PureConnect®

2020 R1

Generated:

17-April-2020

Content last updated:

20-March-2020

See Change Log for summary of changes.

GENESYS™

CX Insights

Installation and Configuration Guide

Abstract

This document contains installation and configuration information for Pureconnect CX Insights, which provides real-time analytics dashboards.

For the latest version of this document, see the PureConnect Documentation Library at: http://help.genesys.com/cic.

Table of Contents

CX Insights Installation and Configuration Guide

This document contains installation and configuration information for PureConnect CX Insights, which provides real-time analytics dashboards. This document is intended for system administrators and users who are installing and setting up CX Insights.

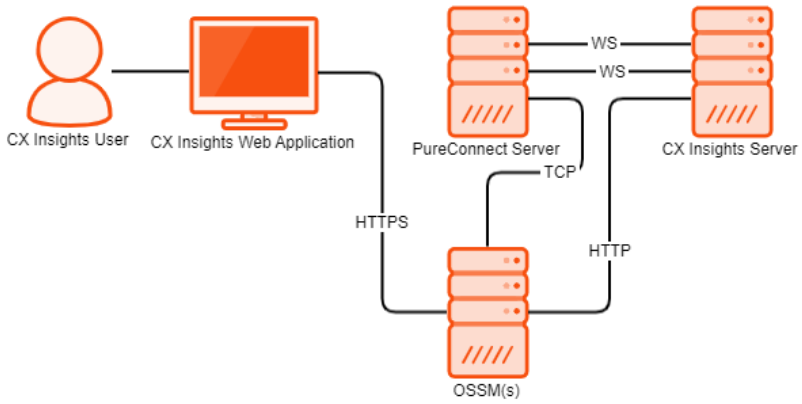
CX Insights overview

CX Insights is a web-based application that allows you to display interactive dashboards to view and analyze real-time agent status and workgroup activity. Agent dashboard visualizations help you monitor agent status and agent interaction details in real-time. Workgroup dashboard visualizations give supervisors a quick look at available agents and their current states. Each agent or supervisors requires an assigned Analytics Core User license in order to log in, and they also need to have access permission to use the dashboards.

CX Insights is built on the MicroStrategy Business Intelligence (BI) platform that runs best in a Linux environment. It is deployed as a set of Docker containers through an Ansible playbook. CX Insights can be accessed on Google Chrome, Mozilla Firefox, and Internet Explorer.

CX Insights architecture

CX Insights deployment model



CX Insights server

The CX Insights server is a Linux server that uses Docker Compose to run the containerized version of the MicroStrategy BI platform, as well as integration containers used for interfacing with PureConnect. The primary driver of the following resource requirements is the MicroStrategy BI platform. It uses in-memory cubes to model incoming real-time statistics for use by visualizations in dashboards.

CX Insights web application

The CX Insights web application is built on the same framework as Interaction Connect and shares the same server requirements.

CX Insights prerequisites

CX Insights requirements

CX Insights server requirements

Hardware

Genesys has tested the following machine specifications to verify a deployment consisting of 1000 PureConnect users taking interactions across an average of 10 workgroups each. Significantly larger deployments may require additional CPU and RAM to retain performance for the increased incoming traffic from the PureConnect Server.

Component	Requirement
Platform	Virtual machine or physical server
CPU	• 8 cores • AMD-V or VT-X VM-extensions
RAM	32 GB
Storage space	512 GB
Swap partition	32 GB

Software

Important!

During installation of Centos, you must include Virtualization Host to minimize the amount of additional configuration required to get Docker running.

Component	Requirement
Operating system	Centos 7
Software components	Virtualization Host: • KVM • QEMU • QEMU+KVM • Libvirt

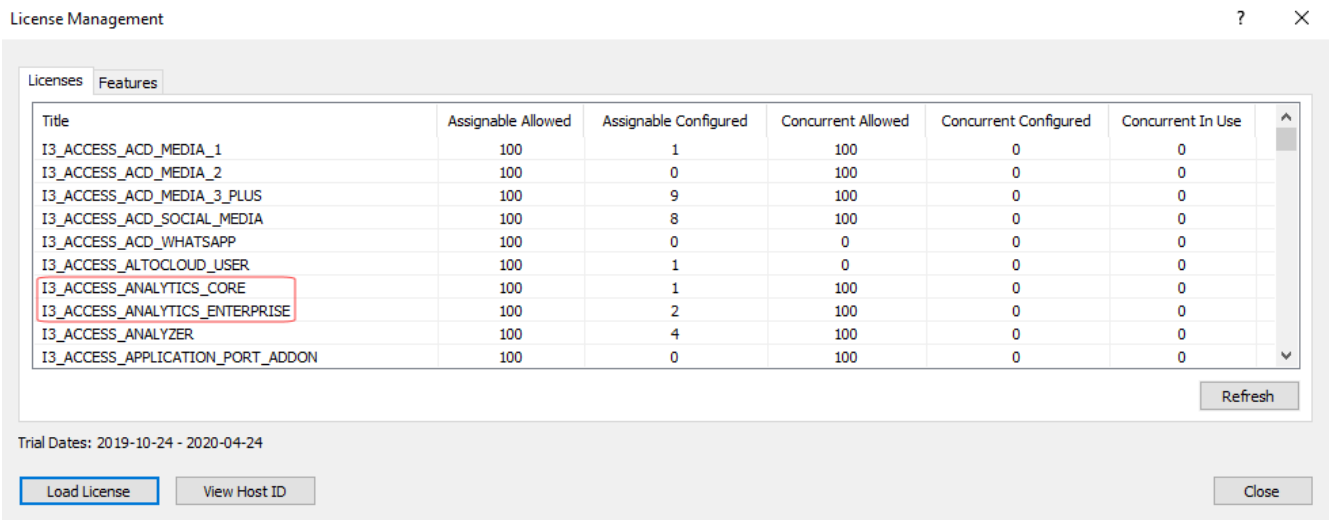
CX Insights licensing

CX Insights requires an Analytics access license for users, and an Analytics feature license.

Analytics access licenses

To verify if you have the Access licenses, go to the **License Management** form in Interaction Administrator and under the **Licenses** tab, verify the following licenses.

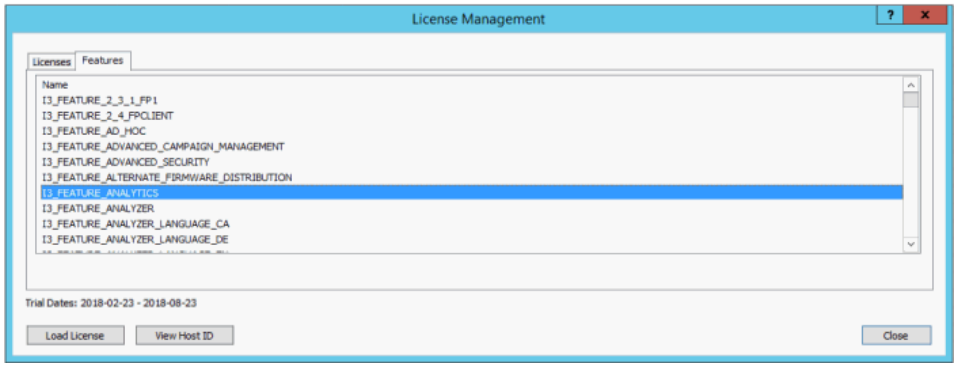
License	Description
I3_ACCESS_ANALYTICS_CORE	Basic dashboard license to view dashboards
I3_ACCESS_ANALYTICS_ENTERPRISE	This license will allow users to create and modify dashboards and also allows external data sources to build dashboards



The **License Management** dialog displays the number of available licenses.

Analytics feature license

To verify if you have the Analytics feature license, go to the **License Management** form in Interaction Administrator and under the **Features** tab, verify the **I3_FEATURE_ANALYTICS** license.



If a license is not present or you do not have enough licenses, contact your sales representative.

CX Insights server installation

CX Insights server installation

The CX Insights server hosts the MicroStrategy BI platform, which is the backend for providing real-time analytics and dashboards in the CX Insights web application. The following server setup and configuration instructions require a knowledgeable Linux administrator and familiarity with Centos.

Install CX Insights server

1. Install Centos7 on either a physical or virtual server that meets the minimum requirements
   - 8+ vcpu
   - 32 GB RAM
   - 512 GB total storage space
   - When installing Centos make sure the swap partition is at least 32 GB
2. Download CX Insights artifacts from the following website:
   https://my.inin.com/products/cic/Pages/Utilities-Downloads.aspx
3. Unzip the CX Insights artifacts archive that contains pcc-cxinsights-playbook, pcon-mstr, and utils.
4. Run the shell script ansible_install.sh to install the dependencies like python, ansible packages with root user account and also creates CX Insights user account to perform all the ansible roles and tasks. If the Centos already has pip installed then ensure that pip is of version 8.1.2, which is compatible with python 2.7.5 else all the following steps will fail.
5. Verify if ansible is installed or not using the command "which ansible", then if installed ansible version appears. If not installed, then re-run the ansible_install shell script again.
6. Verify if CX Insights account is created, using command "cut -d: -f1 /etc/paswd" and login to cxinsights account.

- su cxinsights

- Prerequisites for running ansible-playbook

- Unpack the pcc-cxinsights-playbook file in the CX Insights user home directory.
- Copy helm charts folder pcon-mstr inside cxinsights-playbook folder.
- Generate ansible vault for CX Insights user password, which is required by ansible modules to install k3s, helm and tiller.
  - **Ansible-vault encrypt_string 'passwd' --name 'helm_linux_host_passwd' --vault-id cxinsights@prompt**, replace password with value given while create CX Insights user account. It will ask password for vault usage enter and remember the password which user has to enter while running anisble-playbook command
  - **Ansible-vault encrypt_string 'passwd' --name 'tiller_linux_host_passwd' --vault-id cxinsights@prompt**, again generate the password, in case if you are planning to keep controller and CX Insights server separately, else add the above generate vault value in both helm_linux_host_passwd and till_linux_host_passwd in group_vars/all.yml file as shown below
- Change the hosts details in values.yml file to linux host, where CX Insights to be installed in cxinsights-playbook folder, ensure that the time zone details are mapped with your region where gcxi server is to be installed.

>>>> ORIGINAL //eic/2019r4_systest/products/documentation/source/Technical_Reference_HTML/cx_insights_icg/CX_Insights_installation_procedure.htm#9

[Production]

<host fqdn> ansisble_connection=local pcon_server_timezone=,e.g. America/Indiana/Indianapolis> pcon_server_locale=<e.g. en_us> pcon_server_proxy_rewrite_url="analystics/analytics-route/<PureConnect Server>" websocket_auth _secret=<create a

==== THEIRS //eic/2019r4_systest/products/documentation/source/Technical_Reference_HTML/cx_insights_icg/CX_Insights_installation_procedure.htm#10

[Production]

<host fqdn> ansisble_connection=local ansible_user=cxinsights pcon_server_timezone=,e.g. America/Indiana/Indianapolis> pcon_server_locale=<e.g. en_us> pcon_server_proxy_rewrite_url="analystics/analytics-route/<PureConnect Server>" websocket_auth _secret=<create a

==== YOURS //hyd-arunas-l_eic_2020r1_systest/products/documentation/source/Technical_Reference_HTML/cx_insights_icg/CX_Insights_installation_procedure.htm

```
gcxi:
  gcxiproperties:
    proxyRewriteUrl: /analytics
    pconLocale: en_us
secret: analytics
global:
  tz: America/Indiana/Indianapolis
  hosts:
  - <host QDN>
```

- Below is the inventory.yml file in the cxinsights-playbook directory substitute with appropriate values. This example assumes ansible and k3s running on the same machine. If the controller is different from target machine, then helm_linux_host should be the controller host FQDN and tiller-linux-host should the FQDN of cxinsights server host.

```
---
helm_linux_host:
hosts:
xxx-xxxxx-xxxxx.xxxxxxx.com
vars:
ansible_user: '{{ user }}'
ansible_ssh_pass: '{{ passwd }}'
tiller_linux_host:
hosts:
xxx-xxxxx-xxxxx.xxxxxxx.com
vars:
ansible_user: '{{ user }}'
ansible_ssh_pass: '{{ passwd }}'
```

<<<<

>>>> ORIGINAL //eic/2019r4_systest/products/documentation/source/Technical_Reference_HTML/cx_insights_icg/CX_Insights_installation_procedure.htm#9

8. Unpack the `cxinsights-playbook/group_vars/production.yml` file.
9. Update the value for the `docker_repo` parameter to the repository where the Docker images have been uploaded. If the images were uploaded directly to the cxinsights server, then use `pureconnect`.
10. Create an inventory file in the `cxinsights-playbook` directory. It should look like the following example with the appropriate values substituted:

```
localhost ansible_connection=local pcon_server_timezone=<e.g. America/Indiana/Indianapolis>

pcon_server_locale=<e.g. en_us> pcon_server_proxy_rewrite_url="analytics/analytics-route/<PureConnect

Server>" websocket_auth_secret=<create a password>
```

host FQDN	Current server where you are running ansible play book
ansible_connection	This is the current session for the current server
pcon_server_timezone	PureConnect IC timezone
pcon_server_locale	PureConnect IC locale
pcon_server_proxy_rewrite_url	Rewrite URL for web proxy `analytics/analytics-route/<PureConneectServer>` *Analytics-> app folder in IIS* `analytic-route` should be change Here `PureConnectServer` should be `CIC Server ip` or `fgdn`
websocket_auth_secret	Secret key for web sockets to be configured in Interaction Administrator

==== THEIRS //eic/2019r4_systest/products/documentation/source/Technical_Reference_HTML/cx_insights_icg/CX_Insights_installation_procedure.htm#10

11. Unpack the `cxinsights-playbook/group_vars/production.yml` file.
12. Update the value for the `docker_repo` parameter to the repository where the Docker images have been uploaded. If the images were uploaded directly to the cxinsights server, then use `pureconnect`.
13. Create an inventory file in the `cxinsights-playbook` directory. It should look like the following example with the appropriate values substituted:

```
localhost ansible_connection=local pcon_server_timezone=<e.g. America/Indiana/Indianapolis>

pcon_server_locale=<e.g. en_us> pcon_server_proxy_rewrite_url="analytics/analytics-route/<PureConnect

Server>" websocket_auth_secret=<create a password>
```

host FQDN	Current server where you are running ansible play book
ansible_connection	This is the current session for the current server
pcon_server_timezone	PureConnect IC timezone
pcon_server_locale	PureConnect IC locale
pcon_server_proxy_rewrite_url	Rewrite URL for web proxy `analytics/analytics-route/<PureConneectServer>` *`Analytics-> app folder in IIS`* `analytic-route` should be change Here `PureConnectServer` should be `CIC Server ip` or `fgdn`
websocket_auth_secret	Secret key for web sockets to be configured in Interaction Administrator

==== YOURS //hyd-arunas-l_eic_2020r1_systest/products/documentation/source/Technical_Reference_HTML/cx_insights_icg/CX_Insights_installation_procedure.htm <<<<

14. Run the Ansible Playbook to start the services on the CX Insights server. At the first time it may be slow as dependencies are installed, and container images need to get downloaded.
    >>>> ORIGINAL //eic/2019r4_systest/products/documentation/source/Technical_Reference_HTML/cx_insights_icg/CX_Insights_installation_procedure.htm#9
    - `cd cxinsights-playbook`
    - `ansible-playbook -i production ./site.yml -b`
      ==== THEIRS //eic/2019r4_systest/products/documentation/source/Technical_Reference_HTML/cx_insights_icg/CX_Insights_installation_procedure.htm#10
    - `cd cxinsights-playbook`
    - `sudo ansible-playbook -i production ./site.yml -b`
      ==== YOURS //hyd-arunas-l_eic_2020r1_systest/products/documentation/source/Technical_Reference_HTML/cx_insights_icg/CX_Insights_installation_procedure.htm
    - Sudo ansible-playbook --vault-id cxinsights@prompt -i inventory.yml site.yml -K
    - Run below command to ensure everything is up and running
      <<<<

kubectl get pods to see all the containers are up and running

```
[cxinsights@pc-k3s-s01 pcc-cxinsights-playbook]$ kubectl get pods --namespace=pcn-cxinsights-system
NAME                                                READY   STATUS    RESTARTS   AGE
pcc-helmcharts-gcxi-postgres-69cf45f787-rgq7b       1/1     Running   0          29m
pcc-helmcharts-mstrdataadapteragent-bfcc7dfd5-lwlpc 1/1     Running   0          29m
pcc-helmcharts-gcxi-75c6df6bbb-n77b8                1/1     Running   0          29m
pcc-helmcharts-mstrconnector-6d595679f8-72r9v       1/1     Running   0          29m
pcc-helmcharts-mstrdataadapterserver-76d6b8869b-sgxjd 1/1   Running   0          29m
pcc-helmcharts-gcxi-6d8d476d4d-zh5jm                0/1     Pending   0          10m
```

kubectl get services

```
[cxinsights@pc-k3s-s01 pcc-cxinsights-playbook]$ kubectl get services --namespace=pcn-cxinsights-system
NAME                                          TYPE        CLUSTER-IP       EXTERNAL-IP   PORT(S)
E
gcxi-postgres                                 ClusterIP   192.168.128.50   <none>        5432/TCP,9090/TCP
m
pcc-helmcharts-mstrdataadapteragent           ClusterIP   192.168.204.172  <none>        9090/TCP
m
pcc-helmcharts-mstrdataadapterserver-agentgateway ClusterIP 192.168.137.193 <none>      8079/TCP
m
pcc-helmcharts-mstrconnector                  ClusterIP   192.168.231.3    <none>        8077/TCP,9090/TCP
m
pcc-helmcharts-gcxi                           ClusterIP   192.168.232.239  <none>        34952/TCP,8080/TCP
m
pcc-helmcharts-pcon-mstr                      ClusterIP   192.168.177.112  <none>        80/TCP
m
pcc-helmcharts-mstrdataadapterserver          ClusterIP   192.168.196.99   <none>        8078/TCP,9090/TCP
m
```

kubect1 get pvc

```
[cxinsights@pc-k3s-s01 pcc-cxinsights-playbook]$ kubectl get pvc --namespace=pcn-cxinsights-system
NAME          STATUS   VOLUME                                     CAPACITY   ACCESS MODES   STORAGECLASS   AGE
cube          Bound    pvc-681d50ef-8a20-4622-9d73-993216adf54d   1Gi        RWO            local-path     33m
gcxi-data     Bound    pvc-d7bc8462-3b64-432d-a081-1f485c23f9f0   8Gi        RWO            local-path     33m
gcxi-log      Bound    pvc-6bc0776a-2246-41ea-97d6-184dbb924c78   2Gi        RWO            local-path     33m
gcxi-shared   Bound    pvc-e0c3409a-c22c-45c6-8a95-fa340dd3fbae   1Gi        RWO            local-path     33m
gcxi-volume   Bound    pvc-1e949392-4bd1-447e-a92f-96481ee9c45b   2Gi        RWO            local-path     33m
```

In case if any of the above commands fails to show the list running helm delete --purge pcc-helmcharts --tiller-namespace pcn-tiller-system to delete deployment and try running the ansible-playbook again

Upgrade containers

The sudo ansible-playbook -i inventory.yml site_upgrade.yml -K. Ensure proper tag names updated in values.yml for the container that need to be upgraded. If there is only one container that need to be upgraded, then add tag for the corresponding container and you can omit rest of the properties.

```
gcxi:
  image:
    tag: team.ase283.build.18
     tagcontrol: team.ase283.build.1
gcxi-postgres:
  image:
    tag: team.ase283.build.1
mstrconnector:
 image
   tag: team.ase283.build.4
mstrdataadapteragent:
  image:
    tag: team.ase283.build.9
mstrdataadapterserver:
  image:
    tag: team.ase283.build.4
```

Rollback containers

To get the list of versions installed then type the command **helm history pcc-helmcharts --tiller-namespace pcn-tiller-system,** sample output as shown

```
[cxinsights@qf-cx-docker pcc-cxinsights-playbook]$ helm history pcc-helmcharts --tiller-namespace pcn-tiller-system
REVISION        UPDATED                         STATUS          CHART           DESCRIPTION
1               Fri Mar  6 06:57:32 2020        SUPERSEDED      pcon-mstr-0.1.0 Install complete
2               Fri Mar  6 07:18:08 2020        SUPERSEDED      pcon-mstr-0.1.0 Upgrade complete
3               Fri Mar  6 07:32:30 2020        DEPLOYED        pcon-mstr-0.1.0 Rollback to 1
```

Replace the version number that need to be rollbacked in roles/helm-chart-rollback/vars/main.yml file and run the **sudo ansible-playbook --vault-id cxinsights@prompt -i inventory.yml site_rollback.yml -K**.command.

Deleting setup

Use the **sudo ansible-playbook --vault-id cxinsights@prompt -i inventory.yml site_delete.yml -K** command to delete all the pods, services, ingress end points and persistent volumes, this equivalent to helm delete command.

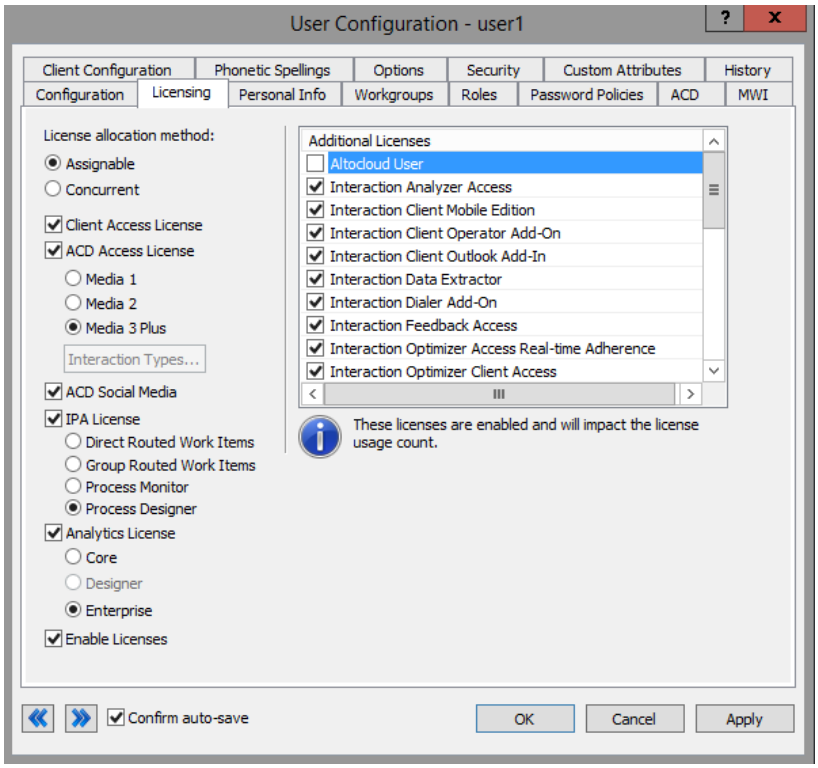
CX Insights server configuration

CX Insights server configuration

To configure the CX Insights server settings in Interaction Administrator, use the following topics.

Allocate Access licenses

Allocate a CX Insights **Analytics License** for each user in Interaction Administrator on the **Licensing** tab.



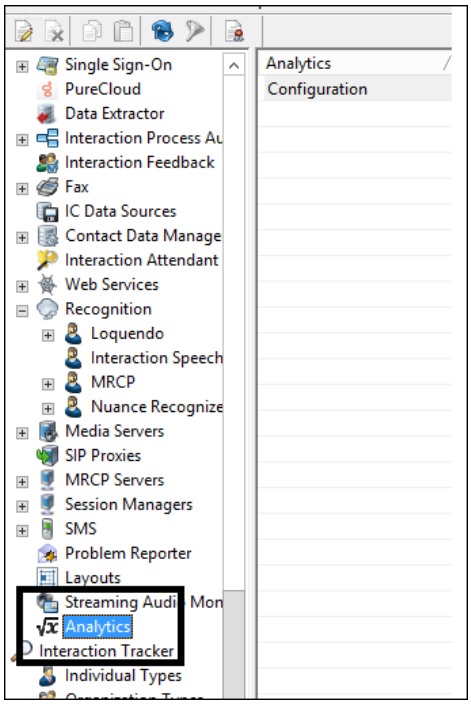
To assign an Analytics license to a user, select the **Analytics License** check box and select one of the following licenses.

CORE	Basic dashboard license to view dashboards
ENTERPRISE	This license will allow users to create and modify dashboards and also allows external data sources to build dashboards

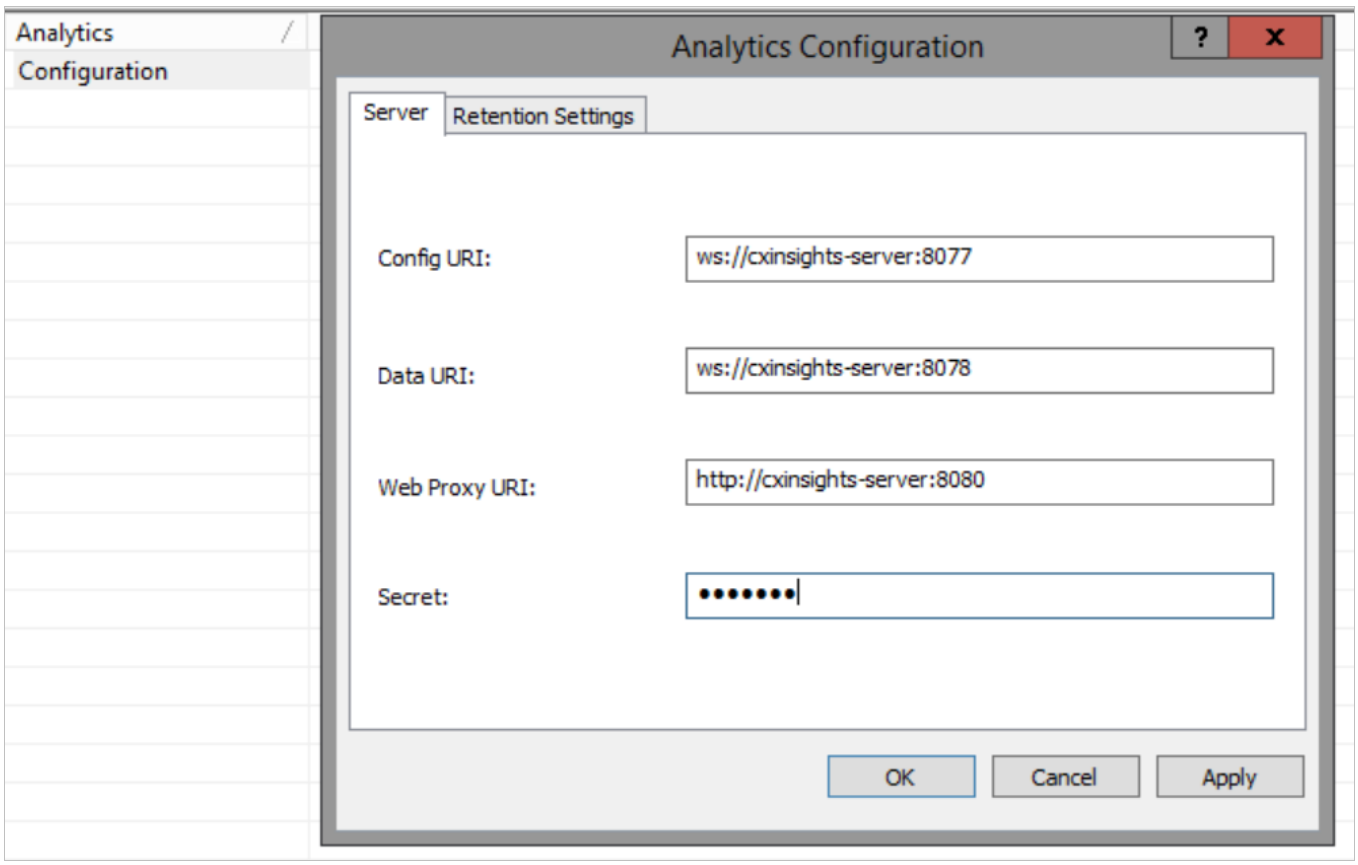
Configure CX Insights server in Interaction Administrator

Once the CX Insights server is up and running, the next step is to configure the PureConnect server to connect to it.

1. Apply the `I3_FEATURE_ANALYTICS` license to the PureConnect server.
2. Open Interaction Administrator and open the Analytics Node under **System Configuration**.



3. In the **Analytics** workspace, click **Configuration**. The **Analytics Configuration** dialog is displayed.



- The Config URI is the websocket address that PureConnect will use to synchronize configuration and security with the CX Insights server. (default port shown)
- The Data URI is the websocket address that PureConnect will stream real-time statistics to the CX Insights server.
- The Web Proxy URI is the target URL used by HttpPluginHost to route web requests.
- The Secret is the websocket_auth_secret that was entered into the inventory file when deploying the CX Insights Server.

Once Configuration is complete, the AnalyticsBridge subsystem will attempt to make the configured websocket connections. If those are successful, the synchronization process will begin. This can take a few minutes to complete if there are a large number of users and workgroups to transfer. Any additional changes to Users, Roles, Workgroups, Access Controls, or Memberships will trigger additional

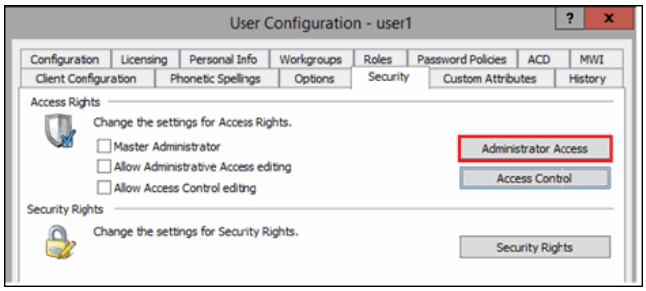
synchronization cycles. Once the servers are synchronized, the AnalyticsBridge Subsystem will begin streaming real-time statistics over the data websocket. At that point, users should be able to view the real-time dashboards.

Configure Administrator Access for CX Insights

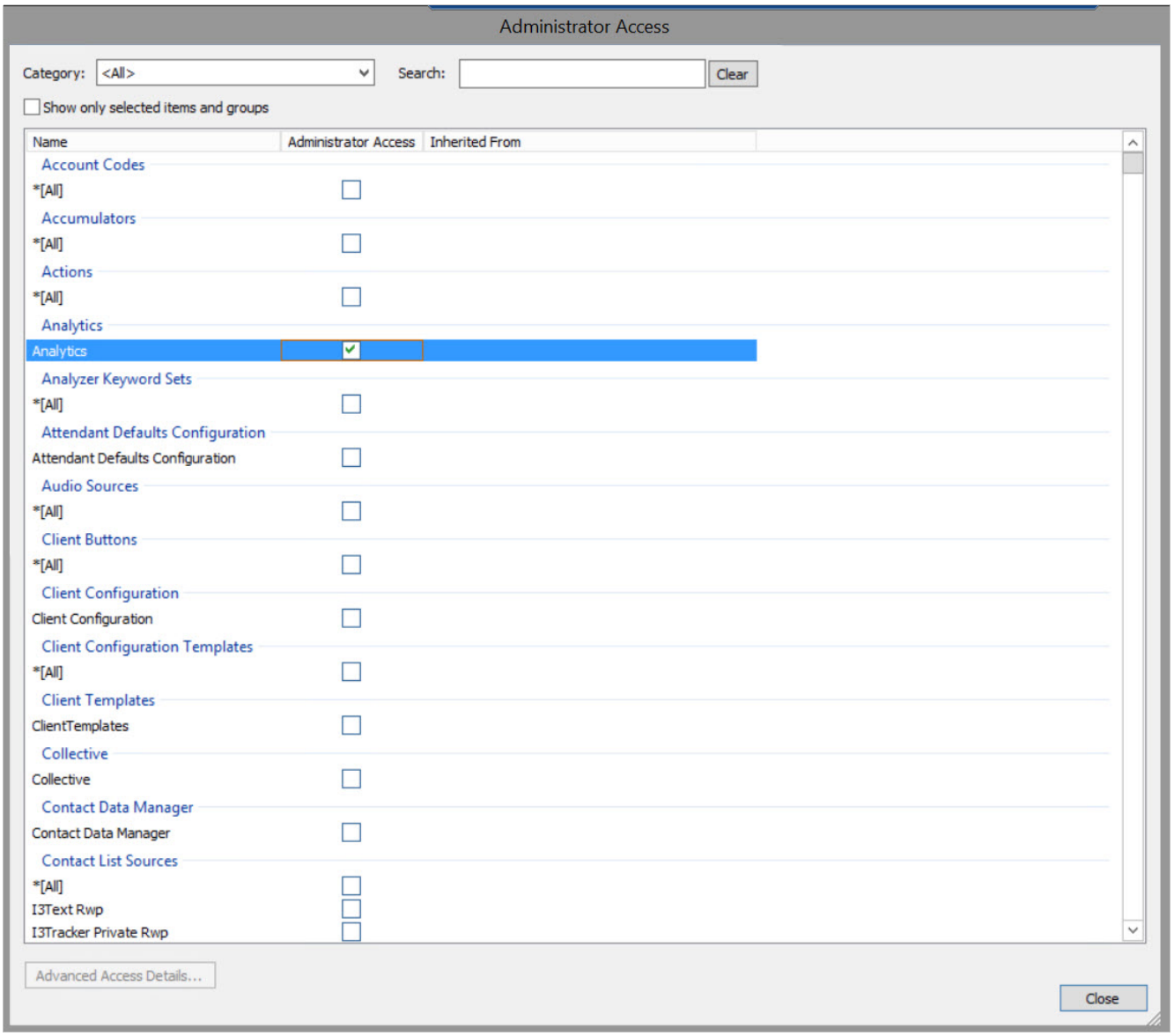
You can restrict which user, workgroup, or role has access to configure the Analytics feature.

To assign administrator access for Analytics:

1. In Interaction Administrator, go to the **User**, **Workgroup**, or **Role** properties dialog box.
2. Select the **Security** tab.



3. Click **Administrator Access**.
4. In the **Administrator Access** dialog, type `analytics` in the **Search** field to filter the list.



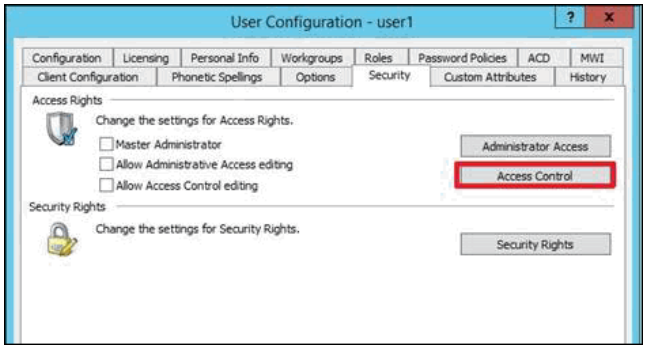
5. To give a user, workgroup, or role Administrator Rights to the Analytics feature, select the **Analytics** check box. You can clear the check box to remove the privilege.
6. Click **Close**.
7. To save the settings, click **OK** or **Apply**.

Configure Access Control for CX Insights dashboards

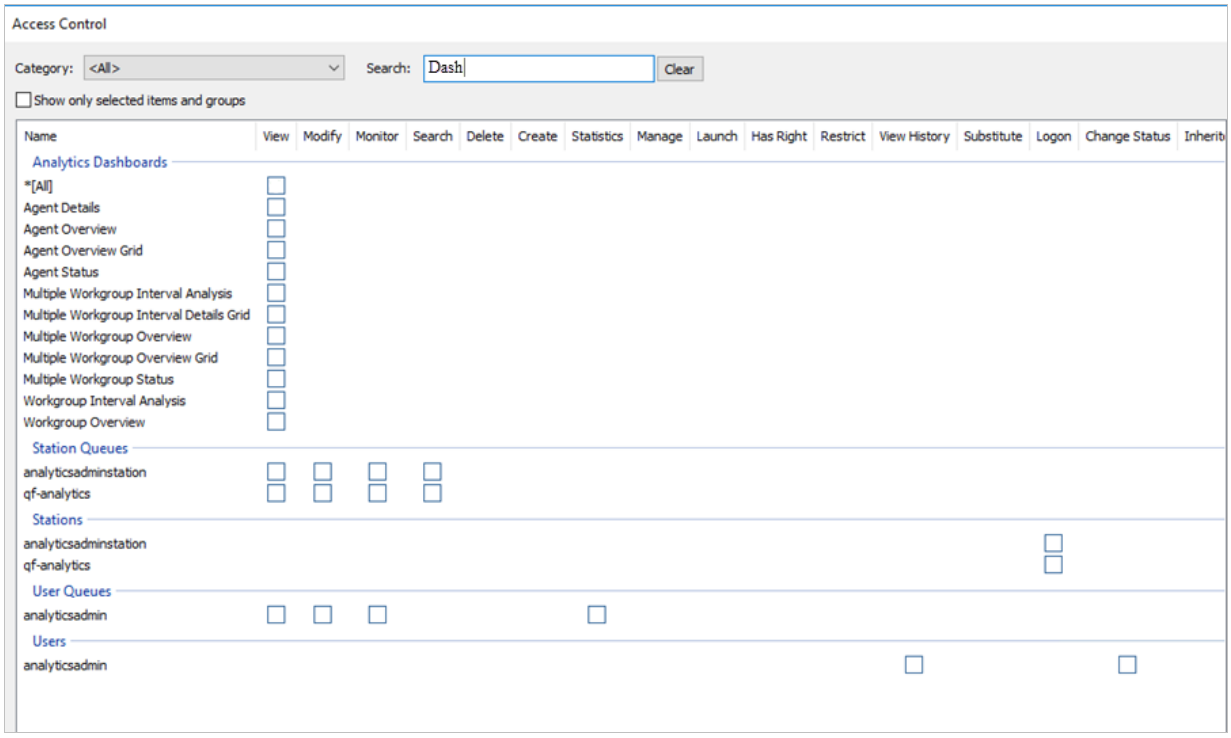
You can restrict which user, workgroup, or role has access to specific dashboards.

To assign dashboard access:

1. In Interaction Administrator, go to the **User**, **Workgroup**, or **Role** properties dialog.
2. Select the **Security** tab.



3. Click **Access Control**.
4. In the **Access Control** dialog, type `dashboards` in the search field to filter the list.



Note:
If the IC Server is in sync with the MicroStrategy server, then the check boxes for all the dashboards are displayed.

5. To assign a user, workgroup, or role access to the dashboard, select the dashboard check box, or select **All** to assign access to all dashboards. Clear a check box to remove the privilege.
6. Click **Close**.
7. Click **OK** or **Apply** to save settings.

Install and configure CX Insights web application

Install CX Insights web application

To host CX Insights web application on web servers, follow the instructions defined in CIC Web Applications Installation and Configuration Guide or download the PDFfile. CX Insights web application does not need any additional inbound or outbound rules to be applied in case of Internet usage.

Public domain purpose

To deploy the CX Insights web application for public domain or on PureConnect Cloud, the following configuration are required:

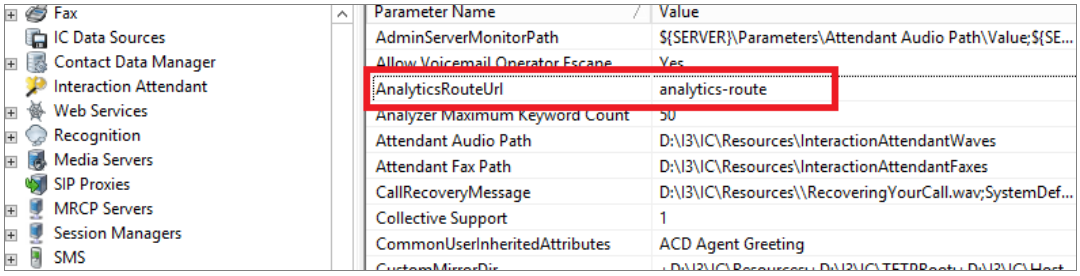
WebServer configuration

You can install and configure CX Insights on anyone of the following web platforms:

- Microsoft Internet Information Server (IIS)
- Apache HTTP Server
- Nginx Server

CIC server configuration

Apart from this configuration on the web server, you must define one server parameter on the CIC server:



MicroStrategy Configuration

In Ansible playbook production inventory file parameter **pcon_server_proxy_rewrite_url** should be defined as "**analytics**" replace **analytics** with full path where web application hosted. For example, if the web application is accessed using url like https://pureconnectprd.simdomain.com/ininapps/analytics/ then the parameter should be defined as "**ininapps/analytics**", this parameter should not be defined in case of intranet usage.

Microsoft Internet Information Server

Install CX Insights web application for Microsoft IIS

For a basic working installation, such as for a test environment, follow the first three sections:

- Step 1: Add Required IIS Services
- Step 2: Download and copy CIC web applications files
- Step 3: Configure IIS

For a production environment, you can also follow the instructions in Configure HTTPS for IIS.

Step 1: Add Required IIS Services

1. In Server Manager, verify that the Web Server Role (IIS 7) is added with the following (minimum required) role services installed:
   - Common HTTP Features
     - Static Content
     - Default Document
   - Performance
     - Static Content Compression
   - Security
     - Request Filtering
   - Management Tools
     - IIS Management Console
2. If you have not installed the **Application Request Routing** and **URL Rewrite extensions**, download them from the following locations and install them.
   - Application Request Routing extension (http://www.iis.net/downloads/microsoft/application-request-routing)
   - URL Rewrite extension (http://www.iis.net/downloads/microsoft/url-rewrite)
3. Enable server as proxy and enable response buffering:
   a. In **IIS Manager**, click your server.
   b. Double-click the **Application Request Routing Cache** module.
   c. In the **Actions** pane, click **Server Proxy Settings**.
   d. Check **Enable proxy**.
   e. Change the **Response buffer threshold (KB)** setting under **Buffer Setting** to `0`.
   f. Click **Apply**.
4. Verify that `index.html` and `index.htm` are present as **Default Documents**.

Step 2: Download and copy CIC web applications files (for analytics only)

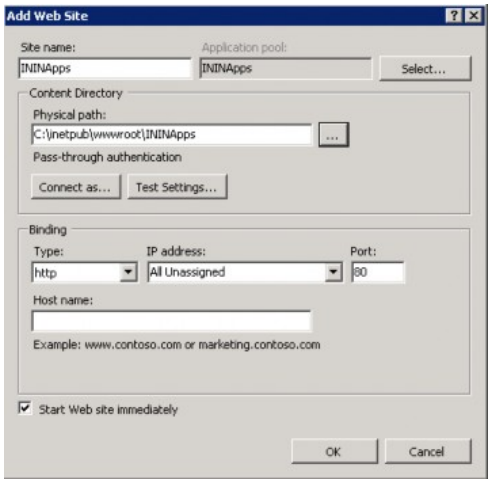
1. In Windows Explorer, create a directory in the Home Directory in IIS for the CIC Web Applications.
   In a default IIS installation, the Home Directory is `C:\inetpub\wwwroot`. Verify that IIS has the appropriate permissions for that newly created directory.

   > **Note:**
   > In this document, the directory is named **ININApps**.

2. Download the CIC Web Applications zip file from https://my.inin.com/products/Pages/Downloads.aspx.
   All the web applications are contained in this single `.zip` archive. You must extract the `analytics` folder only.
3. Unzip the `CIC Web Applications`
4. Navigate to the `web_files` folder inside the unzipped `CIC Web Applications` folder.
5. Copy only the `analytics` folder inside of `web_files`.
6. Paste the folders copied in the previous step into the directory you created in step 1.
   Doing so places the appropriate directory structure and files for CIC Web Applications (**only analytics folder**) on your web server.

Step 3: Configure IIS

1. Create a new Site named `ININApps` in IIS:
   a. Right-click on **Sites** and choose **Add web site**.
   b. In the dialog box, set the **Content Directory** - **Physical path** to the CIC Web Applications folder you previously created in your server's `Home` directory.



2. Remove the .NET Framework version of the application pool:
   a. In the **IIS Manager** side pane, click **Application Pools**.
   b. Right-click the newly created **ININApps** application pool.
   c. Click **Basic Settings**.
   d. Change the .NET Framework version to **No Managed Code**.
   e. Click **OK**.
3. **Enable static content compression** on the new Site:
   a. Click the site in **IIS Manager**.
   b. Double-click the **Compression** module.
   c. Check **Enable static content compression**.
   d. Click **Apply**.
4. Update the **Maximum URL Length** and **Maximum Query String** size in **Request Filtering**, if enabled:
   a. Click the site in the **IIS Manager**.
   b. Double-click on the **Request Filtering** module, if enabled.
   If the module does not appear, **Request Filtering** is not enabled.
   c. Select the **URL** tab in the **Request Filtering** view.
   d. Click on **Edit Feature Settings** in the **Actions** pane.
      i. Update **Maximum URL Length (bytes)** to `8192`.
      ii. Update **Maximum Query String (bytes)** to `8192`.
      iii. Update **Maximum allowed content length (bytes)** to something greater than or equal to `20971520`.
   e. Click **OK**.
5. Add allowed server variables:
   a. Click the site in the **IIS Manager**.
   b. Double-click on the **URL Rewrite** module.
   c. In the **Actions** pane, click **View Server Variables**.
   d. Create the following three server variables by clicking **Add** in the **Actions** pane.
      - **WEB_APP**
      - **ICWS_HOST**
      - **HTTP_ININ-ICWS-Original-URL**

> **Note:**
> Steps 6 through 10 can alternatively be completed using XML configuration files.

6. Create the rewrite map.
   a. Click the site in the **IIS Manager**.
   b. Double-click the **URL Rewrite** module.
   c. In the **Actions** pane on the right, click **View Rewrite Maps**.
   d. Click **Add Rewrite Map**.
   e. Enter `MapScheme` for the rewrite map name.
   f. In the **Actions** pane, click **Add Mapping Entry**.
   g. Enter the following:

Original value	New value
on	https

   h. Repeat steps f and g with the following information:

Original value	New value
off	http

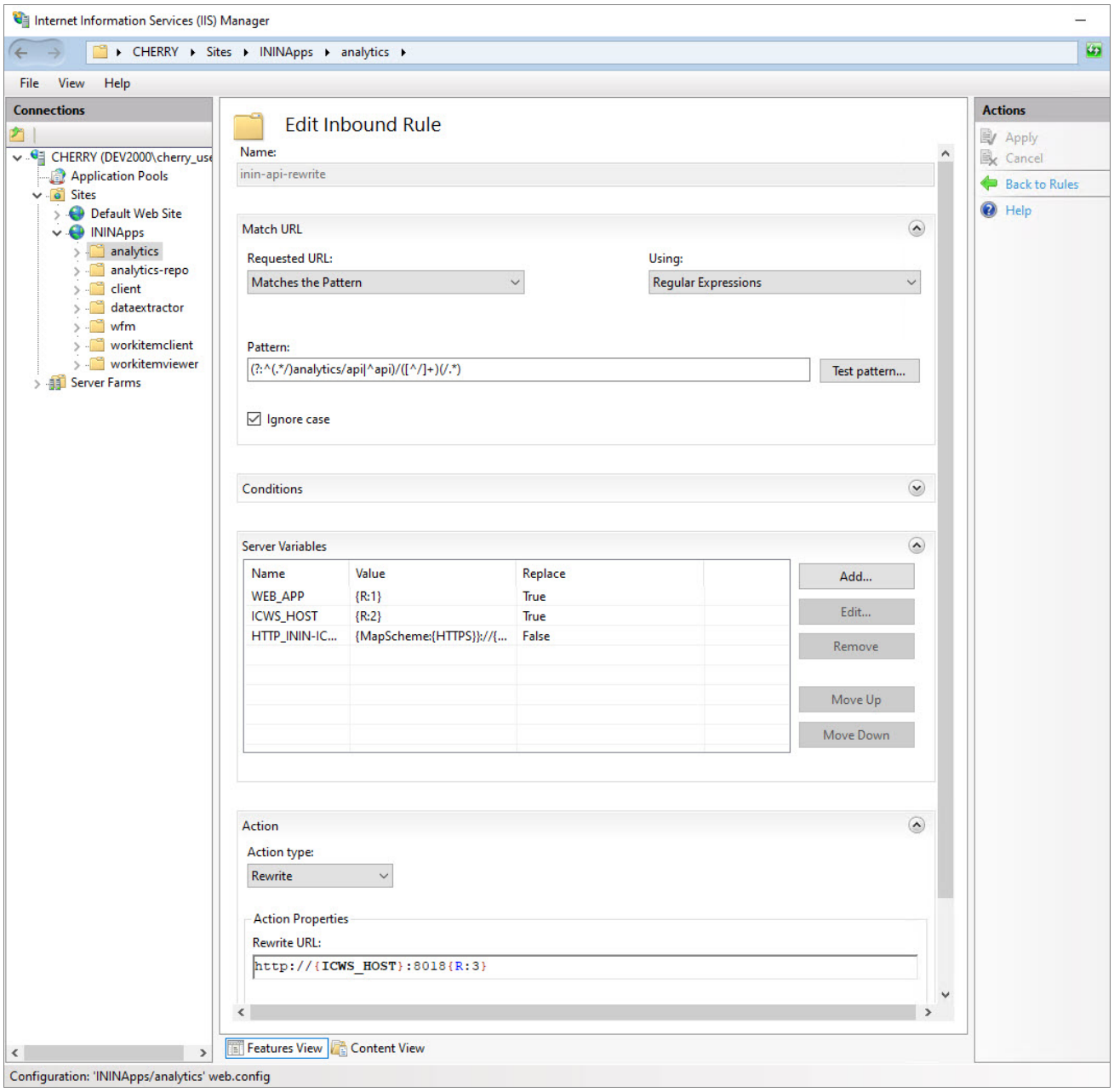
7. Create URL rewrite rules. You will create two inbound rules and four outbound rules.
   a. Click the site in the **IIS Manager**.
   b. Double-click the **URL Rewrite** module.
   c. Navigate to the **Actions** pane and select **Add Rule(s)**.
   d. For each rule, select **Blank rule** under the appropriate type (**Inbound rule** or **Outbound rule**).
   e. Enter the following information for each rule. Tables are provided for ease of copying values, followed by screenshots for each rule.

> **Note:**
> Do not add conditions for any of the rules.

Inbound rule1 This rule allows the client to reach the Session Manager host that ICWS is served from.	
Name>	inin-api-rewrite
Requested URL	Matches the Pattern
Using	Regular Expressions
Pattern	(?:^(.*/)analytics/api|^api)/([^/]+)(/.*)
Ignore case	Enabled
Server Variables	See Server Variables table below
Action type	Rewrite
Rewrite URL (see Configure HTTPS for IIS for HTTPS)	http://{ICWS_HOST}:8018{R:3}
Append query string	Enabled
Log rewritten URL	Enabled
Stop processing of subsequent rules	Enabled

Server Variables

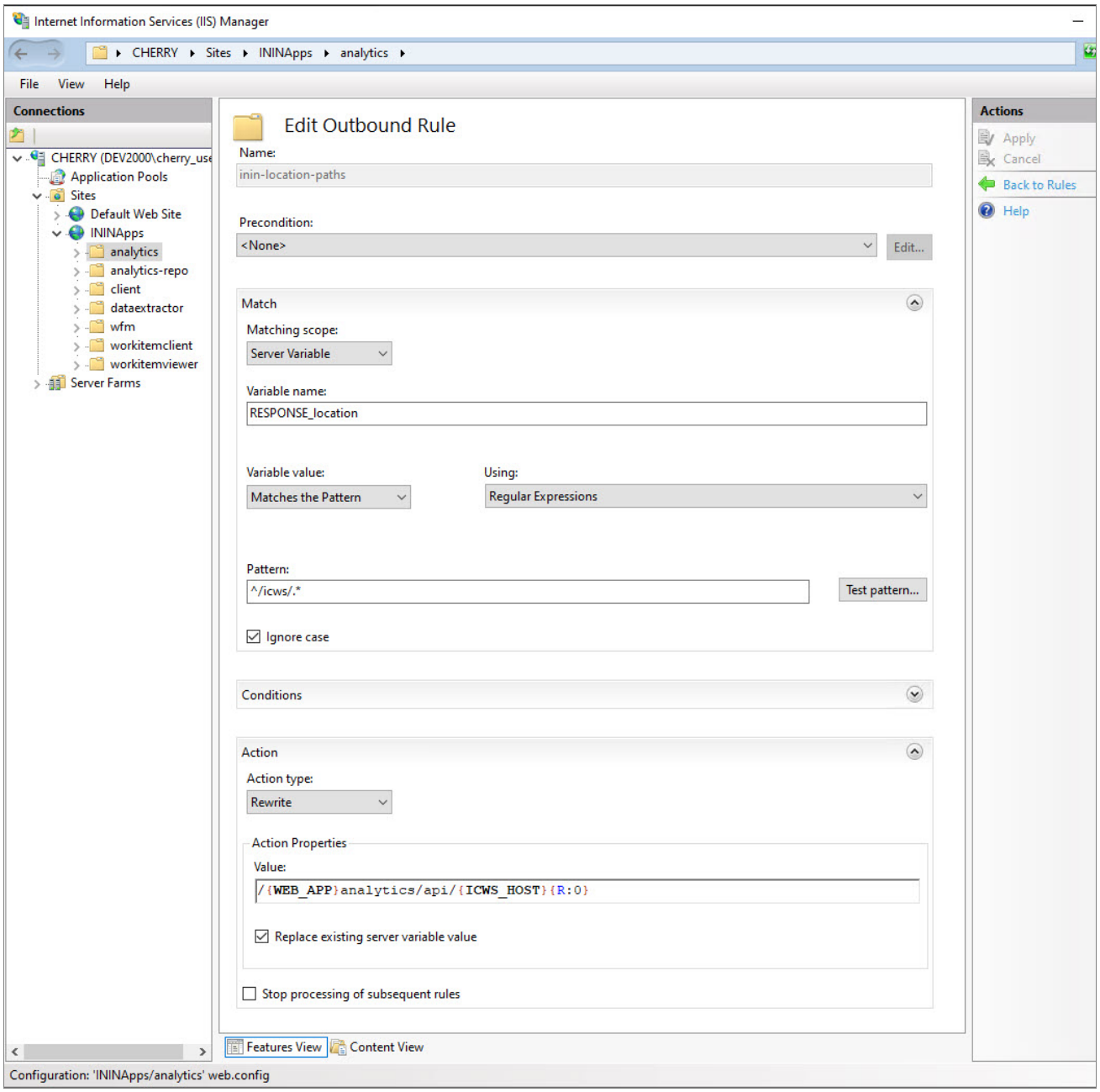
Name	Value	Replace
WEB_APP	{R:1}	True
ICWS_HOST	{R:2}	True
HTTP_ININ-ICWS-Original-URL	{MapScheme:{HTTPS}}://{HTTP_HOST}{UNENCODED_URL}	False



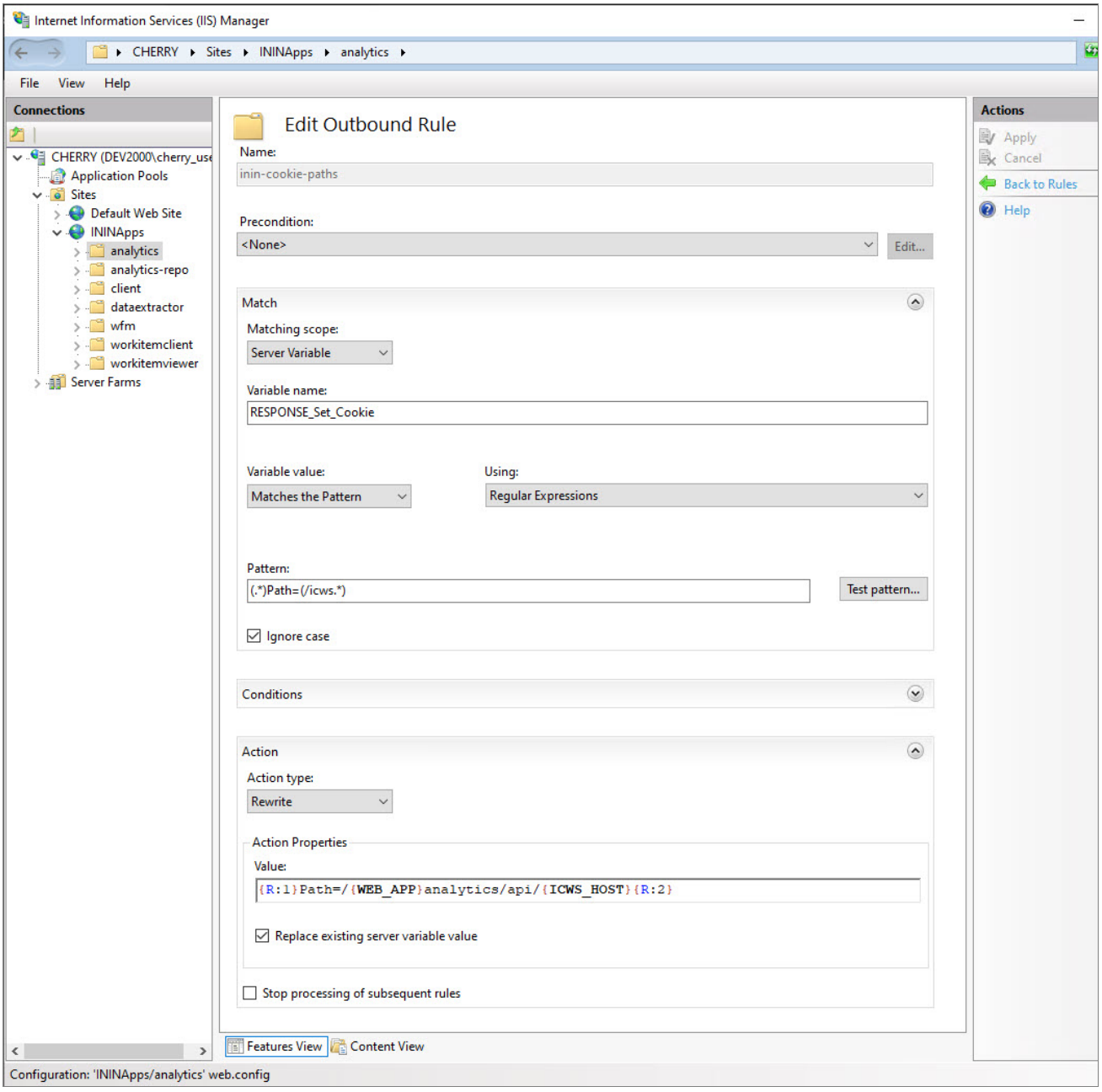
Inbound rule2 This rule allows the client to reach the Session Manager host that Microstrategy calls is served from.	
Name	analytics-route
Requested URL	Matches the Pattern
Using	Regular Expressions
Pattern	(?:^(.*/)analytics-route|^analytics- route)/([^/]+)(/.*)
Ignore case	Enabled
Server Variables	See Server Variables table below
Action type	Rewrite
Rewrite URL (see Configure HTTPS for IIS for HTTPS)	http://{ICWS_HOST}:8018{R:3}
Append query string	Enabled
Log rewritten URL	Enabled
Stop processing of subsequent rules	Enabled

Server Variables

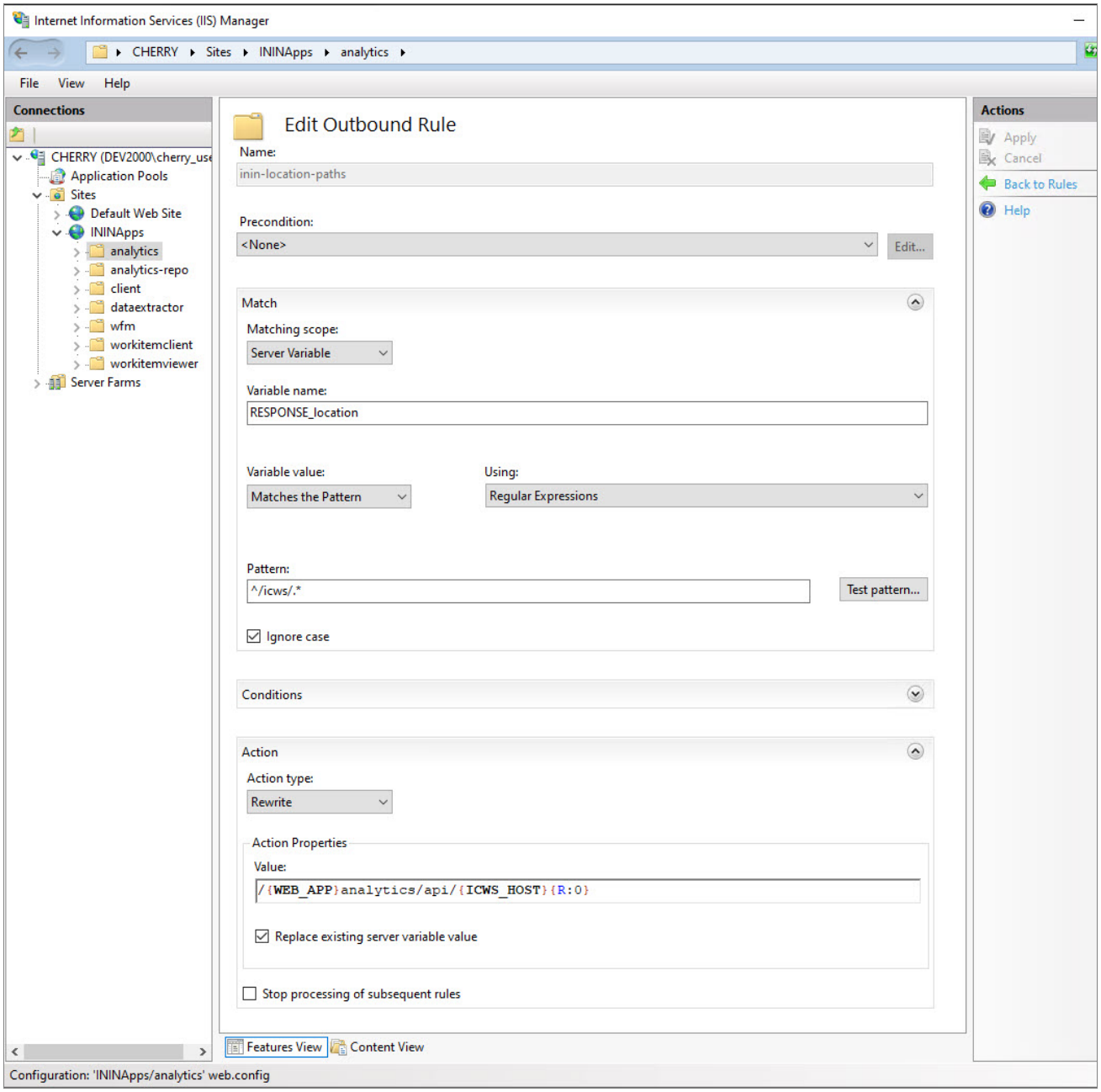
Name	Value	Replace
WEB_APP	{R:1}	True
ICWS_HOST	{R:2}	True
HTTP_ININ-ICWS-Original-URL	{MapScheme:{HTTPS}}://{HTTP_HOST}{UNENCODED_URL}	False



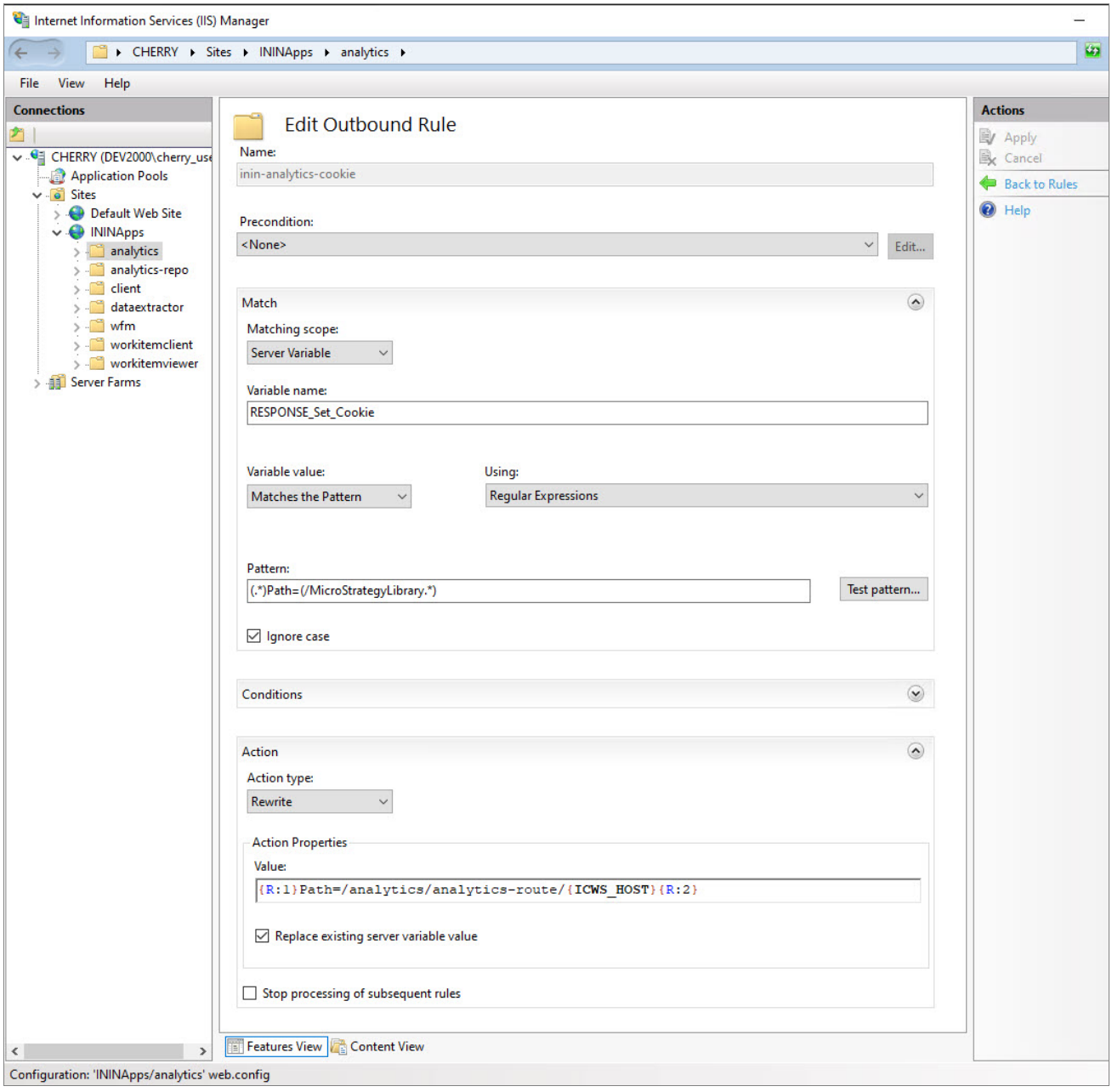
Outbound rule 1 This rule allows the cookies required by ICWS and the client to be located where the client needs them.	
Name	inin-cookie-paths
Precondition	<None>
Matching scope	Server Variable
Variable name	RESPONSE_Set_Cookie
Variable value	Matches the Pattern
Using	Regular Expressions
Pattern	(.*)Path=(/icws.*)
Ignore case	Enabled
Action type	Rewrite
Value	{R:1}Path=/{WEB_APP}analytics/api/{ICWS_HOST}{R:2}
Replace existing server variable value	Enabled
Stop processing of subsequent rules	Disabled



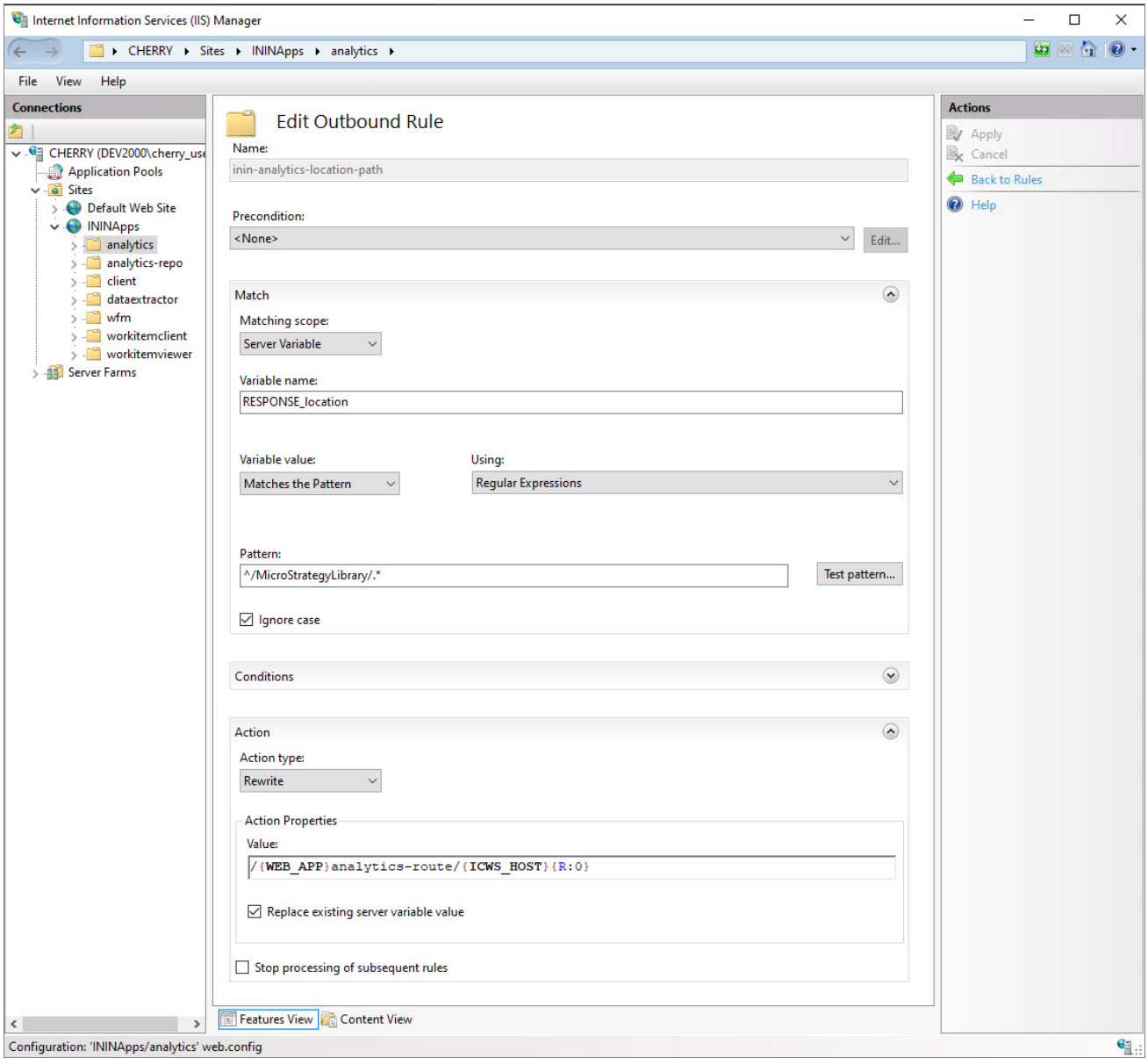
Outbound rule 2 This rule adjusts the location header	
Name	inin-location-paths
Precondition	<None>
Matching scope	Server Variable
Variable name	RESPONSE_location
Variable value	Matches the Pattern
Using	Regular Expressions
Pattern	^/icws/.*
Ignore case	Enabled
Action type	Rewrite
Value	/{WEB_APP}analytics/api/{ICWS_HOS T}{R:0}
Replace existing server value	Enabled
Stop processing of subsequent rules	Disabled



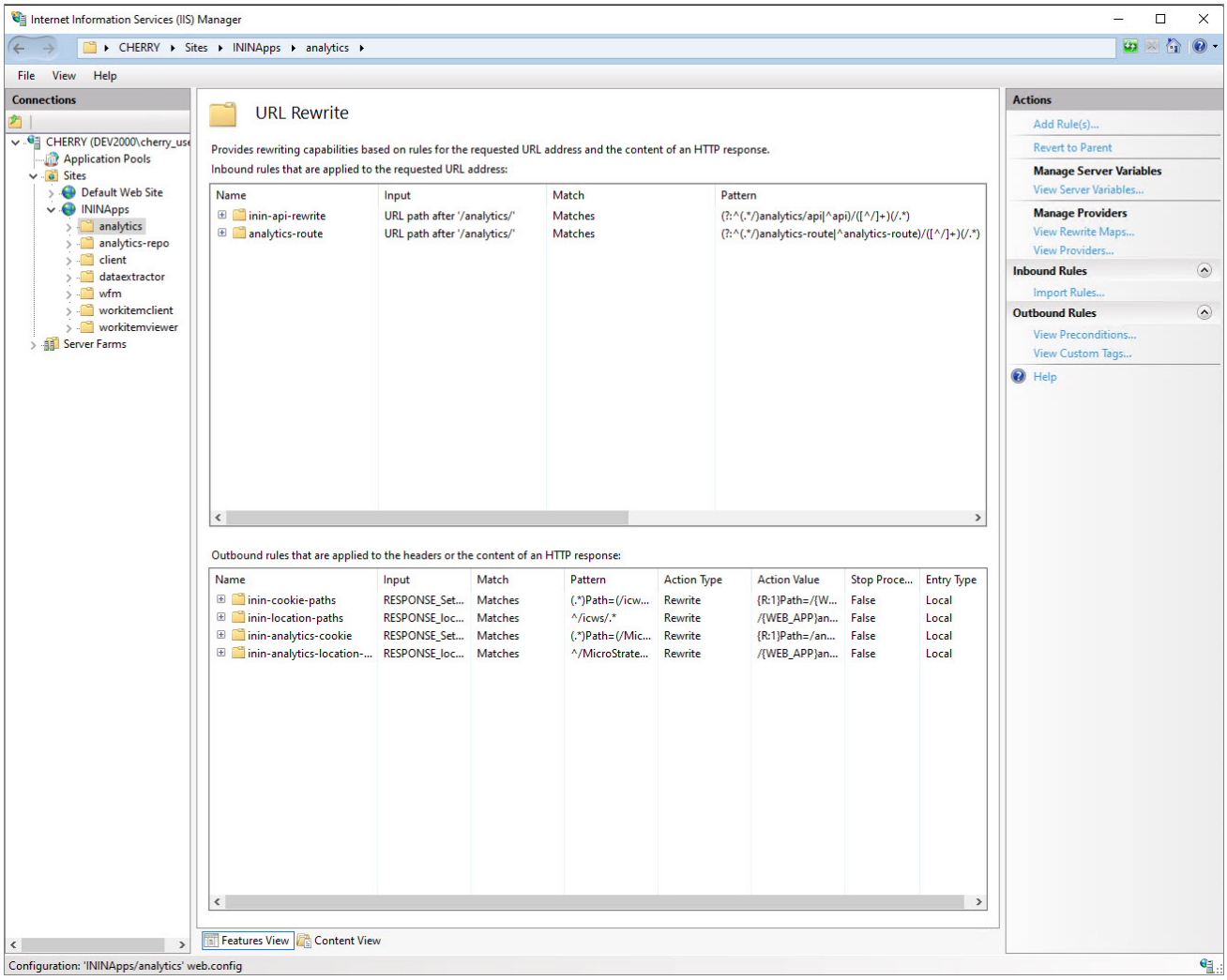
Outbound rule 3 **This rule allows the cookies required by MicroStrategyLibrary and the client to be located where the client needs them.**	
Name	inin-analytics-cookie
Precondition	<None>
Matching scope	Server Variable
Variable name	RESPONSE_Set_Cookie
Variable value	Matches the Pattern
Using	Regular Expressions
Pattern	(.*)Path=(/MicroStrategyLibrary.*)
Ignore case	Enabled
Action type	Rewrite
Value	{R:1}Path=/{WEB_APP}analytics- route/{ICWS_HOST}{R:2}
Replace existing server variable value	Enabled
Stop processing of subsequent rules	Disabled



Outbound rule 4 This rule adjusts the location header	
Name	inin-analytics-location-path
Precondition	<None>
Matching scope	Server Variable
Variable name	RESPONSE_location
Variable value	Matches the Pattern
Using	Regular Expressions
Pattern	^/MicroStrategyLibrary/.*
Ignore case	Enabled
Action type	Rewrite
Value	/{WEB_APP}analytics-route/{ICWS_HOST}{R:0}
Replace existing server value	Enabled
Stop processing of subsequent rules	Disabled



When you are finished, you will have two inbound rules and four outbound rules:



8. (Optional) Increase the cache sensitivity thresholds if you have application load performance issues.
   a. In **Configuration Editor**, select the **system.webServer/serverRuntime** section.
   b. Update **frequentHitThreshold** to **1**.
   c. Update **frequentHitTimePeriod** to **00:10:00**.
9. Enable static content caching for Interaction Connect:
   The following table summarizes the cache settings. Steps to configure cache settings follow.

> **Note:**
> **Client/addins** and **client/config** do not exist in a new installation. If you plan to use `servers.json` or create custom add-ins, use the cache settings below for those folders.

Configure HTTPS for Microsoft IIS

Enable HTTPS between the web browser and IIS

Follow these instructions to encrypt the connection between the web browser and the web server.

Step 1: Add a Certificate to the Web Server

You can use either a *self-signed certificate* or a *third-party certificate*.

If you choose a self-signed certificate, client workstations need to trust that certificate after it is installed on the web server. For this reason, self-signed certificates are usually used for testing only.

To use a third-party certificate, you need to first create a certificate signing request.

Create a self-signed certificate

1. On the web server, open **IIS Manager**.
2. In the **Connections** pane, select the CIC web applications server.
3. Double-click the **Server Certificates** module.
4. In the **Actions** pane, click **Create Self-Signed Certificate**.
5. In the **Create Self-Signed Certificate** window:
   a. Enter a name for the certificate.
   b. Select **Web Hosting** for the certificate store.
6. Click **OK**.

Use a third-party certificate - Generate Certificate Signing Request

1. On the web server, open **IIS Manager**.
2. In the **Connections** pane, select the CIC web applications server.
3. Double-click the **Server Certificates** module.
4. Click **Create Certificate Request** to create a Certificate Signing Request (CSR).
5. In the **Request Certificate** window, enter the information for your organization.

> **Tip:**
> For **Common** name, enter the Fully-Qualified Domain Name (FQDN) of the server, e.g.: `www.example.com`.

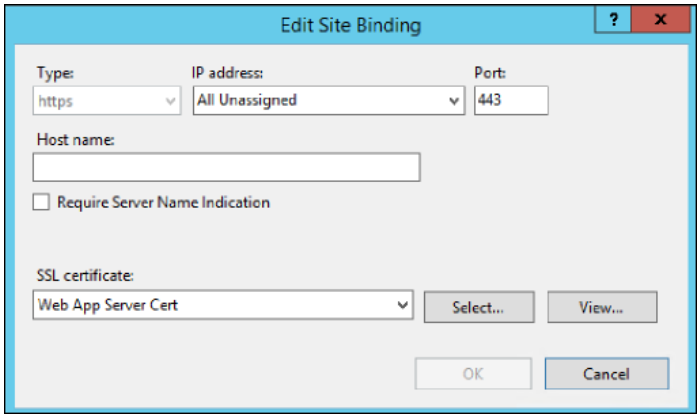
6. Click **Next**.
7. Choose the appropriate cryptographic service provider properties. Ask your third-party Certificate Authority (CA) which options to choose.
8. Click **Next**.
9. Enter a file name and location for the CSR.
10. Click **Finish**.
11. Send the generated CSR to your CA for signing.

Complete certificate request

1. Copy the signed certificate you received from the certificate authority to your web server.
2. In IIS Manager, open the **Server Certificates Module**.
3. Click **Complete Certificate Request**.
4. In the **Specify Certificate Authority Response** window:
   - Select the signed certificate you copied to your web server.
   - Enter a friendly name for the certificate.
   - Select **Web Hosting** for the certificate store.
   - Click **OK**.

Step 2: Bind the certificate to the HTTPS port

1. In the **Connections** pane, click the Site for the CIC Web Applications named **ININApps** in this document.
2. In the **Actions** pane, click **Bindings**.
3. Click **Add**.



4. Change the Type to **https**.
5. In the **SSL certificate** list, select the certificate you previously created or imported.
6. Click **OK**.
7. Click **Close**.

Step 3: Enable SSL on the Site

1. In the **Connections** pane, click the Site for the CIC Web Applications named **ININApps** in this document.
2. Double-click the **SSL Settings** module.
3. Check **Require SSL**.
4. In the **Actions** pane, click **Apply**.

If you used a self-signed certificate, you or the users of client workstations must trust the certificate manually.

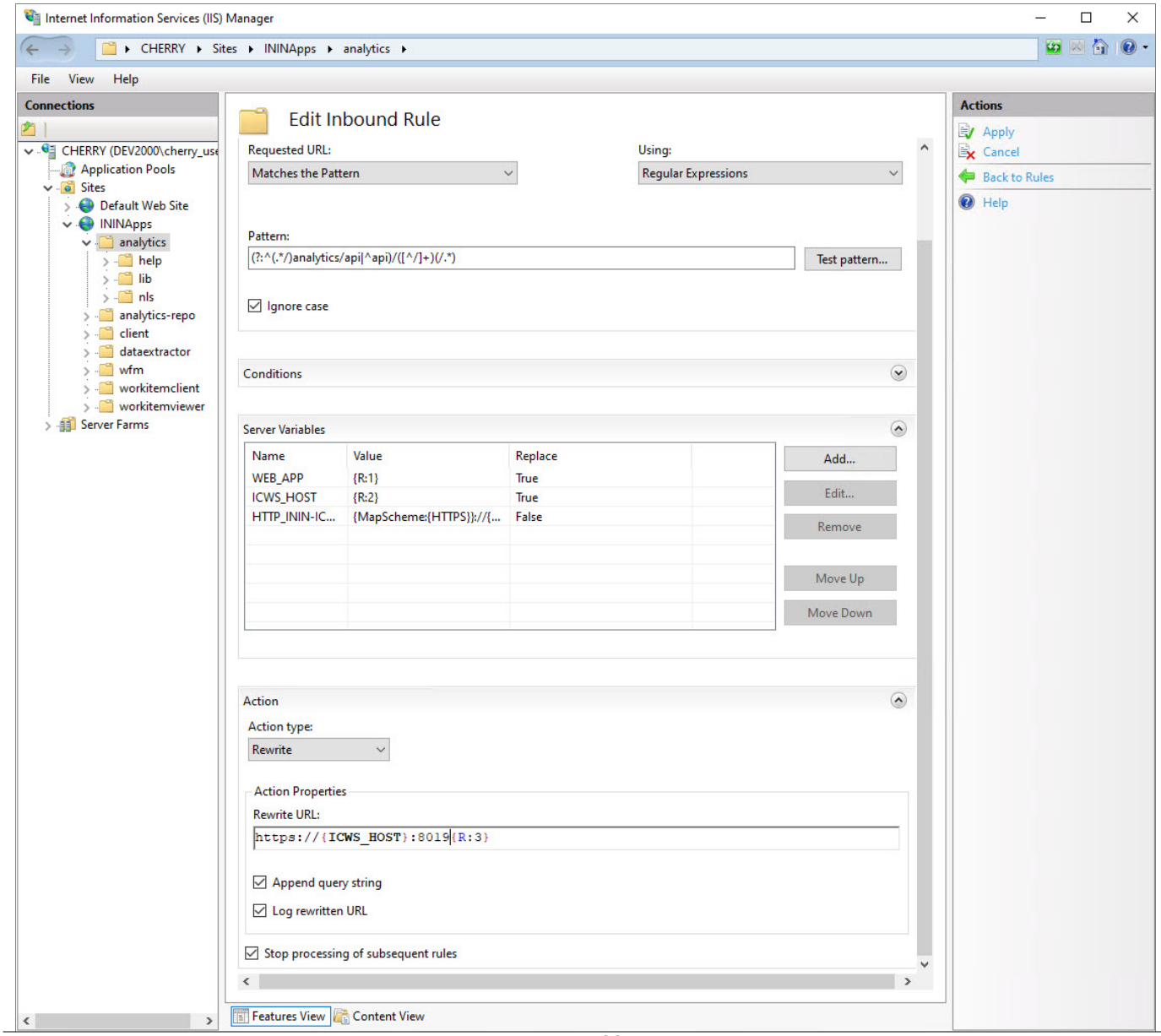
Enable HTTPS between IIS and CIC

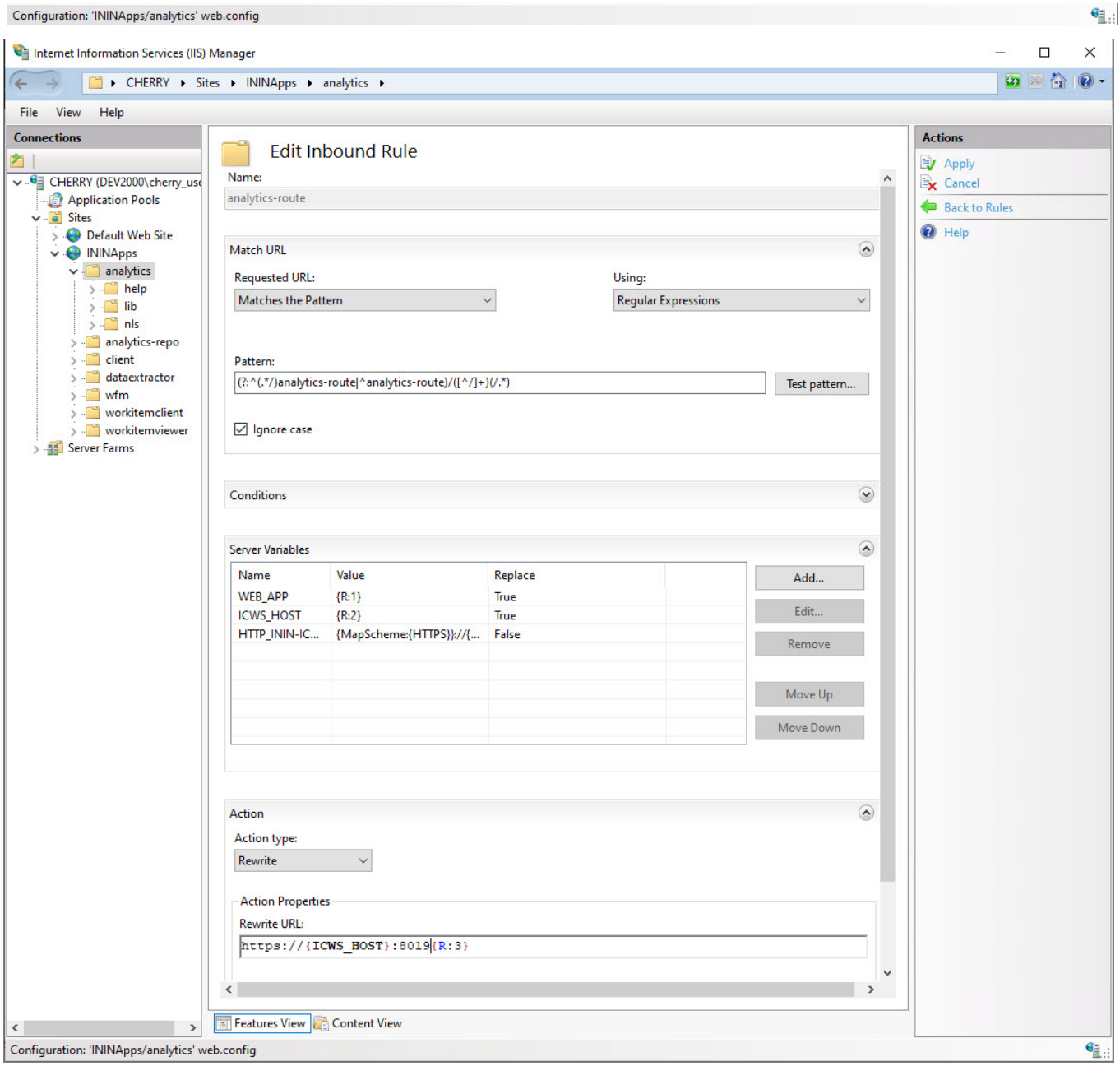
> **Tip:**
> The best practice is to use HTTPS from CIC to IIS and from IIS to the web browser, or from IIS to the web browser only. Securing traffic from IIS to CIC only can cause issues with Secure cookies.

These directions encrypt the connection between the web server and the CIC server.

Step 1: Change Inbound rule to use HTTPS

1. On your web server, open IIS Manager.
2. Expand **Sites**.
3. Select your website, i.e.: ININApps.
4. Double-click the **URL Rewrite** module.
5. Open both the Inbound Rule **inin-api-rewrite** and **analytics-route**.
6. In the **Rewrite URL** field, change the **Rewrite URL** to use **HTTPS** for the two Inbound Rules:
   a. Change the protocol to **https**
   b. Change the port to **8019**.
7. In the **Actions** pane, click **Apply**.


Internet Information Services (IIS) Manager
CHERRY ▸ Sites ▸ ININApps ▸ analytics ▸
File View Help
Connections
CHERRY (DEV2000\cherry_use
Application Pools
Sites
Default Web Site
ININApps
analytics
help
lib
nls
analytics-repo
client
dataextractor
wfm
workitemclient
workitemviewer
Server Farms
Edit Inbound Rule
Requested URL:
Matches the Pattern
Using:
Regular Expressions
Pattern:
(?:^(.*/)analytics/api|^api)/([^/]+)(/.*)
Test pattern...
Ignore case
Conditions
Server Variables
Name
Value
Replace
WEB_APP
{R:1}
True
ICWS_HOST
{R:2}
True
HTTP_ININ-IC...
{MapScheme:{HTTPS}}://{...
False
Add...
Edit...
Remove
Move Up
Move Down
Action
Action type:
Rewrite
Action Properties
Rewrite URL:
https://{ICWS_HOST}:8019{R:3}
Append query string
Log rewritten URL
Stop processing of subsequent rules
Features View
Content View
Actions
Apply
Cancel
Back to Rules
Help



Step 2: Trust the CIC server HTTPS Certificate

Note:
If the `Servername_Certificate.cer` file has a Certificate Chain, then you must trust all the certificates in the chain. Check to see if **Issued To** and **Issued By** are different names. If you do not trust all the certificates in the chain, Session Manager cannot validate the certificate cannot and the SSL handshake will fail. Repeat this task for each Session Manager device in your environment, including both CIC Servers and any Off-Server Session Managers (OSSM).

1. Locate the HTTPS certificate on your CIC server.
   The default location is as follows:
   `\I3\IC\Certificates\HTTPS`
2. Copy **Servername_Certificate.cer** to your web server.
3. On your web server, locate the copied HTTPS certificate.
4. Double-click the certificate.
5. Click **Install Certificate**.
6. Select **Local machine**.
7. Click **Next**.
8. Select **Place all certificates in the following store**.
9. To choose the certificate store, click **Browse** and select **Trusted Root Certification Authorities**.
10. Click **OK**.
11. Click **Next**.
12. Click **Finish**.

Apache HTTP server

Install CX Insights web application for Apache (Only for Analytics)

1. Create a folder in the document root of your web server for the CIC Web Applications.
   Verify that your web server software has the appropriate permissions for that newly created folder.

   Note:
   In this document, the folder is named `ININApps`.
2. Download the CIC web applications zip archive file from https://my.inin.com/products/Pages/Downloads.aspx.
   All the web applications are contained in this single zip archive. You will use only the `Analytics` folder from the zip archive.
3. Unzip the `CIC Web Applications` folder.
4. Navigate to the `web_files` folder inside the unzipped `CIC Web Applications` folder.
5. Copy only `Analytics` folder inside of `web_files`.
6. Paste the `Analytics` folder copied in the previous step into the directory you created in step 1. Doing so places the appropriate directory structure and files for `Analytics` folder on your web server.

Configure HTTP for Apache

1. Download the Apache installer zip archive file (ex: `httpd-2.4.39-win64-VC15.zip`) from the Internet and extract it on `C:` drive.
   It will create a folder similar to `C:\Apache24`.
2. The following actions take place in the Apache server's `/conf/httpd.conf` file. Set the following minimally required modules to be loaded:
   One or more `auth*` modules that are appropriate for your web server
   - `actions_module modules/mod_actions.so`
   - `alias_module modules/mod_alias.so`
   - `allowmethods_module modules/mod_allowmethods.so`
   - `asis_module modules/mod_asis.so`
   - `auth_basic_module modules/mod_auth_basic.so`
   - `authn_core_module modules/mod_authn_core.so`
   - `authn_file_module modules/mod_authn_file.so`
   - `authz_core_module modules/mod_authz_core.so`
   - `authz_groupfile_module modules/mod_authz_groupfile.so`
   - `authz_host_module modules/mod_authz_host.so`
   - `authz_user_module modules/mod_authz_user.so`
   - `autoindex_module modules/mod_autoindex.so`
   - `cgi_module modules/mod_cgi.so`
   - `dir_module modules/mod_dir.so`
   - `env_module modules/mod_env.so`
   - `expires_module modules/mod_expires.so`
   - `headers_module modules/mod_headers.so`
   - `mime_module modules/mod_mime.so`
   - `negotiation_module modules/mod_negotiation.so`
   - `proxy_module modules/mod_proxy.so`
   - `proxy_http_module modules/mod_proxy_http.so`
   - `rewrite_module modules/mod_rewrite.so`
   - `setenvif_module modules/mod_setenvif.so`
3. Change the `DocumentRoot` as well as the single `<Directory>` section to point to the CIC Web Applications folder.
   For example, set—as in this case—the CIC Web Applications folder is extracted in `C:\www`:

   ```
   DocumentRoot "C:/www/"
   <Directory "C:/www">
   ```
4. Change the `DirectoryIndex` property to contain `index.html` and `index.htm`.
5. If `LimitRequestBody` is set to something other then `0`, ensure that you increase it to a value greater than or equal to `20971520` (bytes).
6. Provide the port number on which the web application will be listening.
   Example:

   ```
   Listen 8000
   ServerName localhost:1700
   ```
7. Set up the proxy rewrite rules as follows. Replace `serverName` with the physical name of the server.

   ```
   ServerName {servername}
   RewriteEngine On
   RewriteRule "^(/.*|)analytics/api/([^/]+)([\s\S]*)" "http://$2:8018$3" [P,E=WEB_APP:$1,E=ICWS_HOST:$2,E=ICWS_PATH:$3,E=HTTP_HOST:%
   {HTTP_HOST},E=REQUEST_URI:%{REQUEST_URI},E=SCHEME:%{REQUEST_SCHEME}]
   Header edit Set-Cookie "(.*)Path=(/icws.*)" "$1Path=%{WEB_APP}eanalytics/api/%{ICWS_HOST}e$2"
   Header edit Location "^(/icws.*)" "%{WEB_APP}eanalytics/api/%{ICWS_HOST}e$1"
   SetEnvIf "ININ-ICWS-Original-URL" ".+" HAVE_ININICWSOriginalURL
   RequestHeader set "ININ-ICWS-Original-URL" "%{SCHEME}e://%{HTTP_HOST}e%{REQUEST_URI}e" env=!HAVE_ININICWSOriginalURL
   RewriteRule "^(/.*|)/analytics-route/([^/]+)([\s\S]*)" "http://$2:8018$3" [P,E=WEB_APP:$1,E=ICWS_HOST:$2,E=ICWS_PATH:$3,E=HTTP_HOST:%
   {HTTP_HOST},E=REQUEST_URI:%{REQUEST_URI},E=SCHEME:%{REQUEST_SCHEME}]
   Header edit Set-Cookie "(.*)Path=(/MicroStrategyLibrary.*)" "$1Path=%{WEB_APP}e/analytics-route/%{ICWS_HOST}e$2"
   Header edit Location "^(/MicroStrategyLibrary.*)" "%{WEB_APP}e/analytics-route/%{ICWS_HOST}e$1"
   SetEnvIf "ININ-ICWS-Original-URL" ".+" HAVE_ININICWSOriginalURL
   RequestHeader set "ININ-ICWS-Original-URL" "%{SCHEME}e://%{HTTP_HOST}e%{REQUEST_URI}e" env=!HAVE_ININICWSOriginalURL
   ```
8. Restart the Apache process.
9. Verify that all applications work as expected.

Configure HTTPS for Apache

1. To achieve HTTPS, we need SSL certificate. So, SSL certificate we need to generate via OpenSSL.
   a. Download OpenSSL Windows installer (`Win64OpenSSL-1_1_0k.exe`) from https://slproweb.com/products/Win32OpenSSL.html.
   You can use a more recent version, if available.
   b. Create a directory anywhere (example: `C:\certs`).
   SSL certificate will be generated here.
   c. Open a **Command Prompt** window in Administrator mode and navigate to the directory where SSL certificate will be generated.
   d. Set these configuration variables
      - `set RANDFILE=C:\<directory name>\.rnd`
        Example: `C:\certs\.rnd`
      - `set OPENSSL_CONF=C:\OpenSSL-Win32\bin\openssl.cfg`
        (# as per installation)
   e. In the **Command Prompt** window, enter the following command:
   `"C:\OpenSSL-Win32\bin\openssl.exe" req -out CSR.csr -new -newkey rsa:2048 - nodes -keyout PrivateKey.key`
   f. In the **Command Prompt** window, enter the following command:
   `"C:\OpenSSL-Win32\bin\opensl.exe" x509 -req -days 365 -in CSR.csr -signkey Private.Key -out server.crt`
   g. Verify that the directory contains the following files:
      - `CSR.csr`
      - `PrivateKey.key`
      - `server.crt`
2. Rest of the configuration will be almost same as **HTTP configuration**. Just modify the following steps of **HTTP configuration**
   - At step 2, add module `ssl_module modules/mod_sll.so` for SSL.
   - Add the generated SSL certificate details in server via Apache server's `/conf/httpd.conf` file.

```
<VirtualHost *:{port}>
ServerName {servername}
SSLEngine on
SSLCertificate "C:/certs/server.crt"
SSLCertificateKeyFile "C:/certs/Private.key"
SSLProxyEngine on
RewriteRule "^(/.*|)analytics/api/([^/]+)([\s\S]*)" "http://$2:8018$3"    [P,E=WEB_APP:$1,E=ICWS_HOST:$2,E=ICWS_PATH:$3,E=HTTP_HOST:%
{HTTP_HOST},E=REQUEST_URI:%{REQUEST_URI},E=SCHEME:%{REQUEST_SCHEME}]
Header edit Set-Cookie "(.*)Path=(/icws.*)" "$1Path=%{WEB_APP}eanalytics/api/%{ICWS_HOST}e$2"
Header edit Location "^(/icws.*)" "%{WEB_APP}eanalytics/api/%{ICWS_HOST}e$1"
SetEnvIf "ININ-ICWS-Original-URL" ".+" HAVE_ININICWSOriginalURL
RequestHeader set "ININ-ICWS-Original-URL" "%{SCHEME}e://%{HTTP_HOST}e%{REQUEST_URI}e" env=!HAVE_ININICWSOriginalURL
RewriteRule "^(/.*|)/analytics-route/([^/]+)([\s\S]*)" "http://$2:8018$3" [P,E=WEB_APP:$1,E=ICWS_HOST:$2,E=ICWS_PATH:$3,E=HTTP_HOST:%
{HTTP_HOST},E=REQUEST_URI:%{REQUEST_URI},E=SCHEME:%{REQUEST_SCHEME}]
Header edit Set-Cookie "(.*)Path=(/MicroStrategyLibrary.*)" "$1Path=%{WEB_APP}e/analytics-route/%{ICWS_HOST}e$2"
Header edit Location "^(/MicroStrategyLibrary.*)" "%{WEB_APP}e/analytics-route/%{ICWS_HOST}e$1"
SetEnvIf "ININ-ICWS-Original-URL" ".+" HAVE_ININICWSOriginalURL
RequestHeader set "ININ-ICWS-Original-URL" "%{SCHEME}e://%{HTTP_HOST}e%{REQUEST_URI}e" env=!HAVE_ININICWSOriginalURL
</VirtualHost>
```

   - In the above rule, locate `SSLCertificateFile` and `SSLCertificateKeyFile` and edit them as per your certificate name and location.
   - Set up the proxy rewrite rules as follows. Replace `serverName` with physical name of server.

```
ServerName {servername}
RewriteEngine On
RewriteRule "^(/.*|)analytics/api/([^/]+)([\s\S]*)" "https://$2:8019$3" [P,E=WEB_APP:$1,E=ICWS_HOST:$2,E=ICWS_PATH:$3,E=HTTP_HOST:%
{HTTP_HOST},E=REQUEST_URI:%{REQUEST_URI},E=SCHEME:%{REQUEST_SCHEME}]
Header edit Set-Cookie "(.*)Path=(/icws.*)" "$1Path=%{WEB_APP}eanalytics/api/%{ICWS_HOST}e$2"
Header edit Location "^(/icws.*)" "%{WEB_APP}eanalytics/api/%{ICWS_HOST}e$1"
SetEnvIf "ININ-ICWS-Original-URL" ".+" HAVE_ININICWSOriginalURL
RequestHeader set "ININ-ICWS-Original-URL" "%{SCHEME}e://%{HTTP_HOST}e%{REQUEST_URI}e" env=!HAVE_ININICWSOriginalURL
RewriteRule "^(/.*|)/analytics-route/([^/]+)([\s\S]*)" "https://$2:8019$3" [P,E=WEB_APP:$1,E=ICWS_HOST:$2,E=ICWS_PATH:$3,E=HTTP_HOST:%
{HTTP_HOST},E=REQUEST_URI:%{REQUEST_URI},E=SCHEME:%{REQUEST_SCHEME}]
Header edit Set-Cookie "(.*)Path=(/MicroStrategyLibrary.*)" "$1Path=%{WEB_APP}e/analytics-route/%{ICWS_HOST}e$2"
Header edit Location "^(/MicroStrategyLibrary.*)" "%{WEB_APP}e/analytics-route/%{ICWS_HOST}e$1"
SetEnvIf "ININ-ICWS-Original-URL" ".+" HAVE_ININICWSOriginalURL
RequestHeader set "ININ-ICWS-Original-URL" "%{SCHEME}e://%{HTTP_HOST}e%{REQUEST_URI}e" env=!HAVE_ININICWSOriginalURL
```

   - Restart the Apache process.
   - Verify that all applications work as expected.

Nginx Server

Install CX Insights web application for Nginx

1. Create a folder in the document root of your web server for the CIC Web Applications.
   Verify that your web server software has the appropriate permissions for that newly created folder.

   > **Note:**
   > In this document, the folder is named `ININApps`.

2. Download the CIC web applications zip archive file from https://my.inin.com/products/Pages/Downloads.aspx.
   All the web applications are contained in this single zip. You will use only the `Analytics` folder from the zip.
3. Unzip the `CIC Web Applications` folder.
4. Navigate to the `web_files` folder inside the unzipped `CIC Web Applications` folder.
5. Copy only `Analytics` folder inside of `web_files`.
6. Paste the `Analytics` folder copied in the previous step into the directory you created in step 1. Doing so places the appropriate directory structure and files for `Analytics` folder on your web server.

Configure HTTP for Nginx

1. Enter the `Nginx.config` information and then change the following:

```
location ~ /client/ {
location ~ /client/help/ {
expires off;
}
location ~ /client/(?:addins|config)/ {
add_header Cache-Control "no-cache";
}
location ~ index.html?$ {
expires 15m;
}
location ~ .(?:js|css|jpe?g|ico|png|gif|svg|ttf|woff|otf|eot|mp3|wav|ogg)$
//eic/2019r2_systest/products/documentation/source/Technical_Reference_HTML/cic_web_applications_icg/Install_CIC_Web_Applications_on_Nginx.htm#2
{
```

```
expires 1y;
}
}
```

a. In the Resolver field, use the DNS server instead of `dl-hq-dc01.ininlab.com`
b. In the upstream object for Server field, use the IC server name instead of `adonis.dev2000.com`.
c. Change the port 8070 to the custom port under server object.
d. In the server object, for `server_name` use the proxy server name instead of `eros.dev2000.com`
e. Set the root entry for the server to the CIC Web Applications folder under location object.
f. Enter the content for cache rules within the server object, given in `nginx_cache.conf`.

```
#user  nobody;
worker_processes  2;
#error_log  logs/error.log;
#error_log  logs/error.log  notice;
#error_log  logs/error.log  info;
#pid        logs/nginx.pid;
events {
    worker_connections  1024;
}
http {
resolver  dl-hq1-dc01.ininlab.com valid=90000000s;
    include       mime.types;
    default_type  application/octet-stream;
default_type  application/json;
    #log_format  main  '$remote_addr - $remote_user [$time_local] "$request" '
    #                  '$status $body_bytes_sent "$http_referer" '
    #                  '"$http_user_agent" "$http_x_forwarded_for"';
    #access_log  logs/access.log  main;
    sendfile        on;
    #tcp_nopush     on;
    keepalive_timeout  60;

    gzip  on;
gzip_types                text/plain
#eic/2019r2_systest/products/documentation/source/Technical_Reference_HTML/cic_web_applications_icg/Install_CIC_Web_Applications_on_Nginx.htm#2
text/css application/javascript application/json image/svg+xml;
index                     index.html index.htm;
#eic/2019r2_systest/products/documentation/source/Technical_Reference_HTML/cic_web_applications_icg/Install_CIC_Web_Applications_on_Nginx.htm#2
client_max_body_size      0;
autoindex                 on;

upstream up {
server adonis.dev2000.com:8018;
keepalive 100;
}
    server {
        listen       8070;
listen       [::]:8070;
server_name  eros.dev2000.com;
server_name  127.0.0.1;
        #charset koi8-r;
        #access_log  logs/host.access.log  main;
        location / {
root   ../www;
            index  index.html index.htm;
        }
        #error_page  404              /404.html;
        # redirect server error pages to the static page /50x.html
        #
        #error_page   500 502 503 504  /50x.html;
        #location = /50x.html {
        #    root   html;
        #}
        # proxy the PHP scripts to Apache listening on 127.0.0.1:80
        #
        #location ~ \.php$ {
        #    proxy_pass   http://127.0.0.1;
        #}
        # pass the PHP scripts to FastCGI server listening on 127.0.0.1:9000
        #
        #location ~ \.php$ {
        #    root           html;
        #    fastcgi_pass   127.0.0.1:9000;
        #    fastcgi_index  index.php;
        #    fastcgi_param  SCRIPT_FILENAME  /scripts$fastcgi_script_name;
        #    include        fastcgi_params;
        #}
        # deny access to .htaccess files, if Apache's document root
        # concurs with nginx's one
        #
        #location ~ /\.ht {
        #    deny  all;
        #}
set $ininIcwsOriginalUrl $http_inin_icws_original_url;
if ($ininIcwsOriginalUrl !~ .+) {
set $ininIcwsOriginalUrl $scheme://$http_host$request_uri;
}
location ~* (?:^(.+)analytics/api|^/api)/([^/]+)(/.+)$ {
set $web_app $1;
set $server $2;
set $icws_path $3;

proxy_read_timeout          600;
proxy_cookie_path /icws/ ${web_app}analytics/api/$server/icws/;
proxy_redirect /icws/ ${web_app}analytics/api/$server/icws/;

proxy_pass  http://up$icws_path$is_args$args;
proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
proxy_set_header ININ-ICWS-Original-URL $ininIcwsOriginalUrl;
proxy_http_version 1.1;
proxy_set_header Connection "";
proxy_set_header Host $host;
add_header P3P "CP=`CAO PSA OUR`";

}
if ($ininIcwsOriginalUrl !~ .+) {
set $ininIcwsOriginalUrl $scheme://$http_host$request_uri;
}
location ~* (?:^(.+)/analytics-route|^/analytics-route)/([^/]+)(/.+)$ {
```

```
        set $web_app $1;
        set $server $2;
        set $icws_path $3;

        proxy_read_timeout          600;
        proxy_cookie_path /MicroStrategyLibrary/ $web_app/analytics-route/$server/MicroStrategyLibrary/;
        proxy_redirect ^(/MicroStrategyLibrary.*/) $web_app/analytics-route/$server/$1;

        proxy_pass  http://up$icws_path$is_args$args;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header ININ-ICWS-Original-URL $ininIcwsOriginalUrl;
        proxy_http_version 1.1;
        proxy_set_header Connection "";
        proxy_set_header Host $host;
        add_header P3P "CP=`CAO PSA OUR`";
        add_header P3P "CP=`CAO PSA OUR`";
        }
            }
            # another virtual host using mix of IP-, name-, and port-based configuration
            #
            #server {
            #    listen       8000;
            #    listen       somename:8080;
            #    server_name  somename  alias  another.alias;
            #    location / {
            #        root   html;
            #        index  index.html index.htm;
            #    }
            #}
            # HTTPS server
            #
            #server {
            #    listen       443 ssl;
            #    server_name  localhost;
            #    ssl_certificate      cert.pem;
            #    ssl_certificate_key  cert.key;
            #    ssl_session_cache    shared:SSL:1m;
            #    ssl_session_timeout  5m;
            #    ssl_ciphers  HIGH:!aNULL:!MD5;
            #    ssl_prefer_server_ciphers  on;
            #    location / {
            #        root   html;
            #        index  index.html index.htm;
            #    }
            #}
        }
```

g. Restart the Nginx process.
h. Verify that all applications work as expected.

Configure HTTPS for Nginx

1. To achieve HTTPS, we need SSL certificate. So, SSL certificate we need to generate via OpenSSL.
   a. Download OpenSSL Windows installer (Win64OpenSSL-1_1_0k.exe) from this link https://slproweb.com/products/Win32OpenSSL.html. If latest installer is available that can be considered too.
   b. Create a directory anywhere (Ex: C:\certs). SSL certificate will be generated here.
   c. Open Command Prompt via Administrative mode and navigate to the directory where SSL certificate will be generated.
   d. Set these configuration variable

- Set RANDFILE=C:\<directory name> \.rnd (Ex: C:\certs\.rnd. Modify your location accordingly)
- Set OPENSSL_CONF=C:\OpenSSL-Win32\bin\openssl.cfg (# As per installation)

a. Enter "C:\OpenSSL-Win32\bin\openssl.exe" req -out CSR.csr -new -newkey rsa:2048 -nodes -keyout PrivateKey.key via Command Prompt
b. Enter "C:\OpenSSL-Win32\bin\openssl.exe" x509 -req -days 365 -in CSR.csr -signkey PrivateKey.key -out server.crt via Command Prompt.
c. The directory should contain CSR.csr, PrivateKey.key and server.crt.

- The following configuration is similar to HTTP configuration. Change the following to configure:

```
        #user  nobody;
        worker_processes  2;
        #error_log  logs/error.log;
        #error_log  logs/error.log  notice;
        #error_log  logs/error.log  info;
        #pid        logs/nginx.pid;
        events {
            worker_connections  1024;
        }
        http {
        resolver  dl-hq1-dc01.ininlab.com valid=90000000s;
            include       mime.types;
            default_type  application/octet-stream;
        #default_type  application/json;
            #log_format  main  '$remote_addr - $remote_user [$time_local] "$request" '
            #                  '$status $body_bytes_sent "$http_referer" '
            #                  '"$http_user_agent" "$http_x_forwarded_for"';
            #access_log  logs/access.log  main;
            sendfile        on;
            #tcp_nopush     on;
            keepalive_timeout  60;
            gzip  on;
        gzip_types              text/plain
        #eic/2019r2_systest/products/documentation/source/Technical_Reference_HTML/cic_web_applications_icg/Install_CIC_Web_Applications_on_Nginx.htm#2
        text/css application/javascript application/json image/svg+xml;
        index                   index.html index.htm;
        #eic/2019r2_systest/products/documentation/source/Technical_Reference_HTML/cic_web_applications_icg/Install_CIC_Web_Applications_on_Nginx.htm#2
        client_max_body_size     0;
        autoindex               on;
        upstream up {
        server adonis.dev2000.com:8018;
        keepalive 100;
        }
            server {
                listen       8070;
        listen       [::]:8070;
                #server_name  localhost;
        server_name  eros.dev2000.com;
        server_name  127.0.0.1;
                #charset koi8-r;
                #access_log  logs/host.access.log  main;
```

```
        location / {
            #root   html;
#root   "C://www//analytics";
root   ../www;
            index  index.html index.htm;
        }
        #error_page  404              /404.html;
        # redirect server error pages to the static page /50x.html
        #
        #error_page   500 502 503 504  /50x.html;
        #location = /50x.html {
        #    root   html;
        #}
        # proxy the PHP scripts to Apache listening on 127.0.0.1:80
        #
        #location ~ \.php$ {
        #    proxy_pass   http://127.0.0.1;
        #}
        # pass the PHP scripts to FastCGI server listening on 127.0.0.1:9000
        #
        #location ~ \.php$ {
        #    root           html;
        #    fastcgi_pass   127.0.0.1:9000;
        #    fastcgi_index  index.php;
        #    fastcgi_param  SCRIPT_FILENAME  /scripts$fastcgi_script_name;
        #    include        fastcgi_params;
        #}
        # deny access to .htaccess files, if Apache's document root
        # concurs with nginx's one
        #
        #location ~ /\.ht {
        #    deny  all;
        #}
set $ininIcwsOriginalUrl $http_inin_icws_original_url;
if ($ininIcwsOriginalUrl !~ .+) {
set $ininIcwsOriginalUrl $scheme://$http_host$request_uri;
}
location ~* (?:^(.+)analytics/api|^/api)/([^/]+)(/.+)$ {
set $web_app $1;
set $server $2;
set $icws_path $3;
proxy_read_timeout          600;
proxy_cookie_path /icws/ ${web_app}analytics/api/$server/icws/;
proxy_redirect /icws/ ${web_app}analytics/api/$server/icws/;
proxy_pass  http://up$icws_path$is_args$args;
proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
proxy_set_header ININ-ICWS-Original-URL $ininIcwsOriginalUrl;
proxy_http_version 1.1;
proxy_set_header Connection "";
proxy_set_header Host $host;
add_header P3P "CP=`CAO PSA OUR`";
}
#set $ininIcwsOriginalUrl $http_inin_icws_original_url;
if ($ininIcwsOriginalUrl !~ .+) {
set $ininIcwsOriginalUrl $scheme://$http_host$request_uri;
}
location ~* (?:^(.+)/analytics-route|^/analytics-route)/([^/]+)(/.+)$ {
set $web_app $1;
set $server $2;
set $icws_path $3;
proxy_read_timeout          600;
proxy_cookie_path /MicroStrategyLibrary/ $web_app/analytics-route/$server/MicroStrategyLibrary/;
proxy_redirect /MicroStrategyLibrary/ ${web_app}analytics-route/$server/MicroStrategyLibrary/;
proxy_pass  http://up$icws_path$is_args$args;
proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
proxy_set_header ININ-ICWS-Original-URL $ininIcwsOriginalUrl;
proxy_http_version 1.1;
proxy_set_header Connection "";
proxy_set_header Host $host;
add_header P3P "CP=`CAO PSA OUR`";
add_header P3P "CP=`CAO PSA OUR`";
}
    }
    # another virtual host using mix of IP-, name-, and port-based configuration
    #
    #server {
    #    listen       8000;
    #    listen       somename:8080;
    #    server_name  somename  alias  another.alias;
    #    location / {
    #        root   html;
    #        index  index.html index.htm;
    #    }
    #}
    # HTTPS server
    #
    #server {
    #    listen       443 ssl;
    #    server_name  localhost;
    #    ssl_certificate      cert.pem;
    #    ssl_certificate_key  cert.key;
    #    ssl_session_cache    shared:SSL:1m;
    #    ssl_session_timeout  5m;
    #    ssl_ciphers  HIGH:!aNULL:!MD5;
    #    ssl_prefer_server_ciphers  on;
    #    location / {
    #        root   html;
    #        index  index.html index.htm;
    #    }
    #}
}
```

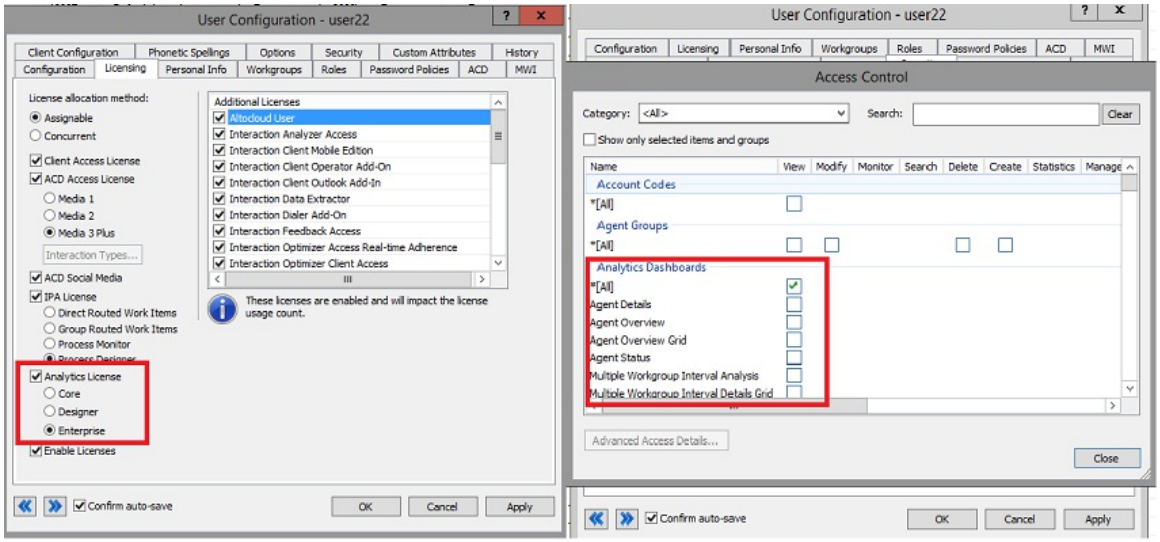
a. In 'resolver' field instead of dl-hq1-dc01.ininlab.com, use the DNS server.
b. Change port 8071 to custom port and provide 'SSL' binding beside o port number under server object.
c. In server object for 'server_name' field instead of eros.dev2000.com, use the proxy server name.
d. Enter the ssl_certificate & ssl_certificate_key under server object (Ex : "C:\certs\server.crt" & "C:\certs\PrivateKey.key" respectively)
e. Set the root entry for the server to the CIC Web Applications folder under location object.
f. Under location object, for proxy_pass instead of http use https and replace 8018 with 8019.
g. Under location object, add proxy_buffering off;

h. Restart the Nginx process.
i. Verify that all applications work as expected.
j. Enter the content for cache rules within the server object, given in nginx_cache.conf.

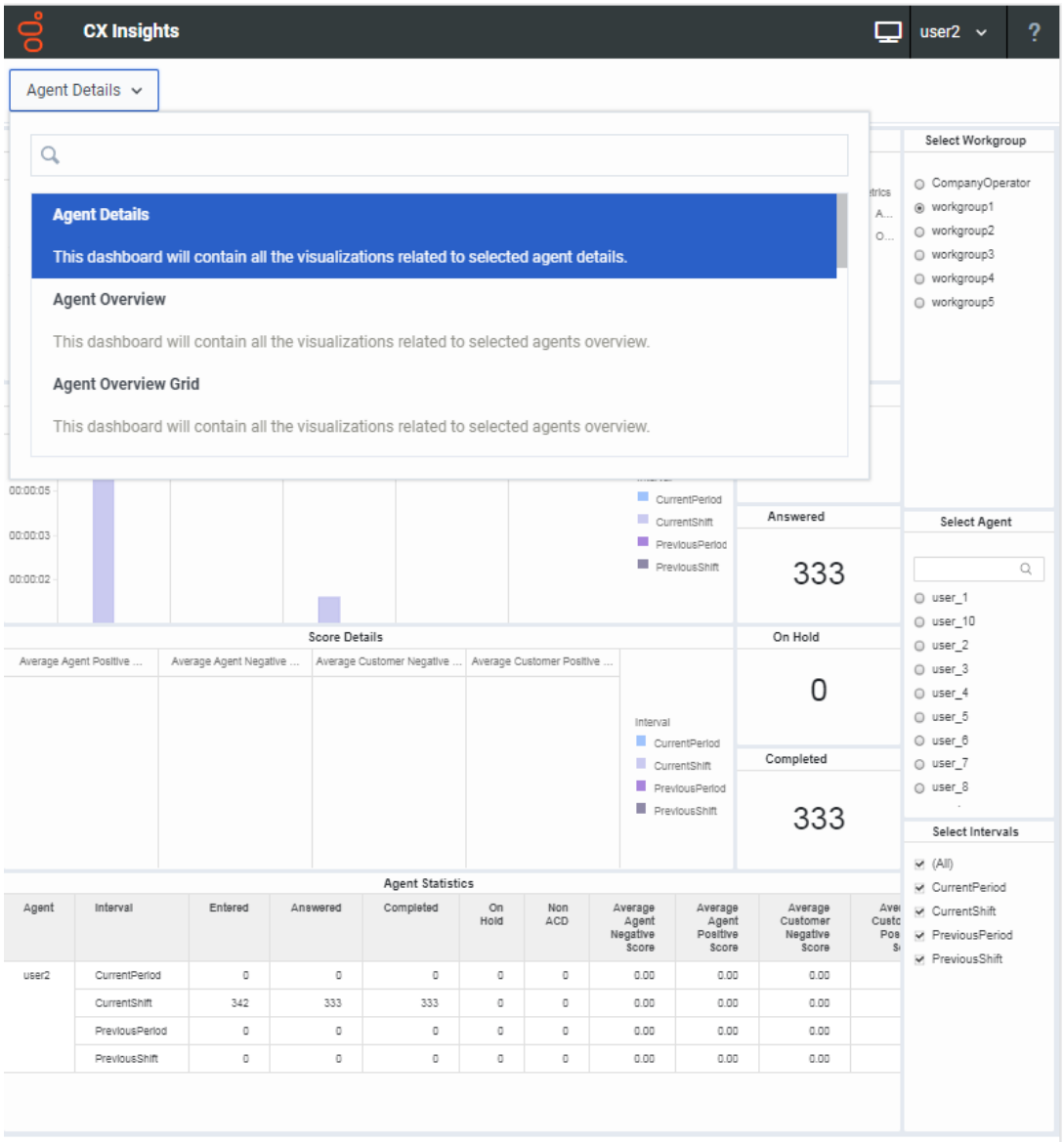
```
location ~ /client/ {
location ~ /client/help/ {
expires off;
}
location ~ /client/(?:addins|config)/ {
add_header Cache-Control "no-cache";
}
location ~ index.html?$ {
expires 15m;
}
location ~ .(?:js|css|jpe?g|ico|png|gif|svg|ttf|woff|otf|eot|mp3|wav|ogg)$
//eic/2019r2_systest/products/documentation/source/Technical_Reference_HTML/cic_web_applications_icg/Install_CIC_Web_Applications_on_Nginx.htm#2
{
expires 1y;
}
}
```

View CX Insights dashboards

You can log in to CX Insights web application with the same PureConnect web application credentials only if you have one of the licenses defined for the analytics feature.



You can select the dashboard from the drop-down selection list as shown in the following image. The list shows the dashboards for which you have access permissions defined in the CIC server. After successful loading, the dashboard refreshes every 30 seconds with real-time statistic values.



Change Log

The following table lists the changes to this document since its initial release.

>>>> ORIGINAL //eic/2019r4_systest/products/documentation/source/Technical_Reference_HTML/cx_insights_icg/Change_Log.htm#4 ==== THEIRS //eic/2019r4_systest/products/documentation/source/Technical_Reference_HTML/cx_insights_icg/Change_Log.htm#5 ==== YOURS //hyd-arunas-l_eic_2020r1_systest/products/documentation/source/Technical_Reference_HTML/cx_insights_icg/Change_Log.htm <<<<

Date	Change
28-June-2019	Initial release
21-November-2019	Updated architecture diagram
02-December-2019	Added Configure HTTPS For Nginx topic
04-December-2019	Updated Analytics Configuration description
06-April-2020	Made changes in CX Insights Server and added troubleshooting topic
20-March-2020	Added troubleshooting, Monitoring and Alerting, and also made few changes to install server topic

Troubleshooting

>>>> ORIGINAL //eic/2019r4_systest/products/documentation/source/Technical_Reference_HTML/cx_insights_icg/Troubleshooting.htm#none ==== THEIRS //eic/2019r4_systest/products/documentation/source/Technical_Reference_HTML/cx_insights_icg/Troubleshooting.htm#1

Error -1: If the below error is seen

```
[root@pcn-cent7-cn24 ~]# docker ps
CONTAINER ID        IMAGE                                                                COMMAND                  CREATED             STATUS              PORTS
        NAMES
efd5a4999acd        docker-dev-pcon-repo.ininlab.com/pureconnect/gcxi:latest.team.ase396           "/bin/bash -c $(GCXI…"   2 weeks ago         Up 2 weeks          0.0.0.0:8080->8080/tcp, 0.0.0.0:34952->34
952/tcp   root_gcxi_1
64209facd51a        docker-dev-pcon-repo.ininlab.com/pureconnect/gcxi-postgres:latest.team.ase396  "docker-entrypoint.s…"   2 weeks ago         Up 2 weeks          0.0.0.0:5432->5432/tcp
        root_gcxi-postgres_1
82667f452197        docker-dev-pcon-repo.ininlab.com/pureconnect/mstr-dataadapter-server:latest    "sh pcon-entrypoint.…"   2 weeks ago         Up 2 weeks          0.0.0.0:8078->8078/tcp
        root_mstr-dataadapter-server_1
df89768adeb8        docker-dev-pcon-repo.ininlab.com/pureconnect/mstr-dataadapter-agent:latest     "sh pcon-entrypoint.…"   2 weeks ago         Up 46 hours
        root_mstr-dataadapter-agent_1
943948a22b41        docker-dev-pcon-repo.ininlab.com/pureconnect/mstr-connector:latest             "sh pcon-entrypoint.…"   2 weeks ago         Up 2 weeks          0.0.0.0:8077->8077/tcp
        root_mstr-connector_1
[root@pcn-cent7-cn24 ~]#
```

Solution: Ensure all the containers are up and running after installation.

Error -2: If the below error is seen

Solution: Check mstrWeb Page opens up using url : http://<host ip>:8080/MicroStrategy/servlet/mstrWeb

Connection between container

Error-3: If errors are seen in the connections between the containers

Solution: Verify the connection between containers mstr-connection with gcxi container after 6-7 minutes.

To access container's log use the cmd: `docker logs container_id --follow`
Example: `docker logs 3bff --follow` where 3bff are the first characters of a container id
To list all running containers with their ids use cmd: `docker ps`

Ensure that connector logs shows up below logs

info [2020-02-06T11:16:03.185Z] - MicroStrategyConnector:Prometheus Prometheus started on port: 9090
info [2020-02-06T11:16:03.191Z] - MicroStrategyConnector:Prometheus Starting the collection of metrics, the metrics are available on /metrics
verb [2020-02-06T11:16:03.329Z] - MicroStrategyConnector:MstrHealthCheck Received: 200
verb [2020-02-06T11:16:03.329Z] - MicroStrategyConnector:MstrHealthCheck changing status to good
verb [2020-02-06T11:16:03.330Z] - MicroStrategyConnector:ConnectorServer In method: onMstrHealthChange
info [2020-02-06T11:16:03.330Z] - MicroStrategyConnector:ConnectorServer MSTR is up
info [2020-02-06T11:16:03.330Z] - MicroStrategyConnector:ConnectorServer Starting container server
info [2020-02-06T11:16:03.331Z] - MicroStrategyConnector:ConnectorServer Listening on port 8077

Ensure that mstr-dataadapter-server logs shows up below logs

info [2020-02-06T16:16:14.036Z] - MicroStrategyDataAdapterServer:DataAdapterServer MSTR is up
info [2020-02-06T16:16:14.036Z] - MicroStrategyDataAdapterServer:DataAdapterServer Starting container server
info [2020-02-06T16:16:14.036Z] - MicroStrategyDataAdapterServer:DataAdapterServer Listening on port 8078
info [2020-02-06T16:16:15.690Z] - iccontainerserver:authorize.js Authorized connection for service Agent

In case if it shows Error EHOSTUNREACH (No route to host), ensure the following ports are open

Commands to open ports are below

sudo firewall-cmd --zone=public --permanent --add-port=8077/tcp
sudo firewall-cmd --zone=public --permanent --add-port=8078/tcp
sudo firewall-cmd --zone=public --permanent --add-port=8080/tcp
sudo firewall-cmd --zone=public --permanent --add-port=5432/tcp

Even after making this changes if till the problem exists, then ensure that there is no IP conflict with Linux host machine and docker container IP. In case if there is conflict following below steps

- Change docker domain IP range. Add the following line { "bip": in file etc/docker/daemon.json }

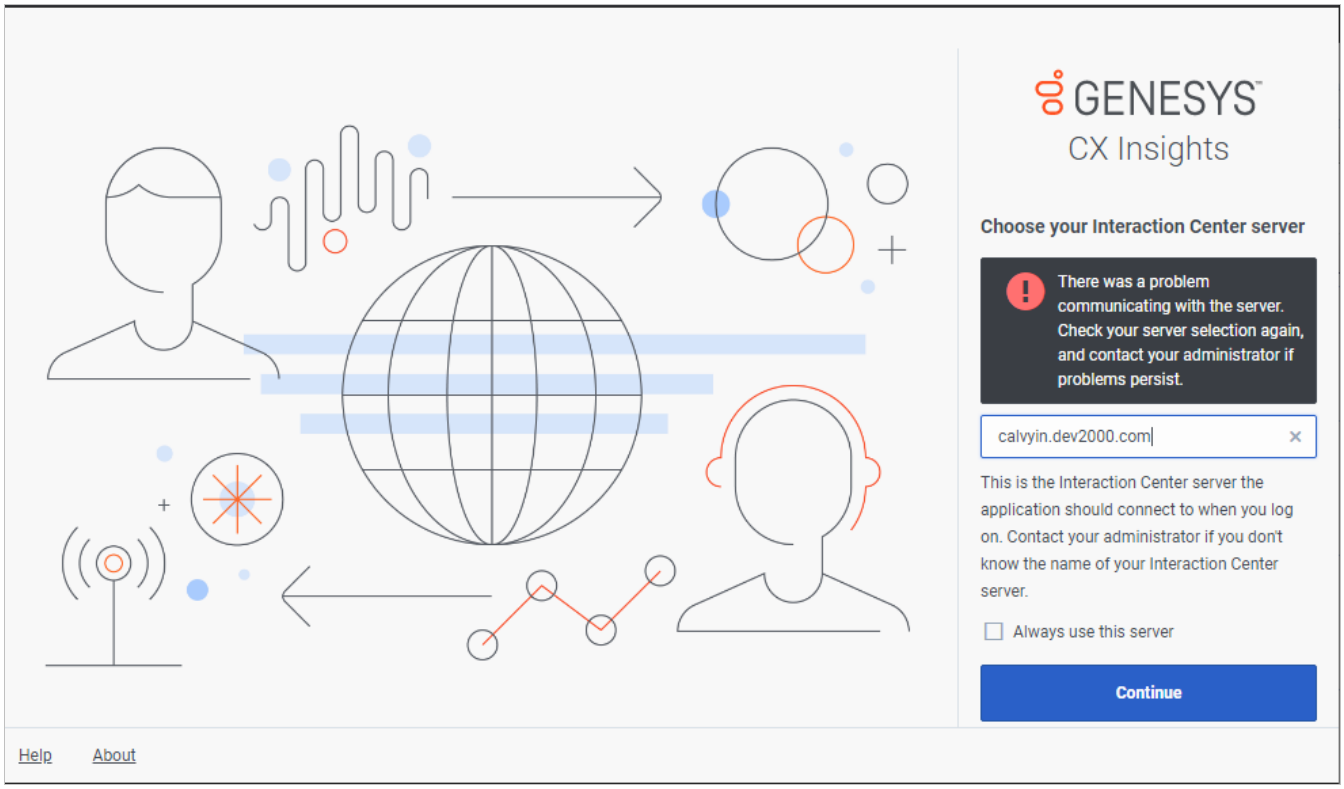
Error-4: If the below error is seen while running ansible-playbook.

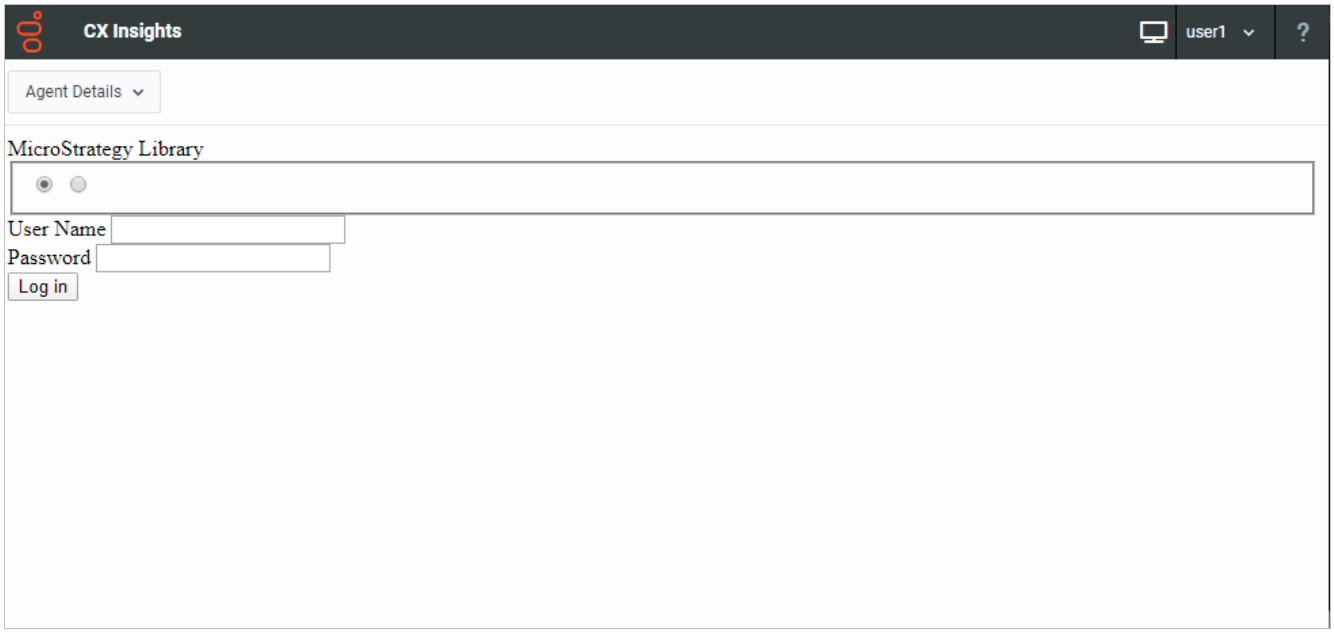
```
[admin@localhost cxinsights-playbook]$ ansible-playbook -i development site.yml -k
SSH password:
/usr/lib/python2.7/site-packages/requests/__init__.py:91: RequestsDependencyWarning: urllib3 (1.25.8) or chardet (2.2.1)
 doesn't match a supported version!
  RequestsDependencyWarning)

PLAY [all] *********************************************************************************************************************

TASK [Gathering Facts] *********************************************************************************************************
fatal: [172.26.20.62]: FAILED! => {"msg": "Using a SSH password instead of a key is not possible because Host Key checki
ng is enabled and sshpass does not support this.  Please add this host's fingerprint to your known_hosts file to manage
this host."}

PLAY RECAP *********************************************************************************************************************
172.26.20.62               : ok=0    changed=0    unreachable=0    failed=1    skipped=0    rescued=0    ignored=0
```

Solution: Run this command ANSIBLE_HOST_KEY_CHECKING=False to resolve the error.

Web Application

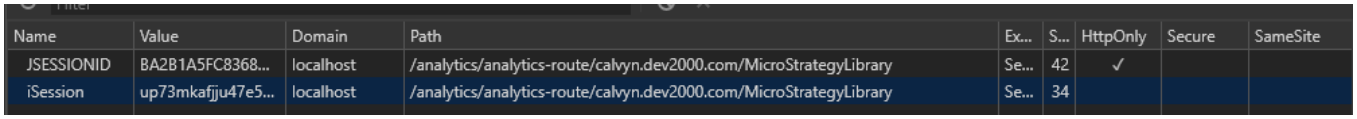
Error-5: While performing Installation on IIS Server, If the below error is seen





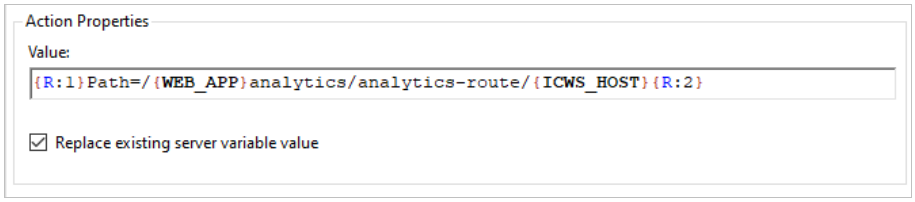
Solution: Ensure all the url rewrite rules are correct, refer to the file web.config.to resolve the error.

Error-6: Check whether the cookie path is same for JSESSION and ISession as shown below. If in case the cookie path is not same for JSession and ISession

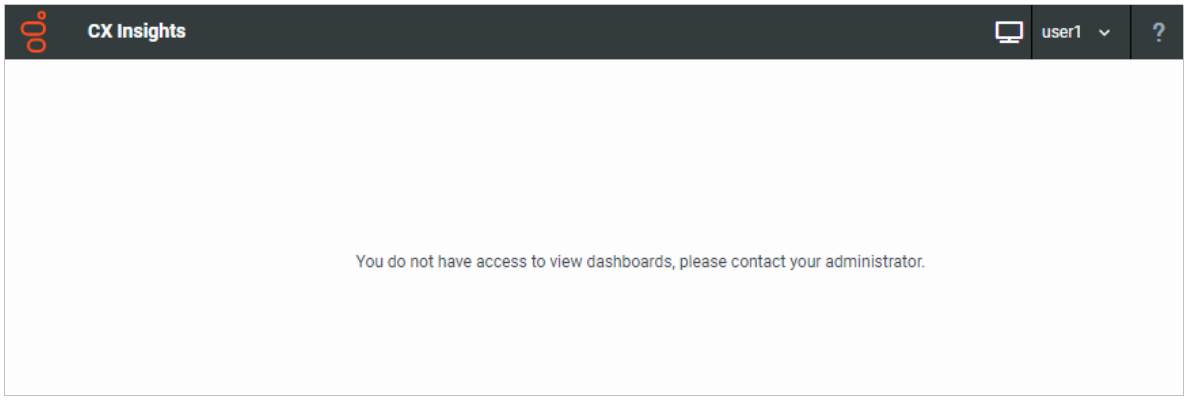


Name	Value	Domain	Path	Ex...	S...	HttpOnly	Secure	SameSite
JSESSIONID	BA2B1A5FC8368...	localhost	/analytics/analytics-route/calvyn.dev2000.com/MicroStrategyLibrary	Se...	42	✓		
iSession	up73mkafjju47e5...	localhost	/analytics/analytics-route/calvyn.dev2000.com/MicroStrategyLibrary	Se...	34			

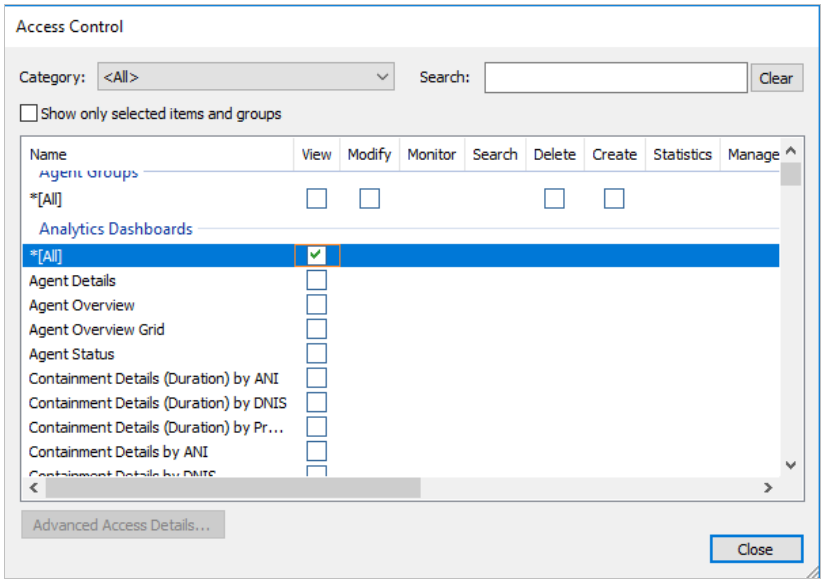
Solution: Change outbound rule inin-analytics-cookie as shown below



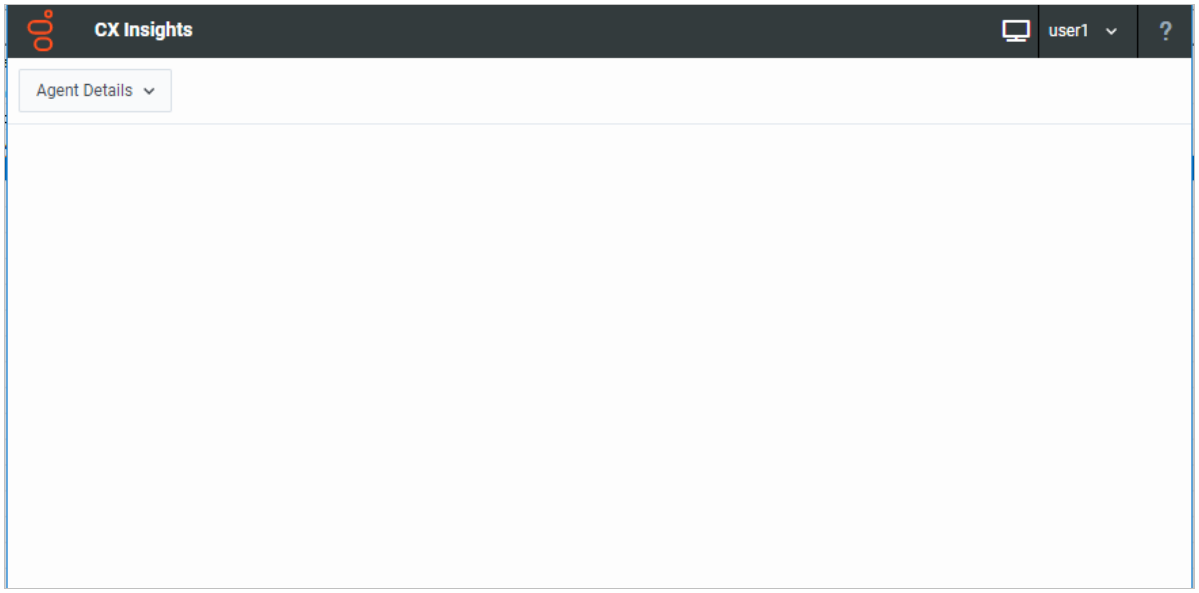
Error-7: If the user doesn't have access to dashboards below error message is displayed



Solution: To permit access to the dashboard in Interaction Administrator → User Properties → Security→AccessControl

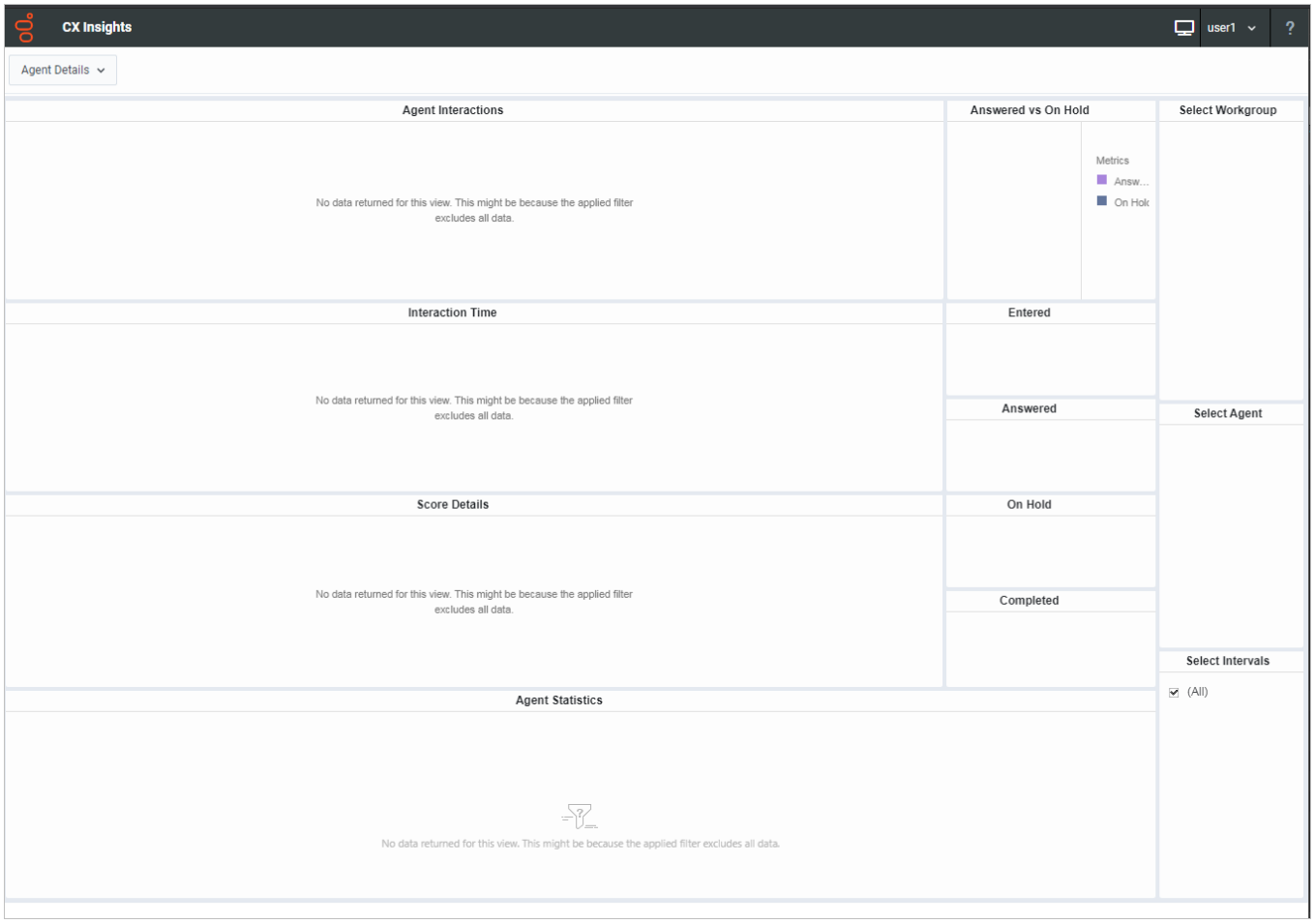


Error-8: If encounter blank screen without any content in the dashboard



Solution: Ensure Analytics Bridge is up and running and also connected to microservice mstr-connector to resolve the error.

Error-9: If data is not updated in dashboard



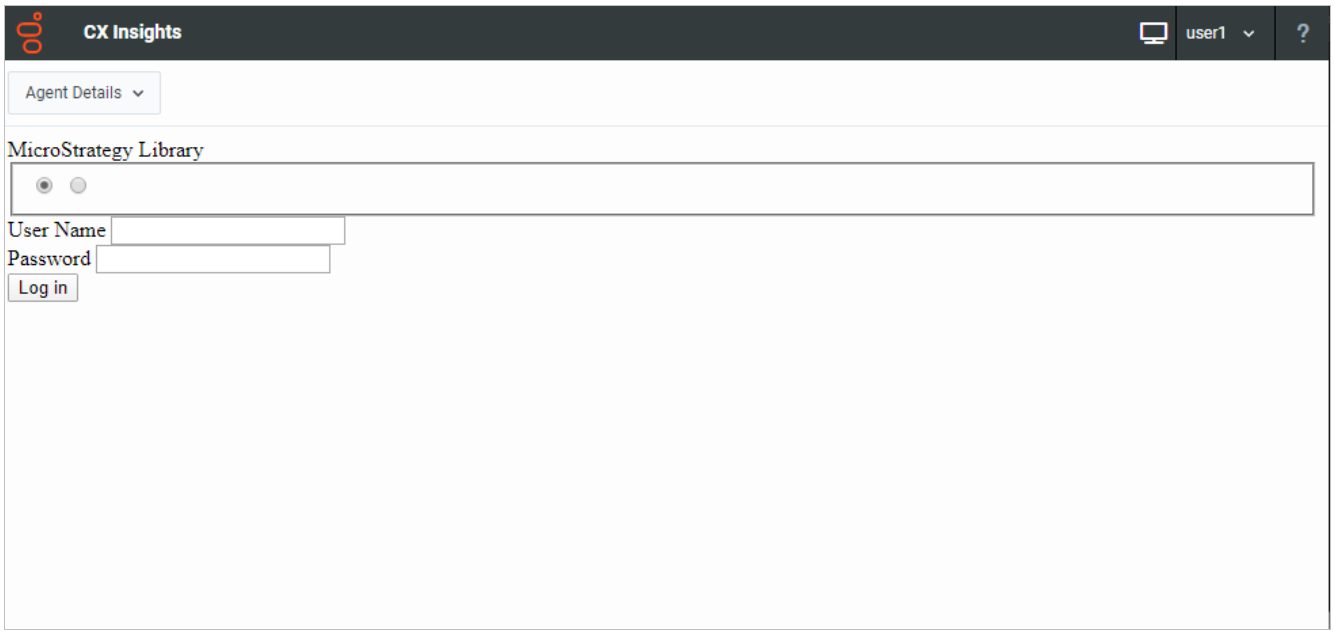
Solution: Ensure that the connection between analytics bridge and mstr-dataadapter-server, even if there any valid statistics shown in ICBM

Known Issues:

Error-10: Sometimes login screen shows "user is not licensed" error message even user had valid license.

Solution: Try to login again to resolve the error.

Error-11: If user login with one account and logged in again in another browser with the same account, then the previous session will be deleted and even if user is idle for long time, it shows below error



Solution: Try to login again to resolve the error

==== YOURS //hyd-arunas-l_eic_2020r1_systest/products/documentation/source/Technical_Reference_HTML/cx_insights_icg/Troubleshooting.htm

Error -1: If the below error is seen

```
TASK [helm-chart-installation : Install Helm Charts for PCC] ******************************************************
fatal: [172.26.20.55]: FAILED! => {"changed": false, "cmd": ["helm", "install", "/home/cxinsights/pcc-cxinsights-playboo
k/pcon-mstr", "--name", "pcc-helmcharts", "--namespace", "pcn-cxinsights-system", "--tiller-namespace", "pcn-tiller-syst
em", "-f", "/home/cxinsights/pcc-cxinsights-playbook/values.yml"], "delta": "0:00:00.713750", "end": "2020-02-21 07:04:1
9.918309", "failed_when_result": true, "msg": "non-zero return code", "rc": 1, "start": "2020-02-21 07:04:19.204559", "s
tderr": "Error: release pcc-helmcharts failed: Ingress.extensions \"pcc-helmcharts-mstrdataadapterserver\" is invalid: s
pec.rules[0].host: Invalid value: \"172.26.20.55\": must be a DNS name, not an IP address", "stderr_lines": ["Error: rel
ease pcc-helmcharts failed: Ingress.extensions \"pcc-helmcharts-mstrdataadapterserver\" is invalid: spec.rules[0].host:
Invalid value: \"172.26.20.55\": must be a DNS name, not an IP address"], "stdout": "", "stdout_lines": []}
```

Solution: Check for the host, as host cannot be an IP address. Use DNS name instead to resolve the error.

Error - 2: If the below error is seen

```
RUNNING HANDLER [k3s : restart k3s] ********************************************************************************
fatal: [172.26.20.55]: FAILED! => {"changed": false, "msg": "Unable to restart service k3s: Failed to restart k3s.servic
e: Connection timed out\nSee system logs and 'systemctl status k3s.service' for details.\n"}

NO MORE HOSTS LEFT *************************************************************************************************
```

Solution: To resolve this error, re-run the sudo ansible-playbook --vault-id cxinsights@prompt -i inventory.yml ./site.yml -K cmd.

Error - 3: If the below error is seen

```
TASK [helm-chart-installation : Install Helm Charts for PCC] ******************************************************
fatal: [172.26.20.55]: FAILED! => {"changed": false, "cmd": ["helm", "install", "pcon-mstr", "--name", "pcc-helmcharts",
 "--namespace", "pcn-cxinsights-system", "--tiller-namespace", "pcn-tiller-system", "-f", "~/values.yml"], "delta": "0:0
0:00.166113", "end": "2020-02-21 06:47:47.533577", "failed_when_result": true, "msg": "non-zero return code", "rc": 1, "
start": "2020-02-21 06:47:47.367464", "stderr": "Error: failed to download \"pcon-mstr\" (hint: running `helm repo updat
e` may help)", "stderr_lines": ["Error: failed to download \"pcon-mstr\" (hint: running `helm repo update` may help)"],
"stdout": "", "stdout_lines": []}
```

Solution: To resolve this error, check for the pcon-folder path. It should be in pcc-cxinsights-playbook/group_vars/all.yml upstream_chart value path.
<<<<

Change Log

The following table lists the changes to this document since its initial release.
>>>> ORIGINAL //eic/2019r4_systest/products/documentation/source/Technical_Reference_HTML/cx_insights_icg/Change_Log.htm#4 ==== THEIRS //eic/2019r4_systest/products/documentation/source/Technical_Reference_HTML/cx_insights_icg/Change_Log.htm#5 ==== YOURS //hyd-arunas-l_eic_2020r1_systest/products/documentation/source/Technical_Reference_HTML/cx_insights_icg/Change_Log.htm <<<<

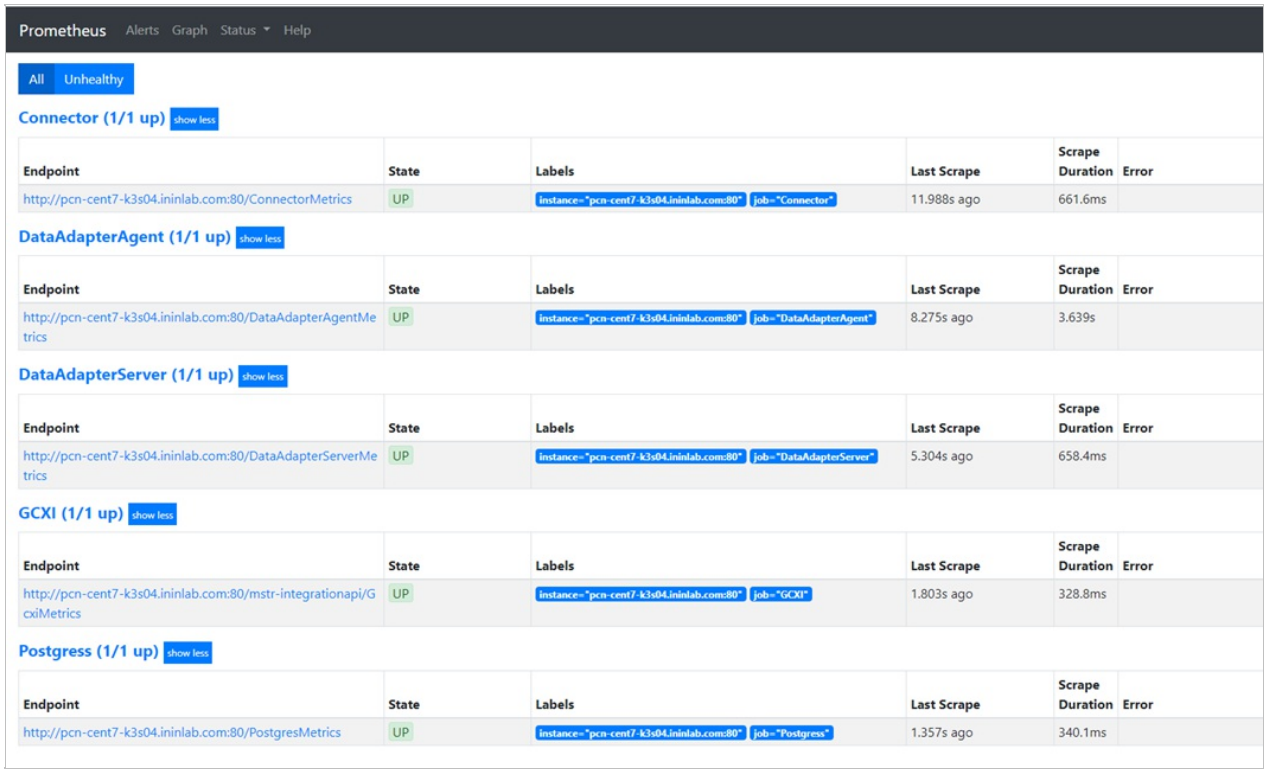
Date	Change		
28-June-2019	Initial release		
21-November-2019	Updated architecture diagram		
02-December-2019	Added Configure HTTPS For Nginx topic		
04-December-2019	Updated Analytics Configuration description		
06-April-2020	Made changes in CX Insights Server and added troubleshooting topic	20-March-2020	Added troubleshooting, Monitoring and Alerting, and also made few changes to install server topic

CX Insights monitoring and alerting

1. Install Prometheus from https://prometheus.io/download/ and extract the files from the folder.
2. Copy alerts.yml inside Prometheus folder and update prometheusus.yml **rule_files** property with **alerts.yml.**
3. Change Prometheus.yml with the below mentioned content and replace <SERVER> with Linux host (Where all the containers are up and running). In rules_files section alerts.yml file reference is provided which contains all the alert scenarios. Scrape_interval is the interval in which data is pulled from all services and evaluation_interval is the internal all rules are evaluated.

```
# my global config
global:
scrape_interval: 15s # Set the scrape interval to every 15 seconds. Default is every 1
minute.
evaluation_interval: 15s # Evaluate rules every 15 seconds. The default is every 1 minute.
# scrape_timeout is set to the global default (10s).
# Alertmanager configuration
alerting:
alertmanagers:
- static_configs:
- targets:
# - alertmanager:9093
# Load rules once and periodically evaluate them according to the global
'evaluation_interval'.
rule_files:
- alerts.yml
# - "first_rules.yml"
# - "second_rules.yml"
# A scrape configuration containing exactly one endpoint to scrape:
# Here it's Prometheus itself.
scrape_configs:
# The job name is added as a label `job=<job_name>` to any timeseries scraped from this
config.
- job_name: 'DataAdapterServer'
metrics_path: /DataAdapterServerMetrics
static_configs:
- targets: ['<SERVER>']
- job_name: 'Connector'
metrics_path: /ConnectorMetrics
static_configs:
- targets: ['<SERVER>']
- job_name: 'Postgress'
metrics_path: /PostgresMetrics
static_configs:
- targets: ['<SERVER>']
- job_name: 'DataAdapterAgent'
metrics_path: /DataAdapterAgentMetrics
static_configs:
- targets: ['<SERVER>']
- job_name: 'GCXI'
static_configs:
- targets: ['<SERVER>']
relabel_configs:
- source_labels:
- __metrics_path__
action: replace
target_label: __metrics_path__
replacement: /mstr-integrationapi/GcxiMetrics
}
```

4. After running Prometheus executable, ensure http://localhost:9090/rules is accessible and all rules are defined properly. We have defined warning and critical alerts, warning is of less priority, if there are any critical alerts fired, rise ticket with proper logs.
5. The http://localhost:9090/targets shows container state.



6. Alerts information can be seen in http://localhost:9090/alerts
7. To receive an e-mail notifications/pagerduty configure alertmanager. More details about alert manger is found in https://prometheus.io/docs/alerting/alertmanager/ and download is available in the https://prometheus.io/download/.
8. After downloading the configure prometheus.yml with alert manager in the # Alertmanager configuration

```
alerting:
alertmanagers:
- static_configs:
- targets:
 - alertmanager:9093
```

9. To receive email notifications from alert manager, configure alertmanager.yml as shown below with details.

```
route:
  group_by: ['alertname']
  group_wait: 30s
  group_interval: 10s
  repeat_interval: 20s
  receiver: 'email-me'
receivers:
 - name: 'email-me'
email_configs:
 - to: xxxxx@gmail.com
   from: xxxxx@gmail.com
   smarthost: smtp.gmail.com:587
   auth_username: "xxxxxx@gmail.com"
   auth_password: "xxxxxx"
```

Troubleshooting

>>>> ORIGINAL //eic/2019r4_systest/products/documentation/source/Technical_Reference_HTML/cx_insights_icg/Troubleshooting.htm#none ==== THEIRS //eic/2019r4_systest/products/documentation/source/Technical_Reference_HTML/cx_insights_icg/Troubleshooting.htm#1

Error -1: If the below error is seen

Solution: Ensure all the containers are up and running after installation.

Error -2: If the below error is seen

Solution: Check mstrWeb Page opens up using url : http://<host ip>:8080/MicroStrategy/servlet/mstrWeb

Connection between container

Error-3: If errors are seen in the connections between the containers

Solution: Verify the connection between containers mstr-connection with gcxi container after 6-7 minutes.

To access container's log use the cmd: `docker logs container_id --follow`
Example: `docker logs 3bff --follow` where 3bff are the first characters of a container id
To list all running containers with their ids use cmd: `docker ps`

Ensure that connector logs shows up below logs

info [2020-02-06T11:16:03.185Z] - MicroStrategyConnector:Prometheus Prometheus started on port: 9090
info [2020-02-06T11:16:03.191Z] - MicroStrategyConnector:Prometheus Starting the collection of metrics, the metrics are available on /metrics
verb [2020-02-06T11:16:03.329Z] - MicroStrategyConnector:MstrHealthCheck Received: 200
verb [2020-02-06T11:16:03.329Z] - MicroStrategyConnector:MstrHealthCheck changing status to good
verb [2020-02-06T11:16:03.330Z] - MicroStrategyConnector:ConnectorServer In method: onMstrHealthChange
info [2020-02-06T11:16:03.330Z] - MicroStrategyConnector:ConnectorServer MSTR is up
info [2020-02-06T11:16:03.330Z] - MicroStrategyConnector:ConnectorServer Starting container server
info [2020-02-06T11:16:03.331Z] - MicroStrategyConnector:ConnectorServer Listening on port 8077

Ensure that mstr-dataadapter-server logs shows up below logs

info [2020-02-06T16:16:14.036Z] - MicroStrategyDataAdapterServer:DataAdapterServer MSTR is up
info [2020-02-06T16:16:14.036Z] - MicroStrategyDataAdapterServer:DataAdapterServer Starting container server
info [2020-02-06T16:16:14.036Z] - MicroStrategyDataAdapterServer:DataAdapterServer Listening on port 8078
info [2020-02-06T16:16:15.690Z] - iccontainerserver:authorize.js Authorized connection for service Agent

In case if it shows Error EHOSTUNREACH (No route to host), ensure the following ports are open

Commands to open ports are below

sudo firewall-cmd --zone=public --permanent --add-port=8077/tcp
sudo firewall-cmd --zone=public --permanent --add-port=8078/tcp
sudo firewall-cmd --zone=public --permanent --add-port=8080/tcp
sudo firewall-cmd --zone=public --permanent --add-port=5432/tcp

Even after making this changes if till the problem exists, then ensure that there is no IP conflict with Linux host machine and docker container IP. In case if there is conflict following below steps

- Change docker domain IP range. Add the following line { "bip": in file etc/docker/daemon.json }

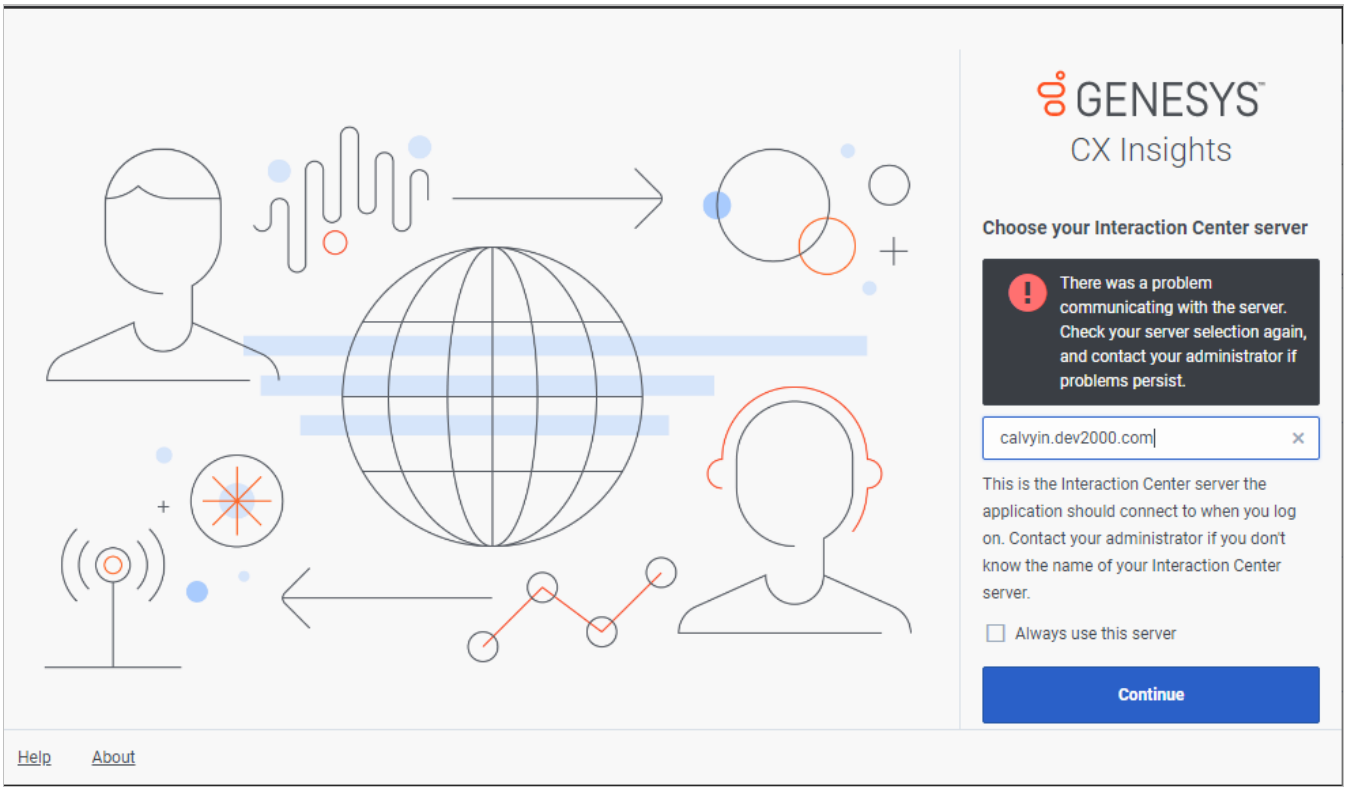
Error-4: If the below error is seen while running ansible-playbook.

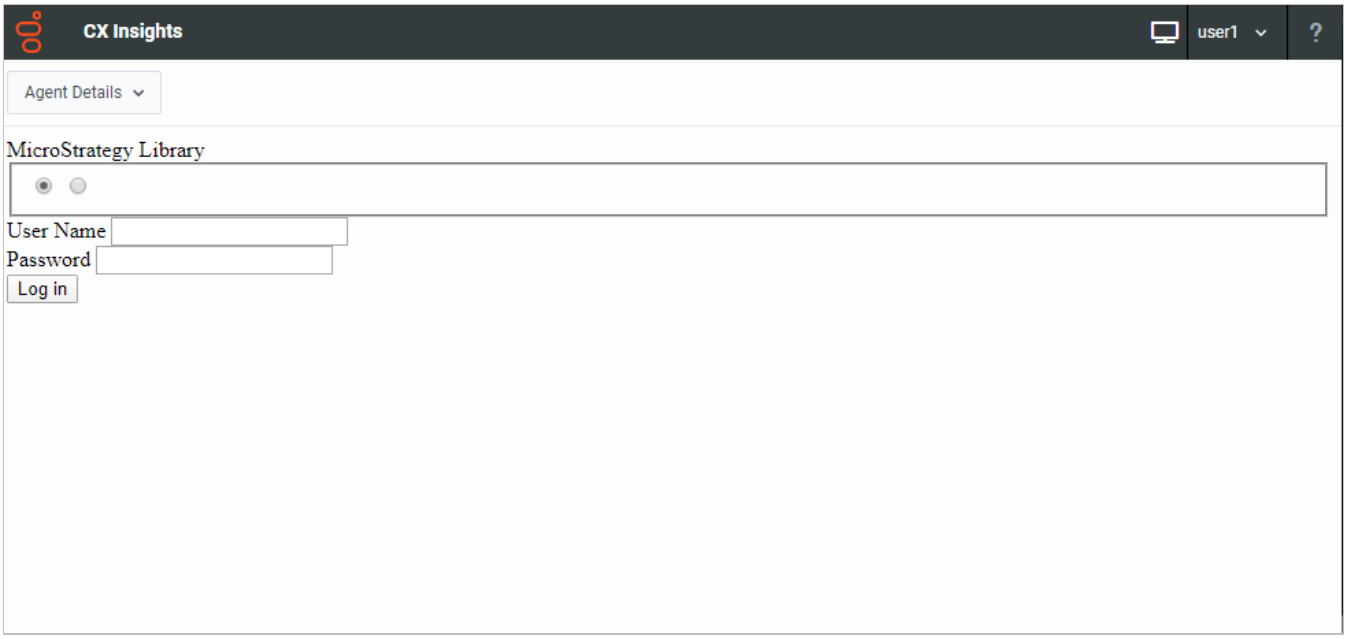
```
[admin@localhost cxinsights-playbook]$ ansible-playbook -i development site.yml -k
SSH password:
/usr/lib/python2.7/site-packages/requests/__init__.py:91: RequestsDependencyWarning: urllib3 (1.25.8) or chardet (2.2.1)
 doesn't match a supported version!
  RequestsDependencyWarning)

PLAY [all] *********************************************************************************************************

TASK [Gathering Facts] *********************************************************************************************
fatal: [172.26.20.62]: FAILED! => {"msg": "Using a SSH password instead of a key is not possible because Host Key checki
ng is enabled and sshpass does not support this.  Please add this host's fingerprint to your known_hosts file to manage
this host."}

PLAY RECAP *********************************************************************************************************
172.26.20.62               : ok=0    changed=0    unreachable=0    failed=1    skipped=0    rescued=0    ignored=0
```

Solution: Run this command ANSIBLE_HOST_KEY_CHECKING=False to resolve the error.

Web Application

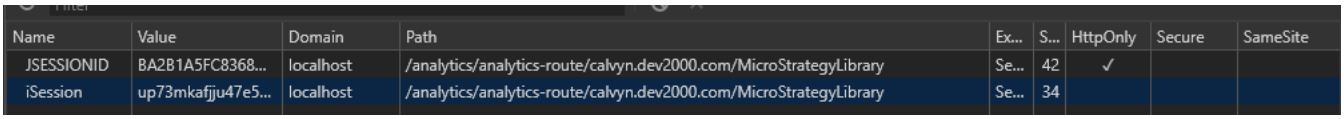
Error-5: While performing Installation on IIS Server, If the below error is seen





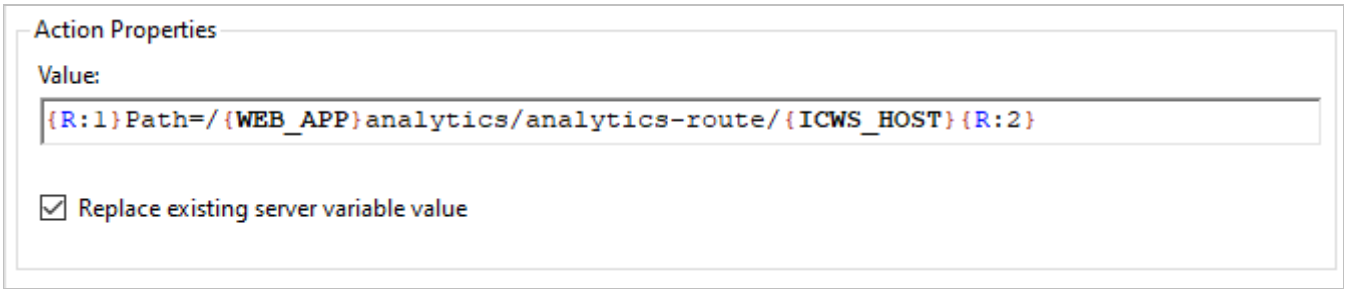
Solution: Ensure all the url rewrite rules are correct, refer to the file web.config.to resolve the error.

Error-6: Check whether the cookie path is same for JSESSION and ISession as shown below. If in case the cookie path is not same for JSession and ISession

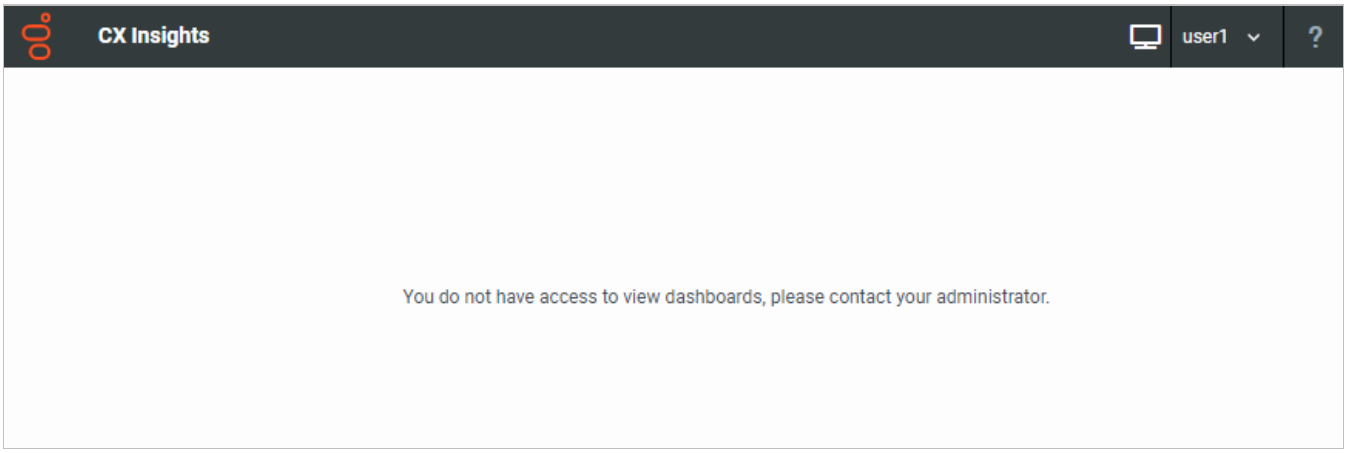


Name	Value	Domain	Path	Ex...	S...	HttpOnly	Secure	SameSite
JSESSIONID	BA2B1A5FC8368...	localhost	/analytics/analytics-route/calvyn.dev2000.com/MicroStrategyLibrary	Se...	42	✓		
iSession	up73mkafjju47e5...	localhost	/analytics/analytics-route/calvyn.dev2000.com/MicroStrategyLibrary	Se...	34			

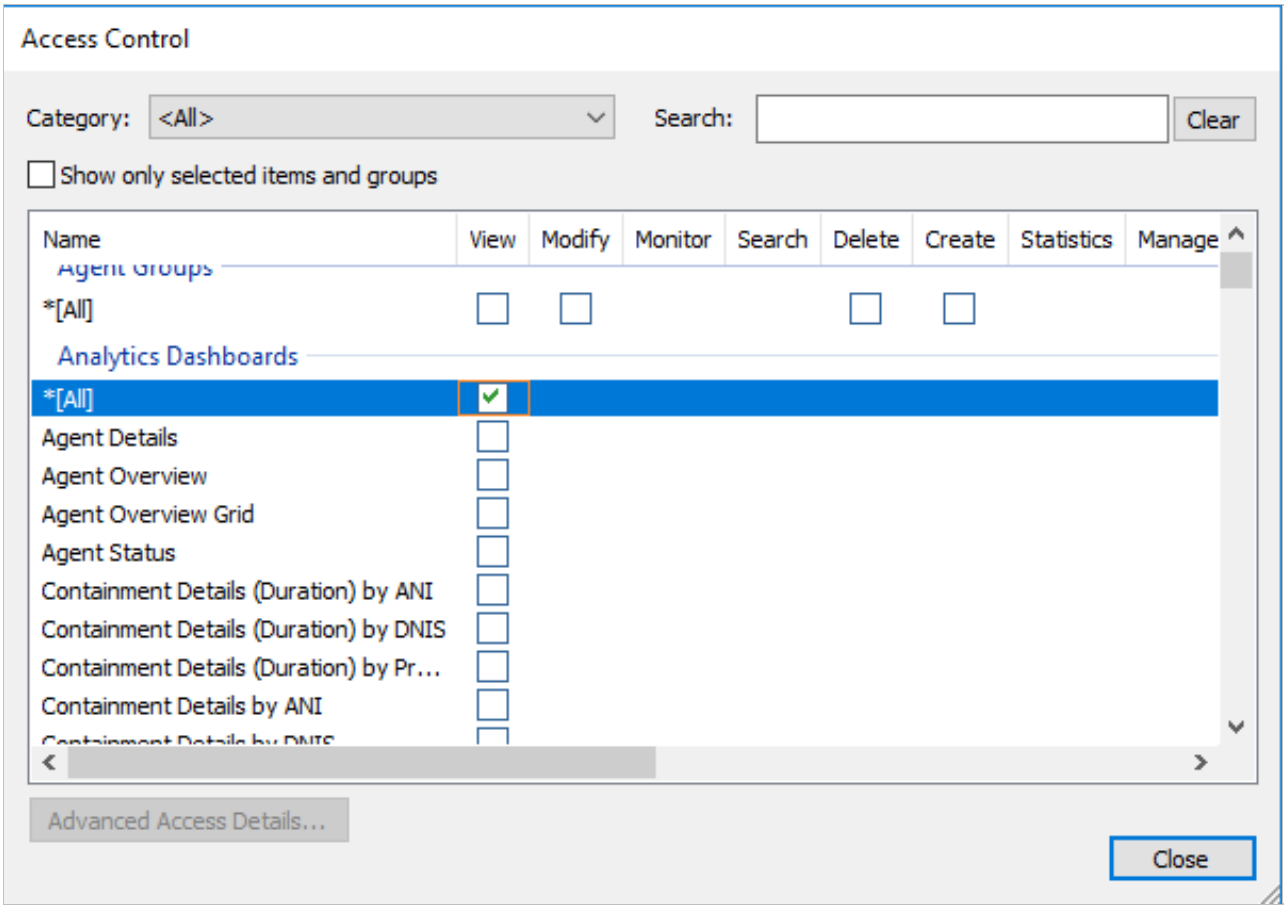
Solution: Change outbound rule inin-analytics-cookie as shown below



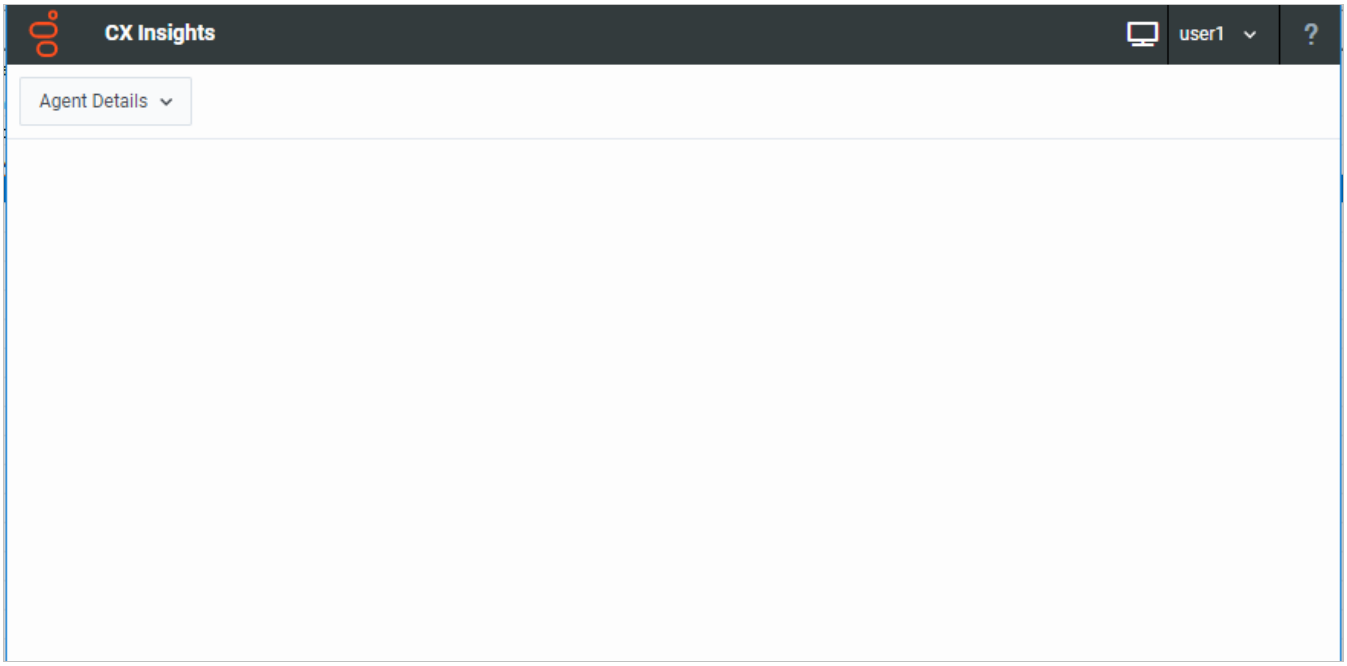
Error-7: If the user doesn't have access to dashboards below error message is displayed



Solution: To permit access to the dashboard in Interaction Administrator → User Properties → Security→AccessControl

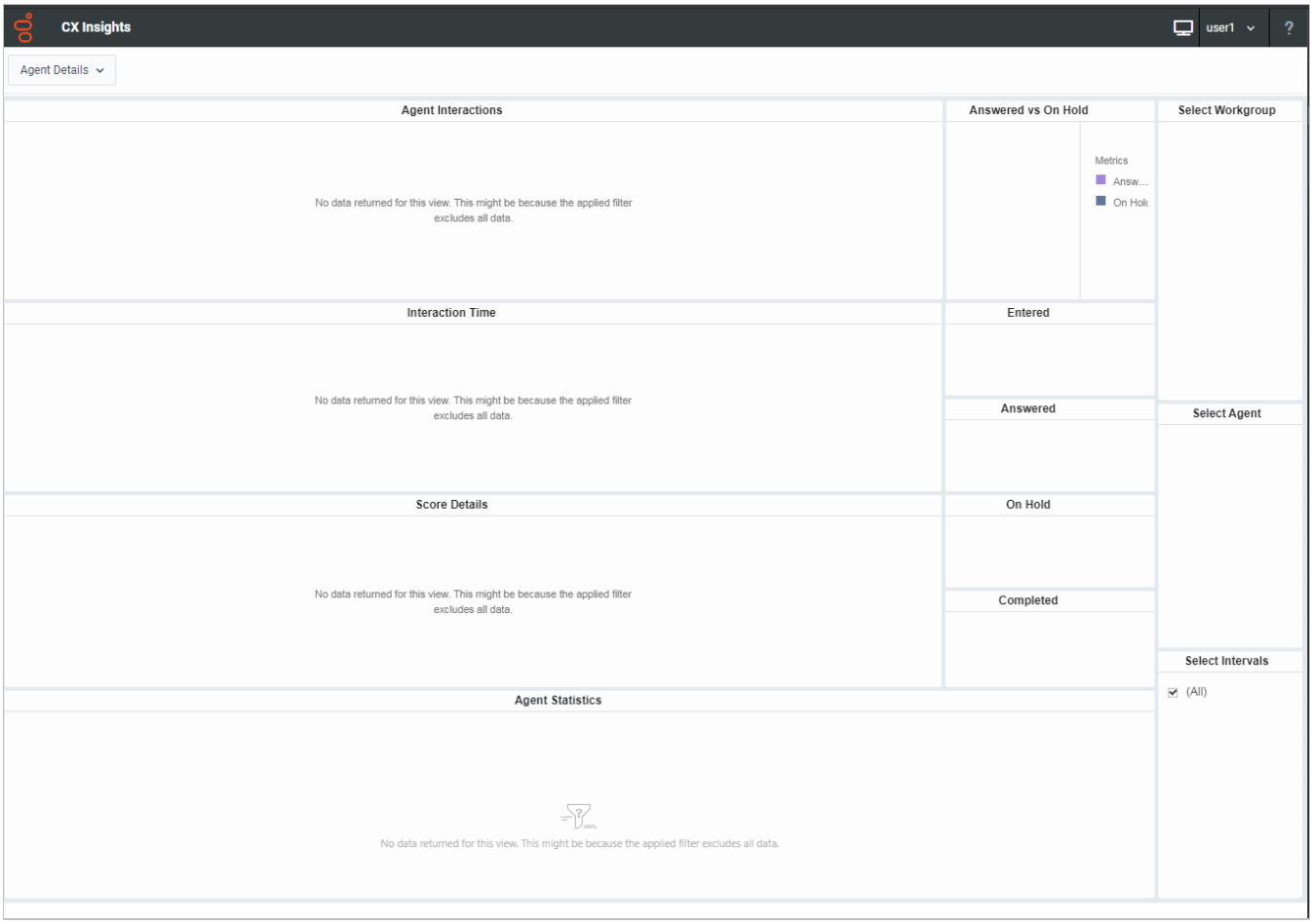


Error-8: If encounter blank screen without any content in the dashboard



Solution: Ensure Analytics Bridge is up and running and also connected to microservice mstr-connector to resolve the error.

Error-9: If data is not updated in dashboard



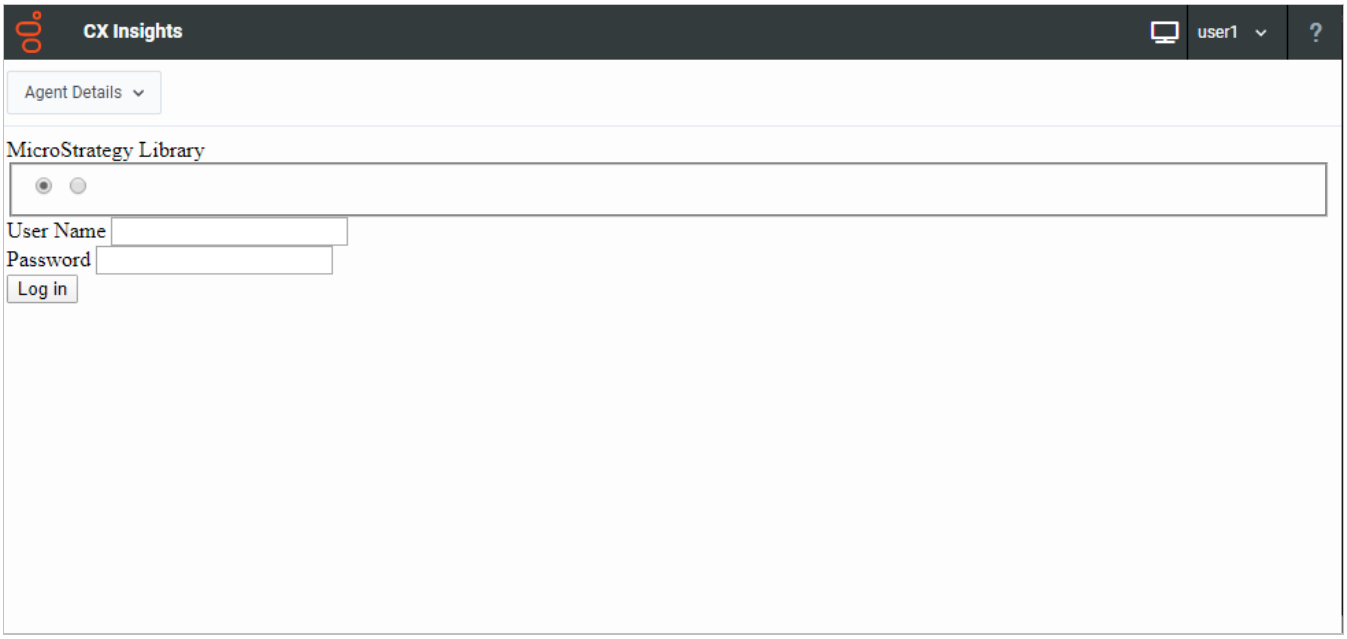
Solution: Ensure that the connection between analytics bridge and mstr-dataadapter-server, even if there any valid statistics shown in ICBM

Known Issues:

Error-10: Sometimes login screen shows "user is not licensed" error message even user had valid license.

Solution: Try to login again to resolve the error.

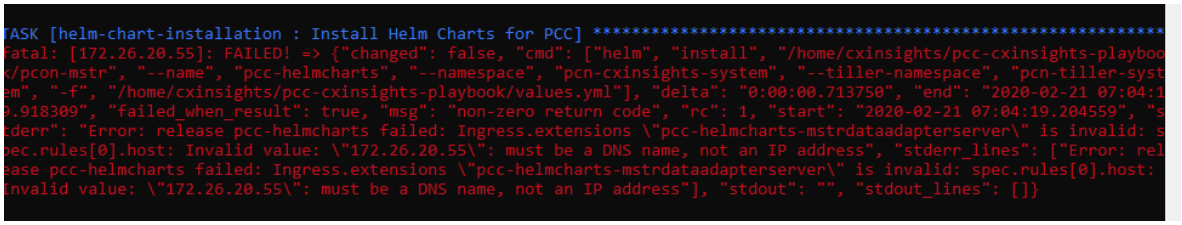
Error-11: If user login with one account and logged in again in another browser with the same account, then the previous session will be deleted and even if user is idle for long time, it shows below error



Solution: Try to login again to resolve the error

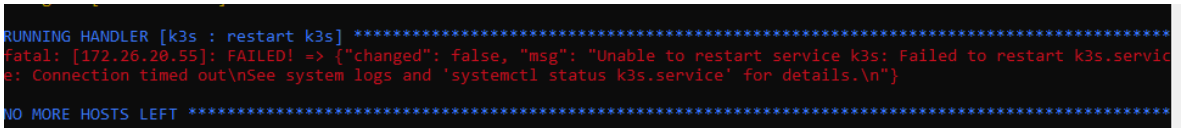
==== YOURS //hyd-arunas-l_eic_2020r1_systest/products/documentation/source/Technical_Reference_HTML/cx_insights_icg/Troubleshooting.htm

Error -1: If the below error is seen



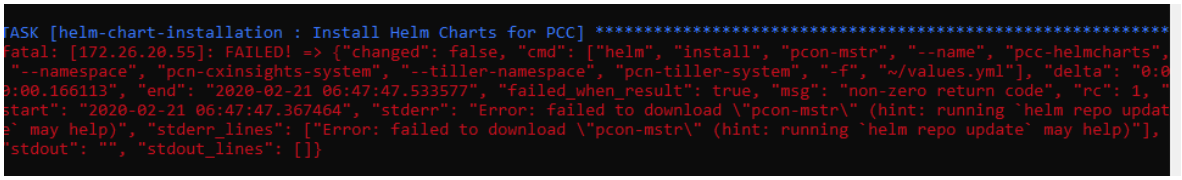
Solution: Check for the host, as host cannot be an IP address. Use DNS name instead to resolve the error.

Error - 2: If the below error is seen



Solution: To resolve this error, re-run the sudo ansible-playbook --vault-id cxinsights@prompt -i inventory.yml ./site.yml -K cmd.

Error - 3: If the below error is seen



Solution: To resolve this error, check for the pcon-folder path. It should be in pcc-cxinsights-playbook/group_vars/all.yml upstream_chart value path.

<<<<

Change Log

The following table lists the changes to this document since its initial release.

>>>> ORIGINAL //eic/2019r4_systest/products/documentation/source/Technical_Reference_HTML/cx_insights_icg/Change_Log.htm#4
==== THEIRS //eic/2019r4_systest/products/documentation/source/Technical_Reference_HTML/cx_insights_icg/Change_Log.htm#5
==== YOURS //hyd-arunas-l_eic_2020r1_systest/products/documentation/source/Technical_Reference_HTML/cx_insights_icg/Change_Log.htm <<<<

Date	Change
28-June-2019	Initial release
21-November-2019	Updated architecture diagram
02-December-2019	Added Configure HTTPS For Nginx topic
04-December-2019	Updated Analytics Configuration description
06-April-2020	Made changes in CX Insights Server and added troubleshooting topic
20-March-2020	Added troubleshooting, Monitoring and Alerting, and also made few changes to install server topic