HLA Workshop

Installation Guide

Anthony Angelo, Netser Heruty, Daniel Smith

5 Oct 2021

Table of Contents

# Introduction

This document provides prescriptive guidance for deploying the HLA Workshop

# Prerequisites

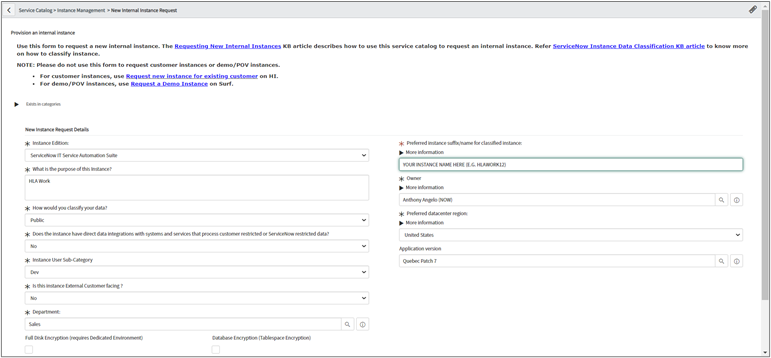
Following is a list of prerequisite tools and accesses needed to perform a full HLA Workshop installation.This document provides prescriptive guidance for using installing the HLA Workshop:

* Access to an AWS Account with full admin privileges
* AWS CLI
* Bash Terminal Access (e.g., WSL for Windows, MacOS Terminal or another Linux)
* Terraform (e.g., Terraform v0.12.31)
* Ansible (e.g., v4.5.0)

# Deploy your NOW Instance

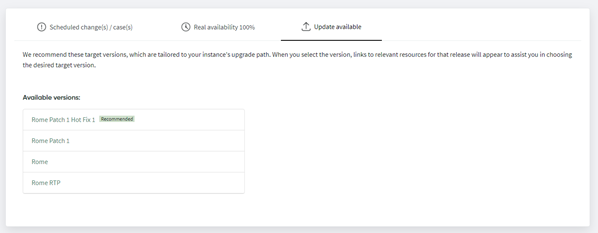
## Request a new NOW Instance

1. Navigate to [NOW HI](https://support.servicenow.com/now)
2. Search for “new internal instance request”
3. Request a new instance as follows, using the latest available application version:

* 
* Figure 1

## Upgrade your NOW Instance to latest Rome version

1. Navigate to [NOW HI](https://support.servicenow.com/now)
2. Select your Instance from the Instances Dashboard
3. Upgrade your instance to latest Rome version & patch level as follows:

* 
* Figure 2

## Install the required ITOM plugins for the Workshop

1. Login to your NOW Instance as an Administrator
2. Navigate to the **System Definition > Plugins** and install the following plugins:

|  |  |  |
| --- | --- | --- |
| * Plugin Name | * Plugin ID | * Notes |
| * Agent Client Collector Monitoring | * com.snc.sa.metric | * 10 Minutes to install |
| * Service Mapping | * com.snc.service-mapping | * 20 Minutes to install |

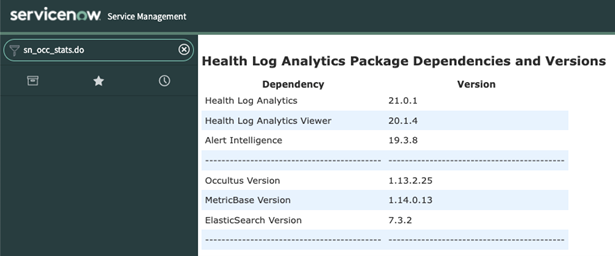
## Install the HLA stack for your NOW Instance

1. Follow the [HLA Installation Guide](https://servicenow.sharepoint.com/:w:/s/itxamssc/EZUQY9F095VEmHuA0A4Od2cBEkjRbiVo8oWMLARC5HfjVg?e=iGEcKR) for the latest installation steps

* NOTE: Please read and follow all the steps carefully as instructed in the “HLA Installation Guide” document as it us updated frequently by the HLA Development Team

## Validate the HLA stack for your NOW Instance

1. Login to your NOW Instance as an Administrator
2. Retrieve the **Health Log Analytics Package Dependencies & Versions** as follows:

* 
* Figure 3

## Optimize your NOW Instance for the Workshop

1. Login to your NOW Instance as an Administrator
2. Navigate to **Health Log Analytics > Health Log Analytics Administration > System Properties** and set the following properties:

|  |  |  |
| --- | --- | --- |
| * Property Name | * Value | * Default Value |
| * aggregator.window\_size\_seconds | * 300 | * 1800 |
| * rules.filter\_detections\_with\_low\_current\_value.threshold | * 1.0 | * 5.0 |
| * incidents.cooldown\_period\_minutes | * 2 | * 5 |

1. Navigate to **Health Log Analytics > Health Log Analytics Administration > Features** and set the following features:

|  |  |  |
| --- | --- | --- |
| * Name | * State | * Default Value |
| * Disable | * Warm Up Time Rule | * ON |
| * Disable Setup Time Rule | * ON | * OFF |
| * Disable All Events Metric Anomaly Detections Rule | * ON | * OFF |

# Deploy the Workshop AWS Environment

## Create the AWS Environment using Terraform

1. Clone Git Project

* $ git clone git@github.com:pangealab/heracles.git  
  $ cd heracles/

1. Configure AWS Profile

* $ export AWS\_PROFILE=YOUR PROFILE; printenv AWS\_PROFILE

1. Create Terraform State Bucket using your VPC ID (e.g., hlawork1)

* $ aws s3 mb s3://YOUR VPC ID-terraform-backend --profile YOUR PROFILE

1. Create SSH Key (e.g., heracles)

* $ ssh-keygen -t rsa -b 4096 -C "heracles@noreply.com" -f $HOME/.ssh/heracles -m PEM

1. Set your backend bucket property in the *backend.tf* file as follows:

* # Save Terraform State to S3 Bucket  
  terraform {  
  backend "s3" {  
   bucket = "YOURCLUSTERID-terraform-backend"  
   key = "terraform.tfstate"  
   region = "us-east-2"  
  }  
  }

1. Initialize Terraform

* $ terraform init

1. Create Infrastructure

* $ terraform apply -auto-approve -var instance\_count=3 -var cluster\_name=YOUR VPC ID

1. Safeguard the Terraform output of server public and private IPs
2. Safeguard the generated Ansible Inventory file (e.g., inventory-hlawork1.cfg)

## Install the Pet Clinic Software stack using Ansible

1. Set SSH Agent

* $ eval `ssh-agent -s`  
  $ ssh-add ~/.ssh/YOUR SSH KEY

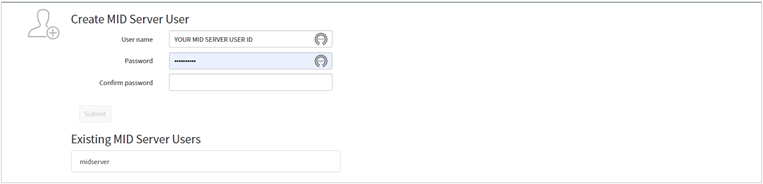
1. Run the Install Pet Clinic Playbook

* $ ansible-playbook -i YOUR INVENTORY FILE.cfg ansible/install-petclinic.yml \  
  -e "github\_token=ghp\_ZZuupQSjnYhU8tEJ5lKZ6X6cQ6AhOS0Wqi8C" \  
  -e "mysql\_host=YOUR MYSQL PRIVATE IP \  
  -e "servers='YOUR SPRING SERVERS PRIVATE IPS'" \  
  -e "frontend\_addr=YOUR NGINX PUBLIC IP:8080"

# Deploy the MID Server and pre-configure ACC, Filebeat Access

## Configure NOW MID Access

1. Login to your NOW Instance as Administrator
2. Navigate to **Guided Setup > ITOM Guided Setup**
3. Click on **MID Server**
4. Click on **Create MID User**

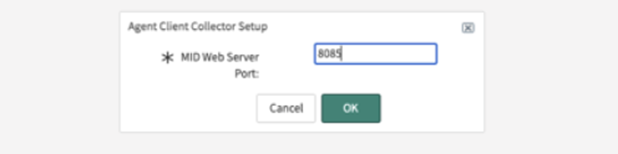
* 
* Figure 4

## Install MID Server Software using Ansible

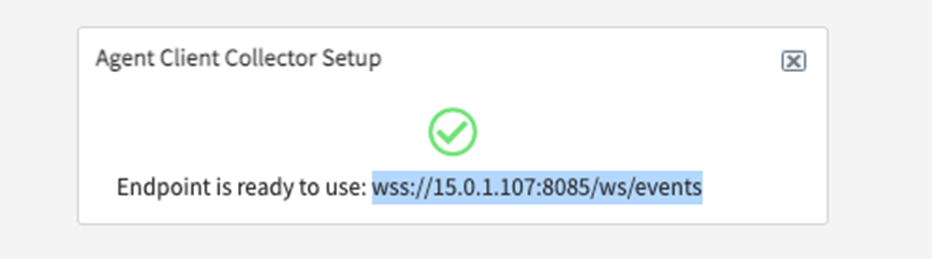
1. Run the Install MID Server Playbook $ ansible-playbook -i YOUR INVENTORY FILE.cfg ansible/install-midserver.yml \ -e "instance\_url=https://YOUR NOW URL " \ -e "mid\_userame=YOUR MID SERVER USER ID" \ -e "mid\_password=YOUR MID SERVER USER PASSWORD" ## Configure NOW ACC & Filebeat Access

* NOTE: After the Ansible script above finished running, a MID entry should appear on the “MID Servers” table in the platform

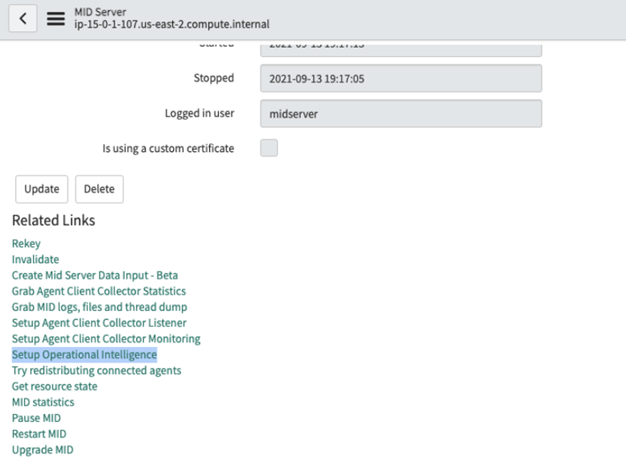
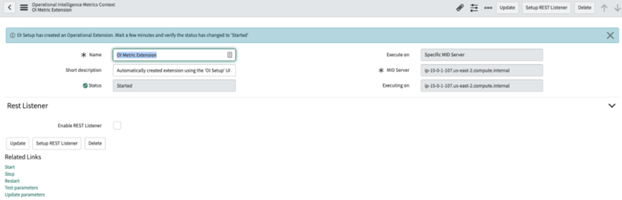
1. Select the MID record name
2. Press the **Validate** button
3. Keep the default options to “ALL” (e.g., IP Ranges, Supported Applications, etc.)
4. Click on **Setup Agent Client Collector Listener**
5. Set the MID Web Server Port to “8085” as follows:

* 
* Figure 5

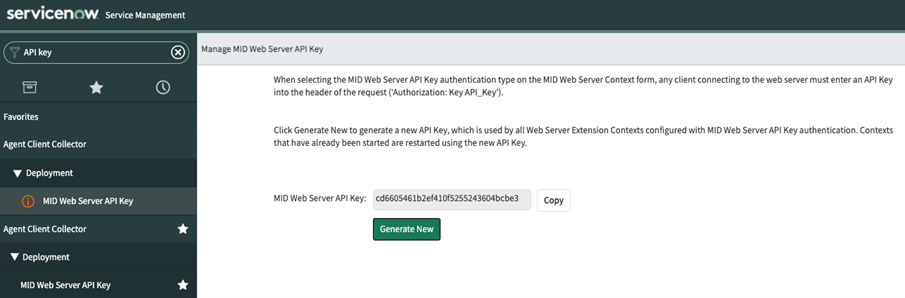
1. Safeguard your Endpoint address (e.g., wss://15.0.1.107:8085/ws/events)

* 
* Figure 6

1. Click on **Setup Operational Intelligence** as follows

* 
* Figure 7
* And you’ll be redirected to the following screen:
* 
* Figure 8
* NOTE: Make sure the status turns to “Started”

1. Navigate to **Agent Client Collector > Deployment > MID Web Server API Key**
2. Safeguard your MID’s API KEY

* 
* Figure 9

## Create your Application Service

1. Login to your NOW Instance as Administrator
2. Navigate to **CSDM > Manage Technical Services > Application Service**
3. Create a **New** Application Service as follows:

|  |  |
| --- | --- |
| * Field | * Value |
| * Name | * Pet Clinic |
| * Operational Status | * Operational |

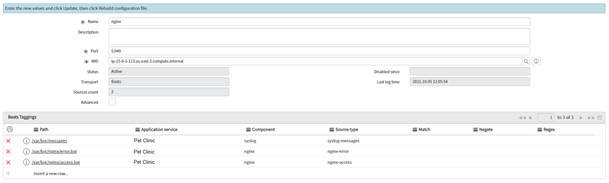
1. Press **Next** followed by **Done**

* NOTE: Do not populate the Application Service

# Create your HLA Data Inputs

## NGINX Data Input

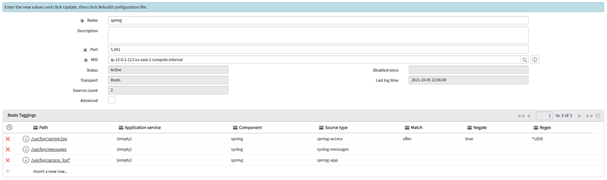
1. Login to your NOW Instance as Administrator
2. Navigate to **Health Log Analytics > Data Input**
3. Create a **Linux using Filebeat Data Input** as follows:

* 
* Figure 10

1. Press **Submit** when done > NOTE: Do not download the ”filebeat.yml” as it is part of an Ansible Playbook already

## Spring Data Input

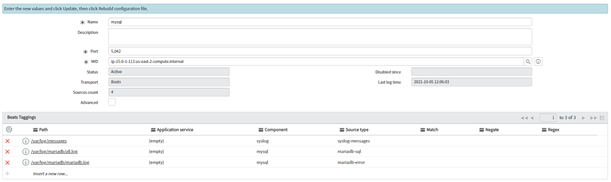
1. Login to your NOW Instance as Administrator
2. Navigate to **Health Log Analytics > Data Input**
3. Create a **Linux using Filebeat Data Input** as follows:

* 
* Figure 11

1. Press **Submit** when done > NOTE: Do not download the ”filebeat.yml” as it is part of an Ansible Playbook already

## MySQL Data Input

1. Login to your NOW Instance as Administrator
2. Navigate to **Health Log Analytics > Data Input**
3. Create a **Linux using Filebeat Data Input** as follows:

* 
* Figure 12

1. Press **Submit** when done > NOTE: Do not download the ”filebeat.yml” as it is part of an Ansible Playbook already

# Install the ACC & Filebeat Software using Ansible

1. Run the Install Agents Playbook

* $ ansible-playbook -i YOUR INVENTORY FILE ansible/install-agents.yml \  
  -e "acc\_mid=YOUR ACC MID URL" \  
  -e "acc\_api\_key=YOUR ACC API KEY" \  
  -e "nginx\_logstash=YOUR MID SERVER PRIVATE IP:5040" \  
  -e "spring\_logstash= YOUR MID SERVER PRIVATE IP:5041" \  
  -e "mysql\_logstash= YOUR MID SERVER PRIVATE IP:5042"

# Configure your NOW HLA instance for a new Workshop

1. Login to your NOW Instance as Administrator
2. Navigate to **Health Log Analytics > Mapping > Source Type Structures**
3. For each Source Type Structure, set the **Custom JS** Function using the scripts located in the cloned Git Project /servicenow folder (e.g., source-type-structure-mariadb-error.js)

# Reconfigure an existing NOW HLA Instance for a new Workshop

## Disable Data Inputs from previous Workshops

1. Login to your NOW Instance as Administrator
2. Navigate to **Health Log Analytics > Data Input > Data Inputs**
3. Select all Data Inputs (e.g., nginx, spring, mysql)
4. **Stop Data Input** using the “Action on Selected rows” field

## Rotate Application Service

1. Login to your NOW Instance as Administrator
2. Navigate to **CSDM > Manage Technical Services > Application Service**
3. Create a **New** Application Service called “Pet Clinic YYYYMMDD-N” as follows:

|  |  |
| --- | --- |
| * Field | * Value |
| * Name | * Pet Clinic 20211006-1 |
| * Operational Status | * Operational |

## Reconfigure Data Input Mappings

1. Login to your NOW Instance as Administrator
2. Navigate to **Health Log Analytics > Mapping > Data Input Mapping**
3. Select each Data Inputs Mapping (e.g., nginx, spring, mysql)
4. Update each **JS Function** as follows function map(sample, metadata) { return { 'applicationService': ‘YOUR NEW APPLICATION SERVICE’, 'component': null, 'sourceType': null, }; } // Do not write code here
5. Test and publish the JavaScript Function

## Re-enable Data Inputs

1. Login to your NOW Instance as Administrator
2. Navigate to **Health Log Analytics > Data Input > Data Inputs**
3. Select all Data Inputs (e.g., nginx, spring, mysql)
4. Select **Start Data Input** using the “Action on Selected rows” field

# Appendix A – Terraform Installation

These instructions apply to a WSL Ubuntu workstation. Please refer to the Terraform Downloads page https://www.terraform.io/downloads.html for more information

1. Start a Bash Shell
2. Install Terraform CLI (e.g., v0.12.31) $ wget -qO- https://releases.hashicorp.com/terraform/0.12.31/terraform\_0.12.31\_linux\_amd64.zip | busybox unzip - $ chmod 775 terraform $ sudo mv terraform /usr/local/bin/

# Appendix B – Ansible Installation

These instructions apply to a WSL Ubuntu workstation and Python PIP which provides broad cross-platform support regardless of the operating system. Please refer to the Ansible Installation Guide https://docs.ansible.com/ansible/latest/installation\_guide for more information.

1. Start a Bash Shell
2. Install Ansible (e.g., v4.5.0) $ pip install ansible==4.5.0
3. Edit Ansible Settings (e.g. vi ~/.ansible.cfg) [defaults] interpreter\_python=auto\_silent ideprecation\_warnings=false

# Appendix C – AWS CLI Installation

These instructions apply to a WSL Ubuntu workstation. Please refer to the AWS CLI Installation Guide https://docs.aws.amazon.com/cli/latest/userguide/install-linux.html for more information.

1. Start a Bash Shell
2. Install Venv $ sudo apt-get install -y python3-venv
3. Install the AWS CLI $ curl "https://s3.amazonaws.com/aws-cli/awscli-bundle.zip" -o "awscli-bundle.zip" $ unzip awscli-bundle.zip $ sudo /usr/bin/python3 awscli-bundle/install -i \ /usr/local/aws -b /usr/local/bin/aws