New comer guide for LXCI SCOM HWMP

1. Apply ICE Firewall Account
   1. Before you can pull code from our GIT repository, you must apply an ICE lab firewall account. Please follow instructions in below link.  
      <http://cowork.us.lenovo.com/departments/ebg_development_lab_tooling/Pages/Requesting-ICE-lab-access.aspx>
   2. After your account is approved, please ask Zhang Yu ([zhangyu38@lenovo.com](mailto:zhangyu38@lenovo.com)) to give you access right to our projects.
2. Pull code from GIT repository
   1. Login ICE lab firewall: <http://10.240.35.120>
   2. Login ICE lab GIT site: <https://git.icelab.lenovo.com/users/sign_in>
   3. Browsing into projects you can find corresponding SSH links to pull code with GIT tool.
   4. The projects for LXCI for SCOM HWMP are “xClarity/scom\_mp” and “xClarity/UnifiedService”, the main branches are ‘scom\_mp\_xxxx’ and ‘uus\_develop’ respectively. Here the xxxx is the release version.
3. Jenkins

LXCI daily builds can be downloaded at our Jenkins site: <http://10.240.196.207>. You should only use builds that pass BVT for feature development and defect verification.

You need to contact Sain Wu ([wusz2@lenovo.com](mailto:wusz2@lenovo.com)) to give you Jenkins access right.

1. Bugizlla

We keep track of defects at our Bugizlla site: <https://bz.labs.lenovo.com>. Product: SCOM; Component: scom\_mp

Below is the typical workflow when a defect is assigned to you:

* 1. Study the defect. Put the defect into “**Working**” state.
  2. If the defect is misreported, put it into “**Rejected**” state and input explanations in comment section. Otherwise fix it, commit code, and put the defect into “**Fixed**” state.
  3. Once a new BVT pass build is available, deploy one and verify whether the defect is truly fixed, then put the defect into “**Verified**” state and update the “**deadline**” field.
  4. During the process you fix a defect, there may be blocking issues or you need additional information from reporter. You should write down your findings in comment section and update the “**Action**” field.

1. Jira

We used Jira to keep track of our Agile Development Process. You can access our Jira site at <http://jira1.labs.lenovo.com:8080>. Our project is “LXCI SCOM”.

You need to ask Emma JL (linjing5@lenovo.com) to give you Jira access right to our projects.

1. Automation Test
   1. Register XRover web page http://10.240.196.222:3000 with Email address. Choose a group which you’re working on, such as SCVMM, SCOM or IVP
   2. Register Jenkins http://10.240.196.207:8080. Ping zhangqian18/wusz2 to grant you a proper permission.
   3. Access XRover web page. Choose product tab and release tab. Then all test suites are listed below. Usually there are Install, BVT,FVT and Password
   4. Click test suite icon to see total result of the latest build. Click pass/fail number to see pass/fail cases only. Click History button to see last 20 builds result. Click build number under History icon to see one build’s result.
   5. For Install/BVT failed cases, there are usually two image links in Fail Detail column. FindFailed means what we are looking for and Screen captured means what we find.
   6. In case list page, click Show button of History column. Last 10 builds’ results are shown in 10 squares. Green means pass and red means fail. Yellow square means there is comment. You could see the comment when hover on this square.
   7. Click Perf button of Perf column to see performance status of this case. X axis is time. Left Y axis is step. Right Y axis is Env’s status, like network delay and CPU usage. Bar chart length is this step’s duration. Baseline is under real execution.
   8. In test suite list page, click Chart button of Performance column to see all cases’ duration of last 5 builds. Click Data button to download these data as a spreadsheet
2. Development Environment
   1. Before you begin, you need to deploy a SCOM server env, please refer to the “How to build a SCOM server”.
   2. The “UnifiedService” Project (we often call it UUS) is the backend service of LXCI for both VMware vCenter and Microsoft System Center. It is where the implementation of most LXCI features locates in. This project is mainly written in Python, and we recommend “Eclipse + Pydev” or “Visual Studio Code” as development environment.

For more information about this project, please refer to the “UnifiedService developer guide”.

* 1. The “scom\_mp” project contains all SCOM HWMP own code.  
     Generally the “scom\_mp” project can be divided to two parts: one is the “Management Pack” ([what is management pack](https://technet.microsoft.com/en-us/library/hh212794(v=sc.12).aspx)) which is in xml format, now there are 9 MPs totally; another part is the tools for supporting “Management Pack”, the tools can be an exe, a VB script, a Powershell script, a DLL, etc.
  2. From the architecture view, SCOM HWMP can divided to some sub-components, as below:

|  |  |  |  |
| --- | --- | --- | --- |
| Component | Management Pack | Source folder | Description |
| In-band monitor | Lenovo.HardwareMgmtPack.xSystems | \Servers\ | monitors and reports hardware health of Lenovo servers via Platform agent |
| Out-band monitor (SNMP - Chassis) | Lenovo.HardwareMgmtPack.BladeCenter.v2 Lenovo.HardwareMgmtPack.FlexSystem.v2 | \BCC\ | monitors and reports hardware health of Lenovo BladeCenter/Flex Chassis and Modules. |
| Out-band monitor (CIM – IMM) | Lenovo.HardwareMgmtPack.IMM2.v2 | \OOB\ | monitors and reports hardware health of Lenovo Integrated Management Module |
| Out-band monitor (LXCA – Thinkserver) | Lenovo.ThinkServer.BMC.Module | \OOB\ | monitors and reports hardware health of Lenovo Thinkserver BMC |
| Common components | Lenovo.HardwareMgmtPack.Common Lenovo.HardwareMgmtPack.HardwareIDs Lenovo.HardwareMgmtPack.Relation.v2 Lenovo.HardwareMgmtPack.RelationCMM.v2 | \common\ | defines common elements for managing Lenovo Hardware. defines the relationship between blade and chassis in SCOM. |

* 1. The architecture diagram of SCOM HWMP
  2. Start to know SCOM and SCOM HWMP  
     Please refer to “Lenovo HWMP L1L2 Training Material.pptx”

1. Build and Debug
   1. UnifiedService

Currently, you cannot debug the “UnifiedService" project locally. This means that you cannot build the whole project and then debug it with the Eclipse IDE / Visual Studio Code. Instead, you need to deploy a LXCI instance, replace python files (.pyc), restart UUS service, and check the log.

* 1. SCOM HWMP management pack

Currently, there is no way to debug the management packs in SCOM. Typically, we need to install the SCOM HWMP, then re-import the management pack you changed, trigger the related task/action, and check the log (in windows viewer).

1. Reference websites and resources
   1. LXCI website for latest release and user guide download:  
      <https://support.lenovo.com/us/en/solutions/lnvo-hwmp>
   2. Operations Manager Management Pack Development Kit :  
      <https://msdn.microsoft.com/en-us/library/ee533840.aspx>