**0. Overview**

SmartSense is a supportability tool from HWX. We use it mainly for log aggregation purpose. Instead of going to different directories on the cluster or different tables from our MDS backend, SmartSense is one easy way for CSS engineers to get most (if not all) of the needed logs (SmartSense does not persist logs, so we need customer to capture the bundle as soon as the issue happens). SmartSense also enables us to get a faster support experience from HWX, as they have an analytical backend that will take the SmartSense Bundle.

This document is organized as following:

1. How SmartSense works
2. Confirm SmartSense Installed
3. How to Capture Bundles
4. How to Download Bundles
5. How to Enable SmartSense View
6. Limitations and Future Releases

**1. How SmartSense works**

SmartSense follows a Master-Slave design pattern. The master service, HST-Server is installed on Headnode 0. The agent is installed on both headnodes, all worknodes, all zookeeper nodes. The agents are responsible for collecting and zip diagnostic package for the nodes hosting them. The per-node package will then be send to the server, and the server will do the aggregation and create a single package for the whole cluster.

The current architecture is demonstrated in the following diagram:

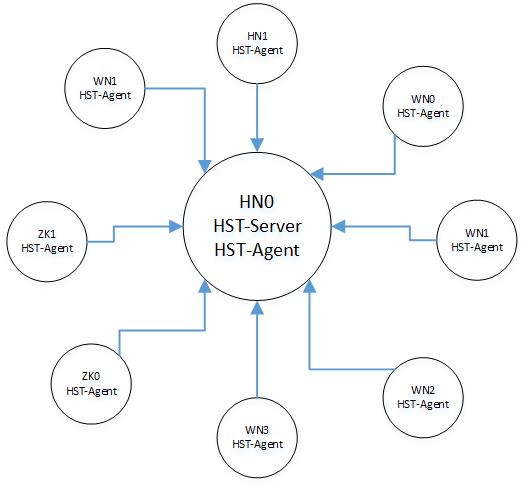


Figure 1 SmartSense Architecture

**2. Confirm SmartSense Installed**

After your cluster created, log into Ambari UI and look for SmartSense service. If you can find the following, means SmartSense is properly installed and stared on your cluster.

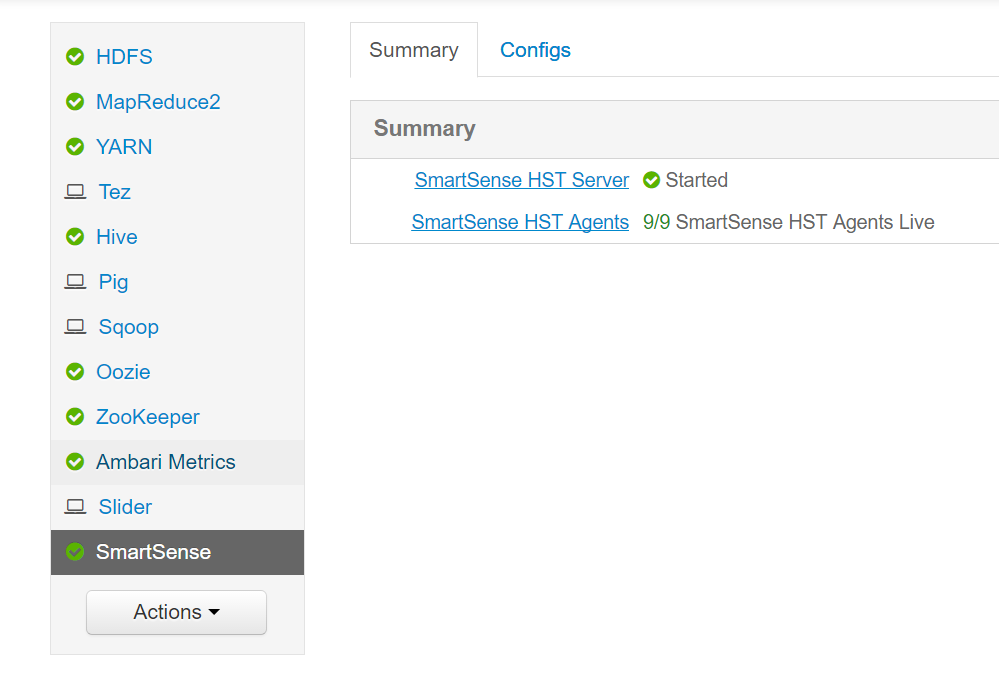


Figure 2 SmartSense Service Page in Ambari UI

**3. How to Capture Bundles**

There are two basic ways people can capture bundles:

1) Via SmartSense View

2) Using script action

3) Tiger Script

As a request from Azure Security team, SmartSense View will not come as default for HDI 3.4. Users have to manually enable it. So that is not a recommend way. But we will document how to enable SmartSense view in this document, in case needed.

**3.1 Recommended way for HDI customers**

For HDI customers, the recommend way is to submit a script action through Azure Portal.

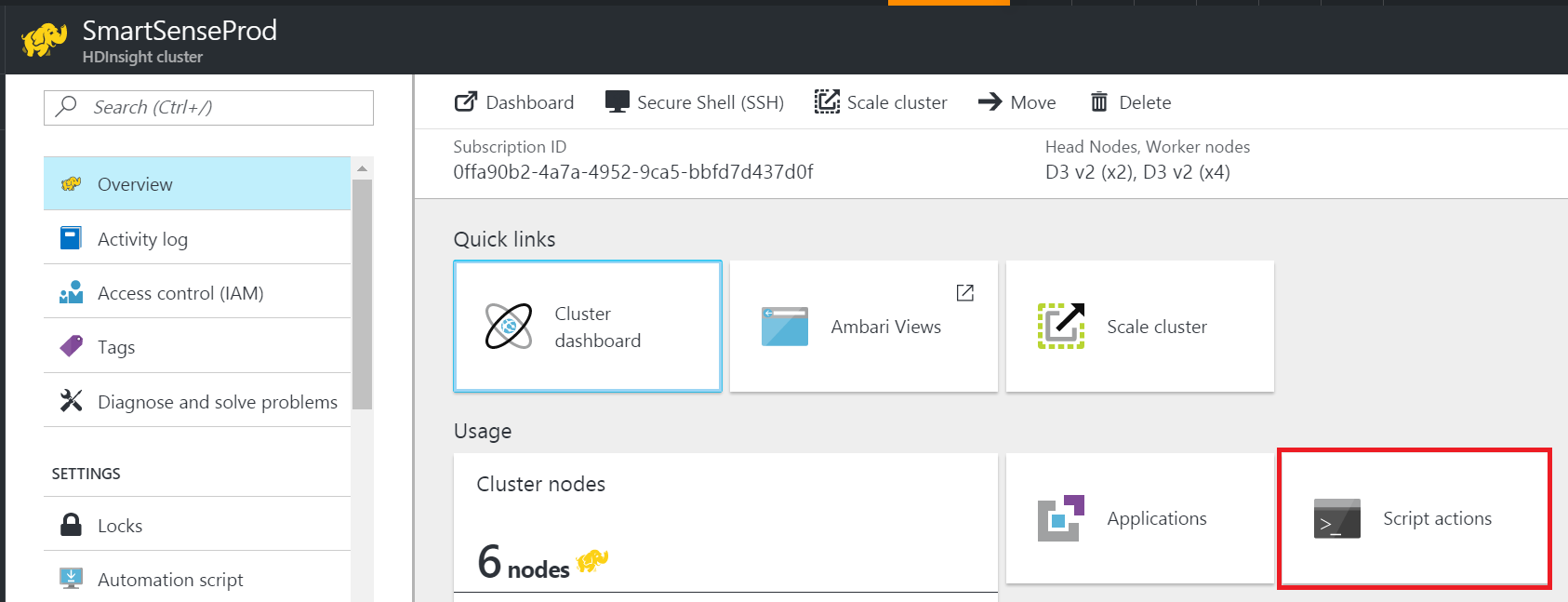


Figure 3. Script Action though Azure Portal

Click “Script actions” and you will see the following:

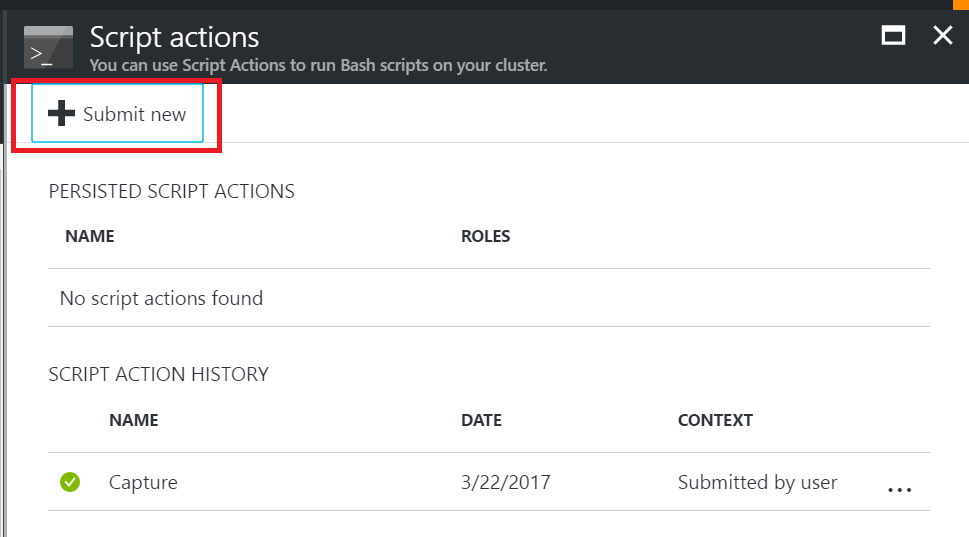


Figure 4. Create Script Action

Click on “Submit new” and you will see the following:

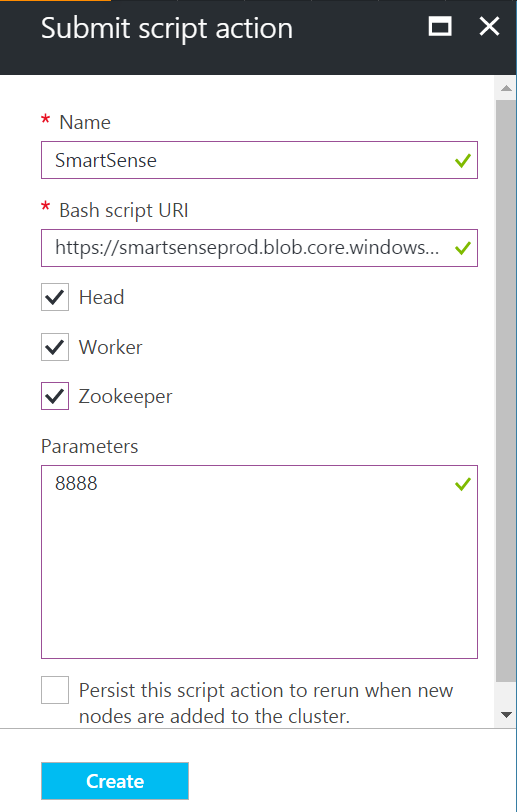


Figure 5. Config Script Action

* Give a name to the script action
* Put the following URI for “Bash script URI”: <https://smartsenseprod.blob.core.windows.net/scriptaction/Capture.sh>
* Check all node types
* Put a CaseID as “Parameters”

Click “Create” and you will see the script been “accepted”. Now go to Ambari UI, you will find the following:

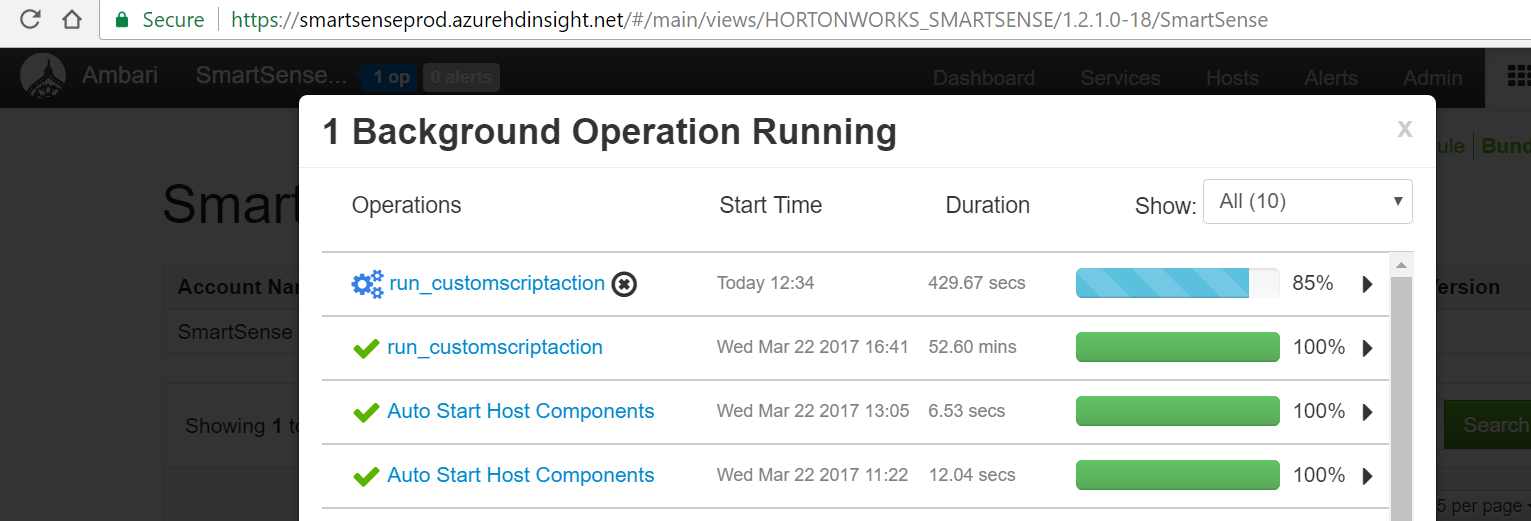


Figure 5. Script Action in Ambari UI

**3.2 Recommended way for HDI Support Staffs**

**Prerequisites: (1) A HDInsight Linux cluster, (2) Install POSTMAN**

**Step 1: Getting the list of hosts available in the cluster:**

Send a GET request through POSTMAN with the following parameters:

URL: [https://[CLUSTER NAME].azurehdinsight.net/api/v1/clusters/[CLUSTER NAME]/hosts/](https://dulehuelinux19.azurehdinsight.net/api/v1/clusters/dulehuelinux19/hosts/)

In the Headers tab: Set the following values:

* X-Requested-By "ambari"
* Need username and password (set in the Authorization tab)

In the example below, I get all hosts for the cluster "**SmartSenseProd**"

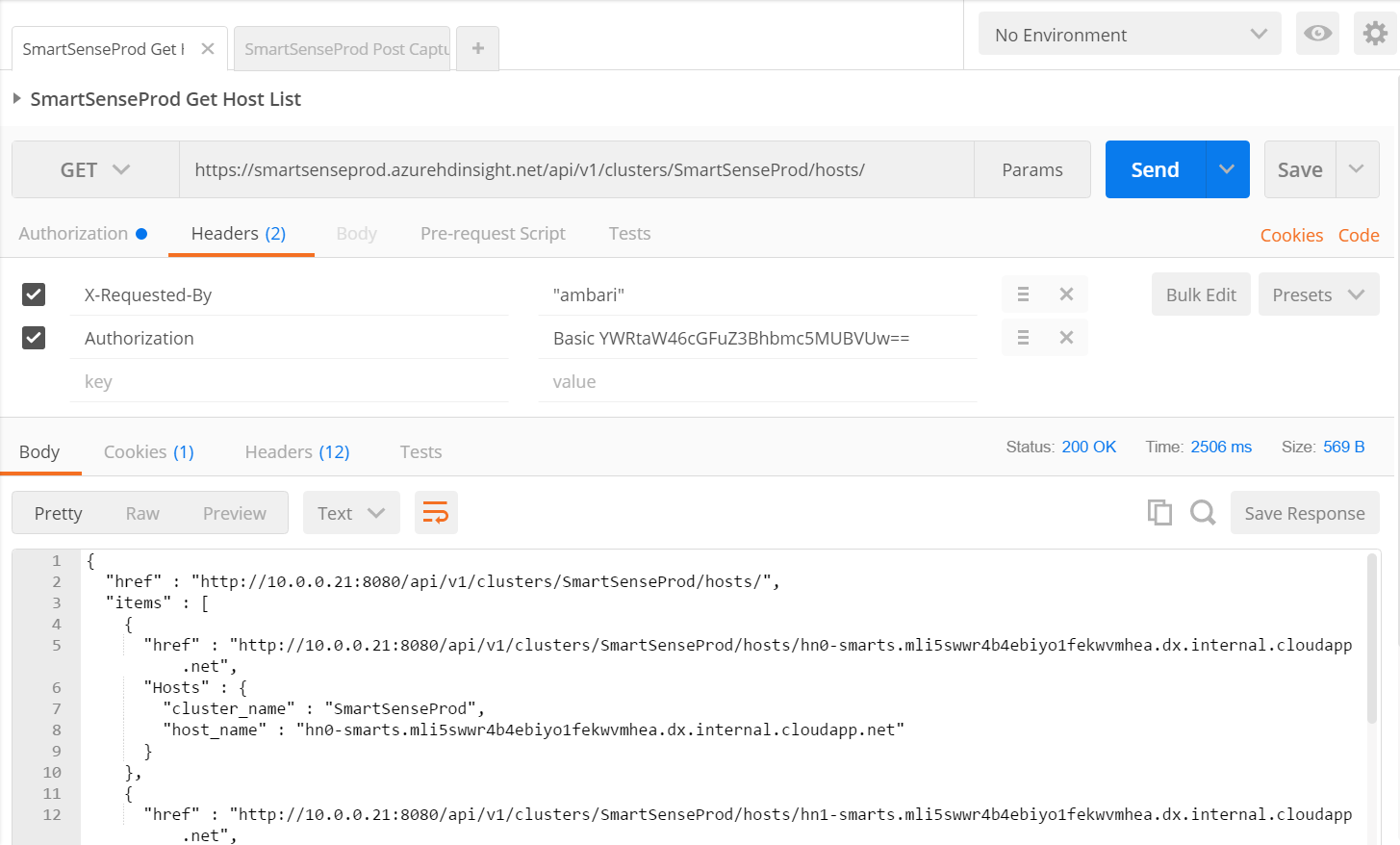


Figure 6. Get Host List through PostMan

**Step 2: Running customization on the cluster on certain hosts:**

Send a POST request to the following URL:

 URL: [https://[CLUSTER NAME].azurehdinsight.net/api/v1/clusters/[CLUSTER NAME]/requests](https://dulehuelinux19.azurehdinsight.net/api/v1/clusters/dulehuelinux19/requests)

In the Headers tab: Set the following values:

* X-Requested-By "ambari"
* Need username and password (set in the Authorization tab)

 In the Body tab, select "raw" and use the following payload:

{

    "RequestInfo": {

        "action": "run\_customscriptaction",

        "context": "run\_customscriptaction",

        "parameters":

        {

          "script\_location":"[script action URI that is publicly available]",

          "script\_params":"[script action parameters if exists]",

          "jdk\_location":"[https://[cluster name].azurehdinsight.net/api/v1/resources](https://ksvijayatestlogs.azurehdinsight.net/api/v1/resources)",

          "storage\_account":"",

          "storage\_key":"",

          "storage\_container":"",

          "blob\_name":""

        }

    },

    "Requests/resource\_filters":[{"hosts":"<<comma separated list of unique host names>>"}]

}

In the following example, I submit a script to run on two headnodes and my request was accepted by Ambari:

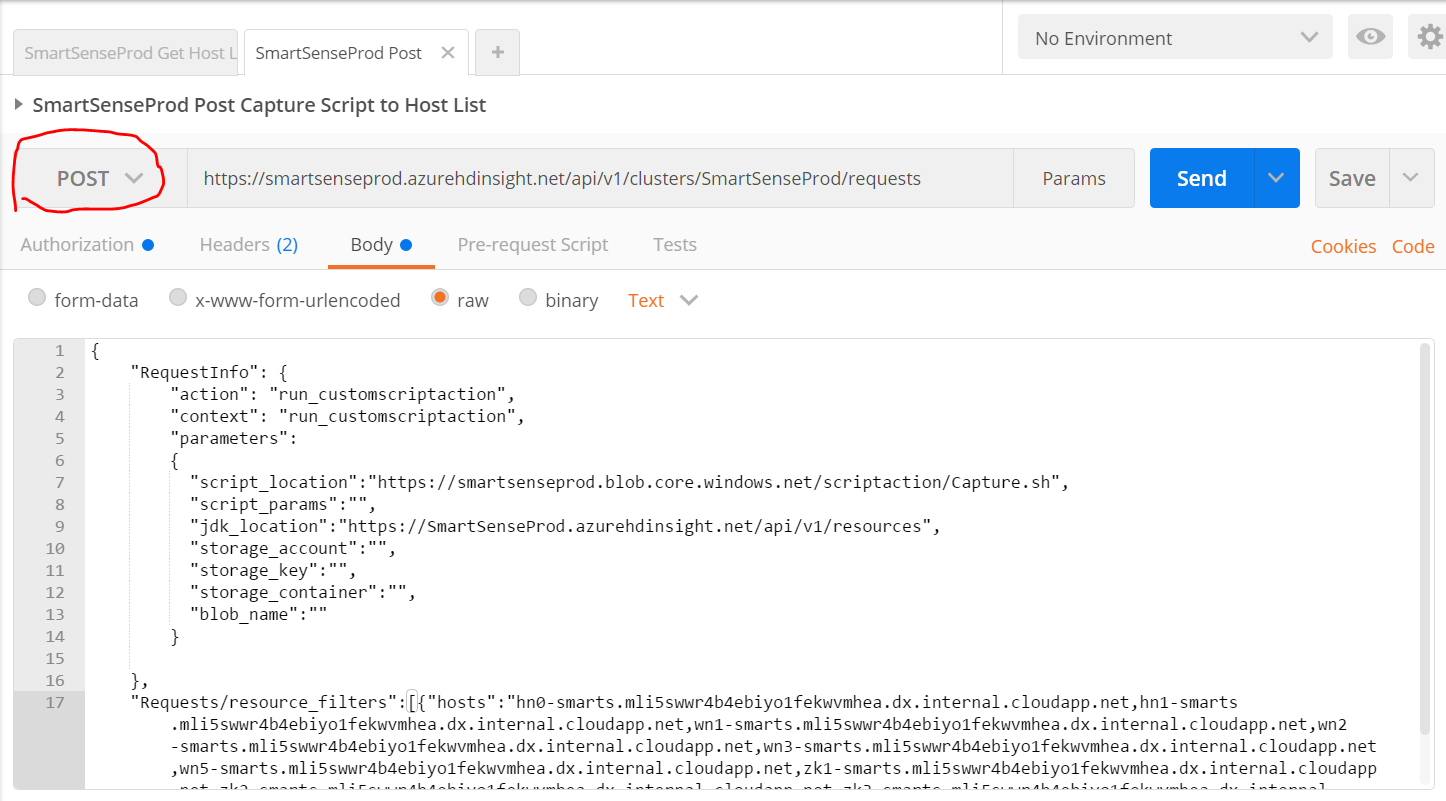


Figure 6. Post Script to Host List through PostMan

I can then go to Ambari UI to see the status of my request as shown in Figure 5.

**3.3 Tiger Scripts**

The most convenient way for MS Support engineers is to use Tiger Scripts. But as Tiger Scripts will install 3rd party tools on customer clusters, so we do not recommend this way. We document it here as an alternative option.

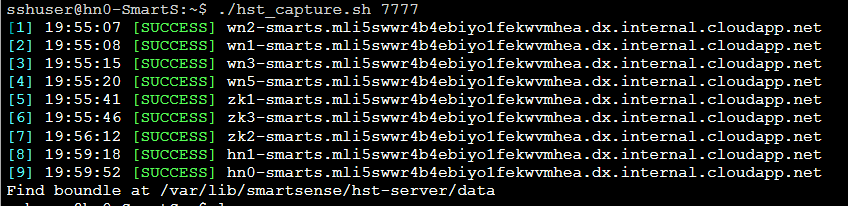


Figure 7. Tiger Scripts for SmartSense Capturing

As you can see with Tiger Scripts, capturing a bundle is as easy as one command.

Please refer to the following [link](TBD) for more details.

**4. How to Download Bundles**

Once finished, the bundles are located on hn0, at /var/lib/smartsense/hst-server/data.

We recommend use SmartTTY. You can download like the following:

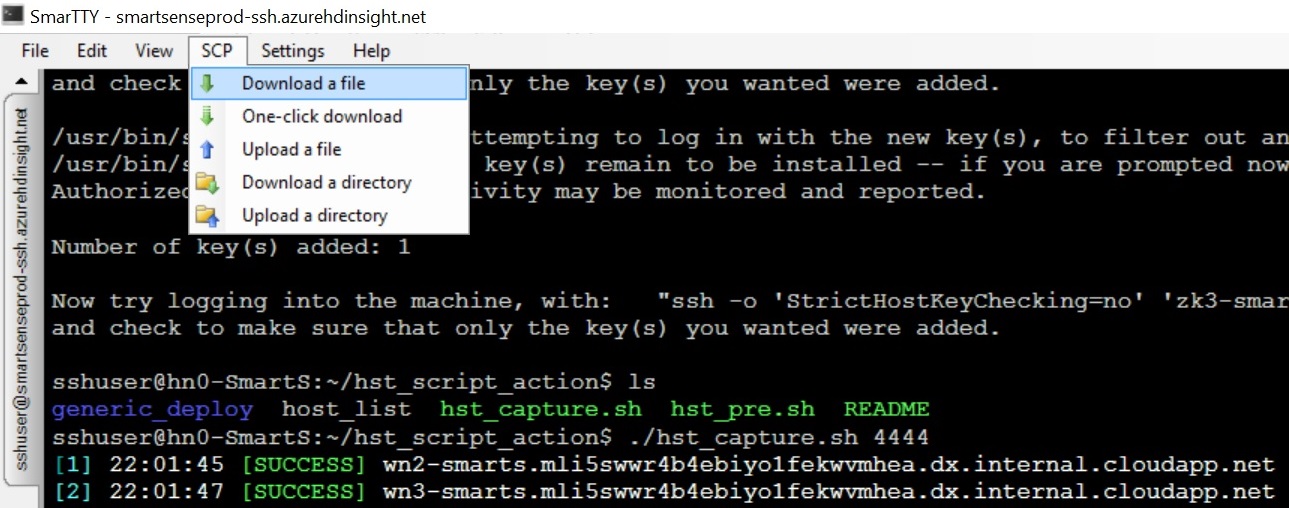


Figure 8. Download bundles.

**5. How to Enable SmartSense View**

**6. Limitations and Future Releases**

* For 3.4 because the way SmartSense 1.2 works, SmartSense view will not come as default. Manual steps needed in order to enable the view.
* Download for bundles are blocked but we may look to enable this for a better UX.
* Yarn App logs are not supported for HDI 3.4. For HDI 3.6, we will have this feature.