

**Azure Sentinel – Create Azure IoT Lab Environment**

Table of Contents

1. Overview
2. Getting Started
   * Prerequisites
   * Installation
3. Usage
4. Contact

**Overview**

Azure IoT lab environment is designed to provide Azure Sentinel Customer a robust and dynamic lab environment to perform Red/Blue team exercise in order to evaluate Azure Sentinel capabilities for IoT monitoring. The lab can also be used for continuous testing and training platform to test new attacks and remediation automations.

The lab consists of docker containers running a .net core simulator app which communicates to Azure IoT hub service using MTTQ over WebSocket simulating a real world IoT solution. The Docker containers are assigned Local-LAN dynamic (DHCP) IP address using a network created with the '[transparent' Network driver](https://docs.microsoft.com/en-us/virtualization/windowscontainers/container-networking/network-drivers-topologies#transparent-network-driver) available via Hyper-V on Windows Servers, enabling the containers to be directly access on Local-LAN and having internet access via the Hyper-V external switch.

The docker container simulates an actual IoT Device by simulating Application, Base OS, and network layers. Defender for IoT will treat each docker container as if it’s new stand-a-lone IoT device on the network.

The lab environment could easily be extended by adding a Kali-Linux VM on the same external Hyper-V switch to launch attacks and perform blue team exercise.

Graphical user interface, diagram

Description automatically generated

**Getting Started**

Use the following steps to correctly configure Azure Sentinel IoT Lab environment.

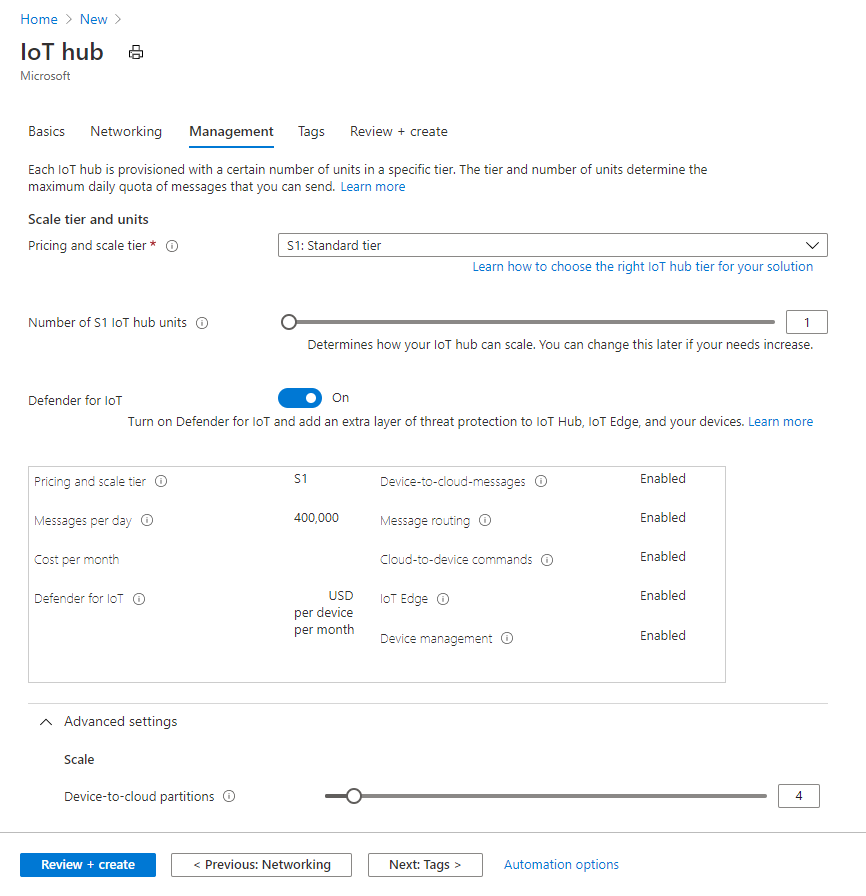
**Prerequisites**

* Hyper-V enable local host machine (Windows 10 or Server 2016 and above)
* Internet connectivity

**Installation**

**Azure IoT Hub** Configuration

1. Follow Microsoft Documentation on how to create IoT Hub [Use the Azure portal to create an IoT Hub | Microsoft Docs](https://docs.microsoft.com/en-us/azure/iot-hub/iot-hub-create-through-portal)
2. Make sure the Pricing and Scale Tier is set to “**S1: Standard Tier**” and **Defender for IoT** is turned on.



**Docker Host VM** Configuration

1. Create new Server 2016 or above VM with a new external network switch. (**Note:** Do not use Default switch as it does not support NIC Spoofing)
2. **Before turning on the VM:**
   1. Run the following PowerShell statement to enable nested virtualization

Set-VMProcessor -VMName <VMName> -ExposeVirtualizationExtensions $true

* 1. Configure MAC Spoofing and Port Mirroring (set as Source)

Graphical user interface, text, application

Description automatically generated

1. Turn on VM and install Windows and update.
2. **Turn Off Windows Firewall**

Set-NetFirewallProfile -Profile Domain,Public,Private -Enabled False

1. **Install Az PowerShell Modules**

Install-Module Az -Force

1. **Install Windows Container Features on VM**

Install-WindowsFeature -Name Hyper-V -IncludeManagementTools

Install-WindowsFeature -Name Containers

Restart-Computer

1. **Install Docker**

Install-Module DockerProvider -force

Install-Package Docker -ProviderName DockerProvider -RequiredVersion preview -force

Restart-Computer

1. **Create Docker Transparent Network called DockerNet**

docker network create -d transparent --subnet=192.168.1.0/24 --gateway=192.168.1.1 DockerNet

**Deploy Docker Containers**

1. Download ‘TemperatureSensorSimulator-Content.zip’ and extract
2. Run the ‘DeployIoTSimulator.ps1’ script
3. Enter the count of IoT simulators to deploy
4. Once the script finishes it will display all the containers deployed with their IP address.

Text

Description automatically generated

1. You can verify that the new simulator devices are communicating to Azure IoT Hub.

Graphical user interface

Description automatically generated

**Clean up**

1. Run the ‘CleanUp.ps1’ script to delete all IoT simulator devices.

**Contact**

Email: [MicrosoftTeam@criticalstart.com](mailto:MicrosoftTeam@criticalstart.com)

**About**

Add Something Here