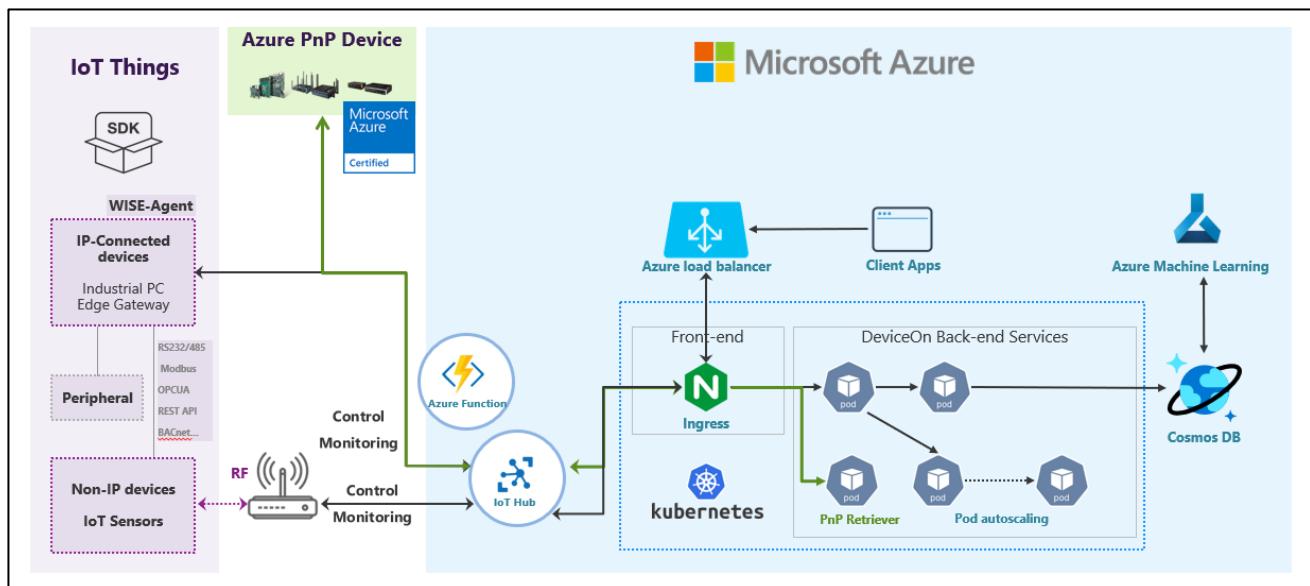
Hardware Minimum Requirements:

- ✓ **Intel® Core™ i5 2.3 GHz CPU and at least 8GB of RAM**
- ✓ **25 GB root partition for the system**
- ✓ **100 GB data storage partition (for documents and indexing)**

### 1.2.2 Azure PaaS, Kubernetes (Cloud)

The Azure Kubernetes Service (AKS) makes it easy to deploy a managed Kubernetes cluster to Azure. AKS reduces the complexity and operational overhead of managing Kubernetes by offloading much of that responsibility to Azure. Azure handles critical tasks like health monitoring and maintenance for those Kubernetes services.

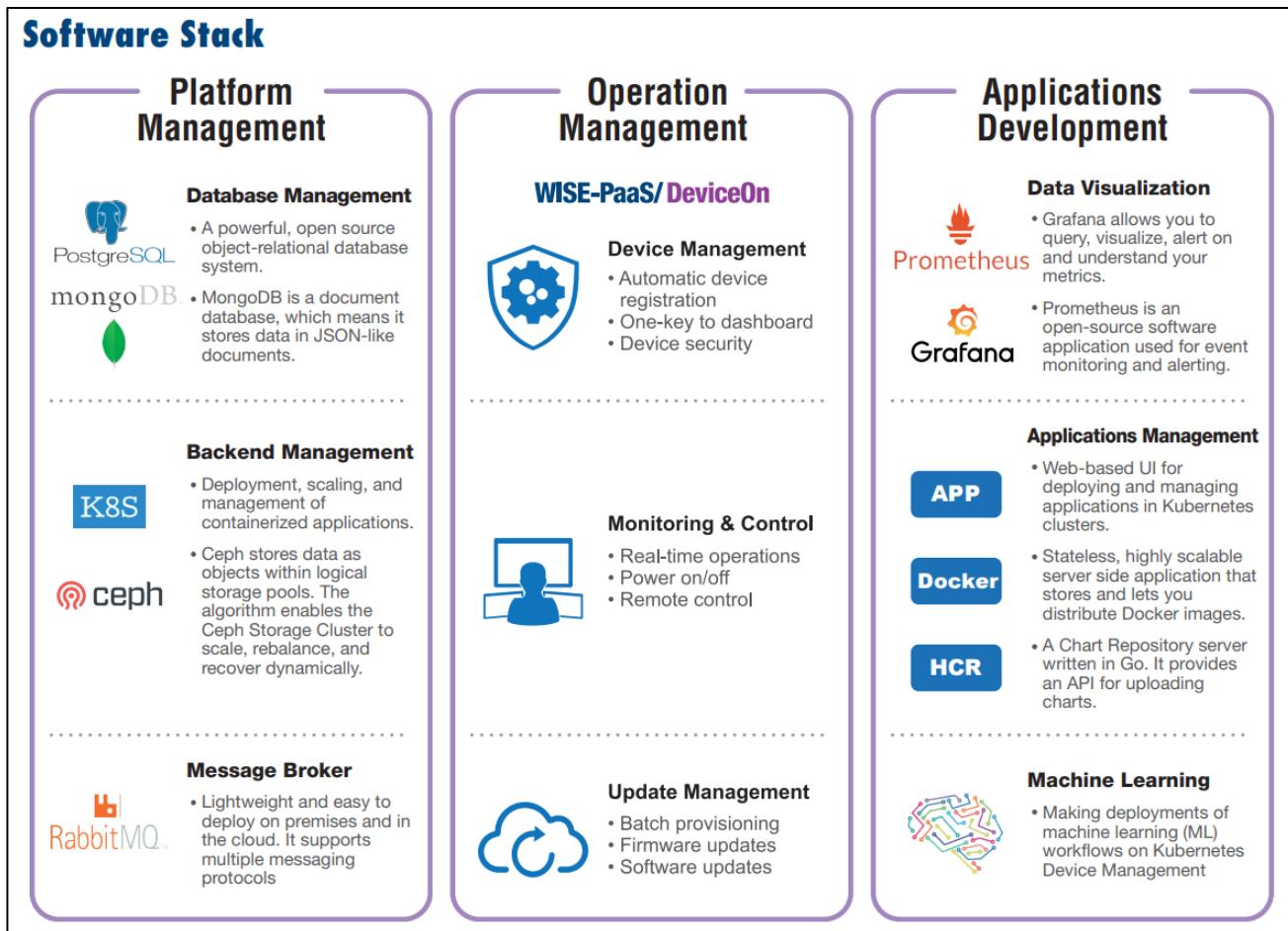
Deploying DeviceOn on the Azure Kubernetes Service is easy and with just a few steps, containers or nodes can be scaled up to manage thousands of devices. Moreover, DeviceOn can leverage the Azure IoT Hub and Cosmos DB for Azure native security and performance. Since the data is already stored on the Azure cloud, it is much easier to leverage the Azure ecosystem – for example using the provided data for Azure Machine Learning. DeviceOn can be deployed to Azure Kubernetes directly from the WISE-PaaS/Marketplace.



### 1.2.3 Data Service Server for Private Cloud

For accelerated IoT application deployment, Advantech offers the Data Service Server [EIS-S230](#) as a stable and reliable all-in-one solution for your back-end data service or light private cloud. It is built around an Intel Xeon or Core i7 CPU to offer best in class computing performance for data services. Moreover, EIS-S230 comes preinstalled with Kubernetes to support micro-services, as well as complete back-end software components including RabbitMQ as IoT Hub, MongoDB and PostgreSQL as database, Grafana for data visualization and Prometheus for back-end management. EIS-S230 also provides a dynamic scale out function that allows extension of resources as necessary. It is a perfect tool to create IoT applications more easily and flexibly and to speed up time to market.

## Software Stack



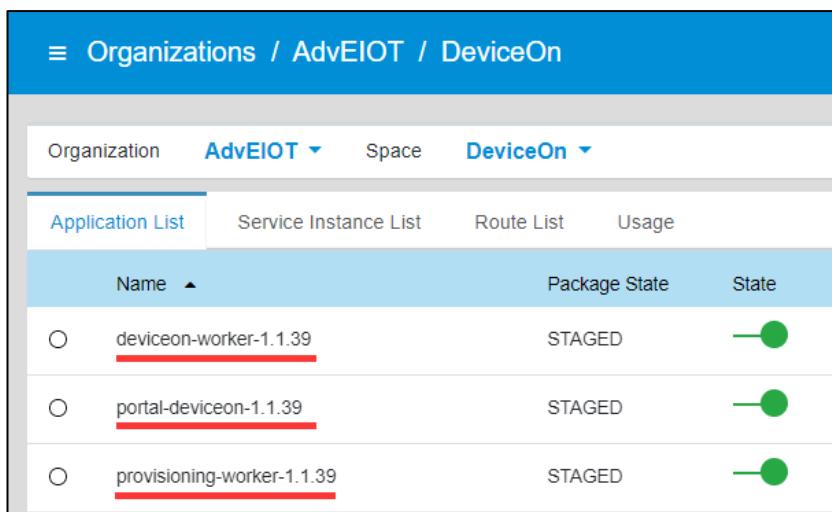
### Features:

- Integrated solution (HW+SW bundle) for back-end data service and light private cloud
- Pre-configured system: Intel Xeon platform with 32GB RAM, 512GB mSATA SSD including Ubuntu Linux OS
- Open and flexible infrastructure: Kubernetes support, multiple database options, on-demand microservices
- Integrated IoT Software: Private Cloud Deployment, Platform Management, Application Integration
- Integrated Applications: WISE-DeviceOn, Grafana, Prometheus, Kubeapps, Kubernetes Dashboard
- Sustainable Management: Condition Monitor, Load Balance, Advanced Recovery
- **WISE-DeviceOn inside for feature-rich IoT Device Management**

#### 1.2.4 WISE-PaaS/EnSaaS (Cloud)

The WISE-PaaS/EnSaaS version consists of three containers as listed below. In this scenario DeviceOn requires 1408 MB of RAM at least.

Application Name	Version	Memory Used	Purpose
deviceon-worker-1.1.x	v-1.1.x	384MB	Worker that processes device messages, status, notification, scheduling etc.
portal-deviceon-1.1.x	v-1.1.x	768MB	Provides the DeviceOn web interface for remote control and monitoring.
provisioning-worker-1.1.x	v-1.1.x	256MB	Worker that provisions devices with configuration, software, firmware etc.



The screenshot shows a web-based interface for managing applications. At the top, there's a header bar with tabs for 'Organization', 'AdvEIOT', 'Space', and 'DeviceOn'. Below that is a sub-header with tabs for 'Application List', 'Service Instance List', 'Route List', and 'Usage'. The main content area displays a table with columns for 'Name', 'Package State', and 'State'. There are three rows in the table, each corresponding to one of the three workers listed above. Each row has a red horizontal underline under the 'Name' column. The 'Package State' column shows 'STAGED' for all three, and the 'State' column shows a green circular icon with a white play symbol.

Name	Package State	State
deviceon-worker-1.1.39	STAGED	
portal-deviceon-1.1.39	STAGED	
provisioning-worker-1.1.39	STAGED	

### 1.3 DeviceOn Agent Versions

Advantech provides a device client that is used to communicate and exchange information between IoT (Internet of Things) devices and the DeviceOn cloud services, called **WISE-Agent**. WISE-Agent provides a rich set of user-friendly features that are intelligent, standardized and scalable.

- Standardization

The communication protocol between client and cloud is based on the industry standard MQTT protocol. The IoT sensor data format is following the IPSO Alliance definition, implemented in JSON.

- Portability

The whole framework is written in C language and follows the ANSI C Standard. C compilers are widely available for most platforms and allow easy porting to different architectures or operating systems.

- Scalability

The WISE-Agent has a modular design and provides a plugin concept that allows flexible addition of new data sources or extra functionality.

#### 1.3.1 WISE-Agent (Client)

WISE-Agent is support on different platforms running Windows 7 (or newer) or Ubuntu 16.04 x64 (or newer). Please contact us for others architectures (e.g. RISC) or operating systems (e.g. Yocto based Linux/Android).

General Operation Systems and Recommendations:

- ✓ **Windows 7 SP1/8/10 32-bit/64-bit**
- ✓ **Ubuntu 16.04, 18.04, 20.04 x64**
- ✓ **CentOS 7.6, 8.2 x64**
- ✓ **Other Linux flavours (e.g. Yocto) on x86 or RISC (on a per project basis)**
- ✓ **Android on RISC (on a per project basis)**

#### *Assigned Ports for Device Communication*

Name & Description		Outbound Port
1	MQTT, MQTTs Message Client	1883, 8883
2	Remote Desktop VNC Client	5501

Hardware Minimum Requirements:

- ✓ **Intel® Celeron™ 1.10 GHz CPU and at least 2GB of RAM**
- ✓ **500 MB root partition for the system**
- ✓ **Advantech HW with respective SUSI driver 3.02/4.0 support is required for the HWM (Hardware Monitoring Management) feature to be available**

## 1.4 Security

#### 1.4.1 Role-Based Access Control (RBAC)

DeviceOn supports three different user roles - “Root” (perpetual version only), “System Admin” and “Device Admin”. There is only one single “Root” account per system, which has the highest permission level and can create “System Admin” or “Device Admin” accounts. The intermediate user level “System Admin” can be used to create “Device Admin” accounts. “Device Admin” accounts have the lowest permission level. Please refer to Section 7.1 for details on access permission levels.

#### 1.4.2 SSL Encryption

- **HTTPS on DeviceOn Web Server**

The principal motivations for HTTPS are authentication of the accessed website, protection of the privacy and integrity of the exchanged data while in transit. It protects against man-in-the-middle attacks. The bidirectional encryption of communications between a client and server protects against eavesdropping and tampering of the communication.

- **SSL Connection on Database (PostgreSQL, MongoDB)**

PostgreSQL and MongoDB have native support for using SSL connections to encrypt client/server communications for increased security.

- **Create Security Credentials on Database**

Databases are by default protected by secure credentials and require explicit authentication for connections. This avoids accidentally deploying platforms with unprotected access.

- **Device Connectivity via MQTT SSL**

RabbitMQ supports multiple protocols including MQTT, which the most popular IoT (Internet of Things) protocol. By default, SSL is used to encrypt all MQTT traffic for device connectivity.

- **Enforce Password Policies**

While DeviceOn allows you to set some of your own passwords, please make sure those meet the minimum complexity requirements established by your specific organization.

#### 1.4.3 Security Scan

The DeviceOn server pass through below famous vulnerability tools to ensure security for your AIoT solutions. Furthermore, all the testing including anti-malware (**Trend Micro** and **Kaspersky**)

- **Web Application Assessment Report (Micro Focus)**

[WebInspect](#) is an automated dynamic testing tool that mimics real-world hacking techniques and attacks, and provides comprehensive dynamic analysis of complex web applications and services.

- **OpenVAS (Open Vulnerability Assessment System)**

[OpenVAS](#) is a full-featured vulnerability scanner. Its capabilities include unauthenticated testing, authenticated testing, various high level and low-level Internet and industrial protocols, performance tuning for large-scale scans and a powerful internal programming language to implement any type of vulnerability test.

The scanner is accompanied by a vulnerability tests feed with a long history and daily updates.

This [Greenbone Community Feed](#) includes more than 50,000 vulnerability tests.

- **Nessus**

[Nessus](#) is the de-facto industry standard vulnerability assessment solution for security practitioners. The latest intelligence, rapid updates, an easy-to-use interface.

- ✓ Covers an industry-leading 47,000+ vulnerabilities
- ✓ Unlimited scans at no extra cost
- ✓ Compliant with PCI, HIPPA, GLBA, CIS, NIST, and more

- **OWASP ZAP**

The [OWASP Zed Attack Proxy \(ZAP\)](#) is one of the world's most popular free security tools and is actively maintained by hundreds of international volunteers\*. It can help you automatically find security vulnerabilities in your web applications while you are developing and testing your applications. It's also a great tool for experienced pen testers to use for manual security testing.

- **Skipfish**

[Skipfish](#) is an active web application security reconnaissance tool. It prepares an interactive sitemap for the targeted site by carrying out a recursive crawl and dictionary-based probes. The resulting map is then annotated with the output from a number of active (but hopefully non-disruptive) security checks. The final report generated by the tool is meant to serve as a foundation for professional web application security assessments.

Key features:

- ✓ High speed: pure C code, highly optimized HTTP handling, minimal CPU footprint – easily achieving 2000 requests per second with responsive targets.
- ✓ Ease of use: heuristics to support a variety of quirky web frameworks and mixed-technology sites, with automatic learning capabilities, on-the-fly wordlist creation, and form auto completion.
- ✓ Cutting-edge security logic: high quality, low false positive, differential security checks, capable of spotting a range of subtle flaws, including blind injection vectors.

- **Nikto**

[Nikto](#) is an Open Source (GPL) web server scanner which performs comprehensive tests against web servers for multiple items, including over 6700 potentially dangerous files/programs, checks for outdated versions of over 1250 servers, and version specific problems on over 270 servers. It also checks for server configuration items such as the presence of multiple index files, HTTP server options, and will attempt to identify installed web servers and software. Scan items and plugins are frequently updated and can be automatically updated.

- **W3af**

[w3af](#) is a **Web Application Attack and Audit Framework**. The project's goal is to create a framework to help you secure your web applications by finding and exploiting all web application vulnerabilities.

- **Arachni**

[Arachni](#) is a fully featured web security scanning tool, it is based on ruby framework. It is an open source, modular and high performance tool. It comes with both command line interface as well as web based gui interface, it is highly versatile tool for security scanning purpose. It supports almost all of the popular web application such as HTML5, Java Script and AJAX etc, additionally it is enables with multi user-multi platform collaboration. It allows you to generate reports in desired format (.txt, XML, HTML).

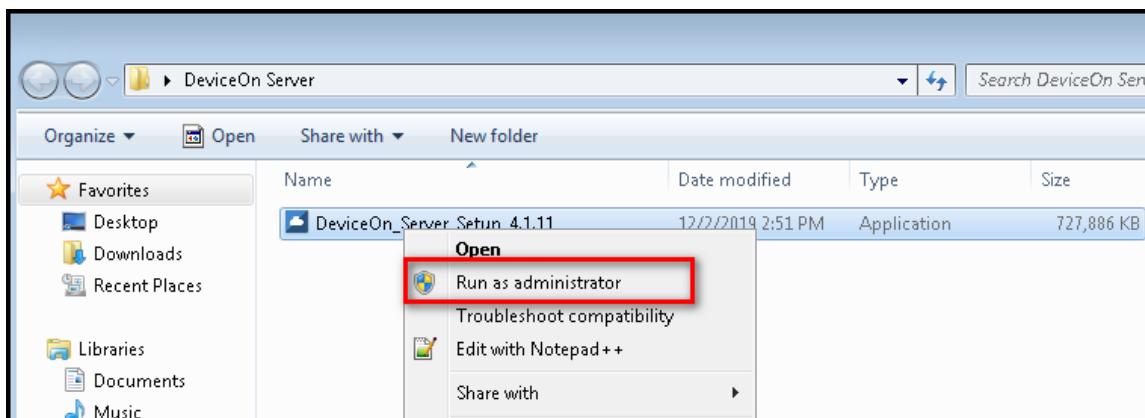
## 2. Getting Started

### 2.1 DeviceOn Cloud Installation

#### 2.1.1 Setup Standalone Version (On-premise)

##### Step 1: Install the DeviceOn package on your system

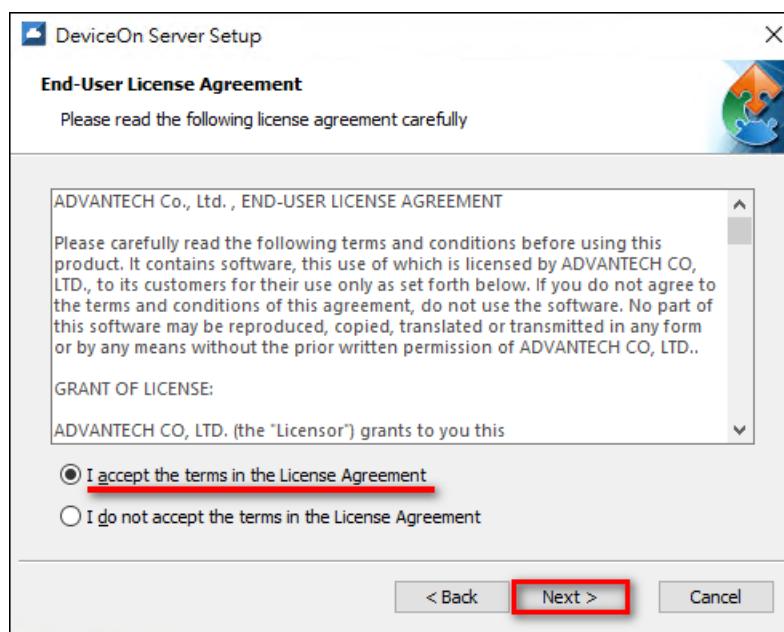
Copy the installation file (**DeviceOn\_Server\_Setup\_4.2.x.exe**) to your target system and run it as administrator.



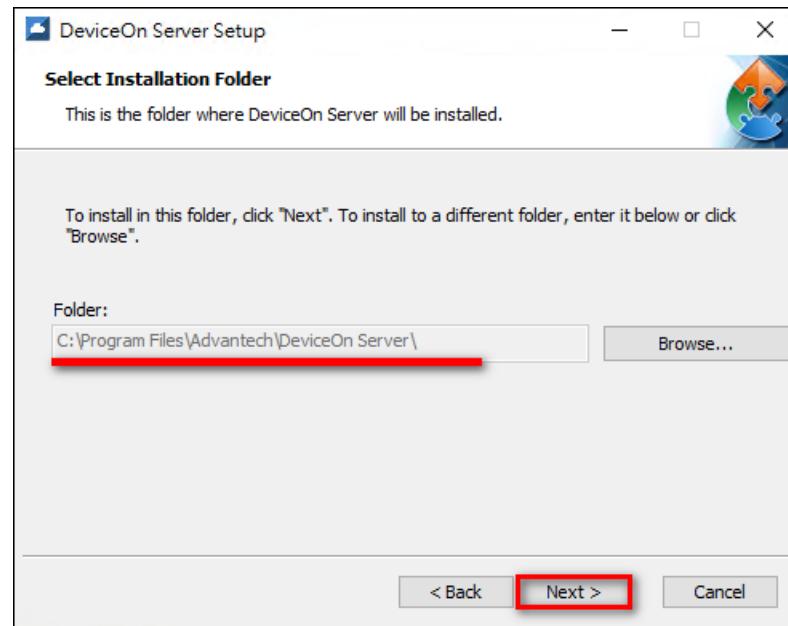
Click “**Next**” to start the installation process.



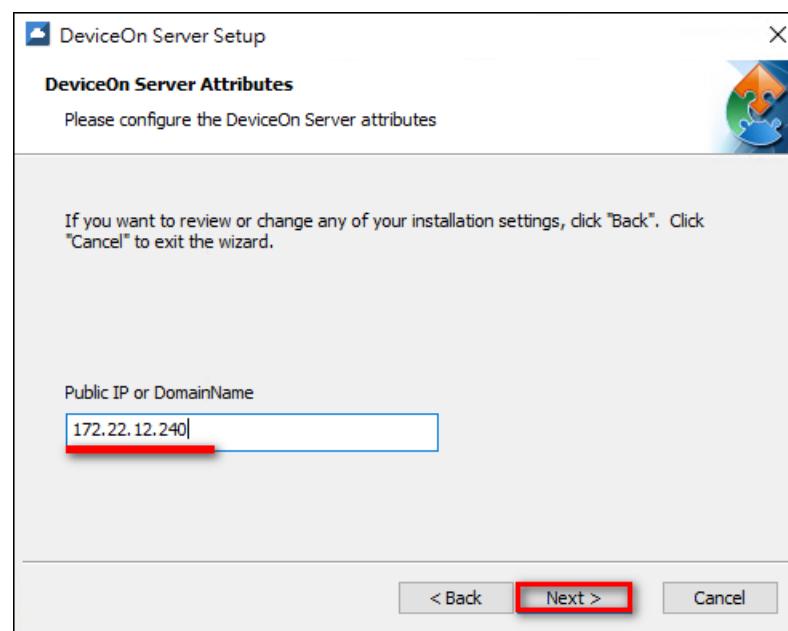
Select “I Accept the terms in the License Agreement” and click “Next”



Select the “Installation Folder” for DeviceOn Server and click “Next”



Enter “**Public IP**” or “**Domain Name**” for this physical/virtual machine and click “**Next**”. This information is required for “Edge Device” connectivity, please make sure your device is reachable under this IP or Domain Name.



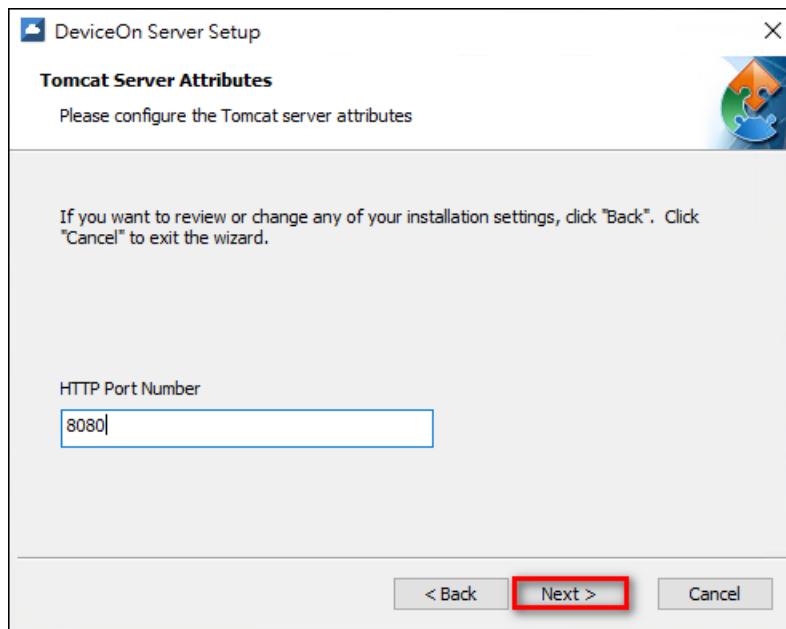
**Note:** You can start a Windows command prompt and type “ipconfig” to retrieve your IP address(es) on this physical/virtual machine.

```
C:\Users\Sephiroth>ipconfig
Windows IP Configuration

Ethernet adapter 乙太網路 2:
  Media State . . . . . : Media disconnected
  Connection-specific DNS Suffix . . .

Ethernet adapter 乙太網路:
  Connection-specific DNS Suffix . . . : ADVANTECH.CORP
  Link-local IPv6 Address . . . . . : fe80::194f:a776:464c:eb9b%9
  IPv4 Address . . . . . : 172.22.12.240
  Subnet Mask . . . . . : 255.255.252.0
  Default Gateway . . . . . : 172.22.15.254
                                         172.22.15.130
```

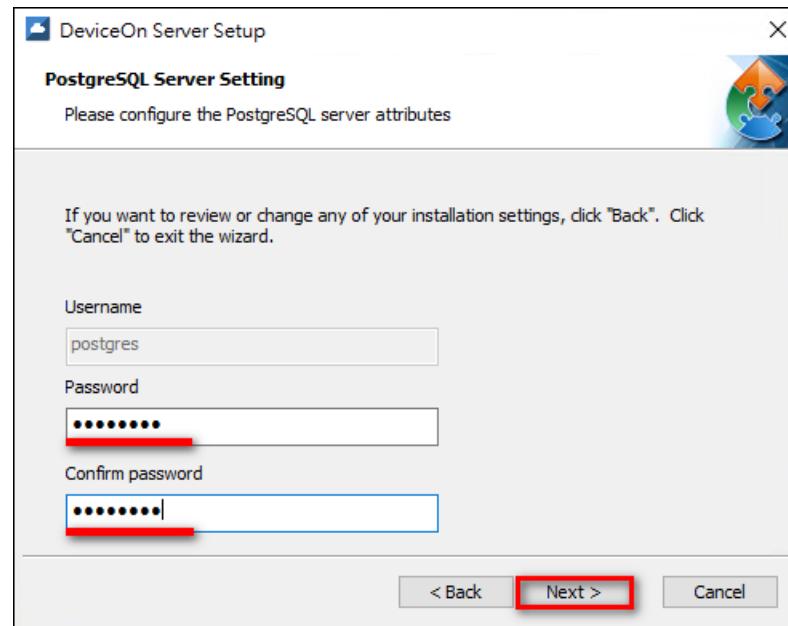
You will need to configure the HTTP port number that is used for web browser-based access the DeviceOn management portal. The default port is 8080, but you can select any other port as long as it does not conflict with any other application or service. Click “**Next**”.



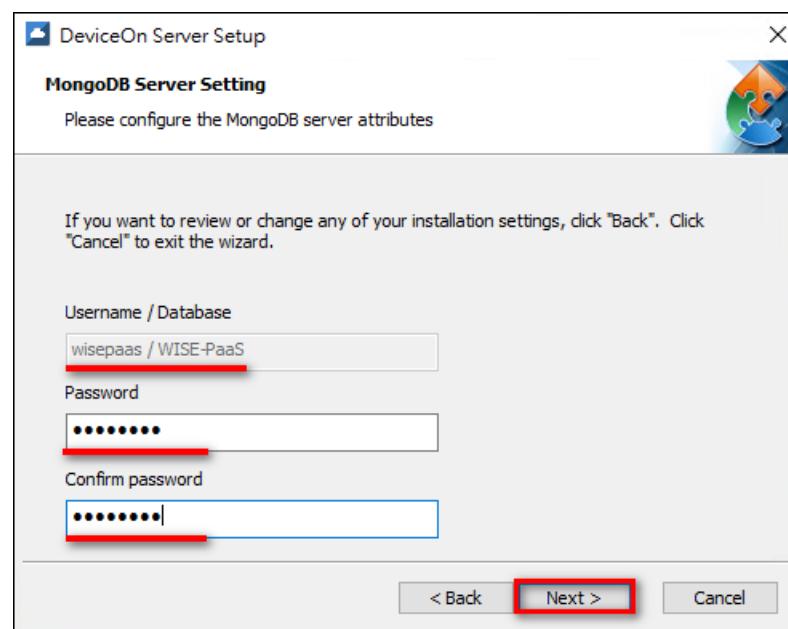
Configure the password of the relational database (PostgreSQL) that DeviceOn uses to manage account, device, permission, and relation data. The default account name is “**postgres**” and the password should follow below guideline.

#### **Strong Password Rules:**

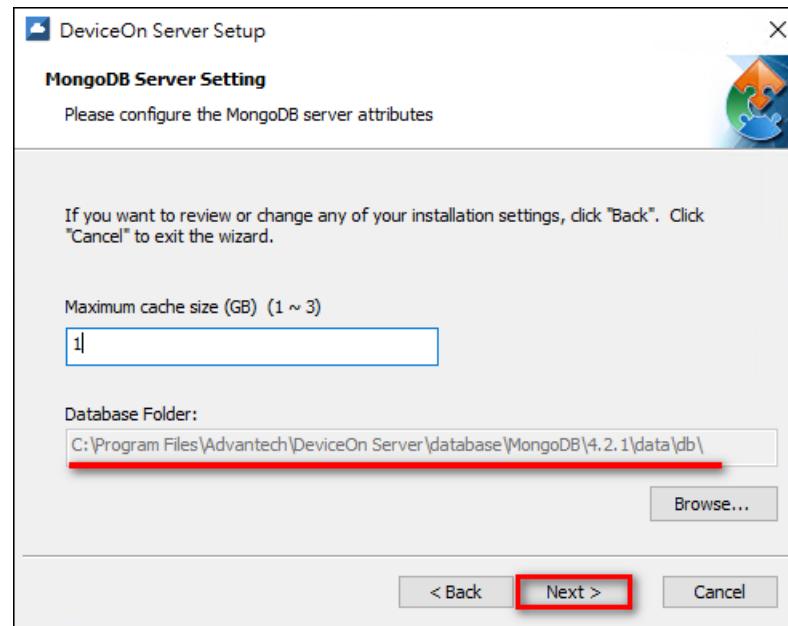
*Minimum eight characters, at least one number, one lowercase letter, one uppercase letter, and one special character (Blank character, Backslash(\), Double quotes(") are prohibited)*



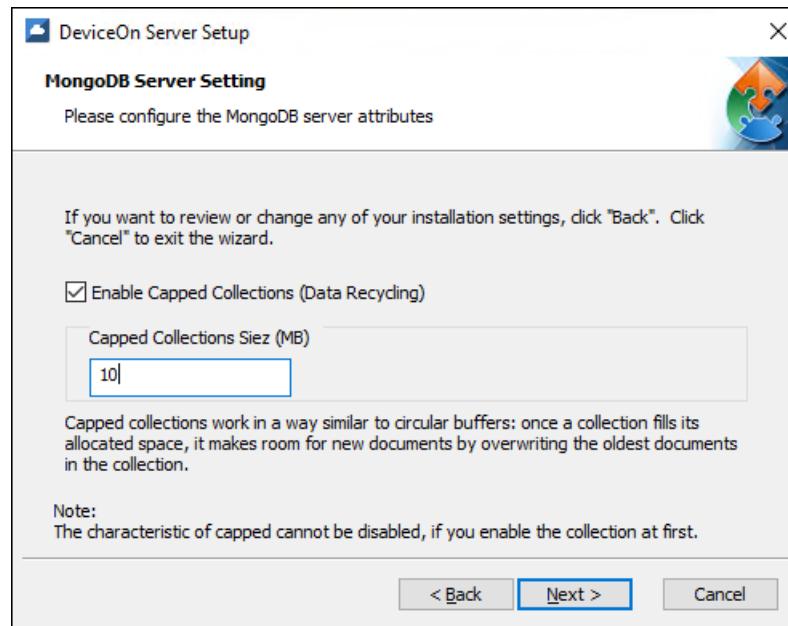
Configure the password of the NoSQL database (MongoDB) that stores device sensor data. The default account and database is “wisepaas/WISE-PaaS”. This password should also follow strong password rules as outlined above.



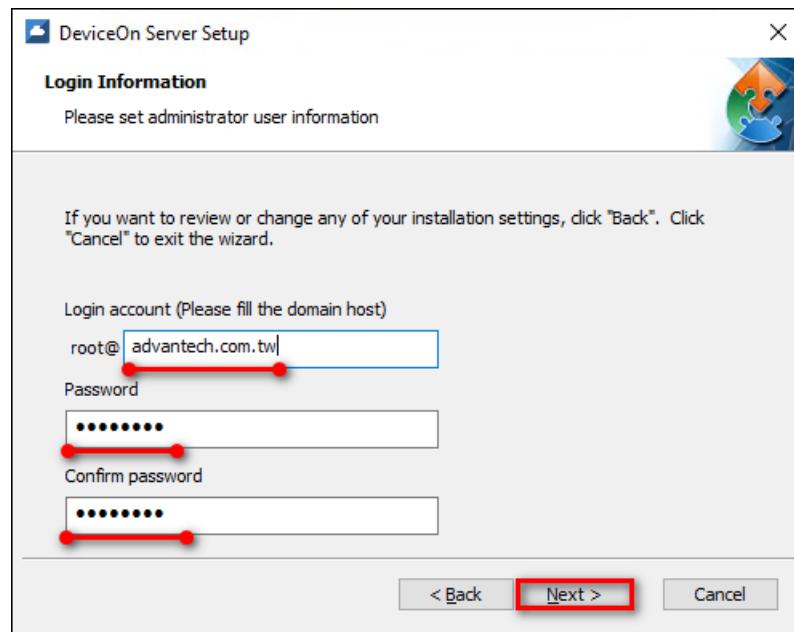
Select the database installation path and cache size of MongoDB and click “**Next**”. A larger cache size will result in better performance. For more information on this parameter, please refer to the [official documentation](#).



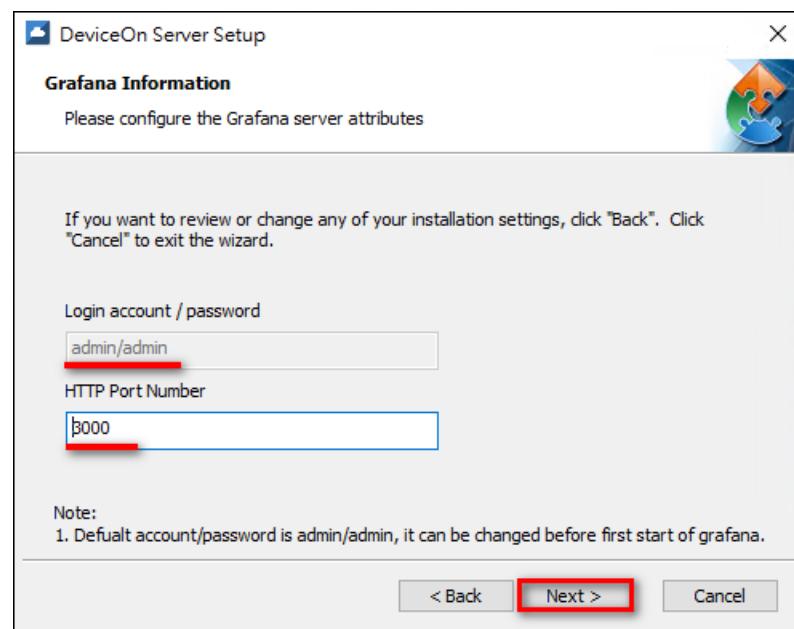
In order to avoid the hard disk space being used up by device data, the MongoDB provide a capped mechanisms, that similar to circular buffers: once a collection fills its allocated space, it makes room for new documents by overwriting the oldest documents in the collection. **Please note, the characteristic of capped cannot be disable, if you enable the collection at first.**



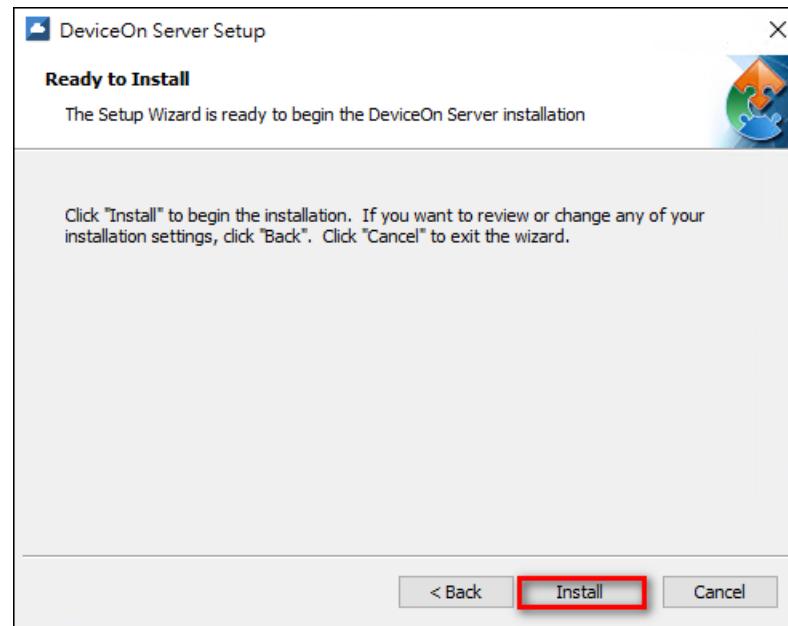
Configure the password and suffix domain of the root account (dummy name “root@”) and click “Next”. This root account has the highest permission level and is used to log in to the DeviceOn web service and create other user accounts.



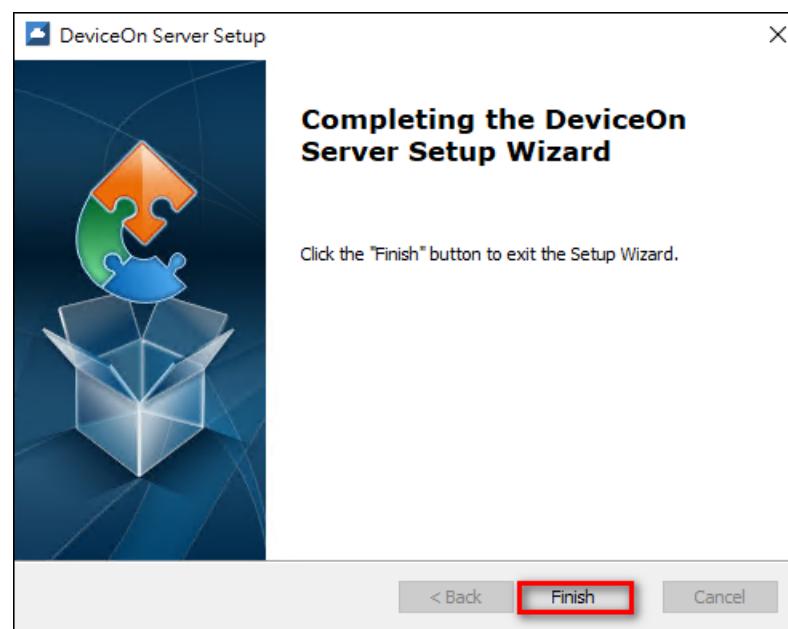
Set up the HTTP service port for Grafana dashboard. The default user name and password is admin/admin. You will be able to modify this at the first login.



Click “Install” to begin the installation.



Click “Finish” to exit the program.



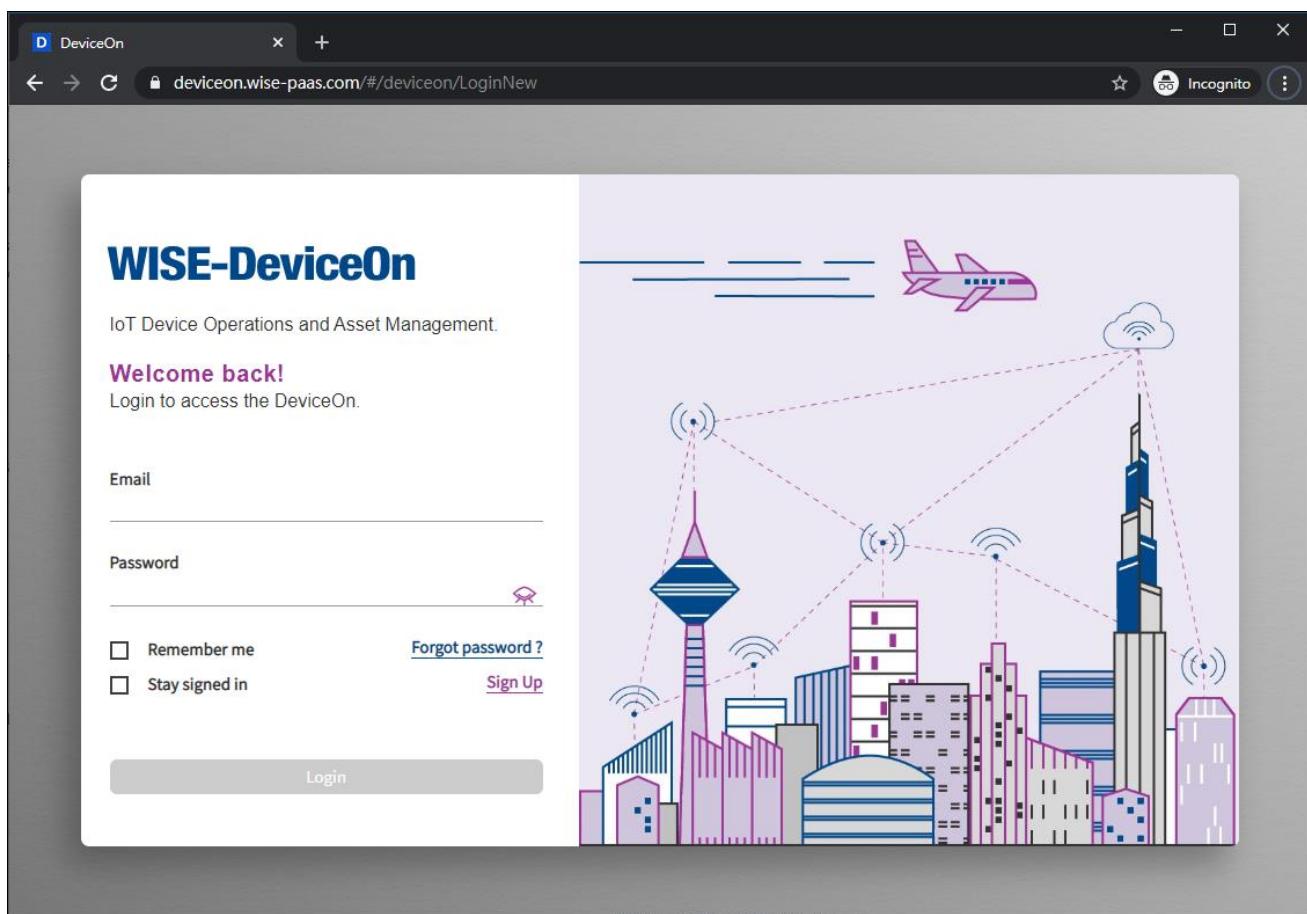
## Step 2: Launch DeviceOn Web Service Shortcut on Desktop

Two shortcuts will be generated on the desktop - one is for the DeviceOn web portal and the other one is for the Grafana dashboard.



Click the “DeviceOn Server” shortcut in order to launch a browser and to start device operation and

management. It is recommended to use **Chrome** for the best user experience.



### 2.1.2 Setup Standalone Version for Ubuntu Linux (On-premise)

If you are interested in DeviceOn and used to Linux platform, On-Premise, we also provide an installer for Ubuntu Linux (one of the most popular Linux distribution). This section will guide you how to install DeviceOn on Ubuntu Linux.

Note here that:

- The DeviceOn Ubuntu Linux installer is named something like "**DeviceOn\_Server\_Ubuntu 18.04\_x64\_4.2.x.run**". To acquire the installer and ensure having the latest version, please contact us.
- If you are running the installer with an account other than "root", you should use "**sudo**" command to obtain higher privileges, or the installation may fail at any step.

#### Step 1: Open a terminal

The installer runs in CLI (Command Line Interface) mode. As such, open a terminal preferable for you.

#### Step 2: Copy the installer to target host

Use the way you like to copy the installer to the target host.

### Step 3: Set the installer as executable

In the terminal, run "**chmod 0755 DeviceOn\_Server\_Ubuntu 18.04\_x64\_4.2.x.run**" so that the installer as an executable file under Ubuntu Linux.

### Step 4: Running the installer

Change your working directory to where the installer is and run "**./ DeviceOn\_Server\_Ubuntu 18.04\_x64\_4.2.x.run**". You may need to run "**sudo ./ DeviceOn\_Server\_Ubuntu 18.04\_x64\_4.2.x.run**" to acquire higher privileges if you were logged in as a normal user.

### Step 5: Answering some questions

Throughout installation process, it's necessary to answer some questions to complete the installation:

- A. The password of user "**postgres**" to login PostgreSQL database.

→ PostgreSQL password setup.  
↳ You need to input a password for super user 'postgres'

When you run into this step the question shows like above. Just input the password you would like to use to login PostgreSQL database for "**postgres**" account.

- B. The password of user "**wisepaas**" to login MongoDB database.

→ MongoDB password setup.  
↳ You need to input a password for user 'wisepaas' within database 'WISE-PaaS'

When you run into this step the question shows like above. Just input the password you would like to use to login MongoDB database for "**wisepaas**" account.

- C. The valid IP or host name of the target host.

→ A valid IP or host name is required.  
↳ The IP or host name you input here will be used by agents to acquire connection information.

When you run into this step the question shows like above. Just input the IP address of the target host. A hostname (even a FQDN) is also acceptable if you are sure that agents can connect to via the name you provide.

- D. If turn MongoDB capped functionality on or not.

→ Turn 'capped' on or not.

- ↳ MongoDB has a feature named 'capped'. It will recycle disk size for those collections turn this functionality on.

When you run into this step the question shows like above. Just input “**yes**” or “**no**” to enable or disable “capped” functionality. If you answer “yes”, a subsequent question followed to ask you “how much capped size, in MB, to be used? ”. Just input the size, in MB, you want to use in “capped” functionality in MongoDB database.

Capped collections are fixed-size collections that support high-throughput operations that insert and retrieve documents based on insertion order. Capped collections work in a way similar to circular buffers: once a collection fills its allocated space, it makes room for new documents by overwriting the oldest documents in the collection.

E. The password of user “**root@advantech.com.tw**” to login DeviceOn portal, and the rule should follow below guideline.

#### **Strong Password Rules:**

*Minimum eight characters, at least one number, one lowercase letter, one uppercase letter, and one special character (Blank character, Backslash(\), Double quotes(") are prohibited)*

→ DeviceOn portal password setup.

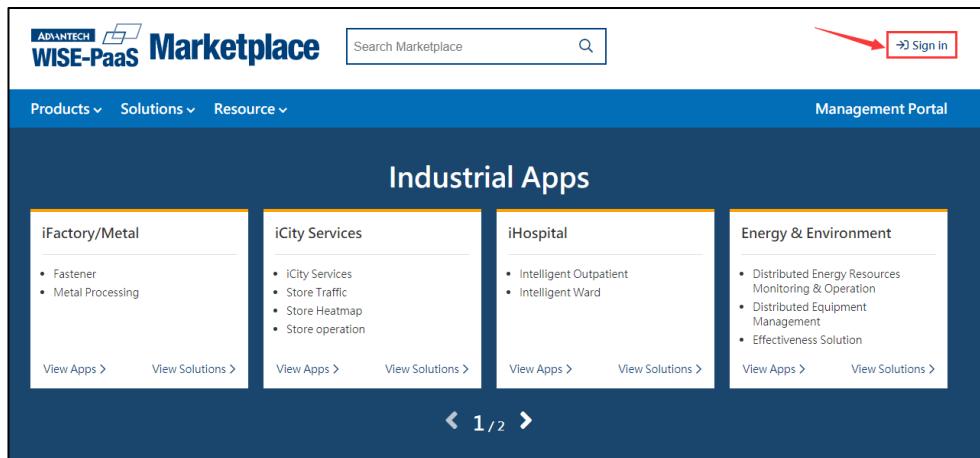
- ↳ You need to input a password for super user 'root' to login DeviceOn portal
- ↳ NOTE THAT A VALID PASSWORD TO LOGIN PORTAL MUST CONTAIN:
  - ↳ 1) at least eight characters
  - ↳ 2) at least a number
  - ↳ 3) at least a lowercase letter
  - ↳ 4) at least an uppercase letter
  - ↳ 5) at least a special character but ' ', '\', and '\"'.

When you run into this step the question shows like above. Just input the password you would like to use to login DeviceOn portal for “**root@advantech.com.tw**” account.

Finally, a workable DeviceOn server should be there the target host. Open a browser and input <http://{IP-US-ED-IN-QUESTION-C}>, you should see the DeviceOn login page.

#### **2.1.3 Redeem AKS Version from WISE-PaaS Marketplace**

**Step 1:** Sign in to your MyAdvantech Account on [Marketplace or create one](#)



**WISE-PaaS Marketplace**

Search Marketplace

Management Portal

Products ▾ Solutions ▾ Resource ▾

## Industrial Apps

**iFactory/Metal**

- Fastener
- Metal Processing

[View Apps >](#) [View Solutions >](#)

**iCity Services**

- iCity Services
- Store Traffic
- Store Heatmap
- Store operation

[View Apps >](#) [View Solutions >](#)

**iHospital**

- Intelligent Outpatient
- Intelligent Ward

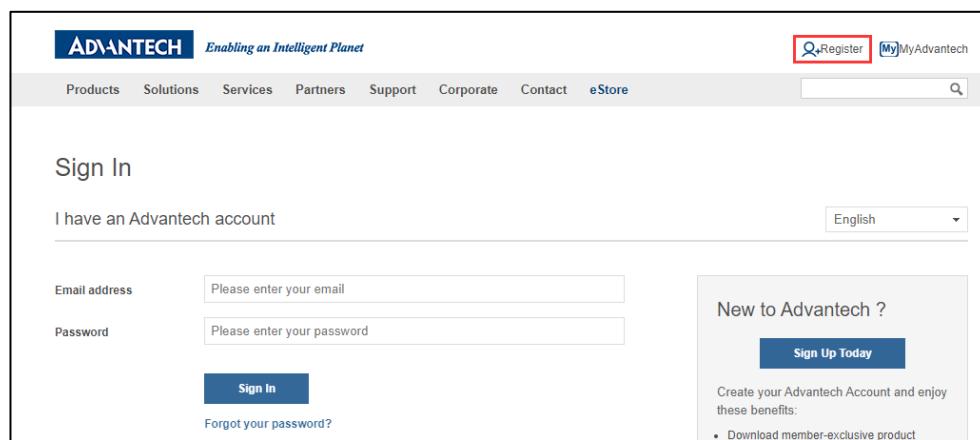
[View Apps >](#) [View Solutions >](#)

**Energy & Environment**

- Distributed Energy Resources Monitoring & Operation
- Distributed Equipment Management
- Effectiveness Solution

[View Apps >](#) [View Solutions >](#)

◀ 1 / 2 ▶



**ADVANTECH Enabling an Intelligent Planet**

Products Solutions Services Partners Support Corporate Contact eStore

## Sign In

I have an Advantech account

Email address  Please enter your email

Password  Please enter your password

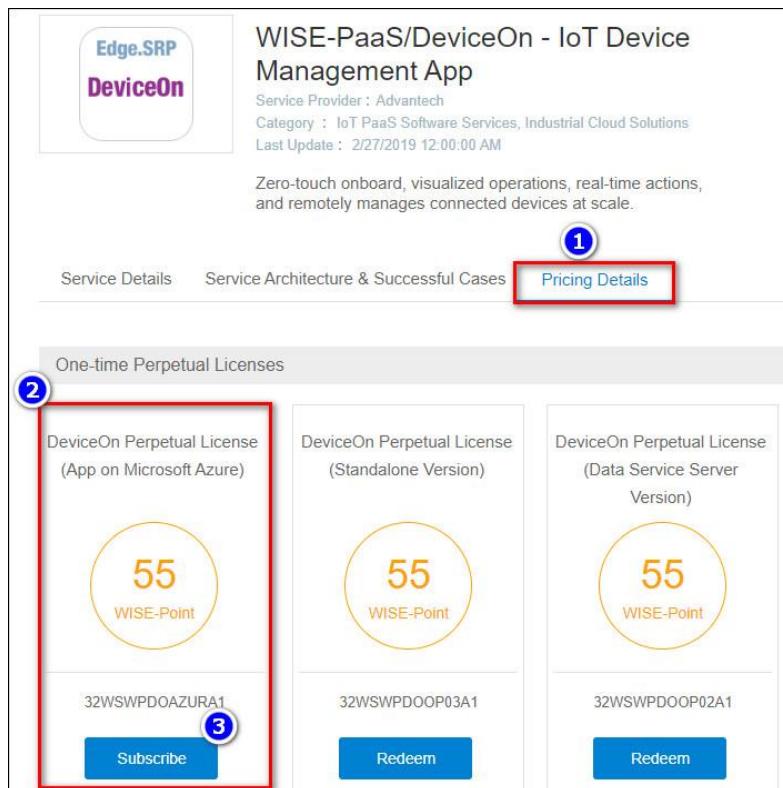
**Sign In** [Forgot your password?](#)

New to Advantech ? [Sign Up Today](#)

Create your Advantech Account and enjoy these benefits:

- Download member-exclusive product

## Step 2: Redeem "[WISE-DeviceOn](#) (App on Microsoft Azure)"



**WISE-PaaS/DeviceOn - IoT Device Management App**

Service Provider : Advantech  
 Category : IoT PaaS Software Services, Industrial Cloud Solutions  
 Last Update : 2/27/2019 12:00:00 AM

Zero-touch onboard, visualized operations, real-time actions, and remotely manages connected devices at scale.

**Pricing Details** 1

One-time Perpetual Licenses			
<span style="color: blue; border: 1px solid blue; border-radius: 50%; padding: 2px 5px;">2</span>	<b>DeviceOn Perpetual License (App on Microsoft Azure)</b> <div style="text-align: center;"> <span style="font-size: 2em; border: 1px solid orange; border-radius: 50%; padding: 10px; display: inline-block;">55</span>            WISE-Point            32WSWPDOAZURA1  <a href="#">Subscribe</a> <span style="color: blue; border: 1px solid blue; border-radius: 50%; padding: 2px 5px;">3</span> </div>	<b>DeviceOn Perpetual License (Standalone Version)</b> <div style="text-align: center;"> <span style="font-size: 2em; border: 1px solid orange; border-radius: 50%; padding: 10px; display: inline-block;">55</span>            WISE-Point            32WSWPDOOP03A1  <a href="#">Redeem</a> </div>	<b>DeviceOn Perpetual License (Data Service Server Version)</b> <div style="text-align: center;"> <span style="font-size: 2em; border: 1px solid orange; border-radius: 50%; padding: 10px; display: inline-block;">55</span>            WISE-Point            32WSWPDOOP02A1  <a href="#">Redeem</a> </div>

Select your license quantity and click “**Confirm**”.

New Purchase – Select Your License Quantity  
Product Category: IoT PaaS Software Services, Industrial Cloud Solutions

DeviceOn Perpetual License (App on Microsoft Azure) WISE-Point  
32WSWPDOAZURA1 License: **50.00**

10 devices ▾

+ Microsoft Azure Infrastructure Cost (*monthly estimated USD\$ 194*)

Please carefully check the purchase details before continue.

**Confirm**

Confirm your license quantity and click “**Confirm Redeem**”.

Confirm Your License Quantity  
Product Category: IoT PaaS Software Services, Industrial Cloud Solutions

DeviceOn Perpetual License (App on Microsoft Azure) WISE-Point  
32WSWPDOAZURA1 License: **50.00**

10 devices

+ Microsoft Azure Infrastructure Cost (*monthly estimated USD\$ 194*)

No refund policy applied to this digital product. Please carefully check purchase details before confirming the redeem transaction.

**Previous** **Confirm Redeem**

To deploy DeviceOn to your Azure subscription, a set of Azure account information is required. You will be prompted to enter required information on the WISE-PaaS/Marketplace when you choose “Create Deployment”. WISE-PaaS Marketplace will use the provided information to automatically deploy DeviceOn to your Azure subscription. There are two methods to retrieve those parameters for your Azure subscription, please reference Section 7.2.

**Processing Your Purchase**  
 A license key will be sent to your email shortly.

**Would You Like To Deploy This App To Microsoft Azure?**  
 Please enter your Microsoft Azure information to initiate deployment  
*(don't have Azure account yet? contact us to create one)*

**Azure Subscription ID:**  
 **a**

**Application (client) ID:**  
 **b**

**Directory (tenant) ID:**  
 **c**

**Client Secret:**  
 **d**

How to retrieve my Azure account information?

---

**Done. Skip Deployment** **Create Deployment**

After deployment, you will receive a mail to get server information, including account, password and URL.

**ADVANTECH**

Dears,

**Thank you for purchasing WISE-PaaS/DeviceOn**

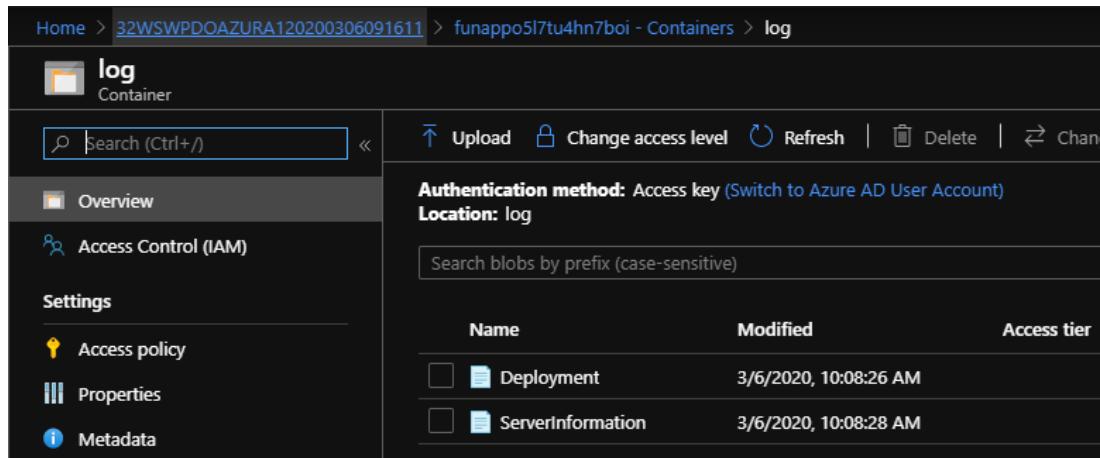
This letter informs you that the service has been deployed on Azure PaaS successfully with following access information.

● DeviceOn Portal: <a href="http://10.70.20.100">http://10.70.20.100</a>	● Grafana Dashboard Portal: <a href="http://10.70.20.170">http://10.70.20.170</a>
Username: <b>[REDACTED]</b>	Username: <b>[REDACTED]</b>
Password: <b>[REDACTED]</b>	Password: <b>[REDACTED]</b>

Note: Auto-activation apply to first-time purchase and deployment.  
 Product information and additional license purchase: <https://wise-paas.advantech.com/en-us/marketplace/detailinfo/52>

This is an automatically generated email, please do not reply.  
 Best regards,  
 Advantech WISE-PaaS Alliance

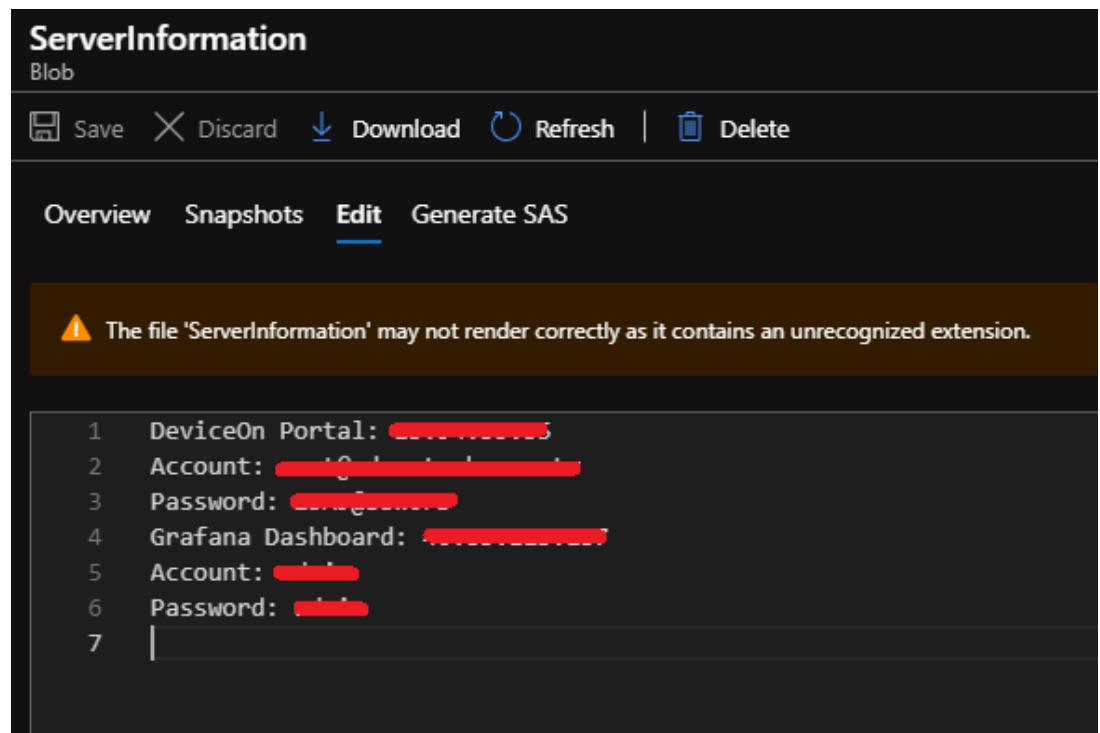
To prevent your mail blocked, we write the server information in Azure blob simultaneously.



The screenshot shows the WISE-PaaS Container log interface. The left sidebar has a 'log' icon and the word 'Container'. Below it are 'Overview', 'Access Control (IAM)', 'Settings' (with 'Access policy', 'Properties', and 'Metadata' options), and a search bar. The main area shows blob details with 'Authentication method: Access key (Switch to Azure AD User Account)' and 'Location: log'. A search bar below says 'Search blobs by prefix (case-sensitive)'. A table lists blobs:

Name	Modified	Access tier
Deployment	3/6/2020, 10:08:26 AM	
ServerInformation	3/6/2020, 10:08:28 AM	

The credential and access information also on the “ServerInformation”.



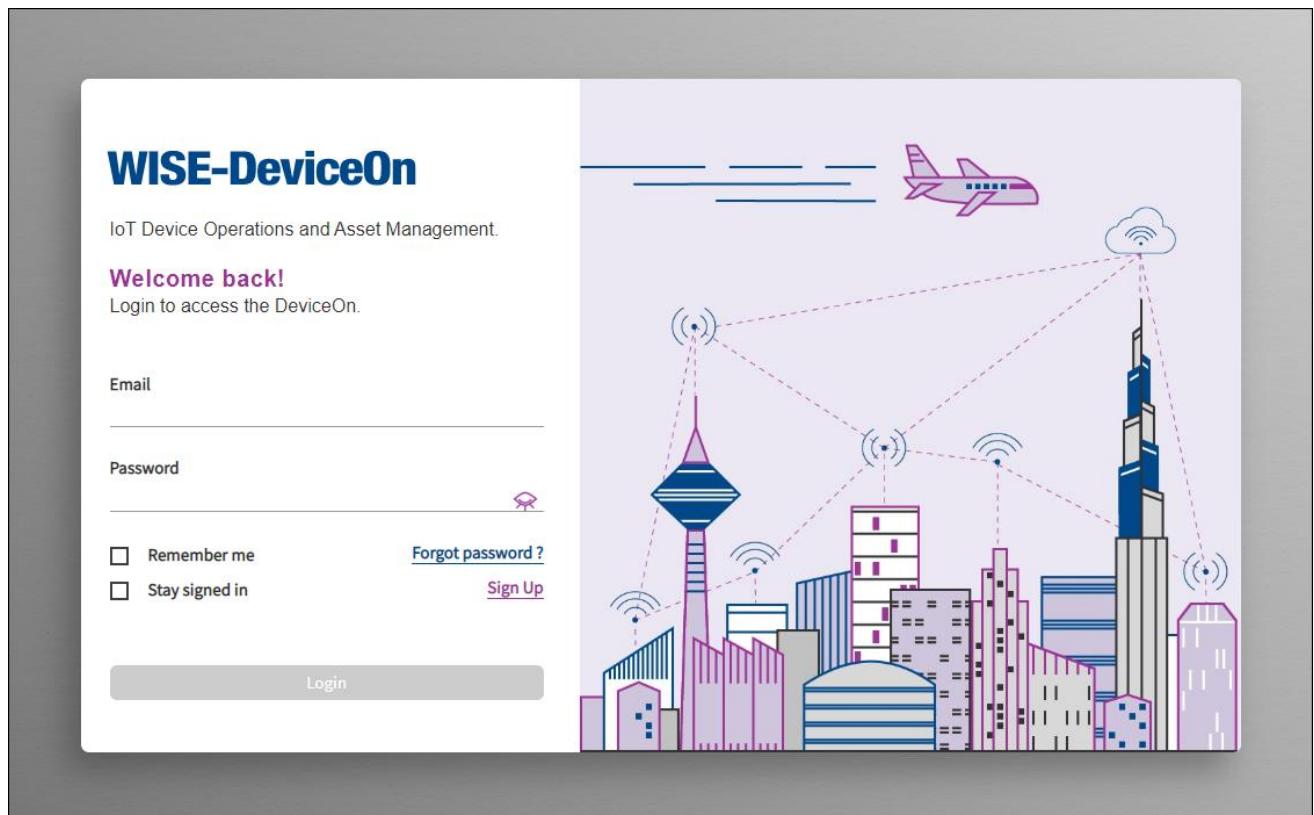
The screenshot shows the 'ServerInformation' blob content. The top navigation includes 'Save', 'Discard', 'Download', 'Refresh', and 'Delete'. Below is a menu with 'Overview', 'Snapshots', 'Edit' (which is underlined), and 'Generate SAS'. A warning message in a yellow box says: '⚠ The file 'ServerInformation' may not render correctly as it contains an unrecognized extension.' The main content area displays a list of items numbered 1 to 7, each followed by a redacted value:

- 1 DeviceOn Portal: [REDACTED]
- 2 Account: [REDACTED]
- 3 Password: [REDACTED]
- 4 Grafana Dashboard: [REDACTED]
- 5 Account: [REDACTED]
- 6 Password: [REDACTED]
- 7 [REDACTED]

## 2.2 DeviceOn Client Installation

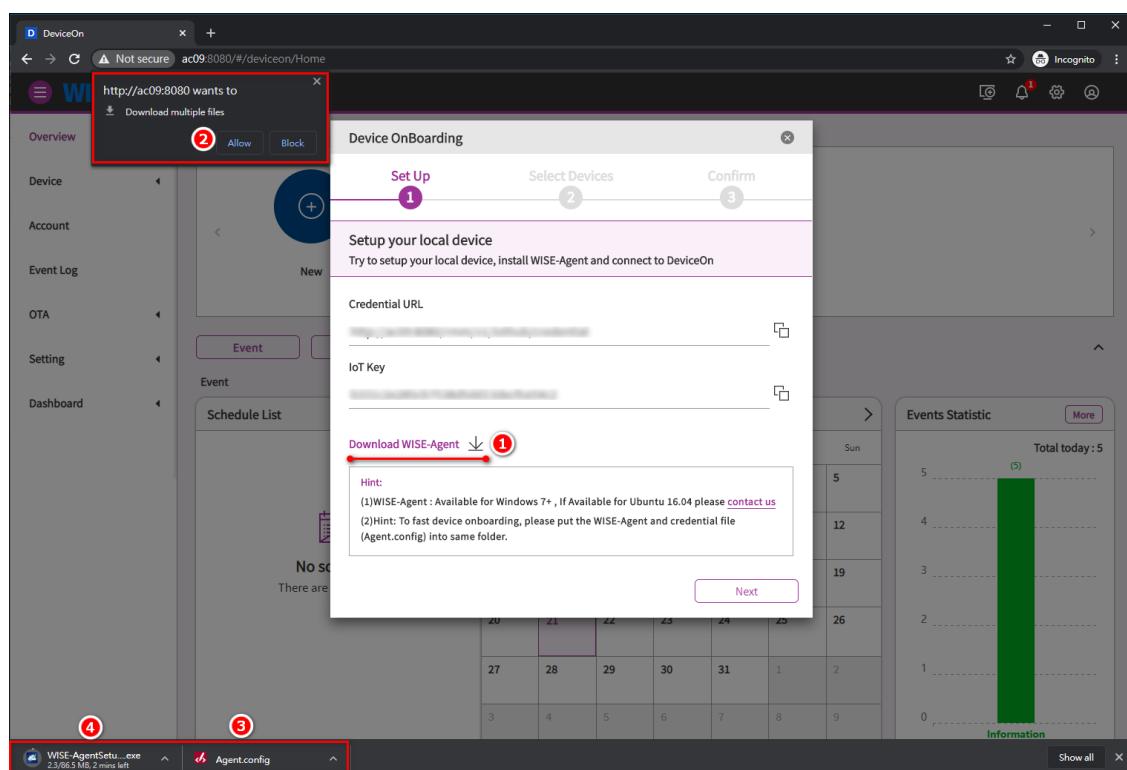
### 2.2.1 Setup Device Onboarding (Windows)

**Step 1: Log in to the DeviceOn Cloud Service with Your Account and Password**

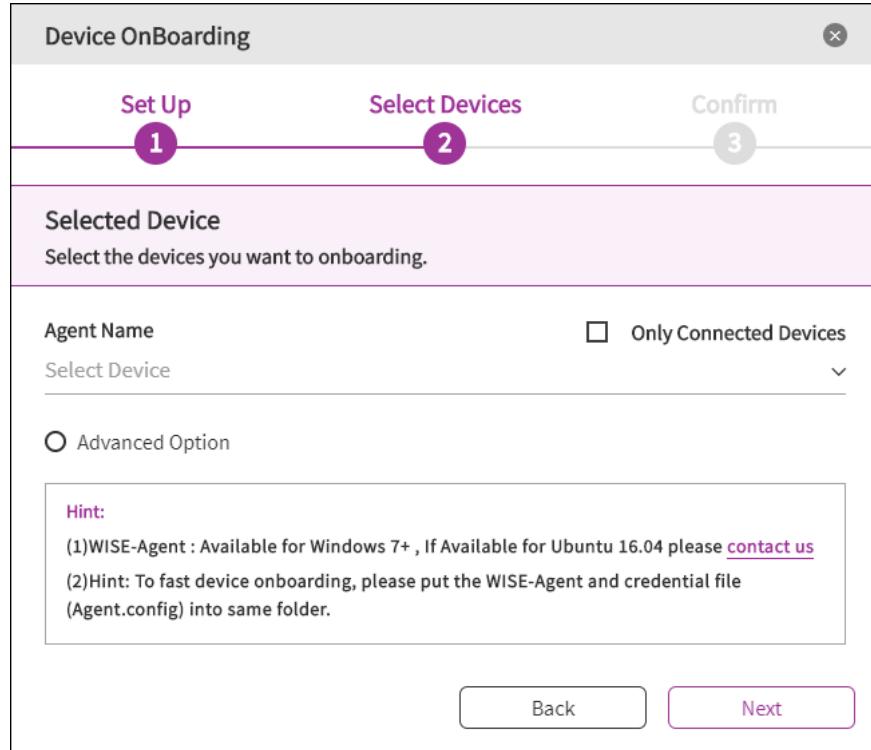


## Step 2: Download WISE-Agent and Connection Configuration (Agent.config)

At the first login, the “Device Onboarding” dialog will pop up automatically. Please click “Download” to get the latest version of **WISE-AgentSetup.exe** and the respective connection configuration. (**Agent.config**)

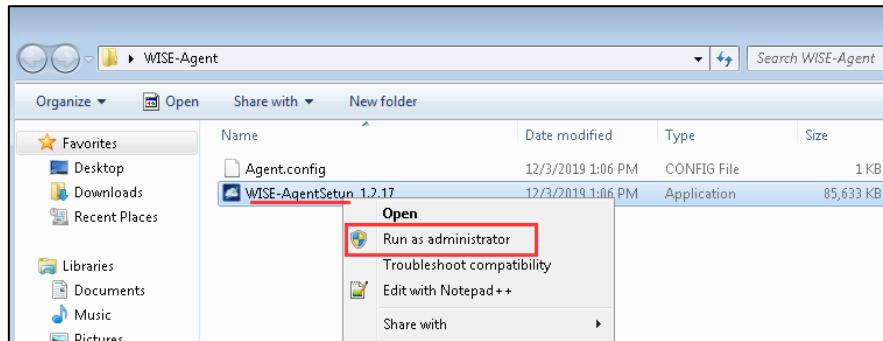


Click “**Next**” to wait for connecting devices.



### Step 3: Set up Your Local Device

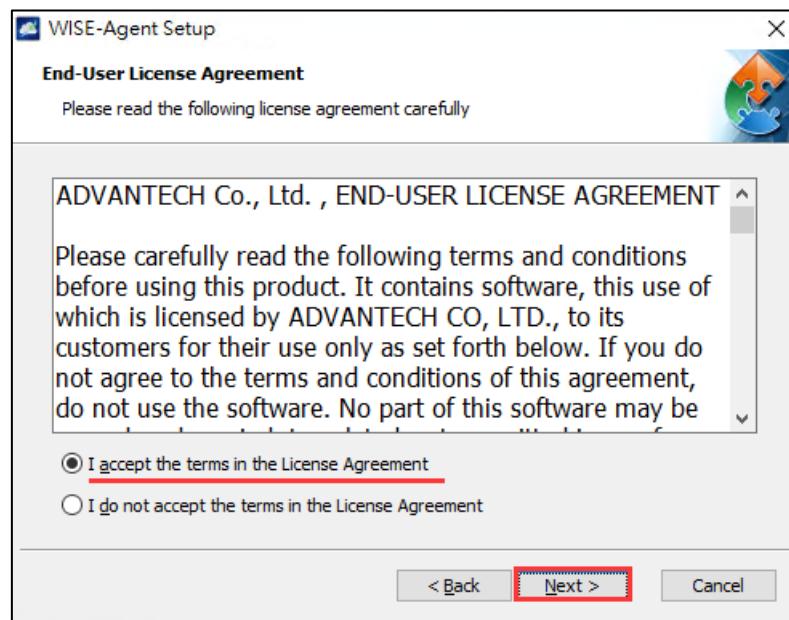
Copy those two files (**WISE-AgentSetup\_1.x.x.exe** and **Agent.config**) to the target device and launch “**WISE-AgentSetup\_1.x.x.exe**” as administrator.



Click “**Next**” to set up the WISE-Agent program.

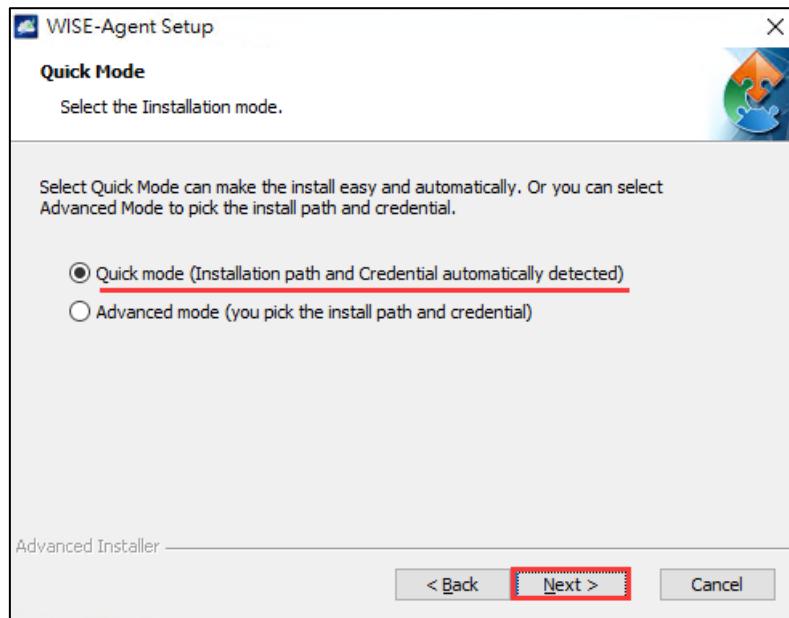


Select “I Accept the terms in the License Agreement” and click “Next”



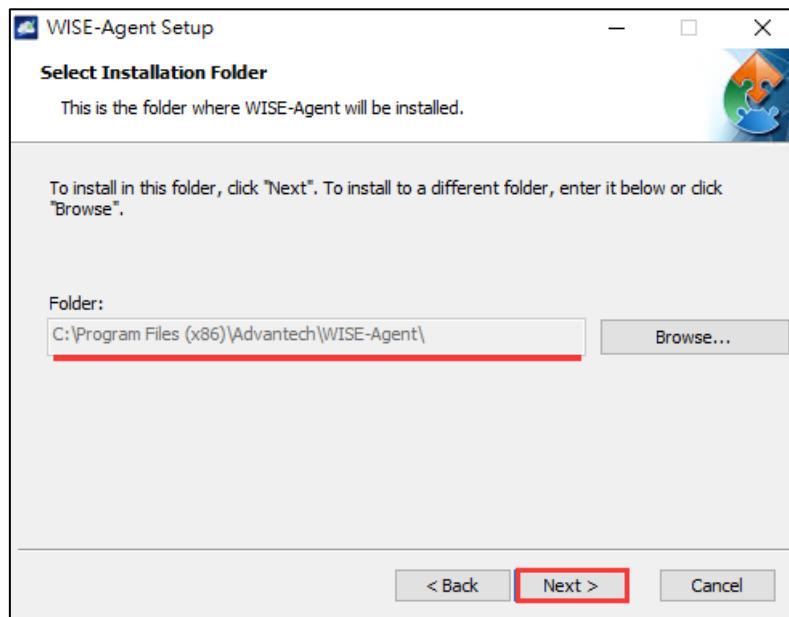
When the “WISE-AgentSetup\_1.x.x.exe” program detects a cloud connection configuration file (**Agent.config**) in the same folder, “**Quick Mode**” as shown in this dialog will be available. For “**Quick Mode**”, the installation path is fixed to “C:\Program Files (x86)\Advantech\WISE-Agent”. If you would like to adjust the installation location, please select “**Advanced Mode**”.

**Quick Mode:**



### Advanced Mode:

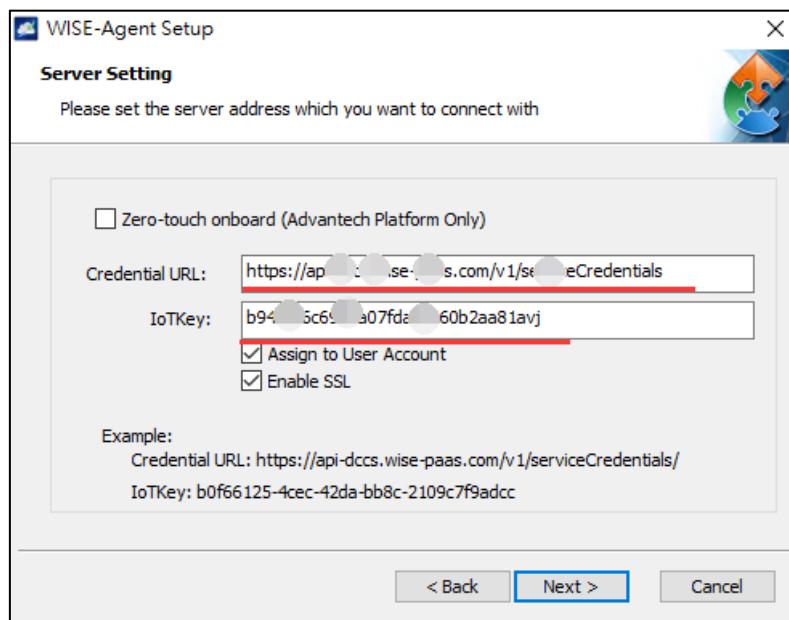
Select the Installation folder for WISE-Agent



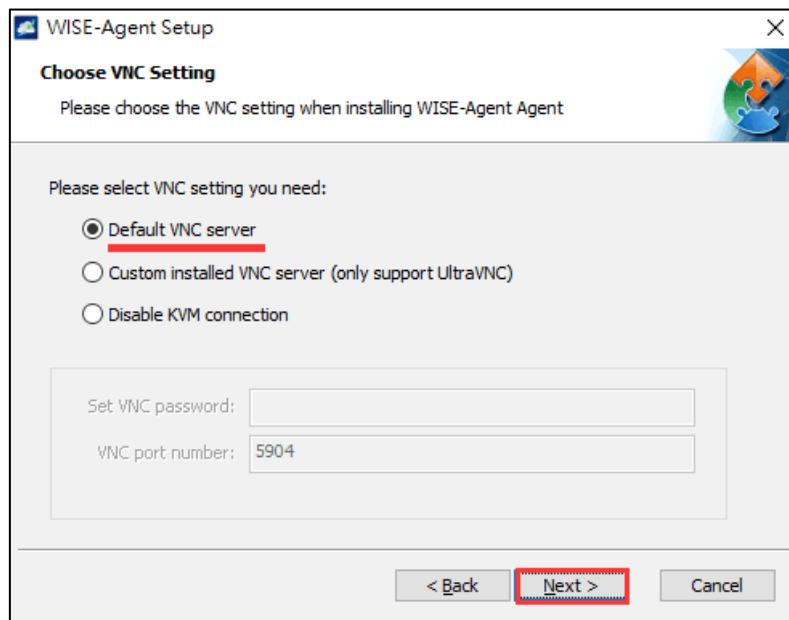
Set up the cloud connection configuration (**Credential URL & IoTKey**). This information can be retrieved from the DeviceOn web portal as shown in Step2, and click “**Next**”.

- “Zero-touch onboarding”: Only supported on Advantech platforms with SUSI Driver and pre-configuration on the provisioning server
- “Assign to User Account”: Each account has its own connection IoTKey. If checked, the device will be assigned to this account automatically.

- “Enable SSL”: The communication between WISE-Agent and DeviceOn Cloud is MQTT. If checked (default setting), all the messages and content are SSL encrypted (MQTT SSL port: 8883). Otherwise, port 1883 is used for MQTT without SSL.

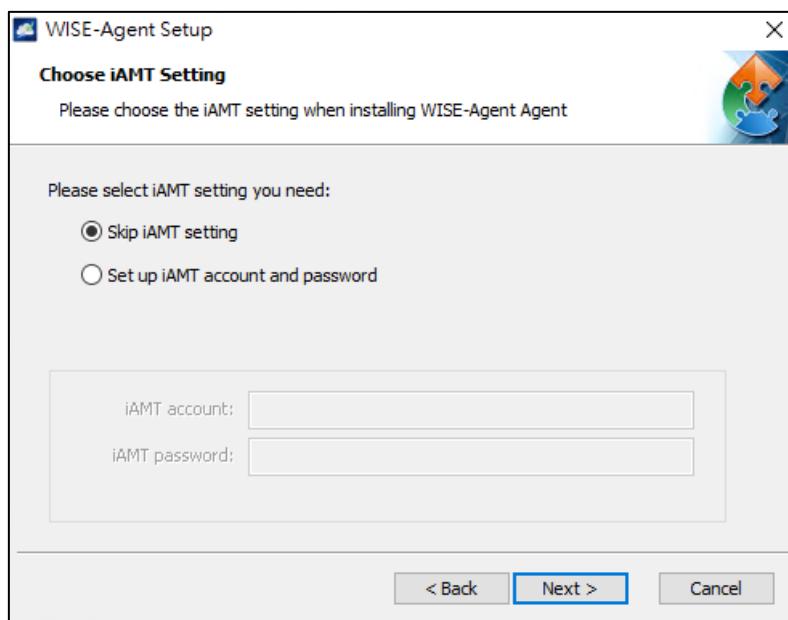


WISE-Agent supports remote desktop through built-in UltraVNC. You can manually specific the location of your own UltraVNC installation if preferred. If you do not want the remote desktop feature to be available, please select “Disable KVM Connection”.

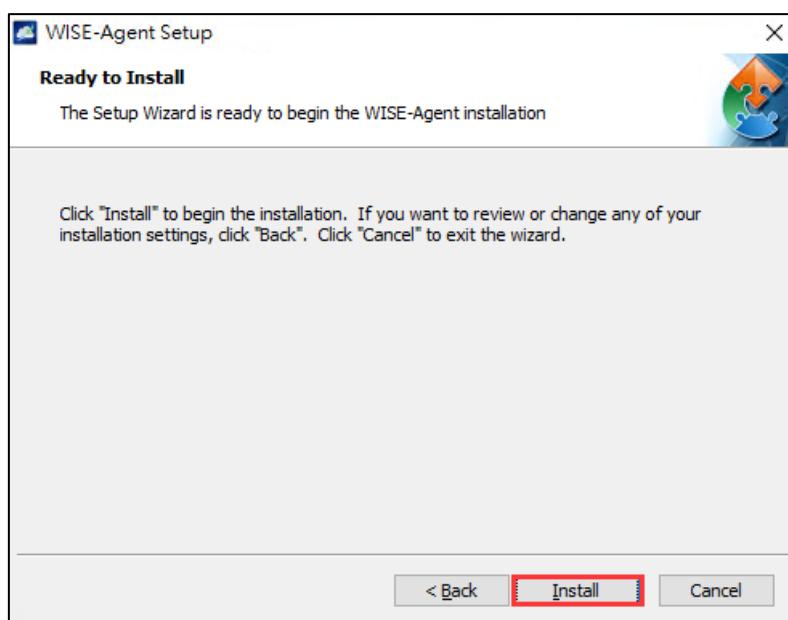


WISE-Agent integrates Intel AMT (Intel Active Management Technology) for remote power management (Power Up, Down, Cycle and Reset) as well as remote desktop access, even in case the operating system has crashed. However, this feature requires hardware support (Intel Core i5, i7) and the target device needs to be on the same local network as the DeviceOn server. Please

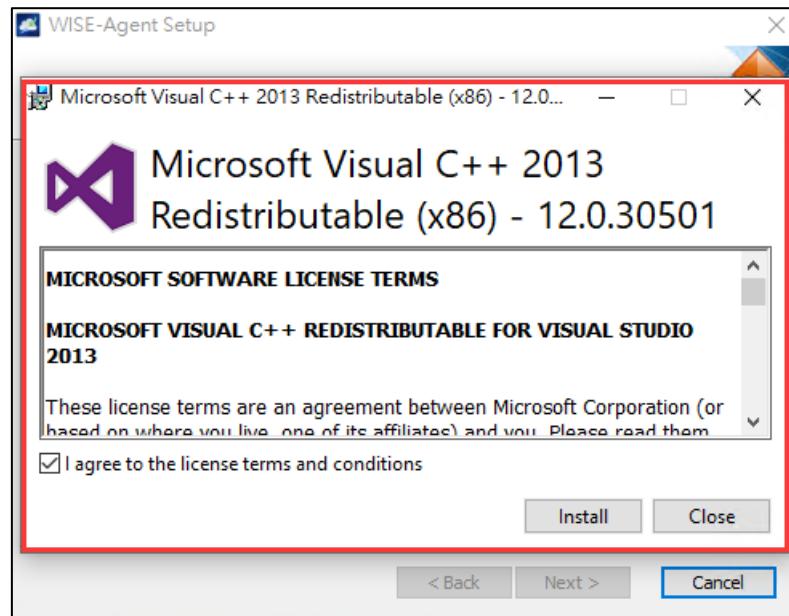
pre-configure iAMT, enable it in the device's BIOS and provide the account and password information in this dialog if you would like to enable iAMT based remote control features.



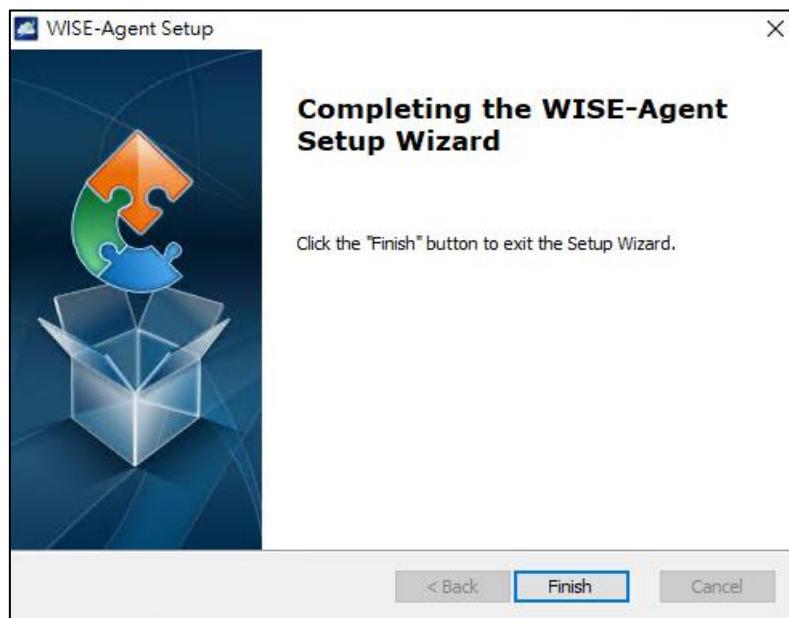
Click “Install” to begin the installation.



WISE-Agent requires the Microsoft Visual C++ Redistributable 2008, 2013, 2015 x86 packages, which will be downloaded from the Internet and set up during the installation process. If you are in an environment with limited or no Internet access, please download the "[Agent Dependency Package](#)" through an Internet connected device and install this package first.



Click “Finish” to exit the program.



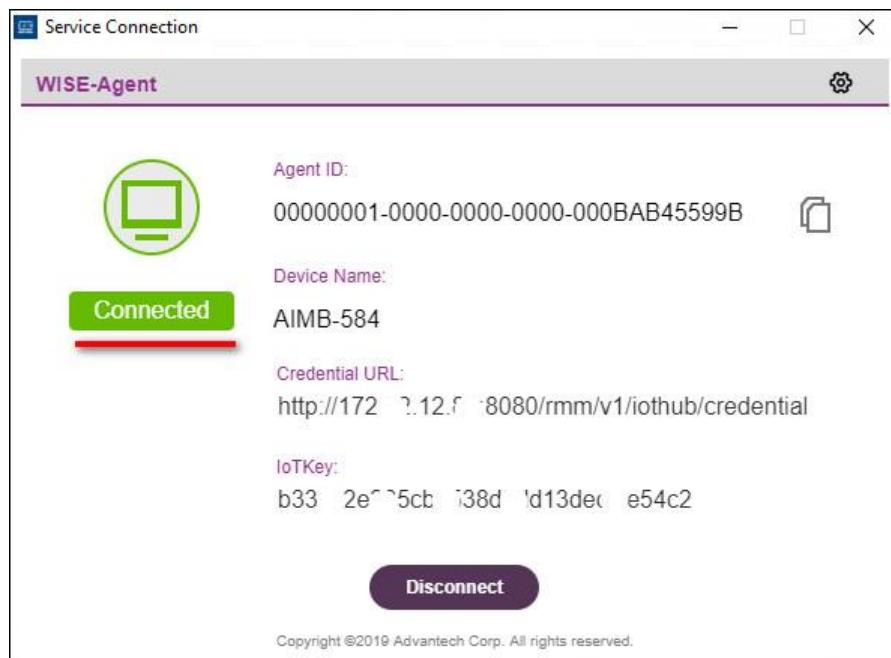
#### Step 4: Launch the WISE-Agent

Click on the “WISE-Agent” icon on the Windows Desktop to open the WISE-Agent user interface.



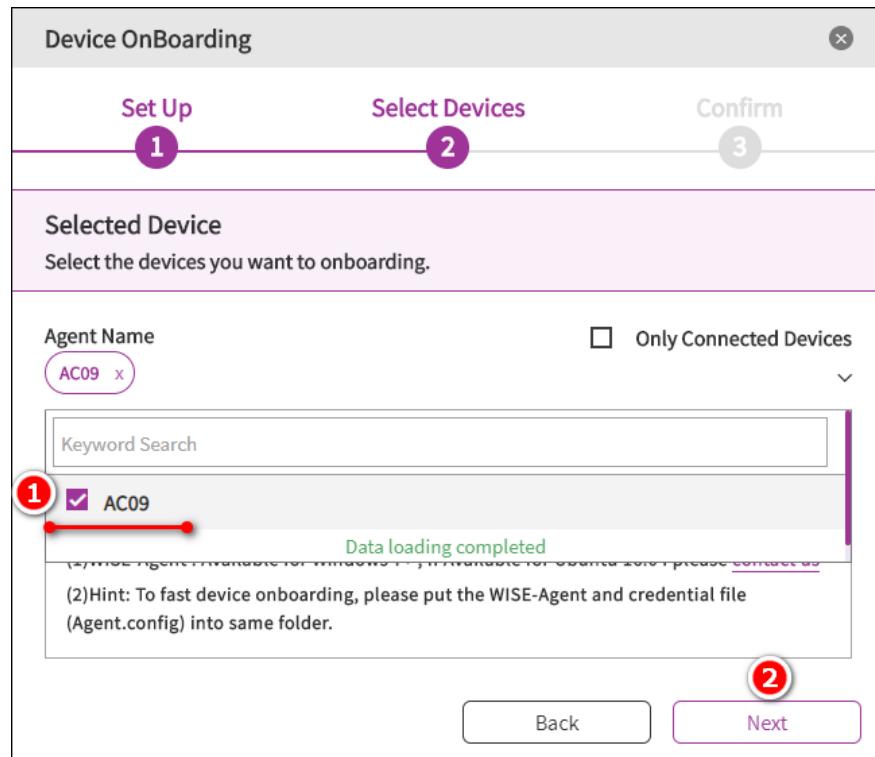
If the status shows “Disconnected”, please make sure your network settings are configured correctly and that you have access to the DeviceOn server-side application, either located in a public cloud

(WISE-PaaS, MS Azure) or on premise (standalone server version) depending on deployment scenario. Then, please click the “Connect” button to try to reconnect.

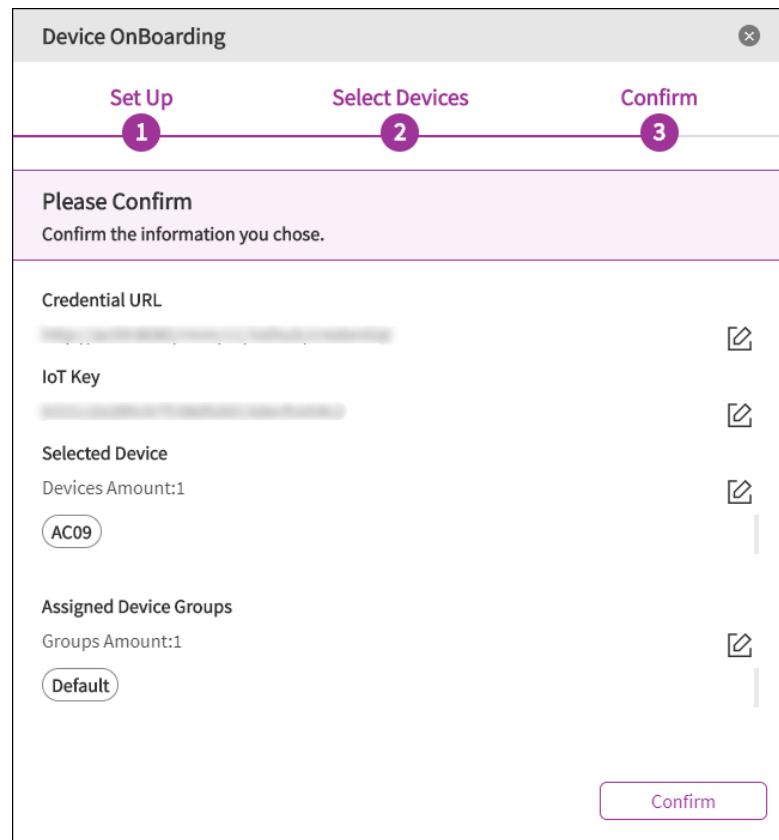


## Step 5: Grouping Your Devices

Once the device connects, go to the device grouping page, where the device group for these devices can be selected.



There is a “**Default**” group that can be used, or other groups for this device can be created after checking “Advanced options”. Click “**Confirm**” to start device management.



## Step 6: Start Device Management

By default, two “Real-time Actions” are created for a group, one is “Screenshot” and the other one is “Reboot”. The overview page further shows the online status of registered.

The screenshot shows the WISE-DeviceOn dashboard. The left sidebar includes links for Overview, Device, Account, Event Log, OTA, Setting, and Dashboard. The main area has a title 'Real-time Actions' with three icons: '+', 'Reboot', and 'Screenshot'. Below this is a 'Event' section with tabs for Event, System, Device Status, Device Rank, and Device Map. The 'Event' tab is selected, showing a 'Schedule List' with a note 'No schedule' and 'There are no schedule.' A calendar for '7Month 2020Year' is displayed, with days from 29 to 5 of the next month. To the right is an 'Events Statistic' chart showing 'Total today : 7' with a green bar reaching the value 7. The chart has a legend 'Information' at the bottom.

### 2.2.2 Setup Device Onboarding (Linux)

We also provide an installer for Ubuntu Linux (one of the most popular Linux distribution). This section will guide you how to install WISE-Agent on Ubuntu Linux. Note here that:

The WISE-Agent Ubuntu Linux installer is named something like "**wise-agent-Ubuntu 18.04 x86\_64-1.x.x.0.run**". To acquire the installer and ensure having the latest version, please contact us. If you are running the installer with an account other than "root", you should use "**sudo**" command to obtain higher privileges, or the installation may fail at any step.

#### Step 1: Open a terminal

The installer runs in CLI (Command Line Interface) mode. As such, open a terminal preferable for you.

#### Step 2: Copy the installer to target host

Use the way you like to copy the installer to the target host.

#### Step 3: Set the installer as executable

In the terminal, run "**chmod 0755 wise-agent-Ubuntu 18.04 x86\_64-1.x.x.0.run**" so that the installer as an executable file under Ubuntu Linux.

```
sephiroth@sephiroth-VirtualBox:~$ chmod 0755 wise-agent-Ubuntu\ 18.04\ x86_64-1.4.10.0.run  
sephiroth@sephiroth-VirtualBox:~$
```

#### Step 4: Running the installer

Change your working directory to where the installer is and run "**./wise-agent-Ubuntu 18.04 x86\_64-1.x.x.0.run**". You may need to run "**sudo ./wise-agent-Ubuntu 18.04 x86\_64-1.x.x.0.run**" to acquire higher privileges if you were logged in as a normal user.

```
sephiroth@sephiroth-VirtualBox:~$ sudo ./wise-agent-Ubuntu\ 18.04\ x86_64-1.4.10.0.run  
[sudo] password for sephiroth:  
Verifying archive integrity... 100% All good.  
Uncompressing The Installer for WISE-Agent 100%  
Install AgentService.  
/tmp/selfgz28285  
INFORMATION: Target device (Ubuntu 18.04) matched with (Ubuntu 18.04).  
Copy AgentService to /usr/local.  
'./AgentService' -> '/usr/local/AgentService'
```

#### Step 5: Start WISE-Agent and Connect to DeviceOn

Change your directory to **/usr/local/AgentService** and run **sudo ./setup.sh** to answer connection information, such as credential URL, IoTKey, Device Name and etc.

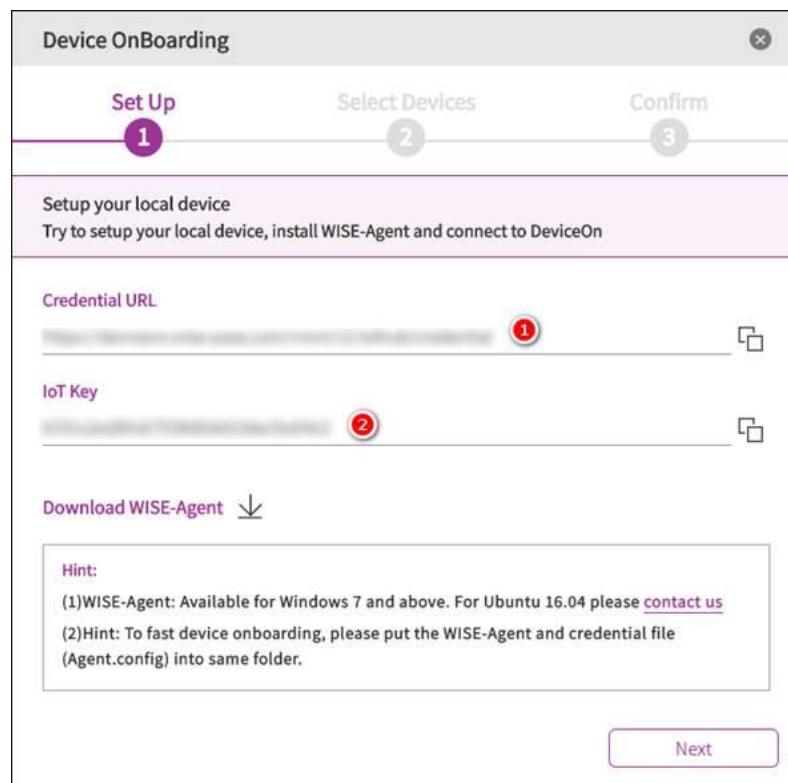
```

sephiroth@sephiroth-VirtualBox:~$ cd /usr/local/AgentService/
sephiroth@sephiroth-VirtualBox:/usr/local/AgentService$ sudo ./setup.sh
=====
***** AgentService Linux Setup ****=
*****
FireWall is disabled
Pid: 28496
find app dir /usr/local/AgentService.
AgentService Path: /usr/local/AgentService
sending request to stop AgentService
*****
Do you want to configure WISE-Agent now? [y/n](default: y)y
Zero-touch onboard [y/n](default: n): n
Input Credential API URL(default:https://api-dccs.wise-paas.com/v1/serviceCreden
tials()): 
Input IoT Key(default:):
Assign device to User Account [y/n](default: n): y
Enalbe TLS [y/n](default: n): y
Input Device Name[Len:4--35](default:sephiroth-VirtualBox):
Input AMT ID[Len:4--35, or na](default:):
Input AMT password[Len:8--16, or na](default:):
Select KVM Mode[0:default, 1:custom VNC, 2:disable](default:0):
Input VNC Port[1--65535](default :5900):
*****
Do you want to start WISE-Agent now? [y/n](default: y)

WISE-Agent Service Starting...
RMM Linux setup successfully!
sephiroth@sephiroth-VirtualBox:/usr/local/AgentService$ 

```

1. Zero-touch onboard is a zero-configuration and quick connection mode for a special purpose. The default is disabled (n).
2. Enter **Credential URL** and **IoT Key** that information could retrieve from the DeviceOn portal.



3. Assign device to User Account: You can bind the target device into a “Default” group in your account on the portal automatically
4. Enable TLS: Turn ON/OFF the TLS/SSL mode.
5. Input Device Name: Give your device name and show it on the portal.
6. Input AMT ID and password: If your device support Intel AMT, please enter AMT ID and Password to enable these functions.
7. Select KVM Mode [0:**default**, 1:**Custom VNC**, 2:**disable**]: User can use our default VNC to support the Remote Desktop function by entering 0 and give a listen port if you don't want to use the default port. Second, select **Custom Mode**, if they already have a VNC server by entering 1 and provide the listen port and password. To disable the KVM function by entering 2.

When you run into this step the question shows like above, device is connected and under your account.

### 3. DeviceOn User Interface & Functions

#### 3.1 DeviceOn Server (Standalone)

The standalone version provides all packages of the DeviceOn software in one installer package, including RabbitMQ as a message broker, MongoDB, PostgreSQL as databases, Grafana for visualization, Tomcat for web services, and a watchdog service that protects DeviceOn core components from crashing or becoming unresponsive.

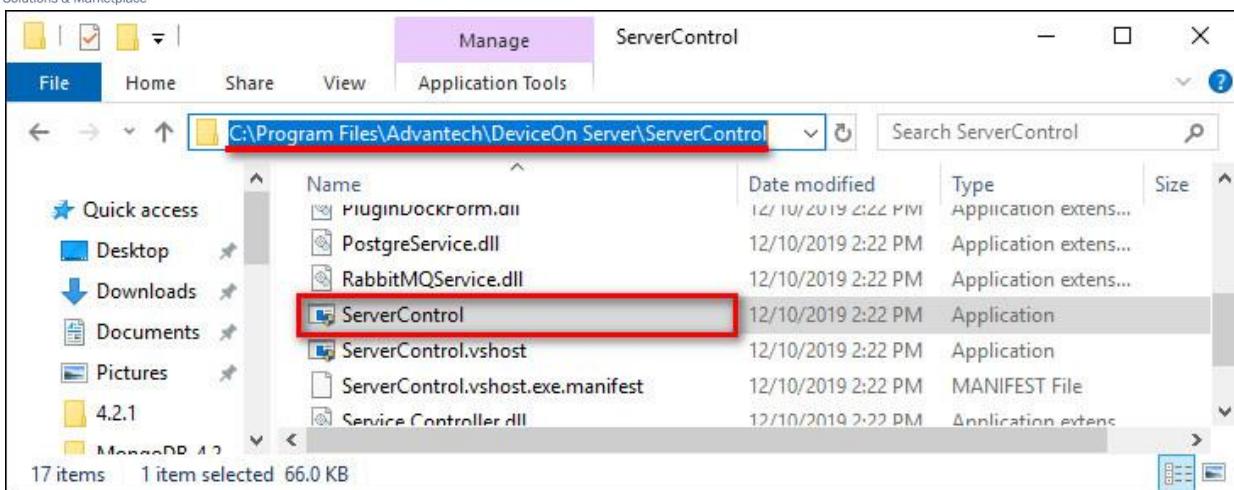
The following section (3.1.1) introduces the “Standalone Server Control” tool that allows to monitor and enable/disable DeviceOn core components. The watchdog service is explained in section 3.1.2.

##### 3.1.1 Standalone Server Control

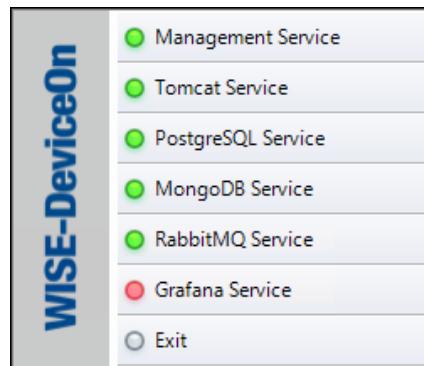
After the DeviceOn standalone version has been installed, a “**Server Control**” icon should show up in the system tray.



If it does not show up for some reason, please go to installation path and launch the program (ServerControl.exe) manually as shown here:

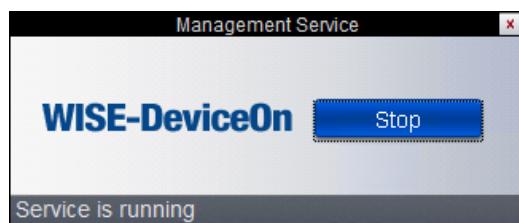


Right click on the tray icon to bring up an overview of each core component status. The green light indicates normal status and a red light means the respective service is stopped.



- Management Service

The “Management” service includes the DeviceOn backend core function and consists of two Java processes (DeviceOn and Provisioning Worker) that handle messages and process OTA traffic between the client and server. Click “Stop” to stop the management service.

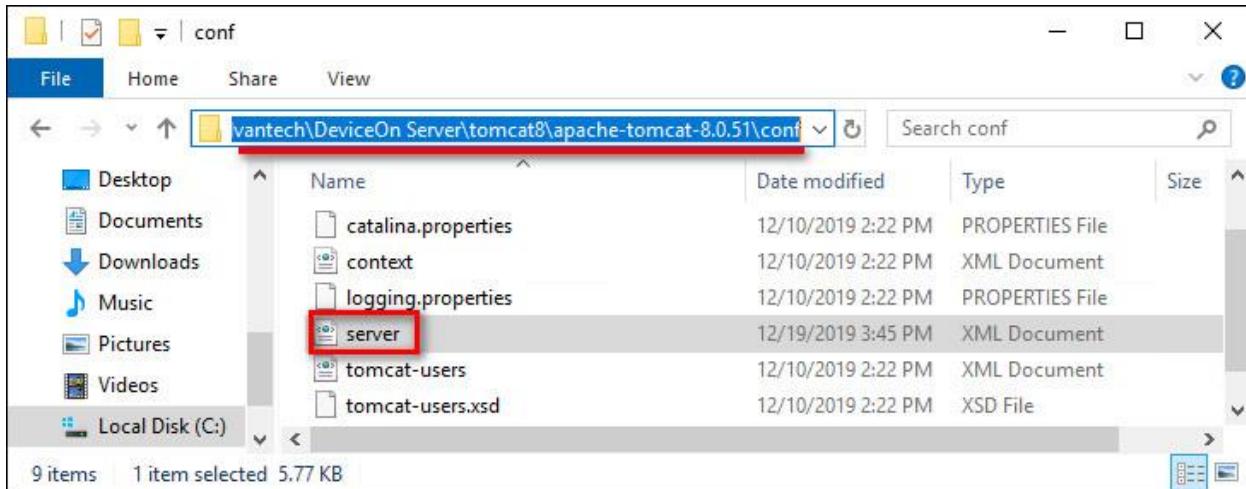


- Tomcat Service

The DeviceOn web service uses Apache Tomcat to provide the user interface, APIs and WebSockets. Click “Stop” to stop the Apache Tomcat service.



For advanced configuration (SSL, connection pool, etc.), you may modify “**server.xml**” located in the installation folder.

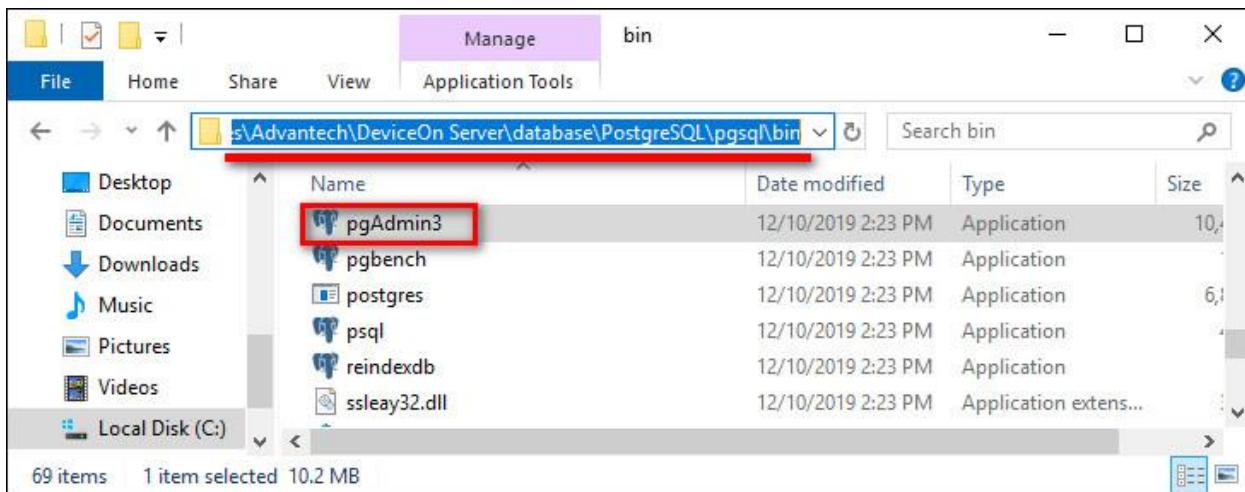


- PostgreSQL Service

The relational database (PostgreSQL) is used to store account, device, map, permission data etc. Click “**Stop**” to stop the PostgreSQL service.



A GUI tool called “**pgAdmin3.exe**” providing access to the PostgreSQL database comes with the PostgreSQL installation and is located in the installation folder as shown below. The default account is “**postgres**” and the password is the one you defined during the installation. We recommend you do not delete/edit any schema, table or data , since DeviceOn might stop to work if data is corrupt or missing.



- MongoDB Service

To process sensor data from client devices, DeviceOn leverages MongoDB to provide better performance and compression rates than relational databases. Click “**Stop**” to stop the MongoDB service.



- RabbitMQ Service

RabbitMQ is one of the most popular open source message brokers, and is used as “IoTHub” to exchange messages between the server and client devices. Click “**Stop**” to stop the RabbitMQ service.



- Grafana Service

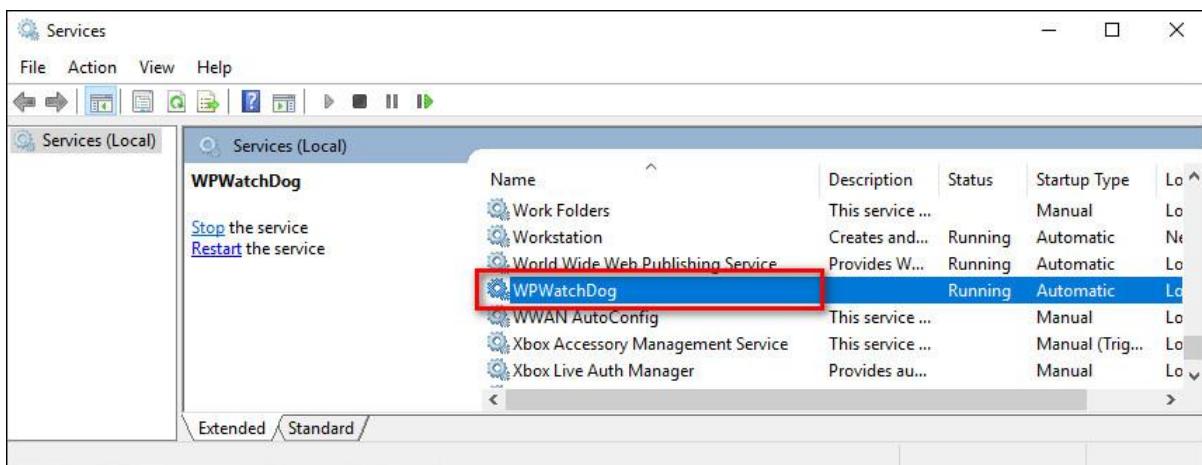
Grafana is a popular framework that allows you to query, visualize and alert on data from various data sources. DeviceOn supports a simple JSON API that can be used as data source in Grafana, effectively making all DeviceOn data available to Grafana. Click “**Stop**” to stop the Grafana service.



### 3.1.2 Background Watchdog Service

- Watchdog Service

There is a Watchdog service (WP) that monitors the management service (DeviceOn and Provisioning Worker) and ensures all the functions work as expected.



## 3.2 DeviceOn WISE-Agent

WISE-Agent runs as Windows service, so even without any user logged in, WISE-Agent will establish a connection to the DeviceOn server and the most of the features are supported. Section 3.2.1 explains how to use the WISE-Agent user interface to verify the current connection status and retrieve basic information of the client device. There is another Watchdog service monitoring the WISE-Agent client in order to avoid impact due to crashed or hanging processes.

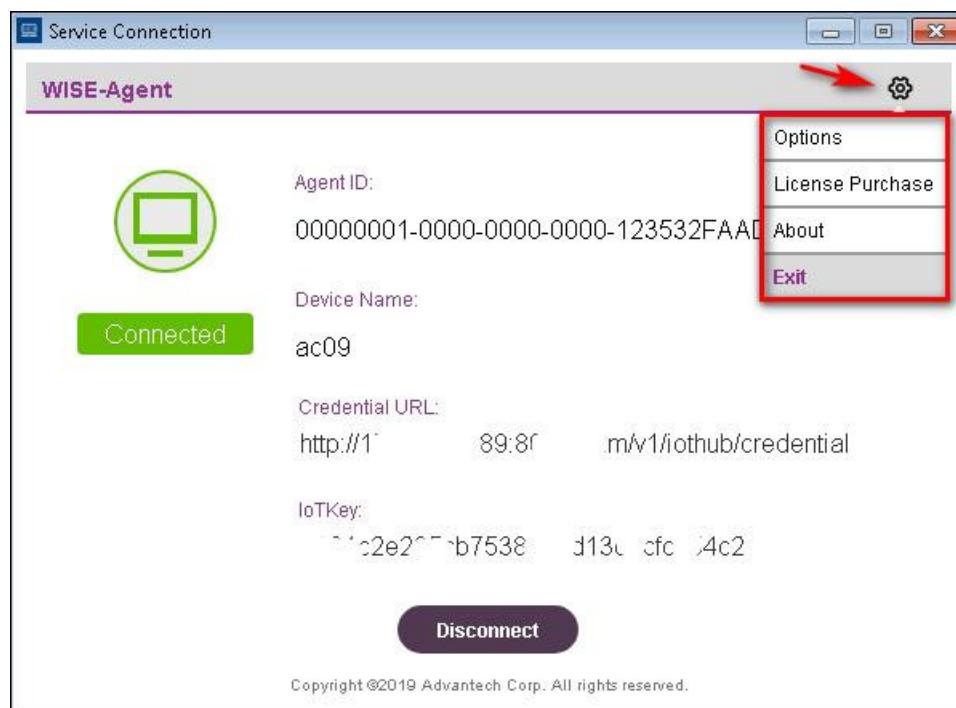
### 3.2.1 WISE-Agent Connection

If you followed the instructions to set up WISE-Agent and connect to the DeviceOn server/cloud, there should be a WISE-Agent shortcut on your desktop. If not, please refer to Section 2.3 to install WISE-Agent. After launching the WISE-Agent user interface, it will provide an overview of the connection status, device information (AgentID, Device Name) as well as connection credentials (Credential URL, IoTKey).



- **Agent ID:** Device unique ID - the default is 32 characters, prefix (20 characters) and MAC address (12 Characters)
- **Device Name:** Device name as shown on the DeviceOn server
- **Credential URL:** Connection URL, used to authenticate to DeviceOn Server
- **IoTKey:** Connection Key - each DeviceOn client has a unique key that will be used to establish the MQTT session
- **Disconnect:** To stop the device connection and data transmission, you can click “Disconnect” to stop the WISE-Agent service

If you would like to adjust the device name or connection parameters, please click the “Settings” icon on the top right and select “**Options**”.



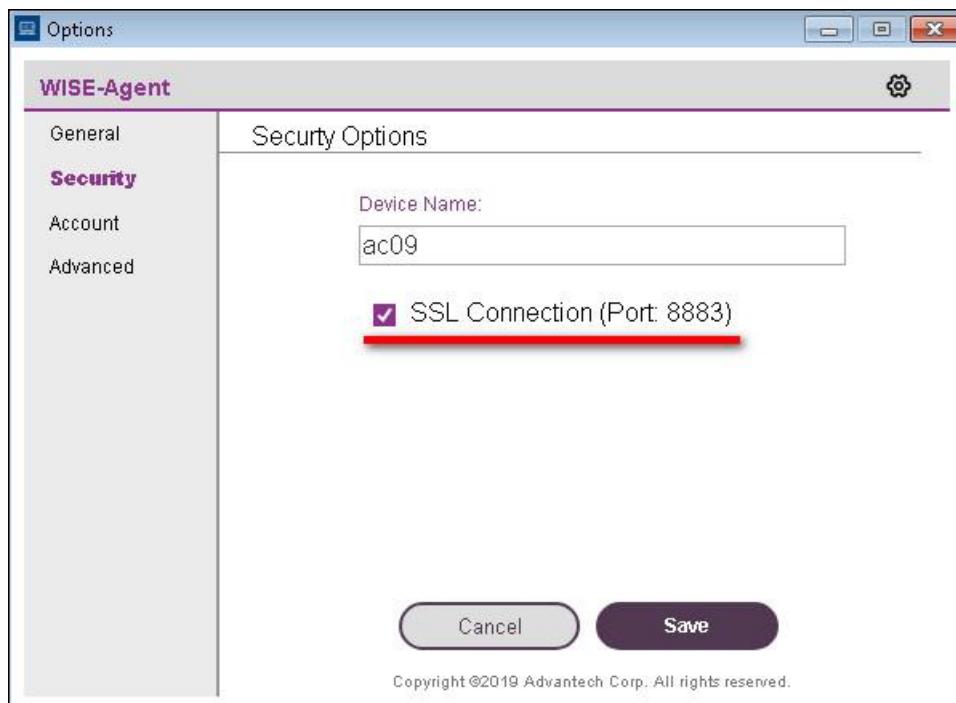
- Option -> General

This overview page provides information about “Device Name”, “Operating System” (Windows 7, 8, 10), “MAC Address” of the client, “Memory Capacity” and version of the Advantech SUSI Driver (if applicable). The version of the “Operating System” represents the [Windows kernel version](#). If the client device is an Advantech platform that is supported by SUSI, we recommend to download the latest SUSI driver from the [Advantech Support](#) site first. Please click [here](#) to obtain the latest driver version.



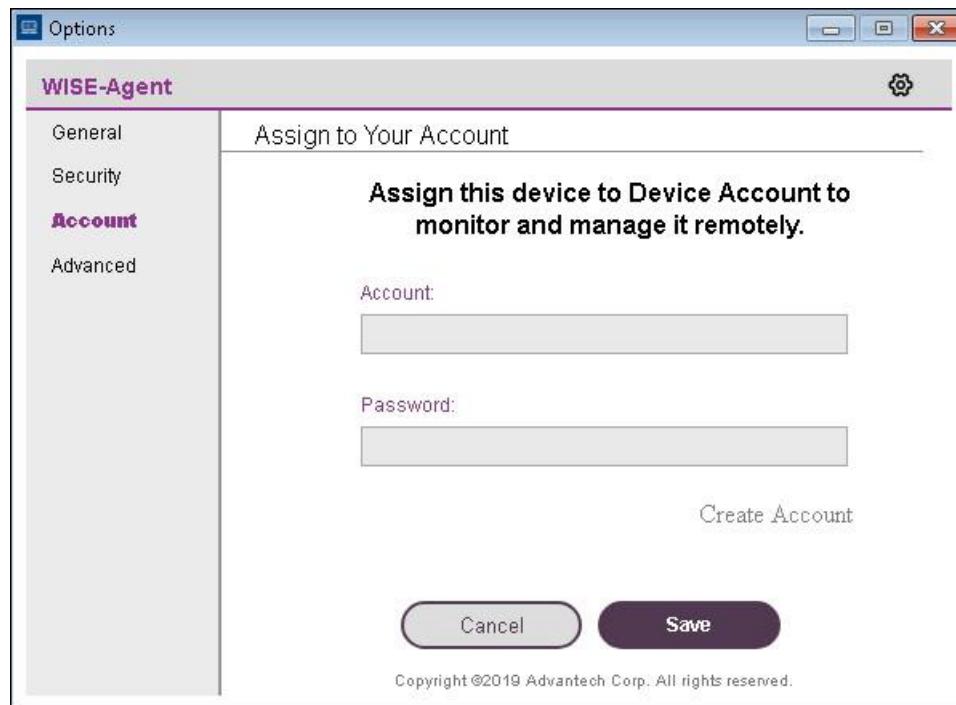
- Option -> **Security**

The communication protocol used for message exchange between the server and client is MQTT, an industry standard lightweight messaging protocol for small sensors and mobile devices. WISE-Agent provides the option to use MQTT with SSL encryption on port 8883, or MQTT without SSL on port 1883.



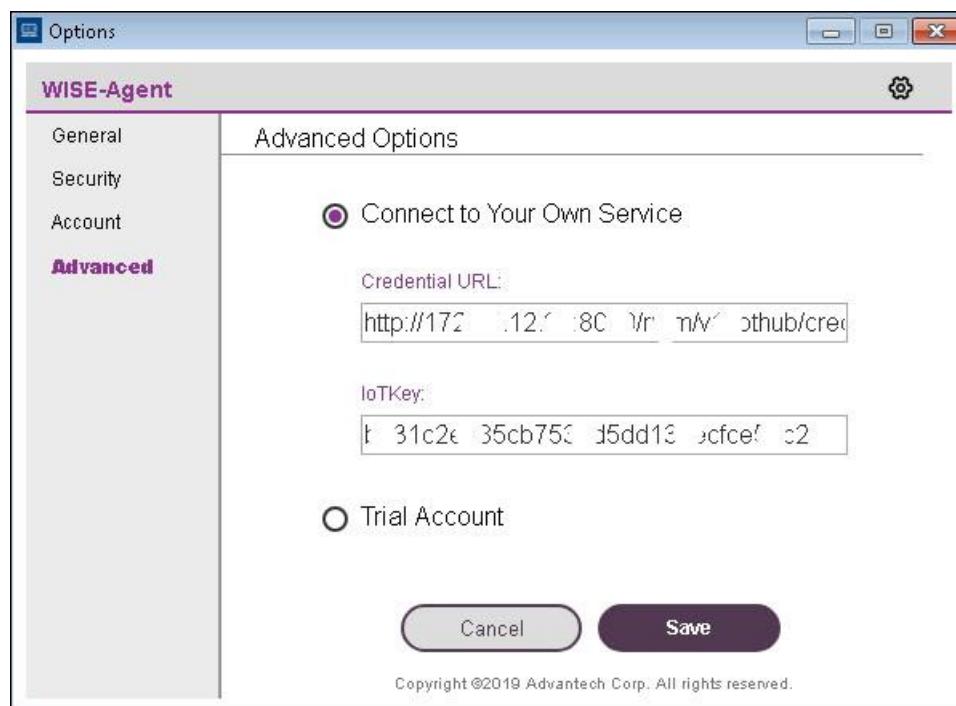
- Option -> **Account**

You can register on the DeviceOn trial site (<https://deviceonapp.wise-paas.com>) for a six-month trial account and use it with your device. Before you can create a trial account or enter trial account information, please go to the “Advanced” tab and select “Trial Account”.



- Option -> **Advanced**

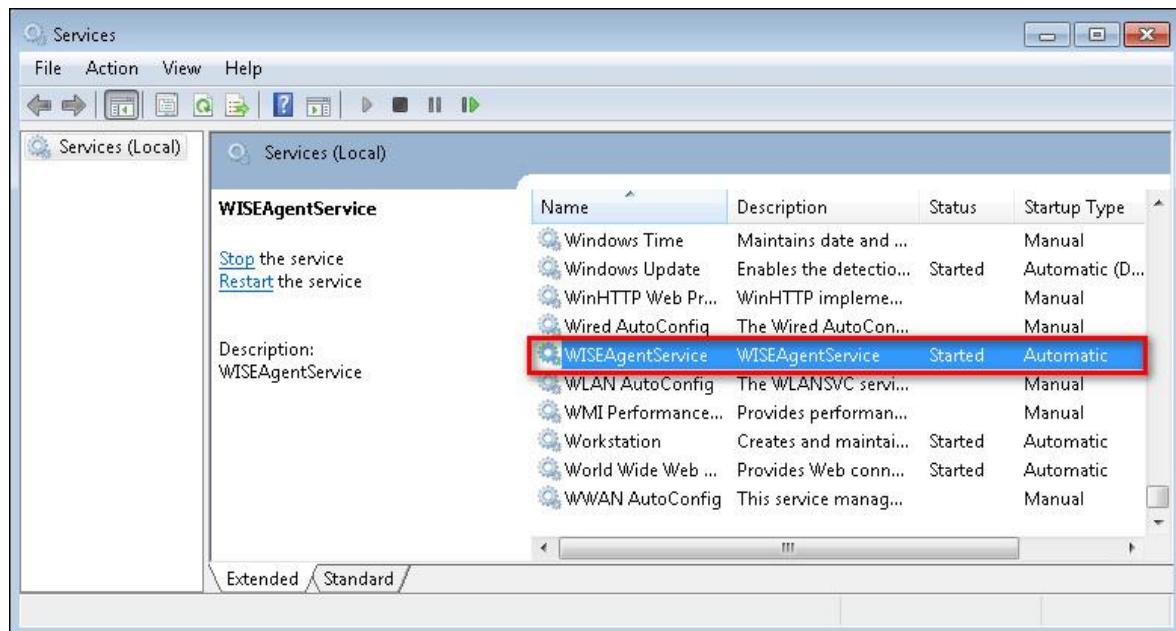
Under the “Advanced” tab, you can select whether to connect to a DeviceOn server/cloud service, or whether to connect to the DeviceOn trial site (<https://deviceonapp.wise-paas.com/>). In case of trial site, you need to enter account information under the “Account” tab (see previous step) while for a regular DeviceOn server or cloud service, you need to enter the “Credential URL” and “IoT Key” here. Refer to “Step 2” in Section 2.3 on information how to obtain those.



### 3.2.2 WISE-Agent Services

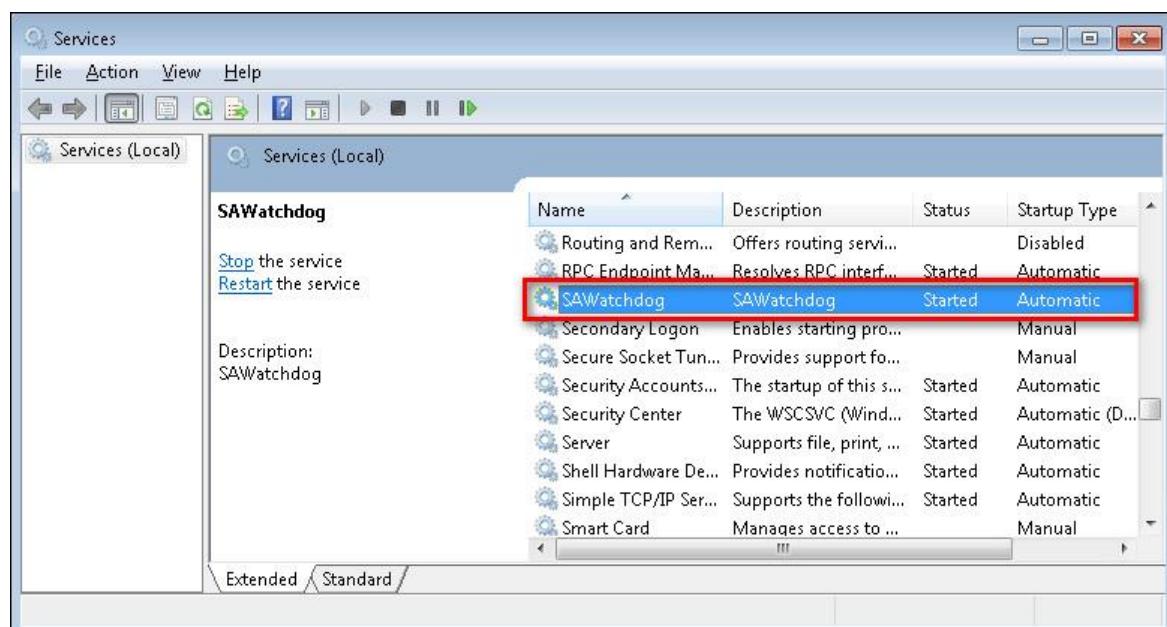
- Main Service

“WISEAgentService” is the main services that connects to the DeviceOn server/cloud service. The service is set to start automatically by default.



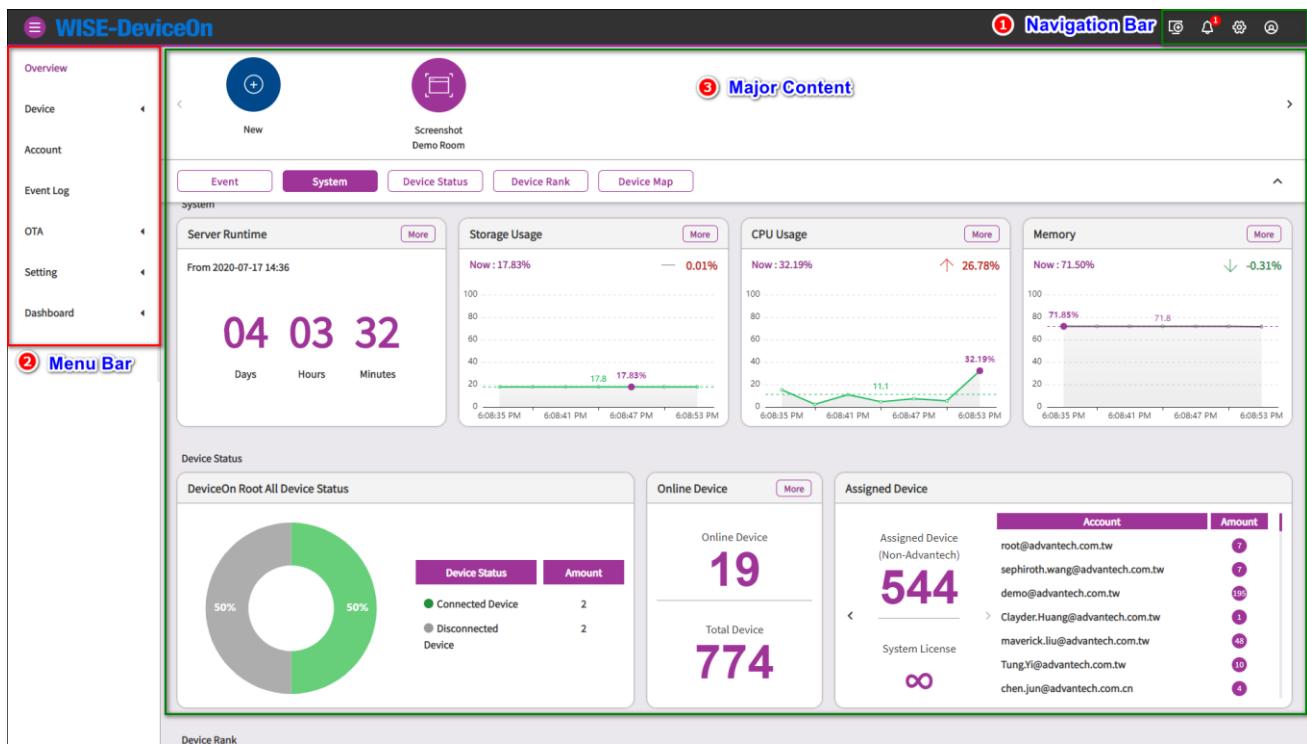
- Watchdog Service

The “SAWatchdog” service is a basic watchdog governing “WISEAgentService” in order to ensure service quality.



## 3.3 DeviceOn User Interface

The DeviceOn web interface is based on the VUE framework and leverages the [Vuestic Admin](#) template. The user interface is divided into three main parts - the navigation bar at the top, the menu bar at the left and the main content in the center with.



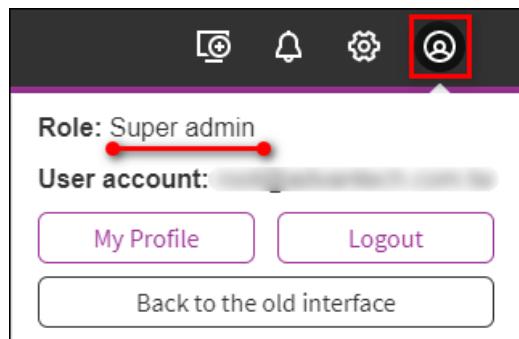
The screenshot illustrates the DeviceOn web interface layout:

- Navigation Bar:** Located at the top right, featuring icons for search, notifications, and settings.
- Menu Bar:** Located on the far left, listing categories: Overview, Device, Account, Event Log, OTA, Setting, and Dashboard. The "Device" item is highlighted with a red box.
- Major Content:** The central area displays various metrics and charts. It includes a clock showing 04 03 32, a "Storage Usage" chart (Now: 17.83%, 0.01%), a "CPU Usage" chart (Now: 32.19%, 26.78%), and a "Memory" chart (Now: 71.50%, -0.31%). Below these are sections for "Device Status" (DeviceOn Root All Device Status: 50% Connected Device, 50% Disconnected Device), "Online Device" (19 Online Device, 774 Total Device), and "Assigned Device" (544 Assigned Device (Non-Advantech), System License: infinity).

## Navigation Bar:

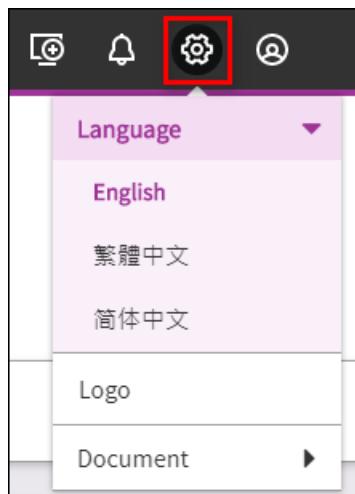
- Account Information

Click the account icon to show the currently logged in account and respective role. For more information, click “My Profile” to open the account page. (Menu Bar -> Account). Click “Logout” to log out from DeviceOn and remove personal information like cookies or tokens.



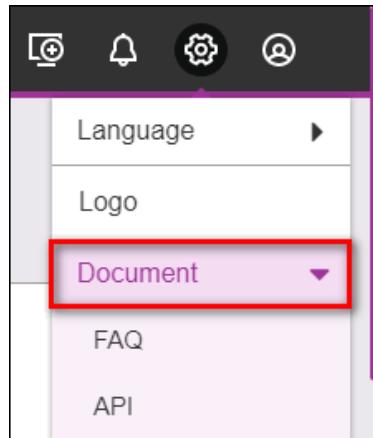
- Languages

DeviceOn supports multiple languages that can be changed by clicking the globe icon in the navigation bar. Currently there are three languages to choose from: English, Traditional Chinese and Simplified Chinese.

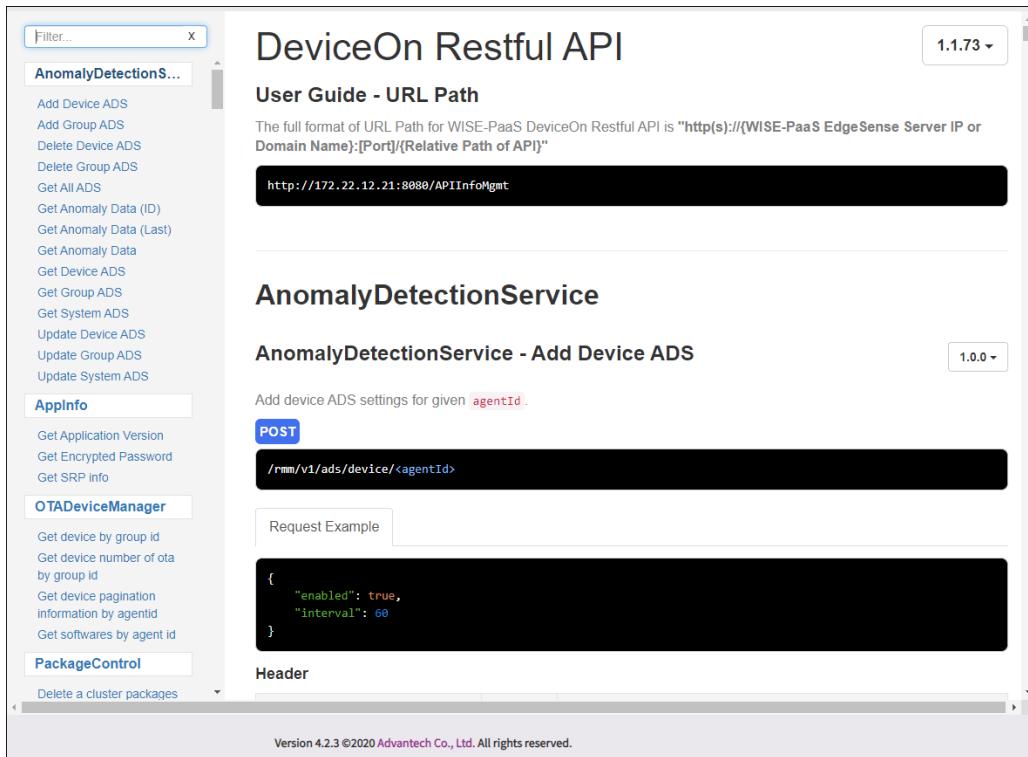


- Document (FAQ & API Reference)

There are two documents on DeviceOn user interface, one is Restful APIs and another is FAQ that including technical and general questions.



DeviceOn provide hundreds of API for App engineer to build up their AIoT solution, through the APIs to get account, map, device data, and remote diagnostic on devices. The API document is generated by APIDoc, includes API method, request, response, header and testing.

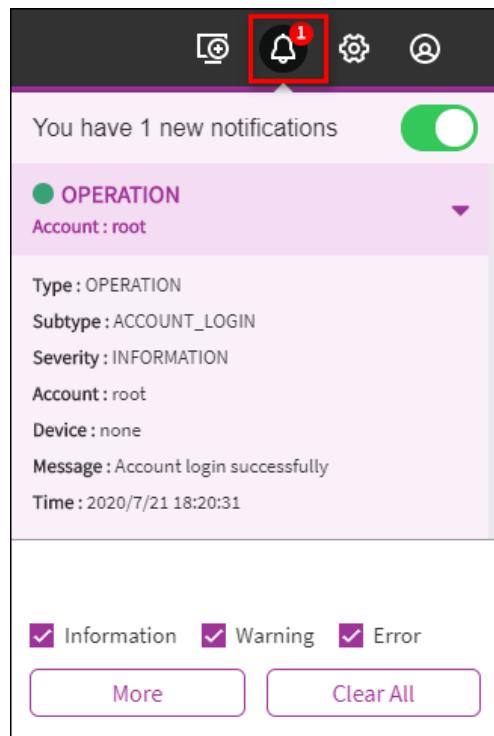


The screenshot shows the WISE-PaaS DeviceOn Restful API User Guide. The main title is "DeviceOn Restful API". On the left, there's a sidebar with a "Filter..." search bar and a list of service categories: "AnomalyDetectionService", "AppInfo", "OTADeviceManager", and "PackageControl". Under "AnomalyDetectionService", there's a sub-list of actions: "Add Device ADS", "Add Group ADS", "Delete Device ADS", "Delete Group ADS", "Get All ADS", "Get Anomaly Data (ID)", "Get Anomaly Data (Last)", "Get Anomaly Data", "Get Device ADS", "Get Group ADS", "Get System ADS", "Update Device ADS", "Update Group ADS", and "Update System ADS". The main content area has a header "User Guide - URL Path" with a note about the full URL format: "http(s)://[WISE-PaaS EdgeSense Server IP or Domain Name]:[Port]/[Relative Path of API]". Below this is a code snippet: "http://172.22.12.21:8080/APIInfoMgmt". A section titled "AnomalyDetectionService" contains a sub-section "AnomalyDetectionService - Add Device ADS" with a version "1.0.0". It describes adding device ADS settings for a given agentId via a POST request to "/v1/ads/device/<agentId>". It includes a "Request Example" with a JSON payload: {"enabled": true, "interval": 60}. A "Header" section is also present. At the bottom of the page is a footer: "Version 4.2.3 ©2020 Advantech Co., Ltd. All rights reserved."

Actually, the developer could design a plugin on WISE-Agent to aggregate edge data (Reference Section 5.1), and get these data via Restful APIs, visualize on Grafana Dashboard (Reference Section 4.4) or develop a UI plugin to customize. (Reference 5.2)

### ● Notification

If there are any active notifications, the number of event log messages is shown on the notification icon. Click the notification icon to see the event message summary. Three levels of events are supported: “**Information**”, “**Warning**” and “**Error**”, and the user can select which type of events should be shown on the user interface. For example, clicking the “**Unsubscribe Notification**” would disable any events in the screenshot shown below. Please note that after disabling events, the UI will not refresh automatically but needs to be refreshed manually. Click “**More**” to open the event log page (**Menu Bar -> Event Log**)



### ● Device Onboarding

To onboard devices, click the onboarding icon in order to download the WISE-Agent installer and in order to look up the required connection credentials. For more details on onboarding, please refer to Section 2.2.

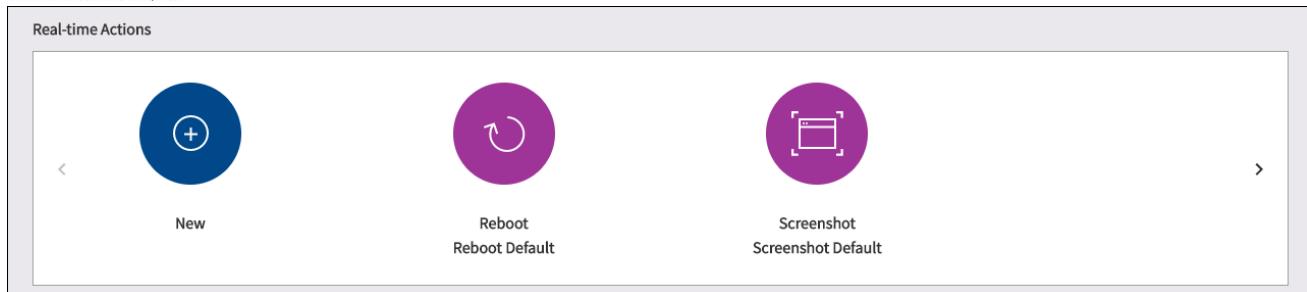


#### 3.3.1 DeviceOn Overview

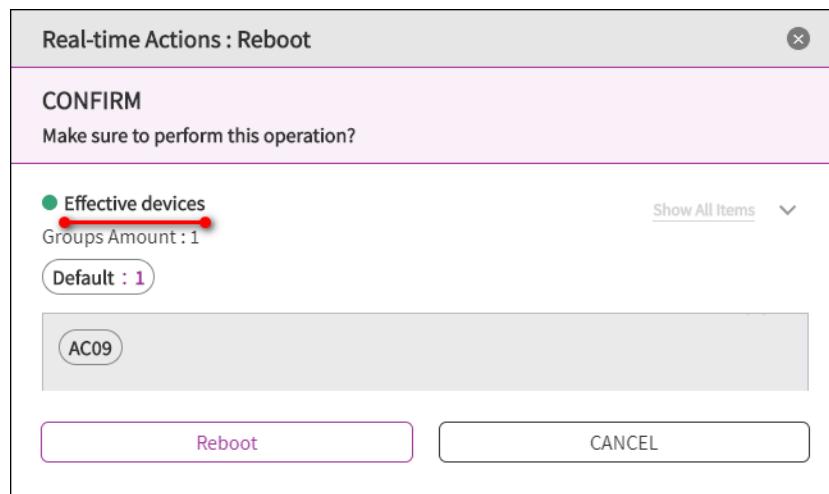
The overview provides quick access to real-time statistics for your managed devices. This information helps to monitor overall status as well as identifying high risk devices. Currently the overview includes Real-time Action, Scheduled Tasks, System Analysis, Device Ranking, and Device Map.

### ● Real-time Actions

Real-time actions provide one-click access to certain actions defined for specific device groups, providing a shortcut for efficient management. Examples for actions are batch reboot, batch screenshot or batch updates.



For example, once you click “Reboot”, a confirmation dialog will pop up and will indicate which devices will actually be affected. Click on the device group button to get more details (individual devices names).



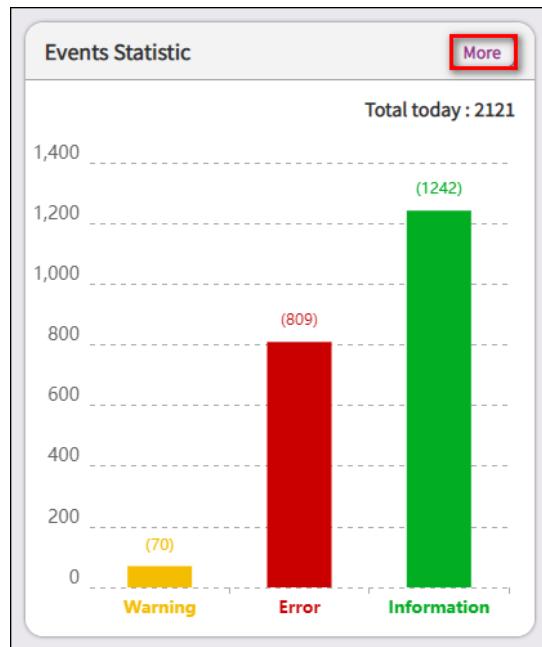
### ● Scheduled Tasks

In addition to real-time action, actions can be scheduled. An example for this is powering off or rebooting devices at a certain time of day. A calendar view is used to visualize upcoming tasks.

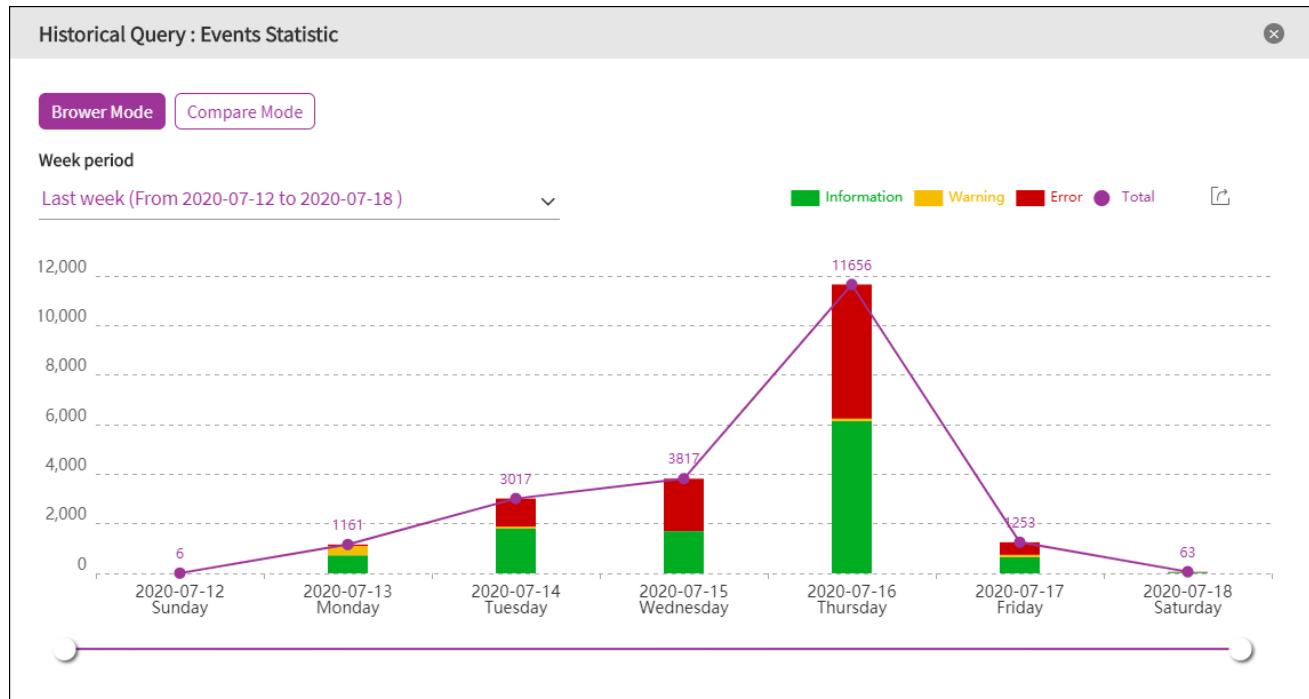
Event		7 Month 2020 Year						
Schedule List		< 7 Month 2020 Year >						
Test Download	2020-07-22 14:42   Cycle	Mon	Tue	Wed	Thu	Fri	Sat	Sun
Testing	2020-07-22 15:02   Cycle	29	30	1	2	3	4	5
		6	7	8	9	10	11	12
		13	14	15	16	17	18	19
		20	21	22	23	24	25	26
		27	28	29	30	31	1	2
		3	4	5	6	7	8	9

- Event Log Statistic

There are three levels of event log on DeviceOn system, such as **Warning**, **Error**, and **Information**. Gives a summary and statistic result for current day. Click on the bar chart to redirect **Event Log** tab to check detail log information.

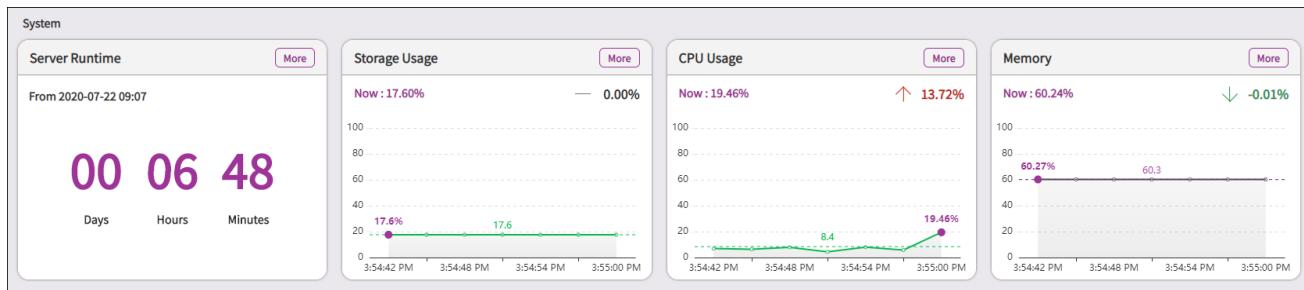


For historical data, please click on “**More**” to compare the daily, weekly result.



- System

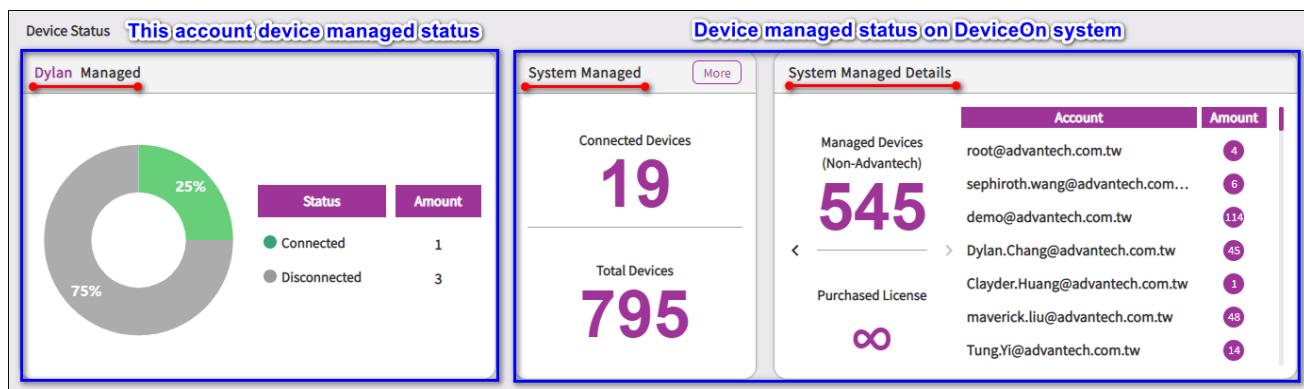
The system is focus on DeviceOn server loading and usage, including storage, CPU and Memory. The administrator could realize the real-time server uptime, downtime through the overview and based on the matrix to scale cloud performance.



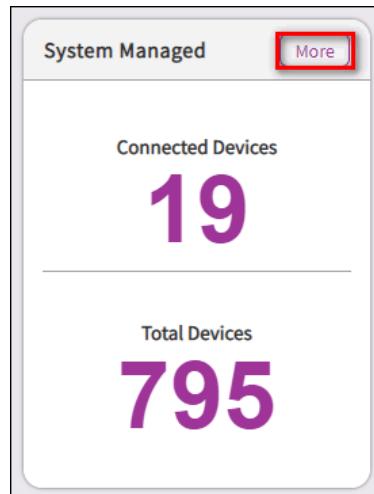
### ● Devices Status

Shows the total devices of current account and DeviceOn system managed.

Shows the number of currently online devices as well as total number of managed devices (assigned to account).



Click on the more information on system managed to show the number of currently online devices as well as total number of managed devices.

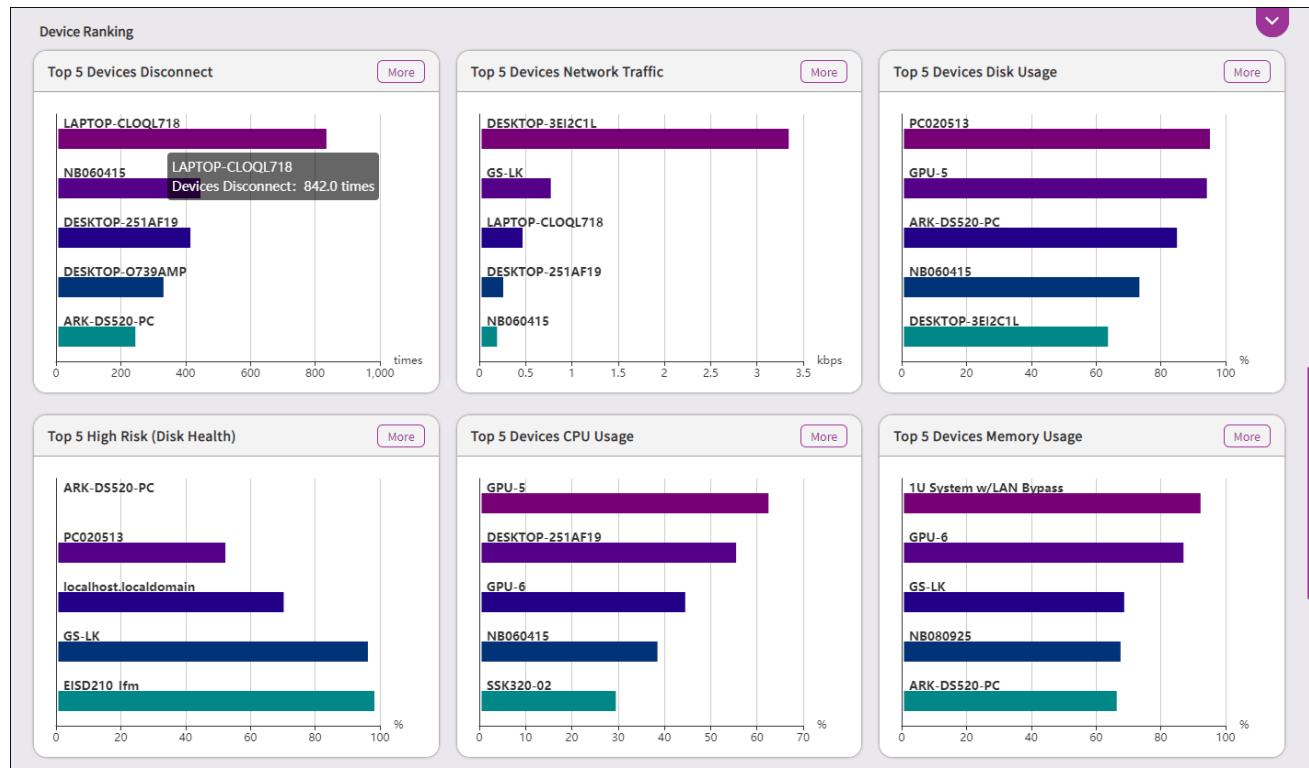


Clicking this overview will bring up a detailed device list including status as well as group membership information.

List			
<b>Connected</b>	<b>Disconnected</b>	Keyword Search <input type="text"/>	 19 Set << < 1 /2 > >>
SETTING STATUS	DEVICE NAME	DEVICE GROUP NAME	WAKE-ON-LAN
●	WISE-3610	Dylan.Chang...	Direct Mode
●	UTX-3117	Kalvin.Yang : ...	Agent Mode : 00000001
●	SSK320-02	SSK : Default	Direct Mode
●	PC020513	demo : Default Hank.Peng : ...	<a href="#">More</a> Direct Mode

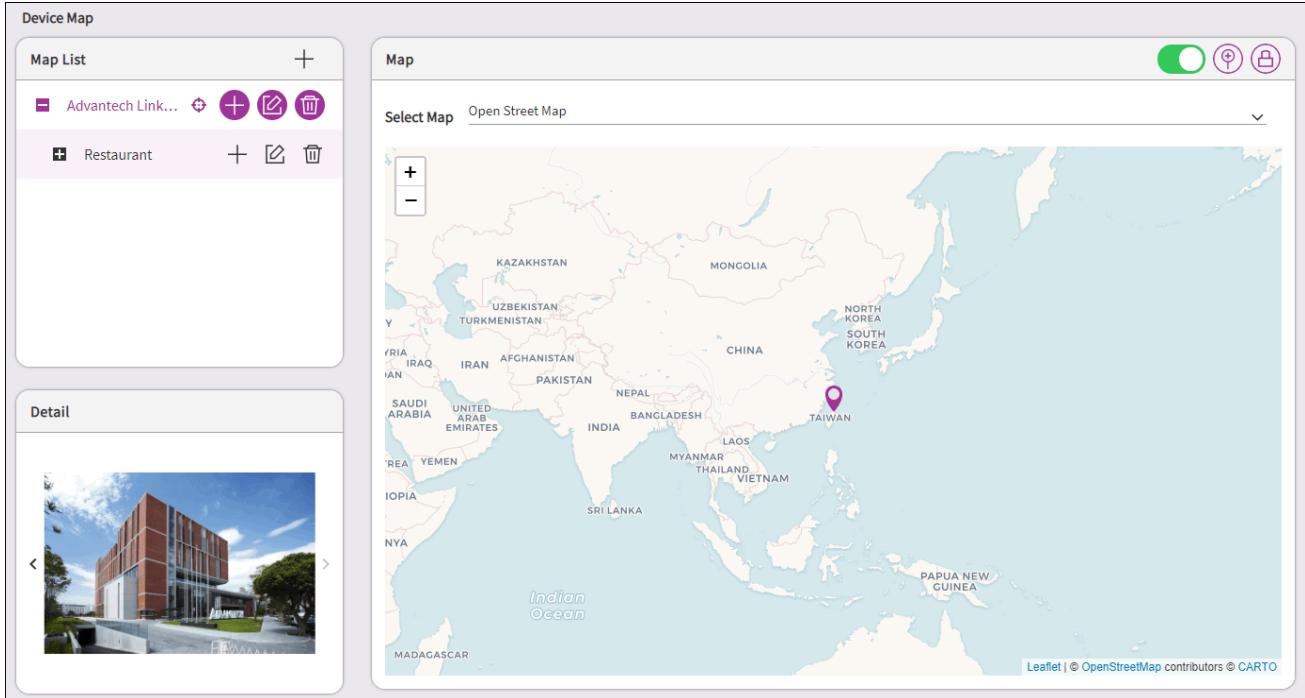
- Top 5 (High-Risk) Statistic

DeviceOn leverages six common sensor types to identify potential high-risk devices. Those sensors are device disconnects, network traffic, disk usage, CPU usage and memory usage. This “top 5” overview allows to quickly identify potential issues and fix or replace the systems to avoid unexpected downtime.



- Device Map (Open Street/Google/Baidu Map)

DeviceOn offers support for maps (latitude and longitude based position) or floor plans in order to visualize the location of managed devices. User could define their location on the map and place the device to the area (floor plans).



The screenshot shows the 'Device Map' feature. On the left, there's a 'Map List' sidebar with icons for adding and deleting items. Under 'Restaurant', there's a list item 'AC09'. On the right, a map of Asia and the Pacific region is displayed with a red location pin over Taiwan. Below the map, there's a detailed view of a modern building with a red brick facade and glass windows. The interface includes standard map controls (zoom in/out) and a legend.



The screenshot shows the 'Device Map' feature using a floor plan of a restaurant. The floor plan is labeled 'SCHEME C' and includes numbered callouts from 1 to 15, each pointing to specific locations in the restaurant. The numbered points correspond to the following descriptions:

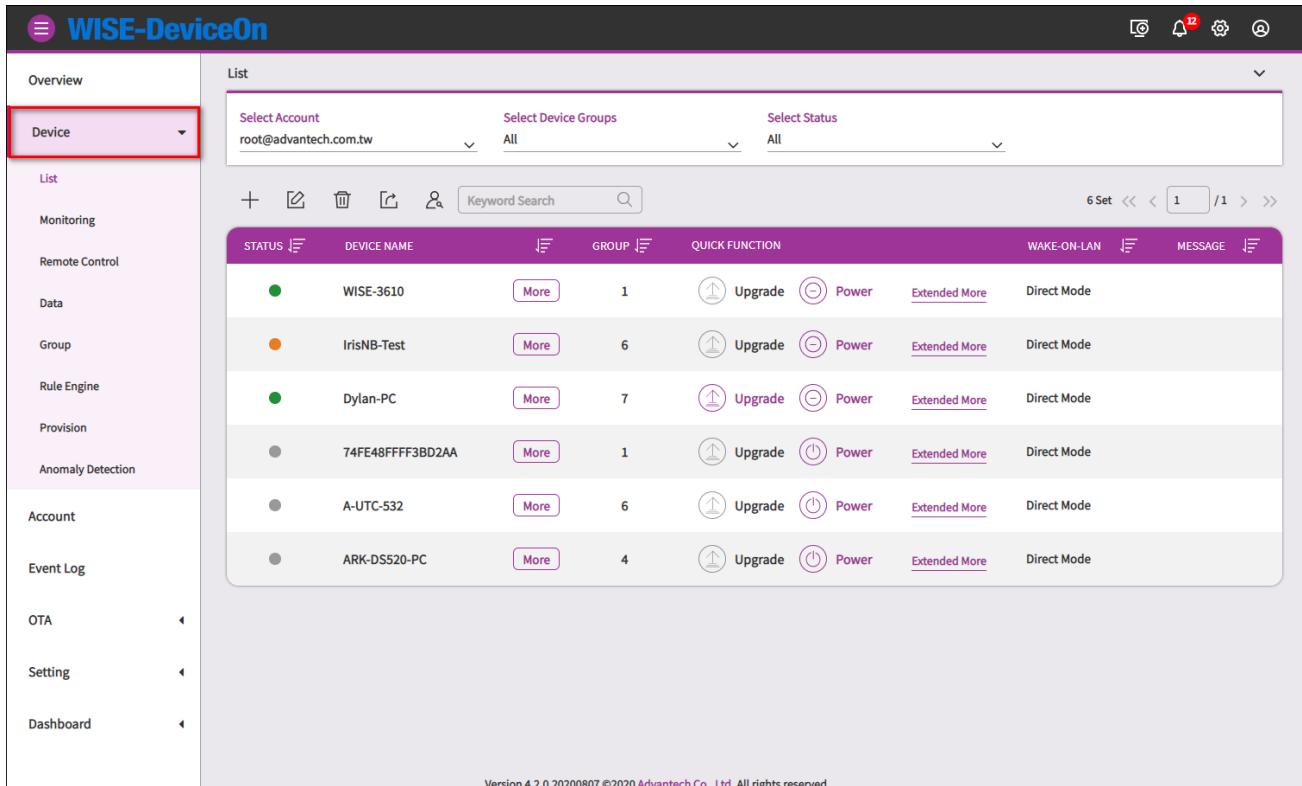
- 1. 雷射點餐機化 - 請中選 (RFID Interactive Restaurant Ordering System)
- 2. 設定主要入口為二樓大廳方向
- 3. 菜單櫃位 (二層櫃位 各式中西點心)
- 4. 自助點餐台 (服務台)
- 5. 自助點餐台 (服務台)
- 6. 自助點餐台 (服務台)
- 7. 自助取餐台 (餐食櫃)
- 8. 自助取餐台 (餐食櫃)
- 9. 自助取餐台 (餐食櫃)
- 10. 咖啡廳點餐櫃 - 一樓
- 11. 咖啡廳點餐櫃 - 二樓 (供應員)
- 12. 咖啡廳點餐櫃 - 二樓 (工作區)
- 13. 低階辦公區域
- 14. 工業4.0展示區 SCHEME B

The interface includes a 'Map List' sidebar with items like 'Advantech Link...', 'Restaurant', and 'AC09', and a 'Detail' section showing a black industrial-grade computer.

### 3.3.2 Device Management

After your device onboarding, you could view, edit device basic information, remote control, and retrieve sensor data on your devices. Eight sub items under Device, Device List contain device name,

upgrade status, power management and etc. Device Monitoring to give device loading at present. To remote diagnostic and debug through Remote Control. Next, the all of plugin sensor data from Device Data. To grouping you device through the Device Group. Rule Engine to set a threshold rule for your devices data in real-time. For advanced configuration, such as WoL, System Backup/Recovery and Protection via Device Provisioning. The last, one of AI solution to detect device screen status on Anomaly Detection.

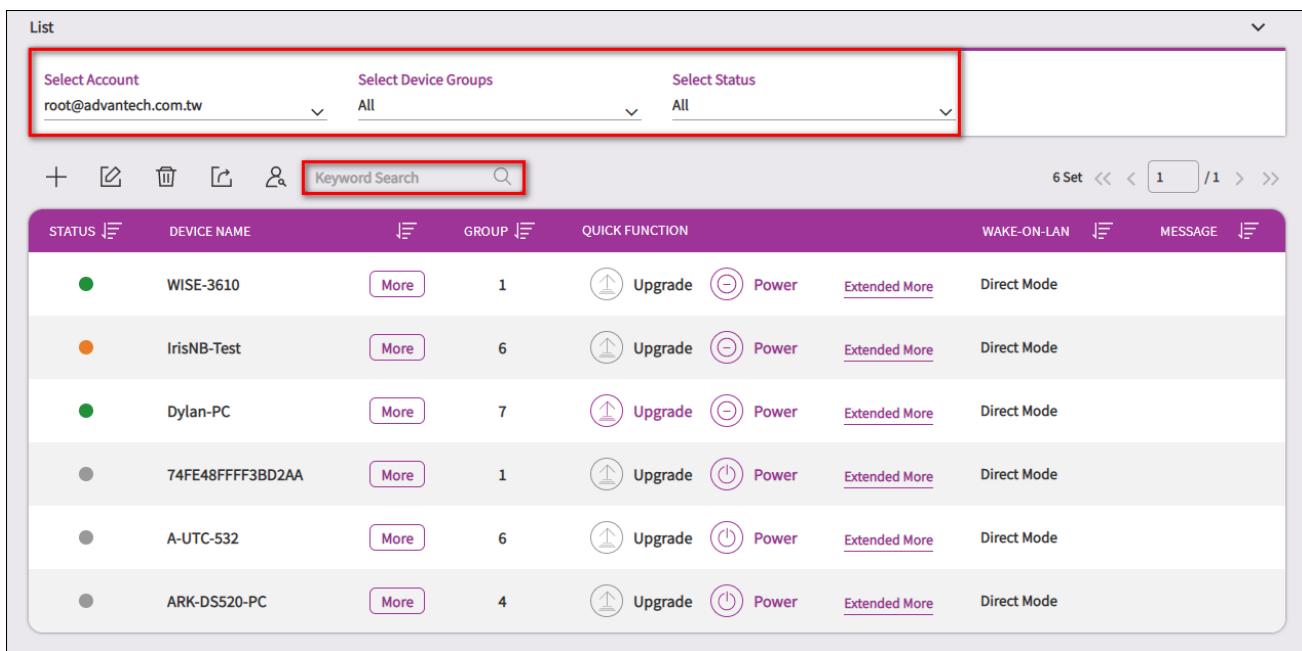


STATUS	DEVICE NAME	GROUP	QUICK FUNCTION	WAKE-ON-LAN	MESSAGE
●	WISE-3610	1	Upgrade Power	Extended More	Direct Mode
●	IrisNB-Test	6	Upgrade Power	Extended More	Direct Mode
●	Dylan-PC	7	Upgrade Power	Extended More	Direct Mode
●	74FE48FFFF3BD2AA	1	Upgrade Power	Extended More	Direct Mode
●	A-UTC-532	6	Upgrade Power	Extended More	Direct Mode
●	ARK-DS520-PC	4	Upgrade Power	Extended More	Direct Mode

### ● Device List

The device could be assigned to multiple accounts and device groups; therefore, you could leverage filter to find your device through **Account**, **Device Group**, **Status** or **Keyword**.

List



Select Account: root@advantech.com.tw | Select Device Groups: All | Select Status: All

Keyword Search:  

6 Set | << < 1 /1 > >>

STATUS	DEVICE NAME	GROUP	QUICK FUNCTION	WAKE-ON-LAN	MESSAGE
●	WISE-3610	More	1  Upgrade  Power <a href="#">Extended More</a>	Direct Mode	
●	IrisNB-Test	More	6  Upgrade  Power <a href="#">Extended More</a>	Direct Mode	
●	Dylan-PC	More	7  Upgrade  Power <a href="#">Extended More</a>	Direct Mode	
●	74FF48FFFF3BD2AA	More	1  Upgrade  Power <a href="#">Extended More</a>	Direct Mode	
●	A-UTC-532	More	6  Upgrade  Power <a href="#">Extended More</a>	Direct Mode	
●	ARK-DS520-PC	More	4  Upgrade  Power <a href="#">Extended More</a>	Direct Mode	

Here is action bar for add, edit, search or export for below table devices.



Click the icon to add devices, that's similar to device onboarding, download WISE-Agent, setup to your local device and grouping.



**Device OnBoarding**

1 Set Up      2 Select Devices      3 Confirm

**Selected Device**  
 Select the devices you want to onboarding.

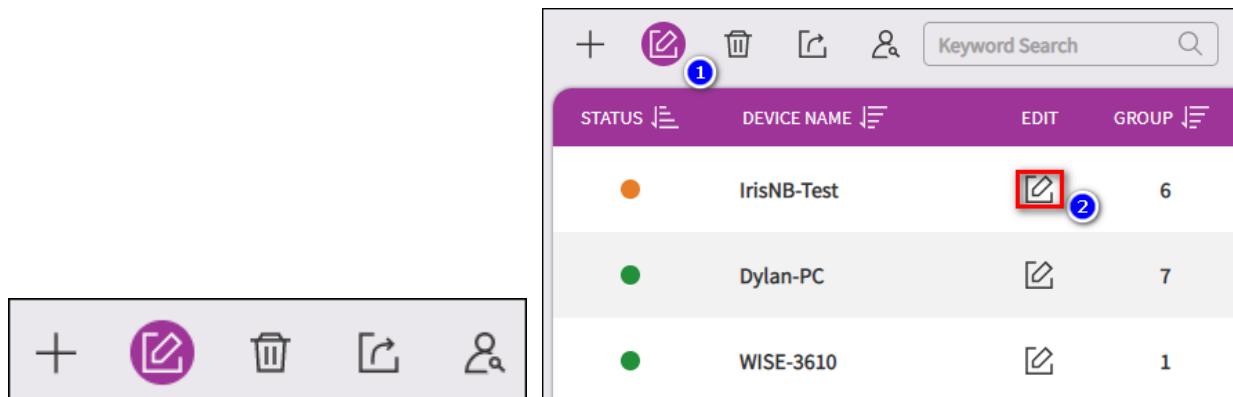
Agent Name  Only Connected Devices  
 Select Device ▾

Advanced Option

**Hint:**  
 (1) WISE-Agent : Available for Windows 7+ , If Available for Ubuntu 16.04 please [contact us](#)  
 (2) Hint: To fast device onboarding, please put the WISE-Agent and credential file (Agent.config) into same folder.

**Back** **Next**

Click the edit icon to display “Edit” options on each device list.



STATUS	DEVICE NAME	EDIT	GROUP
●	IrisNB-Test		6
●	Dylan-PC		7
●	WISE-3610		1

You could edit device name, assign to different accounts, device groups in “Edit Device”

**Edit Device**

---

**Device Name**  
ARK-DS520-PC

**Assigned Account**  
root@advantech.com.tw

**Select Device Groups**

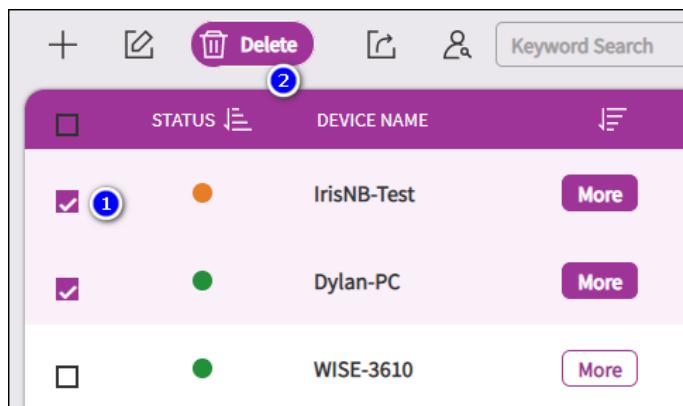
<input checked="" type="checkbox"/> AA-X11	<input type="checkbox"/> AA-X115656
<input type="checkbox"/> Default	<input type="checkbox"/> bgfdbhgbf
<input type="checkbox"/> testonlylrisf~~~	<input type="checkbox"/> wise-test

**Assigned Groups**

Root : AA-X11  scott68.cha...  terry.lu : Def...  000BAB4231... 

**CONFIRM** **CANCEL**

Click the delete icon to display “Delete” options on each device list, pick up the checkbox and confirm to delete these devices.



The screenshot shows a list of devices in a table:

STATUS	DEVICE NAME	More
<input checked="" type="checkbox"/> 1	IrisNB-Test	
<input checked="" type="checkbox"/>	Dylan-PC	
<input type="checkbox"/>	WISE-3610	

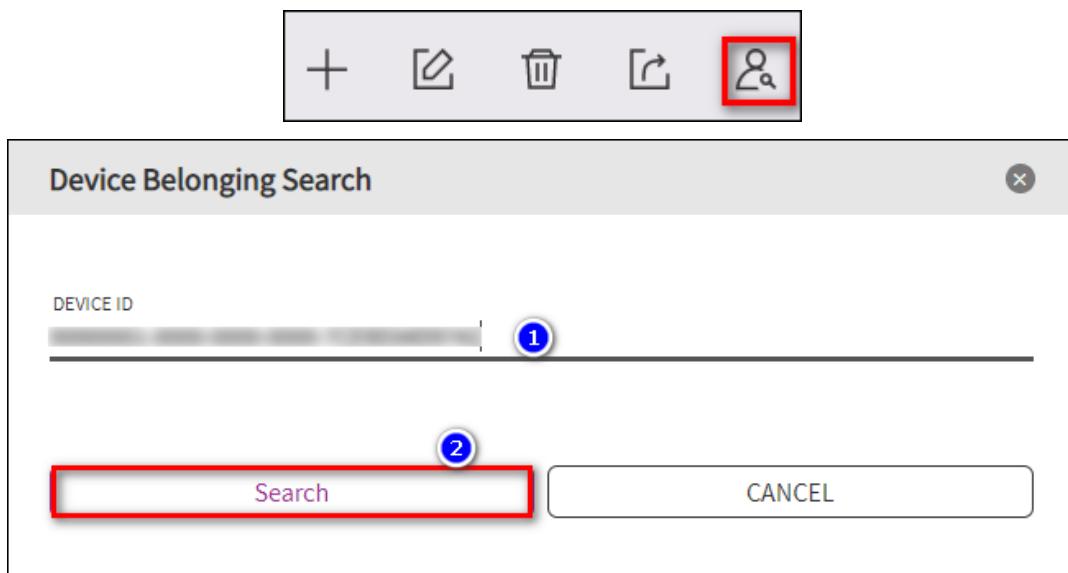
At the top, there are icons for adding (+), editing (pencil), deleting (trash), exporting (CSV), and searching (magnifying glass). A red circle with the number 2 is placed over the delete icon.

Click on export icon to export devices that in the table as CSV file.



Device Name	Agent ID	WAKE-ON-LAN	Mac	Message	Status
DESKTOP-NRB0J2A ac09	0000-0000-0000-0000-0000-0000-0000-0000-1234	Not Set	00E1 7E1 30 AD		Device Online

If you would like to know a device be assigned to which account and device group, click search icon to enter Agent ID (from your WISE-Agent UI) to understand.



Actually, you could do lots of remote action on the device.

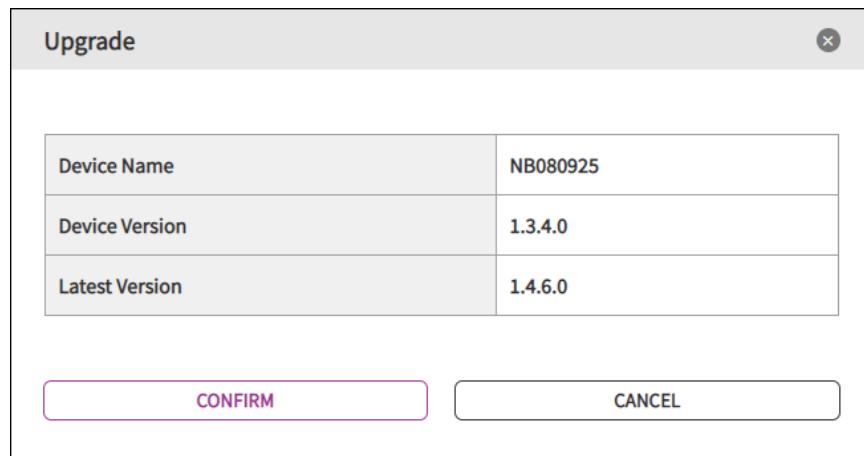
STATUS	DEVICE NAME	GROUP	FULL FUNCTION	WAKE-ON-LAN	MESSAGE
● IrisNB-Test	IrisNB-Test	6	Power		
			Protection		
			Upgrade		Direct Mode
			Backup and Recovery		
			Back		
● Dylan-PC	Dylan-PC	7	Upgrade  Power	<u>Extended More</u>	Direct Mode
● WISE-3610	WISE-3610	1	Upgrade  Power	<u>Extended More</u>	Direct Mode

- ❖ **Status:** Green light represent device connected, gray for disconnected and orange for device abnormal, due to device over threshold.
- ❖ **Device Name:** Device name, click **More** to get more deice information, such as platform, operation system, MAC, memory, etc.

Device Details		
		Function Information
<b>Basic Information</b>		
<b>Device Name</b>	<b>Wake-on-LAN</b>	
AC09	直接喚醒模式	
<b>Connection Status</b>	<b>Last Connected At</b>	<b>Device ID</b>
 connected	2020/08/13 09:30	00000001-0000-0000-0000-000BAB1255AF
<b>Status Message</b>	<b>Data Upload</b>	<b>Upload Interval</b>
null	Data Upload ON	60 seconds
<b>Device Groups</b>	<a href="#">Show All Items</a>	<b>System Information</b>
Groups Amount : 1		
 Default		
<b>Assigned Account</b>	<a href="#">Show All Items</a>	<b>Operating System</b>
Groups Amount : 1		Windows 10 Enterprise LTSC 2019 X64
 Root : Default		<b>Version</b>
<b>Product</b>		<b>MAC</b>
WISE-Agent		00D0C9123491
		<b>CPU</b>
		Intel(R) Core(TM) i7-2655LE CPU @ 2.20GHz
		<b>Memory</b>
		8272700 KB
		<b>Platform</b>
		SOM-5890
		<b>S/N</b>
		<b>BIOS</b>
		V1.12

- ✧ **Group:** Number of groups for the devices, for example, the device could belong to multiple group.
- ✧ **Quick Functions:** Including Power Management, System Protection, Upgrade WISE-Agent and System Backup/Recovery.

The power management supports On/Off, Restart, Sleep and hibernate, the actions depend on your device supported. Second, protection is power-by McAfee white-list protection mechanism to solidify device system. After enable, 3<sup>rd</sup> execution file, bat, DLL cannot be launch. Please go to **Setting -> Provision-> Protection** to install first. Next, for the WISE-Agent upgrade, if there is new version released by Advantech, it will check and show the icon automatically.



Fourth, System Backup/Recovery is power-by Acronis to backup/recovery device runtime system partition. Please go to **Setting -> Provision-> Backup/Recovery** to install first.

- ❖ **Wake-On-LAN:** Wake-On-LAN mode for device, three mode to power your device up, “**Direct Mode**”, “**Agent Mode**” and “**Repeater Mode**”. The magic package sent by DeviceOn Server call “Direct Mode”, but cannot through different network. Therefore, to overcome this limitation, through another Agent or Router to send, forward magic packet. Please go to **Setting -> Provision-> Power On** to configure.
- ❖ **Message:** Device current status

#### ● Device Monitoring

On this page, you could get real-time information about the device that you selected. The information includes general PC status, such as network speed, software process, disk healthy, CPU and memory usage. If the device is Advantech industrial PC and SUSI driver supported, the RPM (Revolution(s) Per Minute) of CPU FAN, system, board level voltage, temperature is displayed on the page.

## WISE-DeviceOn

☰ WISE-DeviceOn
Logout  5  

- Overview
- Device ▼
- List
- Monitoring**
- Remote Control
- Data
- Group
- Rule Engine
- Provision
- Anomaly Detection
- Account
- Event Log
- OTA
- Setting
- Dashboard

**31**

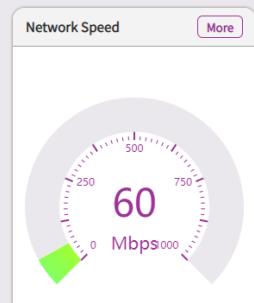
**Monitoring**

Select Account  
root@advantech.com.tw
Select Device Groups  
All
Select Device  
AC09
Refreshing Every  
3 seconds

Network
System
Event

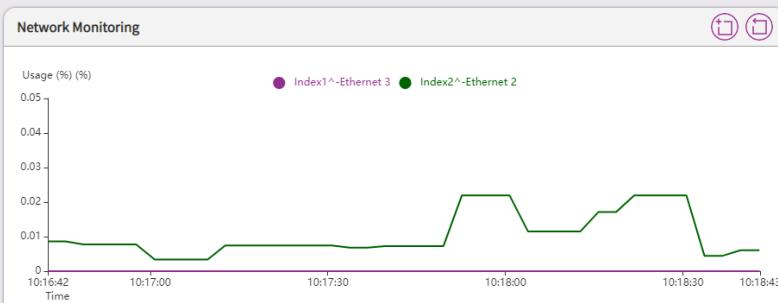
**Network**

Network Speed
More



**Network Monitoring**

Usage (%) (%)
More

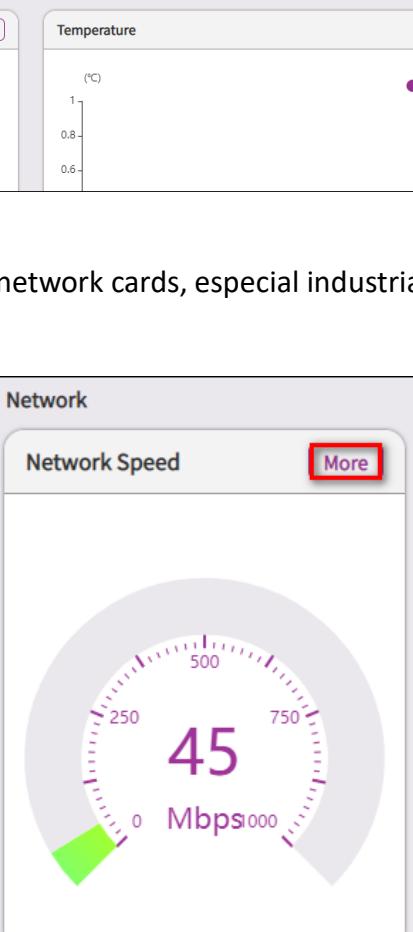


Index1^Ethernet 3   Index2^Ethernet 2

Time: 10:16:42, 10:17:00, 10:17:30, 10:18:00, 10:18:30, 10:18:43

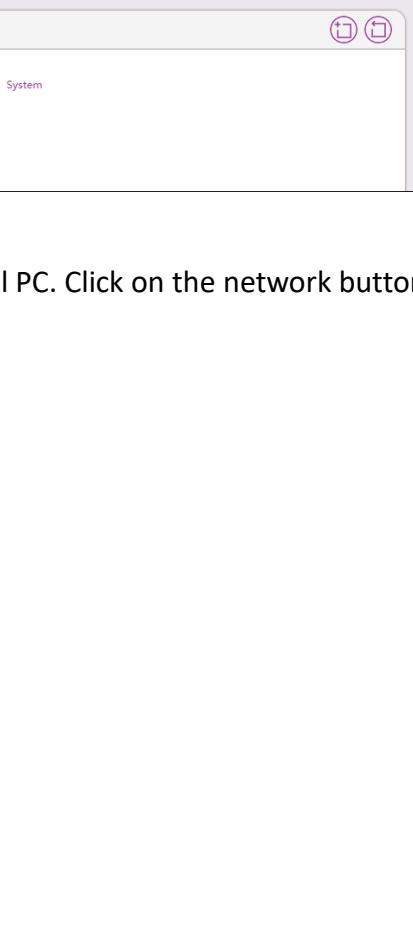
**System**

Software Process
More



**Temperature**

(°C)
More

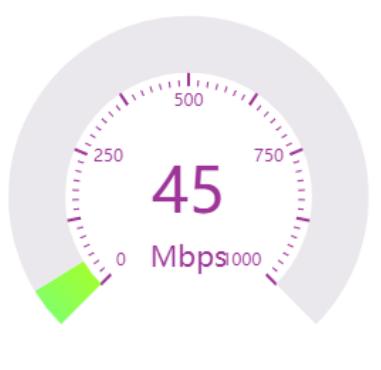


System

Some of devices support multiple network cards, especial industrial PC. Click on the network button to retrieve others.

**Network**

Network Speed
More

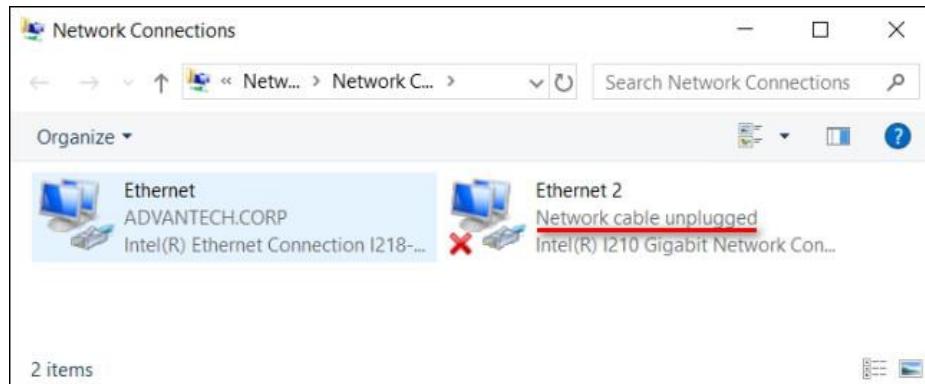


Purchase Information & Product Page

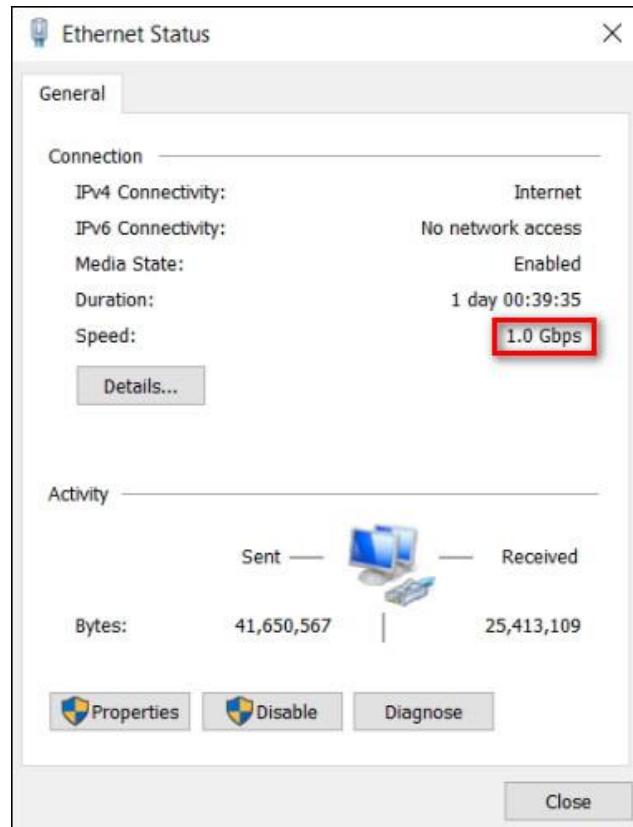
72

Network Usage Details						
Statue	Name	Description	Link Speed (Mbps)	Usage (%) (%)	Speed (Mbps)	
●	Index2^Ethernet 2	Intel(R) 82574L Gigabit Network Connection #2	1000	0.003715	37.15	
●	Index1^Ethernet 3	Intel(R) 82574L Gigabit Network Connection	0	0	0.00	

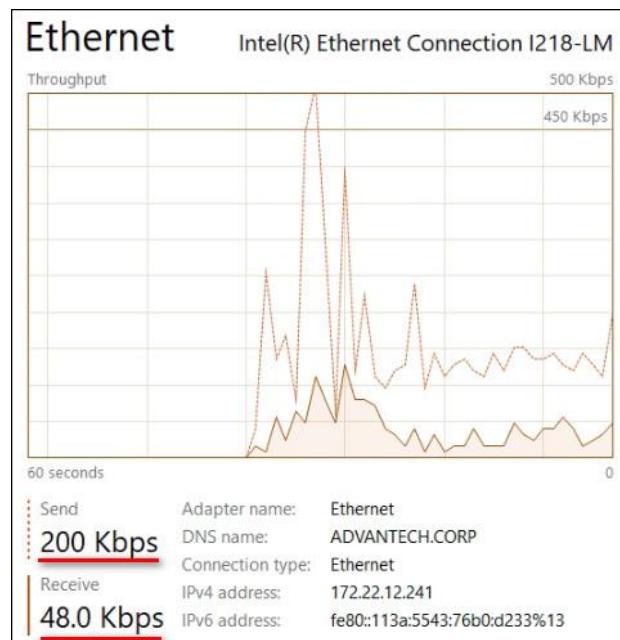
- ✧ **Name:** Name of network card
- ✧ **Description:** network description
- ✧ **State:** Network connected or disconnected, for example, ethernet cable plugin or not.



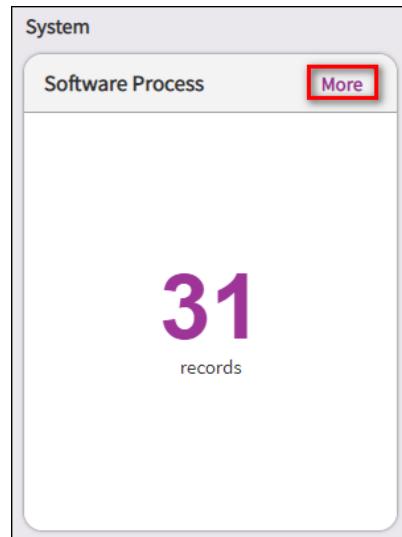
- ✧ **Link Speed (MBPS):** Network maximum link speed



- ◊ **Usage:** Network current usage, **Speed/Link Speed**.
- ◊ **Speed (MBPS):** Send plus receive data rate.



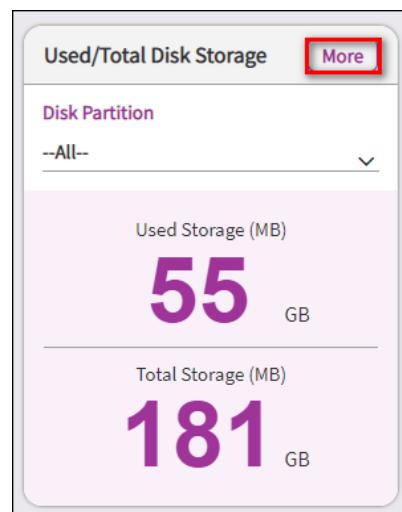
Click on **Software Process** to show **current user process** list, if your device system not login, the result might be zero.



Click on the reset or delete icon, you could restart, terminal your specific process.

Software Process Details						X
PROCESS NAME	RESET	PID	CPU USAGE	MEMORY (GB)		
explorer.exe	(2) 	7664	0	58,544		
ServerConnection.exe		7816	0	57,508		
ServerConnection.exe		5536	0	47,820		
Taskmgr.exe		2212	1	45,848		
ServerConnection.exe		6928	0	41,032		

For hard drive status, not only include current **Used Storage**, but **Healthy** and **Power on Time**. The healthy is based on Acronis healthy model, that calculate on edge side, if you are interested, refer to the [official page](#).



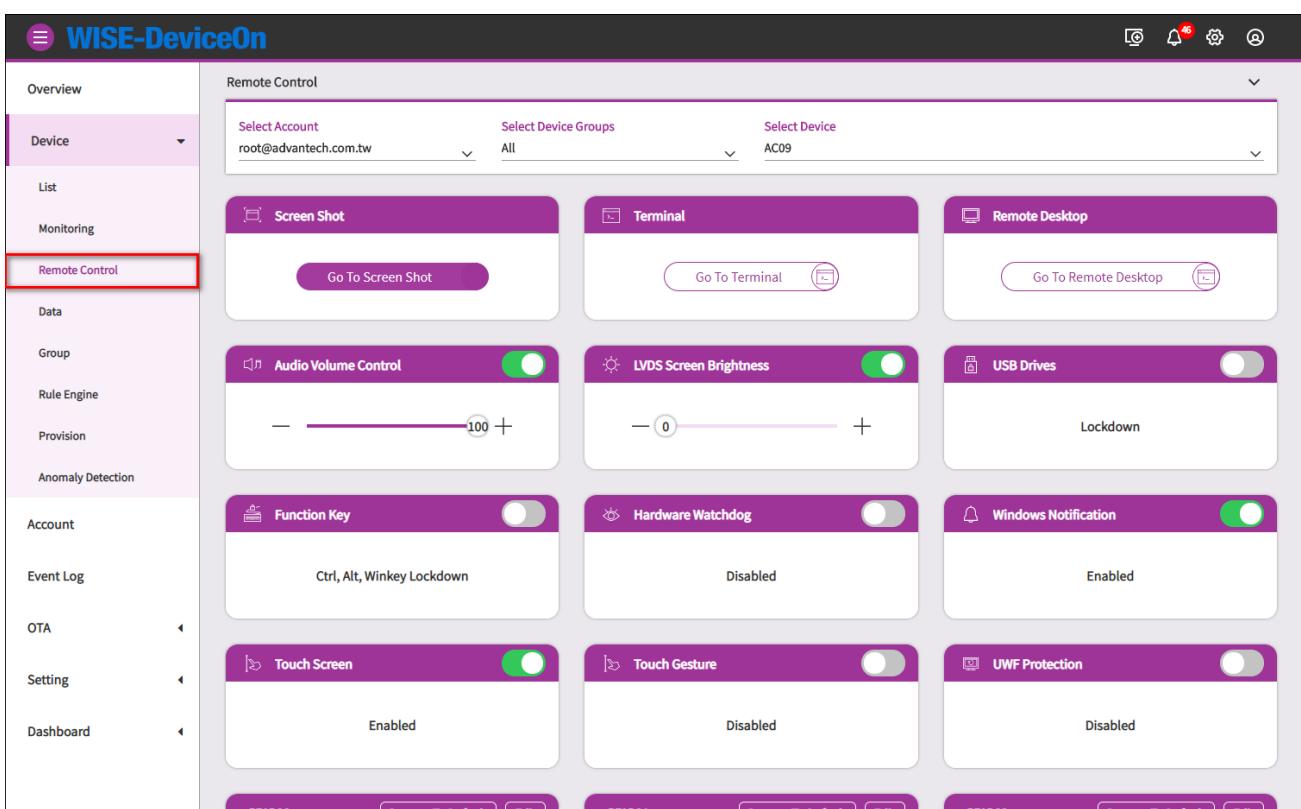
Storage Details						X
Hard Drives		Keyword Search				1 Set << < 1 /1 > >>
INDEX	NAME	TYPE	HEALTH (%)	TEMPERATURE (°C)	POWER ON TIME (HOUR)	
0	ST250LT012-9WS141	STDDisk	18	38	27703	

Disk Partition						X
Disk Partition		Keyword Search				1 Set << < 1 /1 > >>
DISK NAME	EDIT	USED STORAGE (MB) (MB)	EDIT	TOTAL STORAGE (MB) (MB)	EDIT	
Disk C:		56,562		185,548		

- Remote Control

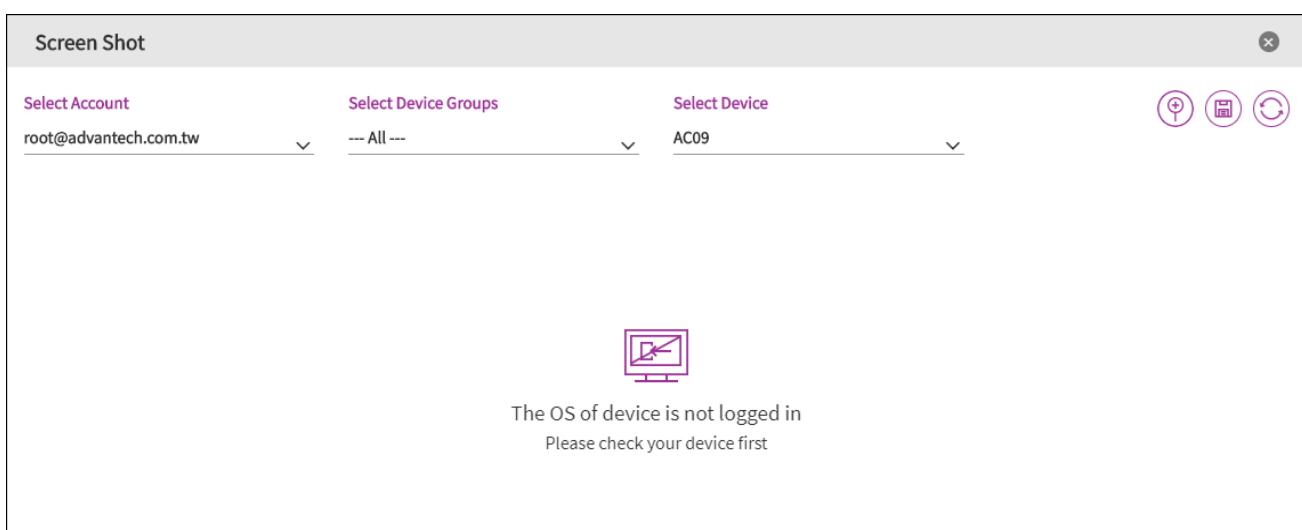
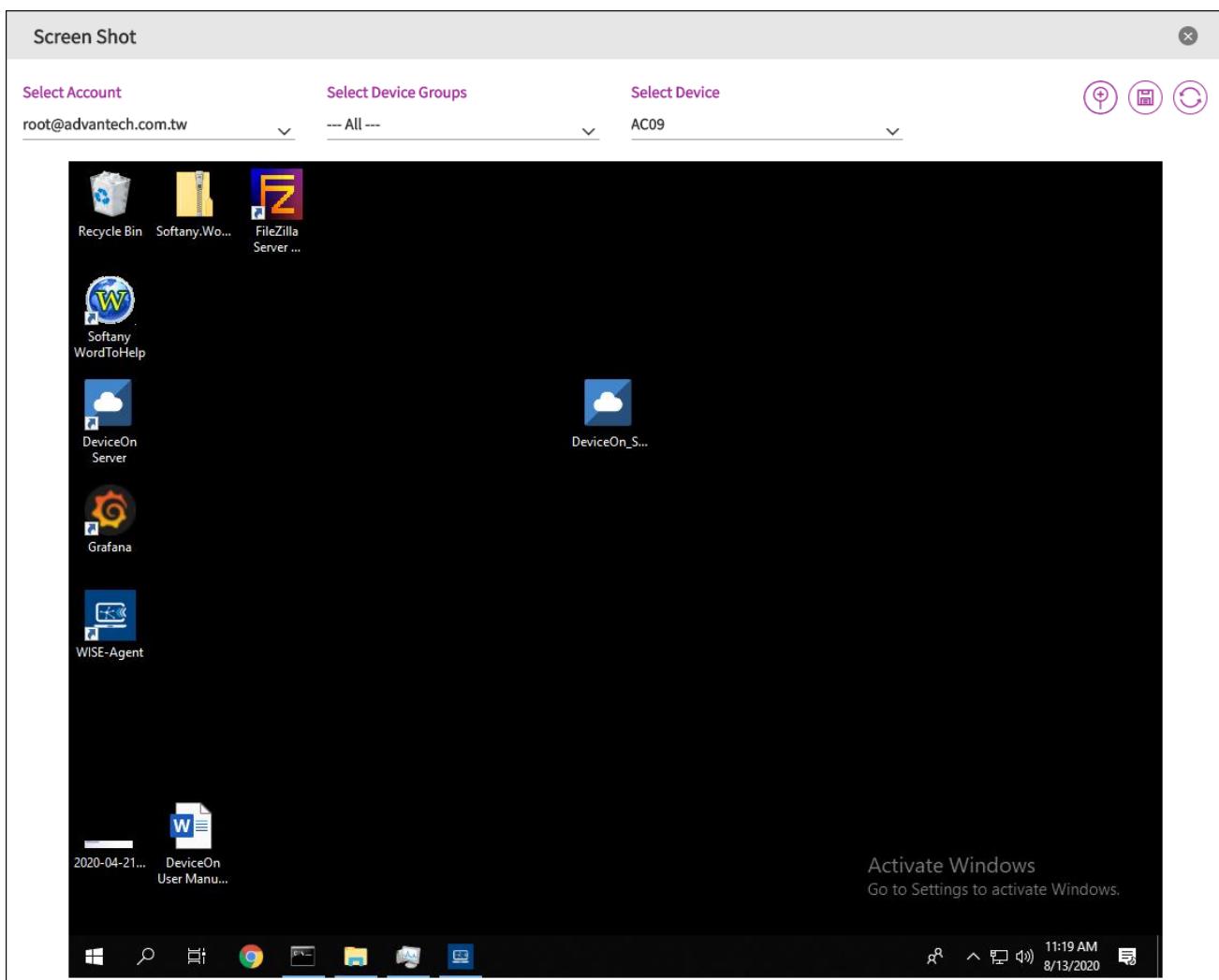
If you need to debug, diagnostic to your devices, actually, do not need go to field side. Through DeviceOn remote control to manage to reduce your operation effort. Basically, there are three functions (**Screenshot**, **Terminal** and **Remote Desktop**) for most devices.



The screenshot shows the WISE-DeviceOn interface. On the left, there's a sidebar with various menu items: Overview, Device (selected), List, Monitoring, Remote Control (highlighted with a red box), Data, Group, Rule Engine, Provision, Anomaly Detection, Account, Event Log, OTA, Setting, and Dashboard. The main area is titled "Remote Control" and contains several sections: "Select Account" (root@advantech.com.tw), "Select Device Groups" (All), and "Select Device" (AC09). Below these are three cards: "Screen Shot" (Go To Screen Shot), "Terminal" (Go To Terminal), and "Remote Desktop" (Go To Remote Desktop). Further down are two rows of four cards each, each with a toggle switch: "Audio Volume Control" (on), "LVDS Screen Brightness" (on), "USB Drives" (off), "Function Key" (off) with "Ctrl, Alt, Winkey Lockdown" below it; "Hardware Watchdog" (off), "Windows Notification" (on), "Touch Screen" (on), "Touch Gesture" (off); and "UWF Protection" (off).

### [Screenshot]

Through the Screenshot to get device real-time screen, there is a limitation, your device **must login to operation system**, otherwise, cannot capture screen and shown “**The OS of device is not logged in**”



### [Terminal]

To terminal support any command to your devices, for instance, realize your device IP, traceroute the network or copy/view file on the device.

### Terminal

Select Account Select Device Groups Select Device

root@advantech.com.tw All AC09 

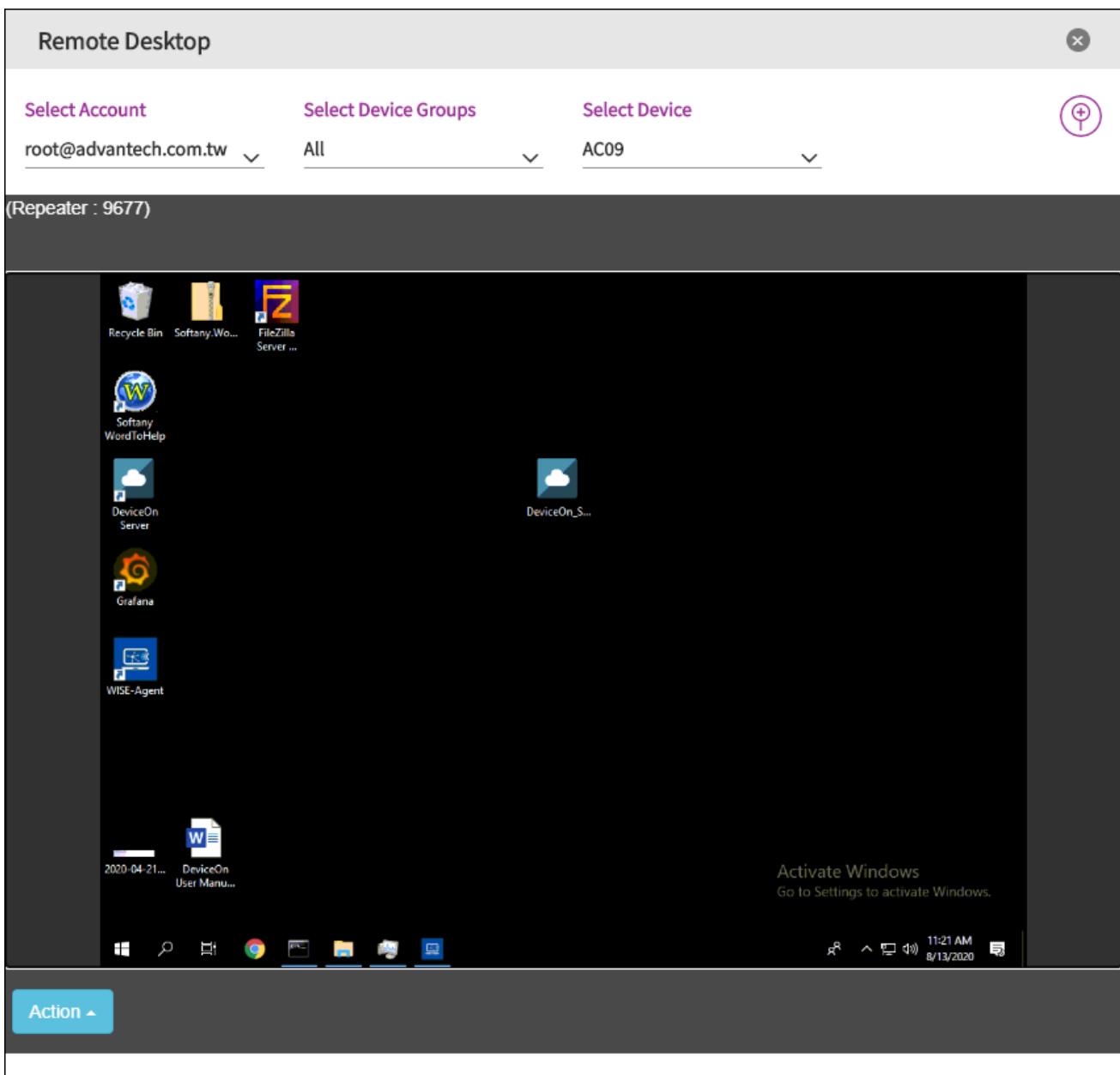
```
Microsoft Windows [Version 10.0.17763.1339]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Windows\system32>
```

Enter Command... Send

#### [Remote Desktop]

DeviceOn leverage VNC (Virtual Network Computing) technology to achieve remote desktop, to bridge different network between public and private. User do not need to install any program, App on their laptop or mobile devices. Through DeviceOn website to remote desktop to debug and diagnostic.



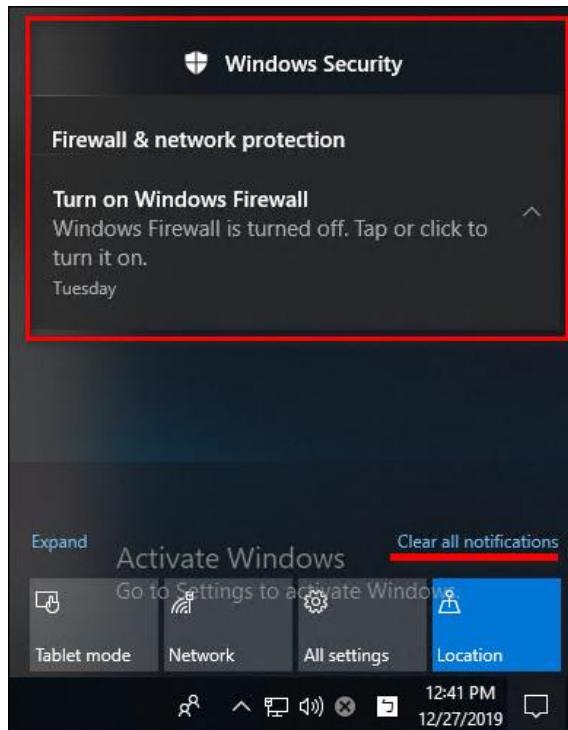
### [Advanced Control]

For others features depend on your device operation system and hardware. DeviceOn integrate Windows Lockdown features on LTSC (Long Time Service Channel) and LTSB (Long Time Service Branch) to provide advanced control, such as “**Block USB Drives**”, “**Keyboard Filter**”, “**Block Windows Notification**”, “**Block Touch, Gesture**” and “**UWF (Unified Write Filter)**”.

**[USB Drive]:** Prevent threats from outside **USB drives**, not include keyboard, mouse.

**[Function Key]:** Disables **Ctrl**, **Alt**, and **WinKey**.

**[Windows Notification]:** Block application notification.



**[Touch Screen]:** Disable touch control

**[Tough Gesture]:** Disable gesture control

**[UWF Protection]:** To protect your drives by intercepting and redirecting any writes to the drive (app installations, settings changes, saved data) to a virtual overlay. The virtual overlay is a temporary location that is usually cleared during a reboot or when a guest user logs off.

**Benefits:**

- Provides a clean experience for thin clients and workspaces that have frequent guests, like school, library or hotel computers. Guests can work, change settings, and install software. After the device reboots, the next guest receives a clean experience.
- Increases security and reliability for kiosks, IoT-embedded devices, or other devices where new apps are not expected to be frequently added.
- Can be used to reduce wear on solid-state drives and other write-sensitive media.

For **backlight, brightness, GPIO** and **Watchdog** only support on Advantech hardware platform with SUSI driver, please download from [Advantech Support](#) site.

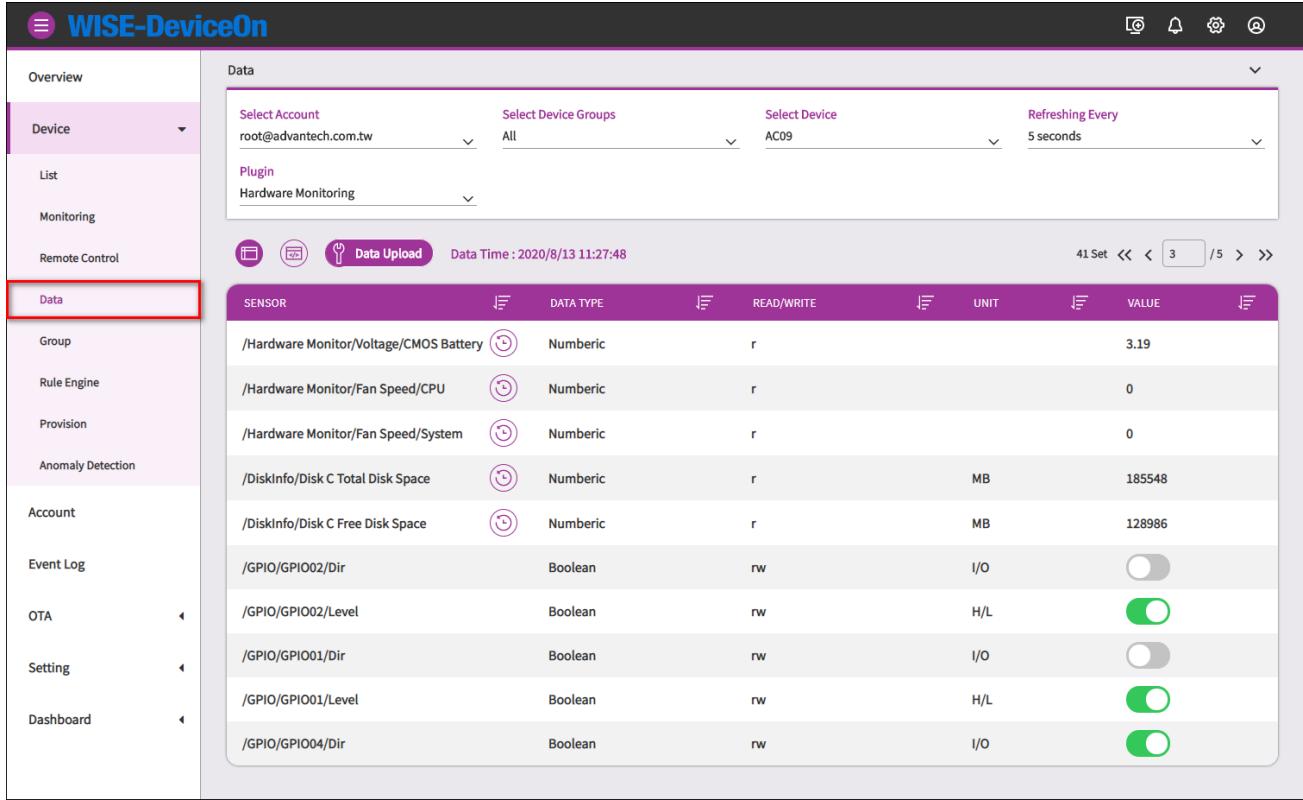
**[LVDS, Backlight and Brightness]:** Turn on/off LVDS backlight for power saving.

**[Watchdog Protection]:** Hardware level watchdog to prevent BSOD (Blue Screen of Death) or system hang without any response. If happened, watchdog will restart your device automatically. There is a tool call [NotMyFault](#) that you can use to crash, hang, and cause kernel memory leaks on your Windows system.

**Benefits:** Avoid embarrassing moment, if BSOD on your Signage devices over the airport, department store and public area.

### ● Device Data

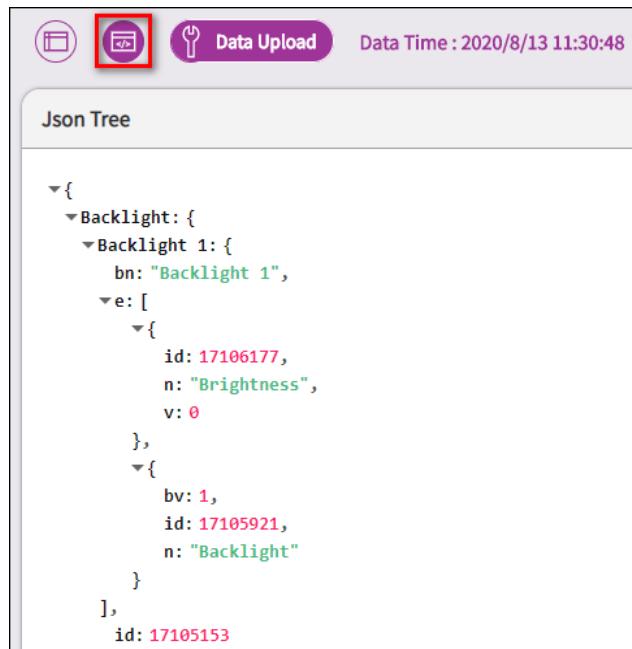
Raw data of each plugin on devices, user could get real-time and historical data on this page. To data analysis and aggregation, user could adjust data report interval or reset to default (60s) for basic sensors.



The screenshot shows the WISE-DeviceOn interface. On the left, there's a sidebar with various tabs: Overview, Device (selected), List, Monitoring, Remote Control, Data (highlighted with a red box), Group, Rule Engine, Provision, Anomaly Detection, Account, Event Log, OTA, Setting, and Dashboard. The main area is titled "Data" and contains four sections: "Select Account" (root@advantech.com.tw), "Select Device Groups" (All), "Select Device" (AC09), and "Refreshing Every" (5 seconds). Below these are sections for "Plugin" (Hardware Monitoring) and "Data Upload". A table titled "SENSOR" lists various monitoring points with their details: Sensor Path, Data Type, Read/Write, Unit, and Value. Some values are numeric (e.g., 3.19, 0, 0, 185548, 128986), while others are Boolean (e.g., I/O, H/L, MB). To the right of the table are icons for "41 Set", navigation arrows, and a search bar.

SENSOR	DATA TYPE	READ/WRITE	UNIT	VALUE
/Hardware Monitor/Voltage/CMOS Battery	Numeric	r		3.19
/Hardware Monitor/Fan Speed/CPU	Numeric	r		0
/Hardware Monitor/Fan Speed/System	Numeric	r		0
/DiskInfo/Disk C Total Disk Space	Numeric	r	MB	185548
/DiskInfo/Disk C Free Disk Space	Numeric	r	MB	128986
/GPIO/GPIO02/Dir	Boolean	rw	I/O	(Switch icon)
/GPIO/GPIO02/Level	Boolean	rw	H/L	(Green switch icon)
/GPIO/GPIO01/Dir	Boolean	rw	I/O	(Grey switch icon)
/GPIO/GPIO01/Level	Boolean	rw	H/L	(Green switch icon)
/GPIO/GPIO04/Dir	Boolean	rw	I/O	(Green switch icon)

The default display is table mode, you could switch to JSON format through the icon.



This screenshot shows the same interface but with the "Json Tree" icon selected (highlighted with a red box) instead of the table icon. The main area displays a hierarchical JSON tree structure under the heading "Json Tree". The tree shows nested objects and arrays, with some fields like "id" and "n" highlighted in pink and green respectively. The JSON structure includes "Backlight", "Backlight 1", "e", "v", "Backlight 2", and "id: 17105153".

```

{
  "Backlight": {
    "Backlight 1": {
      "bn: "Backlight 1",
      "e: [
        {
          "id: 17106177,
          "n: "Brightness",
          "v: 0
        },
        {
          "bv: 1,
          "id: 17105921,
          "n: "Backlight"
        }
      ],
      "id: 17105153
    }
  }
}

```

- Device Group

Every account could group their device into different groups to manage, for example, device over different floor on the building. User could create 1F, 2F group to easy management.

DEVICE GROUP NAME	DESCRIPTION	DEVICE
1F	Taipei HQ	Devices Amount : 1 WINDOWS-Q...
Default	Default	Devices Amount : 6 AIMB-U233 DESKTOP-85...

Click on the icon to add “**Device Group**”. The option to configure the parent group, that’s means share the device group to parent group owner.



Add Device Group

Device Group Name

2F

Description

Taipei HQ

Select Account

Select Device Groups

Parent Group (optional)

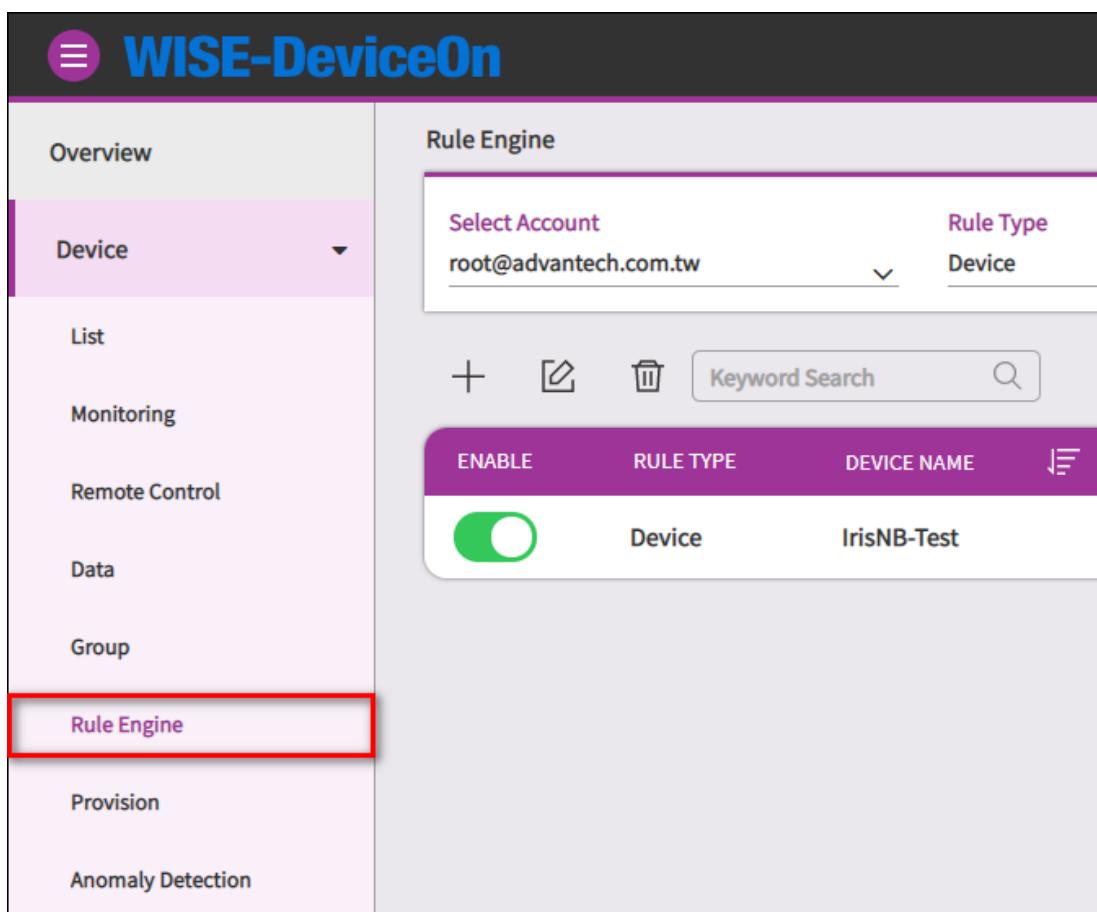
Save CANCEL

Click on the icon to “Edit” or “Delete” account.



- Rule Engine

DeviceOn provides the rule engine. Users can acquire anomaly situations by means of setting thresholds to those interested devices, and, once one or more thresholds meets, receive alerts via event notification services, another one indispensable feature for users.



ENABLE	RULE TYPE	DEVICE NAME
<input checked="" type="checkbox"/>	Device	IrisNB-Test

Click on the add icon to create a Rule.



Pick-up the sensor that you want to monitor, the steps are select **Rule Type**, **Device Group** and **Device**.

### Add Rule

1 Select Sensor      2 Define Threshold      3 Define Action      4 Confirm

**Rule Type** Device 1      **Device Group** AA-X11 2      **Device** ARK-DS520-PC 3

**Default Mode** **Detailed Mode** Keyword Search  24 Set << < 1 / 5 > >>

SELECT	SENSOR NAME	SENSOR ID
<input type="radio"/>	Hard Drive Health	HDDMonitor/hddInfoList/Disk0/health
<input type="radio"/>	Hard Drive Power on Time	HDDMonitor/hddInfoList/Disk0/powerOnTime
<input type="radio"/>	Hard Drive Total Space	HDDMonitor/DiskInfo/Disk C:/Total Disk Space
<input type="radio"/>	Hard Drive Free Space	HDDMonitor/DiskInfo/Disk C:/Free Disk Space
<input type="radio"/>	System, Available Physical Memory	ProcessMonitor/System Monitor Info/availPhysMemKB

**4**

**Next**

Define the threshold, provide 3 types, **more than**, **less than** and **outside the range**. Also, you could realize current value on the page.

- ✧ Lasting Time (Second): means the sensor over the threshold and continue for a period time, avoid peak value to trigger.
- ✧ Notice Interval (Second): If over the threshold, the WISE-Agent will send a notify event, to avoid lots of message, user could adjust notice interval.

Add Rule ×

Select Sensor Define Threshold Define Action Confirm

1 2 3 4

**Sensor Name:**  
Hardware, 3.3V

**Threshold**  
 More than    Less than    Outside the range

**Range (Current Value: 3.424) Unit: V**  
 +  
 0

**Lasting Time (Second):** 10   **Notice Interval (Second):** 60

Back Next

Next, to define the action, if threshold reached. For example, you could power your device off, if the hard drive unhealthy.

Add Rule ×

Select Sensor Define Threshold Define Action Confirm

1 2 3 4

**Take an Action** **Take a Sub Action**

Power On/Off System Power off

**Trigger Frequency**  
 Always    Back to Normal    Once

Back Next

Confirm the rule setting and click confirm.

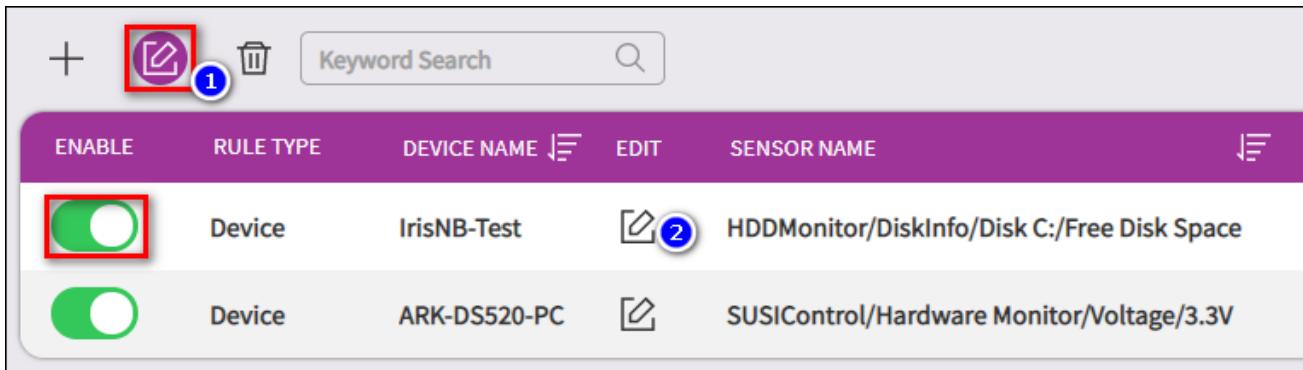
### Add Rule

1 Select Sensor      2 Define Threshold      3 Define Action      4 Confirm

Sensor	Threshold	Take an Action
Rule Type Device	Define Threshold (Unit: V) More than 3	Take an Action Power On/Off
Device Group AA-X11	Lasting Time (Second) 10	Take a Sub Action System Power off
Device ARK-DS520-PC	Notice Interval (Second): 60	Trigger Frequency Once
Plugin SUSIControl	Basic Information	
Sensor ID SUSIControl/Hardware Monitor/Voltage e/3.3V	Enable Rule 	

**Back** **Confirm**

The rule list shown as below, user could edit or disable through the switch.

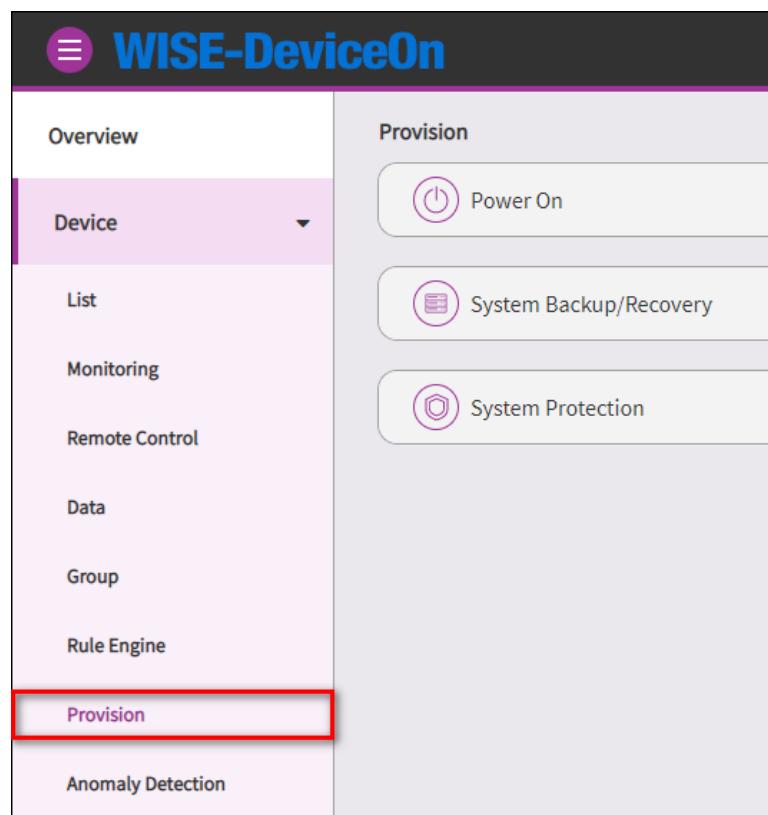


ENABLE	RULE TYPE	DEVICE NAME	EDIT	SENSOR NAME
 1	Device	IrisNB-Test	 2	HDDMonitor/DiskInfo/Disk C:/Free Disk Space
	Device	ARK-DS520-PC		SUSIControl/Hardware Monitor/Voltage/3.3V

- Provision

For device provision, 3 types need be pre-configured. One is “**Power On**”, select which mode to enable device wake up. The others are 3<sup>rd</sup> party tool integration, **Acronis** to **backup/recovery** your device system and **McAfee** for white-list security **protection**. To install 3<sup>rd</sup> tools, you must purchase

the license and activate the product.



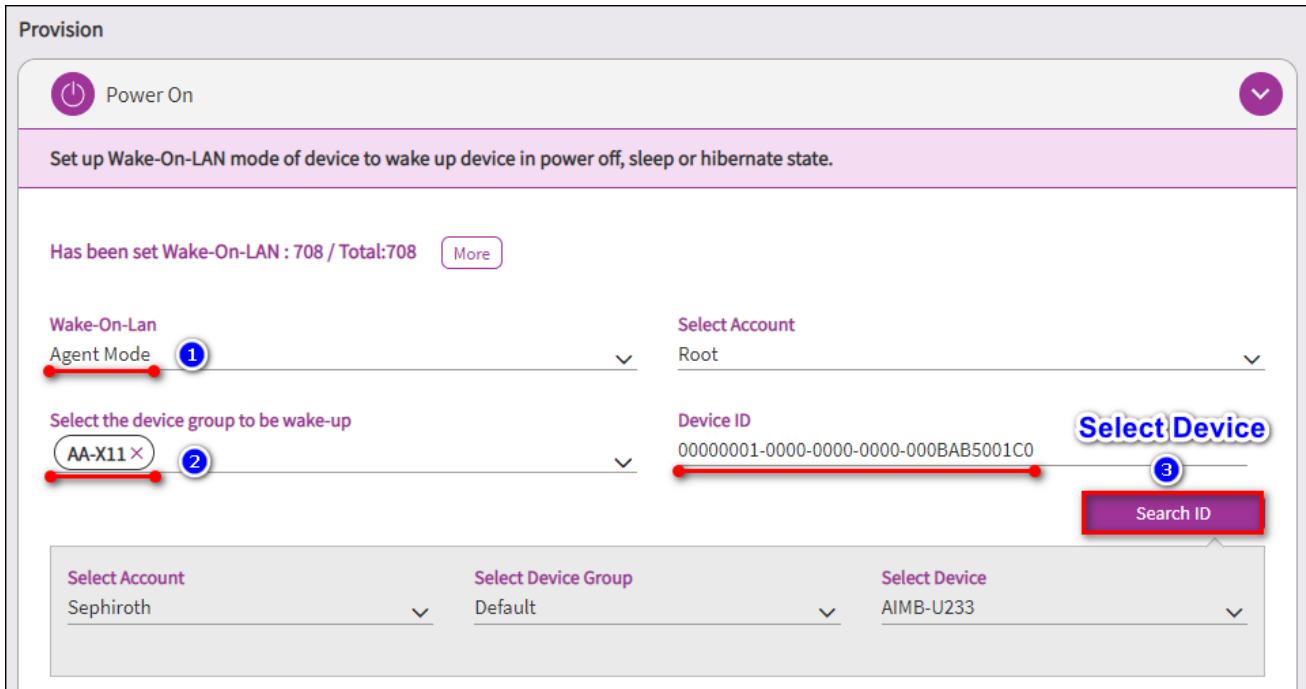
### ❖ Power On

To power your device up, you might to configure the mode for your device. The mechanism is based on Wake-on-LAN to send magic packet to your device. There is a limitation on “**Direct Mode**”, the DeviceOn server and edge device must be on the same network.

This screenshot shows the "Power On" configuration page:

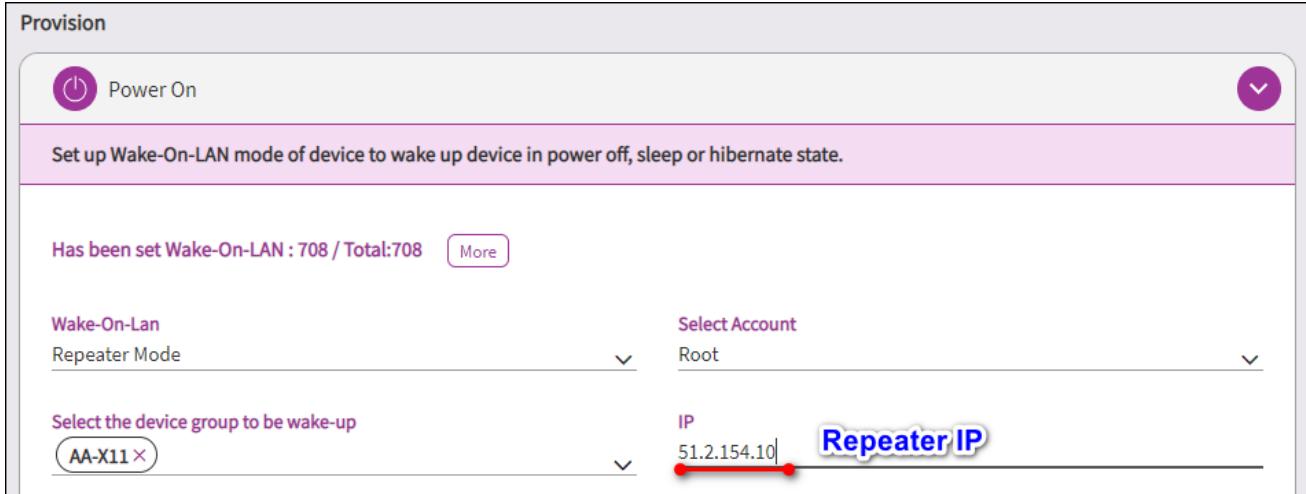
- Power On** button
- Set up Wake-On-LAN mode of device to wake up device in power off, sleep or hibernate state.**
- Has been set Wake-On-LAN : 708 / Total:708** (with a "More" link)
- Wake-On-Lan** settings:
  - Direct Mode (highlighted with a red arrow)
  - Select Account: Root (highlighted with a red circle labeled 2)
  - Select the device group to be wake-up: AA-X11 (highlighted with a red circle labeled 3)
- Device Groups**: Groups Amount : 1
  - Root - AA-X11 (highlighted with a red circle)
  - Root - AA-X11 (link)
  - ARK-DS520-PC
  - A-UTC-532
  - IrisNB-Test

However, through the “**Agent Mode**” or “**Repeater**” could overcome the limitation. You need to pick-up a device that **always on** and on the same network with other devices.



The screenshot shows the 'Provision' section of the WISE-PaaS interface. Under 'Wake-On-Lan', 'Agent Mode' is selected (marked with a red arrow and circled with a blue border). Below it, 'AA-X11' is selected in the 'Select the device group to be wake-up' dropdown (also marked with a red arrow and circled with a blue border). To the right, 'Select Account' is set to 'Root' and 'Device ID' is set to '00000001-0000-0000-0000-BAB5001C0'. A large red arrow points from the 'Device ID' field to a 'Search ID' button. At the bottom, there are three more dropdowns: 'Select Account' (Sephiroth), 'Select Device Group' (Default), and 'Select Device' (AIMB-U233).

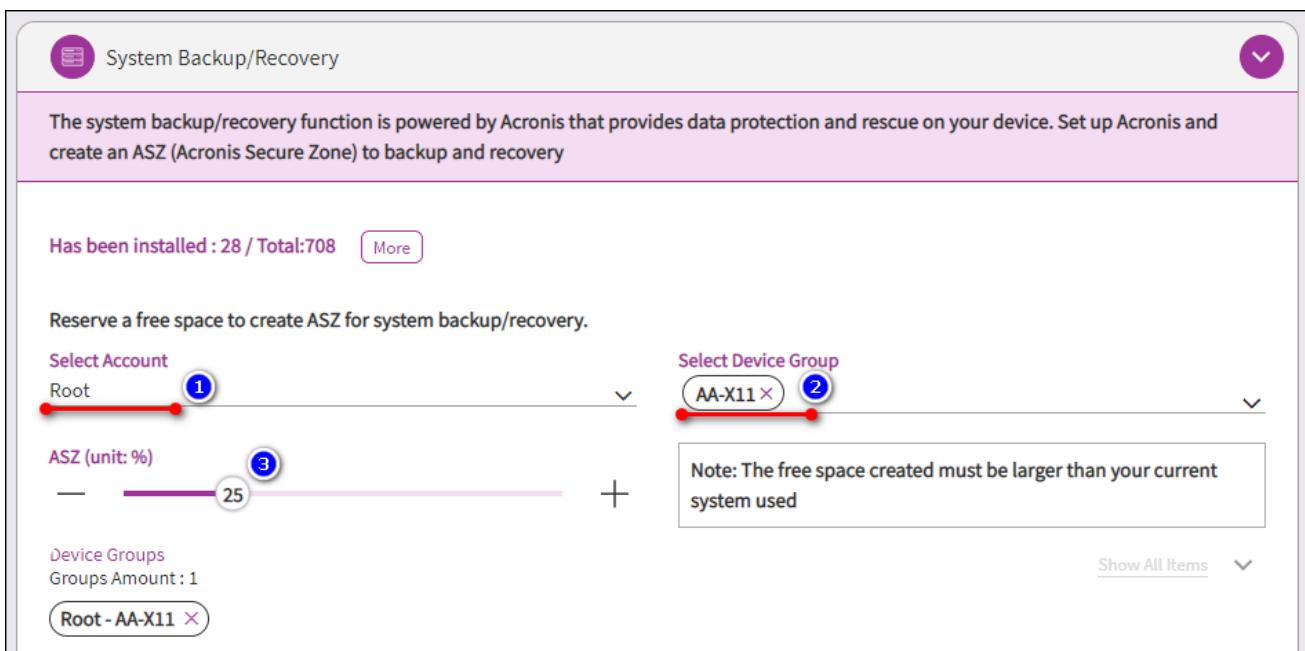
For **Repeater** mode, not only enter your repeater IP, but set your repeater to allow port forwarding (uses UDP port 7 and 9) and permit the packet to be broadcast to the entire LAN.



The screenshot shows the 'Provision' section for Repeater Mode. 'Repeater Mode' is selected in the 'Wake-On-Lan' dropdown. In the 'Select the device group to be wake-up' dropdown, 'AA-X11' is selected. To the right, 'Select Account' is set to 'Root'. Below the account selection, the 'IP' field contains '51.2.154.10' and is highlighted with a red arrow and a blue 'Repeater IP' label.

#### ❖ System Backup/Recovery

Select the free space size to create Acronis Secure Zone (Hidden Partition) to backup system partition. The free space size must larger than system used.



System Backup/Recovery

The system backup/recovery function is powered by Acronis that provides data protection and rescue on your device. Set up Acronis and create an ASZ (Acronis Secure Zone) to backup and recovery

Has been installed : 28 / Total:708 [More](#)

Reserve a free space to create ASZ for system backup/recovery.

Select Account: Root (1)

Select Device Group: AA-X11 (2)

ASZ (unit: %): 25 (3)

Note: The free space created must be larger than your current system used

Show All Items

## ❖ System Protection

Select the device group to install.



System Protection

McAfee Solidcore adopts whitelisting mechanism to prevent your device attacked from unknown malware by allowing only known-good whitelisted applications to run.

Has been installed : 17 / Total:708 [More](#)

Select Account: Root (1)

Select Device Group: AA-X11 (2)

## ● Anomaly Detection

DeviceOn ADS is one of the service that combines anomaly detection algorithm and DeviceOn function. It not only gives customers the high accuracy identification of the error message when advertising is getting interruption but also provides the IoT device remote monitoring and management. Leverage with Azure Custom Vision to continually train the algorithm in order to overcome various errors pop-up under real field.

**Before to detect your anomaly screen, please make sure your device is logged in to capture screen status.**

## WISE-DeviceOn

Overview

Device

List

Monitoring

Remote Control

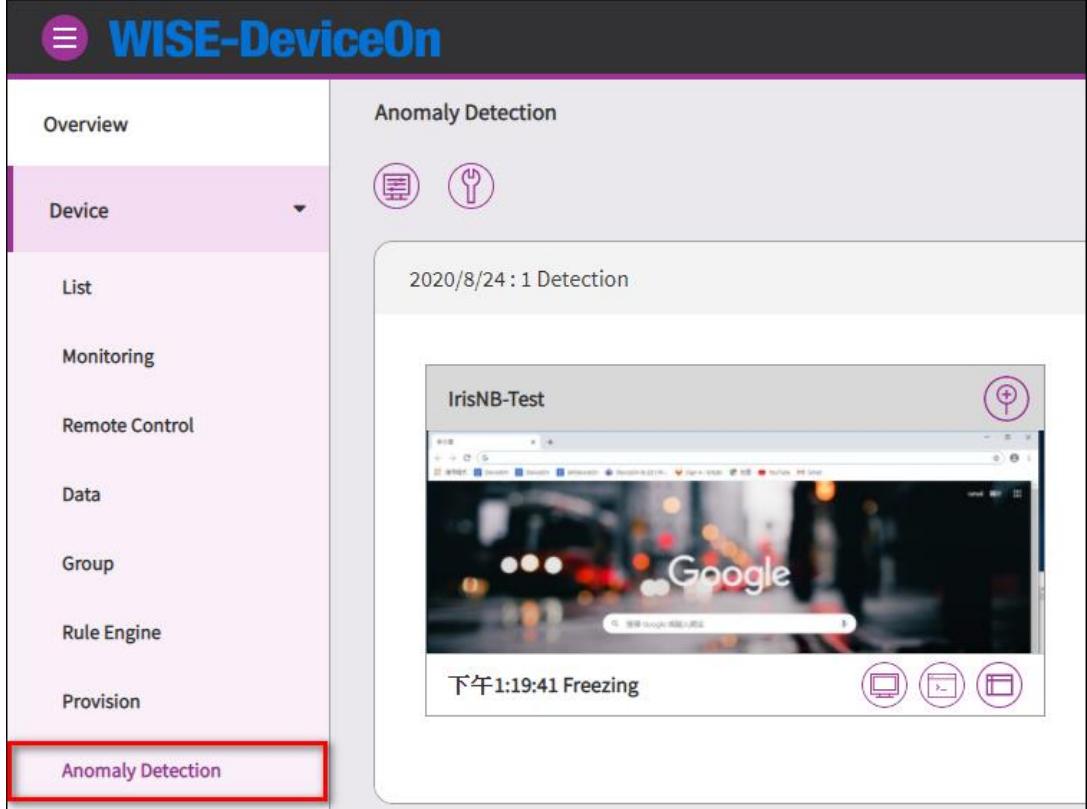
Data

Group

Rule Engine

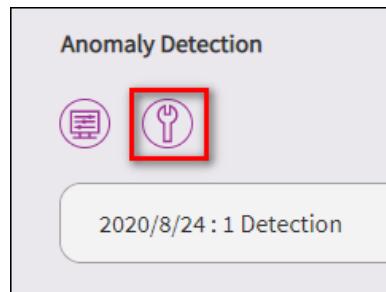
Provision

Anomaly Detection



The screenshot shows the WISE-DeviceOn web interface. On the left, there's a sidebar with various menu items like Overview, Device, List, Monitoring, etc. The 'Anomaly Detection' item is highlighted with a red box. The main area is titled 'Anomaly Detection' and contains two icons: a gear and a wrench. Below that is a timestamp '2020/8/24 : 1 Detection'. Underneath is a thumbnail image of a browser window titled 'IrisNB-Test' showing a blurred Google search result. At the bottom of the main panel, it says '下午1:19:41 Freezing' and has three small circular icons.

Click on the configuration icon to enter API URL and Key.



### Configuration

**API Url**

**API Key**

**Maximum Retention Days(When it is 0, Keep 30 minutes)**

**Interval (Second)**

**Enable** 

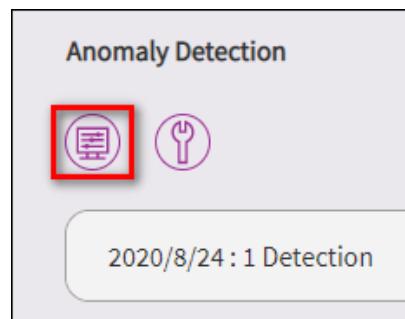
Window PopUp     Freezing Wanted

**Hint:**  
 If want to get API key, [Contact us](#)

**Save** **CANCEL**

- ✧ API URL & Key: Please contact us for AI machine URL and Key, otherwise, deploy total package from [Azure Marketplace](#).
- ✧ Maximum Retention Days: Maximum retention days for the warning images.
- ✧ Interval: Minimal interval to detect devices screen.
- ✧ Window Popup: Enable to detect popup window on the devices.
- ✧ Freezing Wanted: Enable to detect freezing window on the devices.

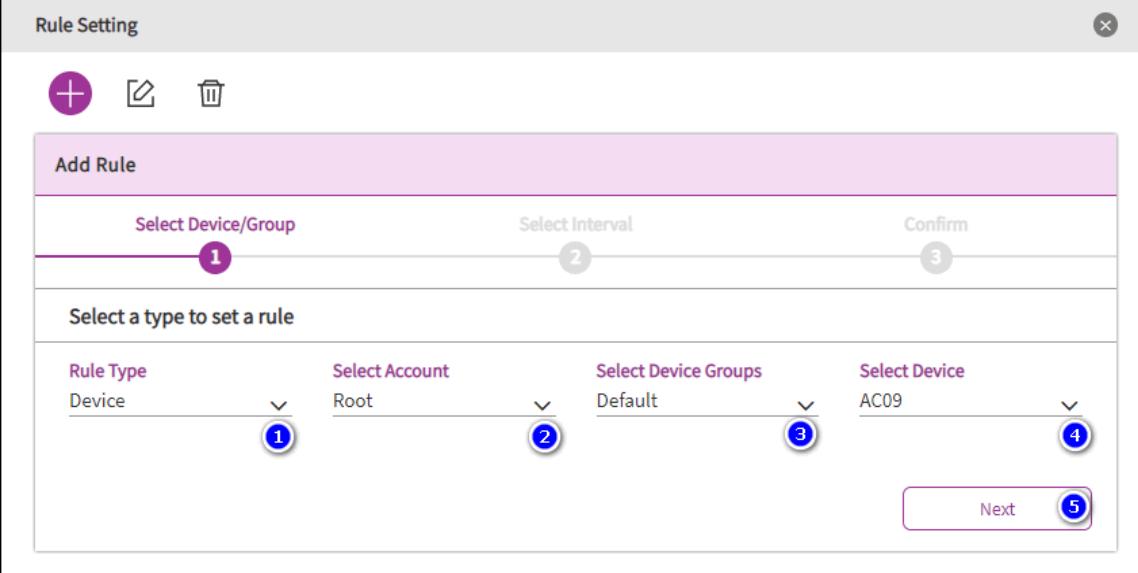
Click on the rule icon to enable detection rules.



### Rule Setting

			2 Set << < 1 /1 > >>
ENABLE	DEVICE/GROUP	INTERVAL (SECOND)	
	Device : ARK-DS520-PC	40	
	Device : IrisNB-Test	20	

Click on the “+” icon to create anomaly rule, select to your device or device group



Rule Setting

Add Rule

Select Device/Group **1**

Select Interval **2**

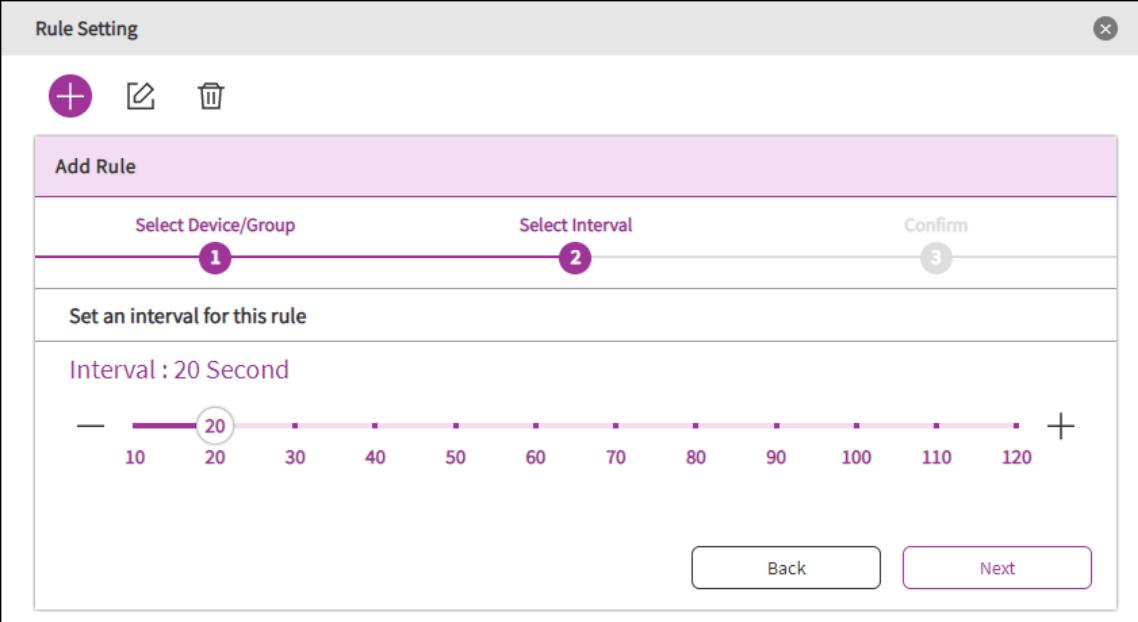
Confirm **3**

Select a type to set a rule

Rule Type: Device **1** Select Account: Root **2** Select Device Groups: Default **3** Select Device: AC09 **4**

Next **5**

Set the detection interval of the rule, the interval cannot be less than the configuration.



Rule Setting

Add Rule

Select Device/Group **1**

Select Interval **2**

Confirm **3**

Set an interval for this rule

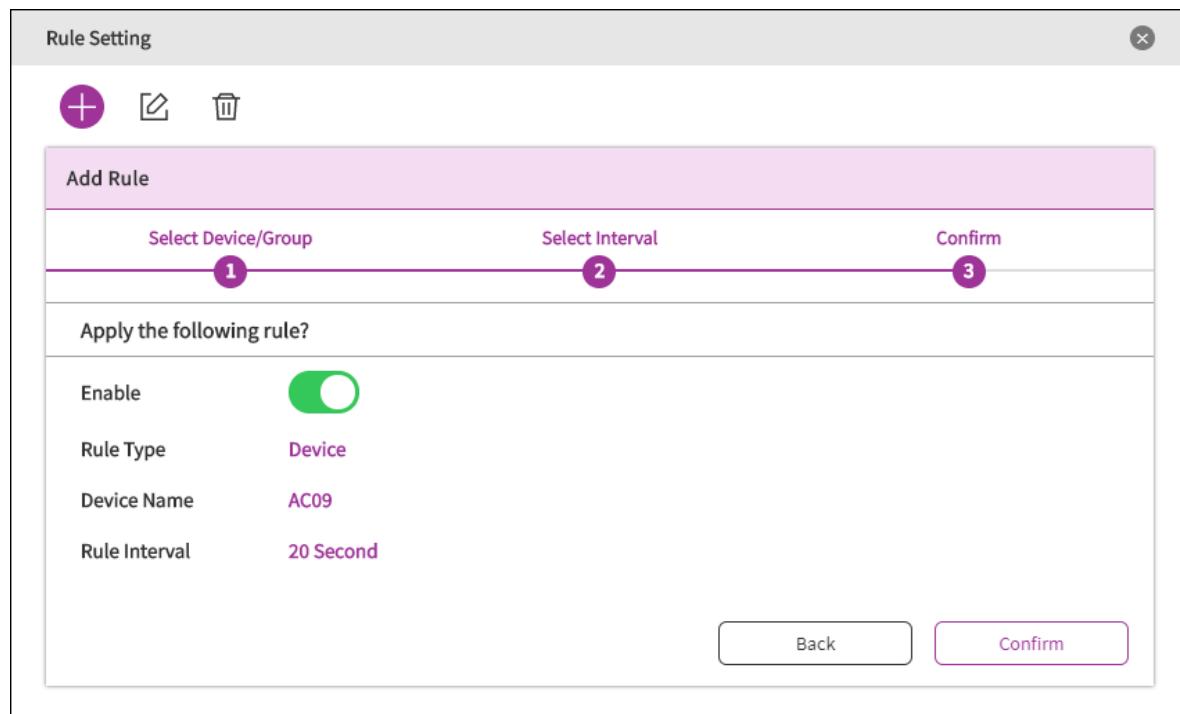
Interval : 20 Second

20

10 20 30 40 50 60 70 80 90 100 110 120

Back Next

Confirm the rule and enable it.



### 3.3.3 Account Management

The first step to manage device is login to DeviceOn, therefore, you could start to invite, edit other accounts on this page. The user profile show your account information and person alert service, such as Email, SMS, WeChat, LINE, Telegram, Slack and Teams status.

Role	Account Name	Email Notify	SMS Notify
Super admin	Root	Off	Off

Email	First Name	WeChat Notify	LINE Notify
root@advantech.com.tw	Dylan	Off	Off

Last Name	Phone (optional)	Telegram Notify	Microsoft-Teams Notify
Root	Not Set	On	On

Login At	Created At	Slack Notify
2020/8/24 14:18:20	2017/1/1 08:00:00	On

To change your password, please click on the “Change Password” on the profile.

### User Profile & Notification Setting

Role Super admin	Account Name Root	 Email Notify Off	 SMS Notify Off
Email root@advantech.com.tw	First Name Dylan	 WeChat Notify Off	 LINE Notify Off
Last Name Root	Phone (optional) Not Set	 Telegram Notify On	 Microsoft-Teams Notify On
Login At 2020/8/24 14:18:20	Created At 2017/1/1 08:00:00	 Slack Notify On	<span style="border: 2px solid red; padding: 2px;">Change Password</span>

### Change Password

**Old Password**

The old password field is required.  

---

**New Password**

---

**Confirm Password**

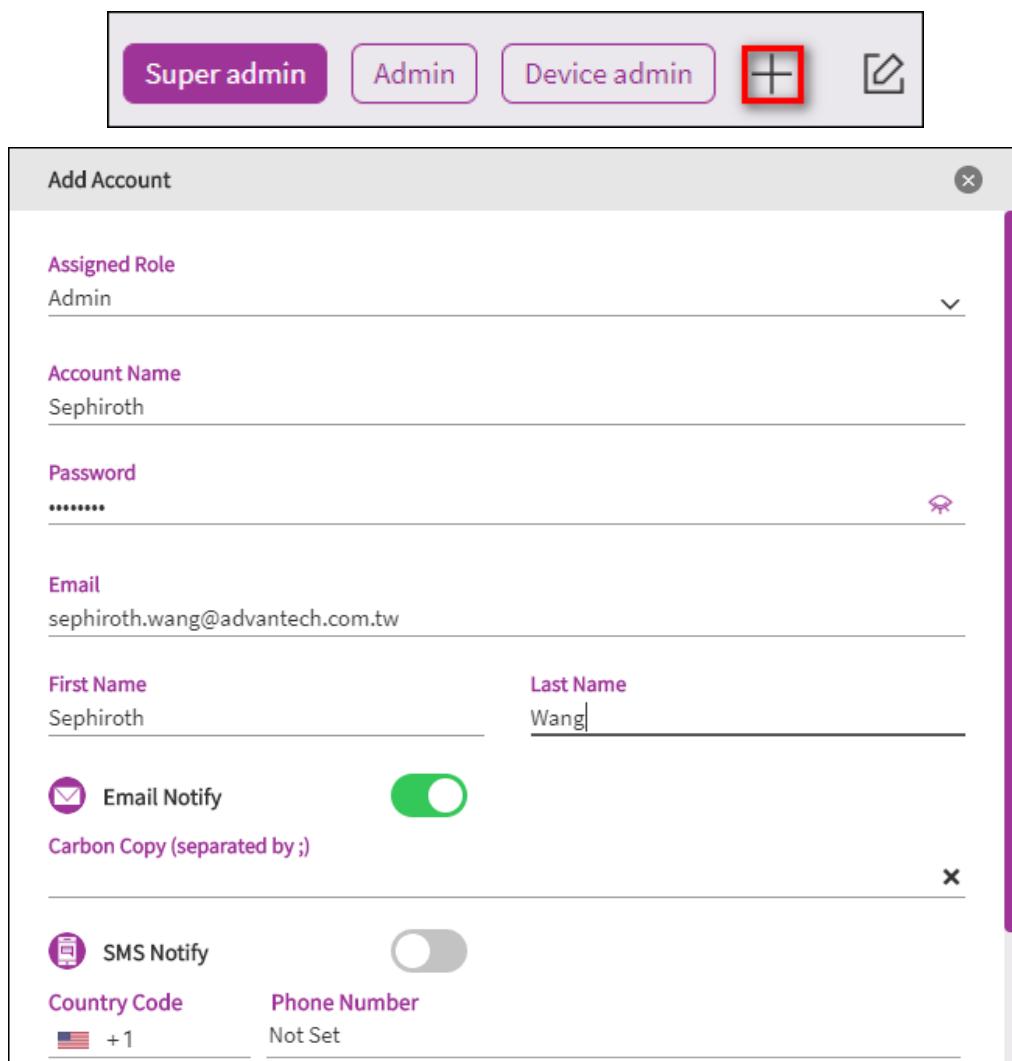
---

Change Password CANCEL

Every account belongs to a role, you could switch the tab to invite/view and edit account. There are 3 roles in the DeviceOn system. One is “**Super Admin**”, only one account in the system belongs to “**Super Admin**”. The other role is “**Admin**” and “**Device Admin**”. For detail role permission, please reference Section 7.1.

Super admin	Admin	Device admin	+		Keyword Search		1 Set	<<	<	1	/1	>	>>
DISABLED	NAME	SOURCE	EMAIL	FULL NAME	PHONE								
	Root	More	DeviceOn	root@advantech.com.tw	Dylan Root	Not Set							

Click on the icon to “Add Account”



Add Account

Assigned Role  
Admin

Account Name  
Sephiroth

Password  
\*\*\*\*\*

Email  
sephiroth.wang@advantech.com.tw

First Name  
Sephiroth

Last Name  
Wang

Email Notify

Carbon Copy (separated by ;)

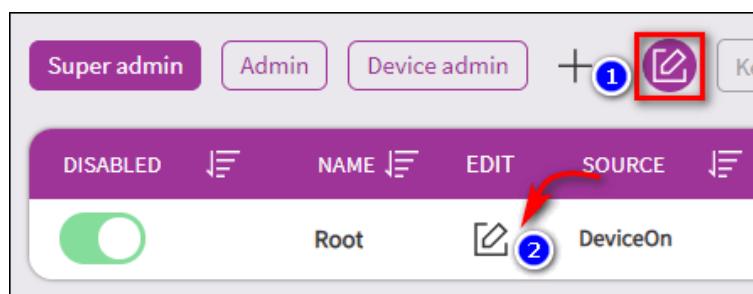
SMS Notify

Country Code  
+1

Phone Number  
Not Set

Enter your account, role, password, etc. to create an account. If the user would to receive notify from device, system alert, please enable these alert services on “Mail”, “SMS”, “WeChat”, “LINE”, “Telegram”, “Teams” and “Slack”. These alert services are personal setting, please make sure the “Setting -> Notification” is configured, enabled on DeviceOn System.

Click on the icon to “Edit” or “Disable” account.



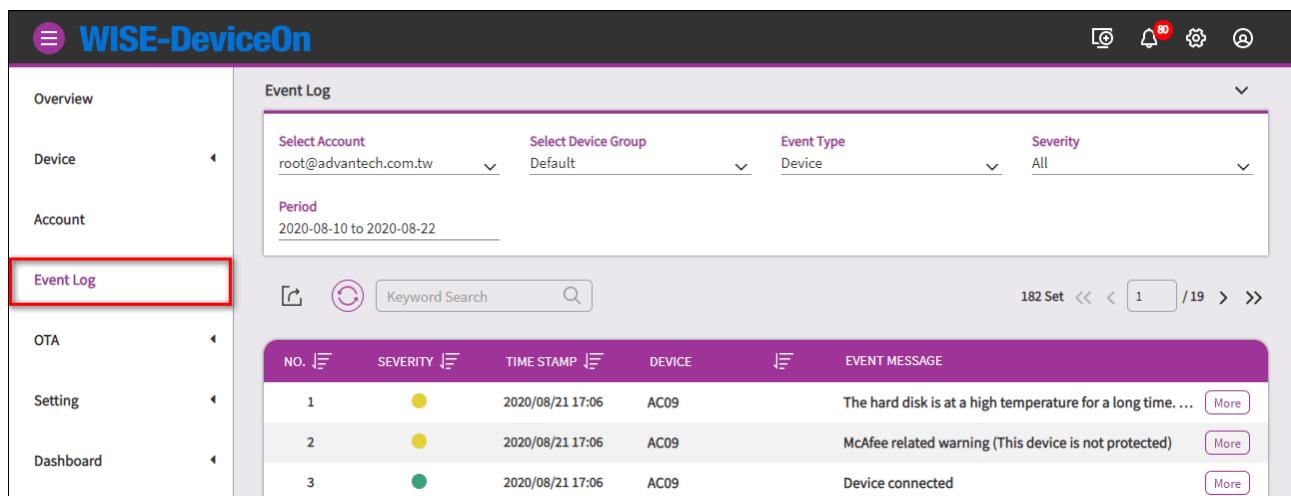
### 3.3.4 Event Logs

Device management is complex with device log and user behavior. Logging data can provide insights about your devices and help you:

- Troubleshoot past problems or prevent potential ones
- Improve device health or maintainability
- Real-time alert through 3<sup>rd</sup> notification

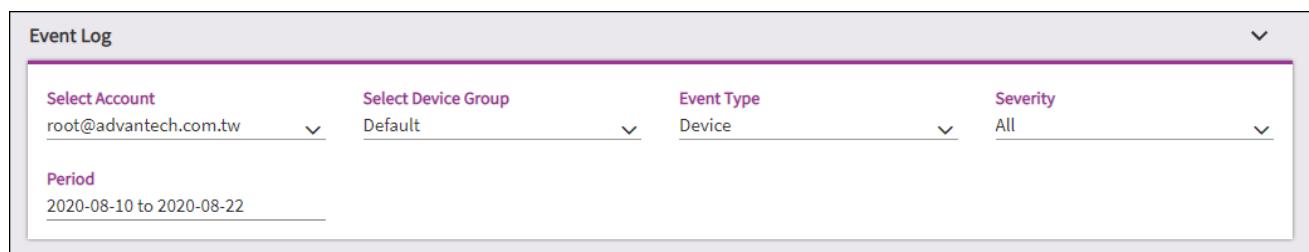
DeviceOn logs are categorized into the following types:

- Operation log provide information about DeviceOn resource CREATE, UPDATE and DELETE operation, like set device power off, update device name or delete account.
- Device log provide information about events raised as device side resources, like connected, disconnected, over the threshold,
- System log provide information about analyzed; scheduling event/alert that have been processed on DeviceOn server. Example of this type are queue buffer alerts where server has processed and measured IoT Hub queue and provides concise alerts.



NO.	SEVERITY	TIME STAMP	DEVICE	EVENT MESSAGE
1	Yellow	2020/08/21 17:06	AC09	The hard disk is at a high temperature for a long time.... <a href="#">More</a>
2	Yellow	2020/08/21 17:06	AC09	McAfee related warning (This device is not protected) <a href="#">More</a>
3	Green	2020/08/21 17:06	AC09	Device connected <a href="#">More</a>

There are three types of Event Logs as mentioned above and each event log with different severity, **Information**, **Warning** and **Error**. Through the filter to find your device log.



Click on the icon to refresh event log by manual.



Click on export icon to export devices that in the table as CSV file.

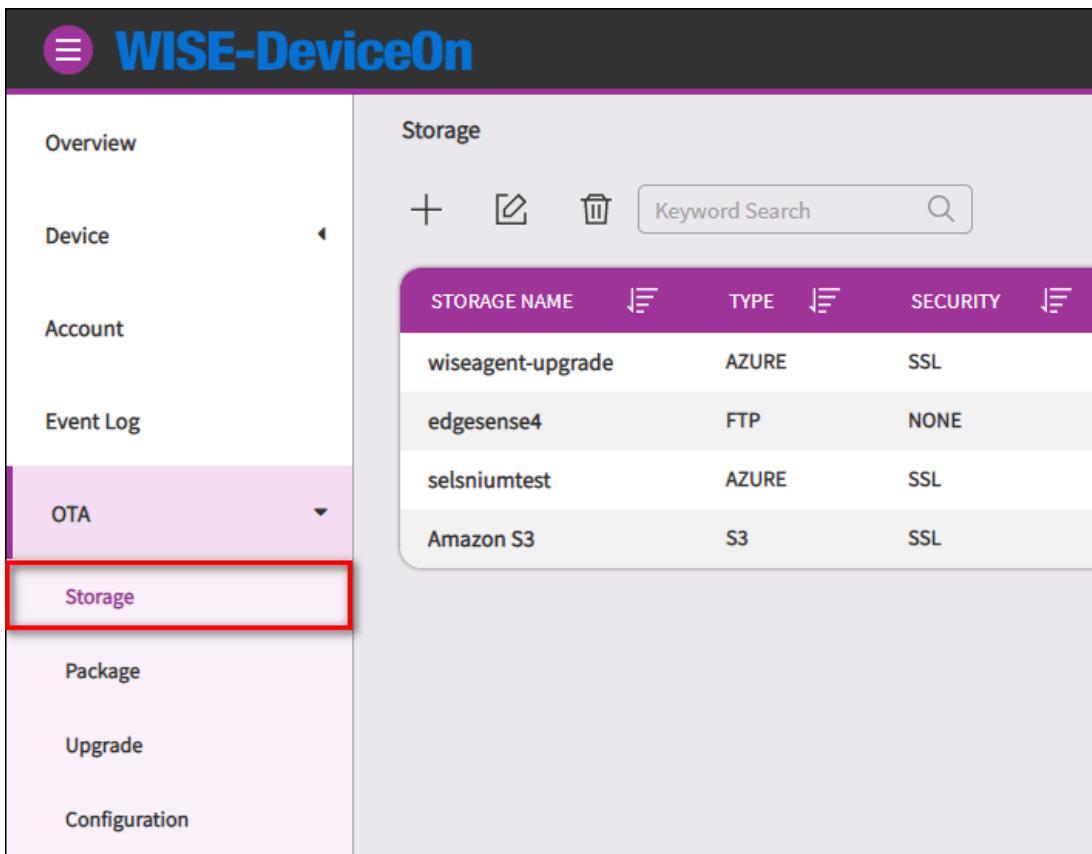


### 3.3.5 OTA (Remote Provisioning)

OTA (Over-The-Air) is one of powerful feature DeviceOn provides. Users can deploy **software** packages, **configuration**, **Windows QFE** (Quick Fix Engineering), **Advantech BIOS** update onto a device remotely, or even many devices broadly.

- Storage

There is a default Azure blob storage called “**wiseagent-upgrade**”, host by Advantech DeviceOn team. If there is a new version of WISE-Agent released, all of user could get the update and upgrade their devices. The storage is read only cannot upload user’s OTA package.



STORAGE NAME	TYPE	SECURITY
wiseagent-upgrade	AZURE	SSL
edgesense4	FTP	NONE
selsniumtest	AZURE	SSL
Amazon S3	S3	SSL

Click on add icon to add new storage.

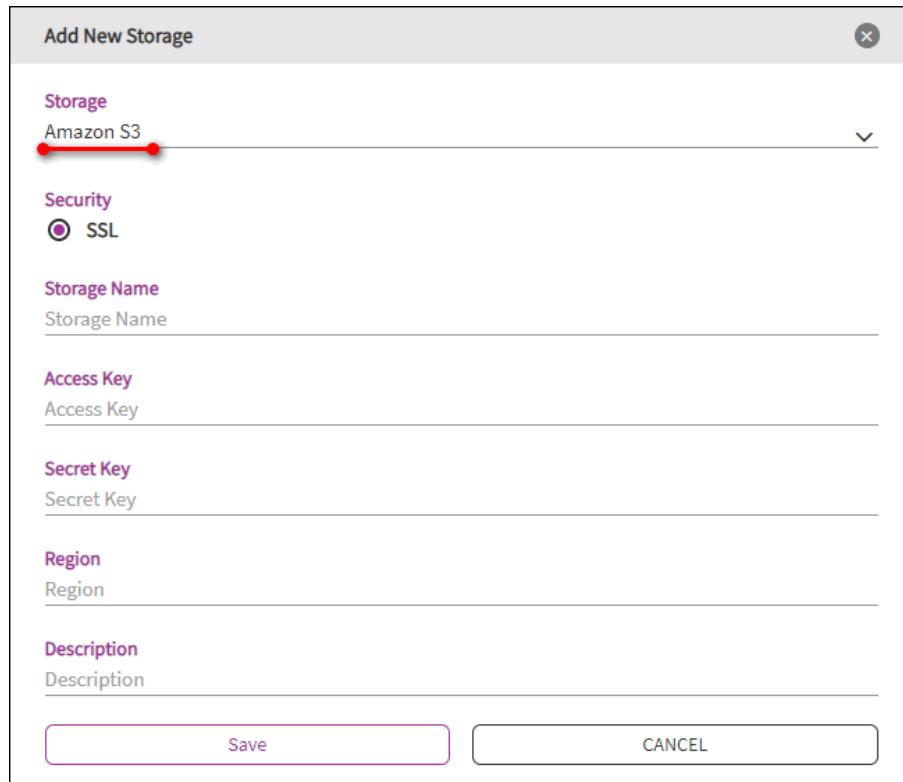


For cloud storage, DeviceOn provide “**Amazon S3**”, “**S3 Compatible**”, “**Azure Blob**” and traditional

## FTP services.

### [Amazon S3]

You could create and get Access Key, Secret Key from Amazon Web service.



Add New Storage

**Storage**  
Amazon S3

**Security**  
 SSL

**Storage Name**  
Storage Name

**Access Key**  
Access Key

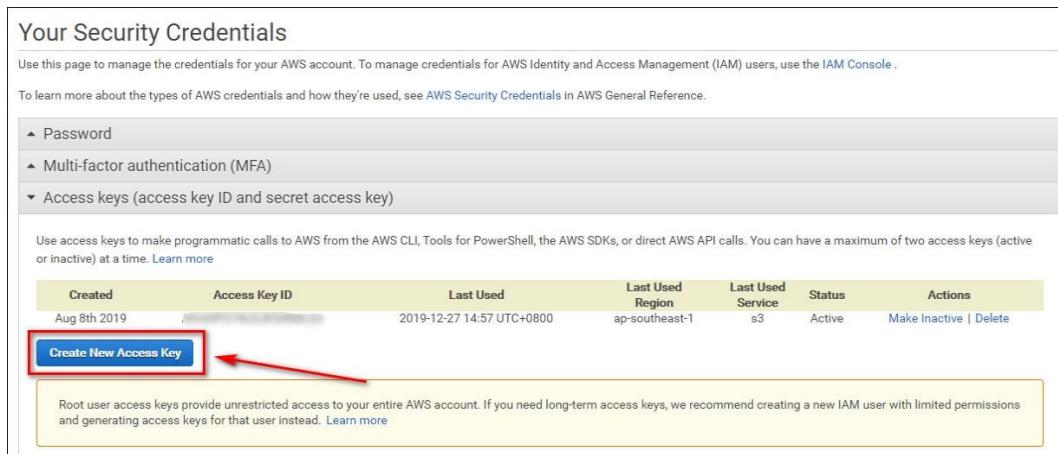
**Secret Key**  
Secret Key

**Region**  
Region

**Description**  
Description

Save CANCEL

- ✧ Storage Name: Your storage name, define by yourself.
- ✧ Region: Region of AWS S3
- ✧ Access Key: Access Key for AWS S3
- ✧ Secret Key: Secret Key for AWS S3



Your Security Credentials

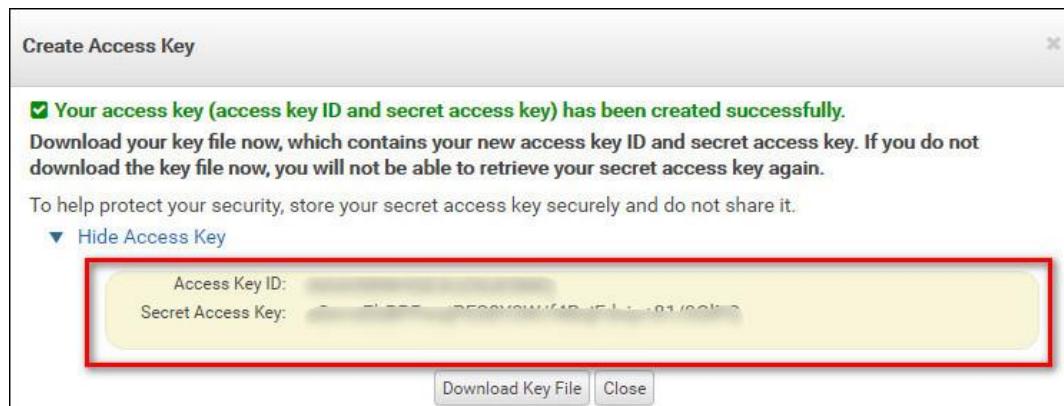
Use this page to manage the credentials for your AWS account. To manage credentials for AWS Identity and Access Management (IAM) users, use the [IAM Console](#).

To learn more about the types of AWS credentials and how they're used, see [AWS Security Credentials](#) in AWS General Reference.

Created	Access Key ID	Last Used	Last Used Region	Last Used Service	Status	Actions
Aug 8th 2019	[REDACTED]	2019-12-27 14:57 UTC+0800	ap-southeast-1	s3	Active	<a href="#">Make Inactive   Delete</a>

**Create New Access Key**

Root user access keys provide unrestricted access to your entire AWS account. If you need long-term access keys, we recommend creating a new IAM user with limited permissions and generating access keys for that user instead. [Learn more](#)



### [S3 Compatible]

The setting similar to Amazon, only **endpoint** must be configured to yourself.

Add New Storage

**Storage**  
S3 Compatible

**Security**  
 NONE    SSL

**Storage Name**  
Storage Name

**Access Key**  
Access Key

**Secret Key**  
Secret Key

**Endpoint**  
Endpoint

**Description**  
Description

Save | CANCEL

### [Azure Blob]

For Azure Blob, supports two mechanisms to access, one is “**Storage Account**” and “**Access Key**” with full access permission of container. The other is “**container**” SAS token generated via [Microsoft Azure Storage Explorer](#).

Add New Storage

**Storage**  
Azure Blob

**Security**  
 SSL

**Storage Name**  
Storage Name

**SAS**  
 Use Shared Access Signature

**Account Name**  
Account Name

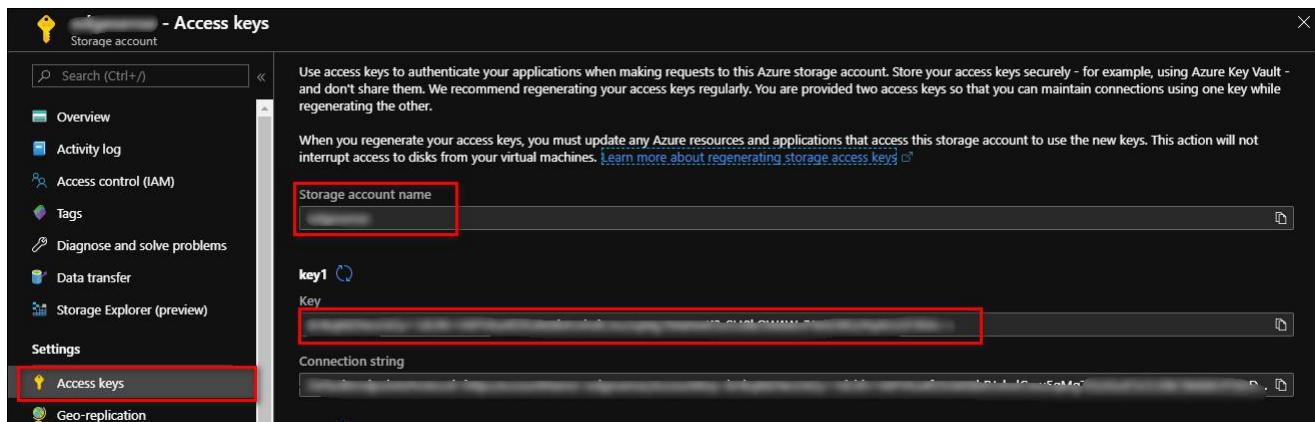
**Access Key**  
Access Key

**Endpoint Suffix**  
core.windows.net (AzureCloud)

**Description**  
Description

**Save** **CANCEL**

Through Azure portal to get your **Storage Account** and **Access Key**.



- Access keys

Storage account

Search (Ctrl+ /)

- Overview
- Activity log
- Access control (IAM)
- Tags
- Diagnose and solve problems
- Data transfer
- Storage Explorer (preview)
- Settings
- Access keys**
- Geo-replication

Use access keys to authenticate your applications when making requests to this Azure storage account. Store your access keys securely - for example, using Azure Key Vault - and don't share them. We recommend regenerating your access keys regularly. You are provided two access keys so that you can maintain connections using one key while regenerating the other.

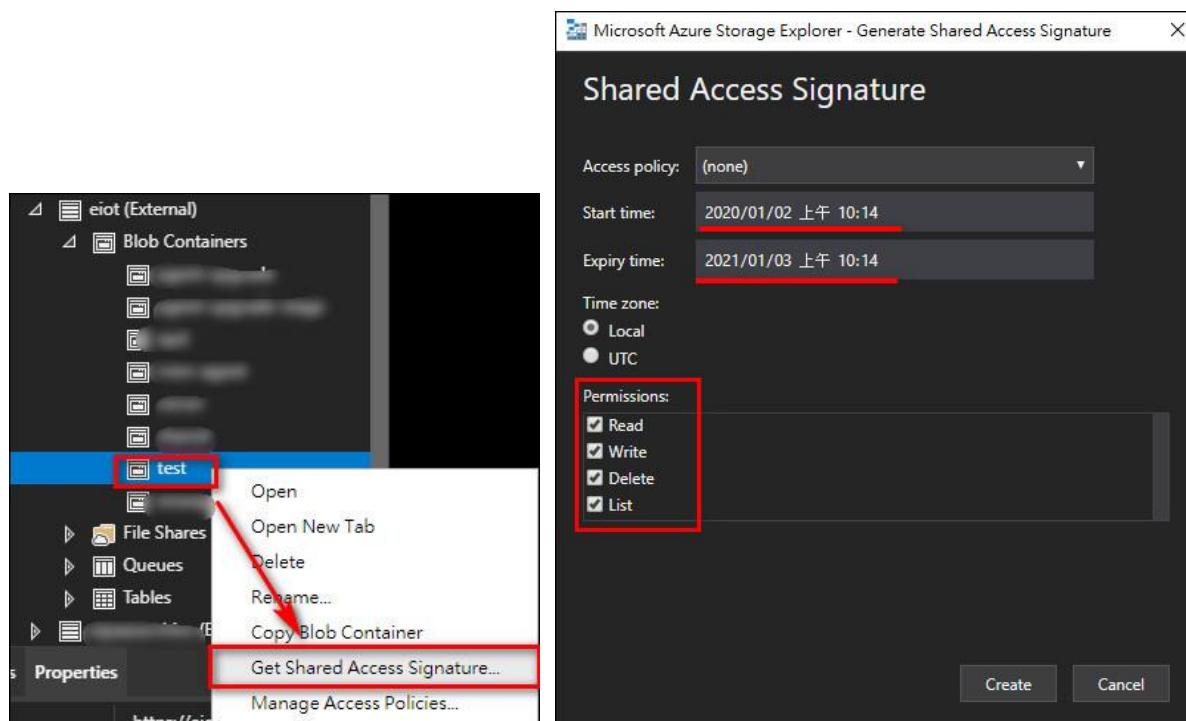
When you regenerate your access keys, you must update any Azure resources and applications that access this storage account to use the new keys. This action will not interrupt access to disks from your virtual machines. [Learn more about regenerating storage access keys](#)

Storage account name: [REDACTED]

key1  
Key: [REDACTED]

Connection string: [REDACTED]

Get **container's SAS token** via Azure Storage Explore, please make sure your permission (Read, Write, Delete, List) and valid period (Start and Expiry time)



## [FTP]

For FTP, you might setup another FTP server with security and account, password.

**Add New Storage**

**Storage**  
FTP

**Security**  
 NONE    FTPS    FTPES

**Storage Name**  
Storage Name

**Domain**  
Domain

**Port (Range: 0 to 65535)**  
21

**Account Name**  
Account Name

**Password**  
Password

**CMC (Client Max Connections)**  
30

**SMC (Server Max Connections)**  
5

**Root Path**  
/

**Description**  
Description

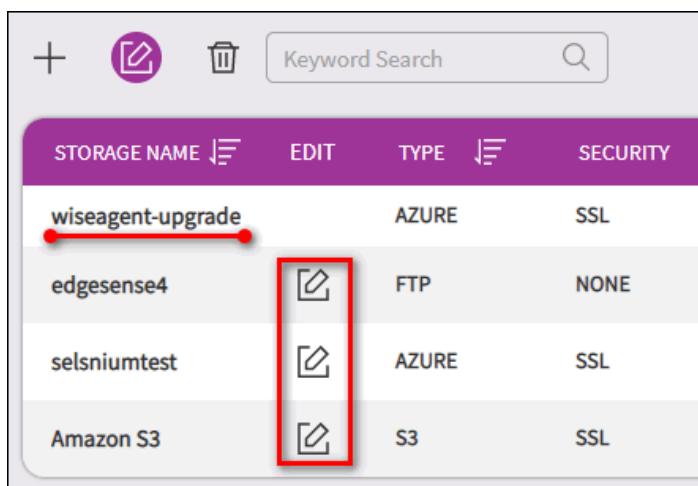
**Save** **CANCEL**

- ✧ **Security:** Leave it as “**NONE**”, the default value. If your FTP server running on FTPS protocol, pick “**FTPS**”.
- ✧ **SOTRAGE NAME:** Enter “**MyFTP**”.
- ✧ **DOMAIN:** Enter the FQDN of your FTP server, or its IP address.
- ✧ **PORT:** Should be **21** if the FTP server runs on a standard port number.
- ✧ **ACCOUNT NAME:** A valid username that can connect to the FTP server, and upload files onto the server as well.
- ✧ **PASSWORD:** The password to login.
- ✧ **CMC/SMC:** Maximum Client & Server Connection.
- ✧ **ROOT PATH:** FTP server access path (root folder)
- ✧ **DESCRIPTION:** It's optional information.

Click on edit icon to adjust a storage.



You could edit yourself storage, but the default storage cannot.



STORAGE NAME	EDIT	TYPE	SECURITY
wiseagent-upgrade		AZURE	SSL
edgesense4		FTP	NONE
selsniumtest		AZURE	SSL
Amazon S3		S3	SSL

**Edit Storage**

**Storage**  
Azure Blob

**Security**  
 SSL

**Storage Name**  
seleniumtest

**SAS**  
 Use Shared Access Signature

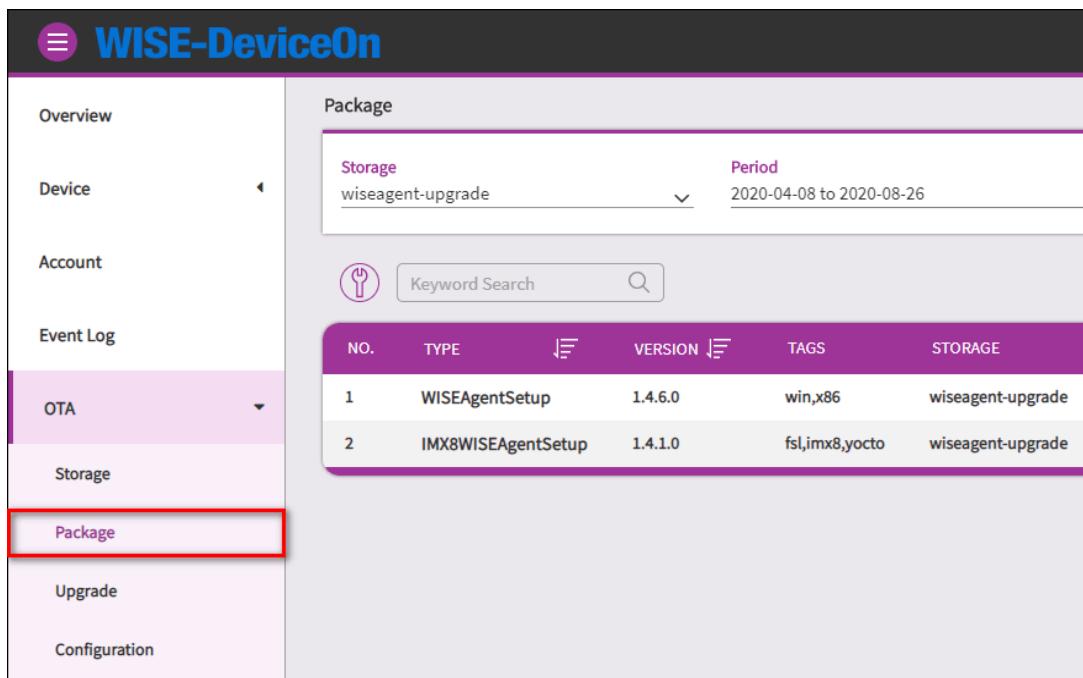
**SAS URI**  
[REDACTED]

**Description**  
Description

**Save** **CANCEL**

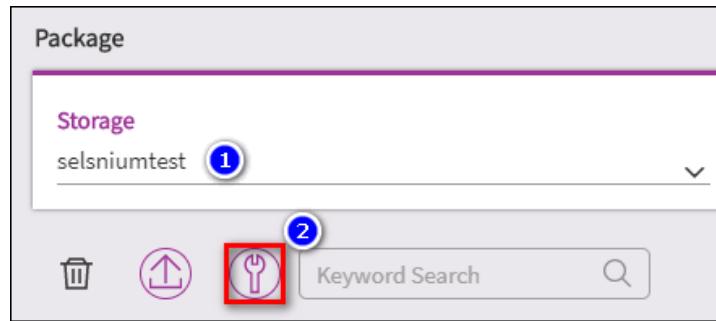
- Package

View and edit OTA package on select storage, user could edit, delete upload their package to selected storage, but default storage (**wiseagent-upgrade**) cannot. To ensure the security and data format on OTA package, user should wrap their software, firmware via DeviceOn toolkit. The toolkit not only command-line tool but support online UI mechanism.



NO.	TYPE	VERSION	TAGS	STORAGE
1	WISEAgentSetup	1.4.6.0	win,x86	wiseagent-upgrade
2	IMX8WISEAgentSetup	1.4.1.0	fsl,imx8,yocto	wiseagent-upgrade

Select to your storage and click on the toolkit icon to start to warp your OTA package.



Prepare your software, configuration and installation script first, gives below information. The operation system and architecture might be different. Therefore, to determine the OTA package be deployed on which devices, please pick-up the “**Tag Name**” on “**Supported Arch**”. All “**Tags**” must match with devices, the OTA package will be executed. For example, there is two devices (ARK-1123, UTC-520) with different tag attribute. The ARK-1123 device is Windows based and support x64 and x86 OTA package. The UTC-520 is Ubuntu system also support x64, x86.

- ARK-1123 (Tags): win, x64, x86
- UTC-520 (Tag): ubuntu, x64, x86

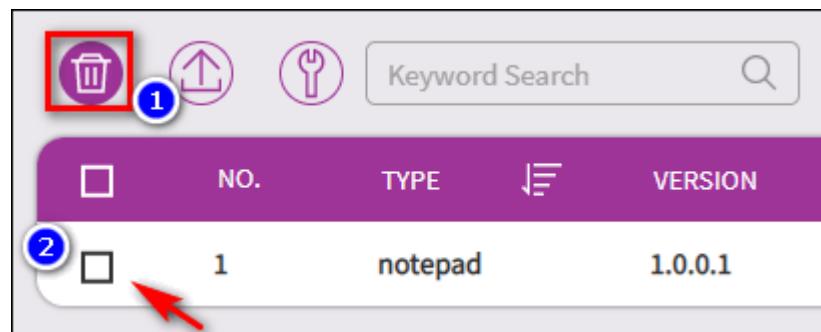
If your OTA package tags are “**win**”, “**x64**”, “**x86**”, the package only support and executed on “ARK-1123”. Otherwise, if the tag is “**x64**”, both devices could be affected.

- ✧ **Package Type:** Name of package
- ✧ **Package Version:** Version of Package
- ✧ **Supported Arch:** Select “**Tag Name**” from of device (Account -> Device Group -> Device)
- ✧ **Deploy File:** Installation script (batch file or shell script)
- ✧ **Storage: Upload to storage or download**
- ✧ **Advanced options:** Reboot or run the script after deployed.

**Package Toolkit**

<b>Package Type</b>	notepad
<b>Package Version</b>	1.0.0.1
<b>Account</b>	Root
<b>Device Group</b>	Default
<b>Device</b>	AC09 
<b>Tags</b>	<input checked="" type="checkbox"/> x64 <input checked="" type="checkbox"/> x86 <input checked="" type="checkbox"/> win
<b>Source Directory</b>	otademo 
<b>Deploy Script</b>	installNotepad.bat
<b>Storage</b>	LocalStorage
<input type="checkbox"/> Advanced options	

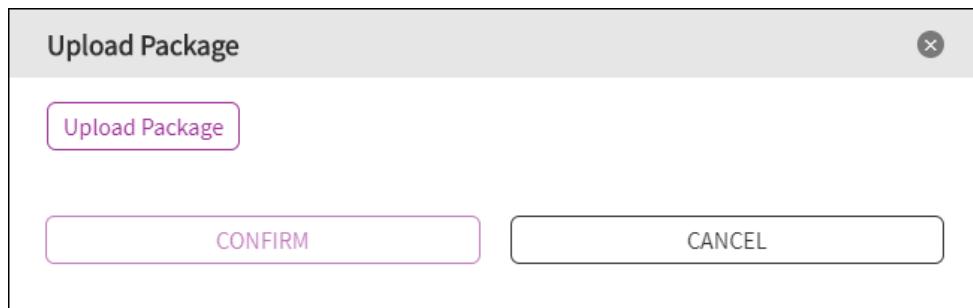
Click on the delete icon to delete your OTA package.



<input type="checkbox"/>	NO.	TYPE	VERSION
 1	1	notepad	1.0.0.1

Click on the upload icon to upload your OTA package.



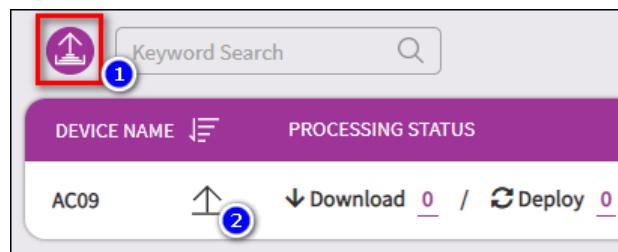


- Upgrade

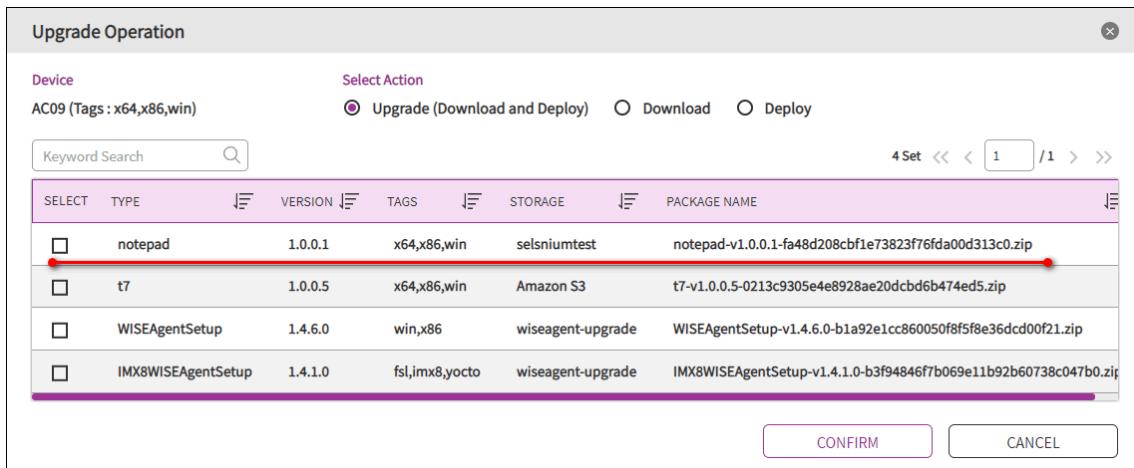
On the upgrade tab, start to select your device or device group and pick-up your OTA package that you upload before. On the device list to configure schedule, check the result status and program list that installed.

The screenshot shows the WISE-DeviceOn web interface. The left sidebar has a navigation menu with options: Overview, Device, Account, Event Log, **OTA**, Storage, Package, **Upgrade** (which is highlighted with a red box), and Configuration. The main content area is titled "Upgrade". It includes a "Select Account" dropdown set to "root@advantech.com.tw", a "select Type" dropdown set to "Device", a "Keyword Search" input field, and a file upload icon. Below this is a table header with columns "DEVICE NAME" and "PROCESSING STATUS". A row for "AC09" is shown with "More" and download/deploy buttons. The "Download" and "Deploy" counts are both 0.

Click on upgrade icon to select OTA package.



Select your package to “**Upgrade**”, “**Download**” or “**Deploy**”. The “**Upgrade**” represents download OTA package from storage and execute (Deploy) immediately. Every package would be kept on device side as “**Upgrade**” or “**Download**”.



To check the deploy status, please click on process icon.

DEVICE NAME	PROCESSING STATUS
AC09	Download 1 / Deploy 0

ACTION	NAME	STAGE	PROGRESS	MESSAGE
download	notepad-v1.0.0.1-fa48d208cbf1e73823f76fda00d313c0.zip	NORMAL	100%	Download task do

To avoid burst download on large number of devices upgrade at the same time, user could add schedule to check and upgrade by schedule.

DEVICE NAME	PROCESSING STATUS	SCHEDULE SETTING
AC09	Download 0 / Deploy 0	Download 3 / Deploy 0

Click on add icon to create a schedule.

Schedule List : AC09

Device Group : Default

①    Keyword Search 

0 Set << < 1 / 0 > >>

Added Schedule			
Package Type	Action Type	Upgrade Mode	Frequency
notepad	<input checked="" type="radio"/> Download <input type="radio"/> Deploy	<input checked="" type="radio"/> Maximum <input type="radio"/> Increment	Daily
Action Start Time	11:45 am	Action End Time	12:15 pm

- ✧ Package Type: Select your OTA package from storage.
- ✧ Action Type: Download or Deploy the package.
- ✧ Upgrade Mode: If the mode is **Max**, the action would download/deploy **the latest version** on the package. Otherwise, if the mode “**Increment**”, The deploy, or download behavior will gradually increase from the lower version to the latest version.
- ✧ Frequency: **Daily, Weekly, Monthly** or **Once** to check.
- ✧ Action Start Time: Check time on start.
- ✧ Action End Time: End time for download, if download exceeds the end time, the action will be terminated.

Click on edit icon to modify, delete OTA schedule.

Device Group : Default

①    Keyword Search 

Type	Package Type
Device	notepad

② 

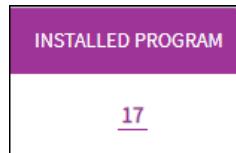
To check deployed software, configure status on device, please click on the numbers.

DEPLOYED SOFTWARE STATUS

Deployed Success 1 / Total Deployed 1

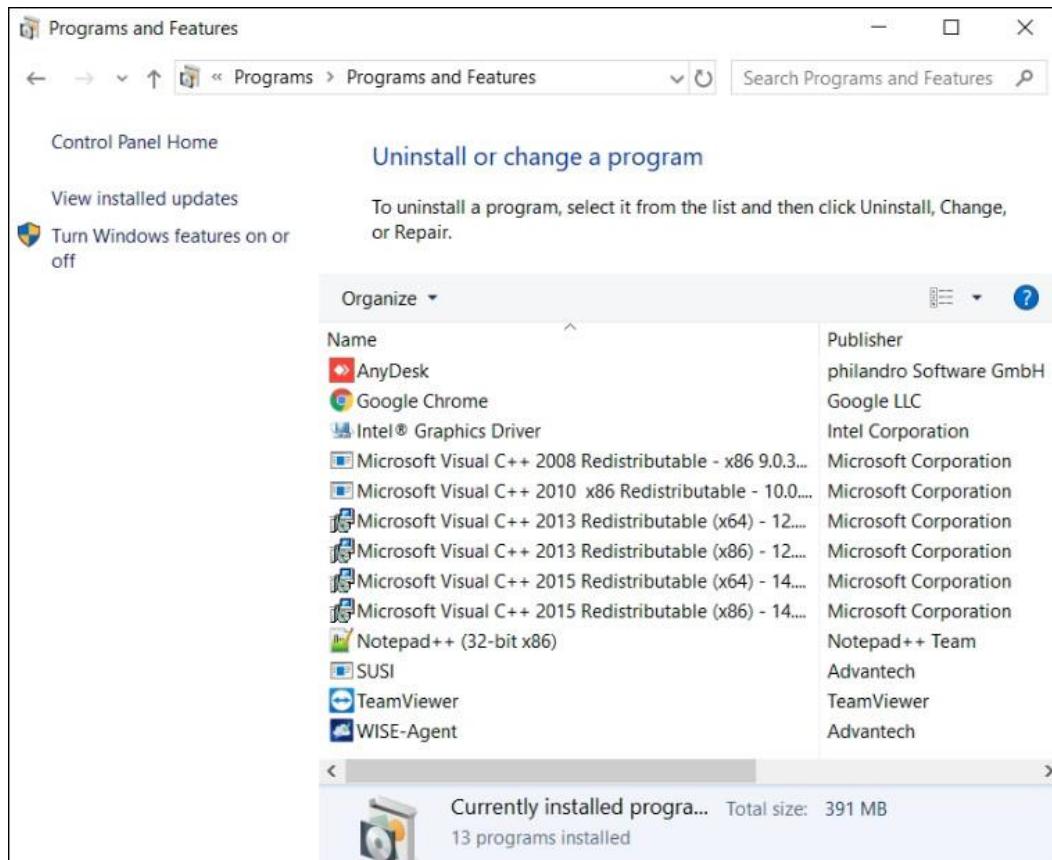
Deployed Software List			
Device : AC09		Keyword Search	1 Set << < 1 /1 > >>
NO.	NAME	VERSION	DEPLOYMENT STATUS
1	notepad	1.0.0.1	✓

Furthermore, user could view the program list on the device. (Windows Only)



Device Program List					
Device : AC09		Keyword Search	21 Set << < 1 /3 > >>		
NO.	DISPLAY NAME	PUBLISHER	INSTALL DATE	ESTIMATED SIZE	DISPLAY VERSION
1	7-Zip 19.00 (x64)	Igor Pavlov	2020318	5082 KB	19.00
2	OWASP Zed Attack Proxy 2.9.0	OWASP ZAP	2020512	0 KB	2.9.0
3	Microsoft Office ????? 2016 - zh-tw	Microsoft Corporation	2020721	0 KB	16.0.13001.20384
4	Sublime Text 3	Sublime HQ Pty Ltd	2020424	35347 KB	
5	SUSI	Advantech	2020323	42518 KB	4.2.15811.0
6	Java 8 Update 251 (64-bit)	Oracle Corporation	2020512	123266 KB	8.0.2510.8
7	McAfee Solidifier	McAfee, Inc.	2020323	64206 KB	7.0.0.666
8	DeviceOn Server	Advantech	2020821	1365820 KB	4.2.2
9	AnyDesk	philandro Software GmbH	202088	2048 KB	ad 6.0.7
10	FileZilla Server	FileZilla Project	202065	6191 KB	beta 0.9.60

This program information retrieves from device operation system, same as below figure.



## ● Configuration

There are three options for OTA to deploy your package, one is “**Rollback**” that means if a new version deploys failed, and the WISE-Agent would try the best to rollback to previous version that successfully. But there is a perquisite, the previous version of the package exists on the device side. The remaining two options are the retry times. Due to network instability or other factors causing the download fail, OTA provide the retry times to ensure successfully deployment as possible.

**Configuration**

**Advanced Setting**

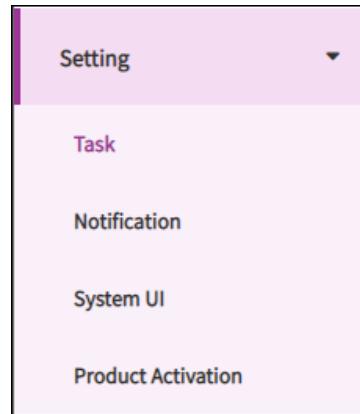
We provide three options that allow you to enable the retry mechanism for downloads and deployments, due to unstable network or environment. Notice that settings provided here are master switches, which make influence to all download or deployment tasks.

**Rollback**  Number of Retry Downloads (0 ~ 10)  Number of Retry Deployments (0 ~ 3)

**Save**

### 3.3.6 System Configuration

A System Configuration define advanced setting include “**Task**”, “**Notification & Event Alert**”, “**System UI**” and “**Product Activation**”. These setting are usually changed less often or only need to be modified once. Some functions require root, admin to modify or be visible, and product activation only shown on prefecture license, such as Standalone, Azure Kubernetes version.



- Task

The real-time actions on the overview that are defined, created on here, you could add a new task and pin to overview. These tasks are binding to personal account, cannot view, edit, and delete others.

Task					
		Keyword Search		1 Set / 1	
PIN TO OVERVIEW	TASK TYPE	TASK DESCRIPTION	GROUP	DATE CREATED	
	Screenshot		Testing	1	2020/08/06 17:14:17

Click on the icon to add action.



Enter your description and select a “**Task**” from three categories, **Power Saving**, **Security** and **System**.

New Action

Select Task 1

Select Device Groups 2

Confirm 3

**Task Description**  
Power off 2F

**Select Task Type**  
Power Saving

Power On       Power Off  
 Reboot       Backlight On  
 Backlight Off       Power Sleep  
 Power Hibernate

Next

Select “Device Groups” for the action that you picked up.

New Action

Select Task 1

Select Device Groups 2

Confirm 3

**Account**  
root@advantech.com.tw

Add Device Groups +

**Groups Amount : 5**

<input type="checkbox"/> AA-X11	<input type="checkbox"/> --Dylan.Chang@advantech.com.tw : DylanTest2
<input checked="" type="checkbox"/> Default	<input type="checkbox"/> Iris
<input type="checkbox"/> wise-test	

**Assigned Groups**

root@advant... X

Back Next

To confirm information, action, group and devices, and enable pin on overview, please click on “Confirm” to complete the wizard.

New Action

1 Select Task      2 Select Device Groups      3 Confirm

**Task Description**  
Power off 2F 

**Task Type**  
Power Off 

**Pin to Overview** 

**Device Groups**  
Groups Amount:1 

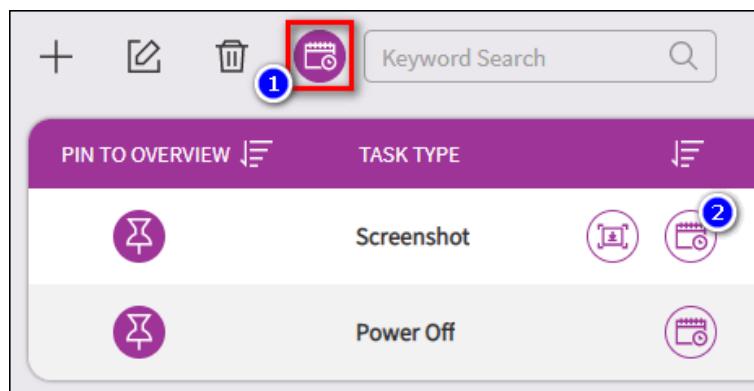
root@advantech.com.tw : Default

**Confirm**

After created, you could find a new action on below actions list, click the PIN icon to determine the action shown on overview or not.

PIN TO OVERVIEW	TASK TYPE		TASK DESCRIPTION	GROUP	DATE CREATED
	Screenshot		Testing	1	2020/08/06 17:14:17
	Power Off		Power off 2F	1	2020/08/27 14:27:16

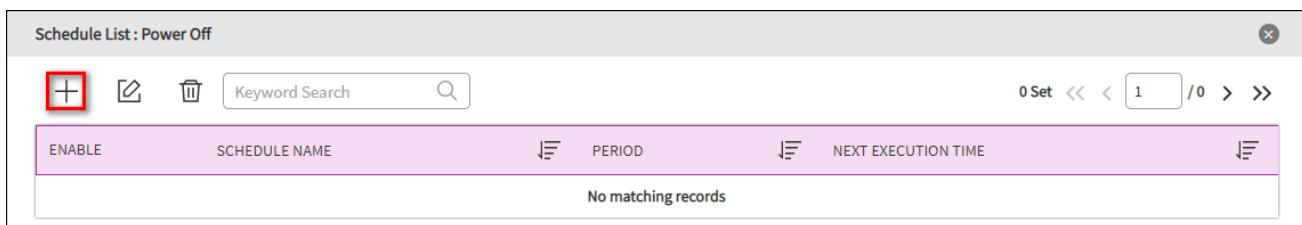
The actions support scheduling, click on the icon to define a schedule, daily, weekly, monthly, yearly or once.



1  Keyword Search

PIN TO OVERVIEW	TASK TYPE
	Screenshot  
	Power Off  

Enter to schedule list for all actions, and click on add icon to create new schedule.



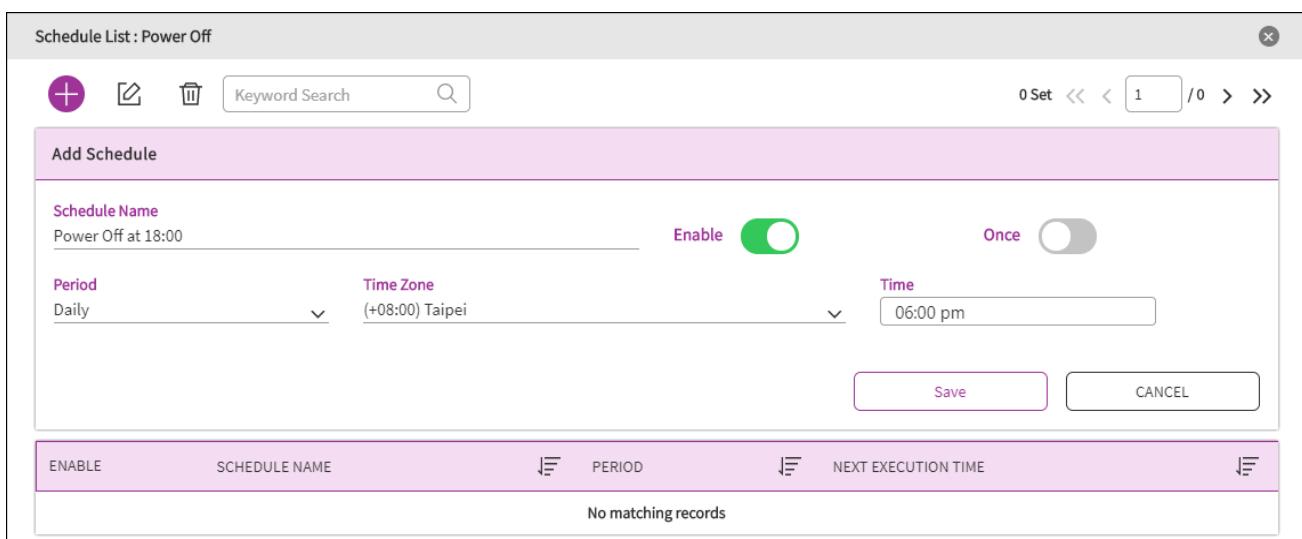
Schedule List : Power Off

0 Set << < 1 /0 > >>

ENABLE	SCHEDULE NAME	PERIOD	NEXT EXECUTION TIME
	No matching records		

Given your schedule name, time zone, period and time and click **Save**.

- ✧ **Schedule Name:** Name of schedule
- ✧ **Time Zone:** Time zones tend to follow the boundaries of countries and their subdivisions instead of longitude, because it is convenient for areas in close commercial or other communication to keep the same time.
- ✧ **Period:** Repeat interval for Daily, Weekly, Monthly, Yearly or once at a time.
- ✧ **Time:** Execution time.



Schedule List : Power Off

0 Set << < 1 /0 > >>

Add Schedule

Schedule Name: Power Off at 18:00

Enable:

Once:

Period: Daily

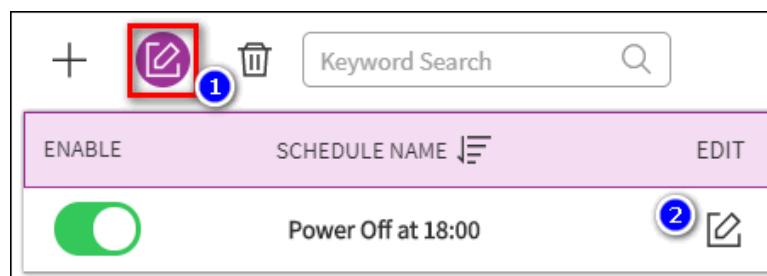
Time Zone: (+08:00) Taipei

Time: 06:00 pm

Save CANCEL

ENABLE	SCHEDULE NAME	PERIOD	NEXT EXECUTION TIME
	No matching records		

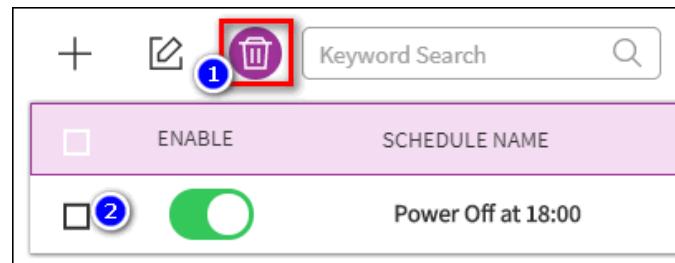
Click on the edit icon to adjust schedule item.



+  Keyword Search

ENABLE	SCHEDULE NAME	EDIT
<input checked="" type="checkbox"/>	Power Off at 18:00	

Click on the delete icon to delete schedule item.



- Notification & Event Alert

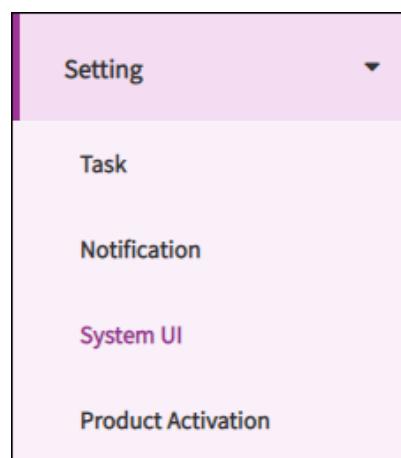
Here are seven notification services, include tradition service (SMS, Email) and popular social media (LINE, WeChat, Telegram, Microsoft Teams, Slack), if you select the event log type on “**Notification Item**”, the notify message will through these services. These notification services are global setting, if your account does not receive, please check the personal setting on **Account**.

To configure these notification service, please reference Section 4.3.2 ~ Section 4.3.5.

- System User Interface

DeviceOn provide an option to customize menu item, theme, logo and user could by setting up the

user interface to meet their needs.



**System UI**

- System Menu
- System Theme
- System Logo
- System Login Page
- Overview Setting
- Language Setting

◇ **System Menu:** To show/hide the items on the menu bar

The default setting do not include “**AddIns**” and “**System Report**”, the “AddIns” is used to customized UI page or embed specific website page to integrate with DeviceOn.

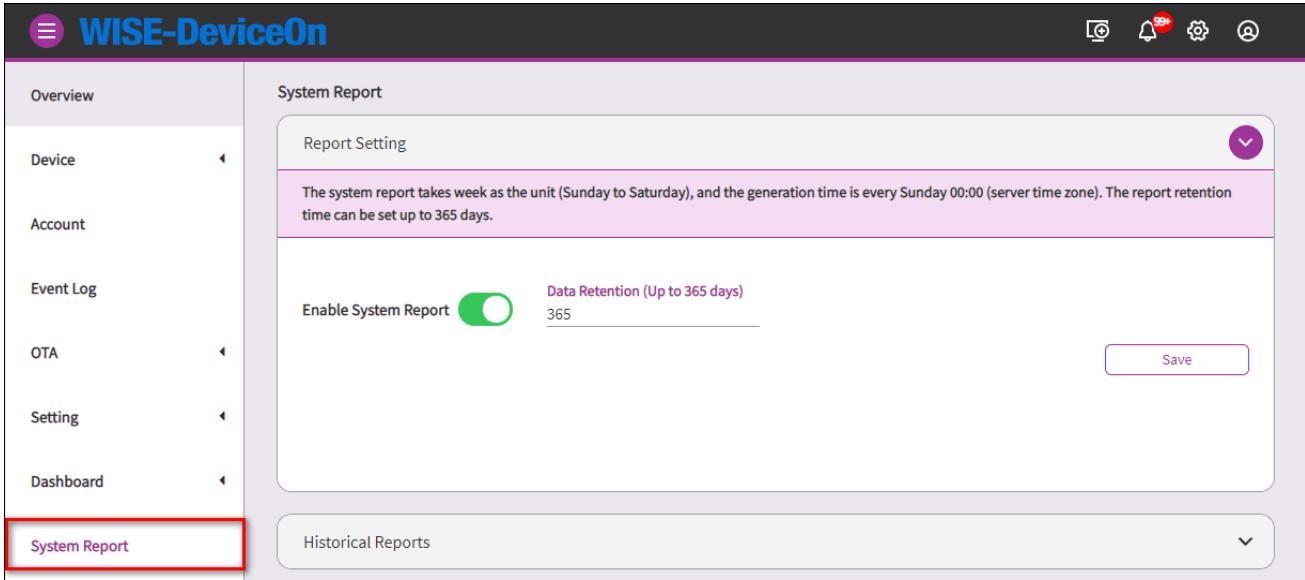
**System UI**

To show/hide the items on the menu bar

Overview	<input checked="" type="checkbox"/>	Device	<input checked="" type="checkbox"/>	Account	<input checked="" type="checkbox"/>	Event Log	<input checked="" type="checkbox"/>
OTA	<input checked="" type="checkbox"/>	Dashboard	<input checked="" type="checkbox"/>	AddIns	<input type="checkbox"/>	System Report	<input type="checkbox"/>

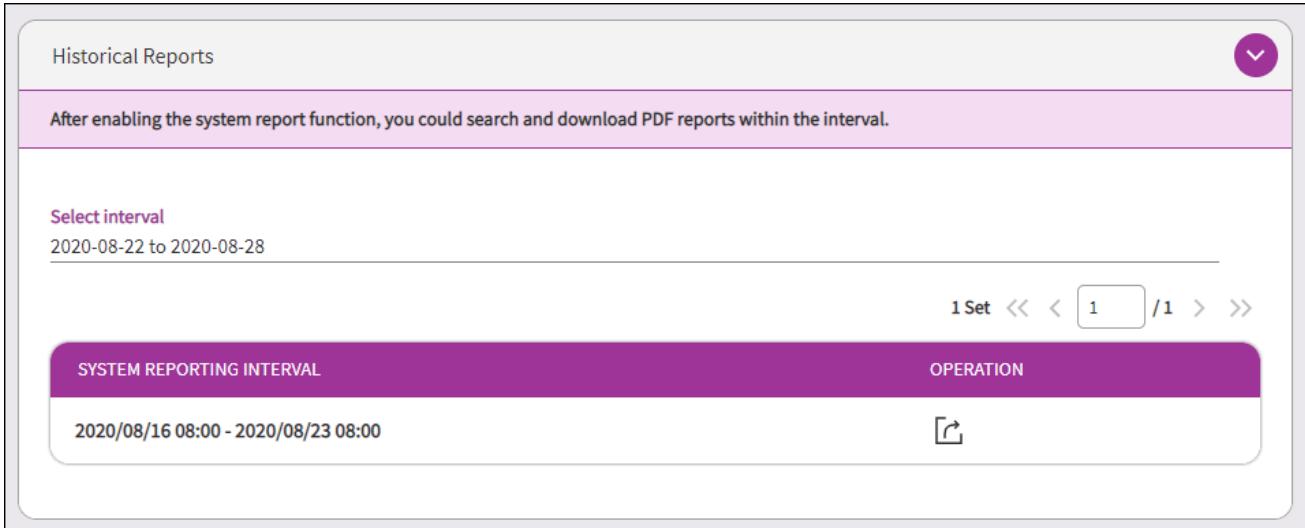
**Restore To Default** **Save**

Second, if the System Report be enabled will appears to the menu item. The system report takes week as the unit (Sunday to Saturday), and the generation time is every Sunday 00:00 (server time zone). The report retention time can be set up to 365 days.



The screenshot shows the WISE-DeviceOn web interface. On the left, there's a vertical navigation bar with items like Overview, Device, Account, Event Log, OTA, Setting, Dashboard, and System Report. The 'System Report' item is highlighted with a red box. The main content area is titled 'System Report' and contains a 'Report Setting' section. It includes a note about the system report taking a week as the unit and generating every Sunday at 00:00. There's a toggle switch labeled 'Enable System Report' which is turned on, and a field for 'Data Retention (Up to 365 days)' with the value '365'. A 'Save' button is located below this section. At the bottom, there's a 'Historical Reports' section with a dropdown arrow.

After enabling the system report function, you could search and download PDF reports within the interval.



This screenshot shows the 'Historical Reports' section. It has a note: 'After enabling the system report function, you could search and download PDF reports within the interval.' Below this is a 'Select interval' section with a date range from '2020-08-22' to '2020-08-28'. To the right, there are navigation buttons for sets and pages, with '1 Set' and '1 / 1' visible. At the bottom, there's a 'SYSTEM REPORTING INTERVAL' box showing '2020/08/16 08:00 - 2020/08/23 08:00' and an 'OPERATION' box with a refresh icon.

From the system report, you may realize the whole status, including server uptime, downtime and managed devices healthy for the pass week.

Weekly report

# WISE-DeviceOn

## DeviceOn Overview

2020-07-05~2020-07-11

### Server Overview

Avg. CPU Usage 8.78 % percent

Average of CPU usage this week.

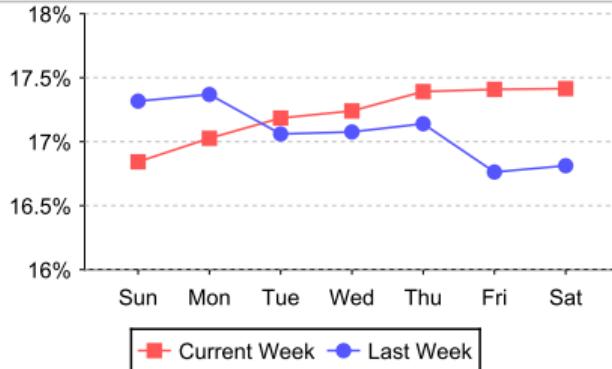
Avg. Mem Usage 74.46 % percent

Average of memory usage this week.

Storage Growth 0.58 % percent

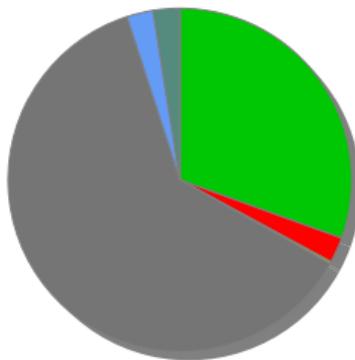
Growth of storage usage this week.

### Overall Storage Usage Trend



Event	Date/Time	Duration	Up Time	Down Time
Up	2020-07-10T09:59:17Z	37 hours, 30 mins		
Down	2020-07-10T09:57:50Z	0 hours, 1 mins	4 Days	5 Minutes
Up	2020-07-10T06:21:35Z	3 hours, 36 mins		
Down	2020-07-10T06:19:54Z	0 hours, 1 mins	99.92 %	0.08 %
Up	2020-07-09T08:45:27Z	21 hours, 34 mins		

### Device Overview



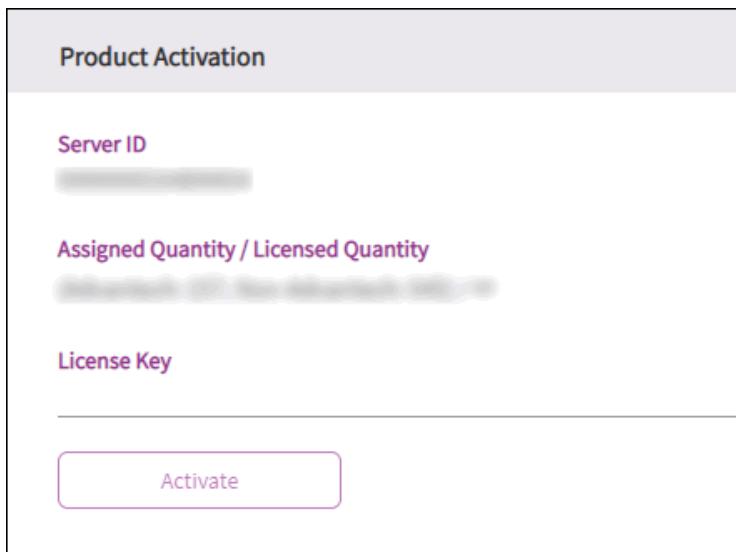
Normal	Error	Warning
989 times	78 times	7 times
30.51 %	30.51 %	30.51 %
Threshold	Disconnect	Loss Connect
2006 times	81 times	81 times
30.51 %	30.51 %	30.51 %

page 1 / 2

- ✧ System Theme: Select the theme style of the system
- ✧ System Logo: Product logo customized, supported formats: GIF, PNG, JPEG/JPG. We recommend the image with a height is less than 55 pixels.
- ✧ System Login Page: Login page customized, supported formats: PNG, JPEG/JPG. We recommend the image with a resolution is less than 860x840 pixels.
- ✧ Overview Setting: To show/hide the functions on the overview
- ✧ Language Setting: Set display language, (English, Traditional Chinese and Simplified Chinese)

#### ● Product Activation

DeviceOn support online and offline to activate product, if your server could access to Advantech License Server that would be simple. Enter your license key that you purchase the product from WISE-PaaS Marketplace.



The screenshot shows a "Product Activation" interface. It has three input fields: "Server ID", "Assigned Quantity / Licensed Quantity", and "License Key". Below these fields is a large "Activate" button.

If your server environment without public network accessible, there is a QR code generated after enter the license key. Please leverage your mobile device to scan and retrieve the "**Activation Code**".

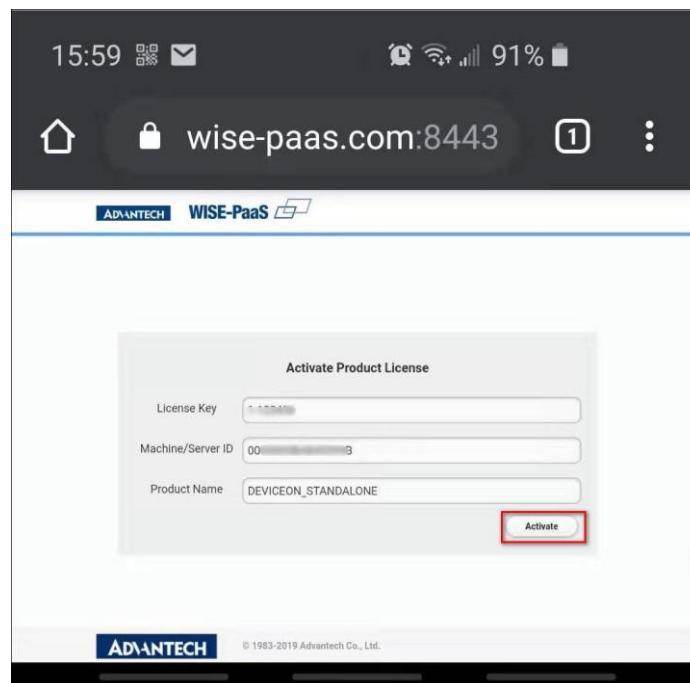
Product Activation

Please scan the QR code through your mobile device and get the activation code



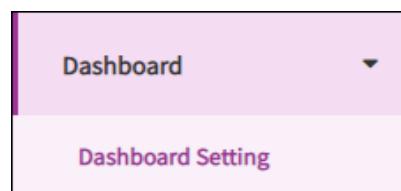
Activation Code

**Activate**

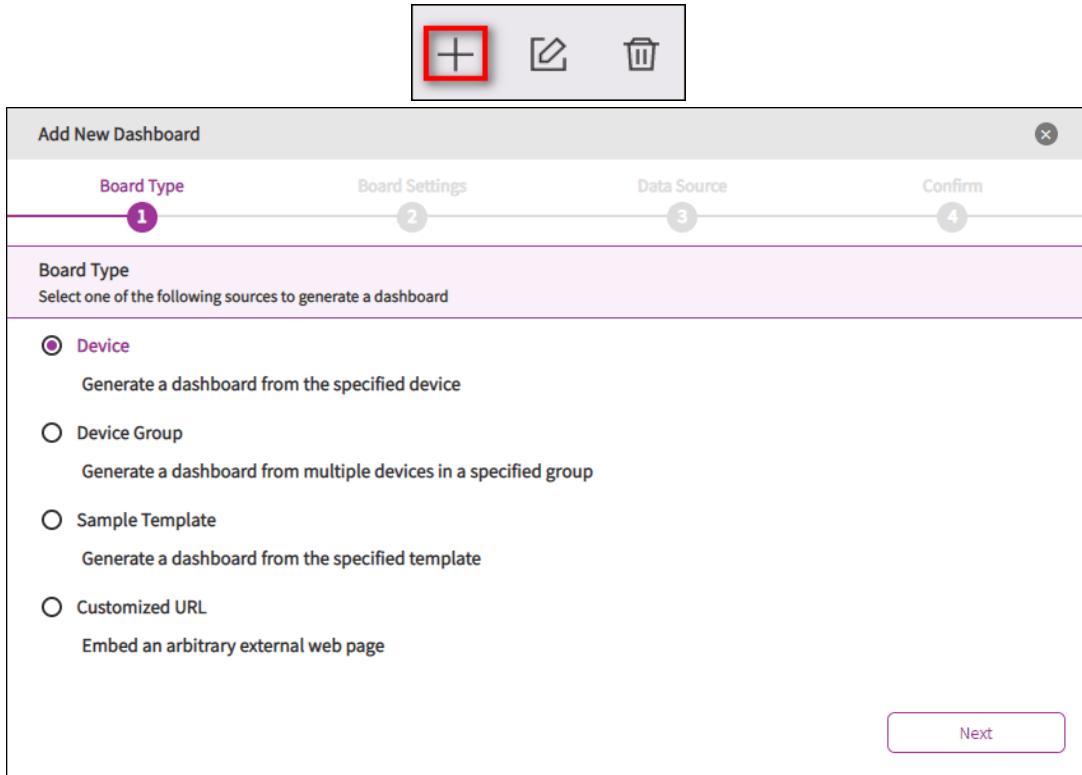


### 3.3.7 Dashboard

The DeviceOn not only native support and build-in Grafana for data visualize, but fully integrate and simplified procedure to 1-Click to generate a dashboard. Grafana is an open-source software for monitoring and analysis. One of its major characteristics is it supports many different data sources, from popular CloudWatch, Elasticsearch, Graphite, and influxDB, to OpenStack Gnocchi or Google Calendar. Its range is very extensive. However, for others data source require to implement SimpleJson to access your data.



Click on the “+” icon to create dashboard item.



Add New Dashboard

Board Type 1

Board Settings 2

Data Source 3

Confirm 4

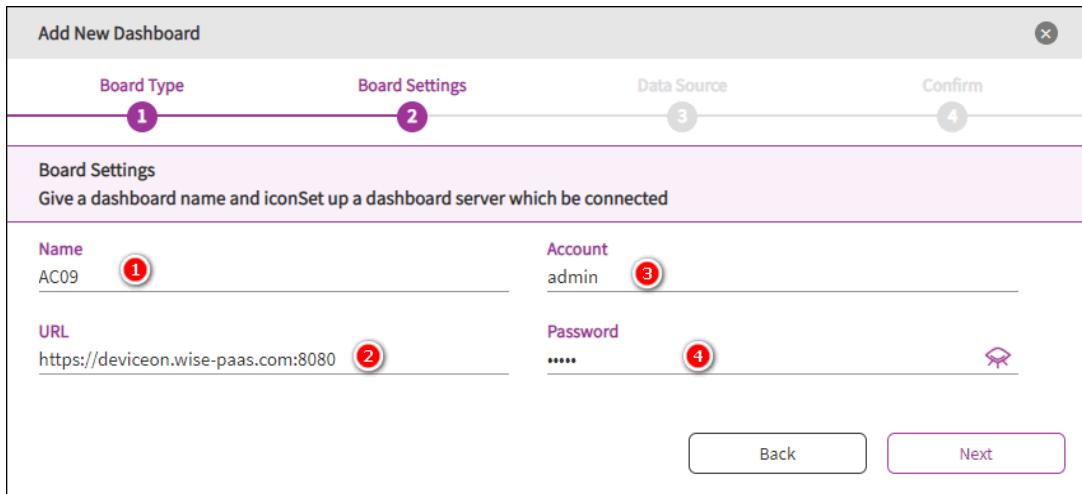
**Board Type**  
Select one of the following sources to generate a dashboard

- Device  
Generate a dashboard from the specified device
- Device Group  
Generate a dashboard from multiple devices in a specified group
- Sample Template  
Generate a dashboard from the specified template
- Customized URL  
Embed an arbitrary external web page

Next

Here, DeviceOn support 4 types of board, select one of method to generate your dashboard. Device, device group, mode or select our default template to generate. Last, embed an arbitrary external web page.

Enter your board name, Grafana URL, account and password. (Default account and password is “admin”)



Add New Dashboard

Board Type 1

Board Settings 2

Data Source 3

Confirm 4

**Board Settings**  
Give a dashboard name and iconSet up a dashboard server which be connected

Name AC09 <span style="border: 1px solid red; border-radius: 50%; padding: 2px 10px;">1</span>	Account admin <span style="border: 1px solid red; border-radius: 50%; padding: 2px 10px;">3</span>
URL <a href="https://deviceon.wise-paas.com:8080">https://deviceon.wise-paas.com:8080</a> <span style="border: 1px solid red; border-radius: 50%; padding: 2px 10px;">2</span>	Password ***** <span style="border: 1px solid red; border-radius: 50%; padding: 2px 10px;">4</span>

Back Next

Select target sensor from the device and click next.

Add New Dashboard

Board Type **1** Board Settings **2** Data Source **3** Confirm **4**

Data Source  
Select target sensors

Select Account Root **1** Select Device Groups AA-X11 **2** Select Device AC09 **3**

Hard Disk Predictive Maintenance

Data Synchronization

Hardware Monitoring

Process Monitor

Lockdown Utility

Hard Disk Monitor

Network Monitor

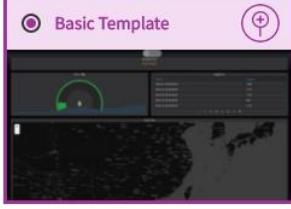
[Back](#) [Next](#)

(Device Method)

Add New Dashboard

Board Type **1** Board Settings **2** Data Source **3** Confirm **4**

Data Source  
Select the template

Basic Template 

IPC Template 

[Back](#) [Next](#)

(Template Method)

Confirm the result and information, and start to generate.

Add New Dashboard

Board Type 1 Board Settings 2 Data Source 3 Confirm 4

Confirm  
Generate a dashboard with this setting?

Board Type	Board Settings
Board Type Device	Name AC09  Account admin
	URL https://deviceon.wise-paas.com:8080  Password 

Data Source 5

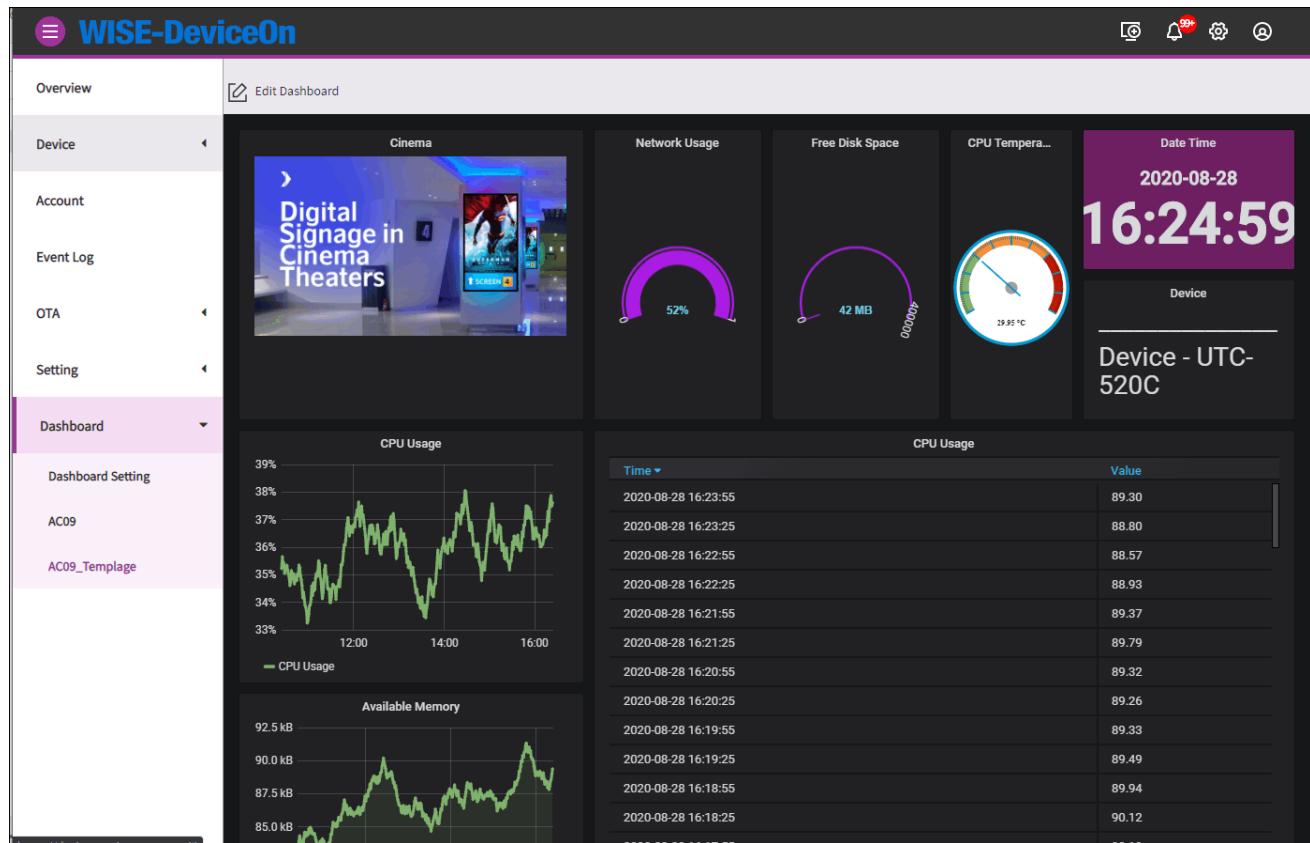
Sensor  
Sensor Amount : 20

Show All Items 

Confirm

After that, the board be generated on the menu item.



The screenshot shows the WISE-DeviceOn interface with a sidebar navigation on the left and a main dashboard area on the right.

- Left Sidebar:**
  - Overview
  - Device
  - Account
  - Event Log
  - OTA
  - Setting
  - Dashboard** (selected)
  - Dashboard Setting
  - AC09
  - AC09\_Template
- Main Area:**
  - Cinema:** Digital Signage in Cinema Theaters (Image)
  - Network Usage:** Gauge chart showing 52% usage.
  - Free Disk Space:** Gauge chart showing 42 MB available.
  - CPU Temperature:** Gauge chart showing 29.95 °C.
  - Date Time:** 2020-08-28 16:24:59
  - Device:** Device - UTC-520C
  - CPU Usage:** Line chart showing CPU Usage over time (33% to 39%) from 12:00 to 16:00.
  - Available Memory:** Line chart showing Available Memory over time (85.0 kB to 92.5 kB).
  - Time Value Table:**

Time	Value
2020-08-28 16:23:55	89.30
2020-08-28 16:23:25	88.80
2020-08-28 16:22:55	88.57
2020-08-28 16:22:25	88.93
2020-08-28 16:21:55	89.37
2020-08-28 16:21:25	89.79
2020-08-28 16:20:55	89.32
2020-08-28 16:20:25	89.26
2020-08-28 16:19:55	89.33
2020-08-28 16:19:25	89.49
2020-08-28 16:18:55	89.94
2020-08-28 16:18:25	90.12

## 4. Hands-On LABs

## 4.1 How to Create a Real-time Action into Overview

The real-time action is a handy way to execute a specific command to a bunch of devices. This lab guides you how to create a real-time action. And, after this lab, you should:

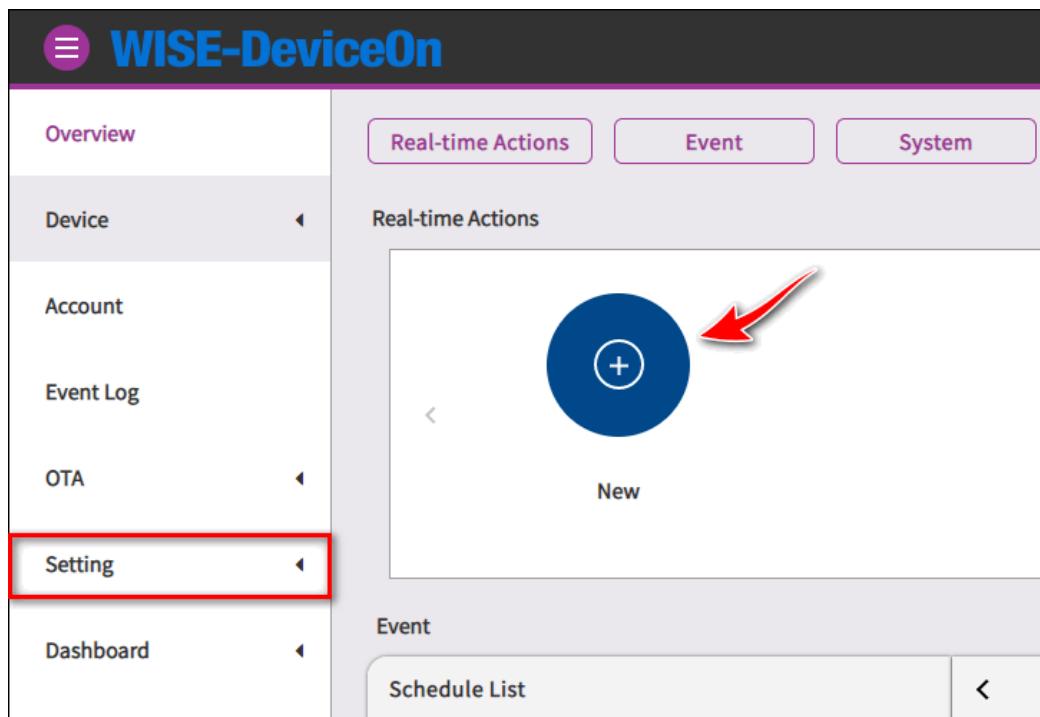
- Learn how to create a real-time action on demand.
- Know of what actions DeviceOn provides.
- Have an action named “**MyTask**” and pinned into your “**Overview**” page, that can reboot devices belong to group “**Default**”.

### 4.1.1 Prerequisite

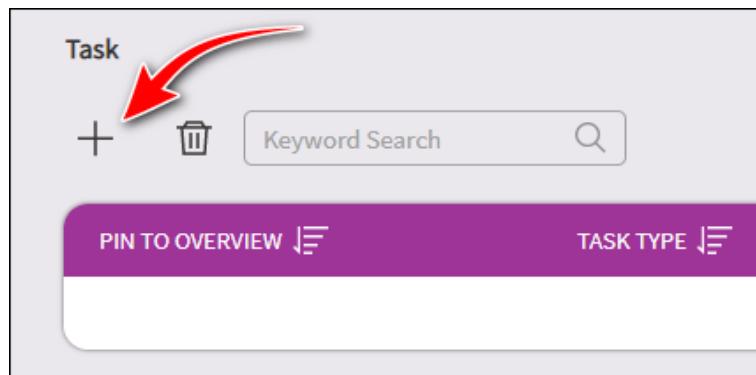
- A running DeviceOn server.
- A device that installed WISE-Agent connects to DeviceOn server.

### 4.1.2 Step-by-Step

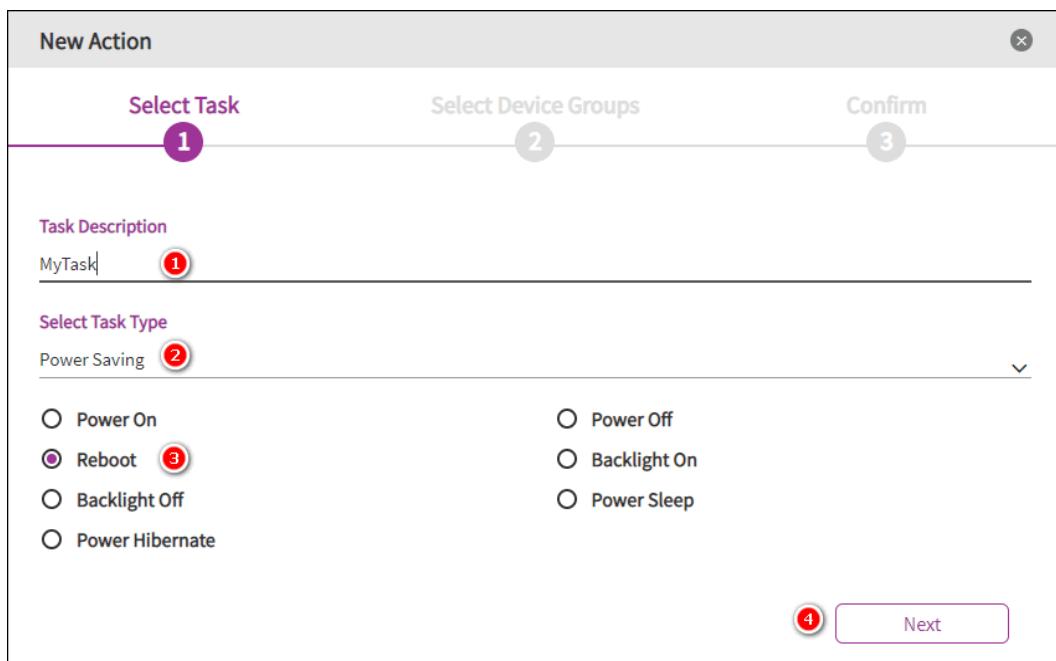
**Step 1:** To create a real-time action, click the “**New**” icon in “**Overview**”. Alternate, click “**Setting**” from the menu populated in left hand side.



**Step 2:** Either way you use in step 1, it leads you into the “**Add Task**” page. Click the “+” symbol.



**Step 3:** You now run into the first page “**Select Task**” to create a new real-time action. Enter the task name “**MyTask**” as well as choose the action “**Reboot**” within this page. From this page you can see all tasksDeviceOn provides. Then end this page up with clicking “**Next**” button.



New Action

1 Select Task      2 Select Device Groups      3 Confirm

Task Description  
MyTask ①

Select Task Type  
Power Saving ②

Power On       Power Off  
 Reboot ③       Backlight On  
 Backlight Off       Power Sleep  
 Power Hibernate

④ Next

**Step 4:** Choose the target group “**Default**” to execute the real-time action in “**Select Device Groups**” page.

New Action

Select Task      Select Device Groups      Confirm

1                          2                          3

Account  
root@advantech.com.tw 

Add Device Groups 

Groups Amount : 5

AA-X11       --Dylan.Chang@advantech.com.tw : DylanTest2  
 Default        Iris  
 wise-test

Assigned Groups  
root@advan... 

  3

**Step 5:** The last page “**Confirm**” provides you a summary like information and, more than those, lets you decide whether this action “**Pin**” to your “**Overview**” page or not. DeviceOn turns this feature on by default. Just toggle it if you don’t want this action pin to your home. Finally, click “**Confirm**” button to finish.

New Action

Select Task      Select Device Groups      Confirm

1                          2                          3

Task Description  
MyTask 

Task Type  
Reboot 

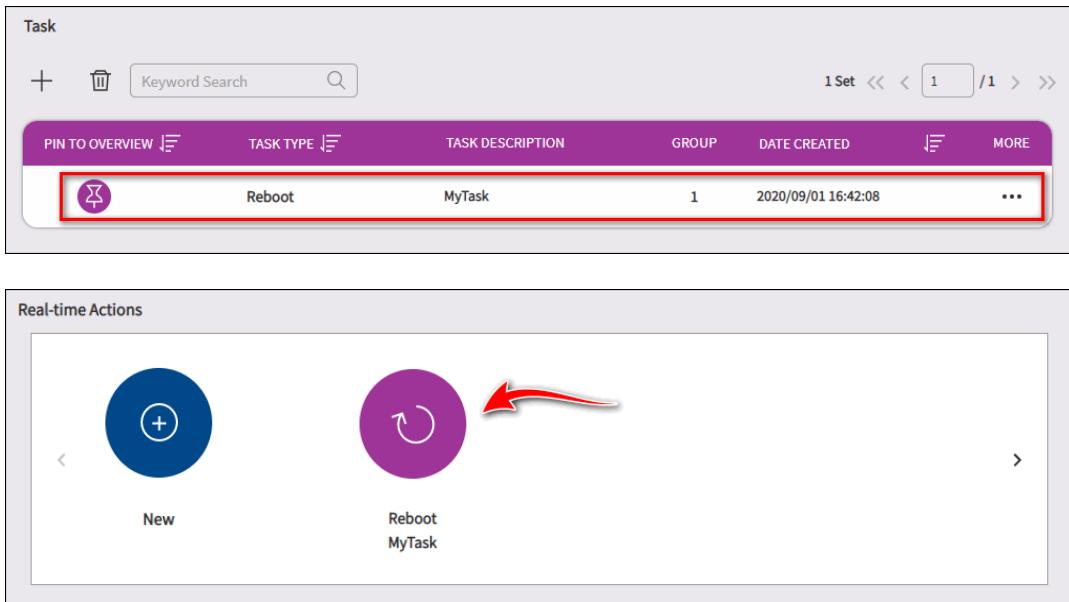
Pin to Overview 

Device Groups  
Groups Amount: 1 

root@advantech.com.tw : Default



If everything goes well, you should see there is a new item generated within “**Task**”. Meanwhile, if you go to your home (page “**Overview**”), you can see a new one action icon is populated there.



What should, or can, you do now? Yes, one-click that icon you created from “**Overview**” page, and watch the devices whether they execute reboot action.

## 4.2 How to Remote Software Provisioning via OTA

OTA (Over-The-Air) is another powerful feature DeviceOn provides. Users can deploy software packages onto a device remotely, or even many devices broadly. This lab guides you how to accomplish remote software provisioning via OTA. And, after this lab, you should:

- Learn how to remote provisioning your software via OTA on demand.
- Learn how to package your software for remote provisioning.
- Have the NotePad++, a popular and famous text editor, populated within the target device.

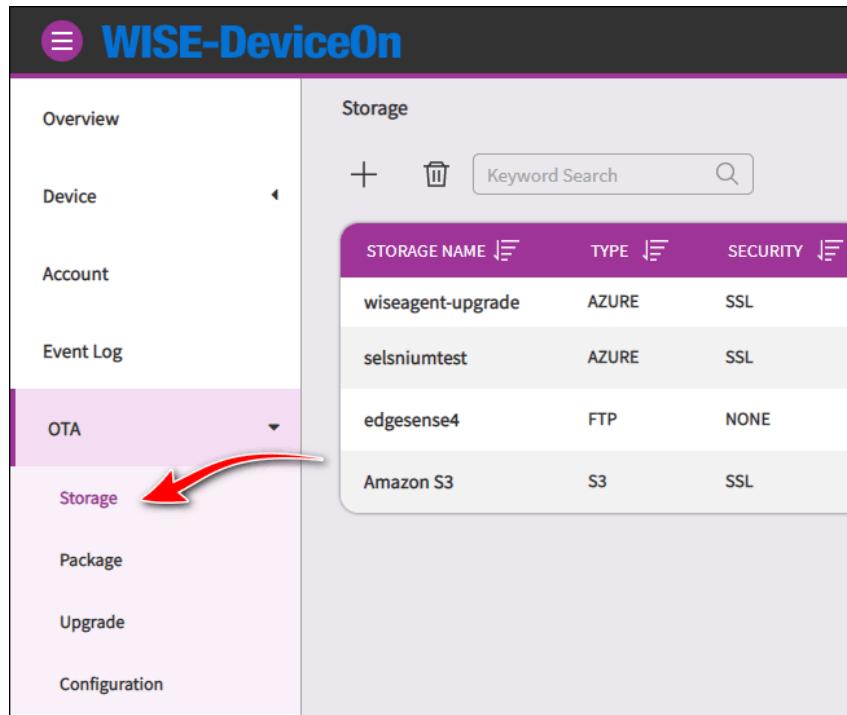
### 4.2.1 Prerequisite

- A running DeviceOn server.
- A device which running on Windows operating system and installed WISE-Agent, that connects to DeviceOn server.
- A running, with well configured, FTP server as the storage.
- A NotePad++ installer, 32-bit edition is recommended. Its name is “**npp.7.8.2.Installer.exe**”, something like that. It can be downloaded from <https://notepad-plus-plus.org/downloads/>.
- Automation skills to install target software package. It is because that user intervention is not possible during provisioning via OTA. For Windows it can be batch file or power shell, while for

Ubuntu it may be shell scripts.

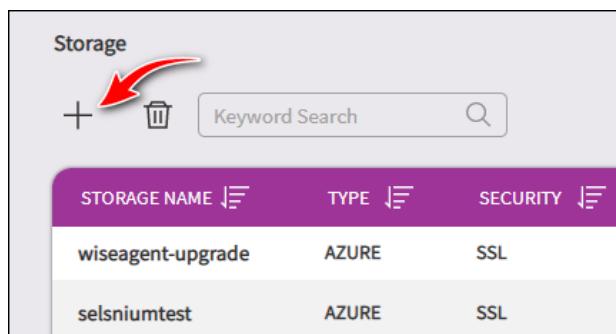
#### 4.2.2 Step-by-Step

**Step 1:** Click “OTA” from the menu on left hand side. It leads you into the “Storage” page.



STORAGE NAME	TYPE	SECURITY
wiseagent-upgrade	AZURE	SSL
selsniumtest	AZURE	SSL
edgesense4	FTP	NONE
Amazon S3	S3	SSL

**Step 2:** In “Storage” page, click the plus (+) sign. This step leads you into the “Add New Storage” page. You have to add a new storage to upload new packages.



STORAGE NAME	TYPE	SECURITY
wiseagent-upgrade	AZURE	SSL
selsniumtest	AZURE	SSL

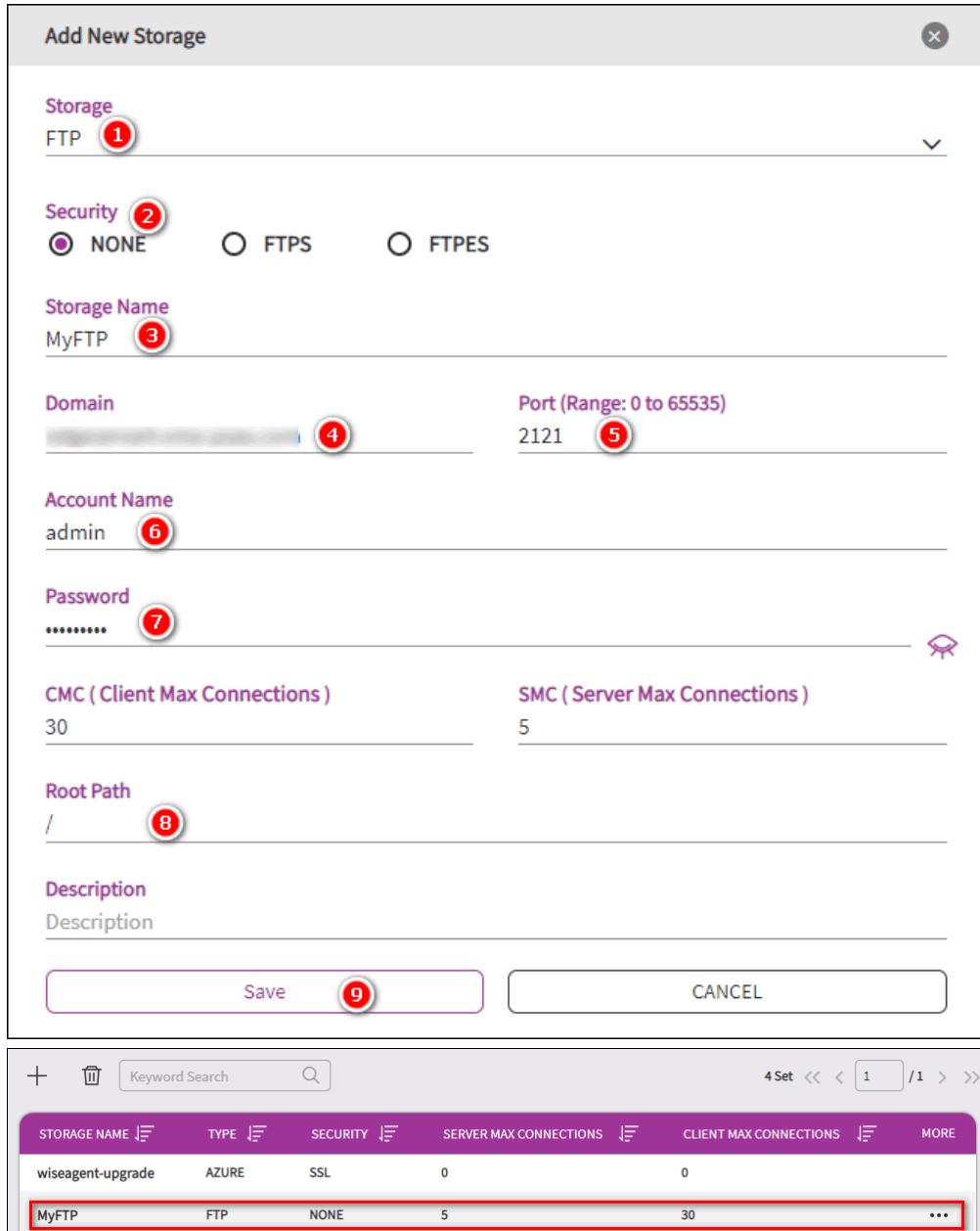
**Step 3:** Fill all fields in with proper values like following:

- **STORAGE:** Pick “FTP” from the dropdown lists.
- **Security:** Leave it as “NONE”, the default value. If your FTP server running on FTPS protocol, pick “FTPS”.
- **STORAGE NAME:** Enter “MyFTP”.
- **DOMAIN:** Enter the FQDN of your FTP server, or its IP address.
- **PORT:** Should be **21** if the FTP server runs on a standard port number.
- **ACCOUNT NAME:** A valid username that can connect to the FTP server, and upload files onto

the server as well.

- **PASSWORD:** The password to login.
- **CMC/SMC:** Use defaults.
- **ROOT PATH:** Simply uses “/”.
- **DESCRIPTION:** Leave it empty. It’s optional information.

Finally, click “**Save**” button to finish this step. If it goes well, you should see a new table row regarding this FTP storage populated in “**Storage**” page.



The screenshot shows two parts of the WISE-PaaS interface. The top part is a modal dialog titled "Add New Storage" with the following fields and their values, each marked with a red circle containing a number:

- Storage:** FTP (1)
- Security:** NONE (2) (radio button selected)
- Storage Name:** MyFTP (3)
- Domain:** (4) (text input field)
- Port (Range: 0 to 65535):** 2121 (5)
- Account Name:** admin (6)
- Password:** (7) (text input field with eye icon)
- CMC (Client Max Connections):** 30
- SMC (Server Max Connections):** 5
- Root Path:** / (8)
- Description:** Description

The bottom part is a table titled "Storage" showing the list of stored configurations:

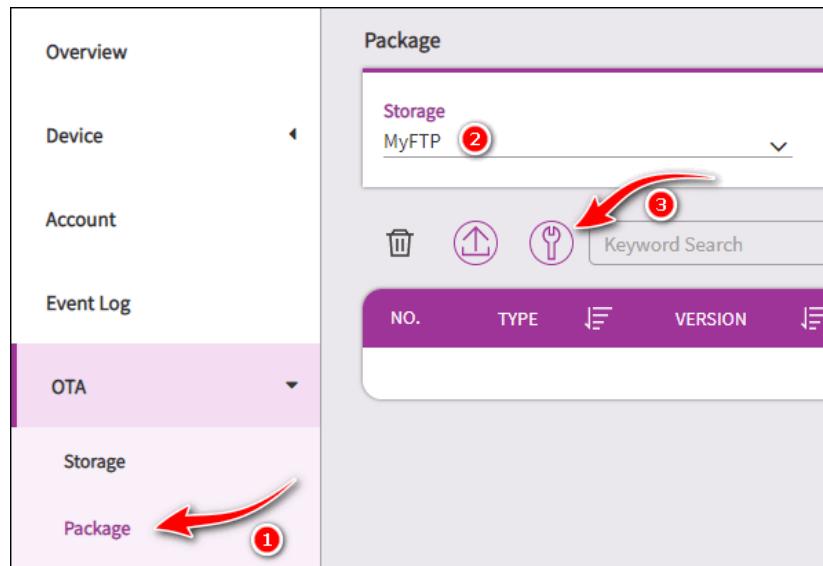
STORAGE NAME	TYPE	SECURITY	SERVER MAX CONNECTIONS	CLIENT MAX CONNECTIONS	MORE
wiseagent-upgrade	AZURE	SSL	0	0	
MyFTP	FTP	NONE	5	30	...

**Step 4:** An extra step we need to execute prior to next step: prepare a valid package for OTA. DeviceOn provides users a toolkit to pack all stuff to be a valid OTA package.

1. Create a new folder names “**NPP**” in, say, your desktop.

2. Move the downloaded file “**npp.7.8.2.Installer.exe**” into.
3. Create a new file “**install.bat**”, contains only **start /wait npp.7.8.2.Installer.exe /S**, inside.  
 This command, per its document in official web site, installs the downloaded NotePad++ software silently.

**Step 5:** Now click the “**Package**” item. And, then, choose “**“MyFTP”** from “**STORAGE**” field. Last, click the “**Package Toolkit**” icon to enter “**Package Toolkit**” page.



**Step 6:** In “**Package Toolkit**” page, fill all mandatory field up with proper values. At last, click “**Generate**” button to package “**NPP**” software, and upload onto “**MyFTP**” storage as well.

- **Package Type:** Fill “**NPP**” up.
- **Package Version:** Fill “**1.0.0.0**” up.
- **Device Group:** Choose “**Default**”.
- **DEVICE:** Choose the target device. “**AA-Win**” in this lab environment.
- **SOURCE DIR:** Click “**Browser**” to point to the location of “**NPP**” folder we created in step 4.
- **DEPLOY FILE:** DeviceOn chooses “**install.bat**” for you.
- **STORAGE:** Choose “**MyFTP**” from dropdown list.

**Package Toolkit**

**Package Type**  
NPP 1

**Package Version**  
1.0.0.1 2

**Account**  
Root

**Device Group**  
Default 3

**Device**  
AC09 4

**Tags**  
 x64  x86  win

**Source Directory**  
otademo 5

**Deploy Script**  
installNotepad.bat 6

**Storage**  
MyFTP 7

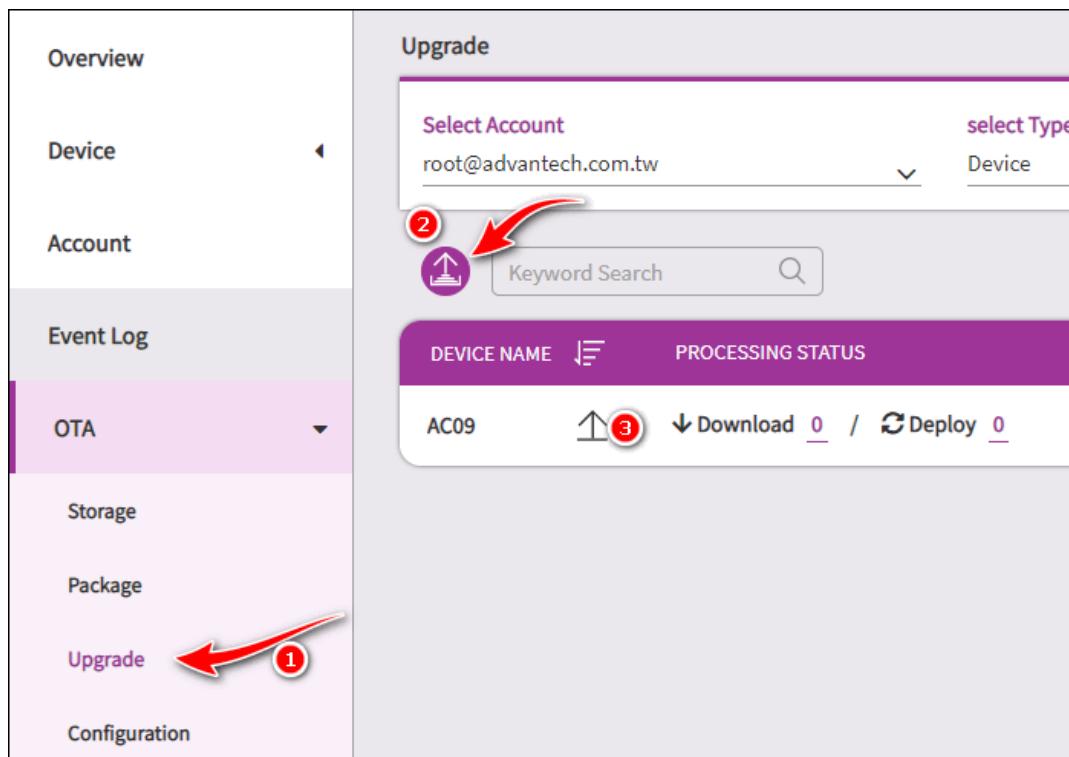
Advanced options

8 Generate      CANCEL

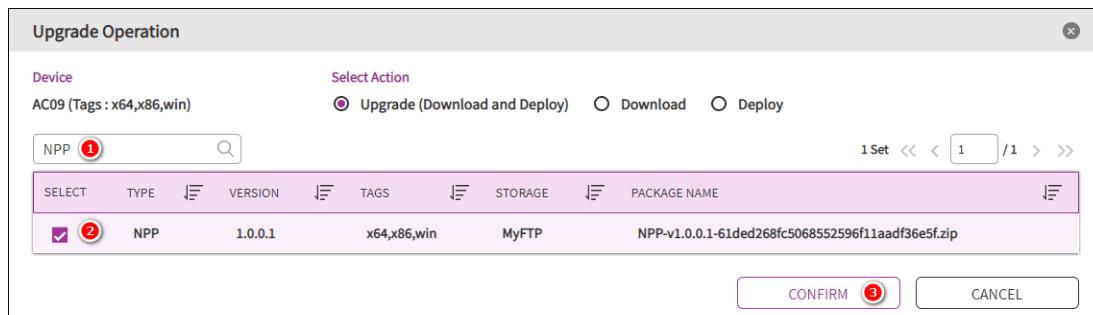
**Step 7:** Now, in “**Package**” page, a new one table row represents the “**NPP**” package has been added.

NO.	TYPE	VERSION	TAGS	STORAGE	PACKAGE NAME	UPLOAD TIME
1	NPP	1.0.0.1	x64,x86,win	MyFTP	NPP-v1.0.0.1-80a19c243c9171a843...	2020/09/01 17:07

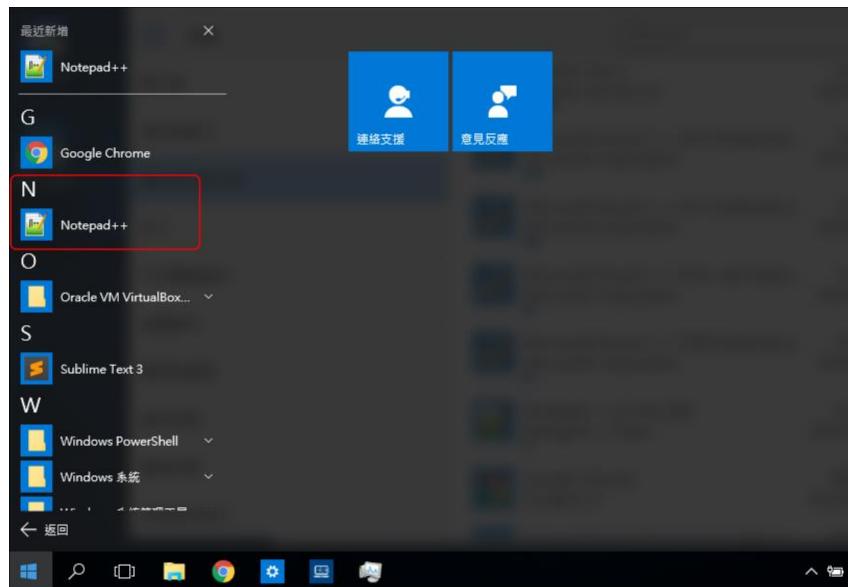
**Step 8:** It is time to install NotePad++ onto the target device remotely. Based on previous step, click “**Upgrade**” tab next to “**Package**” tab. You should find the target device shows there within the table view. Click the icon locates in target device row and “**UPGRADE**” column. It leads you into the “**Upgrade Operation**” page.



**Step 9:** In “Upgrade Operation” page, fill “NPP” up in “KEYWORD SEARCH” field so that the package can be filtered out of all packages. Check the box accordingly and click “CONFIRM” button.



**Step 10:** Now the NotePad++ should been be installing and, after a while, if everything went well, a corresponding application item should be created in Windows menu.



## 4.3 How to Set a Device Threshold and Event Notify Services

For devices monitoring, DeviceOn provides the rule engine. Users can acquire anomaly situations by means of setting thresholds to those interested devices, and, once one or more thresholds meets, receive alerts via event notification services, another one indispensable feature for users. This lab guides you how to set thresholds to a device and how to set event notification services as well. As such, after this lab, you should:

- Learn how to set thresholds to a device on demand.
- Learn how to set event notification services, including email, LINE, and WeChat as well.

### 4.3.1 Prerequisite

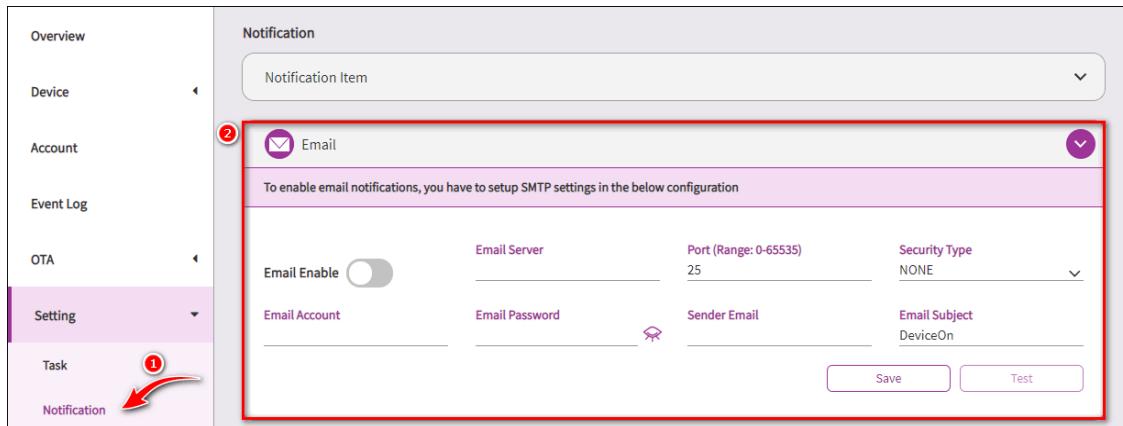
- A running DeviceOn server.
- A device that installed WISE-Agent connects to DeviceOn server.
- A valid, send-able, email account to enable Email notification service.
- A valid LINE account to enable LINE notification service.
- A valid WeChat account, as well as a valid GitHub account, to enable WeChat notification service.

### 4.3.2 Steps to Set Event Notification Service – Email

The configuration of using email as one of event notification services is a system-wide setting. This means DeviceOn uses the server, the one you set in this step, to send all emails. Therefore, uses email settings from your organization is recommended, rather than uses your personal Gmail. If you really want to use Gmail, the situations you are running into may vary and depends on your google account

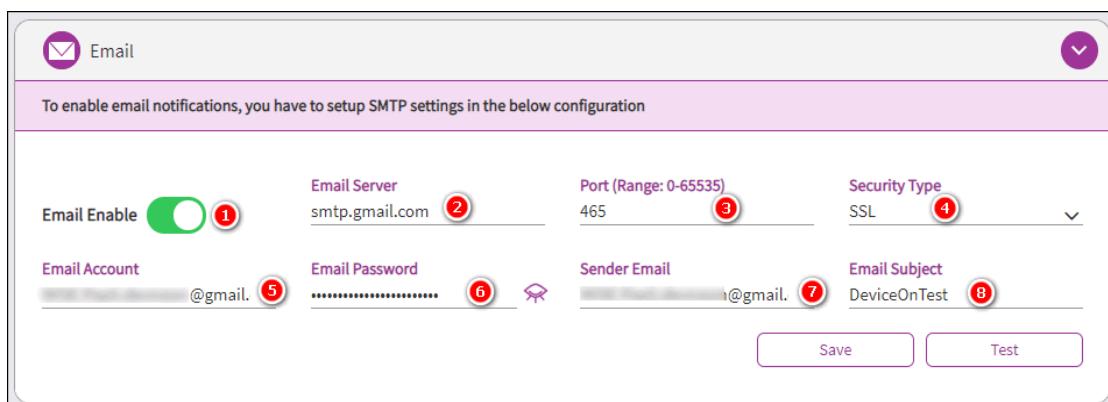
settings. So, in this lab, we assume that you have already a valid business email address from your company.

**Step 1:** Click “**Setting**” menu on the left-hand side of DeviceOn portal and, then, “**Notification**”. Click “**EMail**” bar to open settings regarding email notification service.

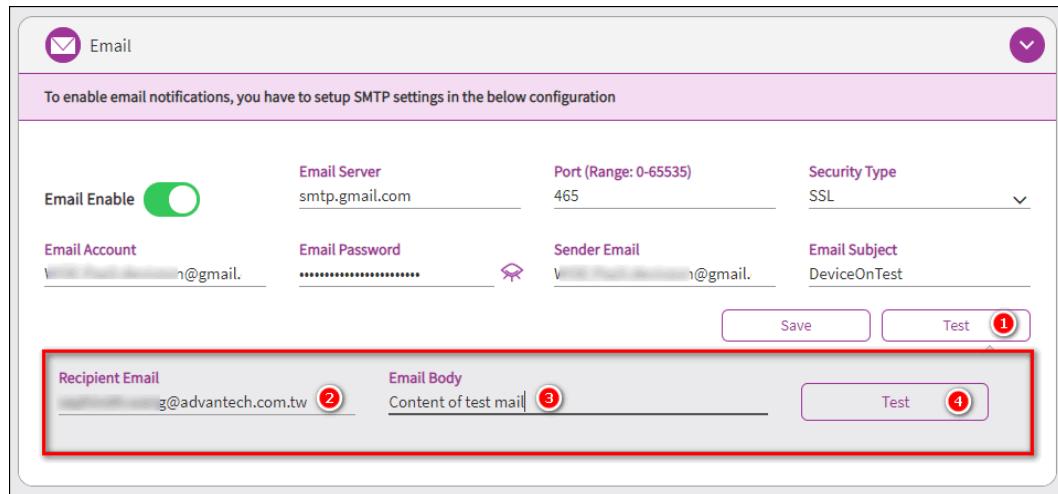


**Step 2:** Toggle “**On/Off**” switch to enable this feature. Then fill fields up with proper values. And end up this step by clicking “**Test**” button.

- **EMAIL SERVER:** The email server host name.
- **PORT:** The email server port. Normally this is 25.
- **SSL/TLS:** Toggle to a proper setting.
- **EMAIL ACCOUNT:** Your email account name. If takes the windows domain into account, a value format like “**DOMAIN\USER**” should be used.
- **EMAIL PASSWORD:** Your password to sign in to the email server.
- **SENDER EMAIL:** Your email address.
- **EMAIL SUBJECT:** Leave it the default.



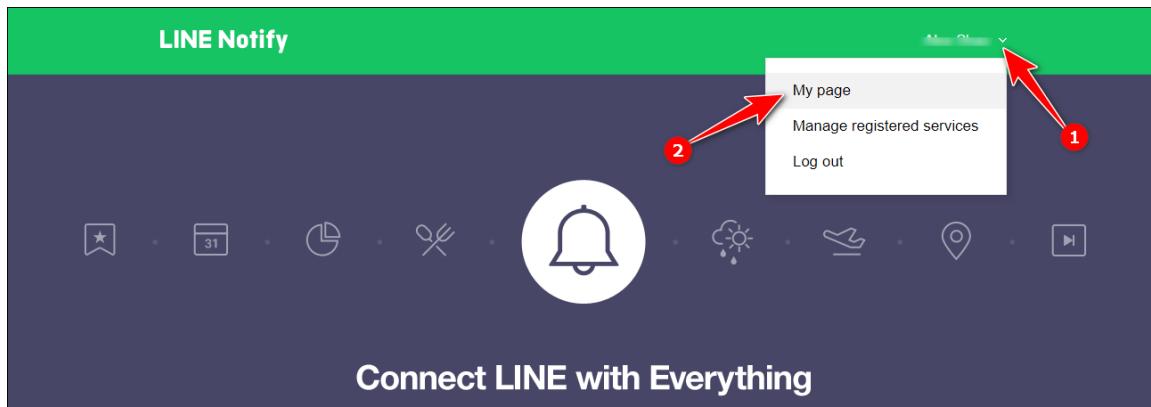
**Step 3:** To assert all values are correct, click “**Test**” button, on the bottom right of the page, to open the “**Send Email for Testing**” dialog for testing purpose. And fill a recipient email as well as email body. Then click “Test” on this dialog. An email you should receive in a while later. Revise them until you got a test email.



**Step 4:** Click “**Save**” on the bottom right of the page that shows in step 2 to keep all settings and enable email notification service.

#### 4.3.3 Steps to Set Event Notification Service – LINE

**Step 1:** Go to <https://notify-bot.line.me/> and sign in with your LINE account. Click “**My Page**” from your account’s dropdown menu in the upper right of the page.



**Step 2:** Click “**Generate token**” under “**Generate access token (For developers)**”. It pops up the “**Generate token**” dialog.

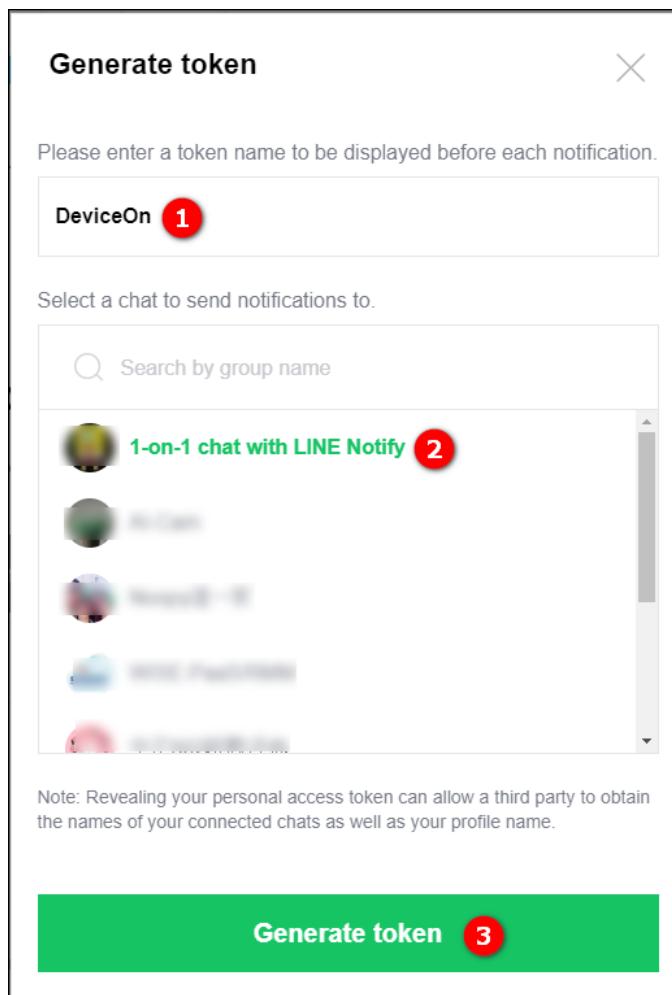
## Generate access token (For developers)

By using personal access tokens, you can configure notifications without having to add a web service.

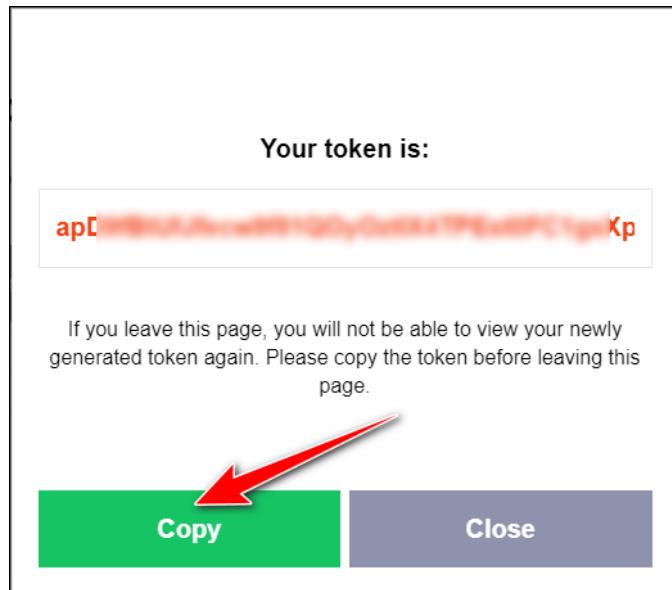
**Generate token**

[LINE Notify API Document](#)

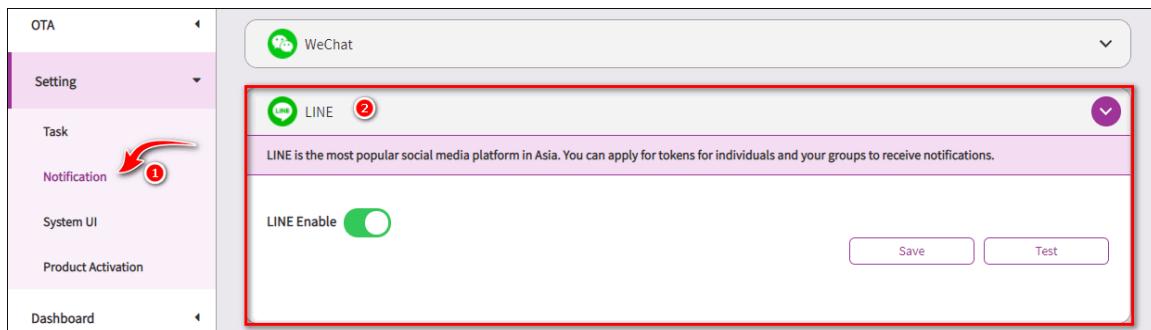
**Step 3:** Fill token field up with “DeviceOn” and click the “1-on-1 chat with LINE Notify” item. Then click the “Generate token” button in green at bottom.



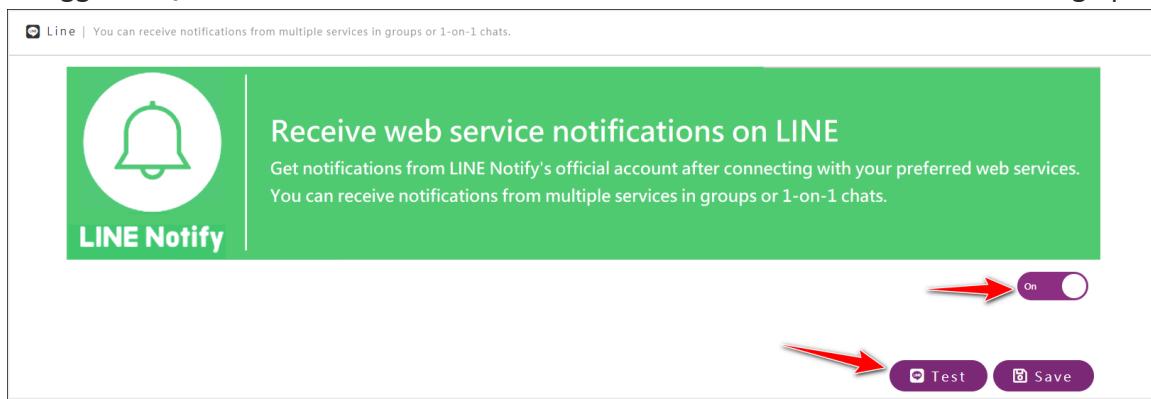
**Step 4:** A new window pops up with token. Meanwhile, a LINE message about this token generation received immediately. Click “Copy” to keep the token in memory, or any file you like.



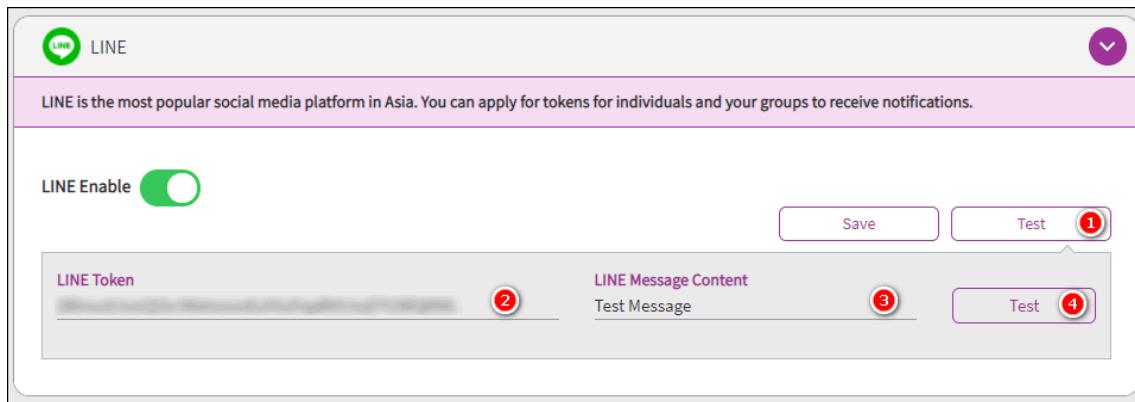
**Step 5:** Now switch your browser to DeviceOn portal. Click “Setting” menu on the left-hand side, then “Notification”, and last “LINE” bar to open settings regarding LINE event notification service.



**Step 6:** Toggle “On/Off” switch to enable this feature. Click “Test” to show the test dialog up.



**Step 7:** Paste the copied token into the first field (LINE Token) and write something into the second field (LINE Message Content). Click “Test”, you should receive the messages you wrote with “DeviceOn” as the prefix.



**Step 8:** Click “Save” button that shows in **Step 6** to keep your settings and enable LINE event notification service.

#### 4.3.4 Steps to Set Event Notification Service – WeChat

**Step 1:** Go to <http://sc.ftqq.com/3.version>. Click “登入网站” hyperlink.

是什么

「Server酱」，英文名「ServerChan」，是一款「程序员」和「服务器」之间的通信软件。

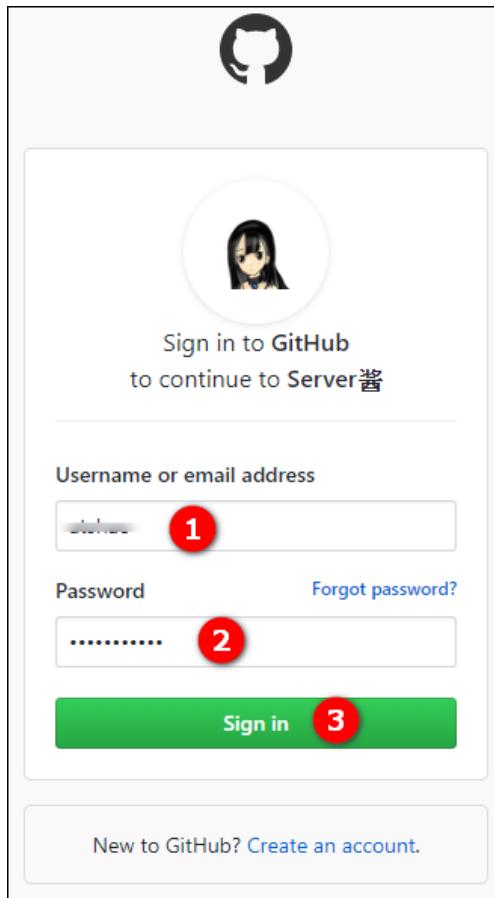
说人话？就是从服务器推报警和日志到手机的工具。

开通并使用上它，只需要一分钟：

1. 登入：用GitHub账号[登入网站](#)，就能获得一个SCKEY（在「发送消息」页面）
2. 绑定：点击「微信推送」，扫码关注同时即可完成绑定
3. 发消息：往 <http://sc.ftqq.com/SCKEY.send> 发GET请求，就可以在微信里收到消息啦

来个示意图：

**Step 2:** Sign in with your GitHub account.



Step 3: Click “微信推送” hyperlink.

# Server酱

## 是什么

「Server酱」，英文名「ServerChan」，是一款「程序员」和「服务器」之间的通信软件。说人话？就是从服务器推报警和日志到手机的工具。

开通并使用上它，只需要一分钟：

1. 登入：用GitHub账号[登入网站](#)，就能获得一个SCKEY（在「发送消息」页面）
2. 绑定：点击「[微信推送](#)」，扫码关注同时即可完成绑定
3. 发消息：往 <http://sc.ftqq.com/SCKEY.send> 发GET请求，就可以在微信里收到消息啦

Step 4: Click “开始绑定”. It opens a QR code image.



**Step 5:** Take your mobile up, swipe and open WeChat App to scan this generated QR code so that the service can bind with your WeChat account.



**Step 6:** Once it is done. The page changes, like below.



**Step 7:** Click “SCKEY” hyperlink and copy, from the opened page, the SCKEY value.

## Server酱

### 是什么

「Server酱」，英文名「ServerChan」，是一款「程序员」和「服务器」之间的通信软件。说人话？就是从服务器推报警和日志到手机的工具。

开通并使用上它，只需要一分钟：

1. 登入：用GitHub账号[登入网站](#)，就能获得一个SCKEY 在「发送消息」页面)
2. 绑定：点击「微信推送」，扫码关注同时即可完成绑定
3. 发消息：往 <http://sc.ftqq.com/SCKEY.send> 发GET请求，就可以在微信里收到消息啦

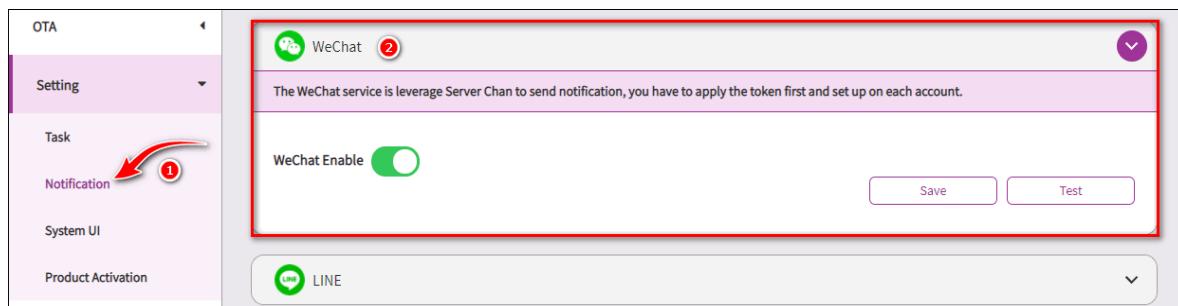
## 调用代码

您的SCKEY如下

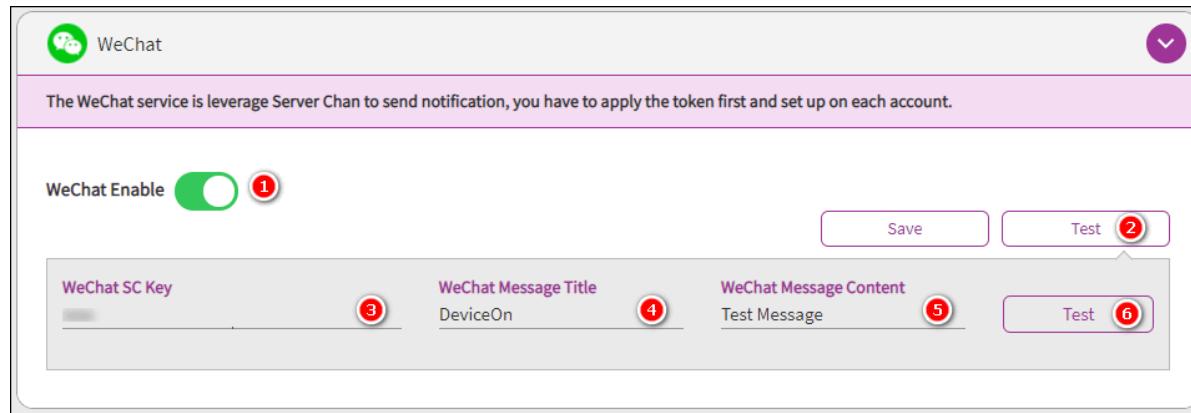
SCU70871...120f57de3f

[重置SCKEY](#)

**Step 8:** Now switch your browser to DeviceOn portal. Click “Setting” menu on the left-hand side, then “Notification”, and “WeChat” to open settings regarding WeChat event notification service.



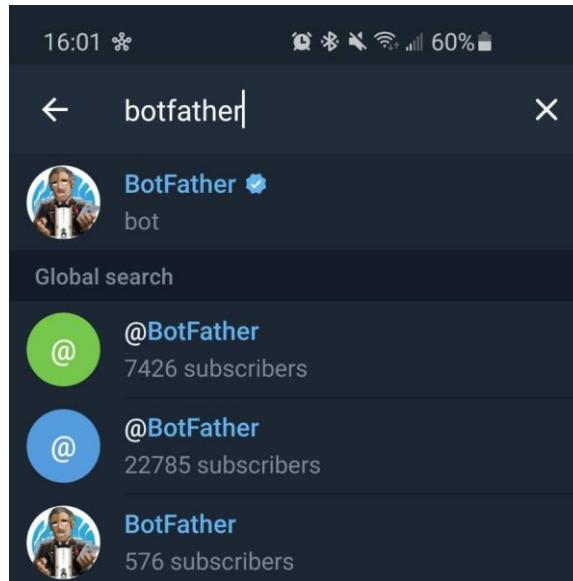
**Step 9:** Toggle “On/Off” switch to enable this feature. Click “Test” to show the test dialog up. Paste the copied SCKEY, copied in step 7, into the first field “WECHAT SC KEY”. Give a title to the second field “WECHAT MESSAGE TITLE”. Write some message content to the last field “WECHAT MESSAGE CONTENT”. And click “Test” to see if it works or not.



**Step 10:** Click “Save” button that shows in step 9 to keep your settings and enable WeChat event notification service.

#### 4.3.5 Steps to Set Event Notification Service – Telegram

**Step 1:** Search “BotFather” and start to chat on your Telegram App.



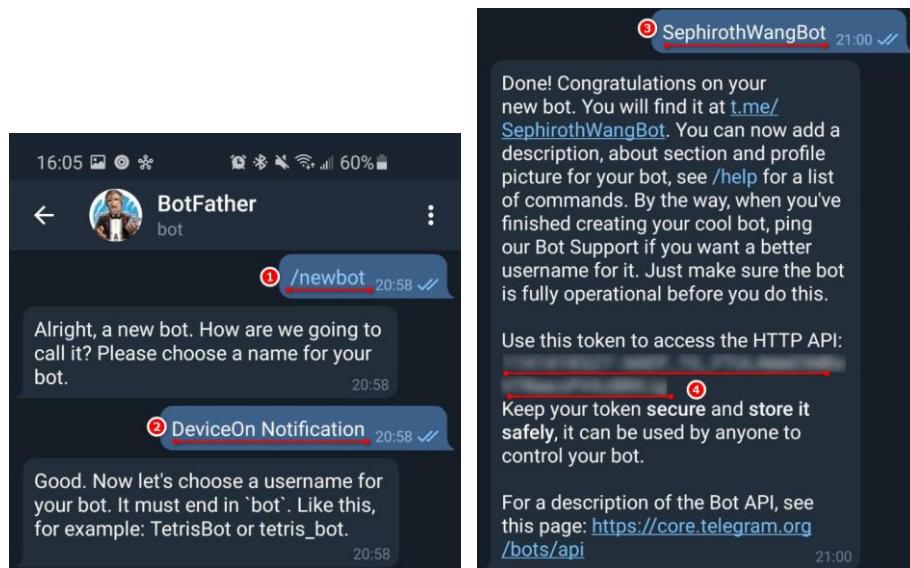
**Step 2:** Create a new bot and generate an **authorization token**.

Use the `/newbot` command to create a new bot. The BotFather will ask you for a name and username, then generate an authorization token for your new bot. The name of your bot is displayed in contact details and elsewhere.

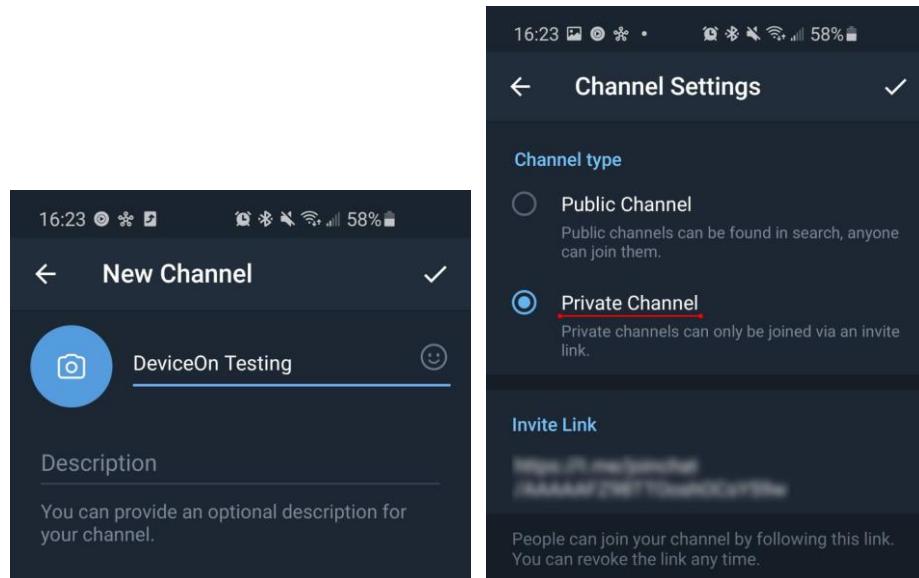
The Username is a short name, to be used in mentions and t.me links. Usernames are 5-32 characters long and are case insensitive, but may only include Latin characters, numbers, and underscores. Your bot's username must end in '**bot**', e.g. 'tetris\_bot' or 'TetrisBot'.

The token is a string along the lines of 110201543:AAHdqTcvCH1vGWJxfSeofSAs0K5PALDsaw that is

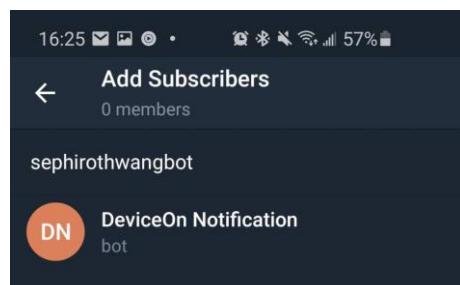
required to authorize the bot and send requests to the Bot API. Keep your token secure and store it safely, it can be used by anyone to control your bot.



## Step 2: Create your private channel on Telegram



## Step 3: Invite your bot into the channel.



## Step 4: Set your bot as “Administrators”



**Step 5:** Enter any txt message in the channel.



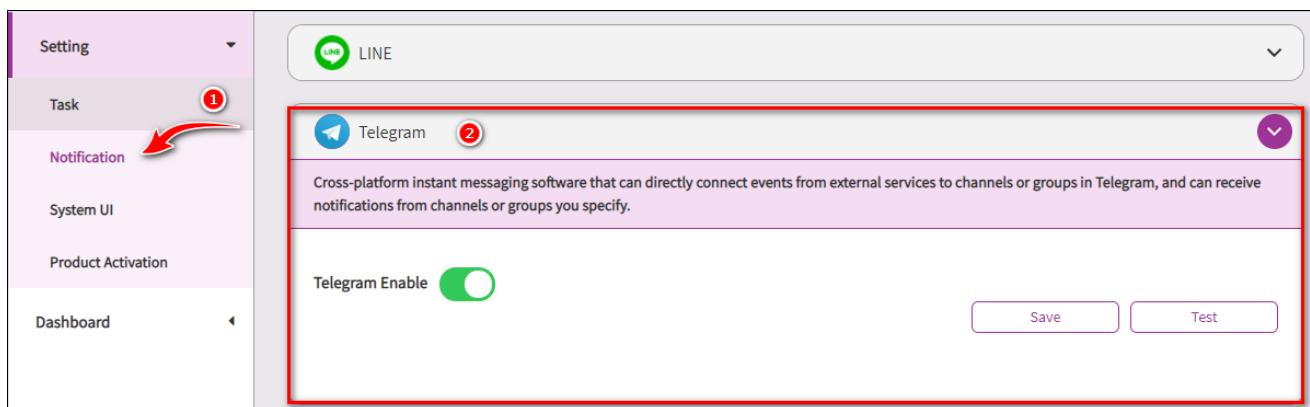
**Step 6:** Retrieve the **chat id** via below URL with your authorization token (Step 2).

⇒ <https://api.telegram.org/botTOKEN/getUpdates>

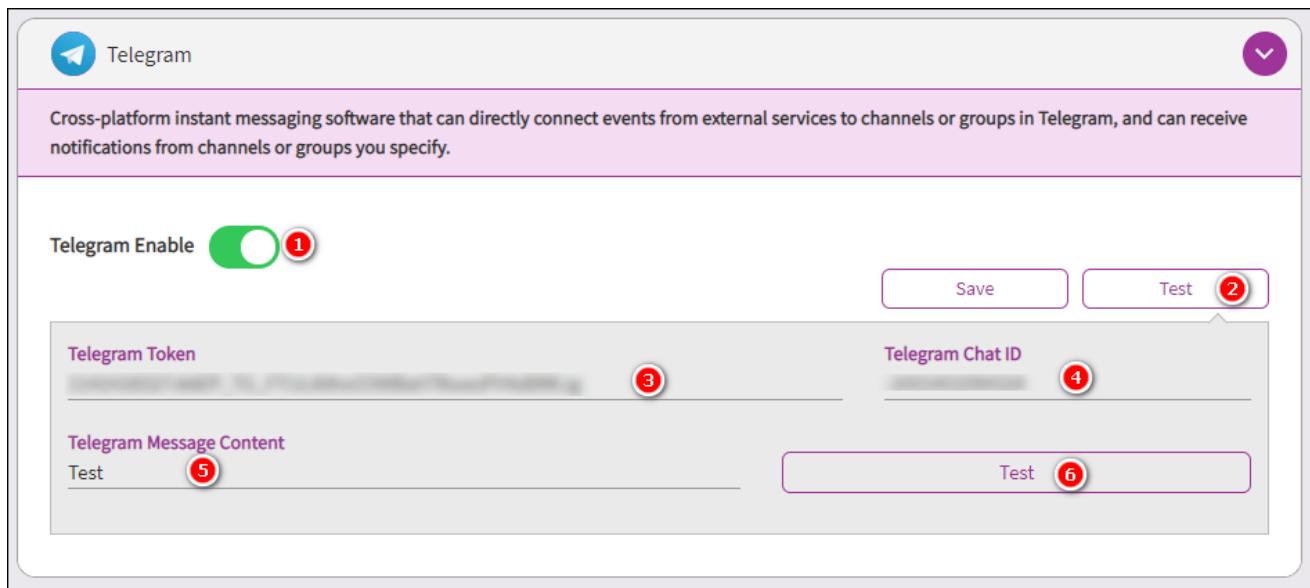
The response that include your chat id as below example.

```
1 // 20200908162812
2 // https://api.telegram.org/
3
4 {
5     "ok": true,
6     "result": [
7         {
8             "update_id": 526215976,
9             "channel_post": {
10                 "message_id": 2,
11                 "chat": {
12                     "id": 1599553687,
13                     "title": "DeviceOn Testing",
14                     "type": "channel"
15                 },
16                 "date": 1599553687,
17                 "text": "Test"
18             }
19         }
20     ]
21 }
```

**Step 7:** Now switch your browser to DeviceOn portal. Click “Setting” menu on the left-hand side, then “Notification”, and “Telegram” to open settings regarding Telegram event notification service.



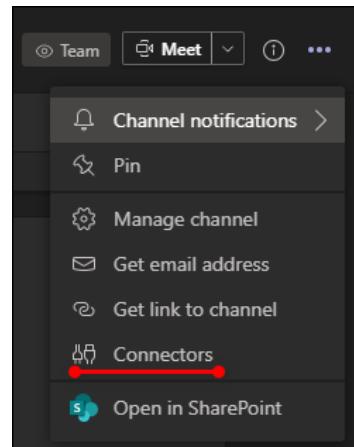
**Step 8:** Toggle “On/Off” switch to enable this feature. Click “Test” to show the test dialog up. Paste the copied Token and chat id, copied in step 2 and step 6. Give a title to the second field “**Telegram Message Content**”. Write some message content to the last field “**Test**”. And click “**Test**” to see if it works or not.



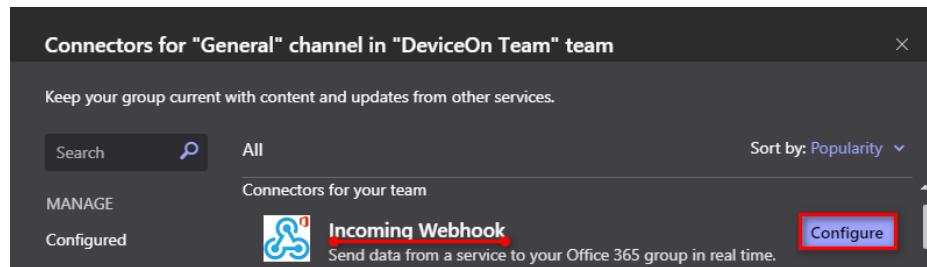
**Step 10:** Click “Save” button that shows in step 8 to keep your settings and enable Telegram event notification service.

#### 4.3.6 Steps to Set Event Notification Service – Microsoft Teams

**Step 1:** In the function menu of the channel where you want to send the message, select...(Other), and select the connector in the menu.



## Step 2: Select “Incoming Webhook”



Connectors for "General" channel in "DeviceOn Team" team

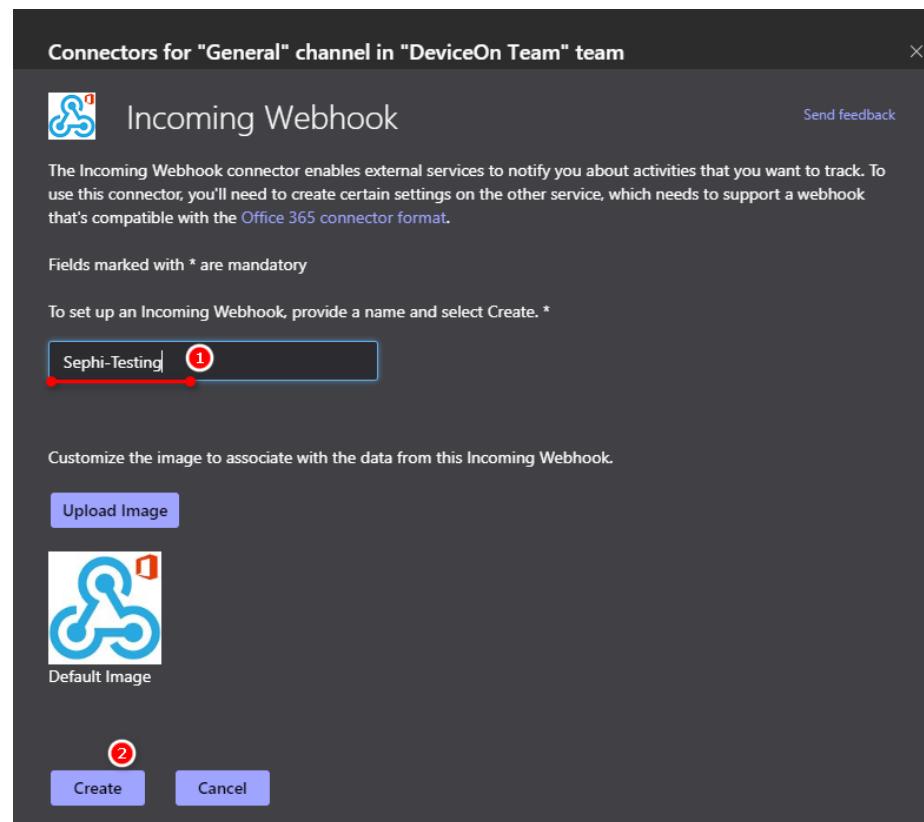
Keep your group current with content and updates from other services.

Search  All Sort by: Popularity ▾

MANAGE Connectors for your team

Configured	Incoming Webhook	Configure
	Send data from a service to your Office 365 group in real time.	<span style="border: 2px solid red; padding: 2px;">Configure</span>

## Step 3: Give this connector a name, then press the create button



Connectors for "General" channel in "DeviceOn Team" team

 Incoming Webhook Send feedback

The Incoming Webhook connector enables external services to notify you about activities that you want to track. To use this connector, you'll need to create certain settings on the other service, which needs to support a webhook that's compatible with the [Office 365 connector format](#).

Fields marked with \* are mandatory

To set up an Incoming Webhook, provide a name and select Create.\*

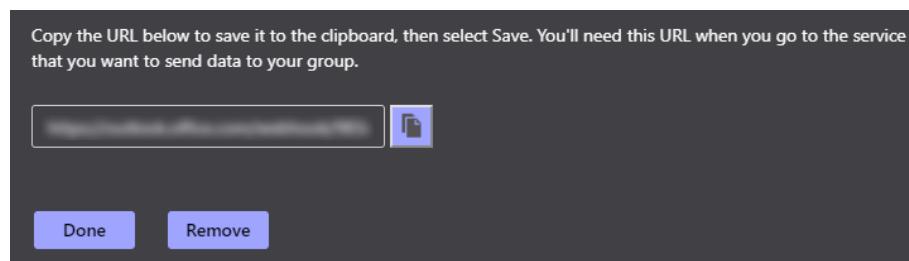
Sephi-Testing 1

Customize the image to associate with the data from this Incoming Webhook.

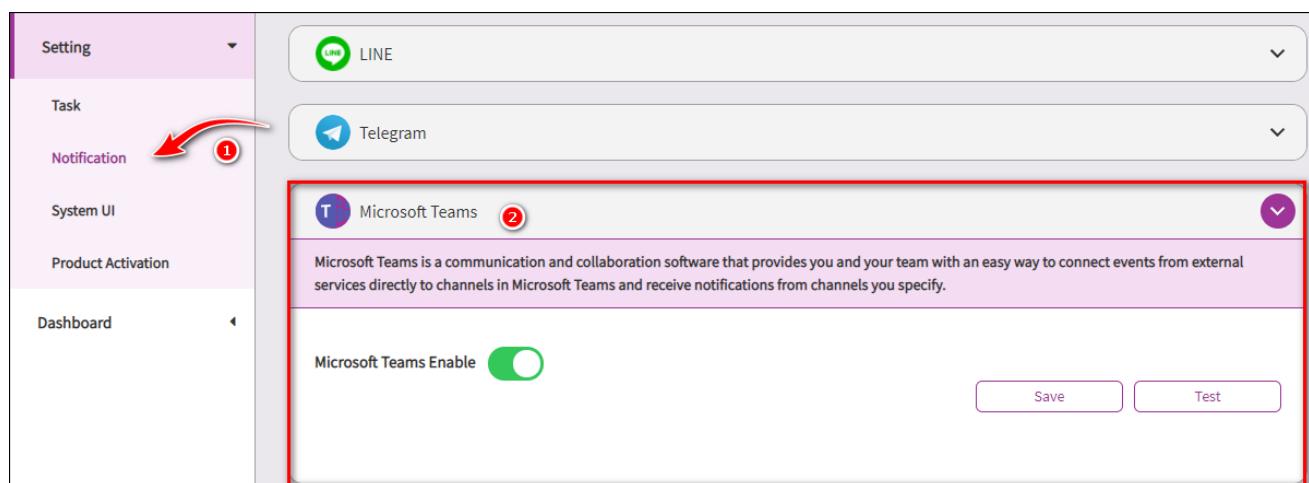
 Default Image

2

**Step 4:** At this time, a set of URLs will appear, which are used to transfer message. After copying, press the “Done” button.



**Step 5:** Now switch your browser to DeviceOn portal. Click “Setting” menu on the left-hand side, then “Notification”, and “Microsoft Teams” to open settings regarding Teams event notification service.



Setting

- Task
- Notification** 
- System UI
- Product Activation

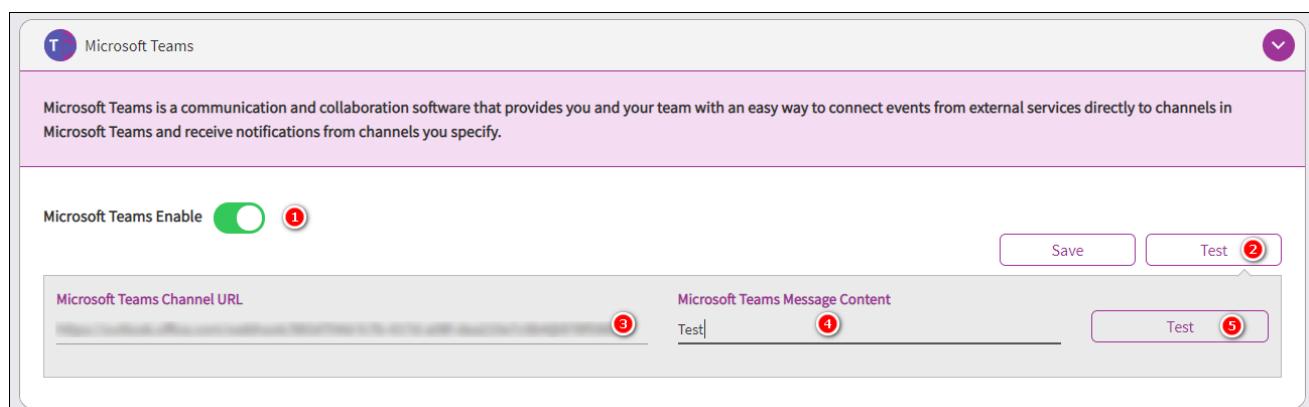
Microsoft Teams

Microsoft Teams is a communication and collaboration software that provides you and your team with an easy way to connect events from external services directly to channels in Microsoft Teams and receive notifications from channels you specify.

Microsoft Teams Enable 

Save Test

**Step 6:** Toggle “On/Off” switch to enable this feature. Click “Test” to show the test dialog up. Paste the URL, copied in step 4. Give a title to the second field “Microsoft Teams Message Content”. Write some message content to the last field “Test”. And click “Test” to see if it works or not.



Microsoft Teams

Microsoft Teams is a communication and collaboration software that provides you and your team with an easy way to connect events from external services directly to channels in Microsoft Teams and receive notifications from channels you specify.

Microsoft Teams Enable  

Save Test 

Microsoft Teams Channel URL 

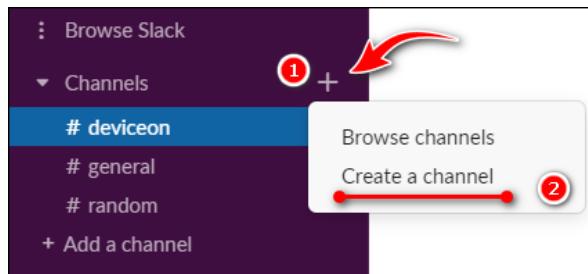
Microsoft Teams Message Content 

Test 

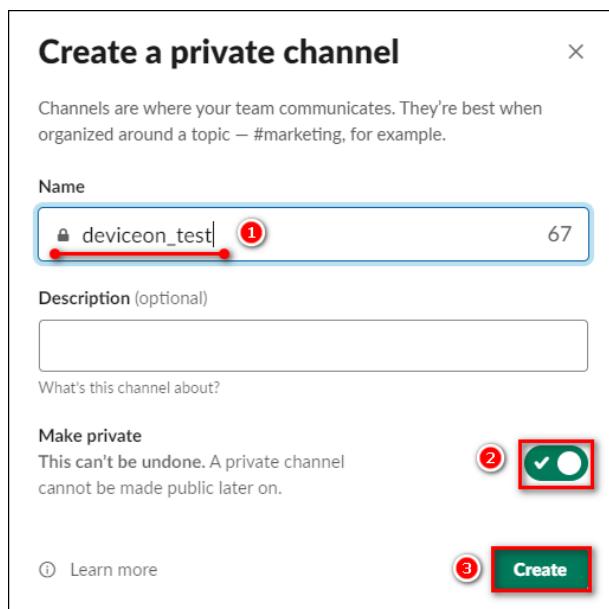
**Step 7:** Click “Save” button that shows in step 6 to keep your settings and enable Microsoft Teams event notification service.

#### 4.3.7 Steps to Set Event Notification Service – Slack

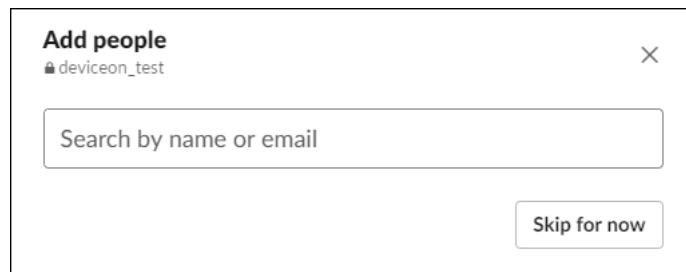
**Step 1:** Create your channel on your Slack.



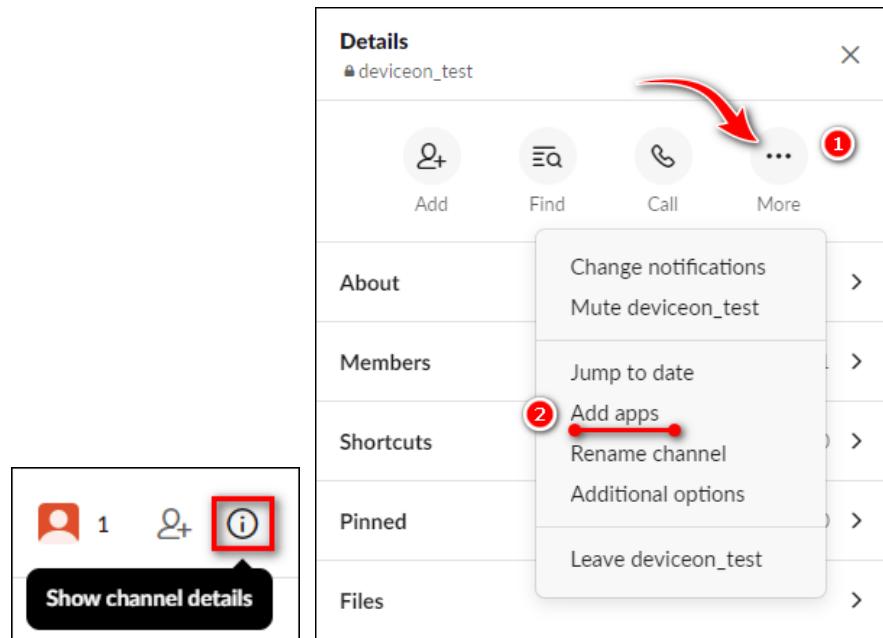
**Step 2:** Give this channel name and set as private.



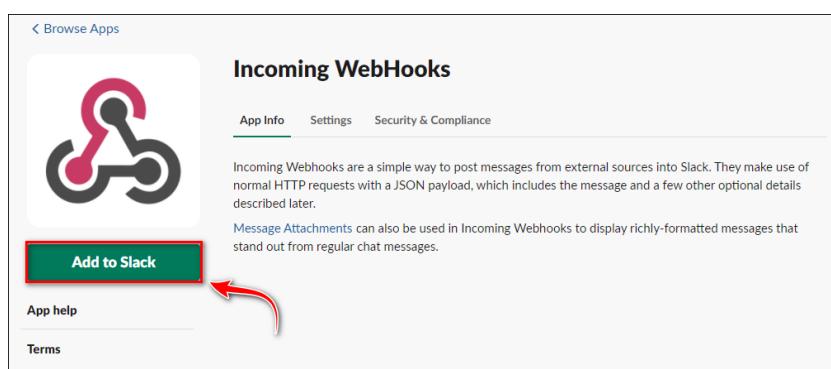
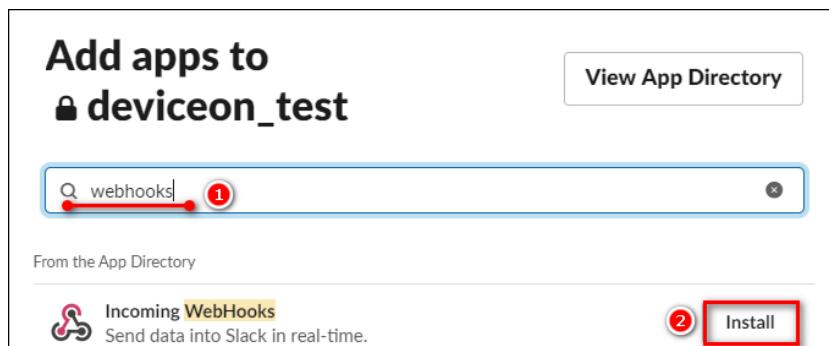
**Step 3:** Skip or add your member into channel.



**Step 4:** After logging in to slack, there will be a row of menus on the right, click "Apps" to expand the sub-menu, and then click "Add Apps"



**Step 5:** A search box will appear, type "webhooks" and you will see the first result is "**Incoming WebHooks**", then click to install and "**Add to Slack**".



**Step 6:** Click "**Add to Slack**", and a menu will appear asking which channel to install on. After selecting it, click "Incoming WebHooks integration".

**Post to Channel**

Start by choosing a channel where your Incoming Webhook will post messages to.

🔒 deviceon\_test 1

or create a new channel

**Add Incoming WebHooks integration** 2

By creating an incoming webhook, you agree to the [Slack API Terms of Service](#).

**Step 7:** After installation, you will enter the setting page of incoming webhooks. The first line of the page "**Webhook URL**" is the most important. We can send out automatic notification messages as long as we post to this url.

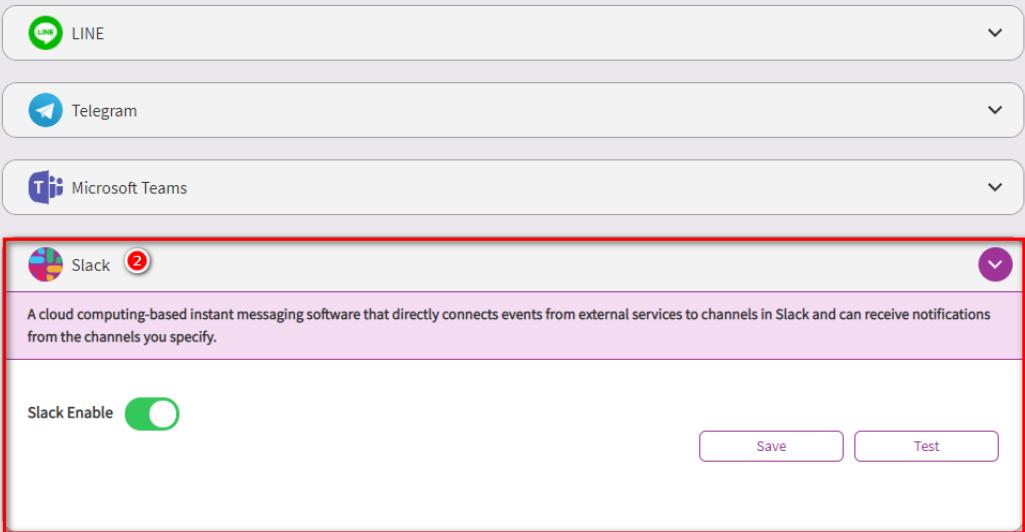
**Setup Instructions**

We'll guide you through the steps necessary to configure an Incoming Webhook so you can start sending data to Slack.

**Webhook URL**



**Step 8:** Now switch your browser to DeviceOn portal. Click "**Setting**" menu on the left-hand side, then "**Notification**", and "**Slack**" to open settings regarding Slack event notification service.



**Setting**

- Task
- Notification** ①
- System UI
- Product Activation

**Dashboard**

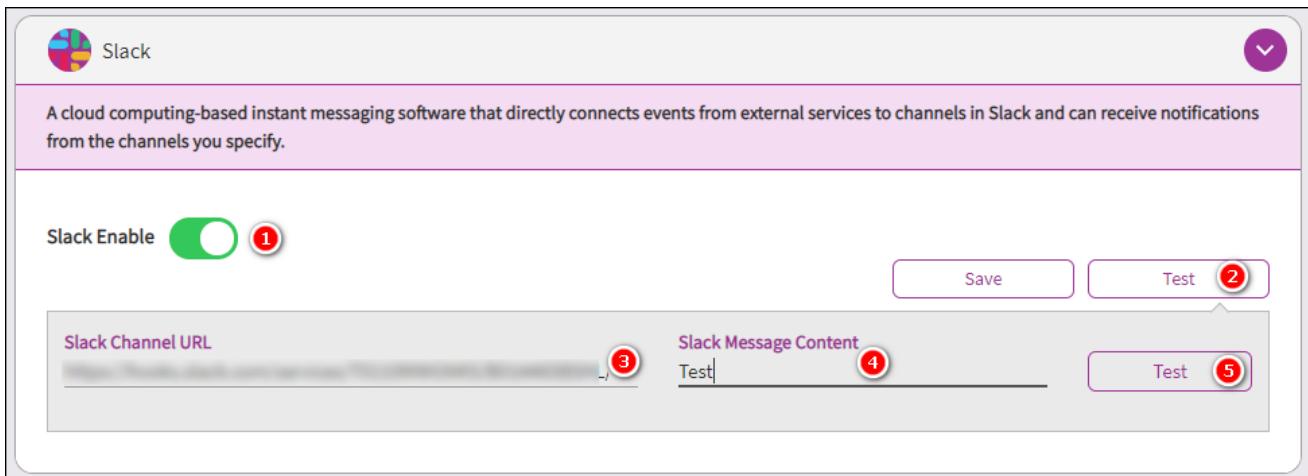
**Slack** ②

A cloud computing-based instant messaging software that directly connects events from external services to channels in Slack and can receive notifications from the channels you specify.

Slack Enable

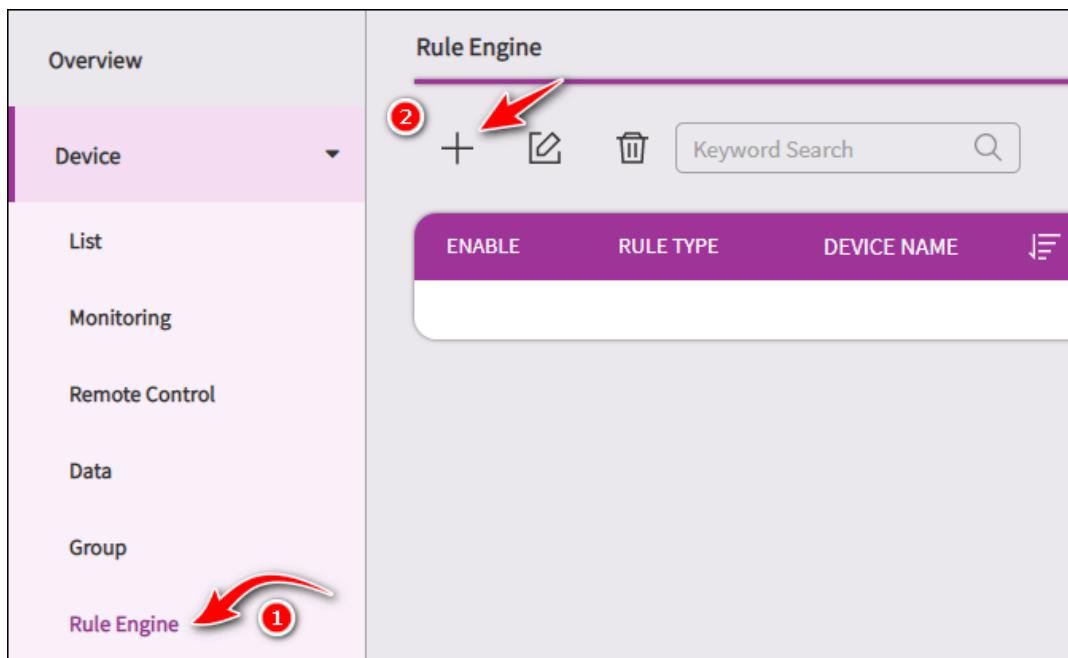
**Save** **Test**

**Step 9:** Toggle “On/Off” switch to enable this feature. Click “Test” to show the test dialog up. Paste the URL, copied in step 4. Give a title to the second field “Slack Message Content”. Write some message content to the last field “Test”. And click “Test” to see if it works or not.



#### 4.3.8 Steps to Set Thresholds to a Device

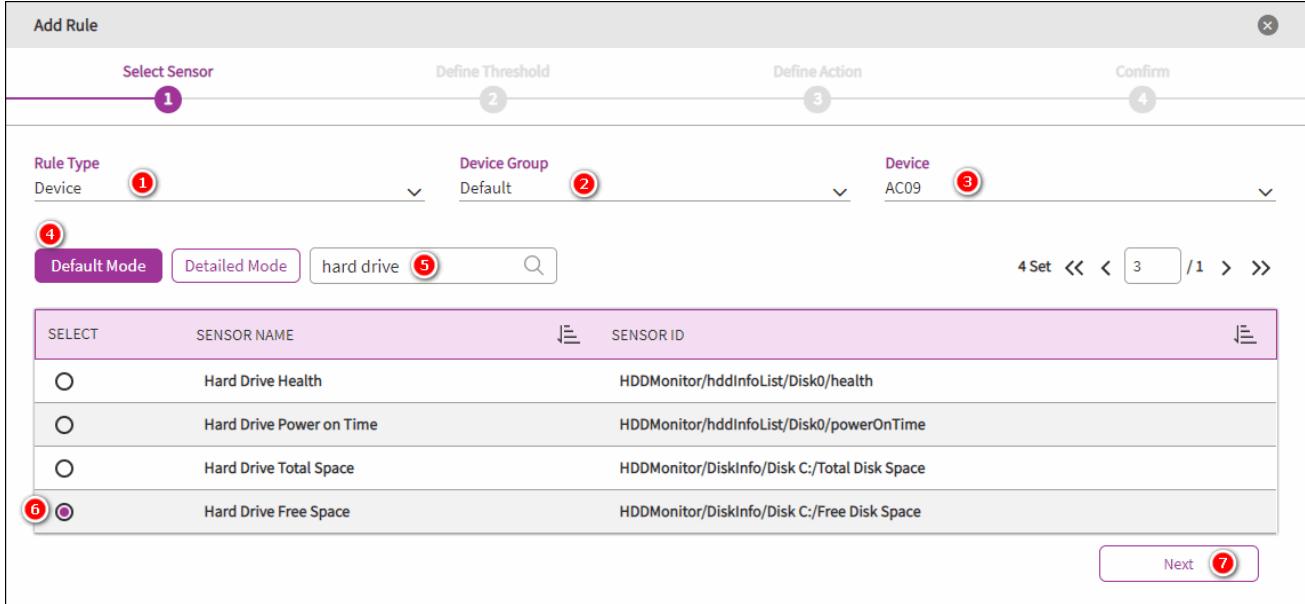
**Step 1:** Click “Setting” menu on the left-hand side of DeviceOn portal and, then, “Rule Engine”, click the plus (+) sign to enter “Rule Engine” page.



**Step 3:** Choose each setting with a proper value within step 1 – Select Sensor.

- **SELECT RULE TYPE:** Shows the new rule engine applies to a single device or a device group. Please pick “Device” here.
- **SELECT DEVICE GROUP:** Also, leave it the default, “Default”.
- **SELECT DEVICE:** Which device the new rule engine will apply? We choose “AA-Win” in this lab environment.

- **KEYWORD SEARCH:** Please enter “hard drive” so that only hard drive relevant items available. Here, to ease this lab, we pick “Hard Drive Free Space” as a threshold of the rule engine. In addition, like the picture shows, it illustrates the disk C is the target hard drive in this lab. Click “Next” to go to next step.



Add Rule

Select Sensor 1

Define Threshold 2

Define Action 3

Confirm 4

Rule Type 1  
Device

Device Group 2  
Default

Device 3  
AC09

Default Mode 4

Detailed Mode 5

hard drive 5 6

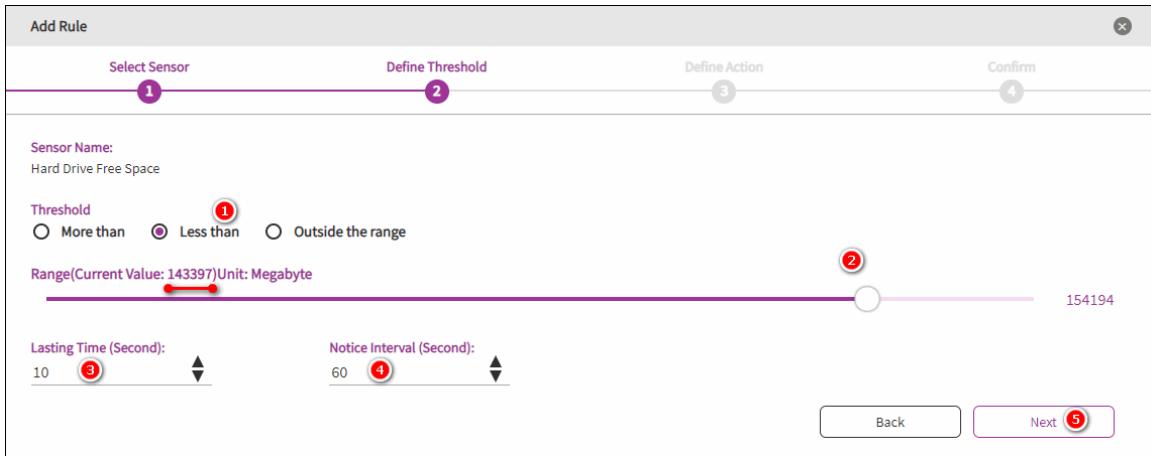
4 Set << < 3 /1 > >>

SELECT	SENSOR NAME	SENSOR ID
<input type="radio"/>	Hard Drive Health	HDDMonitor/hddInfoList/Disk0/health
<input type="radio"/>	Hard Drive Power on Time	HDDMonitor/hddInfoList/Disk0/powerOnTime
<input type="radio"/>	Hard Drive Total Space	HDDMonitor/DiskInfo/Disk C:/Total Disk Space
<input checked="" type="radio"/> <span style="color: #800080;">6</span>	Hard Drive Free Space	HDDMonitor/DiskInfo/Disk C:/Free Disk Space

Next 7

**Step 4:** Now we need to define a threshold for this rule engine in this step. Based on “Current Value” shows on top right, check the “Less than” radio button and slide to a maximum value that just on less than “Current Value”.

Leave “Lasting Time” as well as “Notice Interval” the defaults. “Lasting Time” indicates that the target device runs into the abnormal condition only when it reaches the set threshold and last the set time. While “Notice Interval” tells the interval of users receive an event, until the condition back to normal. Then click “Next” to go to next page.



Add Rule

Select Sensor 1

Define Threshold 2

Define Action 3

Confirm 4

Sensor Name:  
Hard Drive Free Space

Threshold 1  
 More than  Less than  Outside the range

Range(Current Value: 143397)Unit: Megabyte 2

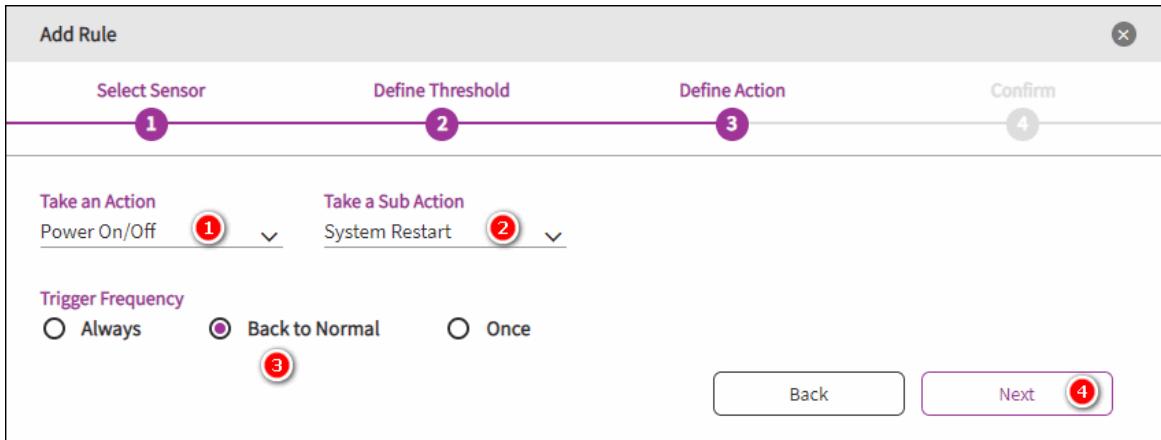
Lasting Time (Second): 3 10

Notice Interval (Second): 4 60

Back Next 5

**Step 5:** We are now in “Define Action” step. Pick “Power On/Off” from “TAKE A ACTION”, “System Restart” from “TAKE A SUB ACTION”, and “Back to Normal” for “Trigger Frequency”. These

combination means that the target device will reboot once it backs to normal, after it enters the threshold we set. Also, click “**Next**” to go to next page.



Add Rule

Select Sensor      Define Threshold      Define Action      Confirm

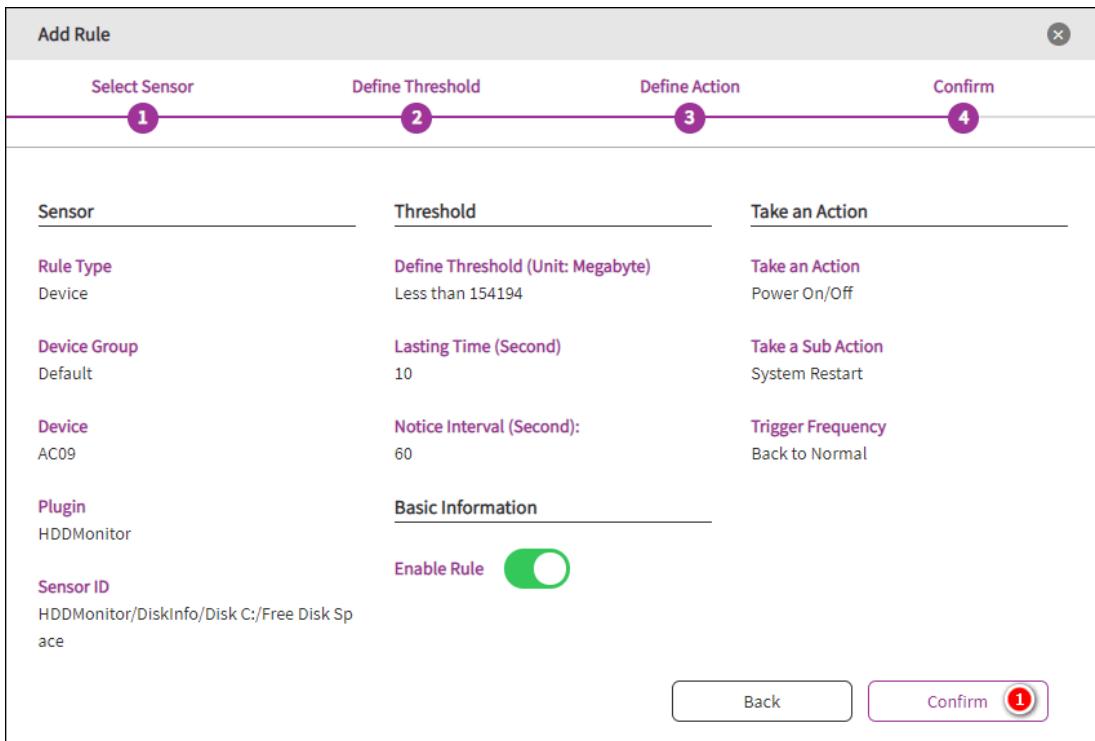
1      2      3      4

Take an Action  
Power On/Off (1) ▾ Take a Sub Action  
System Restart (2) ▾

Trigger Frequency  
○ Always    ● Back to Normal    ○ Once (3)

Back      Next (4)

**Step 7:** Review all information within this page. Leave “**Enable**” the default and click “**Confirm**” button to set this rule, and apply it to the target device as well.



Add Rule

Select Sensor      Define Threshold      Define Action      Confirm

1      2      3      4

Sensor	Threshold	Take an Action
Rule Type Device	Define Threshold (Unit: Megabyte) Less than 154194	Take an Action Power On/Off
Device Group Default	Lasting Time (Second) 10	Take a Sub Action System Restart
Device AC09	Notice Interval (Second): 60	Trigger Frequency Back to Normal
Plugin HDDMonitor	Basic Information	
Sensor ID HDDMonitor/DiskInfo/Disk C:/Free Disk Space	Enable Rule <input checked="" type="checkbox"/>	

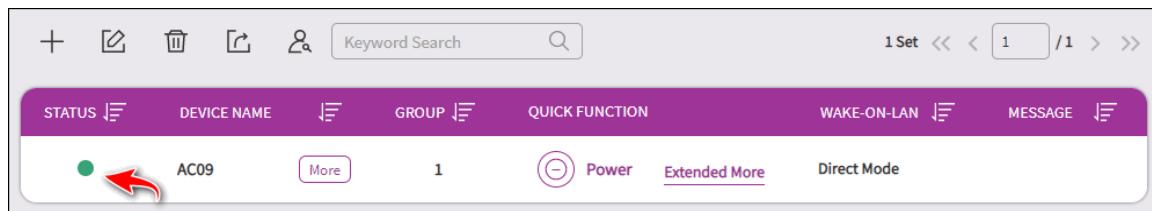
Back      Confirm (1)

**Step 8:** The new item should be populated as the image shows.

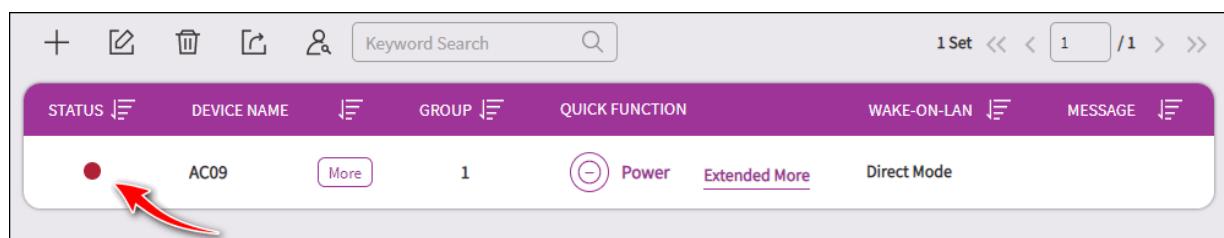


ENABLE	RULE TYPE	DEVICE NAME	SENSOR NAME	ACTION	THRESHOLD
<input checked="" type="checkbox"/>	Device	AC09	HDDMonitor/DiskInfo/Disk C:/Free Disk Space	Power On/Off–System Restart	Less than 154194 Megabyte

**Step 9:** Click “Device” menu item on left hand side of DeviceOn portal. You can see a green circle represents the target device accordingly.



**Step 10:** We can do something so that the target device meets the threshold we set previous. Here we download the newest Ubuntu ISO image to the target device. The green circle shows in step 9 changes, a while later, to an orange one, of which indicates it runs into an abnormal condition.



**Step 10:** Interrupt the download action at any time, or wait until it finishes. Purge the downloaded file so that the target device has free space more than the threshold we set previous. After a while, the target device should reboot due to the rule engine we set. Note here that it may necessary to purge the recycle bin to achieve our goal.

## 4.4 How to Visualize Device Data via Grafana Dashboard

Grafana is an open-source software for monitoring and analysis. One of its major characteristics is it supports many different data sources, from popular CloudWatch, Elasticsearch, Graphite, and influxDB, to OpenStack Gnocchi or Google Calendar. Its range is very extensive. However, for others data source require to implement [SimpleJson](#) to access your data. The DeviceOn native support SimpleJson APIs and data source plugin on Grafana. This lab guides you how to visualize device data via Grafana dashboard.

### 4.4.1 Prerequisite

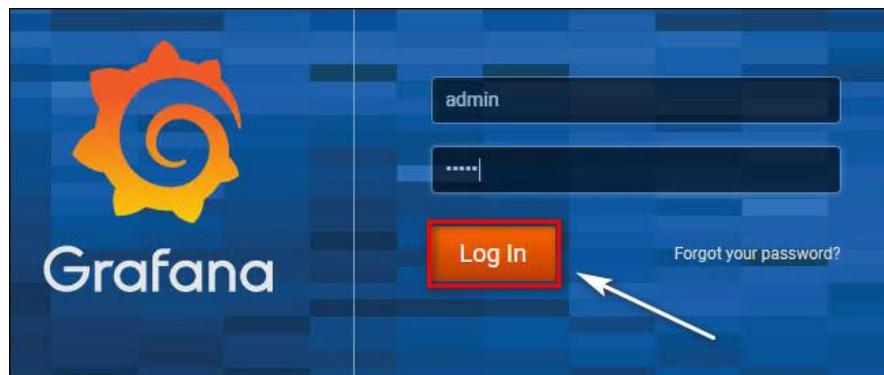
- A running DeviceOn server.
- A running Grafana service with DeviceOn data source plugin.
- A device which installed WISE-Agent, that connects to DeviceOn server.

### 4.4.2 Step-by-Step

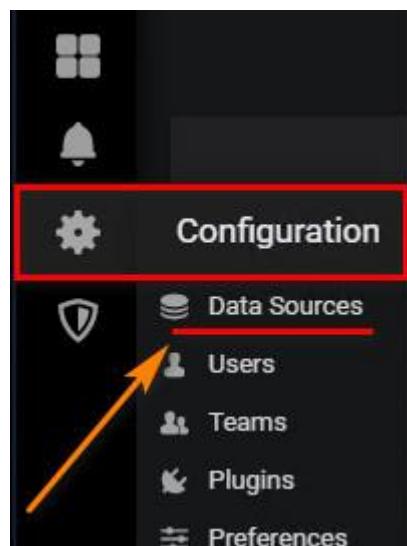
**Step 1:** Launch Grafana Web Service Shortcut on Desktop, or access the Grafana service endpoint.



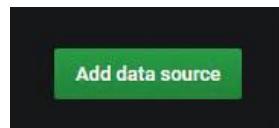
**Step 2:** Login to Grafana portal with your account, password (Default: admin/admin)

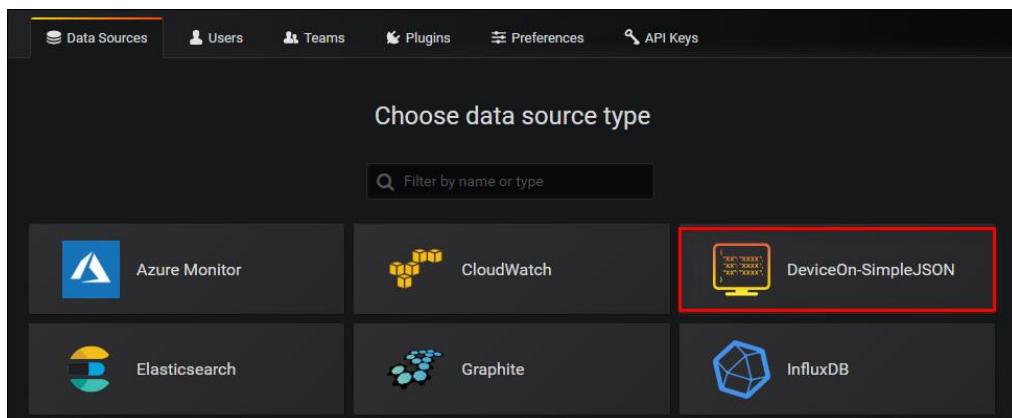


**Step 3:** Create a data source to access DeviceOn SimpleJson API.



Click on “Add data source” and select “DeviceOn-SimpleJson”, (for previous version might be RMM-SimpleJson)





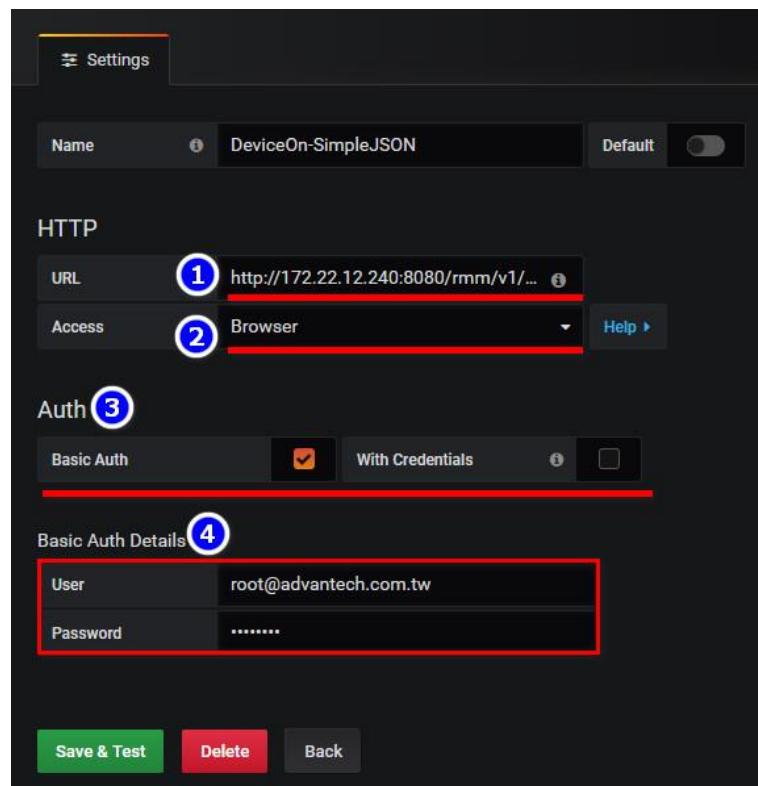
**Step 4:** Given below parameters for data source plugin to retrieve device data from DeviceOn APIs.

**URL:** [http://<DEVICEON\\_SERVER>:8080/rmm/v1/grafana/simplejson](http://<DEVICEON_SERVER>:8080/rmm/v1/grafana/simplejson)

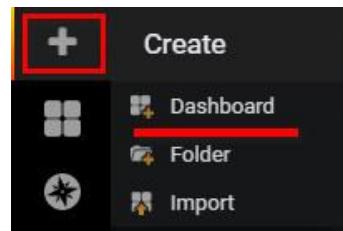
**Access:** Browser

**Auth:** Basic Auth (Support on prefecture version)

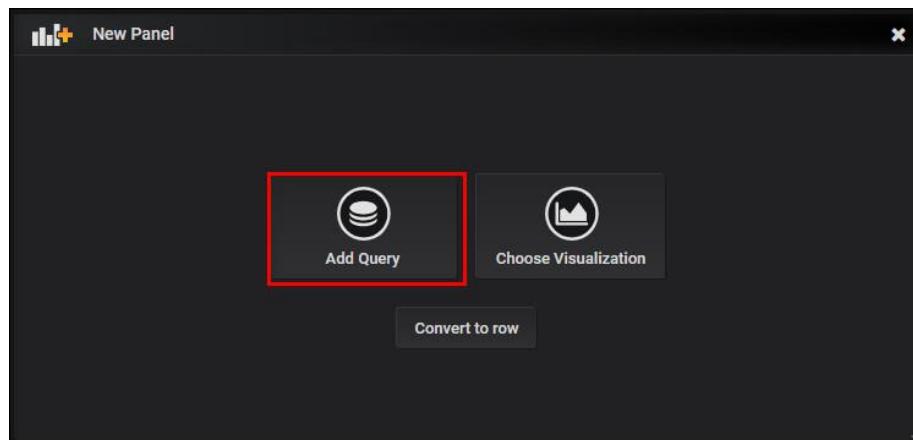
**Basic Auth:** DeviceOn Account & Password



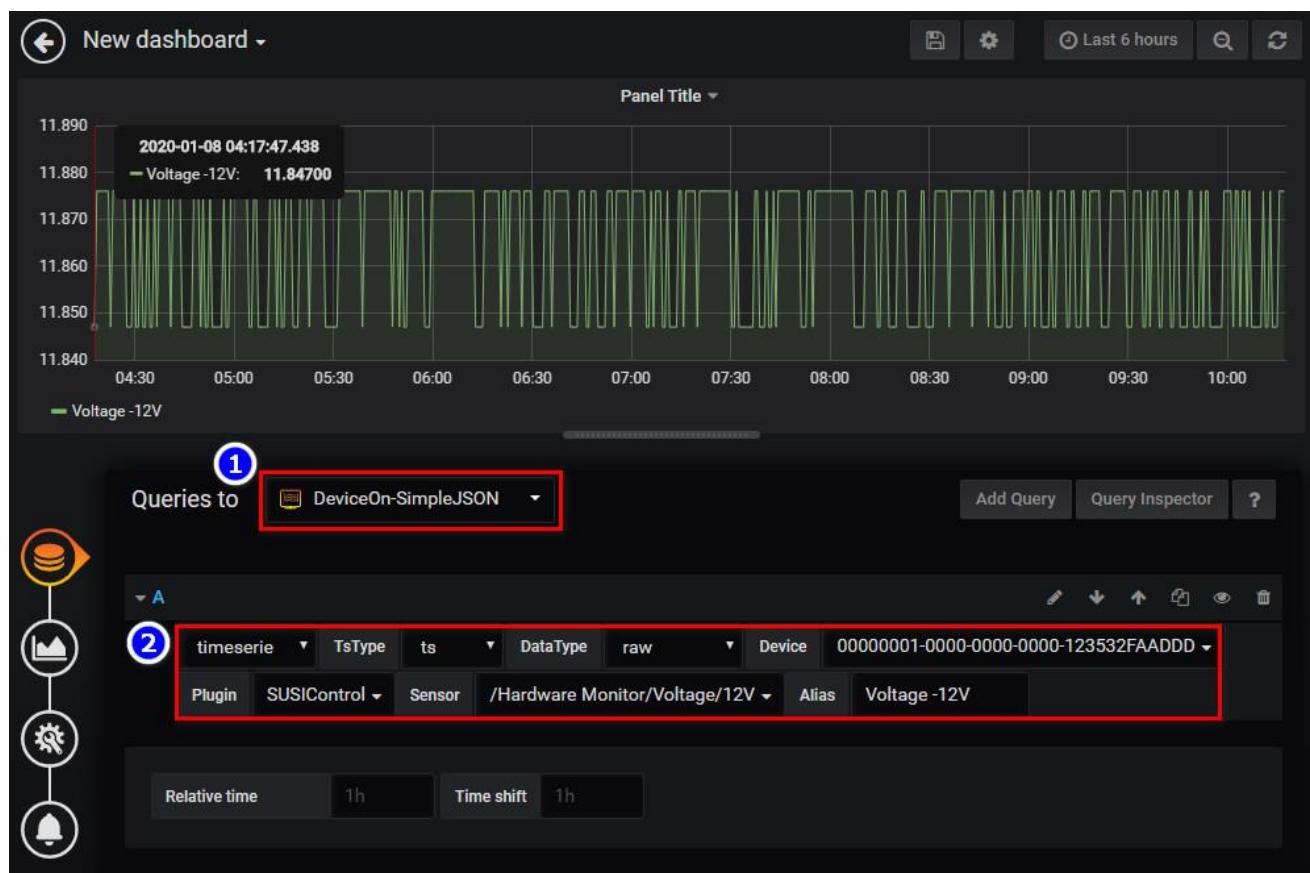
**Step 5:** Create a dashboard to visualize your device data.



Select “Add Query” for your device.



Select **DeviceOn-SimpleJson** from “**Queries to**”, and pick-up your device with **AgentID**, **Plugin**, **Sensor** and **Alias Name** (Option).



## 4.5 How to Enable/Disable Windows Lockdown Features

For devices protection, Windows built many nice features in natively. For instance, function key protection disables Ctrl, Alt, and WinKey. UWF protection guarantees your disk C (System Partition) rollbacks to the original state after you reboot the Windows operating system. This lab guides you how to enable Windows lockdown features, and how to active/inactive them via DeviceOn portal.

After this lab, you should:

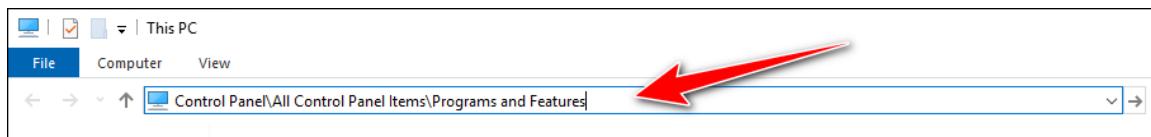
- Learn how to enable “**Keyboard Filter**” and “**Unified Write Filter**” (a.k.a. UWF) in Windows lockdown features.
- Know what lockdown features can be controlled via DeviceOn portal.

### 4.5.1 Prerequisite

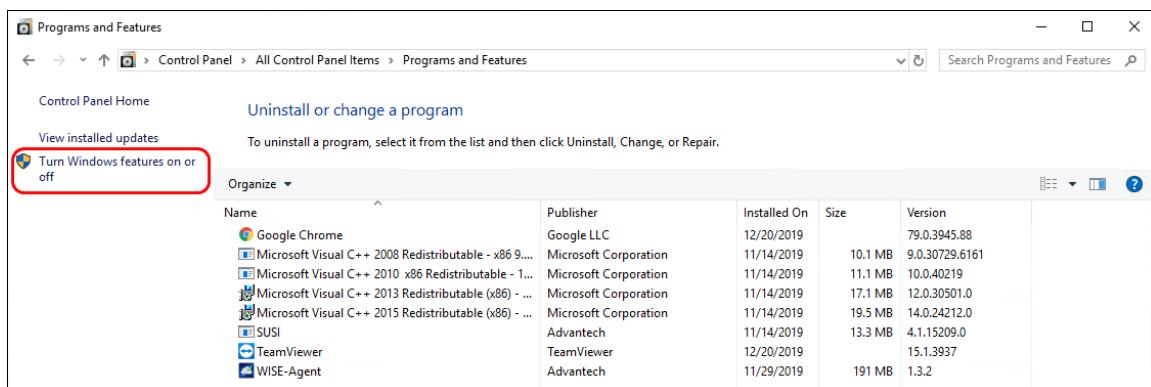
- A running DeviceOn server.
- A device which running on Windows 10 operating system (LTSB, LTSC) and installed WISE-Agent, that connects to DeviceOn server. Besides, this agent must install Advantech SUSI driver, or lockdown feature should not work properly.

### 4.5.2 Step-by-Step

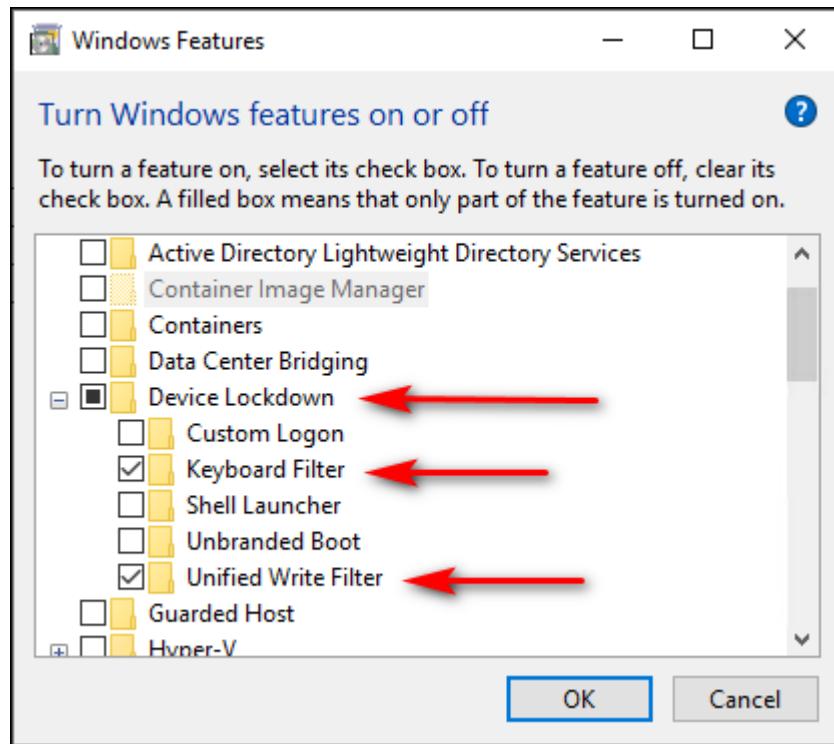
**Step 1:** Go to the target agent device and open the file explorer window. In address bar, key “**Control Panel\All Control Panel Items\Programs and Features**” in and followed by pressing “**ENTER**”. It opens the “**Programs and Features**” window.



**Step 2:** Click “**Turn Windows features on or off**” on left hand side to open “**Windows Features**” window.



**Step 3:** Scroll down the window, find and open the “**Device Lockdown**” item. Make sure both “**Keyboard Filter**” and “**Unified Write Filter**” are checked. Then click “**OK**”.



**Step 4:** Now back to DeviceOn portal. Click “**Device**” menu item, then “**Remote Control**” tab. And choose proper account, group, and device from “**SELECT ACCOUNT**”, “**SELECT DEVICE GROUPS**”, and “**SELECT DEVICE**” fields accordingly. You can see “**Function Key**”, “**UWF Protection**” control buttons there. Also, other than these two mentioned, “**WatchDog Protection**”, “**Windows Notification**” and more relevant features are available as you can see.

The screenshot shows the DeviceOn portal's 'Remote Control' tab. On the left, a sidebar has 'Device' selected. The main area shows several control panels:

- Screen Shot**: Go To Screen Shot
- Terminal**: Go To Terminal
- Remote Desktop**: Go To Remote Desktop
- Audio Volume Control**: A slider set to 99.
- LVDS Screen Brightness**: A slider set to 0.
- USB Drives**: Lockdown button is off.
- Function Key** (highlighted): Ctrl, Alt, Winkey Lockdown. Status: Enabled (button is green).
- Hardware Watchdog** (highlighted): Disabled (button is grey).
- Windows Notification** (highlighted): Disabled (button is grey).
- Touch Screen**: Enabled.
- Touch Gesture**: Disabled.
- UWF Protection** (highlighted): Disabled (button is grey).

**Step 5:** Click “**Function Key**” control button. You would find, after a while, the description of “**Function Key**” changes from “**Available**” to “**Ctrl, Alt, WinKey Lockdown**”. If you try to press such keys on the target device, they should not work as expected. Okay, you learned how to enable, disable “**Function Key**” lockdown. Let’s go ahead and learn something regarding UWF.

**Step 6:** Click “**UWF Protection**” control button. A dialog pops up and the message shows that this action will reboot the device. Click “**CONFIRM**”, its description changes from “**Disabled**” to “**Enabled**”. Just wait for the reboot completed.

**Step 7:** Now, write some data into disk C. You can, for example, download files into disk C, copy files into disk C. Or even generate by programmatically. Just do whatever you can do to mimic that you are working on disk C.

**Step 8:** Once you finish your tasks, reboot the target device. You would find that all those data you made at previous step disappear. The disk C rollbacks to the original state and just like you did nothing at all.

## 4.6 How to Manage DeviceOn on AKS

Since the DeviceOn service container are running the Azure Kubernetes, that’s much easier to upgrade to new version, if released. There

### 4.6.1 Prerequisite

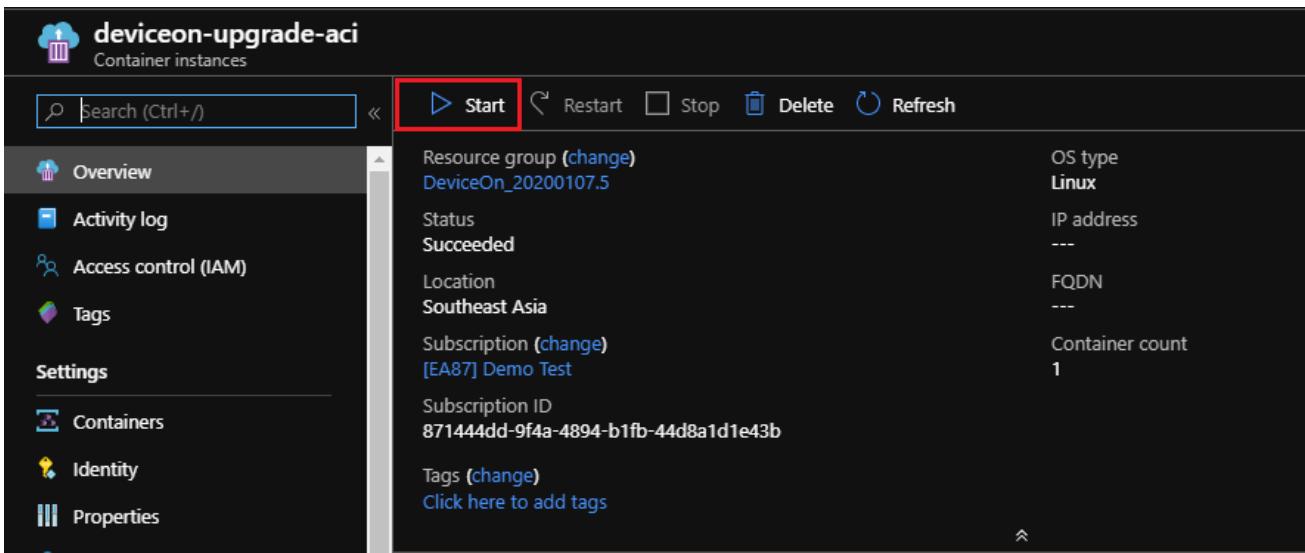
- A running DeviceOn server on Azure Kubernetes
- Azure Account

### 4.6.2 Steps to Upgrade DeviceOn

**Step 1:** Login to Azure Portal and find your AKS solution resource group.

**Step 2:** Click **deviceon-upgrade-aci** service.

**Step 3:** Click **Start** button to upgrade latest version DeviceOn server.



**deviceon-upgrade-aci**  
Container instances

**Start** | Restart | Stop | Delete | Refresh

- Overview
- Activity log
- Access control (IAM)
- Tags
- Settings
- Containers
- Identity
- Properties

Resource group (change)  
DeviceOn\_20200107.5

Status  
**Succeeded**

Location  
**Southeast Asia**

Subscription (change)  
[EA87] Demo Test

Subscription ID  
871444dd-9f4a-4894-b1fb-44d8a1d1e43b

Tags (change)  
Click here to add tags

OS type  
**Linux**

IP address  
---

FQDN  
---

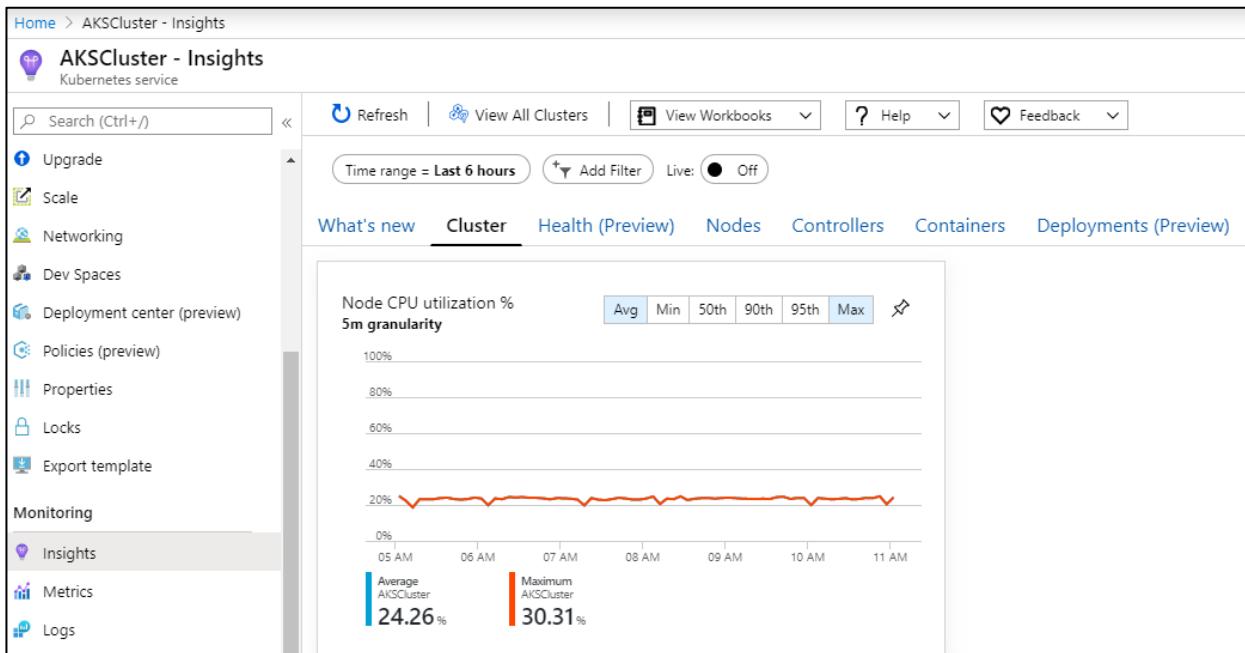
Container count  
**1**

**Step 4:** After upgrade container instances finish, please go to DeviceOn portal check server version.

#### 4.6.3 Step to Monitor Container Healthy and Status

##### ● Monitor Container Status

**Step 1:** In AKS service, select “Insights” on the left tab.



Home > AKSCluster - Insights

**AKSCluster - Insights**  
Kubernetes service

Search (Ctrl+/  
Refresh | View All Clusters | View Workbooks | Help | Feedback

Time range = Last 6 hours | Add Filter | Live: Off

What's new Cluster Health (Preview) Nodes Controllers Containers Deployments (Preview)

Node CPU utilization %  
5m granularity

Avg Min 50th 90th 95th Max

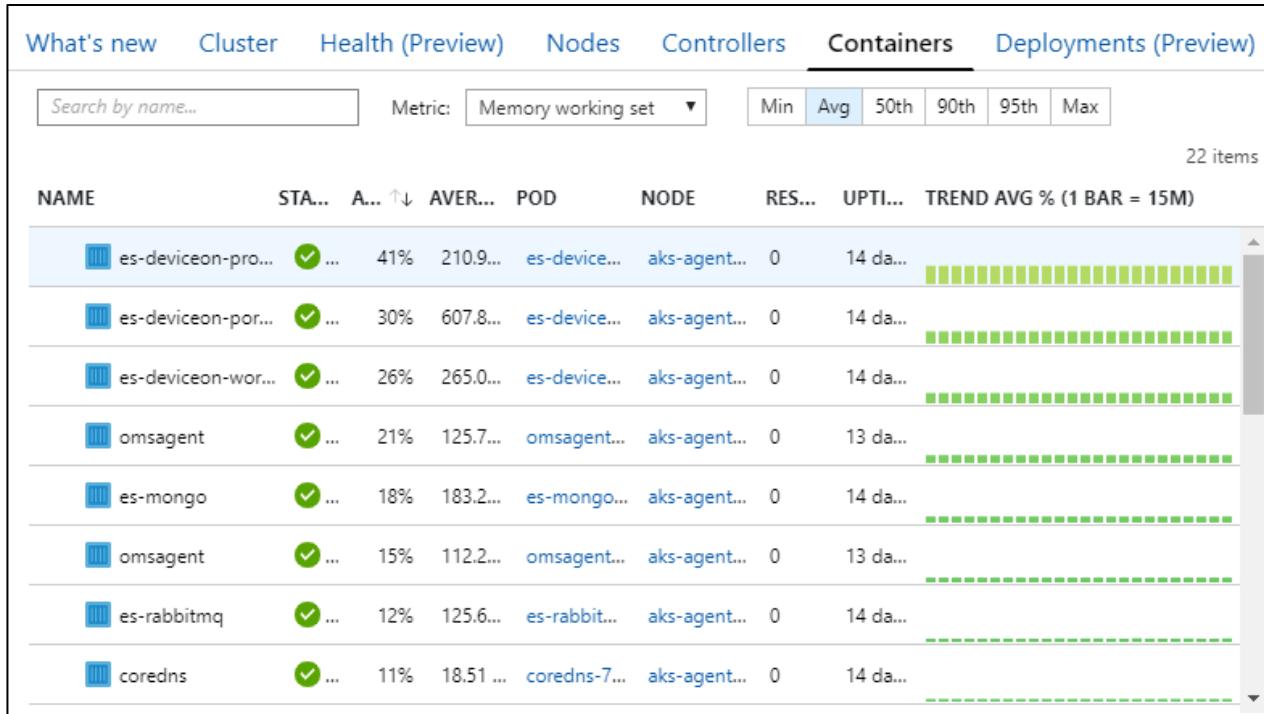
100%  
80%  
60%  
40%  
20%  
0%

05 AM 06 AM 07 AM 08 AM 09 AM 10 AM 11 AM

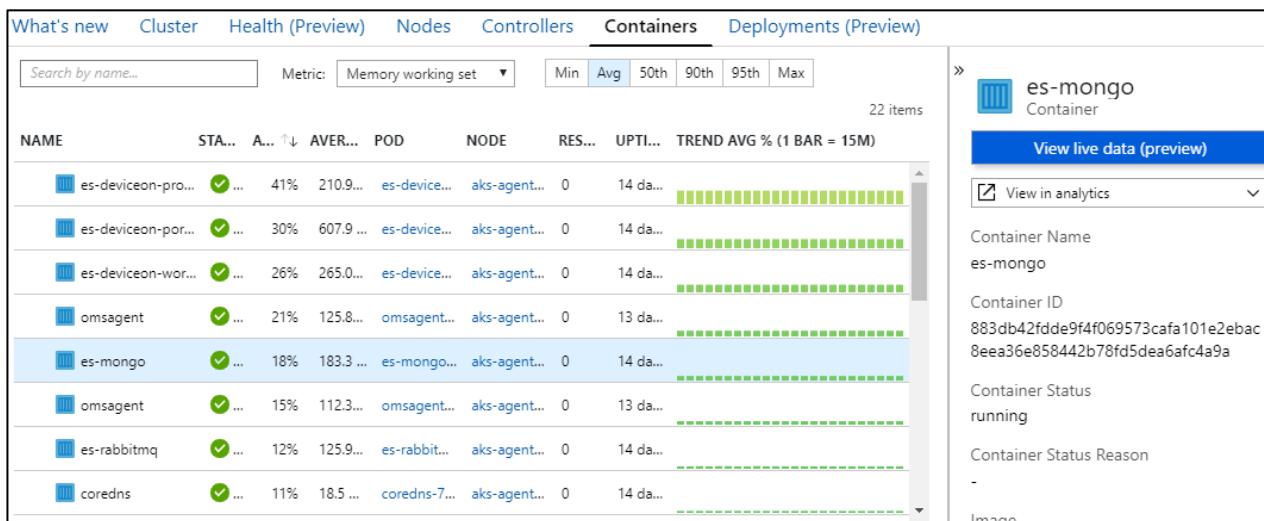
Average AKSCluster: 24.26 % Maximum AKSCluster: 30.31 %

**Step 2:** Click on “Containers” on the top tab. Check if status of each container is running.

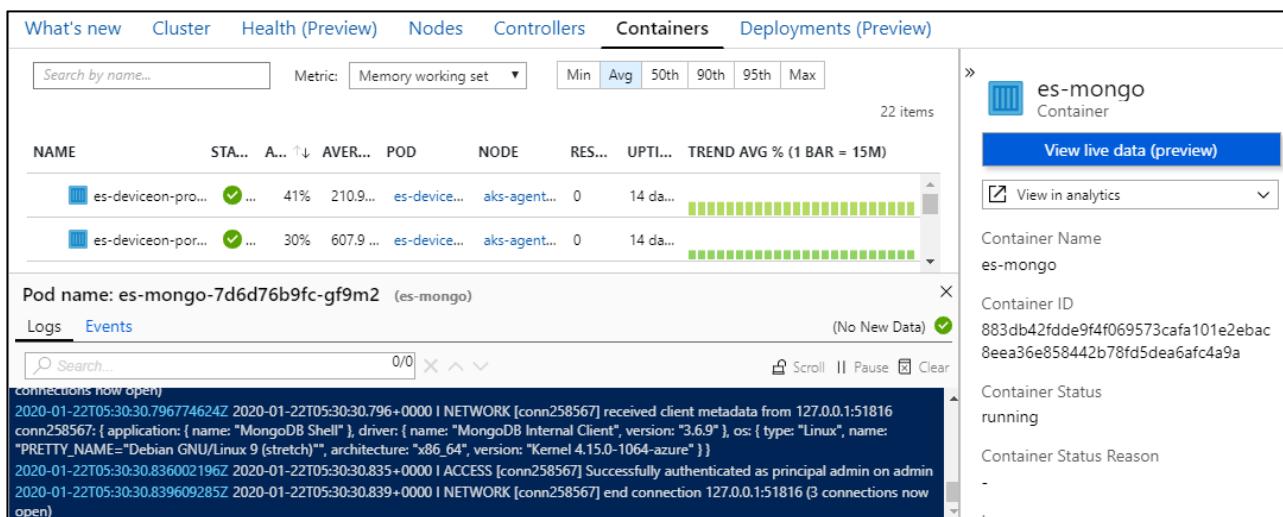
\*\* Init container will show completed. \*\*



**Step 3:** View the container log by clicking on container name. Take es-mongo for example.



**Step 4:** Click on “View live data” to view live log of container.



## ● Monitor Usage of Persistent Volume (PV)

### Step 1: Open PowerShell and login

Follow the instruction to login your account

```
# az login
```

Select your subscription, if you have multiple subscription, you have to set which subscription AKS service is located.

```
# az account set --subscription "SUBSCRIPTION_NAME"
```

Try to install AKS command line.

```
# az aks install-cli
```

To get AKS credential to access.

```
# az aks get-credentials --resource-group RESOURCE_GROUP --name K8S_CLUSTER
```

### Step 2: Use kubectl command to get k8s information

```
# kubectl get pod --all-namespaces (check all pods)
```

NAMESPACE	NAME	READY	STATUS	RESTARTS	AGE
default	es-deviceon-portal-6d6574b65-tp744	1/1	Running	0	13d
default	es-deviceon-provisioning-fc74b9cd7-cnwl	1/1	Running	0	13d
default	es-deviceon-worker-5f88997ccb-292pl	1/1	Running	0	13d
default	es-mongo-7d6d76b9fc-gf9m2	1/1	Running	0	13d
default	es-rabbitmq-7cb946db6c-5sqk5	1/1	Running	0	13d
default	rmm-iothub-bridge-6cf98f5f4-cvqf4	1/1	Running	13	13d
kube-system	coredns-7fc597cc45-ht4jc	1/1	Running	0	13d
kube-system	coredns-7fc597cc45-r9814	1/1	Running	0	13d
kube-system	coredns-autoscaler-7ccc76bfbd-bjwpd	1/1	Running	0	13d
kube-system	kube-proxy-zsfmx	1/1	Running	0	7d9h
kube-system	kubernetes-dashboard-6fbc7f598b-klcgb	1/1	Running	5	13d
kube-system	metrics-server-58b6fcfd54-8jpnn	1/1	Running	0	13d
kube-system	omsagent-nwrnl	1/1	Running	0	12d
kube-system	tiller-deploy-59b99695d8-2x4pf	1/1	Running	0	13d

Try to shows all PVs in the cluster, it also includes which PVCs (Persistent Volume Claim) are requesting for the resources.

```
# kubectl get pv --all-namespaces (check PV)
```

NAME	STORAGECLASS	REASON	AGE	CAPACITY	ACCESS MODES	RECLAIM POLICY	STATUS	CLAIM
pvc-e967a374-3115-11ea-a6bc-6201b0e63488	managed-premium		13d	500Gi	RWO	Delete	Bound	default/es-mongo
pvc-e9eb4084-3115-11ea-a6bc-6201b0e63488	es	managed-premium	13d	8Gi	RWO	Delete	Bound	default/es-postg
pvc-ea66f281-3115-11ea-a6bc-6201b0e63488	mq	managed-premium	13d	32Gi	RWO	Delete	Bound	default/es-rabbi

Take es-mongo as example, the capacity of es-mongo PV is 500Gi, and default/es-mongo is requesting it as PVC. If you'd like to know the PV usage, you need to access es-mongo container.

```
# kubectl exec -it es-mongo-7d6d76b9fc-gf9m2 -- /bin/bash
```

Please replace the **pod** name to yours. To display disk available space on the file system, you could enter:

```
# df -h
```

```
root@es-mongo-7d6d76b9fc-gf9m2:/# df -h
Filesystem      Size  Used Avail Use% Mounted on
overlay        97G   19G   79G  20% /
tmpfs          64M     0   64M   0% /dev
tmpfs          7.9G     0   7.9G   0% /sys/fs/cgroup
/dev/sdc       493G  448M  492G  1% /data/db
/dev/sda1       97G   19G   79G  20% /etc/mongo
shm            64M     0   64M   0% /dev/shm
tmpfs          7.9G   12K   7.9G   1% /run/secrets/kubernetes.io/serviceaccount
tmpfs          7.9G     0   7.9G   0% /proc/acpi
tmpfs          7.9G     0   7.9G   0% /proc/scsi
tmpfs          7.9G     0   7.9G   0% /sys/firmware
```

#### 4.6.4 Steps to Expose Database/RabbitMQ to Access

**Step 1:** Download [Database/RabbitMQ](#) yaml files.

**Step 2:** Open PowerShell and login

Follow the instruction to login your account

```
# az login
```

Select your subscription, if you have multiple subscription, you have to set which subscription AKS service is located.

```
# az account set --subscription "SUBSCRIPTION_NAME"
```

Try to install AKS command line.

```
# az aks install-cli
```

To get AKS credential to access.

```
# az aks get-credentials --resource-group RESOURCE_GROUP --name K8S_CLUSTER
```

**Step 3:** Expose MongoDB/PostgreSQL/RabbitMQ

```
# kubectl create -f service-mongodb.yaml  
  
# kubectl create -f service-postgres.yaml  
  
# kubectl create -f service-rmq.yaml
```

**Step 4:** Use Kubernetes Dashboard to check service public address

```
# az aks browse --resource-group RESOURCE_GROUP --name K8S_CLUSTER
```



Service	Ports	Status	Actions	
es-rabbitmq-worldwide	app: rabbitmq heritage: Tiller release: es-rabbitmq 10.0.230.154	es-rabbitmq-worldwide:5672 TCP es-rabbitmq-worldwide:30050 TCP es-rabbitmq-worldwide:5671 TCP es-rabbitmq-worldwide:30051 TCP es-rabbitmq-worldwide:15672 TCP es-rabbitmq-worldwide:30052 TCP es-rabbitmq-worldwide:1983 TCP es-rabbitmq-worldwide:30053 TCP es-rabbitmq-worldwide:8883 TCP es-rabbitmq-worldwide:30054 TCP	3 分	⋮
es-postgres-worldwide	app: postgres heritage: Tiller release: es-postgres 10.0.38.44	es-postgres-worldwide:5432 TCP es-postgres-worldwide:30066 TCP	3 分	⋮
es-mongo-worldwide	app: mongo heritage: Tiller release: es-mongo 10.0.44.25	es-mongo-worldwide:27017 TCP es-mongo-worldwide:30063 TCP	3 分	⋮

#### Port for DeviceOn Server Used

Name & Description	Inbound Port
--------------------	--------------

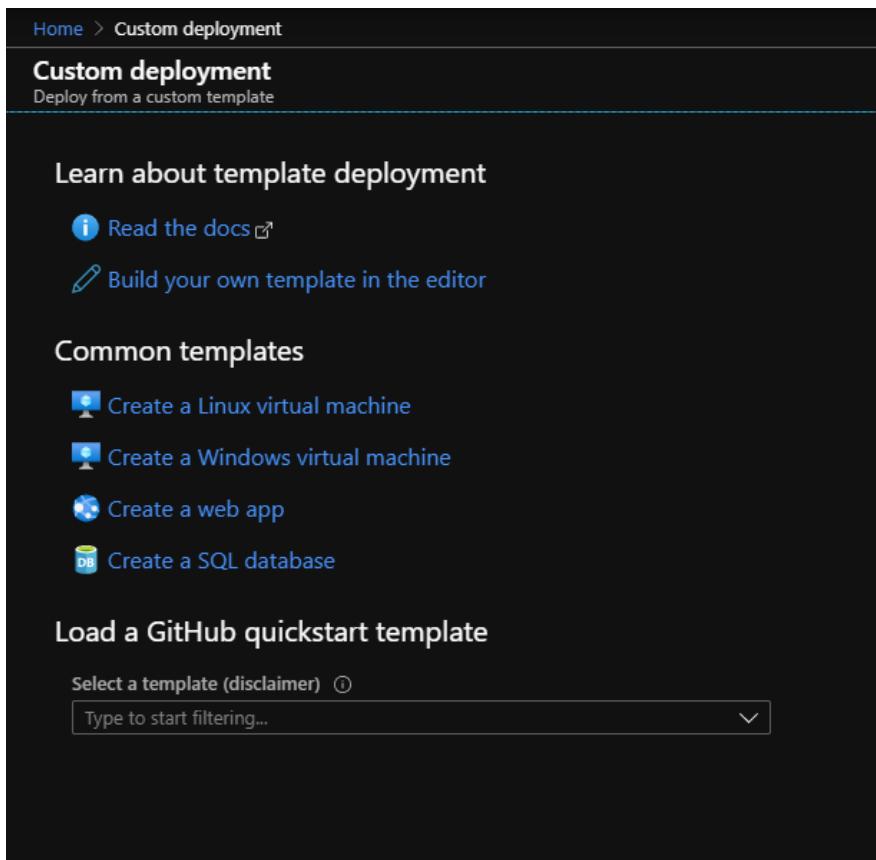
<b>1</b>	Message Broker (RabbitMQ) MQTT, MQTTS	1883, 8883
<b>2</b>	Message Broker (RabbitMQ) AMQP, AMQPs	5671, 5672
<b>3</b>	Message Broker (RabbitMQ) Management Console	15672
<b>4</b>	Database for MongoDB	27017
<b>5</b>	Database for PostgreSQL	5432

#### 4.6.5 Steps to Deploy DeviceOn to AKS by Manual

**Step 1:** Download [ARM template](#) and use notepad to open the file.

**Step 2:** Copy all content from the template

**Step 3:** Login [Azure Portal](#) to start custom deployment



The screenshot shows the 'Custom deployment' section of the Azure portal. It includes links to 'Read the docs' and 'Build your own template in the editor'. Below these are 'Common templates' for creating a Linux virtual machine, Windows virtual machine, web app, or SQL database. At the bottom, there's a search bar for 'Load a GitHub quickstart template' with a placeholder 'Type to start filtering...'.

**Step 4:** Select **Build your own template in the editor** and paste the content from Step 2, then click save button.

Edit template

Edit your Azure Resource Manager template

```
+ Add resource ↑ Quickstart template ⌂ Load file ⌂ Download
Parameters (8)
Variables (22)
Resources (7)
  [concat('edgesense','-aci')](Mic... [concat('edgesense-upgrade','-... [variables('IoTHubName')](Mic... [variables('storageAccountNam... [variables('functionAppName1')... [variables('functionAppName2')... [variables('AKSresourceName')]...
  Parameters (8)
  Variables (22)
  Resources (7)
  apiVersion: "2015-08-01",
  "name": "web",
  "type": "sourcecontrols",
  "dependsOn": [
    "[resourceId('Microsoft.Web/Sites', variables('functionAppName'))]"
  ],
  "properties": {
    "repoUrl": "[variables('repoURL2')]",
    "branch": "[variables('branch')]",
    "isManualIntegration": true
  }
],
{
  "apiVersion": "2018-03-31",
  "type": "Microsoft.ContainerService/managedClusters",
  "location": "[parameters('Location')]",
  "name": "[variables('AKSresourceName')]",
  "properties": {
    "dnsPrefix": "adv",
    "agentPoolProfiles": [
      {
        "name": "agentpool",
        "osDiskSizeGB": "[variables('osDiskSizeGB')]",
        "count": "[variables('agentCount')]",
        "vmSize": "[variables('agentVMsize')]",
        "osType": "[variables('osType')]",
        "storageProfile": "ManagedDisks"
      }
    ],
    "servicePrincipalProfile": {
      "clientId": "[parameters('ApplicationId')]",
      "secret": "[parameters('Password')]"
    }
  }
}
]
]

Save Discard
```

**Step 5:** Please enter these parameters to start deploy, the **Application Id**, **Password (Client Secrets)**, **Tenant Id**, please refer to Section 2.1.2.

Home > Custom deployment

## Custom deployment

Deploy from a custom template

Customized template
7 resources
[Edit template](#)
[Edit paramet...](#)
[Learn more](#)

### TEMPLATE

Customized template  
7 resources

[Edit template](#) [Edit paramet...](#) [Learn more](#)

### BASICS

Subscription *	<input type="text" value="EA87 Demo Test"/>
Resource group *	<input type="text" value="Select a resource group"/> <a href="#">Create new</a>
Location *	<input type="text" value="Asia Pacific Japan East"/>

### SETTINGS

Application Id *	<input type="text"/>
Password *	<input type="text"/>
Tenant Id *	<input type="text"/>
Email *	<input type="text"/>
Location *	<input type="text"/>
IoT Hub Sku *	<input type="text"/>
IoT Hub Unit *	<input type="text"/>
Activate Key *	<input type="text"/>

### TERMS AND CONDITIONS

[Azure Marketplace Terms](#) | [Azure Marketplace](#)

[Purchase](#)

- Email: The email address to get deployment status.
- Location: Please refer below table

Data Center	Location name
Asia East	eastasia
Asia Southeast	southeastasia
Japan East	japaneast
US East	eastus
Europe North	northeurope

- IoT Hub SKU: S1/S2/S3, default is S1
- IoT Hub Unit: 1 to 10, default is 1
- Activate key: Enter **N/A** to skip activate DeviceOn server automatically or please [contact us](#) to purchase license key.

**Step 6:** Pick-up agreement item and click **Purchase** button to start deployment process.

**TERMS AND CONDITIONS**

[Azure Marketplace Terms](#) | [Azure Marketplace](#)

By clicking "Purchase," I (a) agree to the applicable legal terms associated with the offering; (b) authorize Microsoft to charge or bill my current payment method for the fees associated the offering(s), including applicable taxes, with the same billing frequency as my Azure subscription, until I discontinue use of the offering(s); and (c) agree that, if the deployment involves 3rd party offerings, Microsoft may share my contact information and other details of such deployment with the publisher of that offering.

I agree to the terms and conditions stated above

**Purchase**

**Step 7:** After deployment, you will receive a mail to get server information, including account, password and URL.

**ADVANTECH**

Dears,

**Thank you for purchasing WISE-PaaS/DeviceOn**

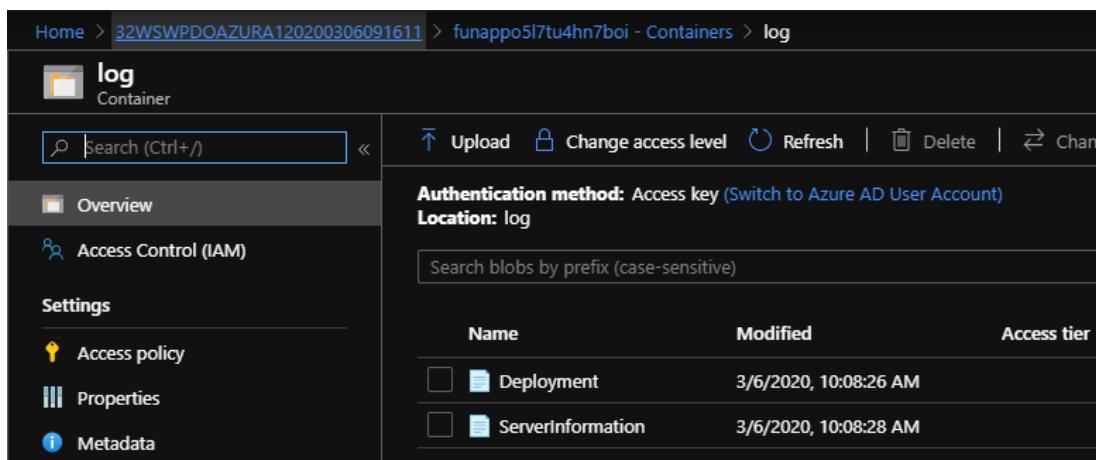
This letter informs you that the service has been deployed on Azure PaaS successfully with following access information.

● DeviceOn Portal: <a href="http://10.70.20.100">http://10.70.20.100</a>	● Grafana Dashboard Portal: <a href="http://10.70.20.170">http://10.70.20.170</a>
Username: <a href="#">admin</a>	Username: <a href="#">admin</a>
Password: <a href="#">7740@www08</a>	Password: <a href="#">7740@www08</a>

Note: Auto-activation apply to first-time purchase and deployment.  
 Product information and additional license purchase: <https://wise-paas.advantech.com/en-us/marketplace/detailinfo/52>

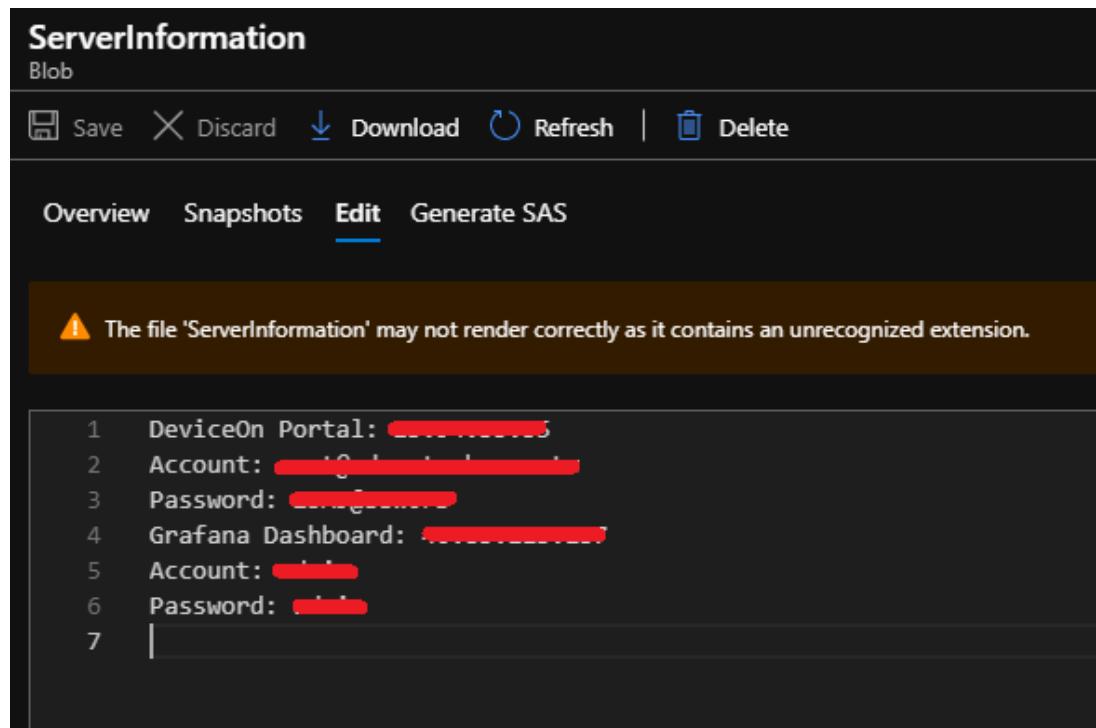
This is an automatically generated email, please do not reply.  
 Best regards,  
 Advantech WISE-PaaS Alliance

To prevent your mail blocked, we write the server information in Azure blob simultaneously.



Name	Modified	Access tier
Deployment	3/6/2020, 10:08:26 AM	
ServerInformation	3/6/2020, 10:08:28 AM	

The credential and access information also on the “ServerInformation”.



```

1 DeviceOn Portal: [REDACTED]
2 Account: [REDACTED]
3 Password: [REDACTED]
4 Grafana Dashboard: [REDACTED]
5 Account: [REDACTED]
6 Password: [REDACTED]
7

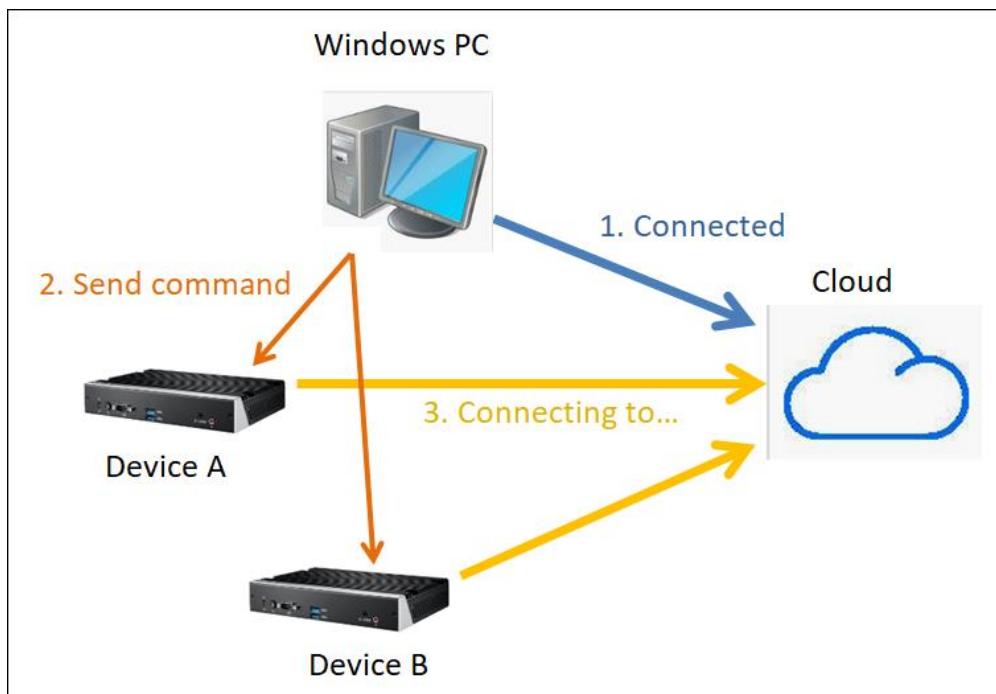
```

## 4.7 How to Batch Provision to Your Devices

WISE-Agent will connect to DeviceOn server through **Credential URL** and **IoT Key** and those setting in **agent\_config.xml**, if you have many devices (that has WISE-Agent in it) need to connect to the server, it takes time to modify agent\_config.xml in each device. Here, we build-in the “**Local Provision**” Plugin to speed up this process. You will learn how to trigger all local devices to connect to the server with the same Credential URL and IoT Key.

The WISE-Agent local provision plugin will send Credential URL and IoT key to other local agent devices,

and the local agent devices can connect to the server successfully. In following figure, you can send trigger command to make device A and B connect to a server with a Windows GUI tool.



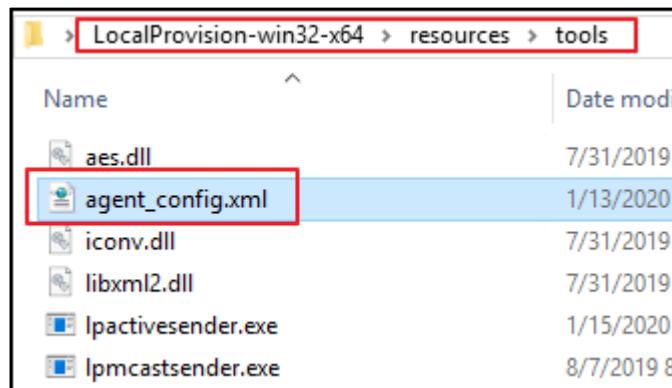
#### 4.7.1 Prerequisite

- All devices must install WISE-Agent in it.
- All devices and the control PC must in the same local network (The multicast packet will not be filtered)
- All devices have the capability to connect to DeviceOn server.

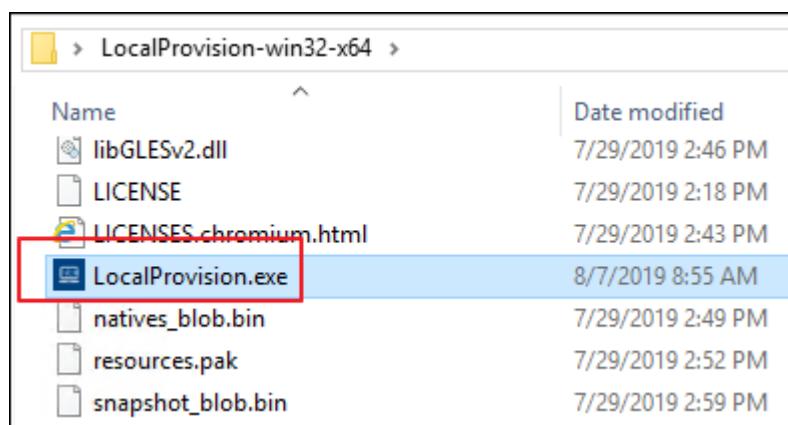
#### 4.7.2 Steps to Local Provisioning

**Step 1:** Download and unzip the [local provision GUI tool](#).

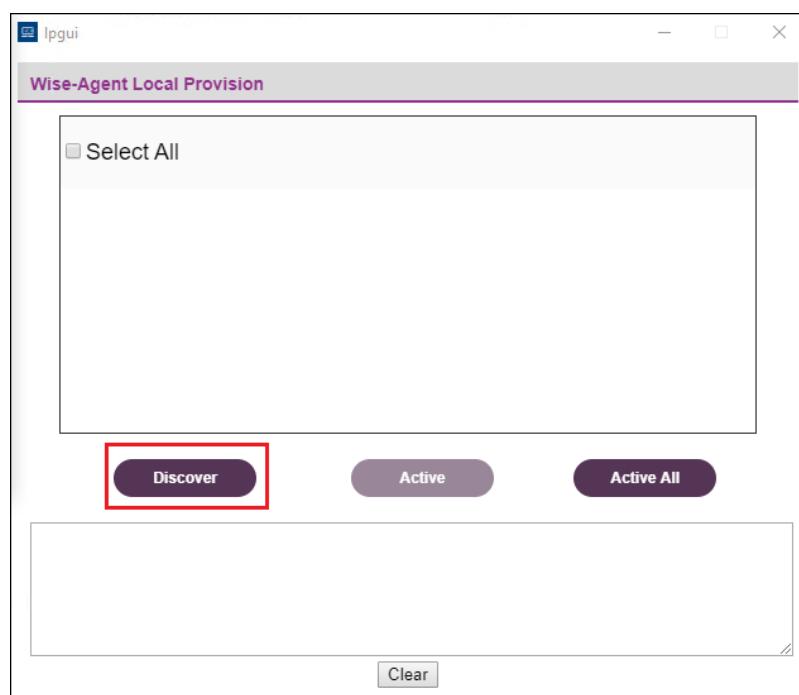
**Step 2:** Place valid “**agnet\_config.xml**” file (with correct Credential URL and IoT Key) to “**GUI tool\resources\tools**” folder



**Step 3: Double click "LocalProvision.exe"**



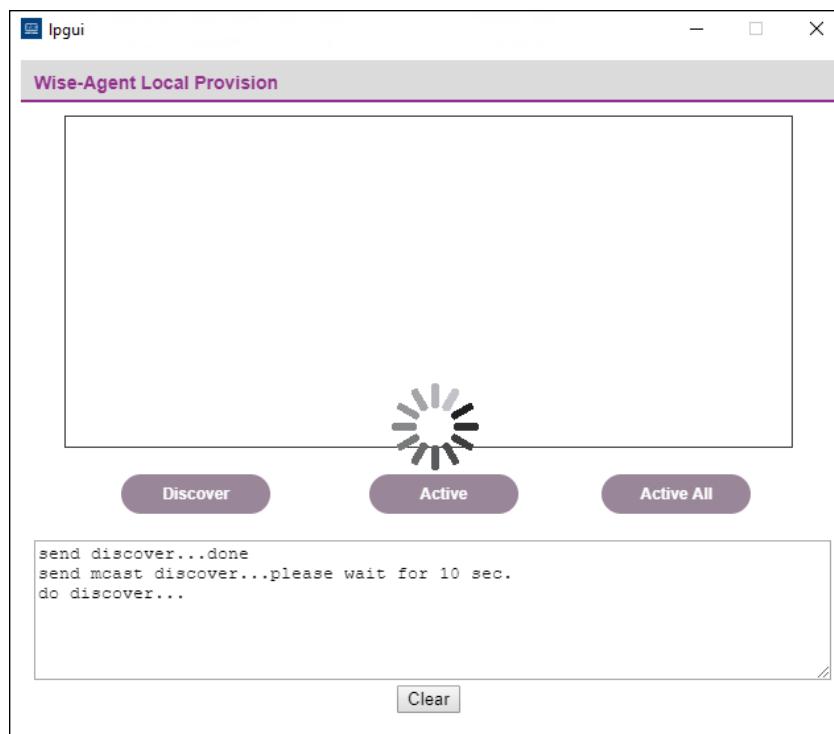
**Step 4: Click Discover button**

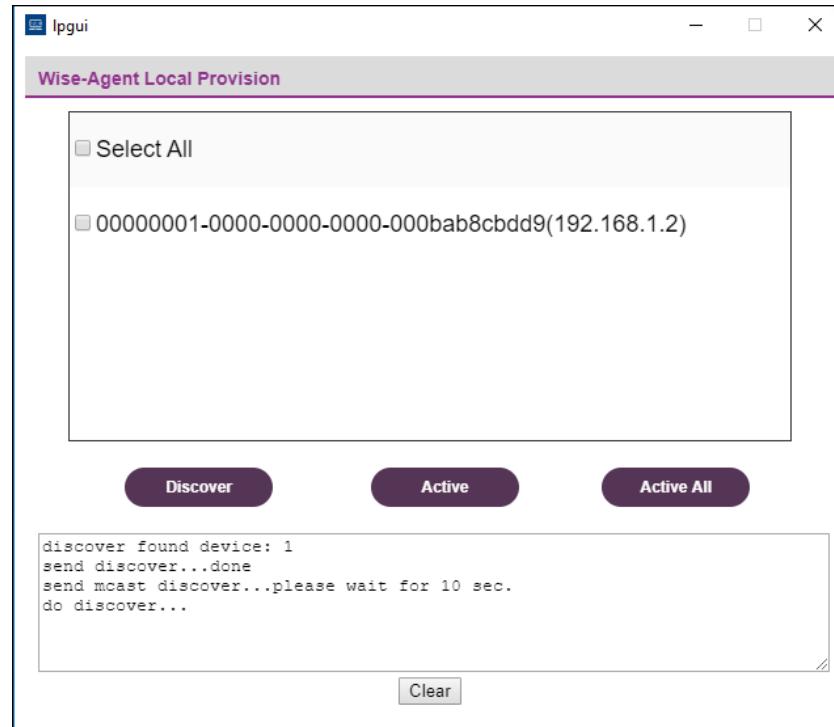


If windows display a firewall dialog, please click allow to enable TCP server permission in tool.

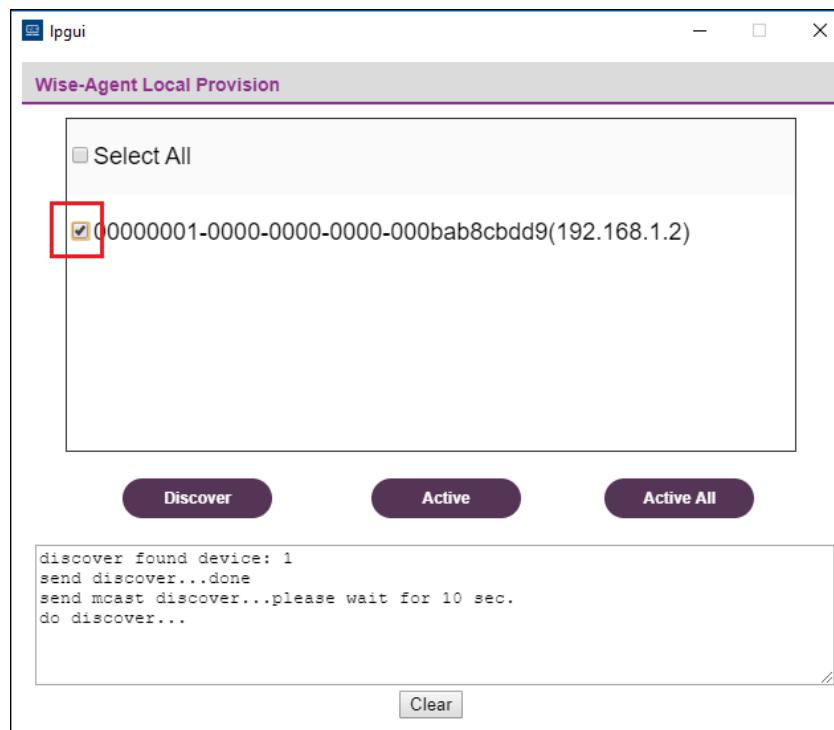


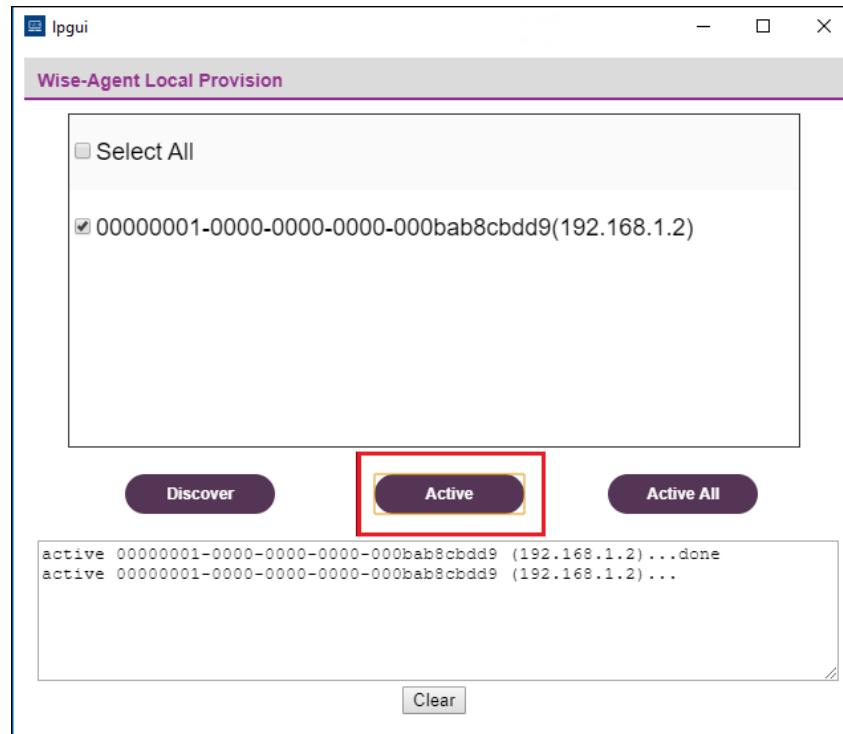
**Step 5:** Wait for 10 second and then you can get the devices on checkbox list.





**Step 6:** Pick-up the device that you would like to connect to the server and click **Active**.





Until now, the checked devices should connect to server after few second later.

#### 4.7.1 Troubleshooting

Why can't I find some WISE-Agent devices? Please help check following:

A. Please check if your local provision plugin is enabled.

Open the **module\_config.xml** in “Installation path\module\” to check if local provision handler is enabled.

```

1  <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
2  <XMLConfigSettings>
3  <BaseSettings>
4    <ModuleNum>17</ModuleNum>
5    <ModuleName1>HDDHandler</ModuleName1>
6    <ModulePath1>\module\HDDHandler.dll</ModulePath1>
7    <ModuleEnable1>TRUE</ModuleEnable1>
8    <ModuleName2>PowerOnOffHandler</ModuleName2>
9    <ModulePath2>\module\PowerOnOffHandler.dll</ModulePath2>

```

```

50  <ModuleName16>EmbIPC</ModuleName16>
51  <ModulePath16>\module\EmbIPC.dll</ModulePath16>
52  <ModuleEnable16>TRUE</ModuleEnable16>
53  <ModuleName17>HDDPMQ</ModuleName17>
54  <ModulePath17>\module\HDDPMQ.dll</ModulePath17>
55  <ModuleEnable17>TRUE</ModuleEnable17>
56  <ModuleName18>LocalProvision</ModuleName18>
57  <ModulePath18>\module\LocalProvisionHandler.dll</ModulePath18>
58  <ModuleEnable18>TRUE</ModuleEnable18>
59
60  </BaseSettings>
</XMLConfigSettings>

```

- B. Please check if your device and windows PC is in the same local network and can transfer multicast packets.
- C. Because the local provision discovers wise-device by UDP port **9178** and TCP port **9177**, please check if your IT block these ports in your local network.

## 5. DeviceOn Development Guide

### 5.1 WISE-Agent Plugin Development

Advantech provides an edge software tool to communicate and exchange information between IoT (Internet of Thing) devices and DeviceOn cloud, called a WISE-Agent. The WISE-Agent not only provides a rich set of users friendly, intelligent, standardization and scalability.

- **Standardization**

The communication protocol is based on the MQTT protocol to communicate and exchange data with DeviceOn cloud. The IoT sensor data report format is following the IPSO Alliance. in JSON format.

- **Portability**

The whole framework is written in C language and follow the ANSI C Standard that C compilers are available for most systems and are often the first compiler provided for a new system, such as OpenWRT, Yacto and Linux based system.

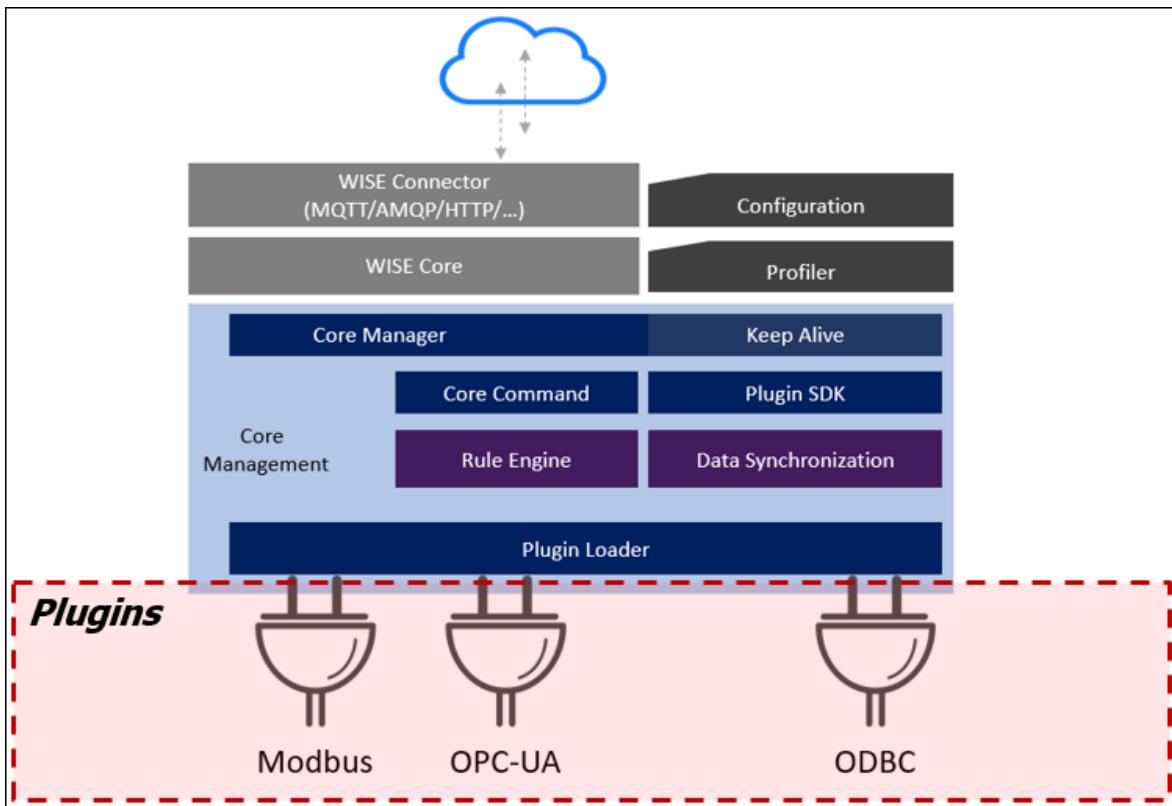
- **Scalability**

The WISE-Agent is modular design and offering plugin concept to Plug & Play (PnP) which is one with a specification that facilitates the discovery of a Plugin in a system without the need for a physical device to advanced configuration or user intervention in resolving resource conflicts.

Besides the basic device connectivity, the WISE-Agent provides an advanced heartbeat solution to synchronize device status. On the different network environment, how to keep your device data without loss? The WISE-Agent has built-in “**Data Synchronization**” to avoid and overcome the disconnect for a long time. For various protocols, we offer a plugin SDK, users only focus on how to retrieve the data, do not worry about the connectivity and stability.

### 5.1.1 WISE-Agent Architecture

WISE-Agent includes two parts, one is the **Core Framework** and **Plugins**.



- **Core Framework**

The main library used to communicate with WISE-PaaS IoTHub or standard MQTT broker and include below components.

- ✧ **Platform Profiler:** describes the target platform (e.g., OS version, SN, Device name, MAC address)
- ✧ **Configuration:** describes how to connect to MQTT broker (e.g., Credential URL, IoTKey, TLS/SSL settings)
- ✧ **Core Manager:** integrates and manages the resources and keeps them alive.
- ✧ **Core Command:** responsible for handling commands that interact with internal components (e.g., rename, update, get capability, auto report start/stop)
- ✧ **Plugin SDK:** A plugin framework that makes plugin implement more easily.

- ✧ **Keep Alive:** A component to detect the connection between WISE-Agent and DeviceOn Server.
- ✧ **Data Synchronization:** kernel plugin that caches and restores data to ensure zero downtime.
- ✧ **Rule Engine:** kernel plugin that supports the threshold rule check and then sends event or trigger actions
- ✧ **Plugin Loader:** responsible for loading and managing plugins indicated in module\_config.xml

## ● The plugins

The plugins include IPC monitoring (Advantech Hardware, HDD/SSD, Networks, Process...etc.), control function (Backup/Recovery, Protection, Remote Desktop, Terminal...), and sensor protocol collection. Following are the list of supported plugins in WISE-Agent.

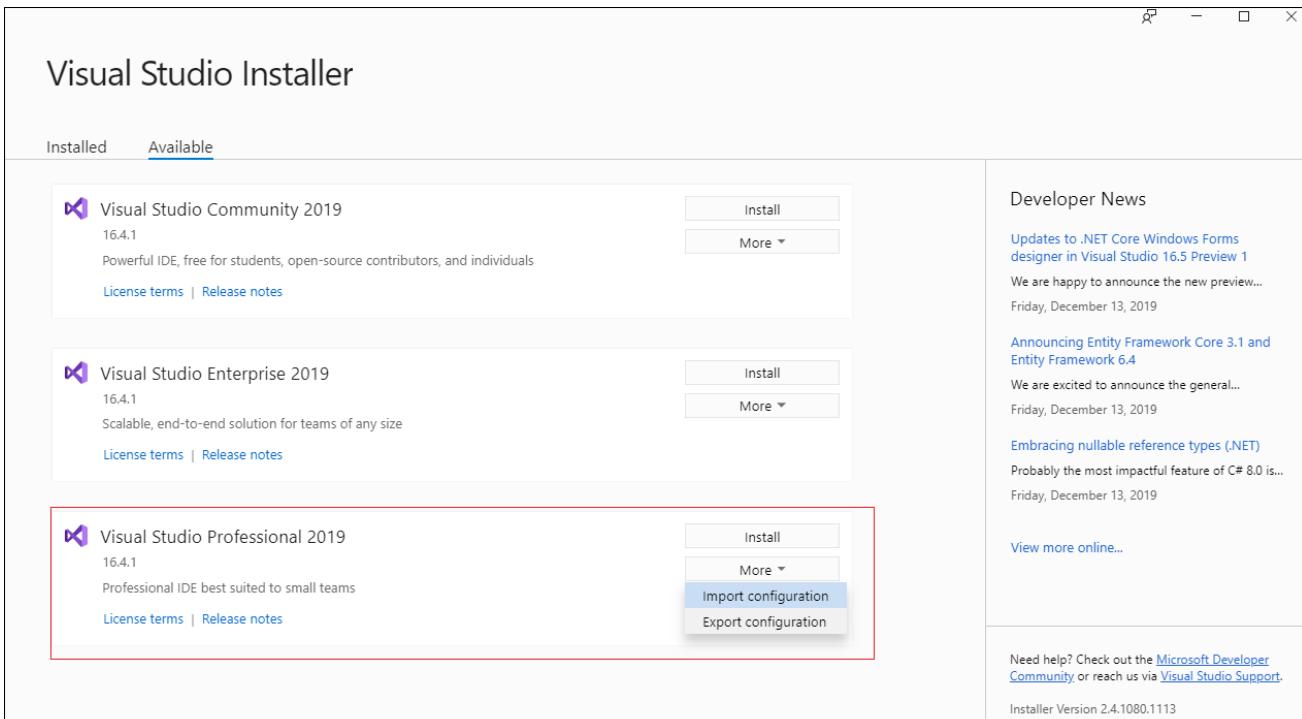
- ✧ **SUSI Control:** Monitoring and Control Advantech Hardware Platform
- ✧ **HDD Monitoring:** Monitoring Hard Drives (HDD, SSD) Usage, Healthy and S.M.A.R.T Information, especially for Advantech SQFlash.
- ✧ **Network Monitoring:** Monitoring Network Interface Usage, Throughput...
- ✧ **Process Monitoring:** Monitoring System Process Status, CPU, Memory Usage.
- ✧ **Power Management:** Remote Control Power On, Off, Reboot, Sleep, Hibernate.
- ✧ **Backup/Recovery:** Remote Backup/Recovery System via Acronis
- ✧ **Protection:** Remote System Protection via McAfee
- ✧ **Remote Desktop:** Remote Desktop via VNC Viewer
- ✧ **Remote Terminal:** Remote Terminal Command
- ✧ **Remote Screenshot:** Remote Screenshot on Current Screen
- ✧ **OTA (Over-the-Air):** Remote Software, Firmware Update
- ✧ **System Program Monitoring:** System Program Information
- ✧ **Embedded Control:** Advanced Control (UWF, USB Lock, Keyboard Filter, ...etc.) for Windows 10 Embedded, LTSC, LTSB
- ✧ **HDD Prediction:** Build-in Hard Drives (HDD, SSD) Failure Prediction Model
- ✧ **Modbus:** Modbus Device Data Gathering
- ✧ **Service Plugin:** Bridge Southbound Device Service

### 5.1.2 Prerequisite

- Visual Studio 2019 for Windows Plugin
- Android NDK for Android Plugin
- A WISE-Agent that is running on your system.

### 5.1.3 Develop a Plugin on Windows Environment

**Step 1:** You can configure Visual Studio across your organization with installation configuration files, [.vsconfig](#)



Visual Studio Installer

Installed Available

**Visual Studio Community 2019**  
16.4.1  
Powerful IDE, free for students, open-source contributors, and individuals  
[License terms](#) | [Release notes](#)

**Visual Studio Enterprise 2019**  
16.4.1  
Scalable, end-to-end solution for teams of any size  
[License terms](#) | [Release notes](#)

**Visual Studio Professional 2019**  
16.4.1  
Professional IDE best suited to small teams  
[License terms](#) | [Release notes](#)

Developer News

Updates to .NET Core Windows Forms designer in Visual Studio 16.5 Preview 1  
We are happy to announce the new preview...  
Friday, December 13, 2019

Announcing Entity Framework Core 3.1 and Entity Framework 6.4  
We are excited to announce the general...  
Friday, December 13, 2019

Embracing nullable reference types (NET)  
Probably the most impactful feature of C# 8.0 is...  
Friday, December 13, 2019

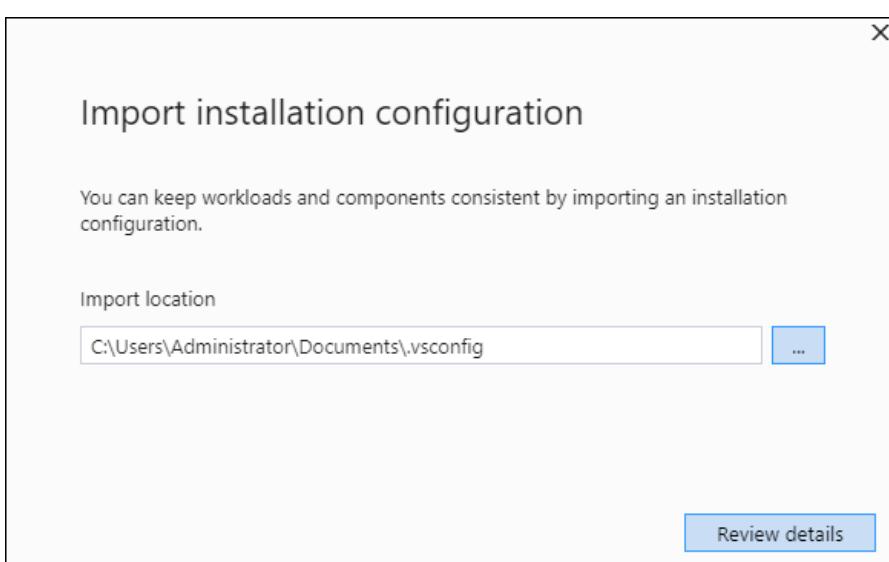
[View more online...](#)

Need help? Check out the [Microsoft Developer Community](#) or reach us via [Visual Studio Support](#).  
Installer Version 2.4.1080.1113

**Visual Studio Professional 2019**  
16.4.1  
Professional IDE best suited to small teams  
[License terms](#) | [Release notes](#)

Install  
More ▾  
**Import configuration**  
Export configuration

Install  
More ▾  
**Import configuration**  
Export configuration



Import installation configuration

You can keep workloads and components consistent by importing an installation configuration.

Import location

C:\Users\Administrator\Documents\vsconfig

[Review details](#)

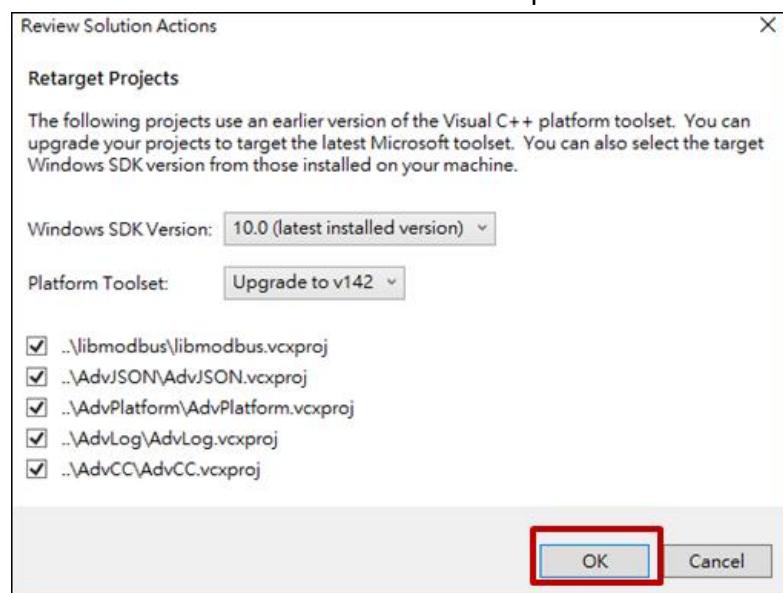
**Step 2:** Download SRP-Plugin,

git clone <http://advgitlab.eastasia.cloudapp.azure.com/SPR-Connect/SPR-Plugin.git>

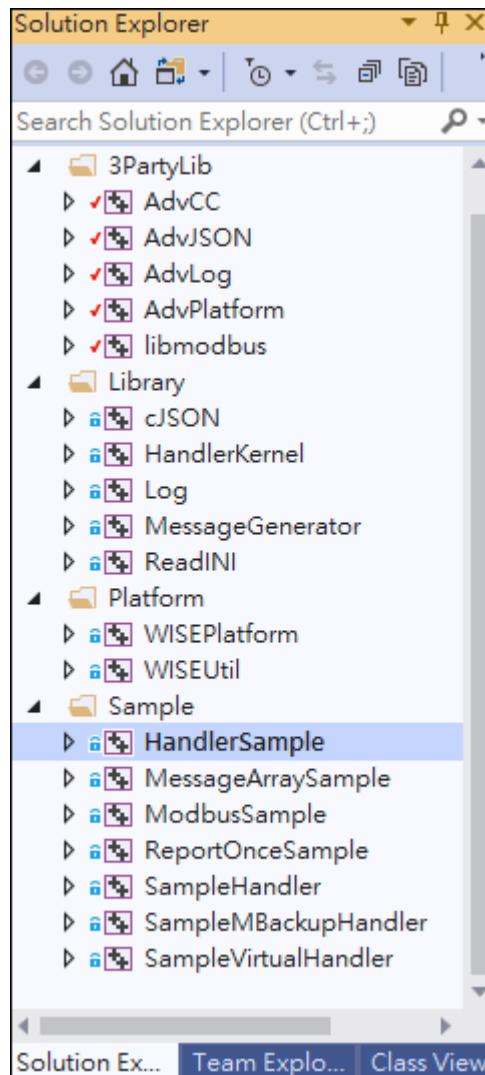
**Step 3:** Open SRP-Plugin solution file, **SRP-Plugin-V2015.sln**

名稱	修改日期
depcomp	2019/11/18 上午 ...
getv.sh	2019/12/11 上午 ...
install-sh	2019/11/18 上午 ...
libtool	2019/11/19 下午 ...
license	2019/3/22 上午 0...
ltmain.sh	2019/11/18 上午 ...
Makefile	2019/11/19 下午 ...
Makefile.am	2019/11/19 下午 ...
Makefile.in	2019/11/19 下午 ...
missing	2019/11/18 上午 ...
pre-install_centos.sh	2019/3/22 上午 0...
pre-install_ubuntu.sh	2019/11/19 下午 ...
README.md	2019/3/22 上午 0...
release-helper.sh	2019/3/22 上午 0...
run.sh	2019/11/18 下午 ...
SRP-Plugin.sln	2019/12/11 上午 ...
SRP-Plugin_VS2015.sln	2019/12/11 上午 ...
stamp-h1	2019/11/19 下午 ...

**Step 4:** Click OK to update the SDK and Toolset for current compile environment



**Step 5:** You can implement new plugin base on plugin sample project.



**Step 6:** It is more easily to create a new plugin by Web-Simulator tools. Web-Simulator is an auxiliary tool that helps you quickly simulate data on the cloud via MQTT over WebSocket (network port: 15675) and directly generate the corresponding code. Following step will introduce how to create a new plugin by Web-Simulator tools. If you want to know exactly how this tool is used, you can refer Web-Simulator [QuickStart](#).

**Step 7:** Download [Web-Simulator](#) tools.

**Step 8:** The sample code can be generated in the fourth step. Please save it as **handler\_data.c** and replace it in the “**SRP-Plugin\Sample\HandlerSample**” path.



Step 3 - Create Grafana board

Now you can open the [RMM Portal](#) to check the retime value or go to the next step, automatically generate a grafana board for you.

Or you can reference the [Sample Handler](#) sample code , according to previous step.

**Grafana url:**

**Name:**

**Password:**

[Previous](#) [Next](#)

**Step 9:** Right click the “**HandlerSampe**” project in Step 5 and choose “**Solution**”.

**Step 10:** Check output without error message. If appear error message, suggest to copy the error message search in google or ask Advantech technical people.

**Step 11:** After successfully completing the compilation, you can find all the .dll files in below path  
**“SRP-Plugin\Debug\module”**

**Step 12:** Download and install [WISE-Agent](#) for Windows. The default installation path is  
**C:\Program Files (x86)\Advantech\WISE-Agent**

**Step 13:** After install the WISE-Agent, copy “**HandlerSample.dll**” file to  
**“C:\Program Files (x86)\Advantech\WISE-Agent\module”** folder.

**Step 14:** Modify **module\_config.xml** on

**“C:\Program Files (x86)\Advantech\WISE-Agent\module\module\_config.xml”**

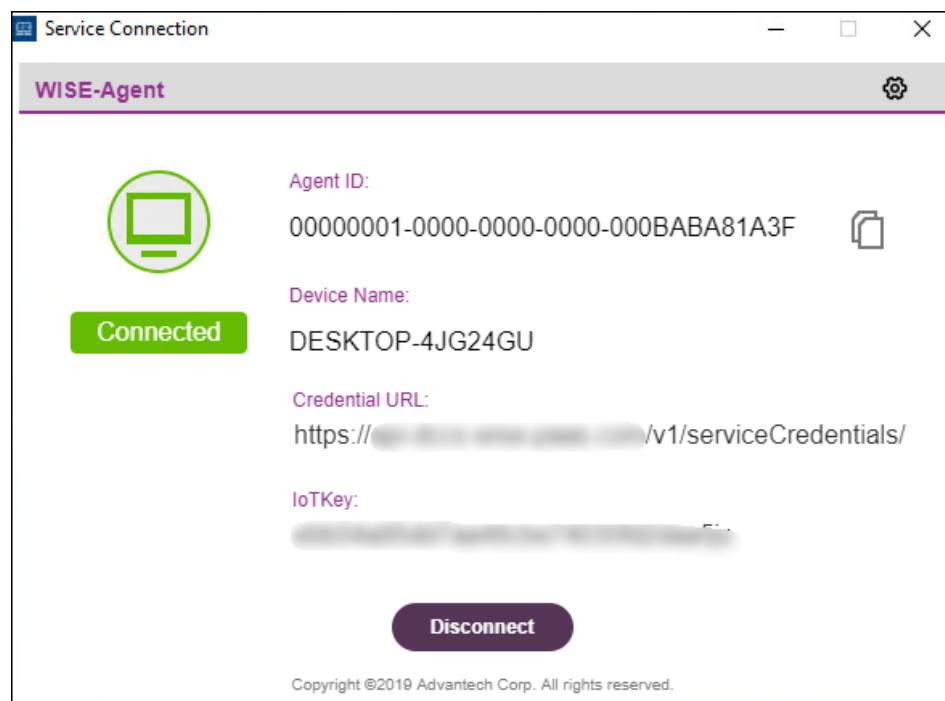
- Increase **ModuleNum** value in below line 3
- Add **HandlerSample.dll** item in below line 7.

```

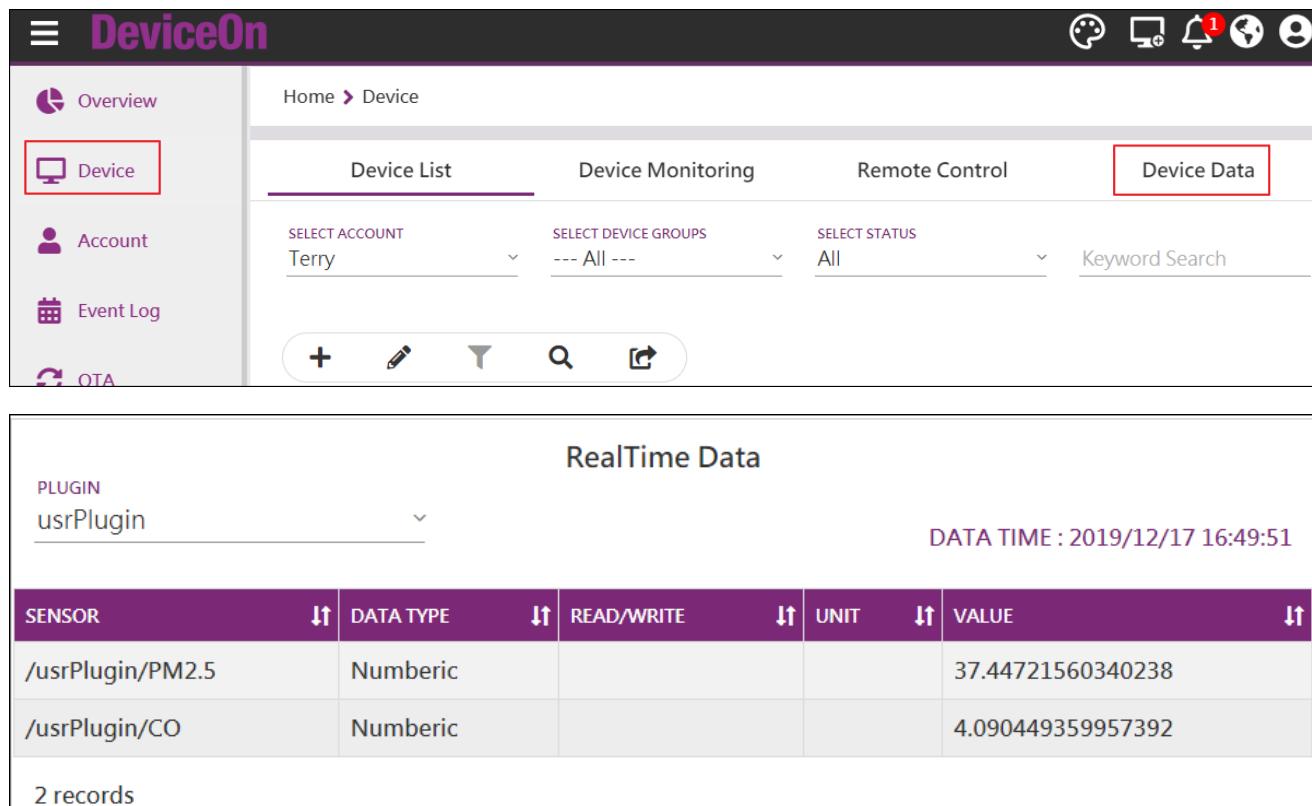
01.  <?xml version="1.0"?>
02.  <XMLConfigSettings><BaseSettings>
03.  <ModuleNum>15</ModuleNum>
04.  <ModuleName1>HDDHandler</ModuleName1><ModulePath1>module/HDDHandler.so</ModulePath1><ModuleEnable1>TRUE</ModuleEnable1>
05.  ...
06.  <ModuleName14>ServiceHandler</ModuleName14><ModulePath14>module/ServiceHandler.so</ModulePath14><ModuleEnable14>TRUE</ModuleEnable14>
07.  <ModuleName15>HandlerSample</ModuleName15><ModulePath15>module/HandlerSample.so</ModulePath15><ModuleEnable15>TRUE</ModuleEnable15>
08.  </BaseSettings>
09.  </XMLConfigSettings>

```

**Step 15:** Reconnect WISE-Agent by “Server Connection” tools. Press “Disconnect” then “Connect”.



**Step 16:** Check if your plugin appears in DeviceOn Page, (**Device** -> **Device Data** -> **PLUGIN**)



The screenshot shows the "DeviceOn" interface. The left sidebar has icons for Overview, Device (which is highlighted with a red border), Account, Event Log, and OTA. The main area shows a navigation path "Home > Device". Below this are tabs for "Device List", "Device Monitoring", "Remote Control", and "Device Data" (which is also highlighted with a red border). There are dropdown menus for "SELECT ACCOUNT" (set to "Terry"), "SELECT DEVICE GROUPS" (set to "--- All ---"), and "SELECT STATUS" (set to "All"). Below these are search and filter icons. The main content area is titled "RealTime Data" and shows a table of data. The table has columns: SENSOR, DATA TYPE, READ/WRITE, UNIT, and VALUE. Two records are listed: "/usrPlugin/PM2.5" with a numeric value of "37.44721560340238" and "/usrPlugin/CO" with a numeric value of "4.090449359957392". The table footer says "2 records". The top right of the main area shows the date and time "DATA TIME : 2019/12/17 16:49:51".

SENSOR	DATA TYPE	READ/WRITE	UNIT	VALUE
/usrPlugin/PM2.5	Numeric			37.44721560340238
/usrPlugin/CO	Numeric			4.090449359957392

#### 5.1.4 Develop a Plugin on Linux Environment

The following steps are handled in Ubuntu or Debian system. If your target device is Yocto Linux, you have to set up cross-compile environment on your host PC. The example below shows how to set up for NXP i.MX8 projects.

```
$ /opt/fsl-imx-xwayland/4.14-sumo
$ source environment-setup-aarch64-poky-linux
```

From now, the DeviceOn supports the following RISC platform, please refer to the SDK links relating to the platform you are developing for details.

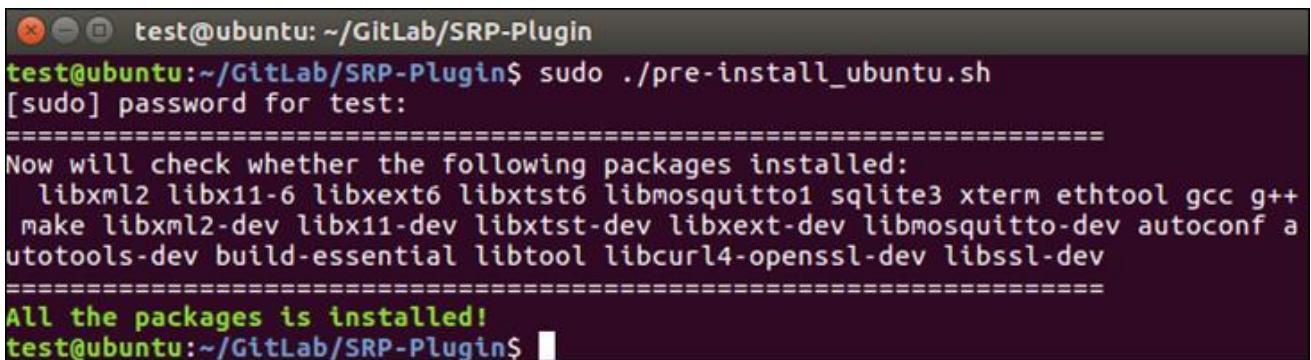
Platform	OS	Architecture	SDK
NXP i.MX8	Yocto 2.5.2	aarch64	<a href="#">Link</a>
NXP i.MX6	Yocto 2.1.1	armv7-a	<a href="#">Link</a>
Qualcomm APQ8016	Yocto 2.1.3	aarch64	<a href="#">Link</a>
TI AM335x	Yocto 2.4	armv7-a	<a href="#">Link</a>
RK3288	Debian 9.8	arm	N/A
RK3399	Debian 9.9	aarch64	N/A

[http://ess-wiki.advantech.com.tw/view/AIMLinux/AddOn/DeviceOn#Supported\\_Platforms](http://ess-wiki.advantech.com.tw/view/AIMLinux/AddOn/DeviceOn#Supported_Platforms)

**Step 1:** Download SRP-Plugin as Section 5.1.3 Step 2.

**Step 2:** In Plugin SDK (SRP-Plugin) folder, execute ‘`sudo ./pre-install_ubuntu.sh`’ with root user authority to install compile tools and dependency libraries

**Note:** *If you are developing with cross-compile, you can skip this step.*



```
test@ubuntu:~/GitLab/SRP-Plugin
test@ubuntu:~/GitLab/SRP-Plugin$ sudo ./pre-install_ubuntu.sh
[sudo] password for test:
=====
Now will check whether the following packages installed:
  libxml2 libx11-6 libxext6 libxtst6 libmosquitto1 sqlite3 xterm ethtool gcc g++
  make libxml2-dev libx11-dev libxtst-dev libxext-dev libmosquitto-dev autoconf autotools-dev build-essential libtool libcurl4-openssl-dev libssl-dev
=====
All the packages is installed!
test@ubuntu:~/GitLab/SRP-Plugin$
```

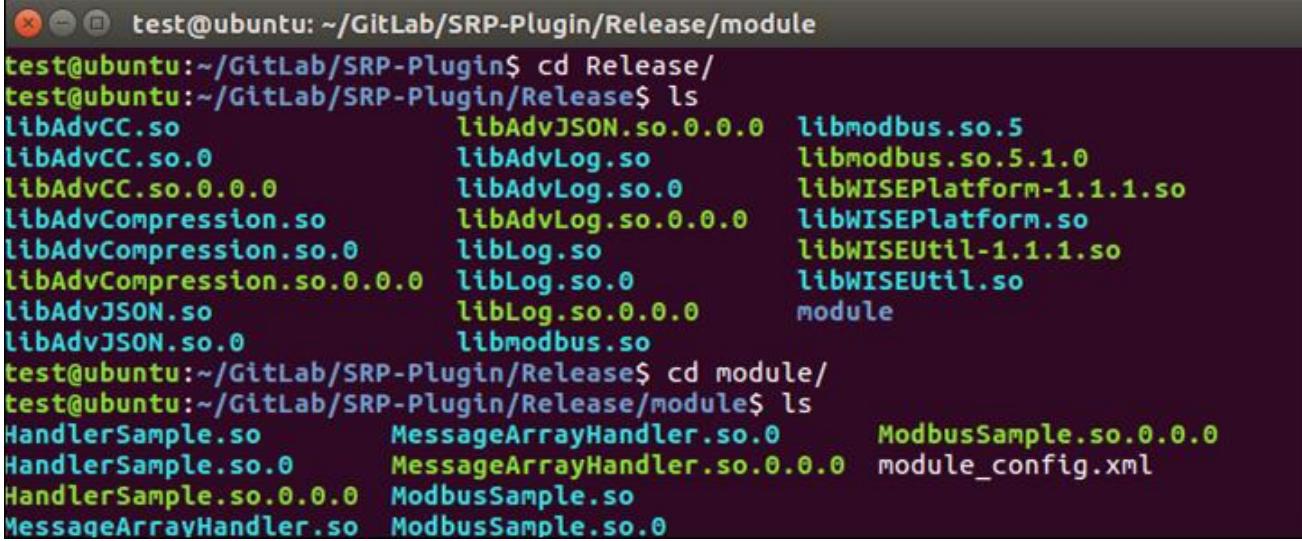
**Step 3:** You can implement new plugin base on plugin sample project or Web-Simulator in Section 5.1.3 Step 6 to Step 8.

**Step 4:** Copy `handler_data.c` that generated by Web-Simulator to  
 “SRP-Plugin/Sample/HandlerSample”.

### Step 5: Build SRP-Plugin by “build-srpplugin.sh”

```
$ ./build-srpplugin.sh
```

### Step 6: You can find the release build file in “SRP-Plugin/Release/module” folder.



```
test@ubuntu:~/GitLab/SRP-Plugin/Release/module
test@ubuntu:~/GitLab/SRP-Plugin$ cd Release/
test@ubuntu:~/GitLab/SRP-Plugin/Release$ ls
libAdvCC.so          libAdvJSON.so.0.0.0  libmodbus.so.5
libAdvCC.so.0          libAdvLog.so      libmodbus.so.5.1.0
libAdvCC.so.0.0.0      libAdvLog.so.0    libWISEPlatform-1.1.1.so
libAdvCompression.so   libAdvLog.so.0.0.0 libWISEPlatform.so
LibAdvCompression.so.0 libLog.so        libWISEUtil-1.1.1.so
libAdvCompression.so.0.0.0 libLog.so.0    libWISEUtil.so
libAdvJSON.so         libLog.so.0.0.0   module
libAdvJSON.so.0        libmodbus.so

test@ubuntu:~/GitLab/SRP-Plugin/Release$ cd module/
test@ubuntu:~/GitLab/SRP-Plugin/Release/module$ ls
HandlerSample.so      MessageArrayHandler.so.0      ModbusSample.so.0.0.0
HandlerSample.so.0     MessageArrayHandler.so.0.0.0  module_config.xml
HandlerSample.so.0.0.0 ModbusSample.so
MessageArrayHandler.so ModbusSample.so.0
```

### Step 7: Please download [WISE-Agent](#) for Ubuntu 16.04 x64. The default installation path is /usr/local/AgentService.

### Step 8: After install the WISE-Agent, copy “HandlerSample.so.0.0.0” and “HandlerSample.so” files to “/usr/local/AgentService/module/” folder.

```
$ sudo cp -a Release/module/HandlerSample.so* /usr/local/AgentService/module/
```

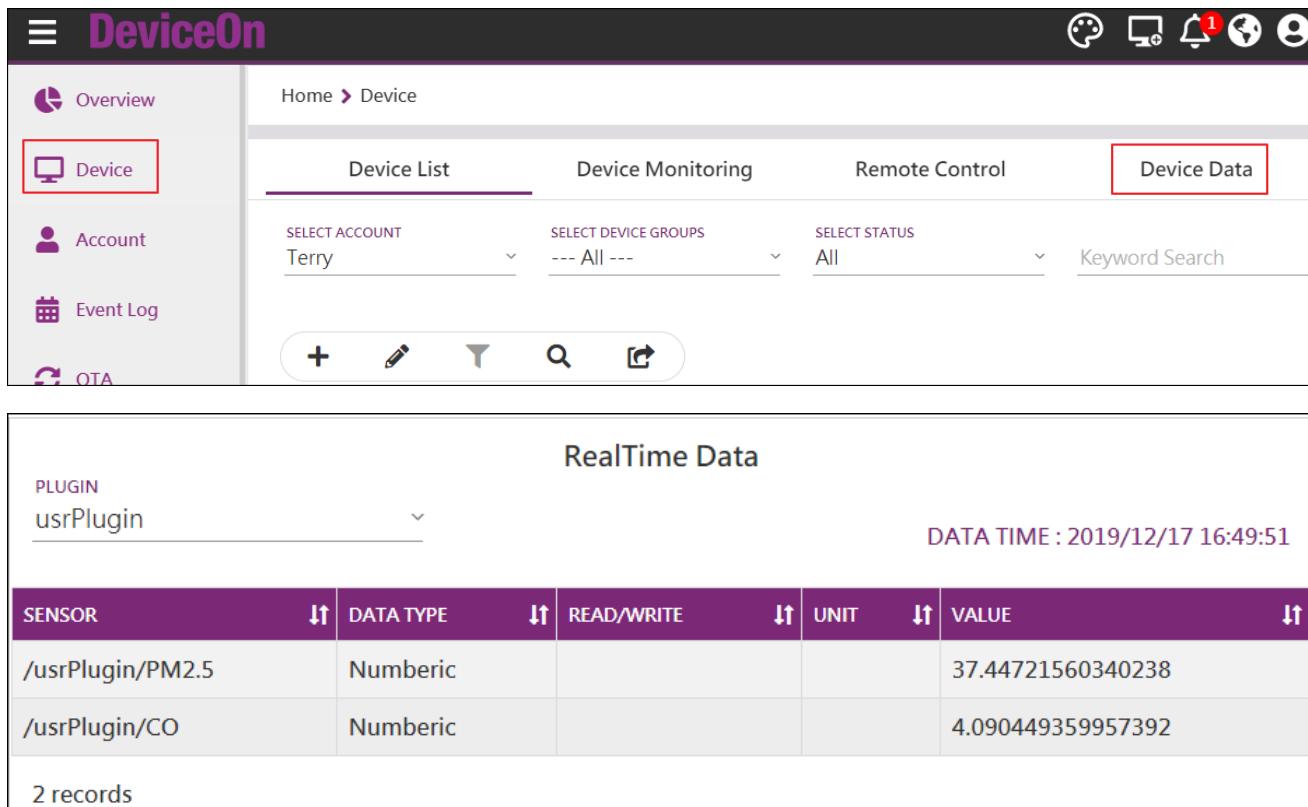
### Step 9: Modify “/usr/local/AgentService/module/module\_config.xml” Increase **ModuleNum** value in below line 3. Add HandlerSample.so item in below line 7.

```
01.  <?xml version="1.0"?>
02.  <XMLConfigSettings><BaseSettings>
03.  <ModuleNum>15</ModuleNum>
04.  <ModuleName1>HDDHandler</ModuleName1><ModulePath1>module/HDDHandler.so</ModulePath1><ModuleEnable1>TRUE</ModuleEnable1>
05.  ...
06.  <ModuleName14>ServiceHandler</ModuleName14><ModulePath14>module/ServiceHandler.so</ModulePath14><ModuleEnable14>TRUE</ModuleEnable14>
07.  <ModuleName15>HandlerSample</ModuleName15><ModulePath15>module/HandlerSample.so</ModulePath15><ModuleEnable15>TRUE</ModuleEnable15>
08.  </BaseSettings>
09.  </XMLConfigSettings>
```

### Step 10: Restart WISE-Agent

```
$ sudo systemctl restart saagent
```

**Step 11:** Check if your plugin appears in DeviceOn Page, (**Device** -> **Device Data** -> **PLUGIN**)



The screenshot shows the WISE-PaaS DeviceOn interface. On the left, there's a sidebar with icons for Overview, Device (which is selected and highlighted with a red border), Account, Event Log, and OTA. The main area has tabs for Device List, Device Monitoring, Remote Control, and Device Data (also highlighted with a red border). Below these tabs are dropdown menus for SELECT ACCOUNT (Terry), SELECT DEVICE GROUPS (--- All ---), and SELECT STATUS (All), along with a Keyword Search input field. At the bottom of this section are several action buttons: a plus sign, a pencil, a funnel, a magnifying glass, and a refresh arrow. The main content area is titled "RealTime Data". It shows a table with two records. The table has columns: SENSOR, DATA TYPE, READ/WRITE, UNIT, and VALUE. The first record is "/usrPlugin/PM2.5" with "Numeric" type, "Read/Write" status, and value "37.44721560340238". The second record is "/usrPlugin/CO" with "Numeric" type, "Read/Write" status, and value "4.090449359957392". Above the table, it says "DATA TIME : 2019/12/17 16:49:51".

SENSOR	DATA TYPE	READ/WRITE	UNIT	VALUE
/usrPlugin/PM2.5	Numeric			37.44721560340238
/usrPlugin/CO	Numeric			4.090449359957392

2 records

### 5.1.5 Develop a Plugin on Android Environment

**Step 1:** Download SRP-Plugin as Section 5.1.3 Step 2.

**Step 2:** In Plugin SDK (SRP-Plugin) folder, execute **android\_build.sh**, where

- CMD : -b : build, -c : clean
- APP\_ABI : x86, armeabi-v7a and so on

For example, to build an armeabi-v7a arch plugin you may enter:

```
source android_build.sh -b armeabi-v7a &
```

**Step 3:** You can find the release build file in

"SRP-Plugin/~/obj/local/armeabi-v7a/libHandlerSample.so" folder.

**Step 4:** Copy **libHandlerSample.so** to DUT in the **/system/lib/module/**.

**Step 5:** Modify **module\_config.xml** in DUT as 5.1.3 step 14 described.

**Step 6:** Check handler as 5.1.3 step 16 described.

## 5.2 DeviceOn UI Plugin Development

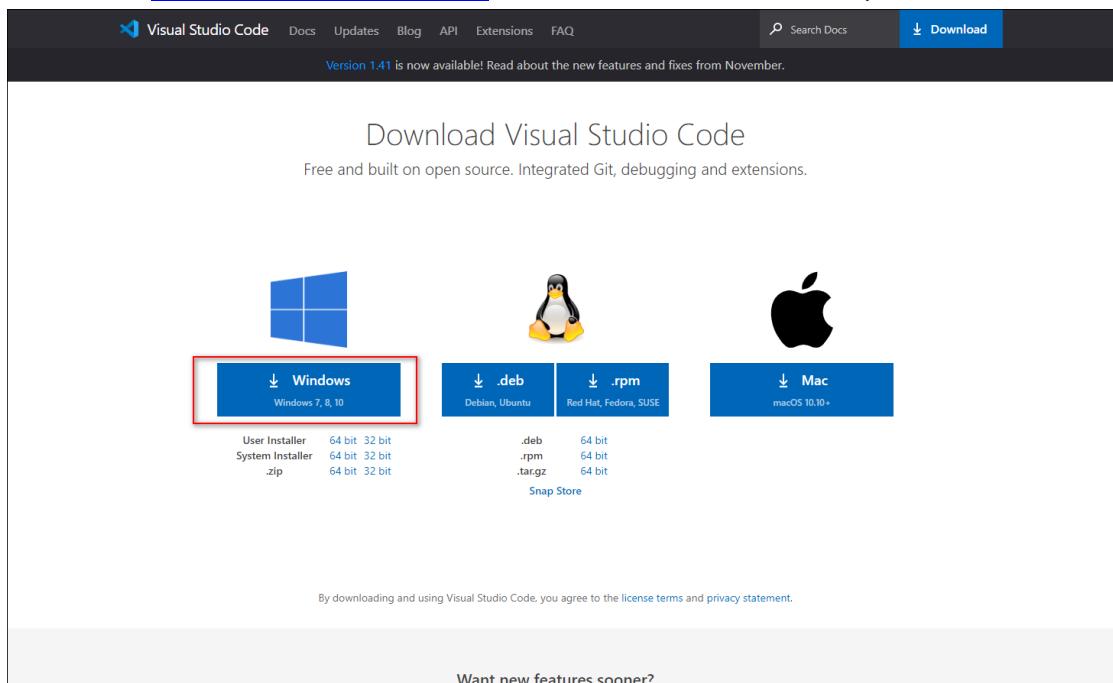
Actually, DeviceOn provide plenty of features to remote management, control to your edge devices, but it's hard to meet all domains application, such as, medical, traffic, energy system and etc. Fortunately, DeviceOn provide APIs and Addins (web user interface) for users to develop their own solution.

### 5.2.1 Prerequisite

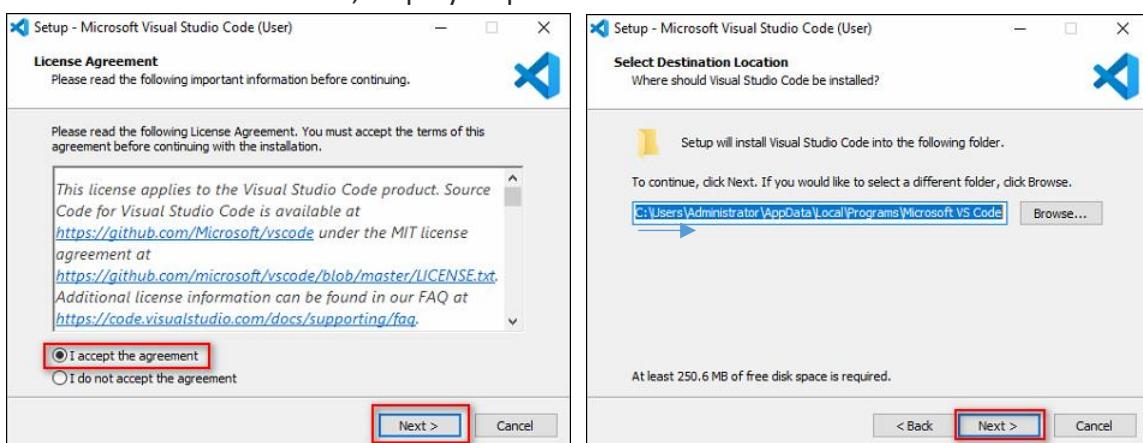
- Visual Studio Code V 1.4.1
- DeviceOn Server

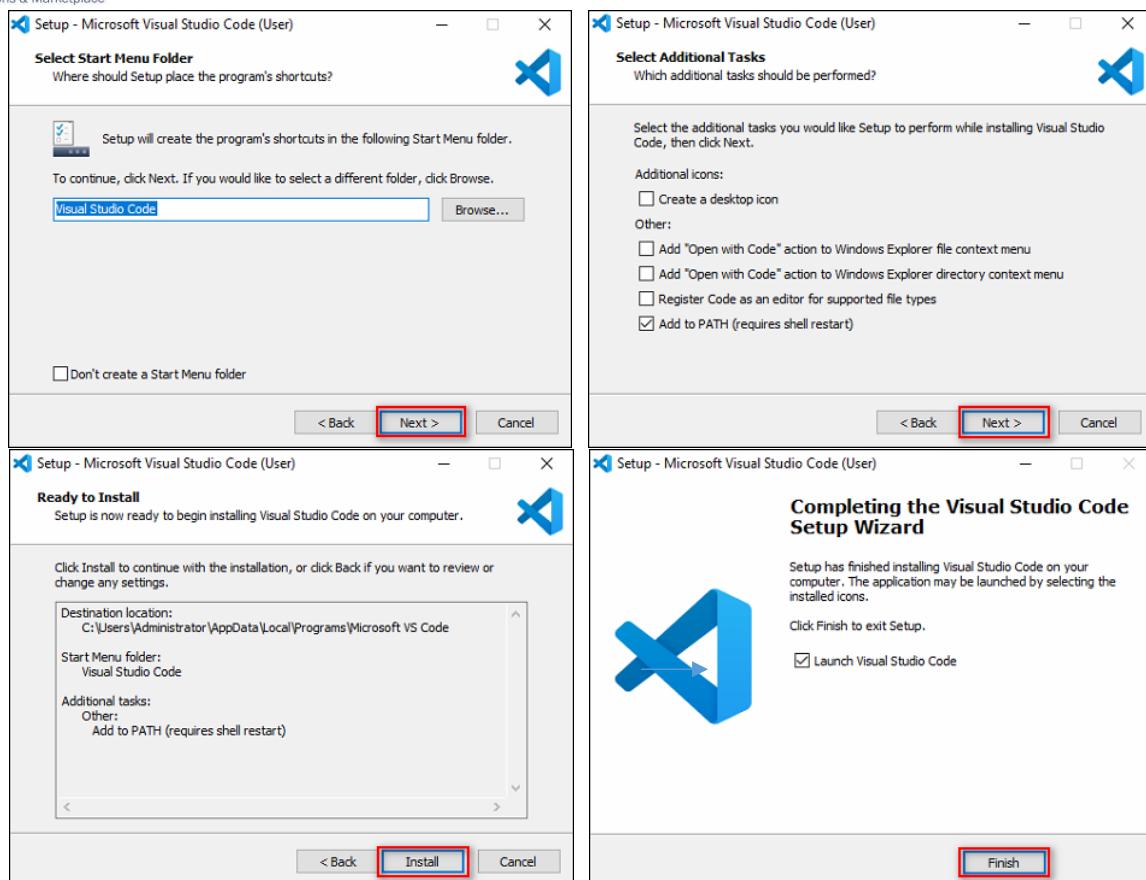
### 5.2.2 Environment Setup

**Step 1:** Download [Visual Studio Code v-1.4.1](#) and launch VSCodeUserSetup-x64-1.41.1.exe.



**Step 2:** Install Visual Studio Code, step by step.





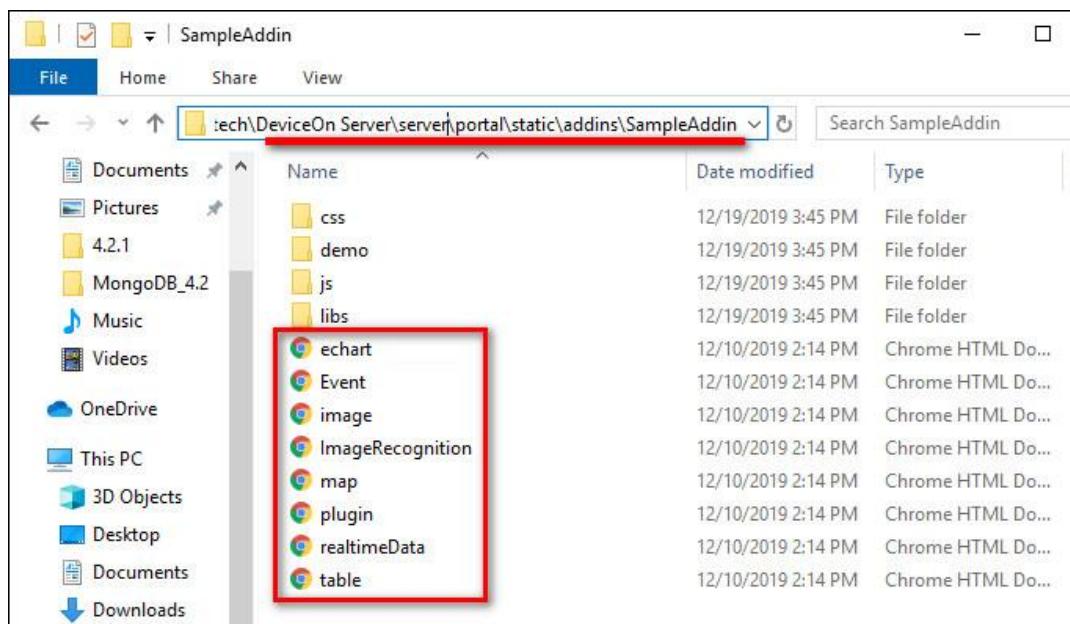
**Step 3:** Install DeviceOn Server, if you don't install DeviceOn Server before, please reference Section 2.2.

### 5.2.3 Develop a Sample Add-in

**Step 1:** Open DeviceOn Server folder and go to the installation path:

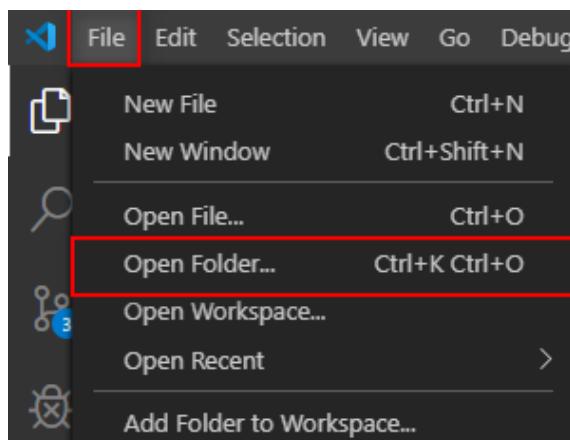
\DeviceOn Server\server\portal\static\addins\SampleAddin.

Here are several Add-in examples (\*.html) that we provide, for your reference.

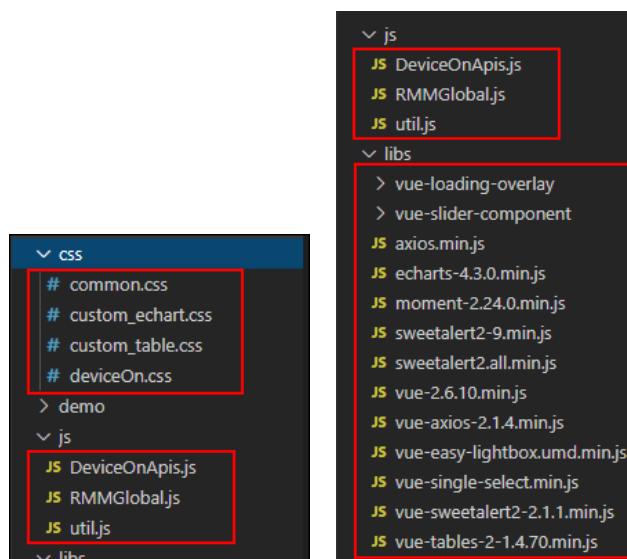


**Step 2:** Open Visual Studio Code -> Open the path:

\DeviceOn Server\server\portal\static\addons\SampleAddin\



**Step 3:** Here are serval resources for you to develop your function.



- **CSS folder** that include \*.css style to describes how HTML elements are to be displayed on screen, paper, or in other media.
- **js folder** provides [DeviceOnApis.js](#) which is the API for get or set Data from Database on the server and [RMMGlobal.js](#) which is the function to get or set the data from the local storage of Website.
- **libs** folder provides simple library, if you need another library, please download from [CDN.js](#) and place in this folder.

**Step 4:** Download [sample code](#), there are two files (demo.html, demo2.html), please place **demo.html** into “**SampleAddin**” folder.

Line 18 to 30 (demo.html) to include java script library, you could place your library in the relative path, or alternatively, given library URL from [CDNjs](#).

```

17      <!-- javascript plugins -->
18      <script src="/static/addins/SampleAddin/libs/vue-2.6.10.min.js"></script>
19      <script src="/static/addins/SampleAddin/libs/vue-tables-2-1.4.70.min.js"></script>
20      <script src="/static/addins/SampleAddin/libs/axios.min.js"></script>
21      <script src="/static/addins/SampleAddin/libs/sweetalert2.all.min.js"></script>
22      <script src="/static/addins/SampleAddin/libs/vue-sweetalert2-2.1.1.min.js"></script>
23      <script src="/static/addins/SampleAddin/libs/echarts-4.3.0.min.js"></script>
24      <script src="/static/addins/SampleAddin/libs/moment-2.24.0.min.js"></script>
25      <script src="/static/addins/SampleAddin/libs/vue-single-select.min.js"></script>
26
27      <!-- javascript common plugins -->
28      <script src="/static/addins/SampleAddin/js/RMMGlobal.js"></script>
29      <script src="/static/addins/SampleAddin/js/DeviceOnApis.js"></script>
30      <script src="/static/addins/SampleAddin/js/util.js"></script>
31

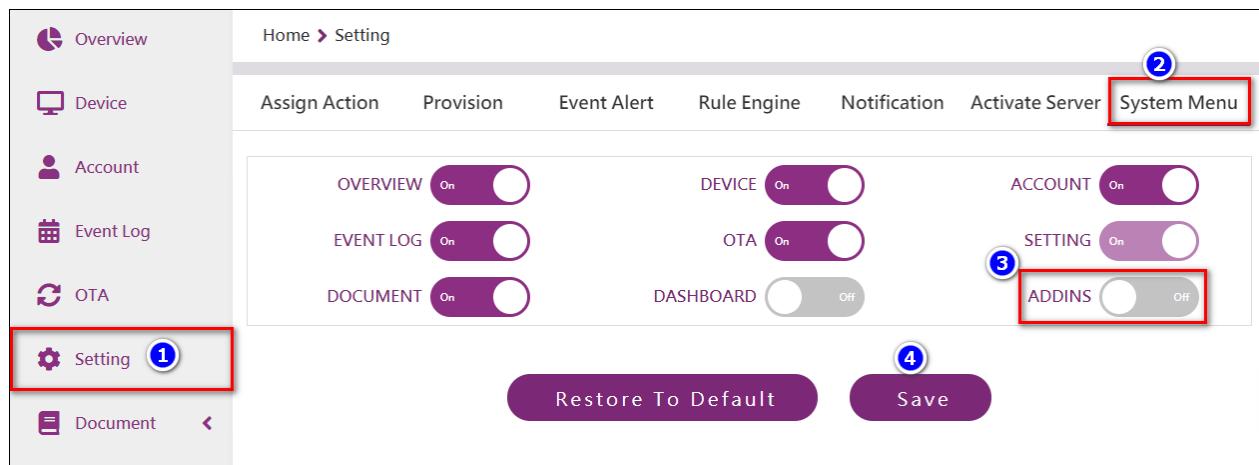
```

```

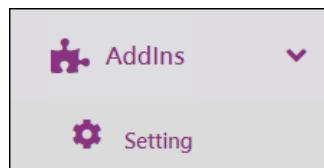
1.  <!-- CDNjs-->
2.  <script src="https://code.jquery.com/jquery.js"></script>
3.  <script src="https://cdnjs.cloudflare.com/ajax/libs/twitter-
    bootstrap/3.3.7/js/bootstrap.min.js"></script>

```

**Step 5:** Enable “**AddIN**” option from DeviceOn Server. (**Setting -> System Menu -> ADDINS**)



After the option is enabled, the “Addins” will appear in the menu item.



**Step 6:** Click on the “Setting” (Addins -> Setting) to add your Addins.

The screenshot shows the 'Add New Addins' dialog. It includes fields for NAME (demo), URL (/static/addins/SampleAddin/demo.html), and ICON (fa-smile-beam). There are also 'DELETE', 'ENABLE', and 'LABEL' columns. At the bottom are 'Save' and 'CANCEL' buttons. Three numbered circles (1, 2, 3) point to the NAME, URL, and ICON fields respectively.

NAME	DELETE	ENABLE	LABEL
demo			Setting

1 record

Add New Addins

NAME  
demo 1

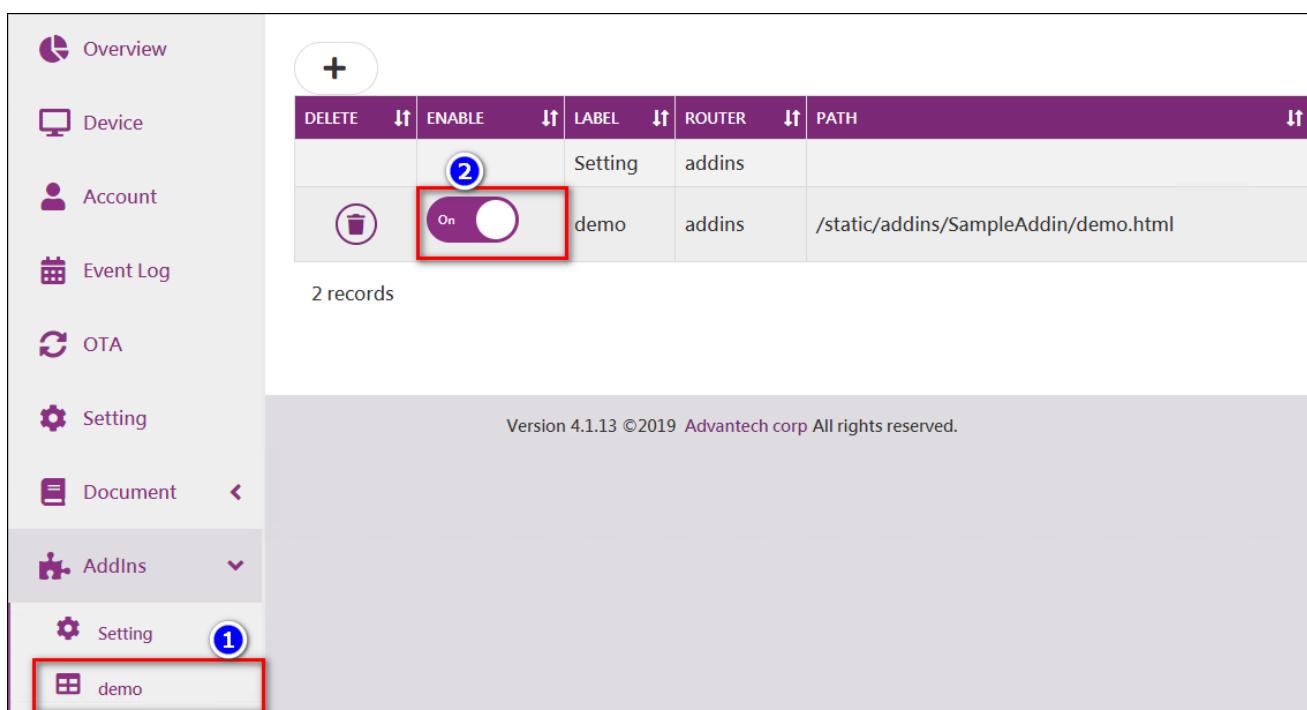
URL  
/static/addins/SampleAddin/demo.html 2

ICON (FONT AWESOME, EX : FA-CHART-LINE)  
fa-smile-beam 3

Save CANCEL

- **Name:** Label name on the menu item
- **URL:** Relative path, /static/addins/SampleAddin/**demo.html**
- **Icon:** Reference [Fontawesome](#) site to get the string of icon

After that, the “**demo**” shown on the menu item, if not, please enable the “**Addin**” on Setting page.

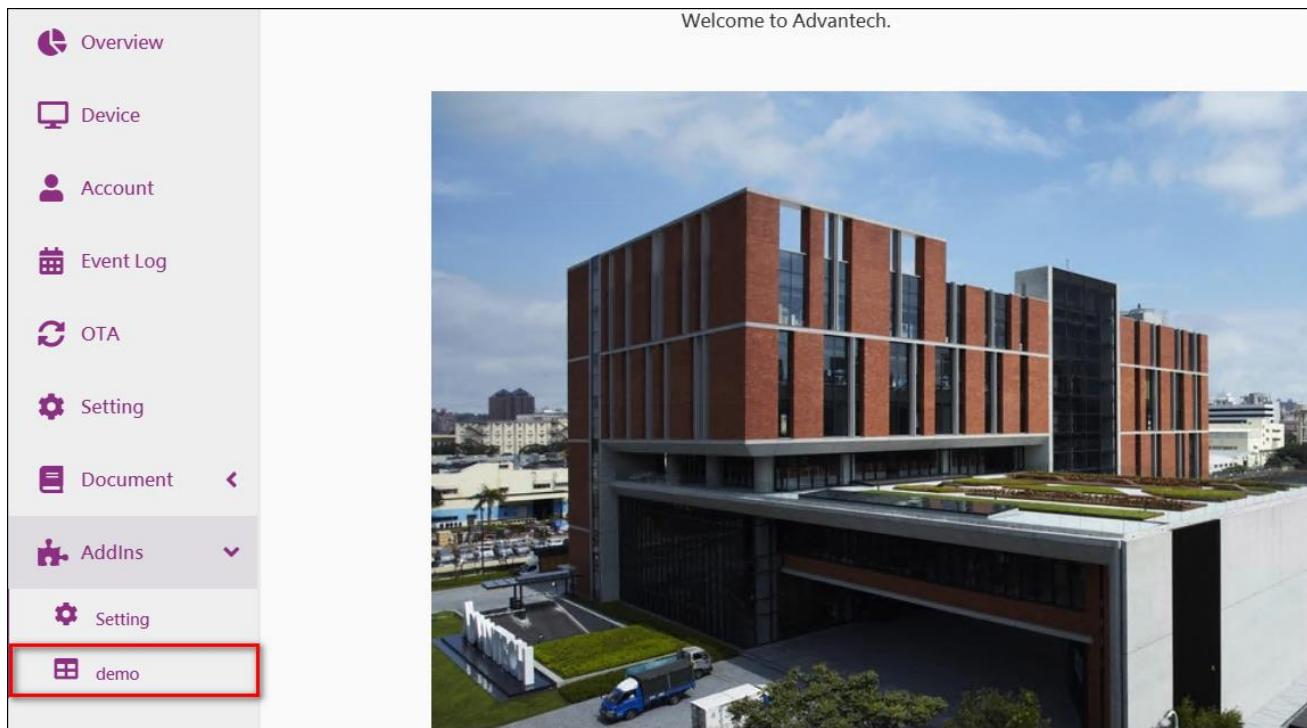


The screenshot shows the WISE-PaaS interface with the following details:

- Left Sidebar:** Overview, Device, Account, Event Log, OTA, Setting, Document, AddIns (selected), Setting (highlighted with a red box).
- Main Content:** A table titled “Addins” with the following data:
 

DELETE	ENABLE	LABEL	ROUTER	PATH
	Setting	addins		
	 (highlighted with a red box)	demo	addins	/static/addins/SampleAddin/demo.html
- Bottom:** Version 4.1.13 ©2019 Advantech corp All rights reserved.

### Step 7: Click on the “**demo**” addins.



The screenshot shows the WISE-PaaS interface with the following details:

- Left Sidebar:** Overview, Device, Account, Event Log, OTA, Setting, Document, AddIns (selected), Setting (highlighted with a red box).
- Main Content:** A large image of a modern building complex under a blue sky.
- Bottom:** Welcome to Advantech.

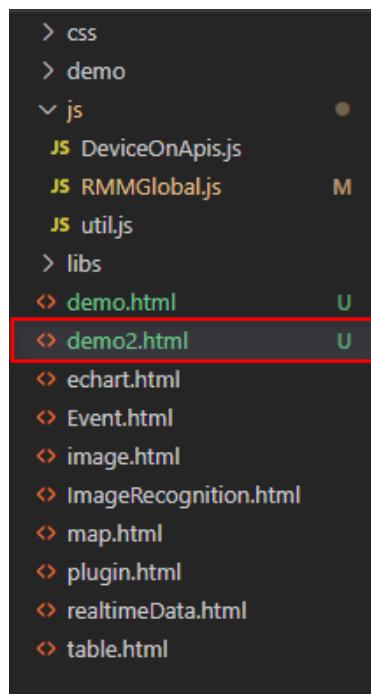
#### 5.2.4 Develop an Add-in to Access DeviceOn API

This example will show you how to get all accounts, groups and devices.

#### APIs used on below sample

1. DeviceOnApis.accounts.get.accounts(aid)  
*To get all accounts information from database.*
2. DeviceOnApis.accounts.get.deviceGroups(aid)  
*To get all groups which under this aid's account from database.*
3. DeviceOnApis.devicegroups.get.devicesAll(data)  
*To get all devices which under this aid's account from database.*
4. DeviceOnApis.devicegroups.get.devices(gid, data)  
*To get all devices which under this gid's group from database.*

**Step 1:** Download [sample code](#), there are two files (demo.html, demo2.html), please place **demo2.html** into “SampleAddin” folder.



**Step 2:** Line 10 ~22 (demo2.html) that describe library used in the Add-in.

```

8      <!-- css plugins -->
9      <link rel="stylesheet" href="/static/addins/SampleAddin/css/deviceOn.css">
10     <link rel="stylesheet" href="/static/addins/SampleAddin/css/common.css">
11     <link rel="stylesheet" href="/static/addins/SampleAddin/css/custom_echart.css">
12
13     <!-- javascript plugins -->
14     <script src="/static/addins/SampleAddin/libs/vue-2.6.10.min.js"></script>
15     <script src="/static/addins/SampleAddin/libs/axios.min.js"></script>
16     <script src="/static/addins/SampleAddin/libs/vue-single-select.min.js"></script>
17
18     <!-- javascript common plugins -->
19     <script src="/static/addins/SampleAddin/js/RMMGlobal.js"></script>
20     <script src="/static/addins/SampleAddin/js/DeviceOnApis.js"></script>
21     <script src="/static/addins/SampleAddin/js/util.js"></script>
22
23

```

Use single-select component to build demo view. (Line 27 ~ 57)

```

26 <body style="background: #FAFAFA;">
27   <div id="app">
28     <div class="content">
29       <div class="row">
30         <div class="col-md-4">
31           <div class="cus-label">Account: </div>
32           <vue-single-select v-model="selectedAccount" :options="accountOptions" option-label="name">
33             <template slot="option" slot-scope="{option, index}">
34               <div>
35                 <span style="margin-left: 1rem;">{{option.name}}</span>
36               </div>
37             </template>
38           </vue-single-select>
39         </div>
40         <div class="col-md-4">
41           <div class="cus-label">Device Group: </div>
42           <vue-single-select v-model="selectedGroup" :options="groupOptions" option-label="name"></vue-single-select>
43         </div>
44         <div class="col-md-4">
45           <div class="cus-label">Device: </div>
46           <vue-single-select v-model="selectedDevice" :options="deviceOptions" option-label="name">
47             <template slot="option" slot-scope="{option, index}">
48               <i :class="option.iconClass" :style="{'color': option.iconColor}" aria-hidden="true"></i>
49               <span style="margin-left: 1rem;">{{option.name}}</span>
50             </template>
51           </vue-single-select>
52         </div>
53       </div>
54     </div>
55   </div>
56 </body>
57
58

```

Use RMMGlobal() to get your login account ID (aid), through the aid as parameter to request API.

```

70   mounted: function () {
71     //get current user aid
72     var aid = RMMGlobal.get().Login.aid;
73     this.getAccounts(aid);
74   },

```

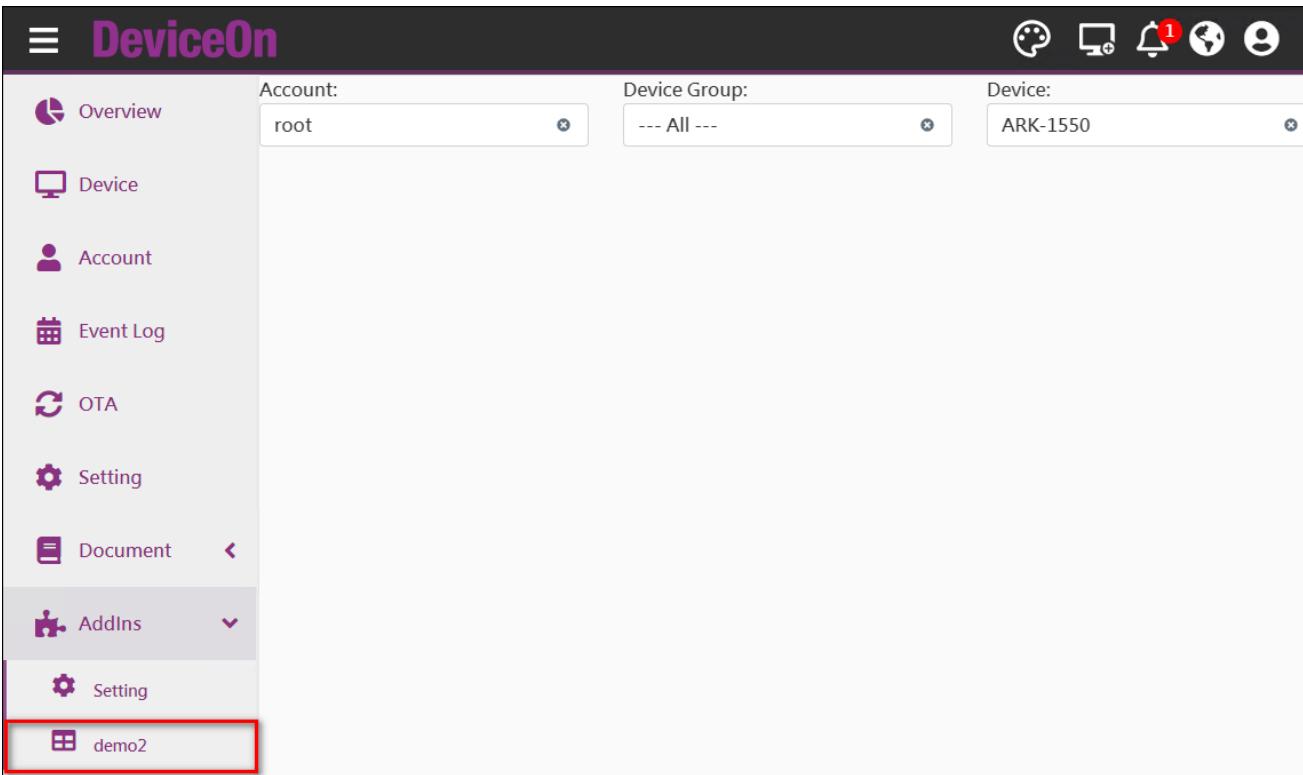
The API (**DeiceOnApis.accounts.get.accounts(aid)**) will send request to server, and return all account data.

```

83   methods: {
84     getAccounts: function (aid) {
85       DeviceOnApis.accounts.get.accounts(aid)
86         .then(function (xhr) {
87           if (xhr && xhr.data && xhr.data.accounts) {
88             vue.accountOptions = xhr.data.accounts;
89             let aAccount = vue.accountOptions.filter(function (g, i) {
90               return g.aid === Number(aid);
91             });
92             if (aAccount.length === 0 && vue.accountOptions.length > 0) {
93               vue.selectedAccount = vue.accountOptions[0];
94             } else {
95               vue.selectedAccount = aAccount[0];
96             }
97           }
98         });
99   },

```

**Step 3:** Add an Addin (demo2) as before steps.



The screenshot shows the DeviceOn web interface. On the left is a vertical navigation menu with icons and labels: Overview, Device, Account, Event Log, OTA, Setting, Document, AddIns (with a dropdown arrow), Setting, and demo2. The 'demo2' item is highlighted with a red box. At the top, there are three search/filter boxes: 'Account:' with 'root', 'Device Group:' with '--- All ---', and 'Device:' with 'ARK-1550'. To the right of these boxes are several small icons representing different functions.

## 6. FAQ

### 6.1 Why Some of Devices Cannot Power On

REF: <https://www.lifewire.com/wake-on-lan-4149800/>

The DeviceOn leverage Wake-on-LAN (WoL) mechanism to remote power your device on, there are 2 steps to should be configured at first. Wake-on-LAN (WoL) is a network standard that allows a computer to be turned on remotely, whether it's hibernating, sleeping, or even completely powered off. It works by receiving what's called a "magic packet" that's sent from a WoL client.

It also doesn't matter what operating system the computer will eventually boot into (Windows, Mac, Ubuntu, etc.), Wake-on-LAN can be used to turn on any computer that receives the magic packet. A computer's hardware does have to support Wake-on-LAN with a compatible BIOS and network interface card, so not every computer is automatically able to use Wake-on-LAN.

#### Two-step WoL Setup

Enabling Wake-on-LAN is done in two steps, both of which are described below. The first sets up the motherboard by configuring Wake-on-LAN through BIOS before the operating system boots, and the next logs into the operating system and makes some small changes there. The first step with the BIOS is valid for every computer, but after following the BIOS setup, skip down to your operating

system instructions, whether it be for Windows, Mac, or Linux.

### Step 1: BIOS Setup

The first thing you need to do to enable WoL is to set up BIOS correctly so that the software can listen for incoming wake up requests.

Every manufacturer will have unique steps, so what you see below may not describe your setup exactly. If these instructions aren't helping, find out your BIOS manufacturer and check their website for a user manual on how to get into BIOS and find the WoL feature.

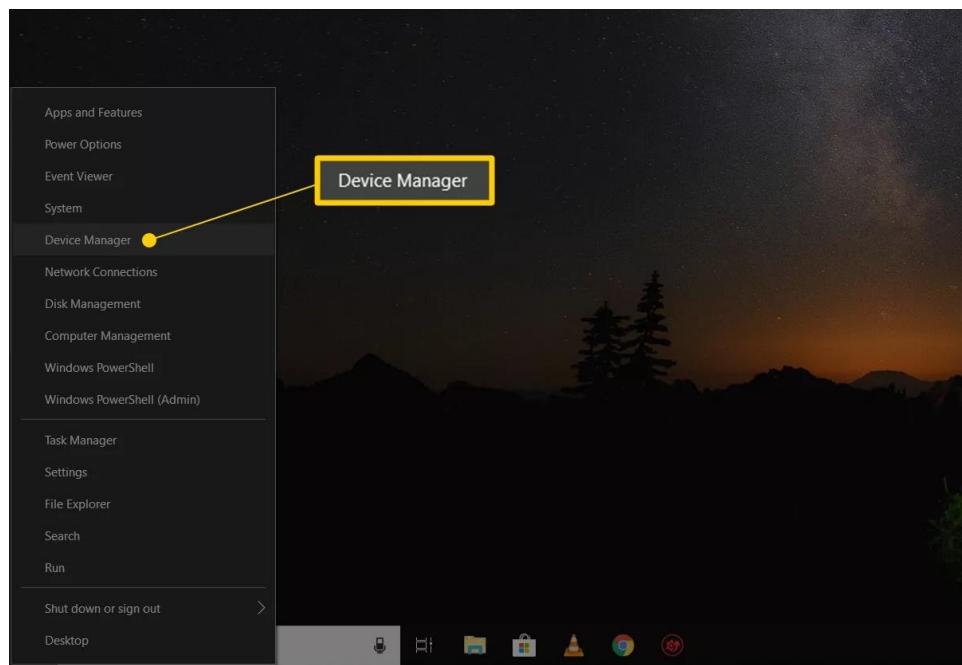
1. Enter BIOS instead of booting to your operating system.
2. Look for a section that pertains to power, such as Power Management. This may be under an Advanced section. Other manufacturers might call it Resume On LAN, such as on the Mac.  
Most BIOS screens have a help section off to the side that describes what each setting does when enabled. It's possible that the name of the WoL option in your computer's BIOS isn't clear.
3. Once you find the WoL setting, you can most likely press **Enter** to either immediately toggle it on or to show a small menu that allows you to toggle it on and off, or enable it and disable it.
4. Save the changes. This isn't the same on every computer, but on many the **F10** key will save and exit BIOS. The bottom of the BIOS screen should give some instructions about saving and exiting.

CMOS Setup Utility - Copyright (C) 1984-2000	
Power Management Setup	
ACPI Function	Enabled
ACPI Suspend Type	S1
Power Management	User Define
Video Off Method	DPMs
Video Off In Suspend	Yes
Suspend Type	CPU Sleep Mode
MODEM Use IRQ	NA
Suspend Mode	Disabled
HDD Power Down	Disabled
Soft-Off by PWR-Button	Delay 4 Sec.
Wake On PCI Card	By OS
Wake On Modem	Disabled
Wake On LAN	Enabled
Wake On RTC	By OS

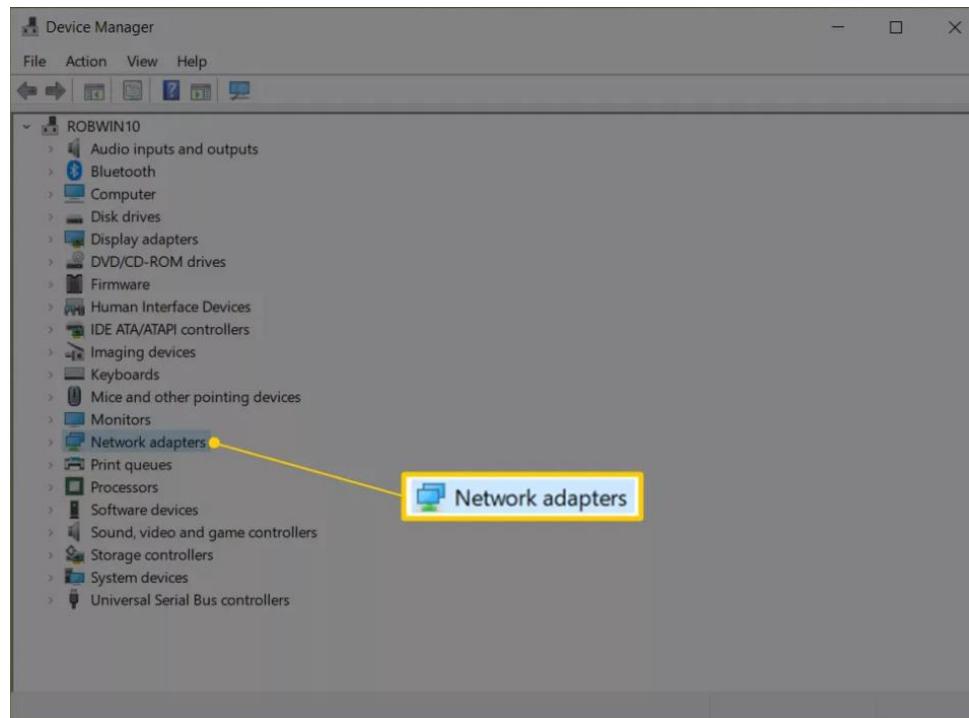
### Step 2: Windows operating system WoL setup

[Windows Wake-on-LAN](#) is set up through Device Manager. There are a few different settings to enable here:

1. Open Device Manager

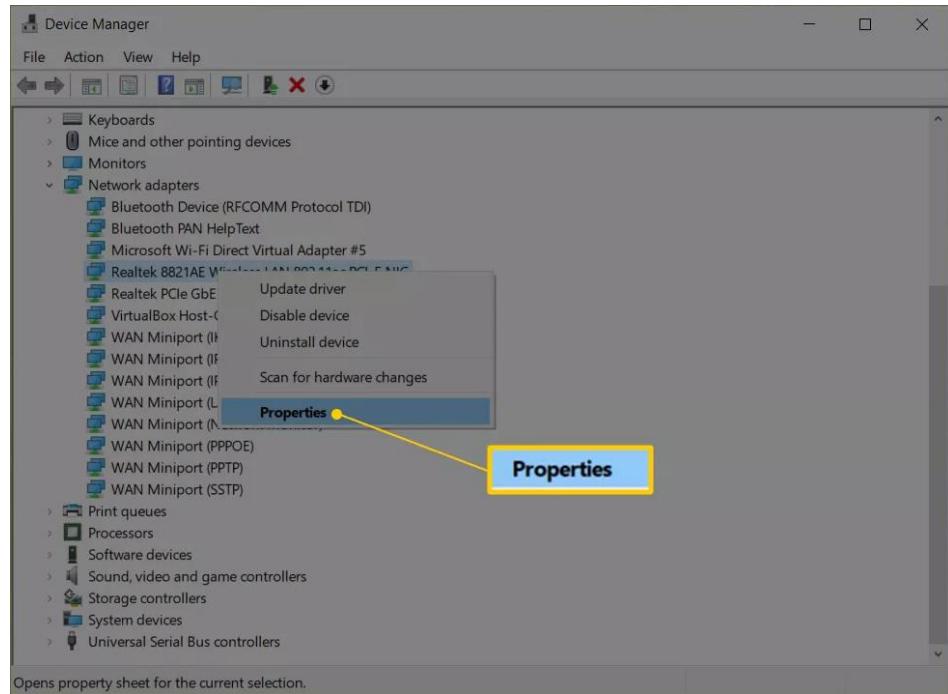


2. Find and open the Network adapters section.

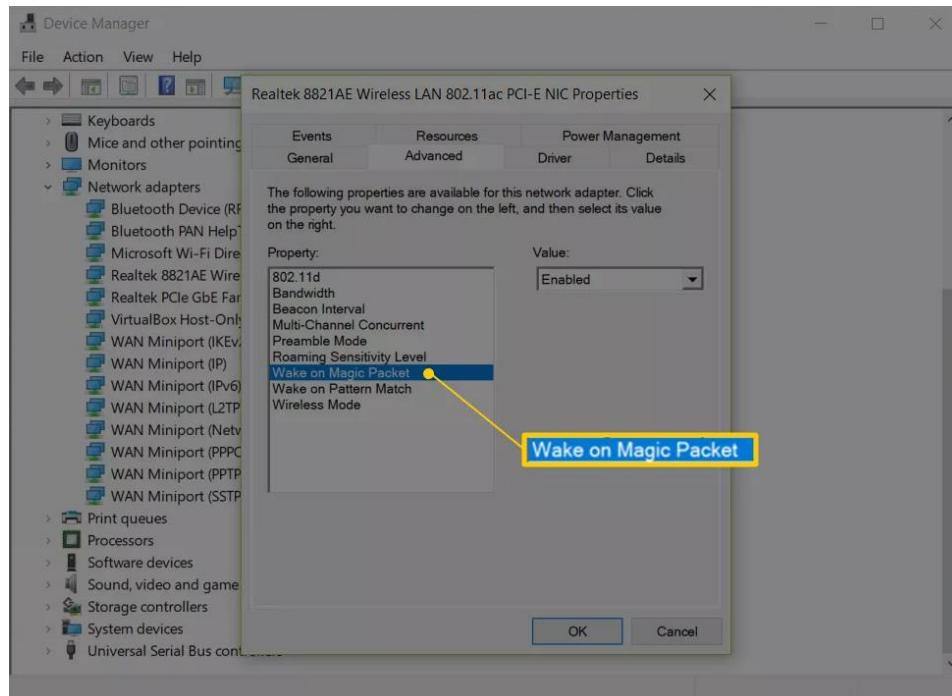


You can ignore any Bluetooth connections and virtual adapters. Double-click (or double-tap) **Network adapters** or select the small + or > button next to it to expand that section.

3. Right-click or tap-and-hold the adapter that belongs to the active internet connection. Examples of what you might see are **Realtek PCIe GBE Family Controller** or **Intel Network Connection**, but it will vary depending on your computer.
  4. Choose **Properties**.

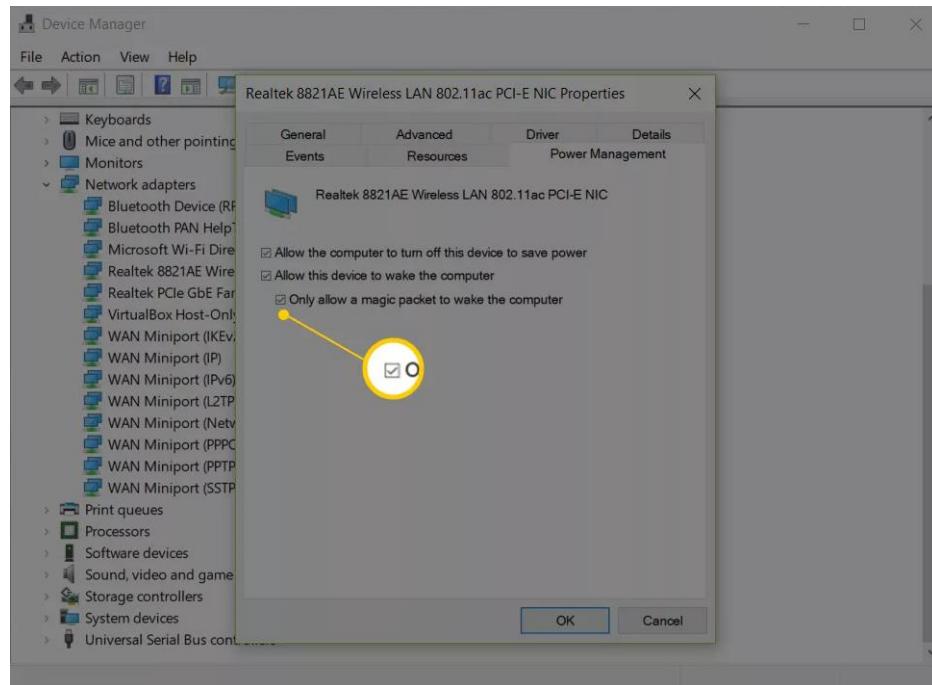


5. Open the **Advanced** tab.
6. Under the **Property** section, click or tap **Wake on Magic Packet**. If you can't find this, skip to Step 8; Wake-on-LAN might still work anyway.



7. From the **Value** menu on the right, choose **Enabled**.
8. Open the **Power Management** tab. It might be called **Power** depending on your version of Windows or network card.
9. Make sure these two options are enabled: **Allow this device to wake the computer** and **Only**

allow a magic packet to wake the computer.

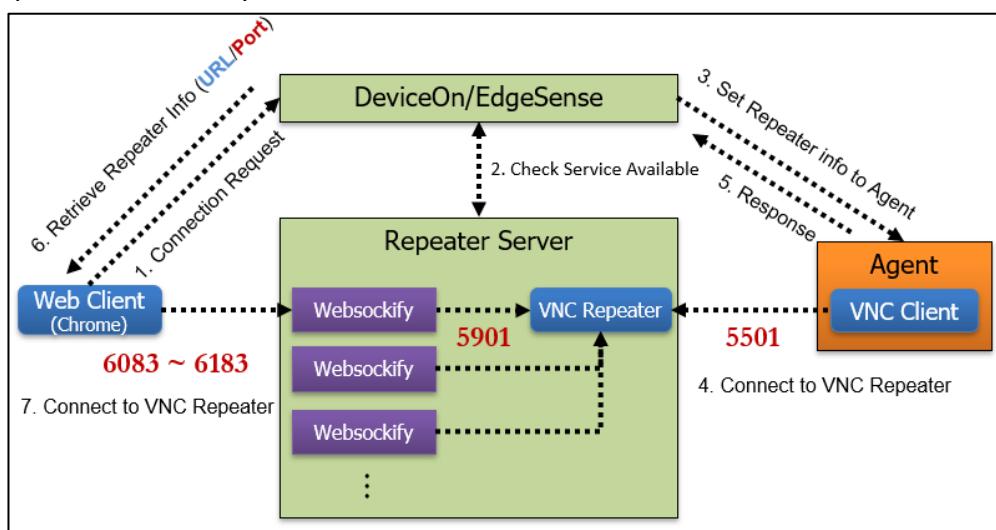


These settings might instead be under a section called Wake-on-LAN and be a single setting called **Wake on Magic Packet**.

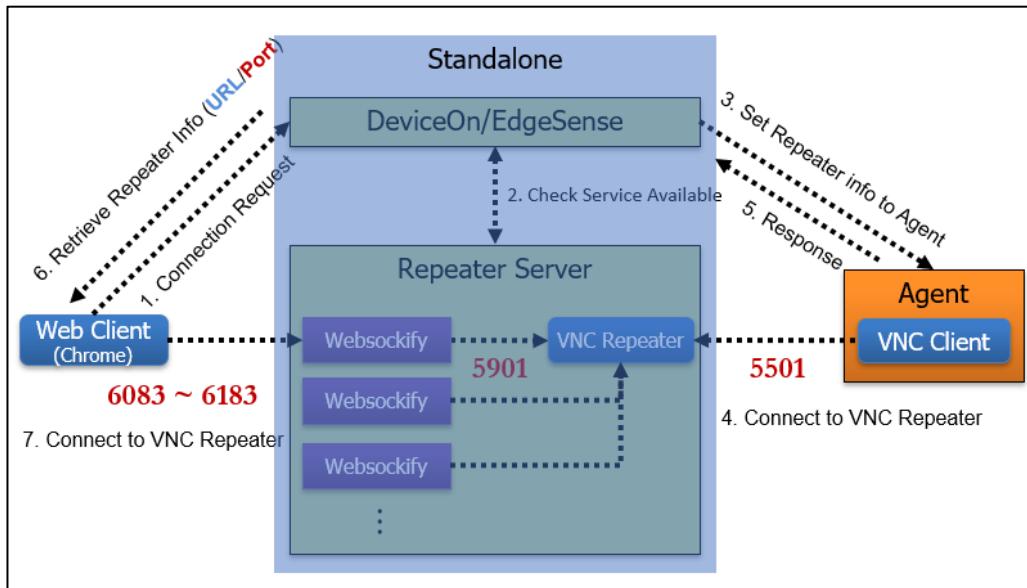
10. Click or tap OK to save the changes and exit that window. You can also close down Device Manager.

## 6.2 Why Cannot Remote Control via KVM (Remote Desktop)

The DevicOn leverage VNC (Virtual Network Computing) technology to achieve remote desktop, to bridge different network between public and private. We build-up a Repeater server on public site for WISE-PaaS/EnSaaS and Azure PaaS. There is a web-client through WebSocket (port: 6083 ~6183) mechanism connect to Repeater and device via 5501 to Repeater, the structure as below. Please help confirm the port available on your browser and device side.

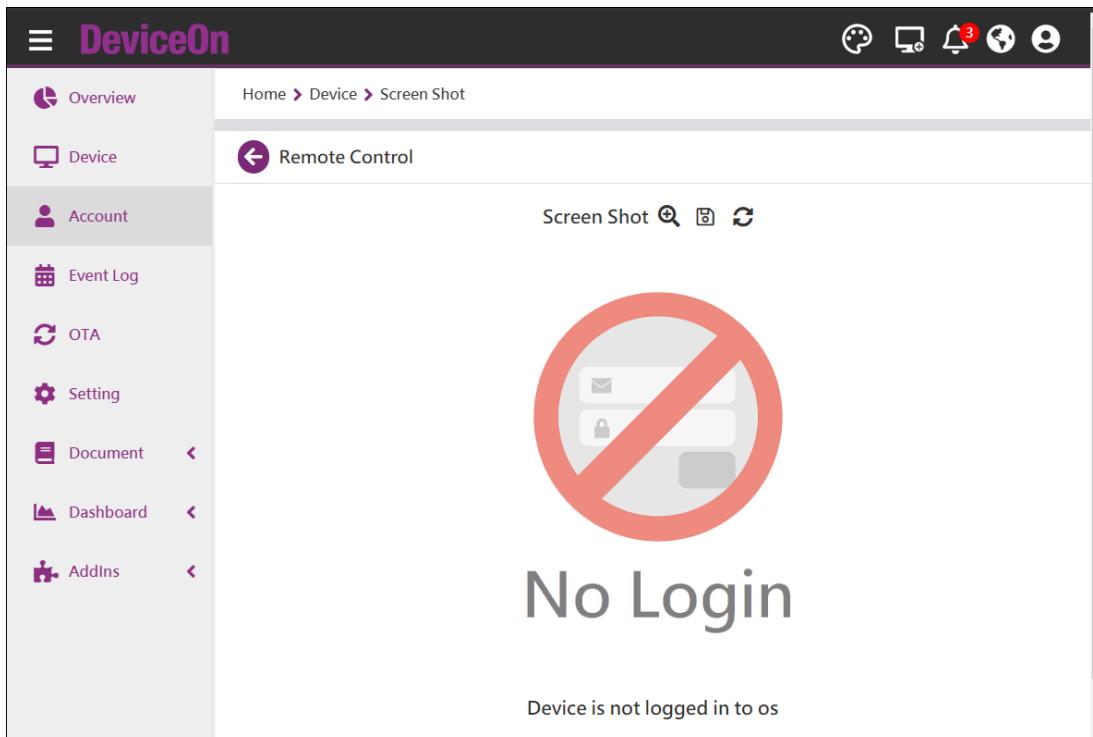


If the DeviceOn running on VM, standalone version, the Repeater also build into same machine, please reference the structure, make sure the VM available for these inbound and outbound ports.

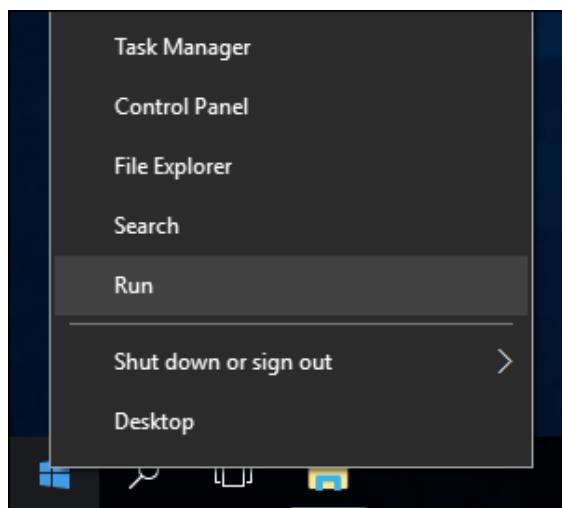


### 6.3 Why Cannot Screenshot and Always Show Device “No Login”

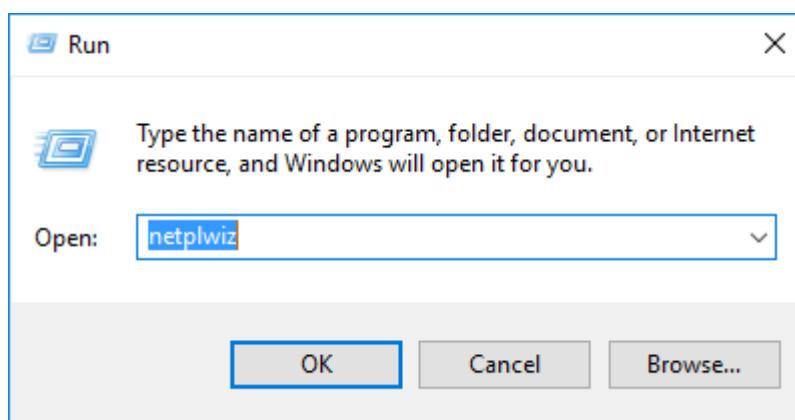
To fix the “No Login” error, you can sign into the system manually, or set the “Automatically Sign in to Windows 10”.



**Step 1:** Right-click the Start button and select Run from the hidden quick access menu, or use the keyboard shortcut Windows Key  + R to bring up the Run dialog.

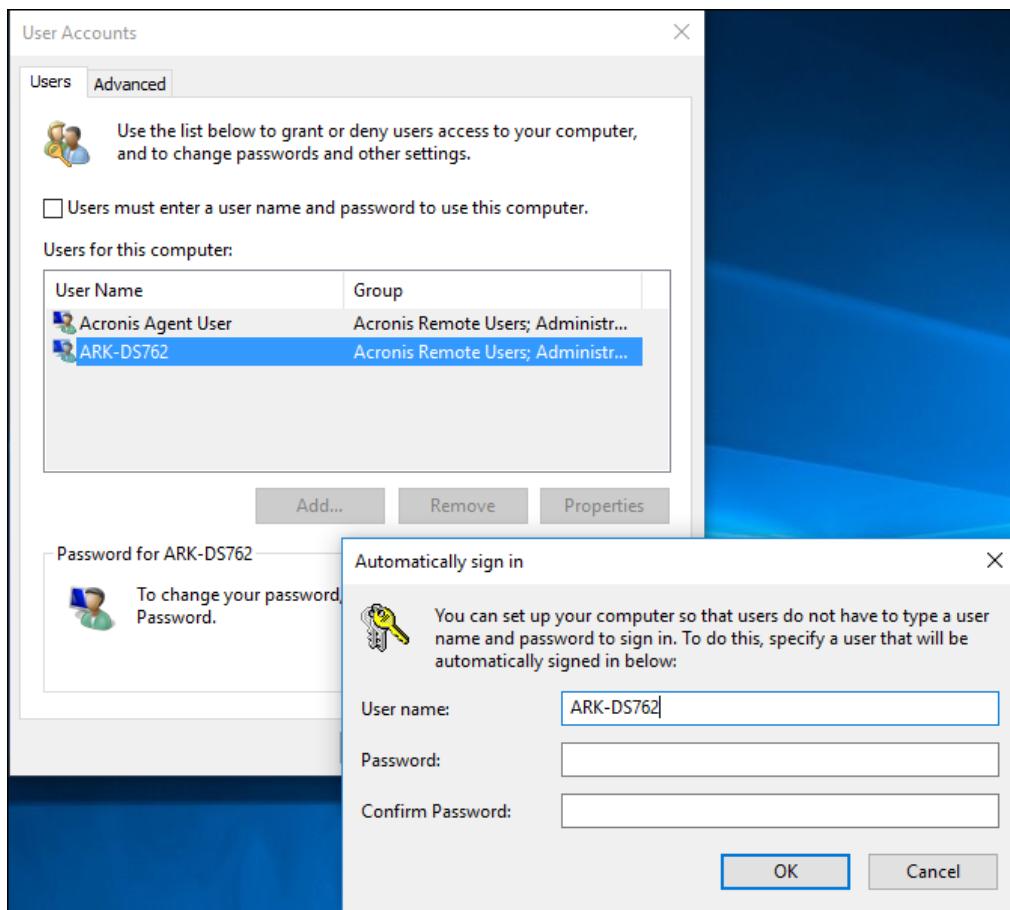


**Step 2:** Now Then Type: *netplwiz* and hit Enter or click OK.



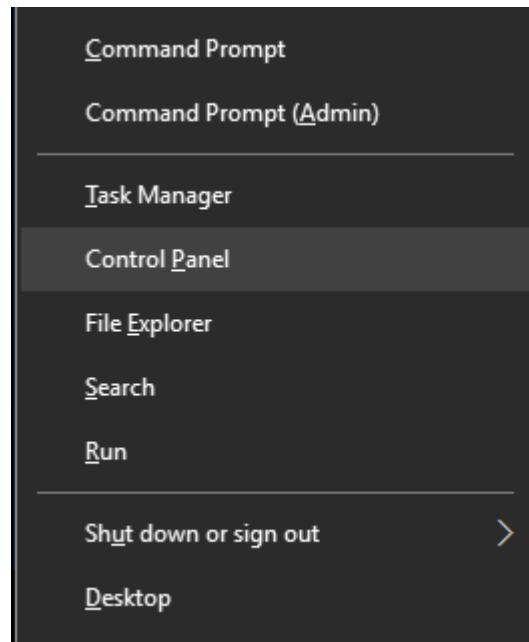
**Step 3:** Uncheck Users must enter a user name and password to use this computer and click OK.

**Step 4:** Enter in your user name and the password you use to log into your system twice and click OK.

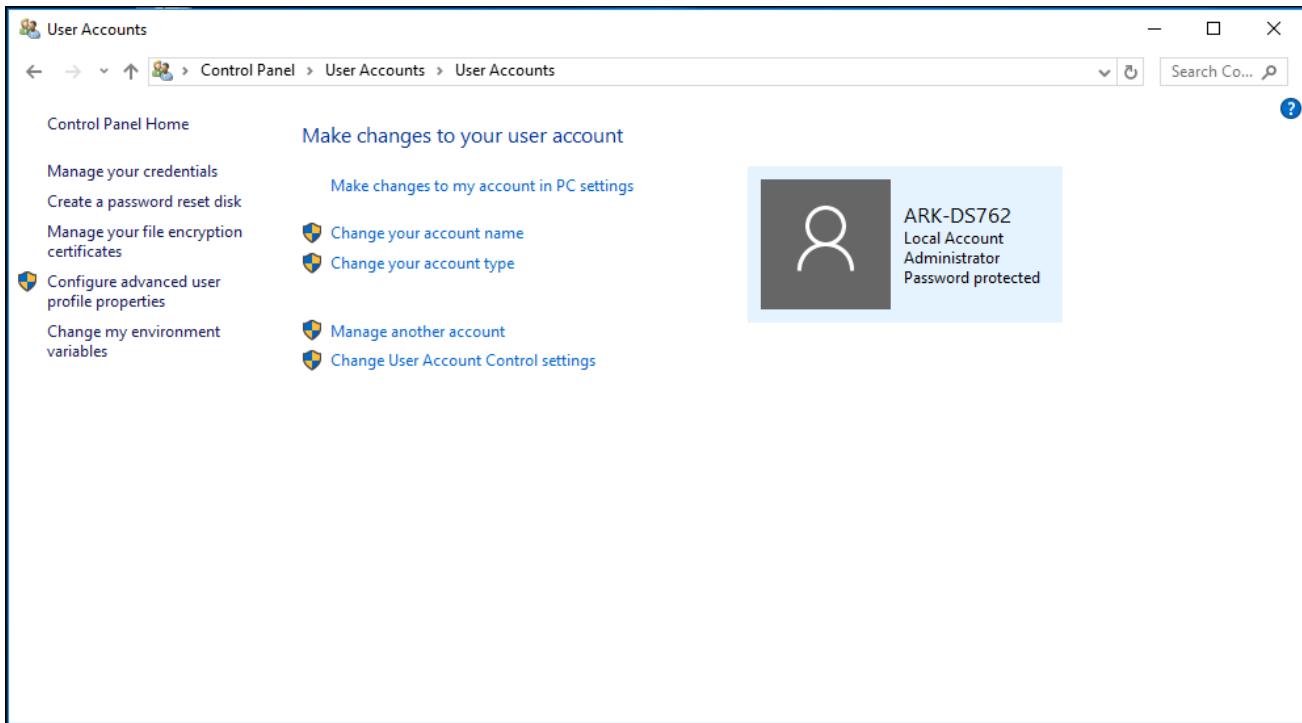


If you still get the “No Login” error or get the “black screen”, then you can try to disable the Windows User Account Control (UAC).

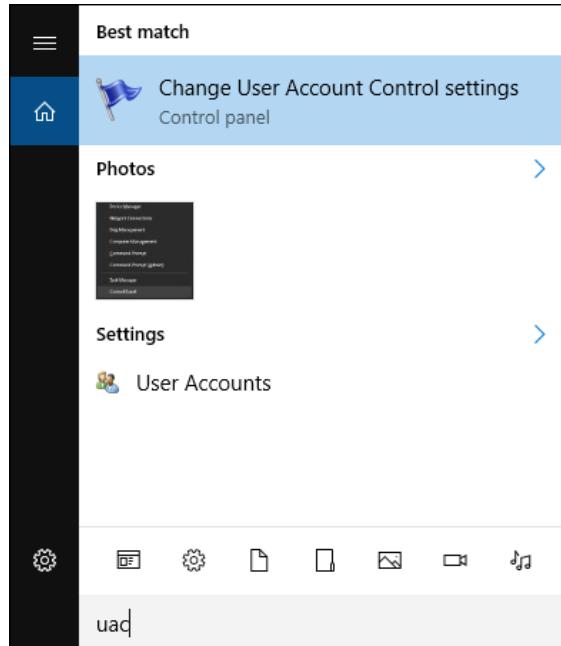
**Step 1:** Press Windows Key  + X hotkeys together on the keyboard and choose the "Control Panel" item.



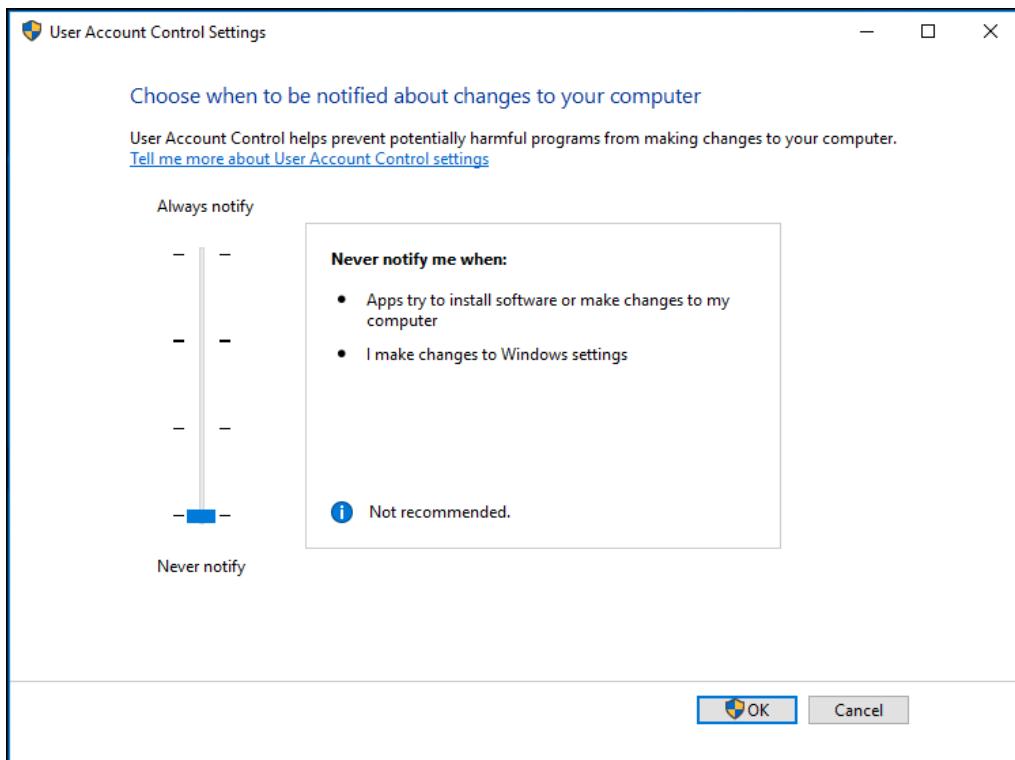
**Step 2:** Go to the following path: “**Control Panel\>User Accounts\User Accounts**” There you will find the Change User Account Control settings link. Click it.



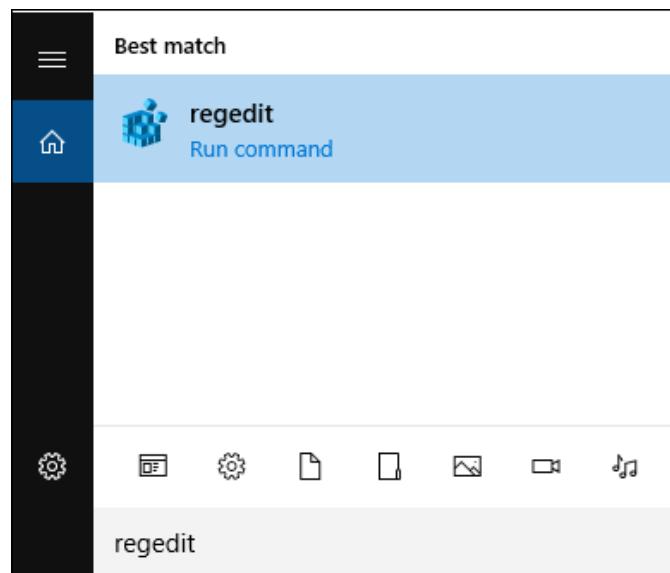
Alternatively, you can enter the “**UAC**” in the Search box to open the User Account Control settings dialog.



**Step 3:** In the User Account Control settings dialog, move the slider to the bottom (Never Notify).

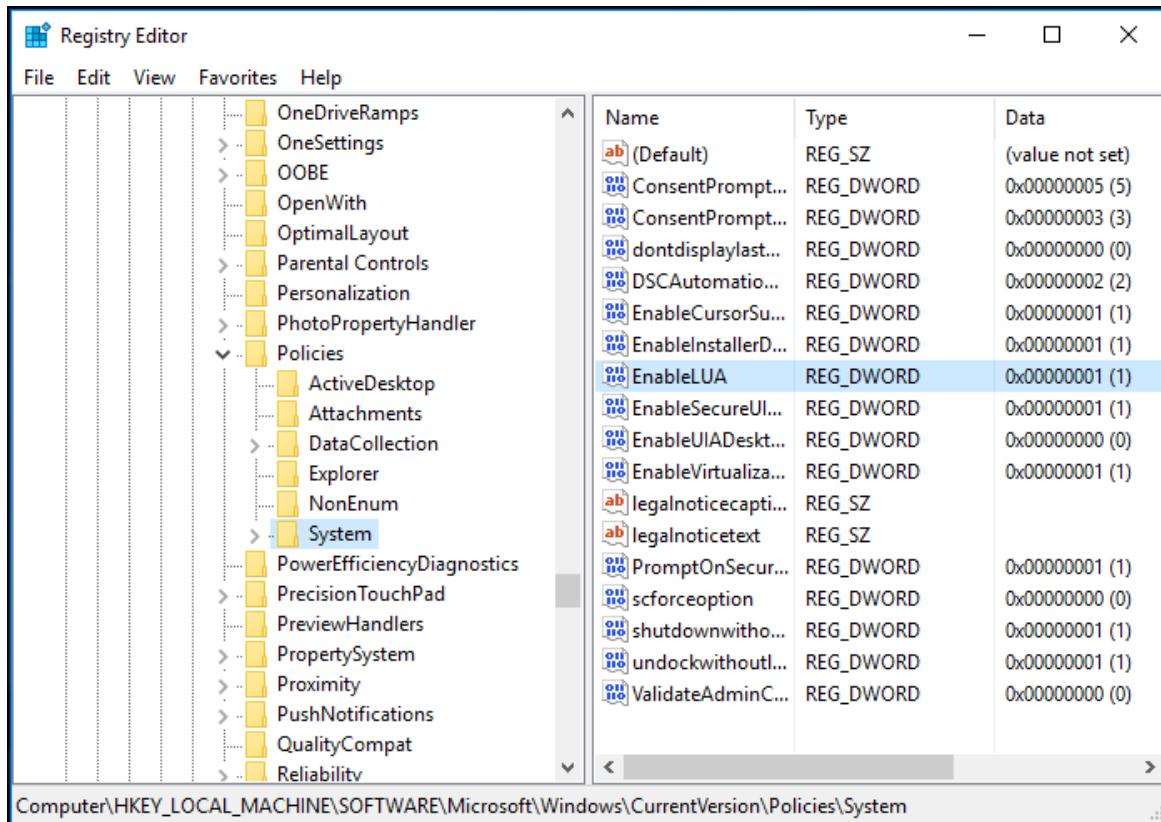


**Step 4:** Enter the “*regedit*” in the Search box to open the Registry Editor.

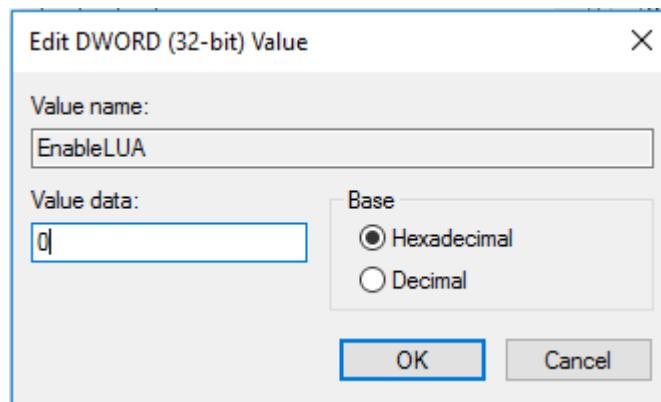


**Step 5:** Navigate to the following key:

“*HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\System*”

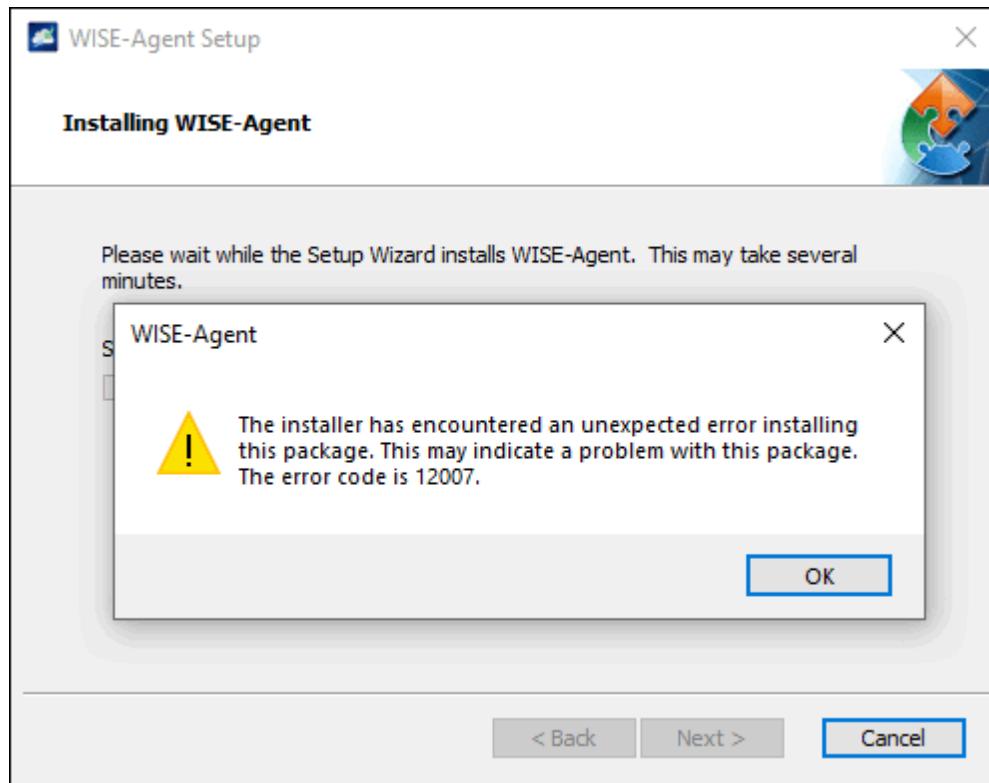


**Step 6:** In the right pane, modify the value of the **EnableLUA** DWORD value and set it to **0**.



**Step 7:** Restart your computer.

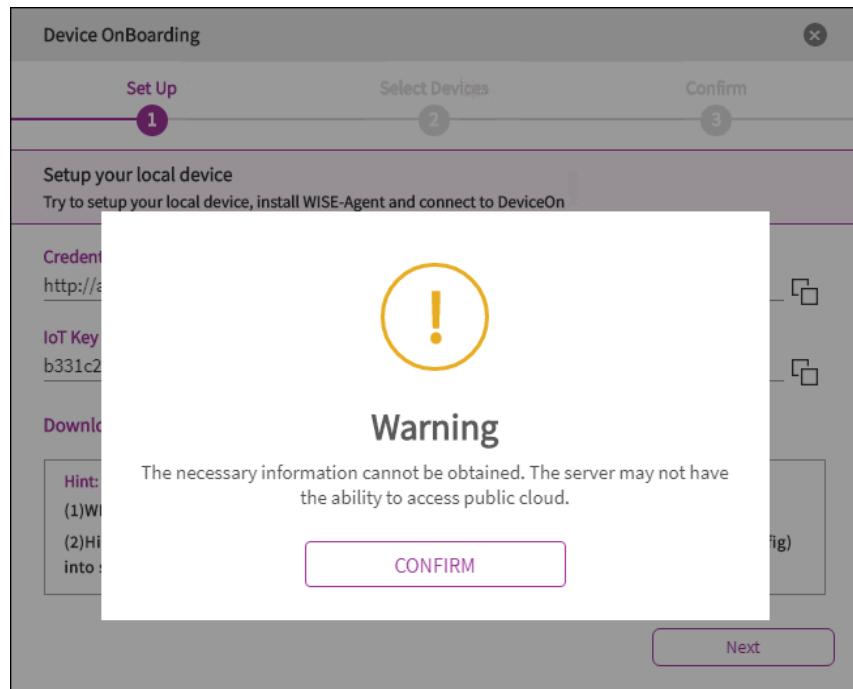
## 6.4 Why the WISE-Agent Cannot Install? With error code 12007?



WISE-Agent requires the Microsoft Visual C++ Redistributable 2008, 2013, 2015 x86 packages, which will be downloaded from the Internet and set up during the installation process. If you are in an environment with limited or no Internet access, please download the [Agent Dependency Package](#) through an Internet connected device and install this package first.

## 6.5 Why the WISE-Agent Cannot Download from Device Onboarding?

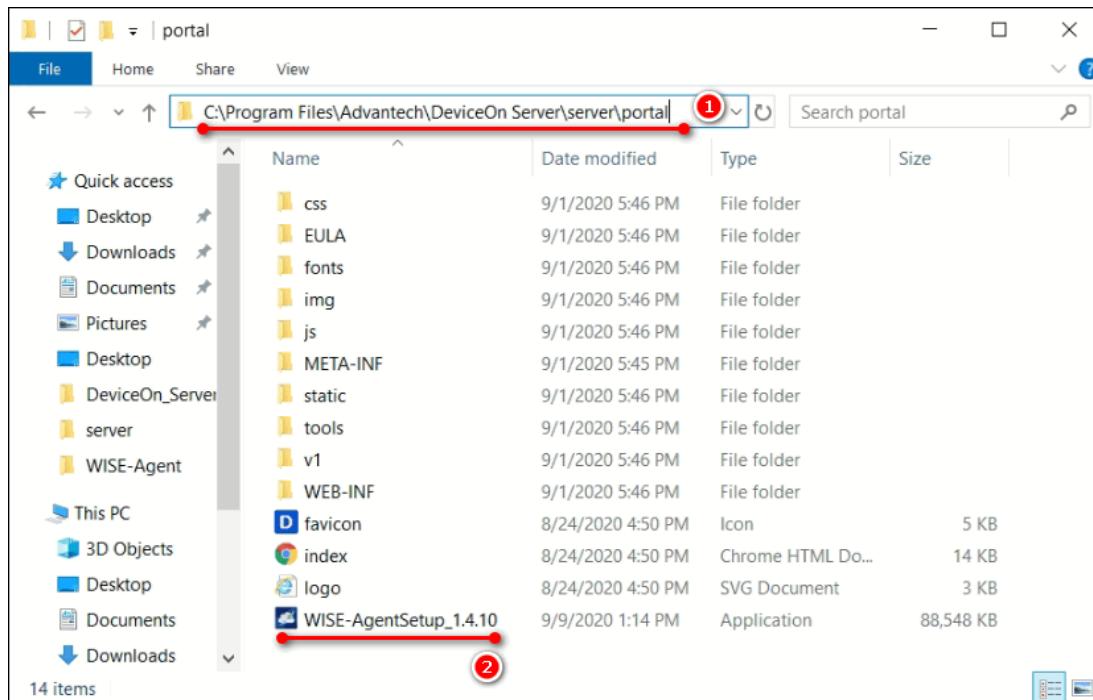
In order to allow users to obtain the latest and stable WISE-Agent, the DeviceOn team will place the latest version on the cloud. When this message appears, it means that your server network cannot access the cloud or does not have network connectivity.



You could download WISE-Agent through your mobile device or laptop and put it in the following path.

The file name must be "**WISE-AgentSetup\_x.y.z**".

Installation Path\DeviceOn Server\server\portal\



## 6.6 Why the Acronis and McAfee failed to install?

Since the installer package require .Net Framework 3.5 dependency, please help to confirm is .Net Framework 3.5 installed on your devices.

## 6.7 Why Your SMTP Server Cannot Send a Mail?

### Case I: Your DeviceOn service is deployed on Azure cloud and your SMTP server adopt port 25.

Starting on November 15, 2017, outbound email messages that are sent directly to external domains (such as outlook.com and gmail.com) from a virtual machine (VM) are made available only to certain subscription types in Microsoft Azure. Outbound SMTP connections that use TCP port 25 were blocked. (Port 25 is primarily used for unauthenticated email delivery.)

This change in behavior applies only to new subscriptions and new deployments since November 15, 2017. [Referenced site>](#)

### Case II: Alway authentication failed through your Gmail account.

Step 1: Less secure apps & your Google Account.

Please enter to the [page](#) with your Google account and set it to **Enable**.

Step 2: Unlocking Google's Gmail CAPTCHA

Please enter to the [page](#) with your Google account and click **Continue**.

## 6.8 How the Device Data Flow and Debug from Edge to Cloud

The **WISE-DeviceOn** offers a general solution of gathering device, equipment and sensor information from the edge device via WISE-Agent. This document will walk you through how the data will be transmitted to **WISE-PaaS/EnSaaS** and the data flow over our architecture. Besides, it also covers how to clarify the issues while using **WISE-DeviceOn**.

For those advanced system integrators, below figure illustrates an overview of data flow among our software components. You also can access several documents (links of these docs...) that will help you to dive deeper of each item. As an example of document item 1, how to retrieve credential for device to connect to **WISE-PaaS/EnSaaS** IoTHub. We implemented a virtual host mechanism in IoTHub to process messages independently for different space in case data confusion of messages in same topic. Each key can only be used for accessing data for one specific space as below figure. Key No. 1 can neither publish nor subscribe data for space B, but only key No. 4 is available to access IoTHub for space B.

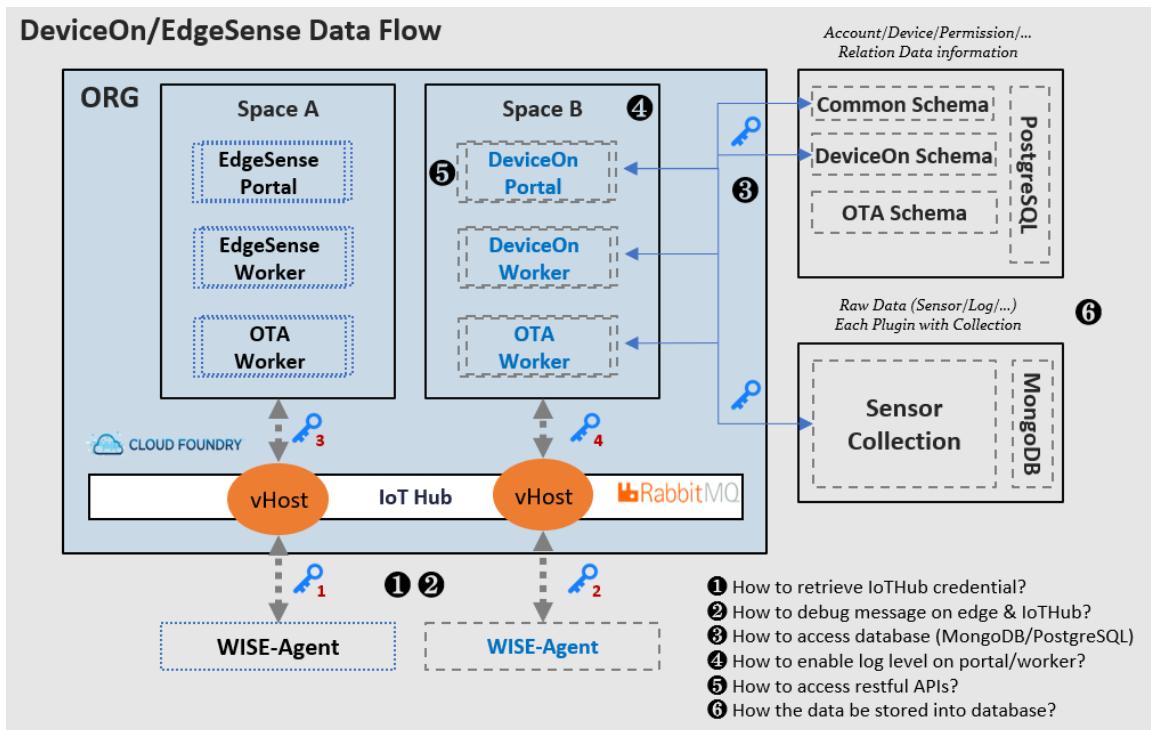
Furthermore, one 3<sup>rd</sup> party tool **node-RED** can help you to check if all the data are transmitted to IoTHub properly. It's an open source project has been uploaded on the GitLab, as [document](#) item2 will walk you through how to publish node-RED application on our **WISE-PaaS/EnSaaS** and how to monitor messages on edge and IoTHub.

Once the data has been published to target space of **WISE-PaaS/EnSaaS**, the worker will process it and store it into corresponding databases. There are two databases we adopted; one is relational

database **PostgreSQL** for storing relational data. For instance, MAC address, device name, platform name, OS information...etc. The other is NoSQL database **MongoDB** for storing sensor raw data. Hence, document as [item 3](#) will walk you through how to access these databases to make sure all the data are stored in databases properly.

In **WISE-DevieOn**, we implemented a log system to record each operation. As document [item 4](#) will walk you through how to enable log message for advanced error tracking. Besides, as document [item 5](#), it will also walk you through how to accessing data from database directly by using RESTful APIs for data visualization or application development.

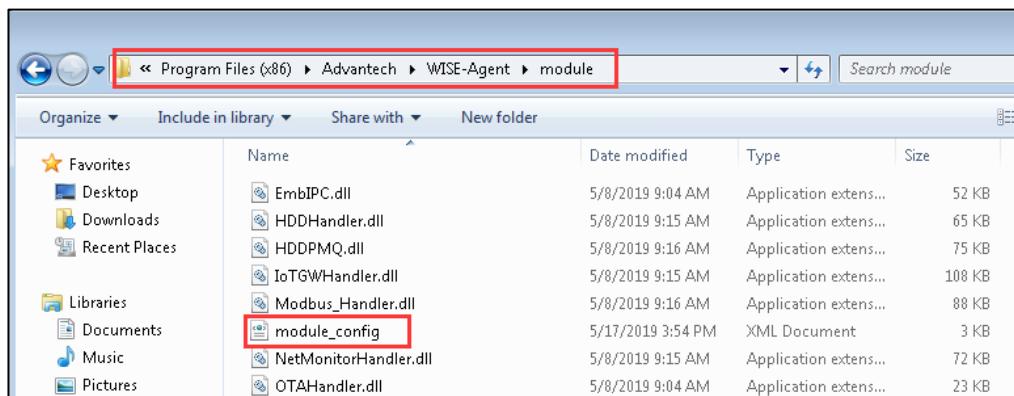
Finally, document [item 6](#) gives an example to show you what kind of data will be stored in **PostgreSQL** and **MongoDB** respectively once edge devices are connected.



## 6.9 How to Enable/Disable plugins on WISE-Agent

### Step 1: Adjust configuration file on WISE-Agent

Open **module\_config.xml** on Installation path\module\



Adjust “ModuleEnable” to TRUE/FALSE to enable and disable.

```

1  <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
2  <XMLConfigSettings>
3      <BaseSettings>
4          <ModuleNum>15</ModuleNum>
5          <ModuleName1>HDDHandler</ModuleName1>
6          <ModulePath1>\module\HDDHandler.dll</ModulePath1> HDD Monitoring
7          <ModuleEnable1>TRUE</ModuleEnable1>
8          <ModuleName2>PowerOnOffHandler</ModuleName2>
9          <ModulePath2>\module\PowerOnOffHandler.dll</ModulePath2>
10         <ModuleEnable2>TRUE</ModuleEnable2>
11         <ModuleName3>ScreenshotHandler</ModuleName3>
12         <ModulePath3>\module\ScreenshotHandler.dll</ModulePath3>
13         <ModuleEnable3>TRUE</ModuleEnable3>
14         <ModuleName4>NetMonitorHandler</ModuleName4>
15         <ModulePath4>\module\NetMonitorHandler.dll</ModulePath4>
16         <ModuleEnable4>TRUE</ModuleEnable4>
17         <ModuleName5>ProcessMonitorHandler</ModuleName5>
18         <ModulePath5>\module\ProcessMonitorHandler.dll</ModulePath5>
19         <ModuleEnable5>TRUE</ModuleEnable5>

```

## Step 2: Restart WISE-Agent

Open “Task Manager” and switch to “Services”

Task Manager			
File Options View			
Processes Performance App history Startup Users Details Services			
Name	PID	Description	Status
WinHttpAutoProxySvc	2892	WinHTTP Web Proxy Auto-Discover...	Running
Winmgmt	2084	Windows Management Instrumentation...	Running
WinRM		Windows Remote Management (WS...	Stopped
WISEAgentService	3636	WISEAgentService	Running
wisvc		Windows Insider Service	Stopped
WlanSvc		WLAN AutoConfig	Stopped
wlidsvc	8888	Microsoft Account Sign-in Assistant	Running

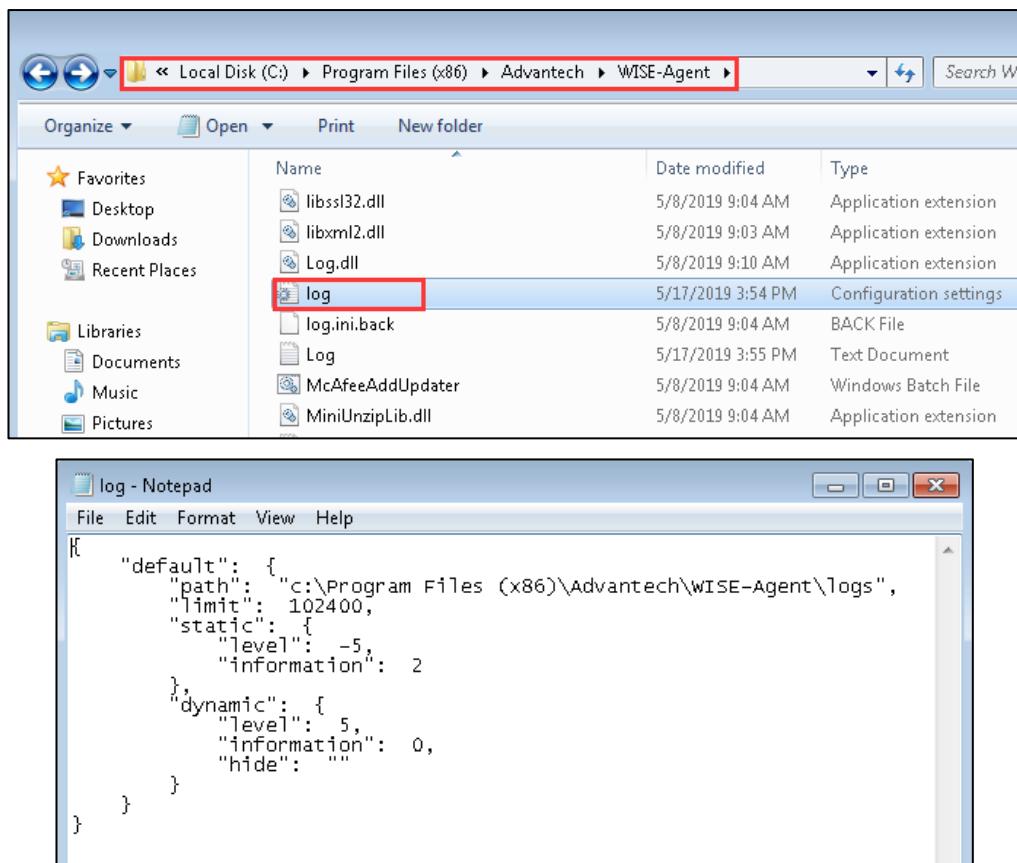
Restart “WISEAgentService” to connect to DeviceOn

## 6.10 How to Enable and Adjust WISE-Agent Log Levels

### [WISE-Agent v-1.3.x & v-1.2.x]

#### Step 1: Adjust configuration file on WISE-Agent

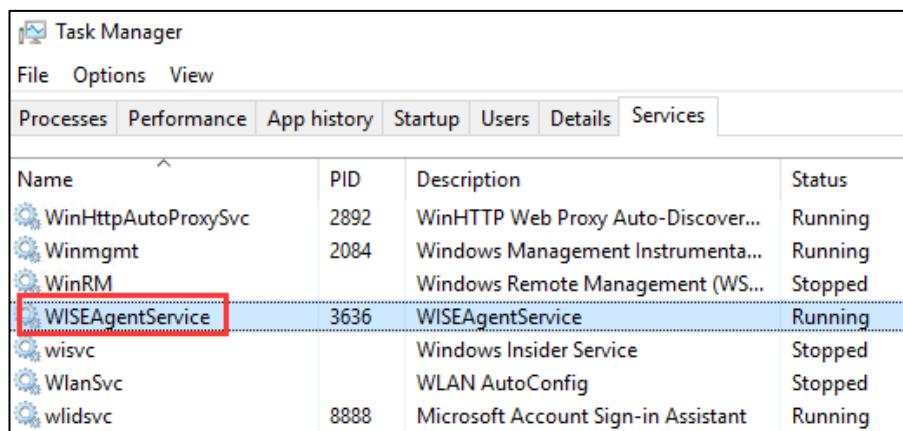
Open **log.ini** on Installation path



Adjust level 5 to 7, minus stand for HTML format.

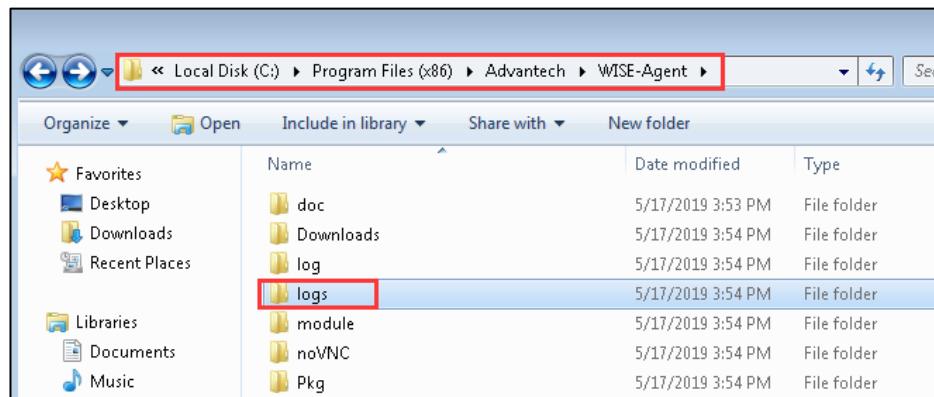
#### Step 2: Restart WISE-Agent

Open “Task Manager” and switch to “Services”, and restart “WISEAgentService”



#### Step 3: Retrieve log files from WISE-Agent

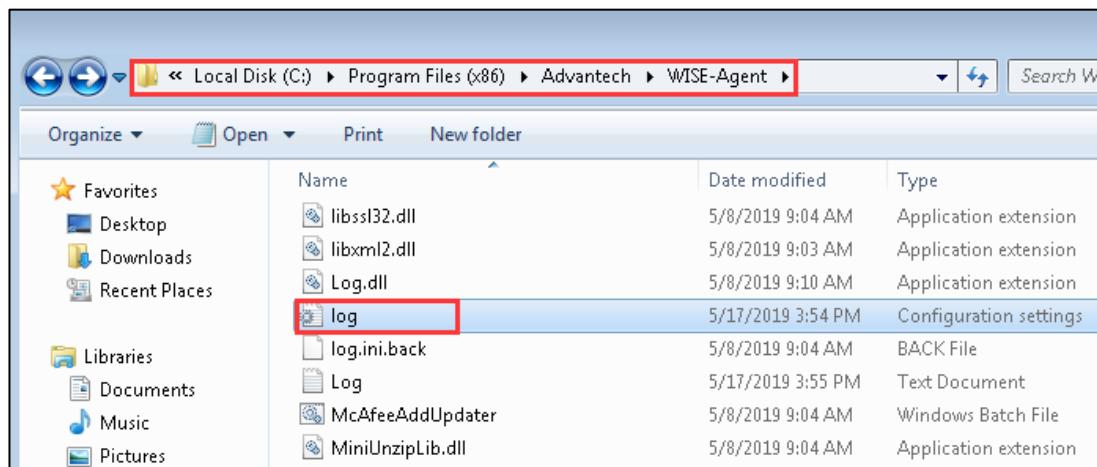
The log files under the Installation path\logs



### [WISE-Agent v-1.4.x and above]

#### Step 1: Adjust configuration file on WISE-Agent

Open **log.ini** on Installation path\module\



```

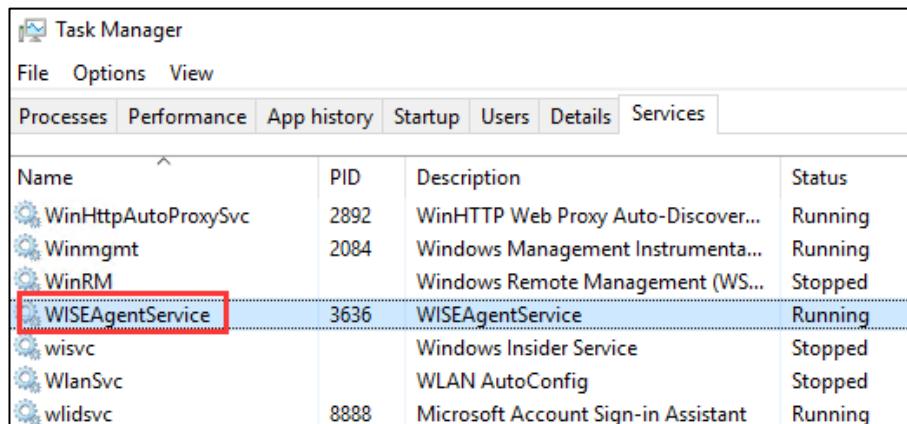
[LogClient]
#log_level=4, LOG_FATAL(0), LOG_ALARM(1), LOG_ERROR(2), LOG_WARNING(2), LOG_NORMAL(4), LOG_DEBUG(5)
log_level=5
#to_stderr=1, 1: print to stderr, 0: doesn't print stderr
#logd_ip=127.0.0.1, ip of logd
#logd_port=9278

```

Adjust **log\_level** from 4 to 5.

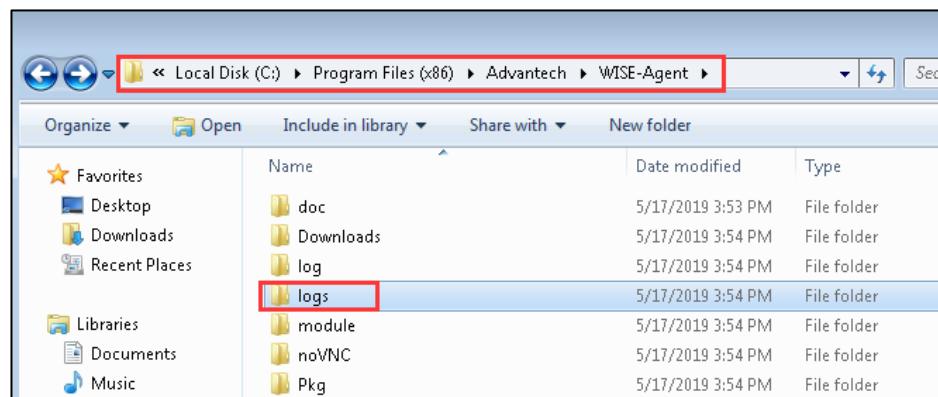
#### Step 2: Restart WISE-Agent

Open “Task Manager” and switch to “Services”, and restart “WISEAgentService”



### Step 3: Retrieve log files from WISE-Agent

The log files under the Installation path\logs

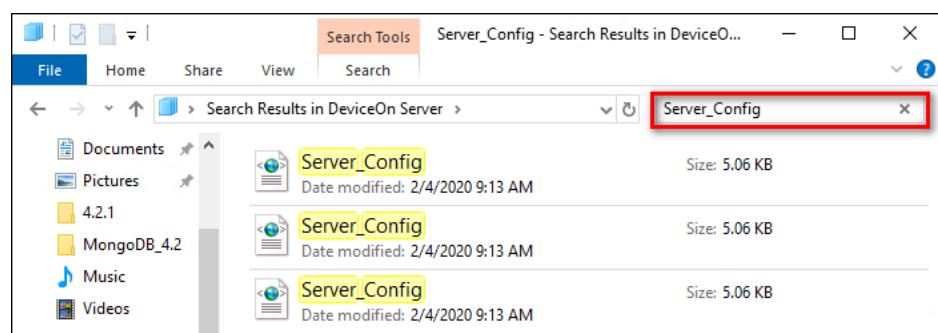


### 6.11 How to Change DeviceOn Server Address (Standalone)

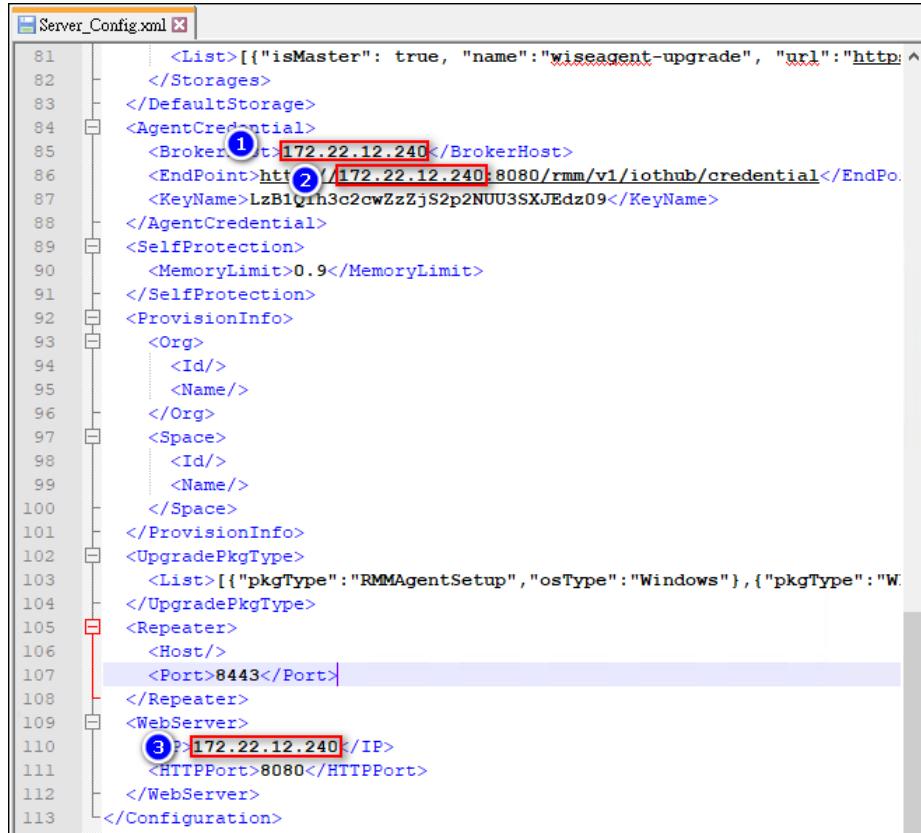
If your DeviceOn Server (Standalone) running on public cloud or on premise environment, and then you would like to update DeviceOn Server address, due to machine/VM IP changed. Here are few steps to update server setting.

#### Step 1: Search Server\_config.xml on installation path,

(example, C:\Program Files\Advantech\DeviceOn Server\)



**Step 2:** Open these files with notepad or other txt editor, and then update host IP address to below path.

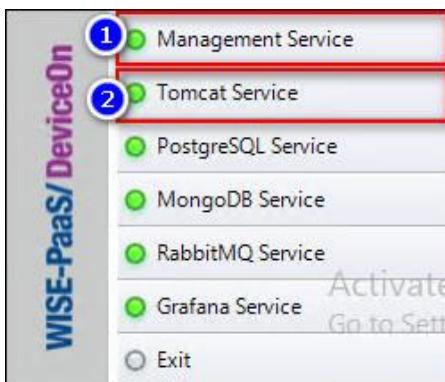


```

<List>[{"isMaster": true, "name":"wiseagent-upgrade", "url":"http://172.22.12.240:8080/rmm/v1/iothub/credential"}]
</BrokerHost>
<EndPoint>http://172.22.12.240:8080/rmm/v1/iothub/credential</EndPoint>
<KeyName>LzB1Q1h3c2cwZzZjS2p2NUU3SXJEdz09</KeyName>
</AgentCredential>
<SelfProtection>
    <MemoryLimit>0.9</MemoryLimit>
</SelfProtection>
<ProvisionInfo>
    <Org>
        <Id/>
        <Name/>
    </Org>
    <Space>
        <Id/>
        <Name/>
    </Space>
</ProvisionInfo>
<UpgradePkgType>
    <List>[{"pkgType":"RMMAgentSetup", "osType":"Windows"}, {"pkgType":"W...</List>
</UpgradePkgType>
<Repeater>
    <Host/>
    <Port>8443</Port>
</Repeater>
<WebServer>
    <IP>172.22.12.240</IP>
    <HTTPPort>8080</HTTPPort>
</WebServer>
</Configuration>

```

**Step 3:** Restart the **Tomcat** and **Management Services** through DeviceOn Server Control.



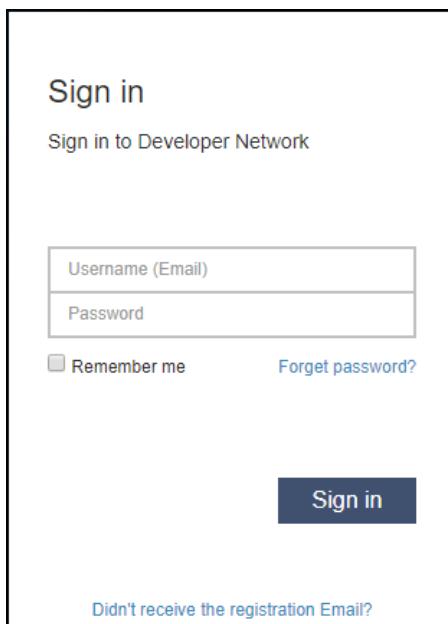
## 6.12 How to Migrate/Transfer EdgeSense Database to DeviceOn (WISE-PaaS/EnSaaS)

Actually, the DeviceOn is a new product for IoT device management and the backend cores, database structure is based on EdgeSense to develop. In the section, we give a few steps to migrate, transfer database from EdgeSense to DeviceOn. Before the steps, you should prepare the database tool, download and install the program.

PostgreSQL: pg\_dump, psql

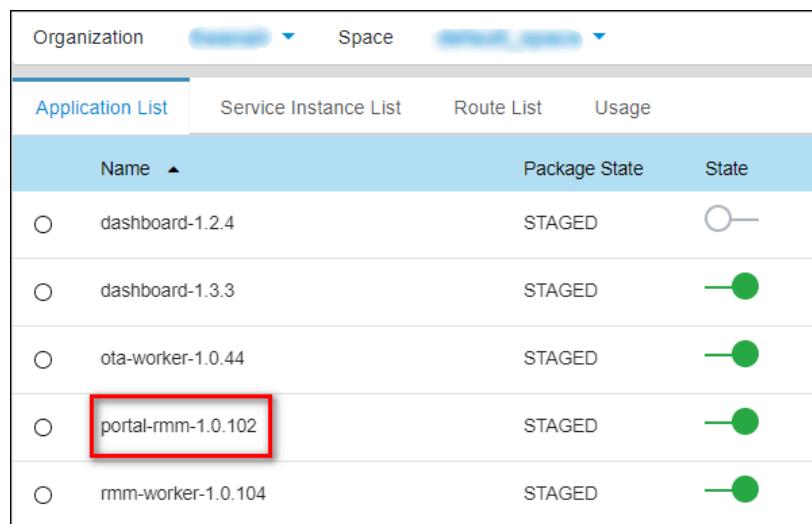
[MongoDB](#): mongodump, mongorestore

**Step 1:** Sign in to your WISE-PaaSEnSaaS Management portal



The image shows the 'Sign in' page of the WISE-PaaS Management portal. It features a header 'Sign in' and a sub-header 'Sign in to Developer Network'. Below these are two input fields: 'Username (Email)' and 'Password'. There is a 'Remember me' checkbox and a 'Forgot password?' link. A large blue 'Sign in' button is centered at the bottom. At the very bottom, there is a link 'Didn't receive the registration Email?'

**Step 2:** Enter to your organization, space and listing your applications.

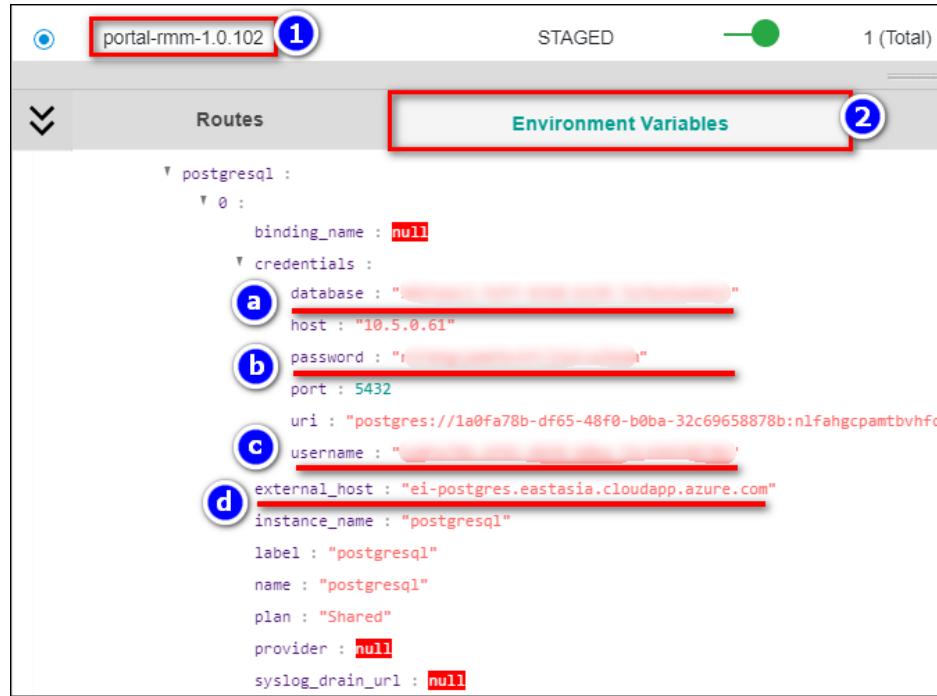


The image displays a table titled 'Application List' within a management interface. The table has columns for 'Name', 'Package State', and 'State'. The 'Name' column is sorted by name. The 'State' column contains green circular icons with white bars. One row, 'portal-rmm-1.0.102', is highlighted with a red rectangular border.

Name	Package State	State
dashboard-1.2.4	STAGED	
dashboard-1.3.3	STAGED	
ota-worker-1.0.44	STAGED	
portal-rmm-1.0.102	STAGED	
rmm-worker-1.0.104	STAGED	

**Step 3:** Retrieve PostgreSQL information via Application ("portal-rmm-1.0.x") environment, click on the application.

- a. DATABASE\_NAME
- b. DATABASE\_PASSWORD
- c. DATABASE\_USERNAME
- d. DATABASE\_HOST

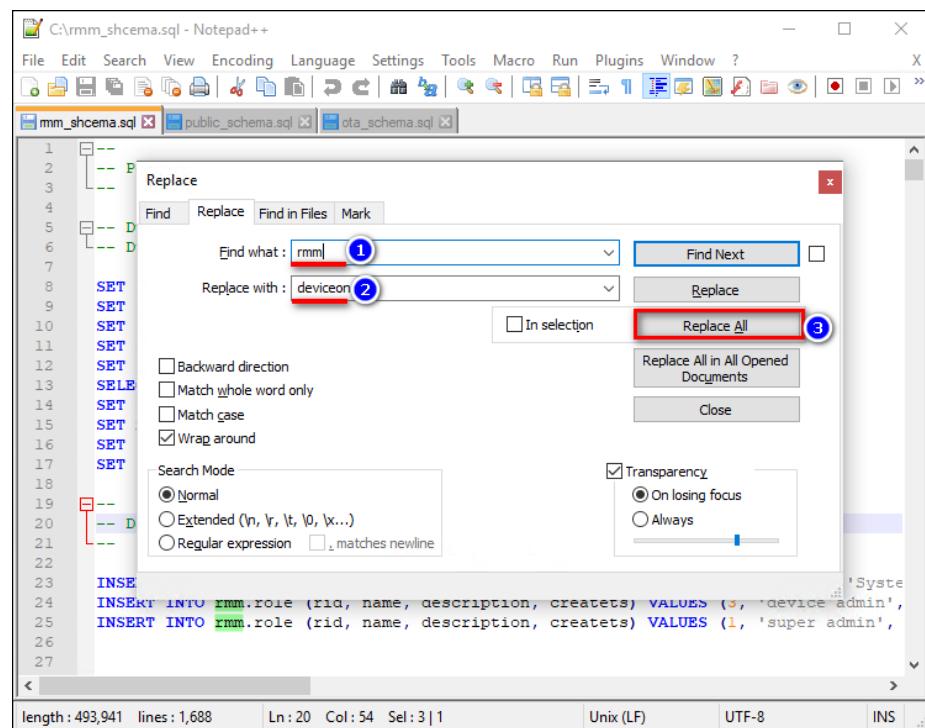


**Step 4:** Start to backup PostgreSQL data, open the terminal and enter to your PostgreSQL tool path, for example, <INSTALLATION\_PATH>\PostgreSQL\11\bin\

Run the following commands and give a password to backup 3 schema data only.

1. pg\_dump.exe -h **DATABASE\_HOST** -U **DATABASE\_USERNAME** --column-inserts --data-only --schema=**rmm** --dbname=**DATABASE\_NAME** --file=d:\rmm\_schema.sql
2. pg\_dump.exe -h **DATABASE\_HOST** -U **DATABASE\_USERNAME** --column-inserts --data-only --schema=**public** --dbname=**DATABASE\_NAME** --file=d:\public\_schema.sql
3. pg\_dump.exe -h **DATABASE\_HOST** -U **DATABASE\_USERNAME** --column-inserts --data-only --schema=**ota** --dbname=<**DATABASE\_NAME**> --file=d:\ota\_schema.sql

**Step 5:** Open **rmm\_schema.sql** on text editor tool, replace “**rmm**” word to “**deviceon**”.



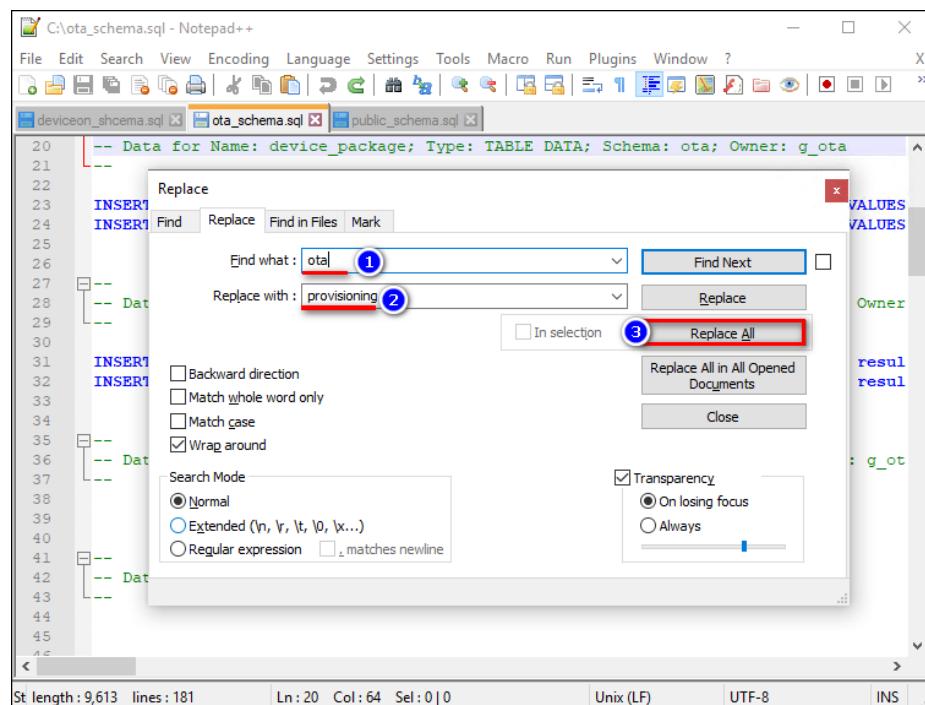
Then, remove or mark the data on “servicekey”, save as another file ([deviceon\\_schema.sql](#))

```

27
28   -- Data for Name: servicekey; Type: TABLE DATA; Schema: deviceon; Owner: g_deviceon
29
30
31
32   INSERT INTO deviceon.servicekey (kid, guid, name, createts, endpoint, enable, md5) VALUES (1, 'ee9a0c28-0429-4f3d-83e3-001648a2a200')
33   INSERT INTO deviceon.servicekey (kid, guid, name, createts, endpoint, enable, md5) VALUES (2, '55feb4ce-d5f2-4a20-8300-001648a2a200')
34
35

```

**Step 6:** Open [ota\\_schema.sql](#) on text editor tool, replace “ota” word to “provisioning”, and save as another file ([provisioning\\_schema.sql](#))



**Step 7:** Before to restore database to **DeviceOn**, please retrieve related information on Management portal, such as Database name, user name, password and host. On WISE-PaaS 3.0, the steps similar to previous, click on the application (portal-deviceon-1.1.x) and get the information via environment.



The screenshot shows the WISE-PaaS Management Portal interface. At the top, there are two applications listed: "portal-deviceon-1.1.53" (selected, highlighted with a red box and circled with a blue number 1) and "provisioning-worker-1.1.53". Both are in the "STAGED" state with 1 total instance. Below the applications, there is a section for "Routes" and "Environment Variables". The "Environment Variables" section is highlighted with a red box and circled with a blue number 2. Inside this section, there is a detailed configuration for a PostgreSQL route named "postgresql". The configuration includes:

- A "binding\_name" field set to **null**.
- A "credentials" object with the following fields:
  - a**: "database": "redacted5", "host": "10.5.0.61", "password": "redacted", "port": 5432, "uri": "postgres://20928b83-44a1-473d-8100-e110692ac8c4:1vmmmk5qfvg7rmu8h@redacted4", "username": "redacted4", "external\_host": "ei-postgres.eastasia.cloudapp.azure.com", "instance\_name": "postgresql", "label": "postgresql", "name": "postgresql", "plan": "Shared", "provider": **null**, "syslog\_drain\_url": **null**
  - b**: "password": "redacted"
  - c**: "username": "redacted4"
  - d**: "external\_host": "ei-postgres.eastasia.cloudapp.azure.com"

**Step 8:** Start to restore PostgreSQL data, open the terminal and enter to your PostgreSQL tool path, for example, <INSTALLATION\_PATH>\PostgreSQL\11\bin\

Run the following commands with the SQL that adjusted and give a password to restore 3 schema data only.

```
1. psql.exe -h DATABASE_HOST -U DATABASE_USERNAME -d DATABASE_NAME -f d:\public_schema.sql
2. psql.exe -h DATABASE_HOST -U DATABASE_USERNAME -d DATABASE_NAME -f d:\deviceon_schema.sql
3. psql.exe -h DATABASE_HOST -U DATABASE_USERNAME -d DATABASE_NAME -
   f d:\provisioning_schema.sql
```

**Step 9:** For MongoDB backup and restore, you could get the credential on application's environment, and start to run below command to dump collection.

```
1. mongodump.exe --host DATABASE_HOST --db DATABASE_NAME --collection COLLECTION_NAME --  
out d:\mongodb --username DATABASE_USERNAME --password DATABASE_PASSWORD
```

Run the following commands to restore collection to new database.

```
1. mongorestore.exe --host DATABASE_HOST --db DATABASE_NAME --  
collection COLLECTION_NAME D:\mongodb\COLLECTION_NAME.bson --username DATABASE_USERNAME --  
password DATABASE_PASSWORD
```

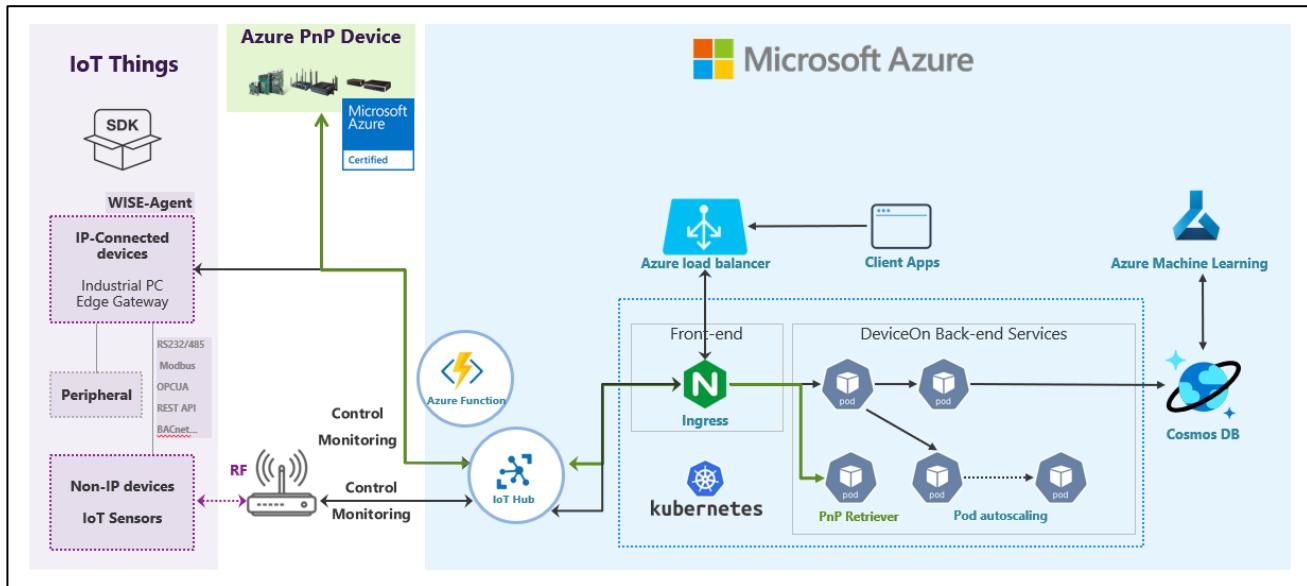
## 6.13 How Does DeviceOn Interact with AI and Machine Learning

Before to realize the value of data, to export a precise model on your field side, you must collect these raw data from edge side through WISE-Agent. The WISE-Agent not only IPC management but data acquisition for various wire/wireless protocols. DeviceOn could deploy on Azure Kubernetes to leverage Azure PaaS resource, such as Azure Function, IoTHub, Cosmos DB, meanwhile, much easier to start training via Azure Machine Learning.

Leverage Azure Machine Learning, automated ML is the process of automating the time consuming, iterative tasks of machine learning model development. It allows data scientists, analysts, and developers to build ML models with high scale, efficiency, and productivity all while sustaining model quality. Automated ML is based on a breakthrough from our Microsoft Research division.

Traditional machine learning model development is resource-intensive, requiring significant domain knowledge and time to produce and compare dozens of models. Apply automated ML when you want Azure Machine Learning to train and tune a model for you using the target metric you specify. The service then iterates through ML algorithms paired with feature selections, where each iteration produces a model with a training score. The higher the score, the better the model is considered to "fit" your data.

With automated machine learning, you'll accelerate the time it takes to get production-ready ML models with great ease and efficiency.



## 6.14 How to Enable Data Retention on DeviceOn

The device's raw data, such as hardware information, voltage, FAN, network or the wireless sensor data are stored into MongoDB. If ran out of disk storage, the MongoDB service would be stopped. To avoid this situation, you could set up the retention size of each collection on MongoDB via the API or MongoDB command, if the collection existed. Second method, you can adjust the configuration (Server\_Config.xml) to enable retention for all collections after the DeviceOn server installed. However, the second method will affect newly created collections only.

- Collection existed:

Please refer to API document to convert collection to capped, the command takes a time (depend on your collection size) to process in the background.

```
/rmm/v1/db/nosql/mongo/convertToCapped
```

**Test API**

URL-Path: <http://aimb-584:8080/rmm/v1/db/nosql/mongo/convertToCapped>

Content-Type: application/json ▾

Accept-Type: application/json ▾

Request-Body:

```
{
  "pluginName": "ProcessMonitor",
  "maxSize": 500
}
```

Response-Body:

```
{
  "result": true
}
```

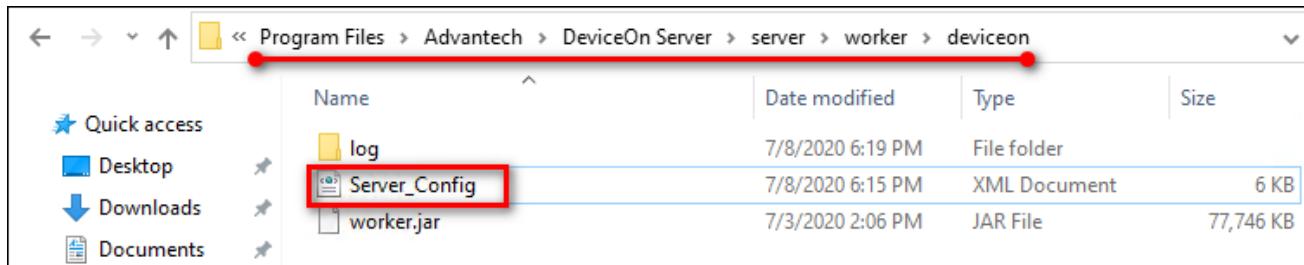
MongoDB Command line:

```
1. db.runCommand({ "convertToCapped": "common_ProcessMonitor", size: 524288000});
```

- Collection non-existed (Apply to newly created):

Please add the “Capped” item into Server\_config.xml that locate on:

\DeviceOn Server\server\worker\deviceon\



The unit of size is MB, that's mean the maximum size (uncompressed) is limited, and then restart the Management service.

```
<NoSQL>
  <MongoDB>
    <IP>127.0.0.1</IP>
    <Port>27017</Port>
    <username>[REDACTED]</username>
    <Password>[REDACTED]</Password>
    <DBName>[REDACTED]</DBName>
    <ssl>false</ssl>
    <Capped>
      <Enable>true</Enable>
      <Size>10240</Size>
    </Capped>
  </MongoDB>
</NoSQL>
```

After that, you could check and confirm the collection applied through third party tool (Rob 3T).

Key	Value
(1)	{ 13 fields }
ns	WISE-PaaS.common_ProcessMonitor
size	1827645
count	10495
avgObjSize	174
storageSize	245760
capped	false
wiredTiger	{ 14 fields }
nindexes	2
indexBuilds	[ 0 elements ]
totalIndexSize	573440
indexSizes	{ 2 fields }
scaleFactor	1
ok	1.0

Key	Value
(1)	{ 17 fields }
ns	WISE-PaaS.common_ProcessMonitor
size	1827645
count	10495
avgObjSize	174
storageSize	4096
capped	true
max	-1
maxSize	524288000
sleepCount	0
sleepMS	0
wiredTiger	{ 14 fields }
nindexes	2
indexBuilds	[ 0 elements ]
totalIndexSize	266240
indexSizes	{ 2 fields }
scaleFactor	1
ok	1.0

## 6.15 How to Enable HTTPS on DeviceOn Web Service

Generate Let's Encrypt certificate using Certbot for DeviceOn.

- [Let's Encrypt](#) is a new free, automated, and open source, Certificate Authority.
- [Certbot](#) is a console based certificate generation tool for Let's Encrypt.

In this recipe, we will generate a Let's Encrypt certificate using Certbot. This certificate will then be deployed for use in the DeviceOn server.

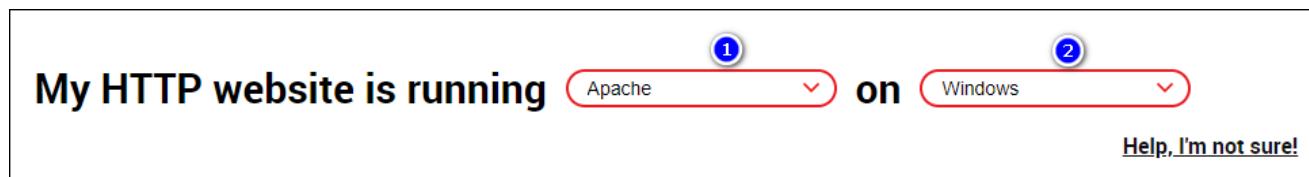
Dependencies:

- Port 443 for https needs to be open and available at time of executing certbot.
- Certbot needs root access while executing because only root is allowed to bind to any port below 1024.
- We will be using our own domain myminio.com as an example in this recipe. Replace with your own domain under your setup.

### Step 1: Install Certbot

Install Certbot by following the documentation at <https://certbot.eff.org/>

Since the DeviceOn Web service is running on Apache Tomcat, please select to "Apache" and "Windows" to download Certbot installer.



Scrolling down the instruction, you may get the installer package on Step 4.

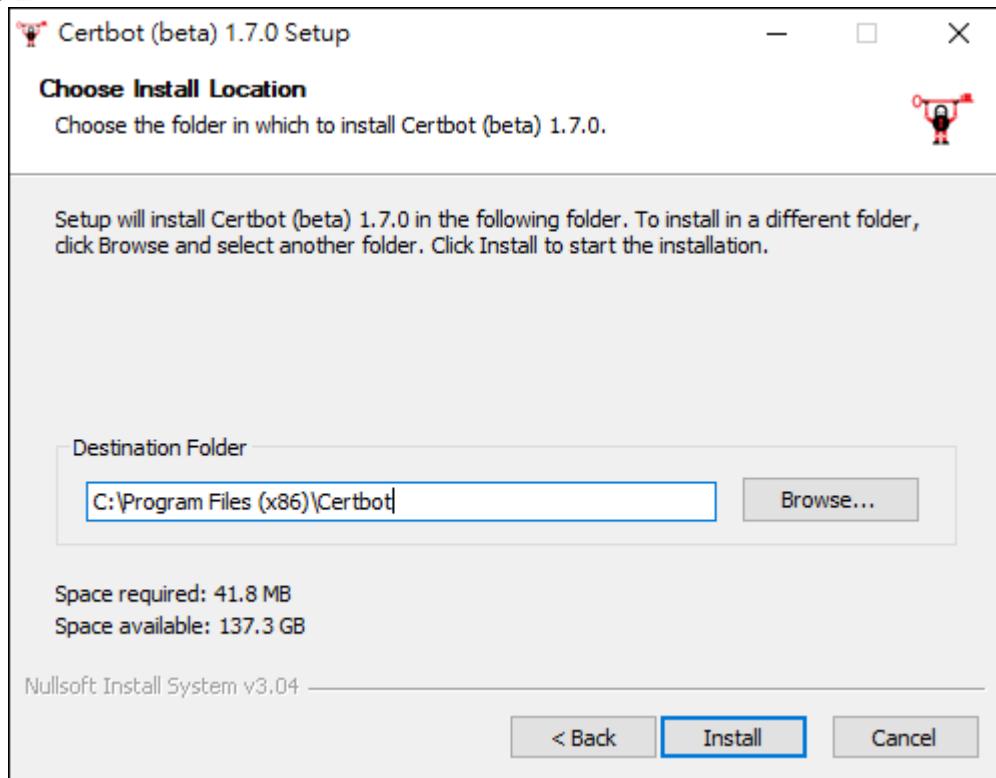
<https://dl.eff.org/certbot-beta-installer-win32.exe>



The image shows a screenshot of a web page with a white background. The title "4. Installation instructions (default)" is centered at the top in bold black font. Below it is a numbered list of steps:

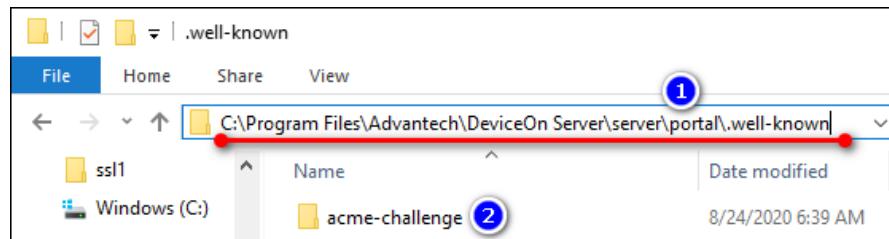
1. Connect to the server.
2. Connect locally or remotely (using Remote Desktop) to the server using an account that has administrative privileges for this machine.
3. Install Certbot.
4. Download the latest version of the Certbot installer for Windows at <https://dl.eff.org/certbot-beta-installer-win32.exe>.
5. Run the installer and follow the wizard. The installer will propose a default installation directory, `C:\Program Files(x86)`, that can be customized.)

Run the installer and follow the wizard. The installer will propose a default installation directory, `C:\Program Files(x86)`, that can be customized.)



### Step 2: Create the folder to authenticate

Create the folder named **acme-challenge** under **<DeviceOn Folder>\server\portal\.well-known**. If the folder “**.well-known**” is not exist, please create it manually.



### Step 3: Choose how you'd like to run Certbot

Run the following command to create credential files and enter your website information. The domain name(s) should input yours and the webroot to (**\DeviceOn Path\server\portal\**)

1. certbot.exe certonly --webroot

```

Administrator: C:\Windows\System32\cmd.exe
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

C:\Program Files <x86>\Certbot\bin>certbot.exe certonly --webroot
Saving debug log to C:\Certbot\log\letsencrypt.log
Plugins selected: Authenticator webroot, Installer None
Enter email address (used for urgent renewal and security notices)
  (Enter 'c' to cancel): @advantech.com.tw ②

Please read the Terms of Service at
https://letsencrypt.org/documents/LE-SA-v1.2-November-15-2017.pdf. You must
agree in order to register with the ACME server at
https://acme-v02.api.letsencrypt.org/directory
(A)gree/(C)ancel: A ③

Would you be willing, once your first certificate is successfully issued, to
share your email address with the Electronic Frontier Foundation, a founding
partner of the Let's Encrypt project and the non-profit organization that
develops Certbot? We'd like to send you email about our work encrypting the web,
EFF news, campaigns, and ways to support digital freedom.
(Y)es/(N)o: Y ④

Please enter in your domain name(s) (comma and/or space separated) (Enter 'c'
to cancel): .wise-paas.com ⑤

Obtaining a new certificate
Performing the following challenges:
http-01 challenge for .wise-paas.com
Input the webroot for .wise-paas.com: (Enter 'c' to cancel): C:\Program
Files\Advantech\DeviceOn Server\server\portal ⑥
Waiting for verification...
Cleaning up challenges
Subscribe to the EFF mailing list (email: @advantech.com.tw).
--[in

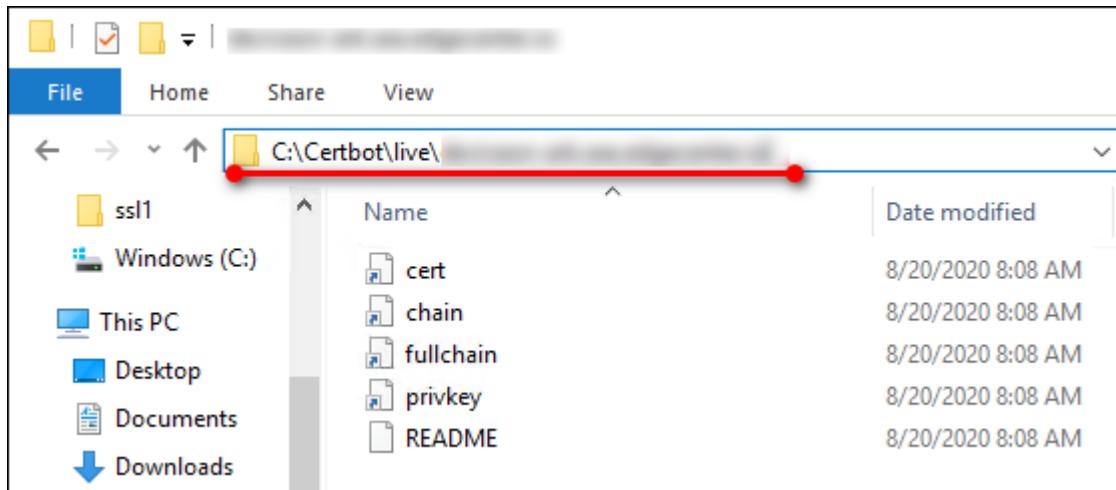
IMPORTANT NOTES:
- Congratulations! Your certificate and chain have been saved at:
  C:\Certbot\live\.wise-paas.com\fullchain.pem
  Your key file has been saved at:
  C:\Certbot\live\.wise-paas.com\privkey.pem
  Your cert will expire on 2020-11-22. To obtain a new or tweaked
  version of this certificate in the future, simply run certbot
  again. To non-interactively renew *all* of your certificates, run
  "certbot renew"
- Your account credentials have been saved in your Certbot
  configuration directory at C:\Certbot. You should make a secure
  backup of this folder now. This configuration directory will also
  contain certificates and private keys obtained by Certbot so making
  regular backups of this folder is ideal.
- If you like Certbot, please consider supporting our work by:
  Donating to ISRG / Let's Encrypt: https://letsencrypt.org/donate
  Donating to EFF: https://eff.org/donate-le
- We were unable to subscribe you to the EFF mailing list because your
  e-mail address appears to be invalid. You can try again later by
  visiting https://act.eff.org.

C:\Program Files <x86>\Certbot\bin>

```

#### Step 4: Install your certificate

You'll need to install your new certificate in the configuration file or interface for your webserver. Certificates are located in `C:\Certbot\live\[certificate_name]`, where `[certificate_name]` is the name of your certificate (usually the first domain if the `--cert-name` flag has not been used on the certonly command)



### Step 5: Install your certificate on DeviceOn Web Services

- Open the **server.xml** on text editor tool that located in the **\DeviceOn Path\tomcat\conf\**
- Append the following XML attribute into **Service** tag and give your certification path (Line, 26 to 28) that generated on Step 4.

```

1. <Connector port="443"
2.     protocol="org.apache.coyote.http11.Http11AprProtocol"
3.     connectionTimeout="20000"
4.     useSendfile="false"
5.     compression="on"
6.     compressionMinSize="2048"
7.     noCompressionUserAgents="gozilla, traviata"
8.     compressableMimeType="text/html,text/xml,text/plain,text/css,text/javascript,application/javascript,application/xml,application/json"
9.     redirectPort="8443"
10.    maxThreads="150"
11.    scheme="https"
12.    secure="true"
13.    SSLEnabled="true">
14.    <UpgradeProtocol className="org.apache.coyote.http2.Http2Protocol"
15.        overheadWindowUpdateThreshold="-1"
16.        overheadDataThreshold="-1"
17.        writeTimeout="-1"
18.        streamWriteTimeout="-1"
19.        streamReadTimeout="-1"
20.        maxHeaderSize="8192"
21.        maxConcurrentStreams="300"

```

```

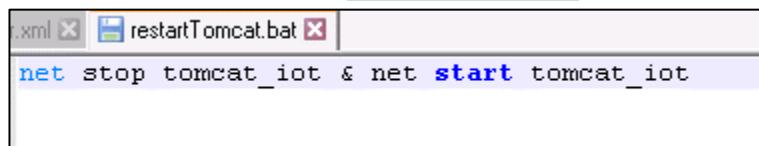
22.      readTimeout="-1"
23.      compressibleMimeType="text/html,text/xml,text/plain,text/css,text/javascript,application/javascript,application/json"
24.      compression="on" compressionMinSize="1024"/>
25.      <SSLHostConfig>
26.          <Certificate certificateKeyFile="C:\Certbot\live\<DNS>\privkey.pem"
27.              certificateFile="C:\Certbot\live\<DNS>\cert.pem"
28.              certificateChainFile="C:\Certbot\live\<DNS>\fullchain.pem"
29.              type="RSA" />
30.      </SSLHostConfig>
31.  </Connector>

```

**Step 6:** Restart DeviceOn web services (Tomcat\_IoT) to reload the configuration

**Step 7:** Enable to automatic renewal

- Create a batch file named `restartTomcat.bat` which content as below.



```
xml < restartTomcat.bat >
net stop tomcat_iot & net start tomcat_iot
```

- Copy the batch file into `C:\Certbot\renewal-hooks\post\`

**Step 8 (Optional),** Test automatic renewal, please run the following command

1. `certbot.exe renew --dry-run`

**Step 9 (Optional),** if you get all renewals succeeded, it means your configuration is correct.

```

Administrator: C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.14393]
(c) 2016 Microsoft Corporation. All rights reserved.

C:\Program Files (x86)\Certbot\bin>certbot.exe renew --dry-run
Saving debug log to C:\Certbot\log\letsencrypt.log

----- Processing C:\Certbot\renewal\*.conf -----
Cert not due for renewal, but simulating renewal for dry run
Plugins selected: Authenticator webroot, Installer None
Renewing an existing certificate
Performing the following challenges:
http-01 challenge for *
Waiting for verification...
Cleaning up challenges

----- new certificate deployed without reload, fullchain is
C:\Certbot\live\*.fullchain.pem

----- ** DRY RUN: simulating 'certbot renew' close to cert expiry
**          (The test certificates below have not been saved.)

Congratulations, all renewals succeeded. The following certs have been renewed:
  C:\Certbot\live\*.fullchain.pem (success)
* DRY RUN: simulating 'certbot renew' close to cert expiry
**          (The test certificates above have not been saved.)

Running post-hook command: C:\Certbot\renewal-hooks\post\restartTomcat.bat
Output from post-hook command restartTomcat.bat:

C:\Program Files (x86)\Certbot\bin>

```

## 7. Reference

### 7.1 User Permission

Item	Action	Description	Root	System Admin	Device Admin
Account Management	Create	Create Account	✓ (Not Include Self)	✓ (Only Device Admin)	
	Edit	Edit Account Basic Information	✓	✓ (Only Self & Device Admin)	✓ (Only Self)
	Edit	Edit Account Role	✓ (Not Include Self)		
	Edit	Disable Account	✓ (Not Include)	✓ (Only Device Admin)	

			Self)		
<b>Device Group Management</b>	View	View Account Information	✓	✓ (Only Self & Device Admin)	✓ (Only Self)
	Create	Create Device Group	✓	✓ (Only Self & Device Admin)	✓ (Only Self)
	Edit	Edit Device Group Information	✓	✓ (Only Self & Device Admin)	✓ (Only Self)
	View	View Device Group Information	✓	✓ (Only Self & Device Admin)	✓ (Only Self)
	Delete	Delete Device Group	✓	✓ (Only Self & Device Admin)	✓ (Only Self)
<b>Device Control &amp; Management</b>	Add	Add Unmanaged Device	✓	✓	✓
	Edit	Edit Device Information	✓	✓ (Only Self & Device Admin)	✓ (Only Self-Managed Devices)
	View	View Device Information	✓	✓ (Only Self & Device Admin)	✓ (Only Self-Managed Devices)
	Edit	Remove Device	✓	✓ (Only Self & Device Admin)	✓ (Only Self-Managed Devices)
	View	Search Unmanaged Devices	✓	✓	✓
	Control	Power, Remote Desktop, Terminal, Screenshot, Backup/Recovery, Protection, Windows Lockdown Actions...	✓	✓ (Only Self & Device Admin)	✓ (Only Self-Managed Devices)
<b>Event Log Management</b>	View	View and Export Device Event	✓	✓ (Only Self-Managed &	✓ (Only Self-Managed

<b>OTA Management</b>				Device Admin Devices)	Devices)
	View	View and Export System Event	✓	✓	
	View	View and Export Operation Event	✓	✓ (Only Self-Managed & Device Admin Devices)	✓ (Only Self-Managed Devices)
	View	Long-polling, Web-Socket	✓	✓	✓
	Create	Create Storage Repository	✓	✓	
	Edit	Edit Storage Repository	✓	✓	
	View	View Storage Repository	✓	✓	✓
	Delete	Delete Storage Repository	✓	✓	
	Upload	Upload OTA Package	✓ (Only Self)	✓ (Only Self)	✓ (Only Self)
	View	View OTA Package	✓ (Only Self)	✓ (Only Self)	✓ (Only Self)
<b>System Setting Management</b>	Delete	Delete OTA Package	✓ (Only Self)	✓ (Only Self)	✓ (Only Self)
	Deploy	Deploy OTA Package	✓	✓ (Only Self-Managed & Device Admin Devices)	✓ (Only Self-Managed Devices)
	Edit	Edit OTA Deploy Configuration	✓	✓	
	Create	Create an Action	✓ (All Groups) on Self Account	✓ (Only Self-Groups & Device Admin Groups) on Self Account	✓ (Only Self-Groups)
	Edit	Update an Action	✓ (All)	✓ (Only Self-)	✓ (Only Self-)

		Groups) on Self Account	Groups & Device Admin Groups) on Self Account	Groups)
View	View Action	✓ Self Account	✓ Self Account	✓ Self Account
Delete	Delete Action	✓ Self Account	✓ Self Account	✓ Self Account
Provisioning	Power Management	✓	✓ (Only Self-Managed & Device Admin Devices)	✓ (Only Self-Managed Devices)
	Backup/Recovery	✓	✓ (Only Self-Managed & Device Admin Devices)	✓ (Only Self-Managed Devices)
	Protection	✓	✓ (Only Self-Managed & Device Admin Devices)	✓ (Only Self-Managed Devices)
Edit	Edit Event Alert Setting	✓ (Only Self)	✓ (Only Self)	✓ (Only Self)
Edit	Configure Alert Service	✓	✓	
Create	Create Rule Engine	✓	✓ (Only Self & Device Admin)	✓ (Only Self-Managed Devices)
Update	Edit Rule Engine	✓	✓ (Only Self & Device Admin)	✓ (Only Self-Managed Devices)
View	View Rule Engine	✓	✓ (Only Self & Device Admin)	✓ (Only Self-Managed Devices)
Delete	Delete Rule Engine	✓	✓ (Only Self & Device Admin)	✓ (Only Self-Managed Devices)
Edit/View	Edit/View System UI	✓	✓	
Edit	Activate	✓	✓	

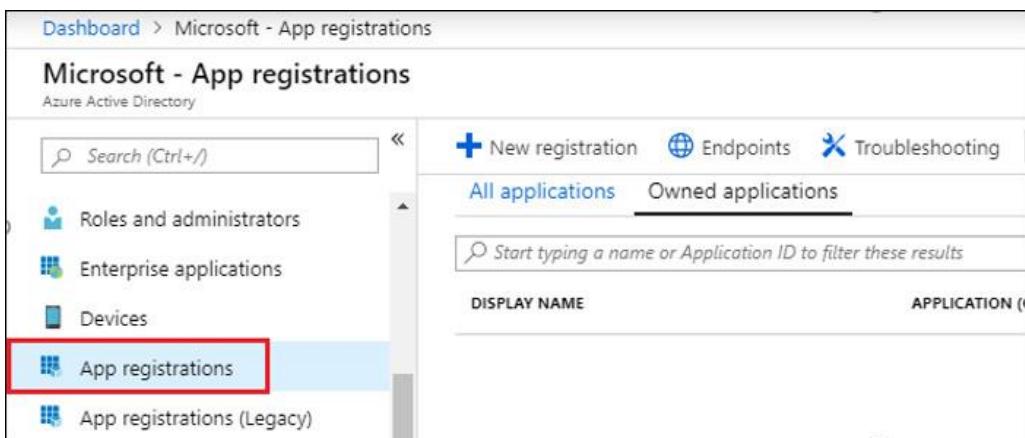
	DeviceOn License (Perpetual Only)		
--	--------------------------------------	--	--

## 7.2 Retrieve My Azure Account Information

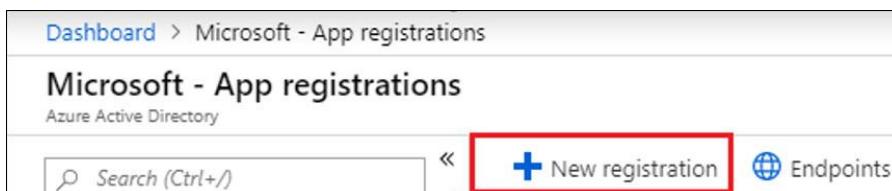
### 7.2.1 Method 1 – Create & Get Information on Azure Portal

#### Step 1: Create Your Application

- 1.1. Log into your [Azure Portal](#)
- 1.2. Select [Azure Active Directory]
- 1.3. Select [App registrations]



- 1.4. Add [New Registration]



- 1.5. Setup your **Application Name** then click [**Register**].
  - Enter your Application display name in **Name** field.
  - Setup **Supported account types** by selecting the respective account type for this API.
  - Under **Redirect URI**, select Web for the type of application you want to create. Enter the URI where the access token is sent to.

Note: You cannot create a Native application credential nor use the type for an automated application.

Dashboard > Microsoft - App registrations > Register an application

## Register an application

**⚠ If you are building an application for external users that will be distributed by Microsoft, you must register as a first party application to meet all security, privacy, and compliance policies. [Read our decision guide](#)**

**\* Name**  
The user-facing display name for this application (this can be changed later).  
 

**Supported account types**  
Who can use this application or access this API?  
 Accounts in this organizational directory only (Microsoft)  
 Accounts in any organizational directory  
 Accounts in any organizational directory and personal Microsoft accounts (e.g. Skype, Xbox, Outlook.com)  
[Help me choose...](#)

**Redirect URI (optional)**  
We'll return the authentication response to this URI after successfully authenticating the user. Providing this now is optional and it can be changed later, but a value is required for most authentication scenarios.  
   

By proceeding, you agree to the [Microsoft Platform Policies](#)

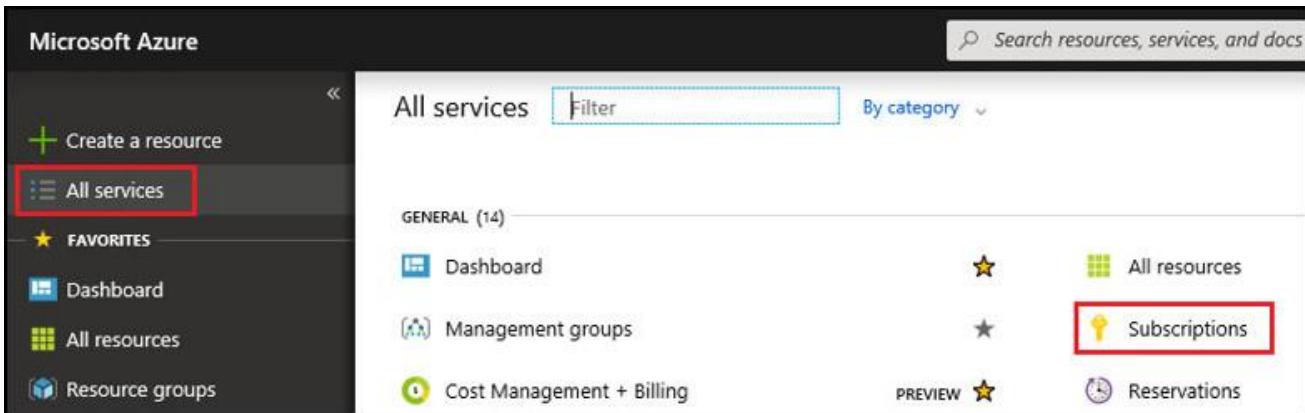
**Register**

### Step 2: Get Subscription ID

To access resources in your subscription, you must assign a role to the Application. You can pick between Subscription, Resource Group or Resource. Permissions are inherited to lower scope levels.

[For more details, see RBAC: Built in Roles](#)

2.1. Select **All services** then select **Subscriptions** to set up the level of scope you wish to assign this application.



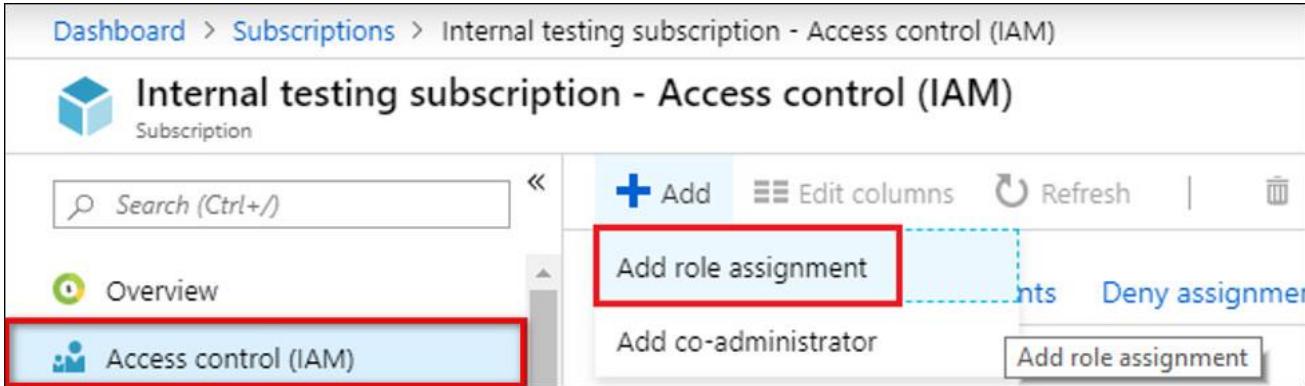
The screenshot shows the Microsoft Azure portal's main dashboard. On the left sidebar, 'All services' is selected and highlighted with a red box. The main content area displays a grid of service tiles under the 'GENERAL (14)' category. The 'Subscriptions' tile, which contains a yellow key icon, is also highlighted with a red box. Other visible tiles include 'Dashboard', 'Management groups', 'Cost Management + Billing', 'All resources', and 'Reservations'.

2.2. Find the Subscription you would like to assign to the Application created in the Step 1. Copy the **Subscription ID**, as this is one of the Azure data fields required on the WISE-PaaS Marketplace later. ([Ref: Marketplace field #A](#))

SUBSCRIPTION	SUBSCRIPTION ID
 Visual Studio Enterprise – MPN	[REDACTED]

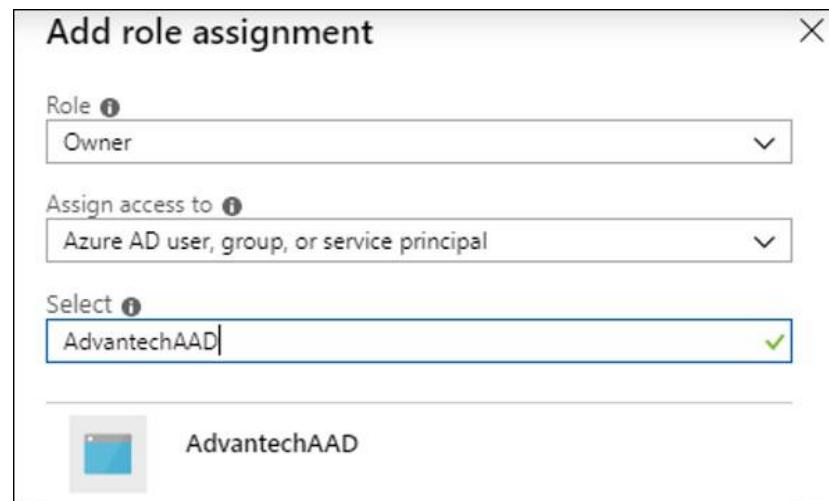
**! Troubleshoot:** If you do not see the subscription you're looking for, select global subscriptions filter. Make sure the subscription you want is selected for the portal.

2.3. Select Access control (IAM) then Add role assignment



The screenshot shows the 'Internal testing subscription - Access control (IAM)' page. In the left sidebar, the 'Access control (IAM)' link is highlighted with a red box. At the top right, the 'Add role assignment' button is also highlighted with a red box. Other buttons visible in the top right include 'Edit columns', 'Refresh', and 'Deny assignment'. A search bar and an 'Overview' link are also present in the interface.

2.4. Select the **Owner** role. By default, Azure AD applications are not displayed in the available options. To find your application, search for the name.



2.5. Click **Save** to finish assigning the role. You will be able to see your application in the list of users assigned to a role for that scope.

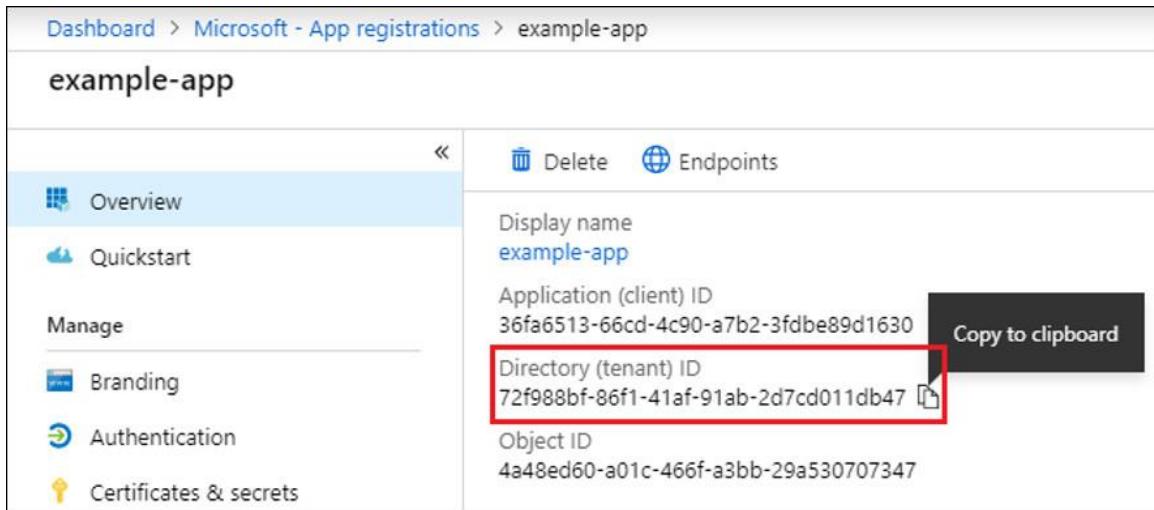
### Step 3: Get Application & Tenant ID

#### 3.1. Select Azure Active Directory

#### 3.2. From App registrations in Azure AD, select your application

DISPLAY NAME	APPLICATION (CLIENT) ID
example-app	

3.3. Copy the **Directory (tenant) ID** as another piece of Azure information that will be required on the WISE-PaaS Marketplace later. ([Ref: Marketplace field #C](#))

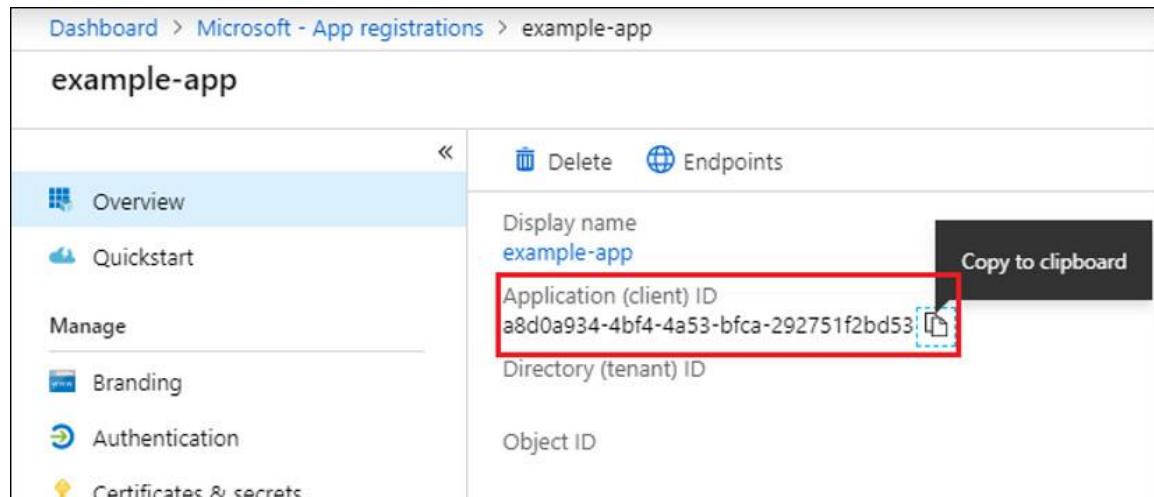


Dashboard > Microsoft - App registrations > example-app

## example-app

Overview	Delete	Endpoints
Display name example-app		
Application (client) ID 36fa6513-66cd-4c90-a7b2-3fdbe89d1630		<b>Copy to clipboard</b>
Directory (tenant) ID 72f988bf-86f1-41af-91ab-2d7cd011db47		
Object ID 4a48ed60-a01c-466f-a3bb-29a530707347		

- 3.4. Copy the **Application (client) ID** as part of Azure information that will be required on the WISE-PaaS Marketplace later. ([Ref: Marketplace field #B](#))



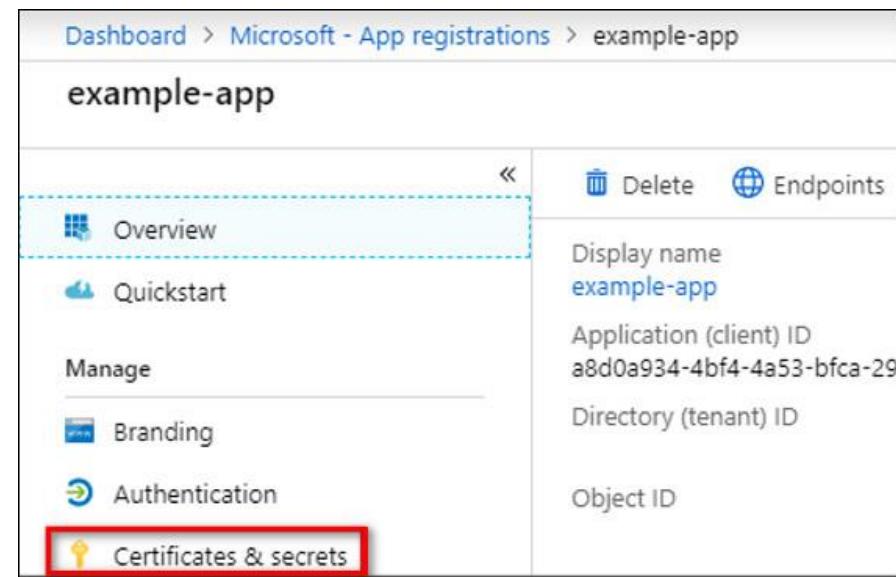
Dashboard > Microsoft - App registrations > example-app

## example-app

Overview	Delete	Endpoints
Display name example-app		<b>Copy to clipboard</b>
Application (client) ID a8d0a934-4bf4-4a53-bfca-292751f2bd53		
Directory (tenant) ID		
Object ID		

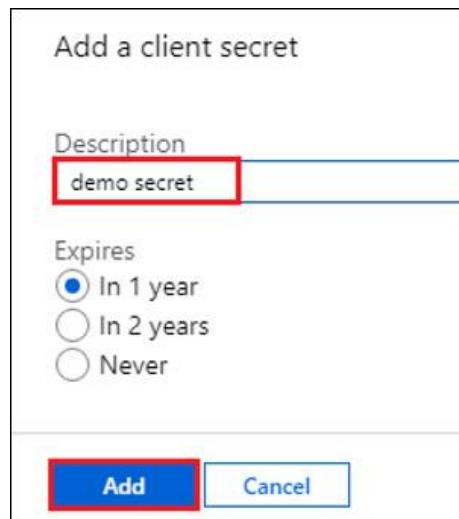
#### Step 4: Add & Get Client Secret

- 4.1. Select [**Certificates & secrets**]
- 4.2. Select **Client secrets** then **New client secret**



The screenshot shows the Microsoft Azure portal interface. At the top, it says "Dashboard > Microsoft - App registrations > example-app". Below this, the app name "example-app" is displayed. On the left, there's a sidebar with links: Overview (selected), Quickstart, Manage, Branding, Authentication, and Certificates & secrets (highlighted with a red box). On the right, there are details for the app: Display name (example-app), Application (client) ID (a8d0a934-4bf4-4a53-bfca-29), Directory (tenant) ID, and Object ID.

4.3. Provide a description for the new client secret, set up the expiration period. Then Click [Add]



The dialog box has a title "Add a client secret". It contains a "Description" field with the value "demo secret" (highlighted with a red box). Under "Expires", there are three radio buttons: "In 1 year" (selected), "In 2 years", and "Never". At the bottom are two buttons: "Add" (highlighted with a red box) and "Cancel".

Copy Client Secret ([Ref: Marketplace field #D](#))

Client secrets		
A secret string that the application uses to prove its identity when requesting a token. Also can be referred to as application password.		
<a href="#">+ New client secret</a>		
DESCRIPTION	EXPIRES	VALUE
demo secret	5/14/2020	nWu9HVZ7Rnj.2y7XSkVvUngZ][x9Z:e 

## 7.2.2 Method 2 – Create via Azure CLI (Command-line Tool)

**Step 1:** Install Azure CLI

[For details, please view this step by step guide](#)

**Step 2:** Sign in to the Azure Account

1. C:\>az login

Note: If the CLI can open your default browser, it will do so and load a sign-in page. Otherwise, you need to open a browser page and follow the instructions on the command line to enter an authorization code after navigating to <https://aka.ms/devicelogin> in your browser. Sign in with your account credentials in the browser.

**Step 3:** Get Subscription ID & Copy Output

2. C:\>az account show --query id

```
C:\>az login
Note, we have launched a browser for you to login. For old experience with device code, use "az login --use-device-code"
You have logged in. Now let us find all the subscriptions to which you have access...
[{"cloudName": "AzureCloud",
"id": "00000000-0000-0000-0000-000000000000",
"isDefault": true,
"name": "Visual Studio Enterprise \u2013 MPN",
"state": "Enabled",
"tenantId": "00000000-0000-0000-0000-000000000000",
"user": {
"name": "John.Doe@contoso.com",
"type": "user"
}
}]
```

**Step 4:** Create service principal and get Application ID, Tenant ID and Client Secret

3. C:\>az ad sp create-for-rbac --name ServicePrincipalName

```
C:\>az ad sp create-for-rbac --name AdvantechAD
Retrying role assignment creation: 1/36
Retrying role assignment creation: 2/36
{
  "appId": "22222222-2222-2222-2222-222222222222",
  "displayName": "AdvantechAD",
  "name": "http://AdvantechAD",
  "password": "XXXXXXXXXXXXXX",
  "tenant": "00000000-0000-0000-0000-000000000000"
}
```

Reference: [Create an Azure service principal with Azure CLI >](#)