PureConnect®

2020 R1

Generated:

15-September-2021

Content last updated:

01-September-2021

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

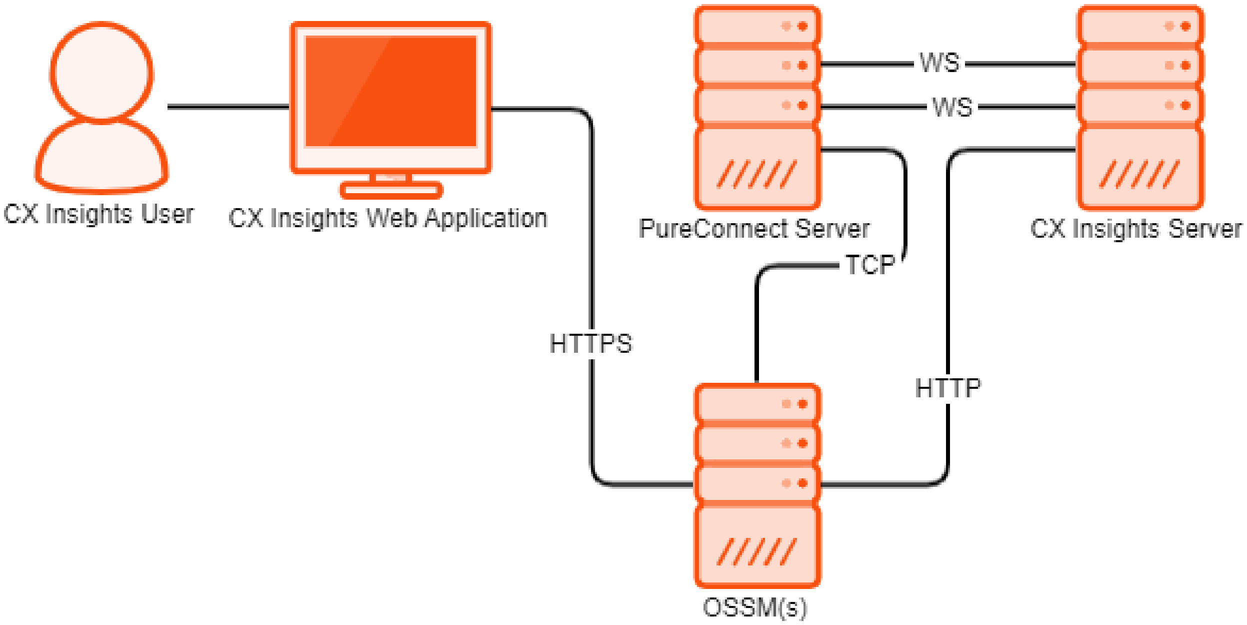
CX Insights overview

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

You need Internet Connectivity while installing CX Insights, to download few packages and modules. After Installation is complete, Internet connectivity is not required.

As part of installation, CX Insights need to download required packages and modules for Ansible and Docker.

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

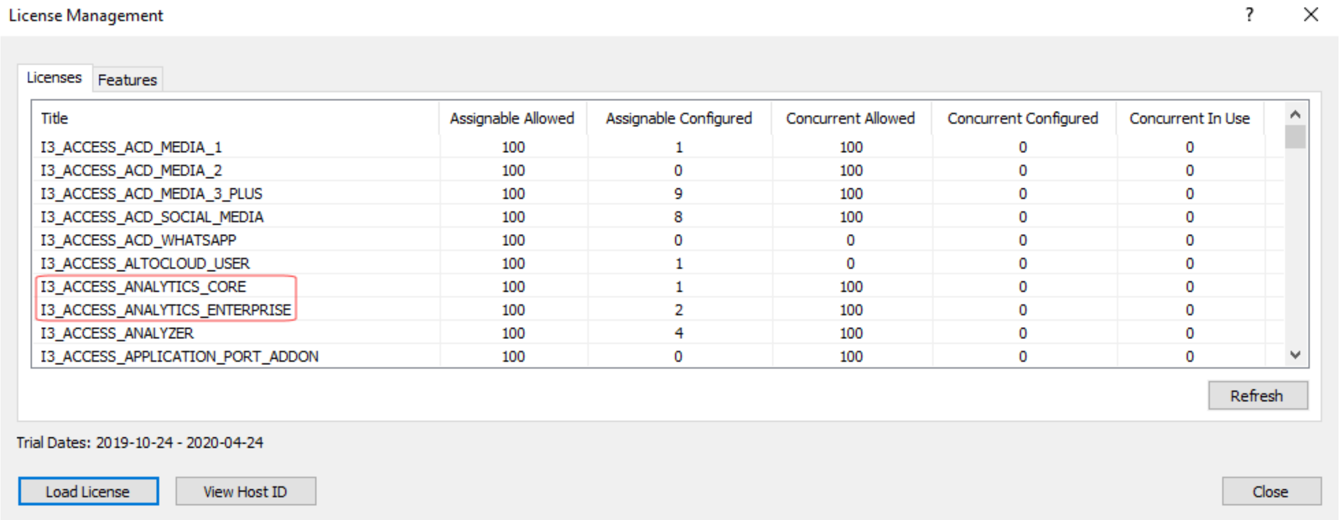
CX Insights licensing

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



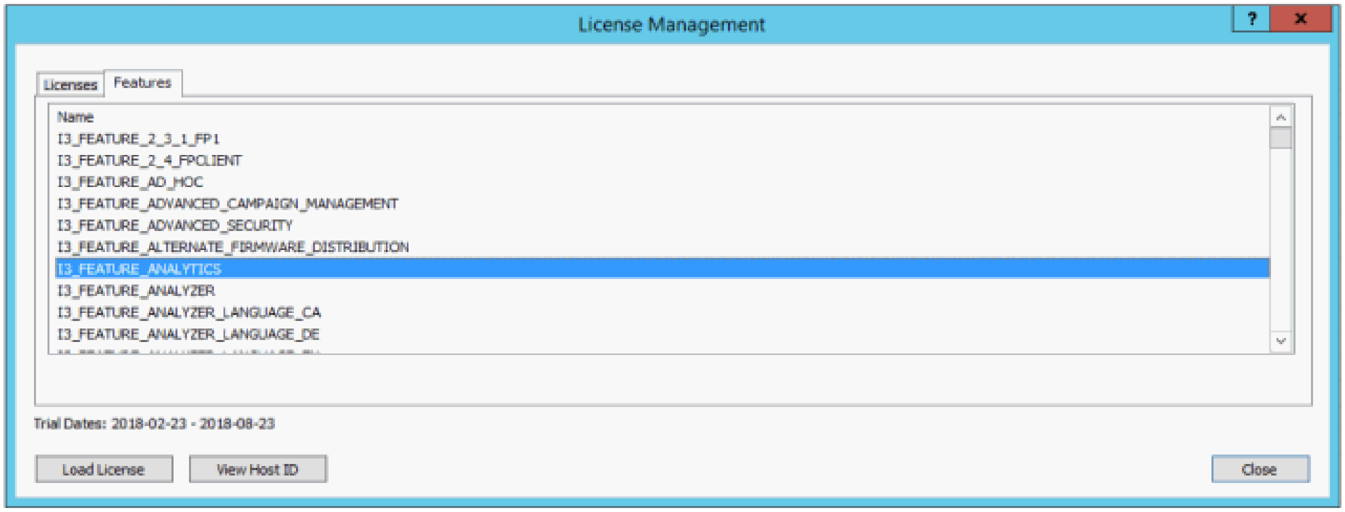
Title	Assignable Allowed	Assignable Configured	Concurrent Allowed	Concurrent Configured	Concurrent In Use
I3_ACCESS_ACD_MEDIA_1	100	1	100	0	0
I3_ACCESS_ACD_MEDIA_2	100	0	100	0	0
I3_ACCESS_ACD_MEDIA_3_PLUS	100	9	100	0	0
I3_ACCESS_ACD_SOCIAL_MEDIA	100	8	100	0	0
I3_ACCESS_ACD_WHATSAPP	100	0	0	0	0
I3_ACCESS_ALTOCLOUD_USER	100	1	0	0	0
I3_ACCESS_ANALYTICS_CORE	100	1	100	0	0
I3_ACCESS_ANALYTICS_ENTERPRISE	100	2	100	0	0
I3_ACCESS_ANALYZER	100	4	100	0	0
I3_ACCESS_APPLICATION_PORT_ADDON	100	0	100	0	0

Trial Dates: 2019-10-24 - 2020-04-24

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.



Name
I3_FEATURE_2_3_1_FP1
I3_FEATURE_2_4_FPCLIENT
I3_FEATURE_AD_HOC
I3_FEATURE_ADVANCED_CAMPAIGN_MANAGEMENT
I3_FEATURE_ADVANCED_SECURITY
I3_FEATURE_ALTERNATE_FIRMWARE_DISTRIBUTION
I3_FEATURE_ANALYTICS
I3_FEATURE_ANALYZER
I3_FEATURE_ANALYZER_LANGUAGE_CA
I3_FEATURE_ANALYZER_LANGUAGE_DE

Trial Dates: 2018-02-23 - 2018-08-23

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 back-end 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 ansible_install, cxinsights-playbook.tgz file, and cx-insights.tgz
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. Before executing sh script, you may need to change file properties. Use the following command to add execute permission: `chmod +x ansible_install.sh`
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 cxinsights=playbook.tgz file in the cxinsights user home directory.
- Copy cxinsights.tgz file inside cxinsights-playbook folder
- Create an inventory file in the cxinsights-playbook directory. It should look like the following with the appropriate values substituted. For example: Assume ansible is running on the CX Insights host. You need to change `<host fqdn>` to the cxinsights server name and the ansible_connection to '`ssh`' if using a remote machine to manage the server.
- For setting pcon_server_timezone ansible parameter in inventory file, please refer the link TZ Column here based on time zone of CX Insights host

[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

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

11. Run the Ansible Playbook to start the services on the CX Insights server. The first time will be slow as dependencies are installed, and container images downloaded.
    - `cd cxinsights-playbook`
    - `sudo ansible-playbook -i production ./site.yml -b`

    Ansible will run the playbook and test the server until its web services are responsive. At this point, the server should be ready to integrate with PureConnect.

    > **Note:** Wait for 6 minutes so that all the containers are ready to use.

Installation clean up

- To stop the existing container and to clean up, use the below command.

```
docker-compose down

docker volume rm `docker volume ls -q -f dangling=true`
```

- To delete existing docker images, `use docker rmi  `docker images -aq` command`

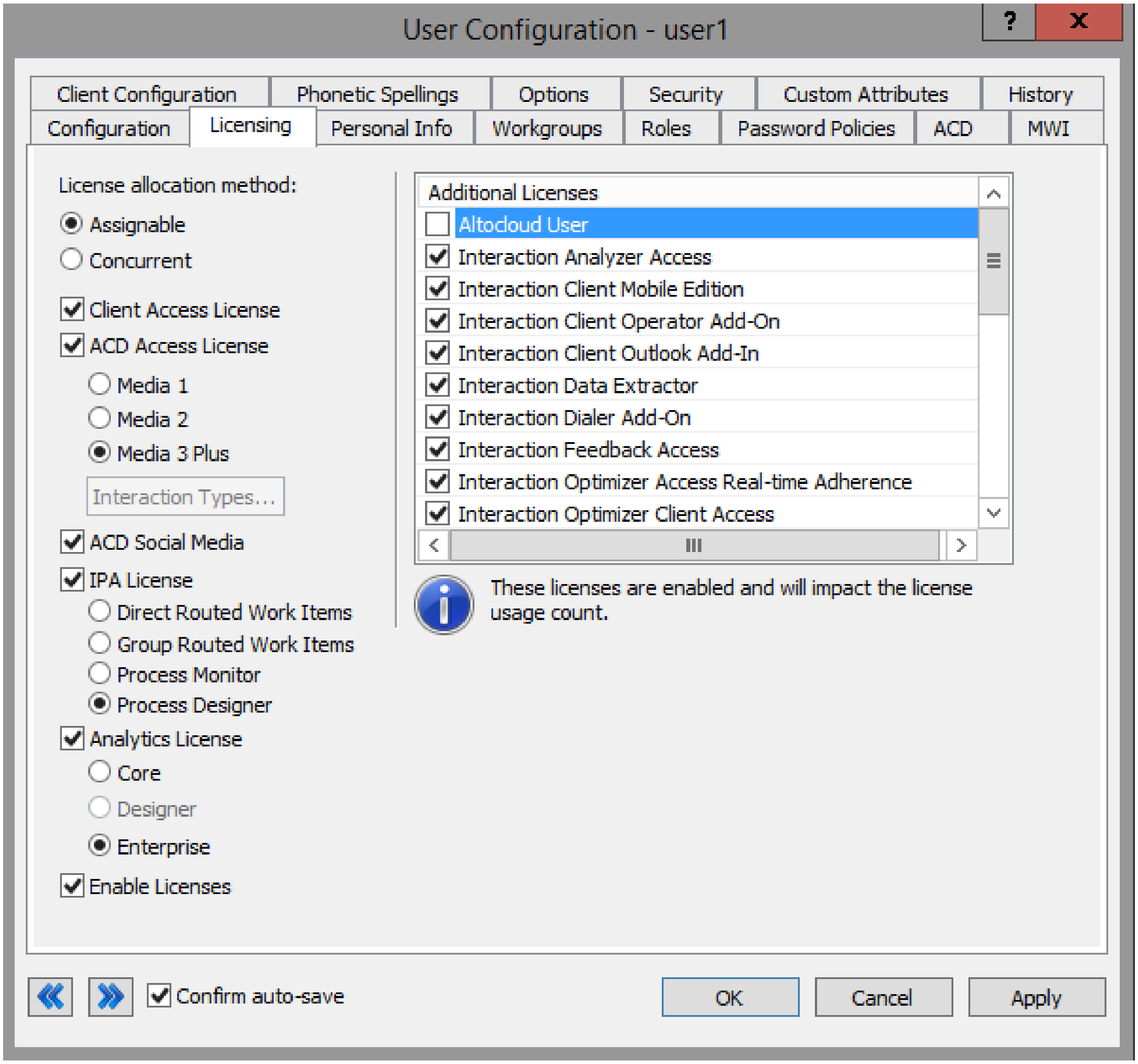
CX Insights server configuration

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.



License allocation method:

● Assignable
○ Concurrent

☑ Client Access License
☑ ACD Access License
○ Media 1
○ Media 2
● Media 3 Plus

☑ ACD Social Media
☑ IPA License
○ Direct Routed Work Items
○ Group Routed Work Items
○ Process Monitor
● Process Designer
☑ Analytics License
○ Core
○ Designer
● Enterprise
☑ Enable Licenses

Additional Licenses

□ AIcloud User
☑ Interaction Analyzer Access
☑ Interaction Client Mobile Edition
☑ Interaction Client Operator Add-On
☑ Interaction Client Outlook Add-In
☑ Interaction Data Extractor
☑ Interaction Dialer Add-On
☑ Interaction Feedback Access
☑ Interaction Optimizer Access Real-time Adherence
☑ Interaction Optimizer Client Access

These licenses are enabled and will impact the license usage count.

☑ Confirm auto-save

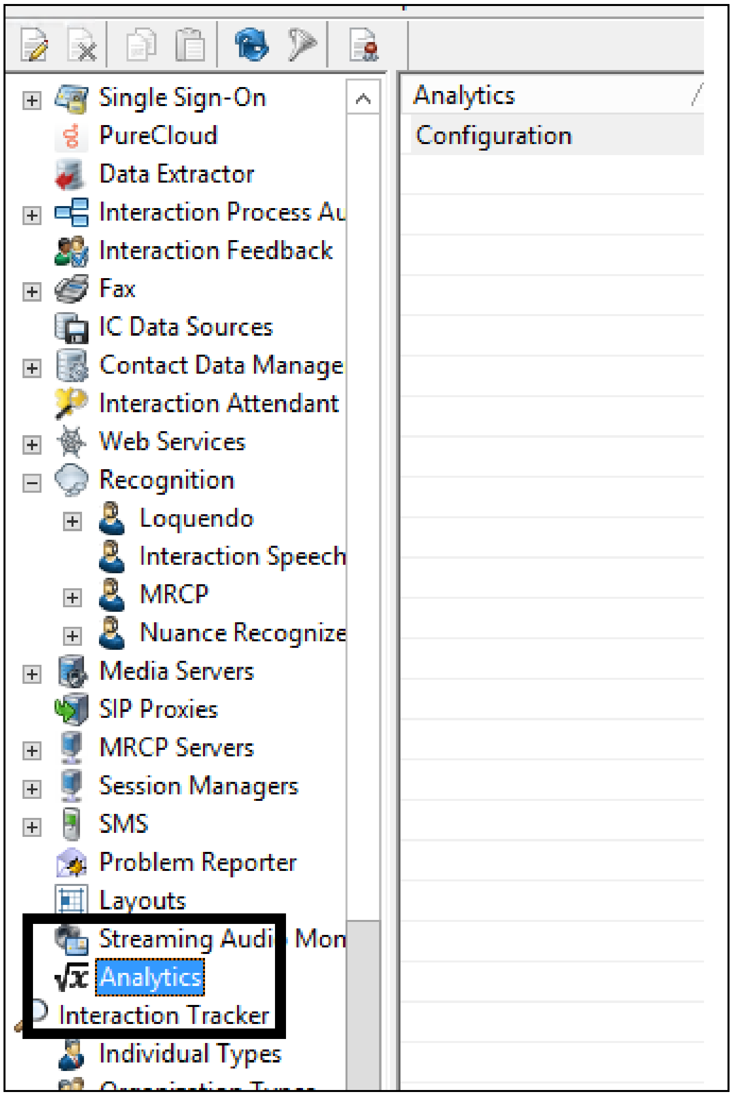
To assign an Analytics license to a user, select the **Analytics License** check box and select one of the following licenses.

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

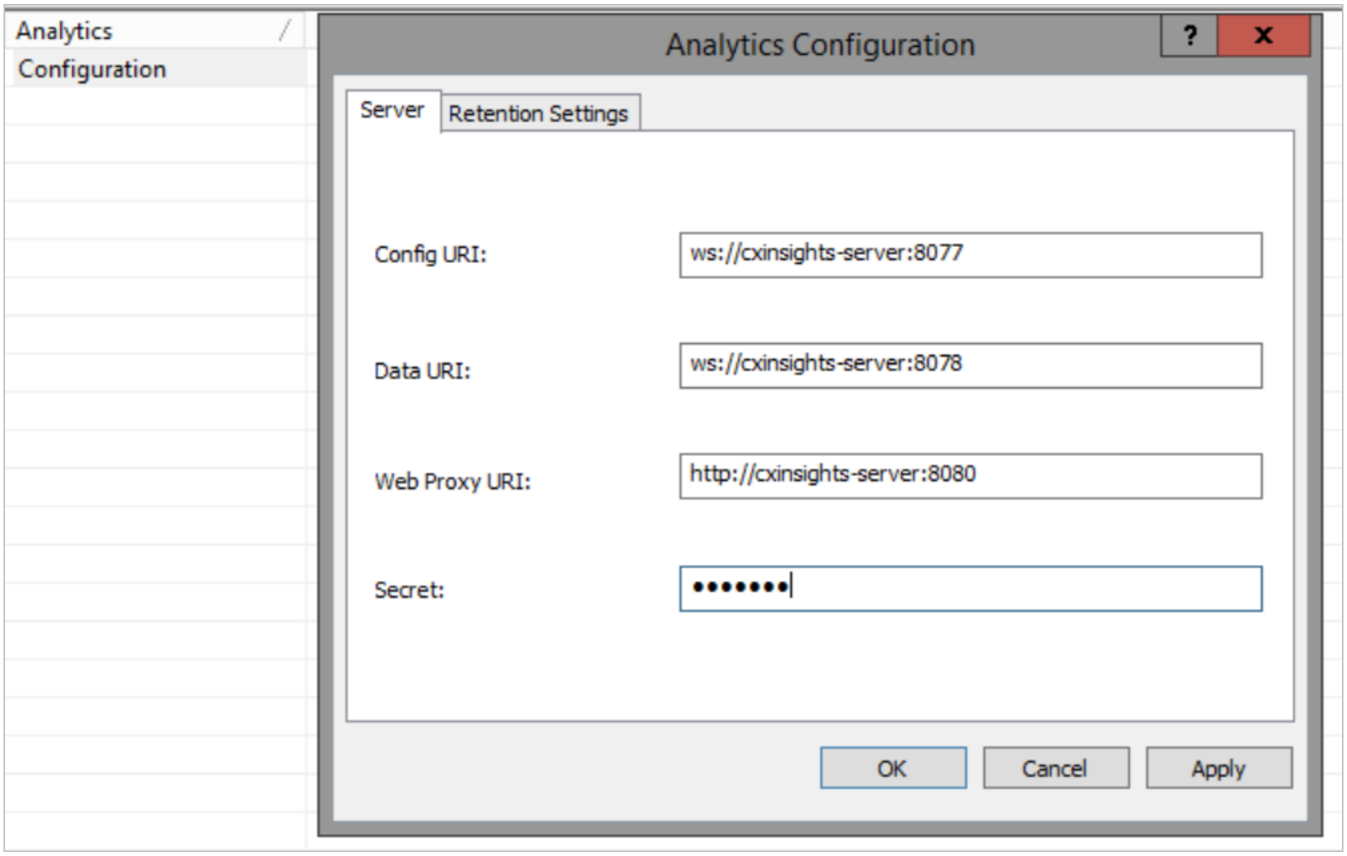
Configure CX Insights server in Interaction Administrator

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.



Config URI: ws://cxinsights-server:8077

Data URI: ws://cxinsights-server:8078

Web Proxy URI: http://cxinsights-server:8080

Secret: •••••••

- 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.

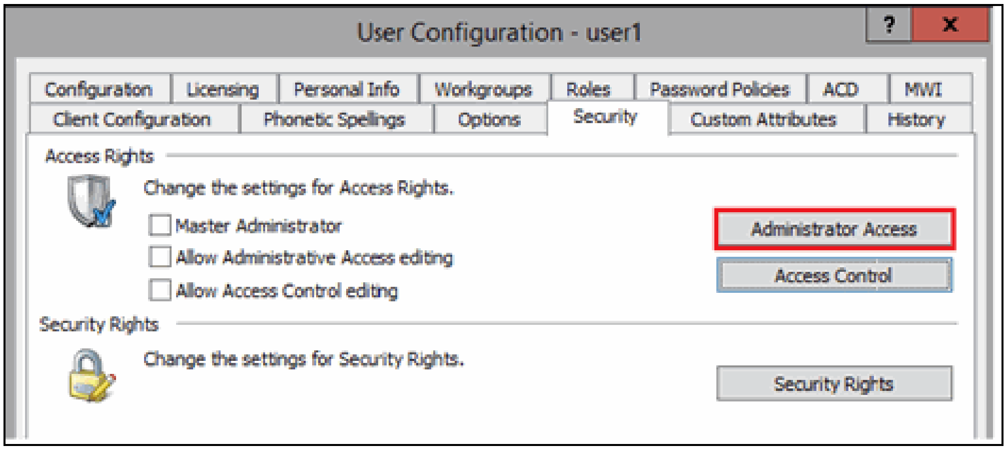
Once Configuration is complete, the AnalyticsBridge subsystem will attempt to make the configured websocket connections. If those are successful, the synchronization process will begin. This can take a few minutes to complete if there are a large number of users and workgroups to transfer. Any additional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



Access Rights

Change the settings for Access Rights.

□ Master Administrator

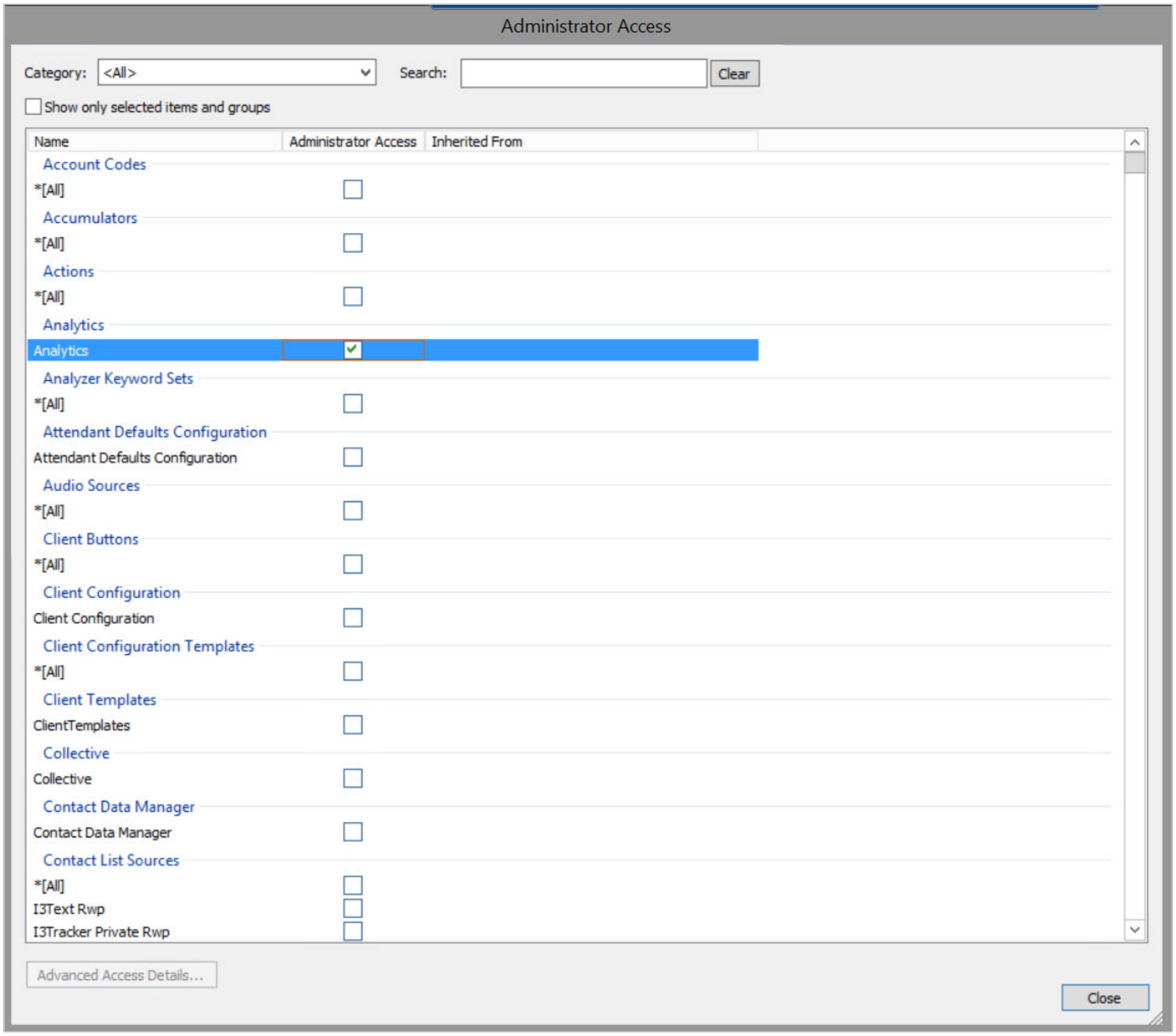
□ Allow Administrative Access editing

□ Allow Access Control editing

Security Rights

Change the settings for Security Rights.

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



Category: <All>

Search:

□ Show only selected items and groups

Name	Administrator Access	Inherited From
Account Codes		
*[All]	□	
Accumulators		
*[All]	□	
Actions		
*[All]	□	
Analytics		
Analytics	☑	
Analyzer Keyword Sets		
*[All]	□	
Attendant Defaults Configuration		
Attendant Defaults Configuration	□	
Audio Sources		
*[All]	□	
Client Buttons		
*[All]	□	
Client Configuration		
Client Configuration	□	
Client Configuration Templates		
*[All]	□	
Client Templates		
ClientTemplates	□	
Collective		
Collective	□	
Contact Data Manager		
Contact Data Manager	□	
Contact List Sources		
*[All]	□	
I3Text Rwp	□	
I3Tracker Private Rwp	□	

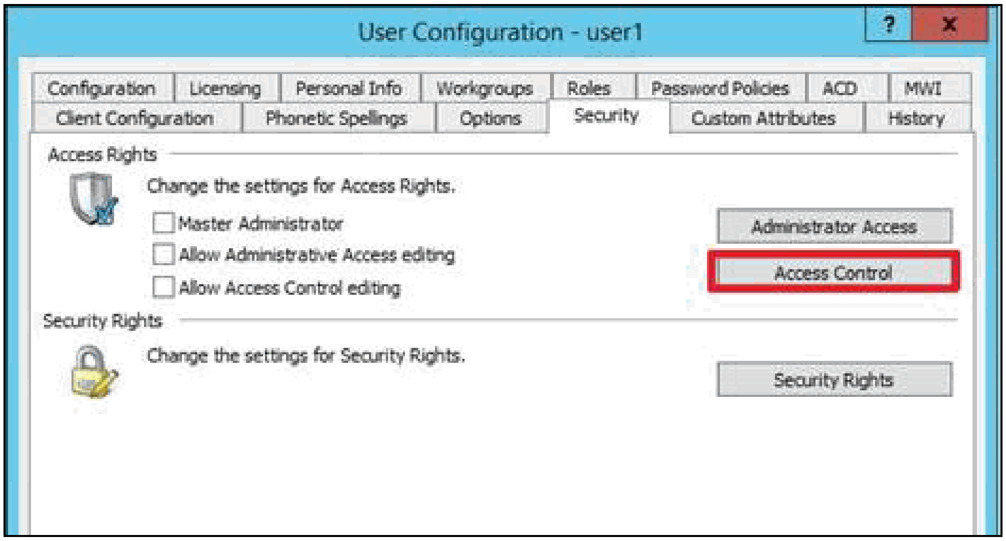
5. To give a user, workgroup, or role Administrator Rights to the Analytics feature, select the **Analytics** check box. You can clear the check box to remove the privilege.
6. Click **Close**.
7. To save the settings, click **OK** or **Apply**.

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.



Access Rights

Change the settings for Access Rights.

□ Master Administrator

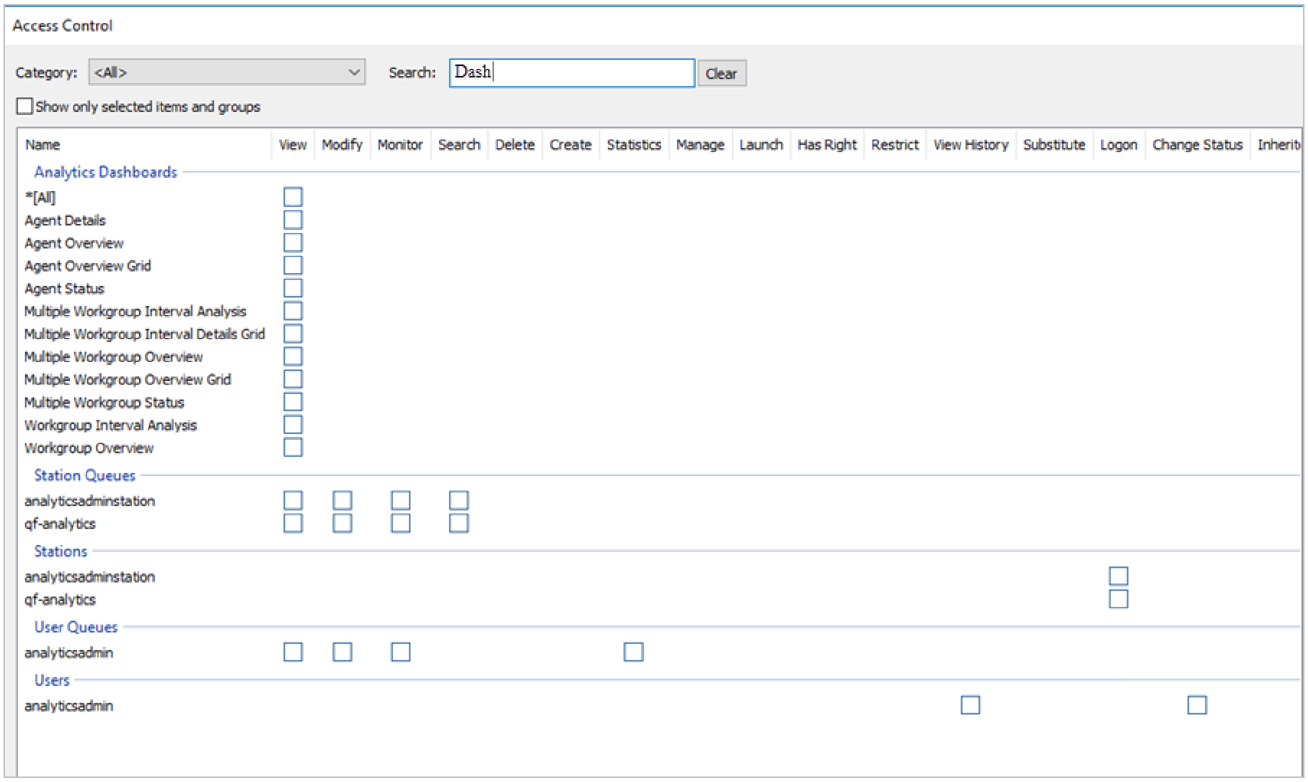
□ Allow Administrative Access editing

□ Allow Access Control editing

Security Rights

Change the settings for Security Rights.

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



Category: <All>

Search: Dash

□ Show only selected items and groups

Name	View	Modify	Monitor	Search	Delete	Create	Statistics	Manage	Launch	Has Right	Restrict	View History	Substitute	Logon	Change Status	Inherit
Analytics Dashboards																
*[All]	□															
Agent Details	□															
Agent Overview	□															
Agent Overview Grid	□															
Agent Status	□															
Multiple Workgroup Interval Analysis	□															
Multiple Workgroup Interval Details Grid	□															
Multiple Workgroup Overview	□															
Multiple Workgroup Overview Grid	□															
Multiple Workgroup Status	□															
Workgroup Interval Analysis	□															
Workgroup Overview	□															
Station Queues																
analyticsadminstation	□	□	□	□												
qf-analytics	□	□	□	□												
Stations																
analyticsadminstation														□		
qf-analytics														□		
User Queues																
analyticsadmin	□	□	□				□									
Users																
analyticsadmin												□			□	

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:

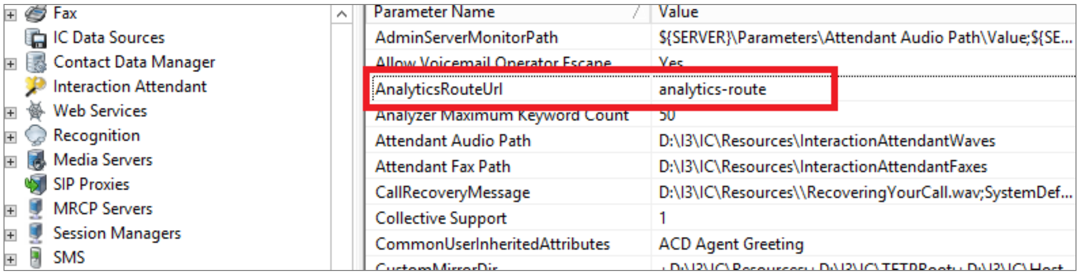
WebServer configuration

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:



Parameter Name	Value
AdminServerMonitorPath	${SERVER}\Parameters\Attendant Audio Path\Value;${SE...
Allow Voicemail Operator Escape	Yes
AnalyticsRouteUrl	analytics-route
Analyzer Maximum Keyword Count	50
Attendant Audio Path	D:\I3\IC\Resources\InteractionAttendantWaves
Attendant Fax Path	D:\I3\IC\Resources\InteractionAttendantFaxes
CallRecoveryMessage	D:\I3\IC\Resources\\RecoveringYourCall.wav;SystemDef...
Collective Support	1
CommonUserInheritedAttributes	ACD Agent Greeting
CustomMirrorDir	;D:\I3\IC\Resources;;D:\I3\IC\TFTPRoot;;D:\I3\IC\Host

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.

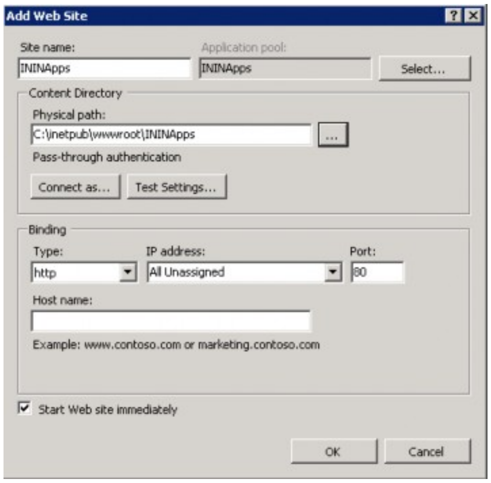
   > **Note:**
   > In this document, the directory is named **ININApps**.

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.



Site name: ININApps
Application pool: ININApps
Content Directory
Physical path: C:\inetpub\wwwroot\ININApps
Pass-through authentication
Binding
Type: http
IP address: All Unassigned
Port: 80
Host name:
Example: www.contoso.com or marketing.contoso.com
☑ Start Web site immediately

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

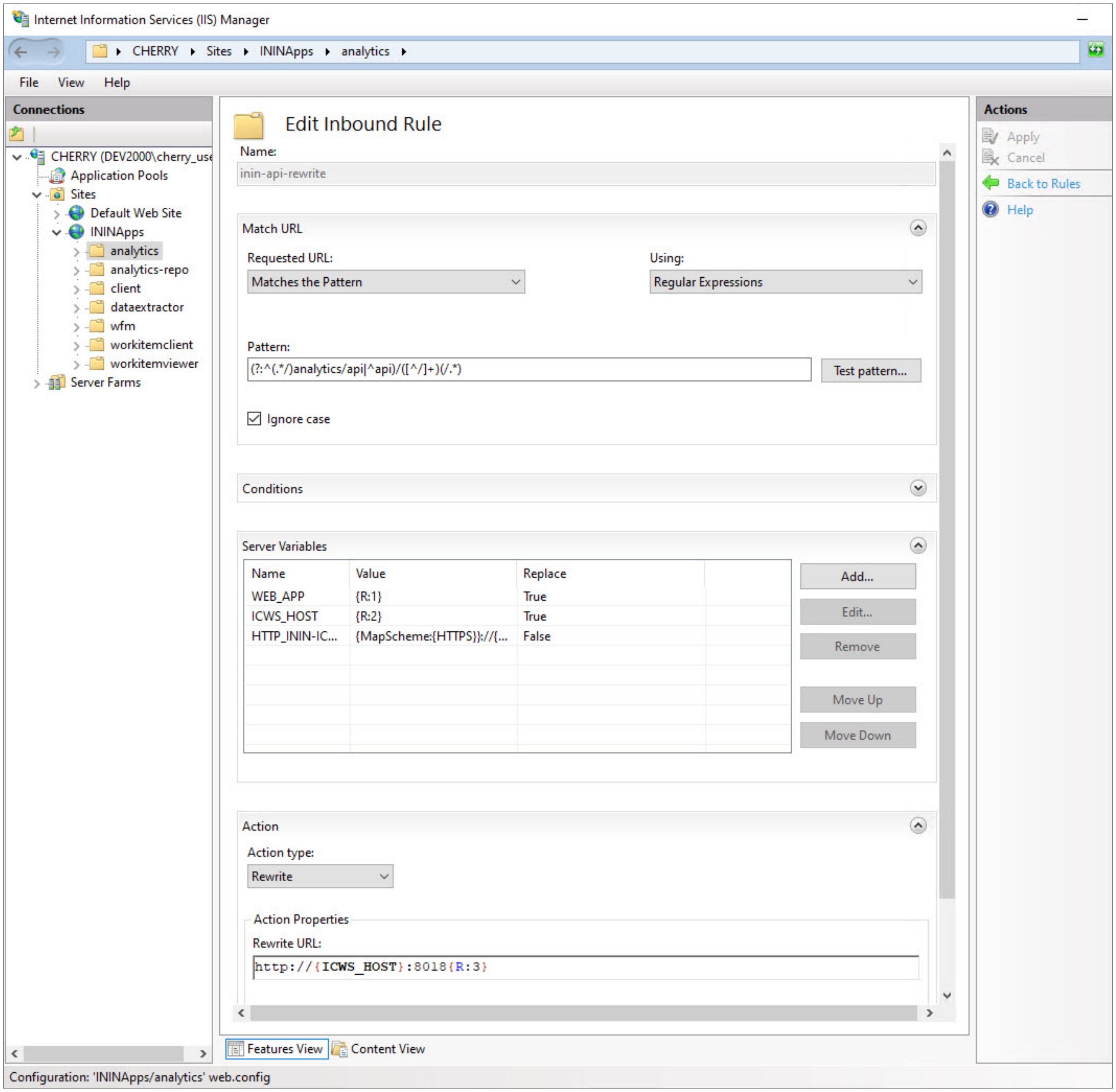
7. Create URL rewrite rules. You will create two inbound rules and four outbound rules.
   a. Click the site in the **IIS Manager**.
   b. Double-click the **URL Rewrite** module.
   c. Navigate to the **Actions** pane and select **Add Rule(s)**.
   d. For each rule, select **Blank rule** under the appropriate 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



Edit Inbound Rule

Name: inin-api-rewrite

Match URL

Requested URL: Matches the Pattern

Using: Regular Expressions

Pattern: (?:^(.*/)analytics/api|^api)/([^/]+)(/.*)

☑ Ignore case

Conditions

Server Variables

Name	Value	Replace
WEB_APP	{R:1}	True
ICWS_HOST	{R:2}	True
HTTP_ININ-IC...	{MapScheme:{HTTPS}}://{...	False

Action

Action type: Rewrite

Action Properties

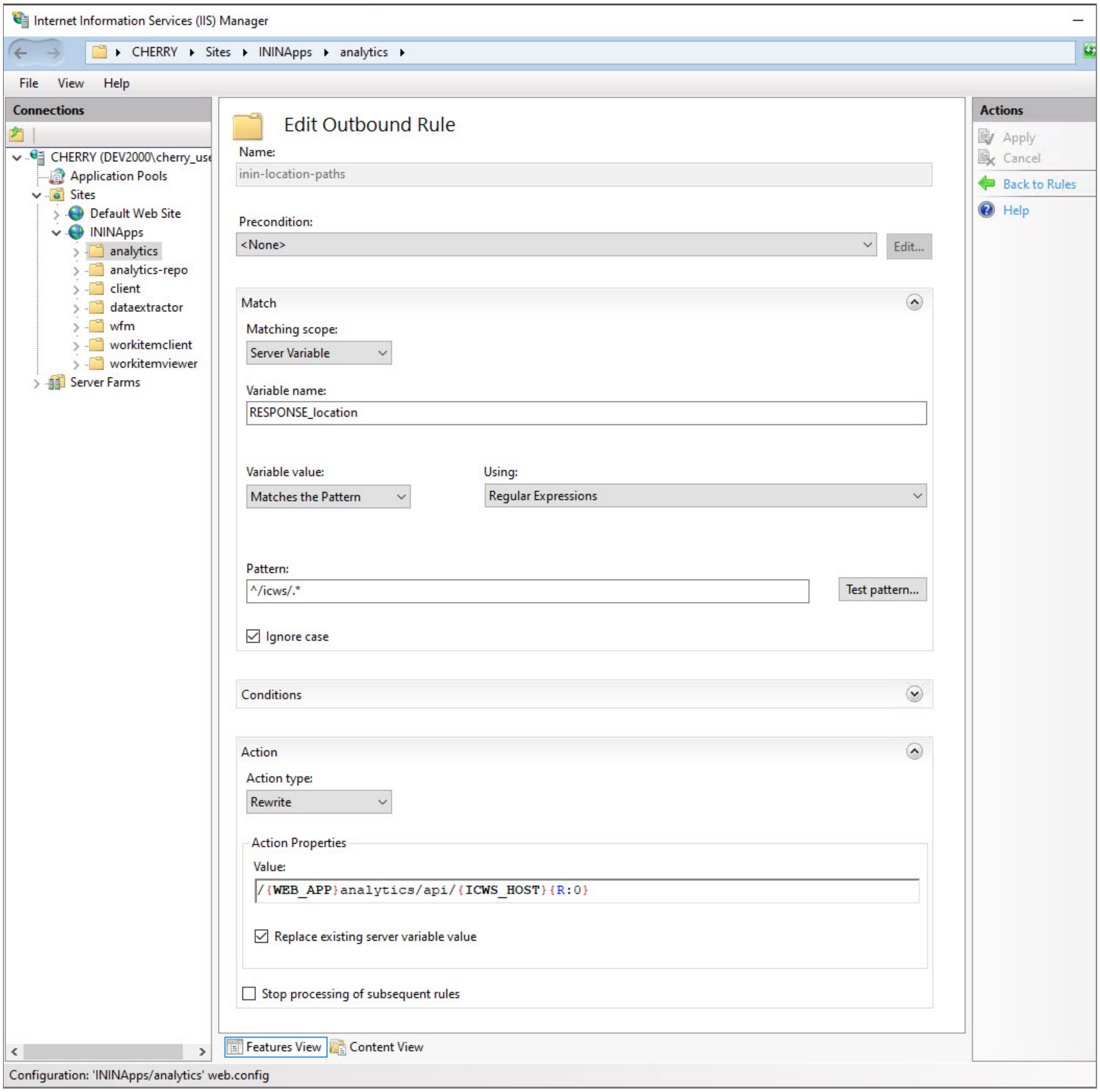
Rewrite URL: http://{ICWS_HOST}:8018{R:3}

Configuration: 'ININApps/analytics' web.config

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



Edit Outbound Rule

Name: inin-location-paths

Precondition: <None>

Match

Matching scope: Server Variable

Variable name: RESPONSE_location

Variable value: Matches the Pattern

Using: Regular Expressions

Pattern: ^/icws/.*

☑ Ignore case

Conditions

Action

Action type: Rewrite

Action Properties

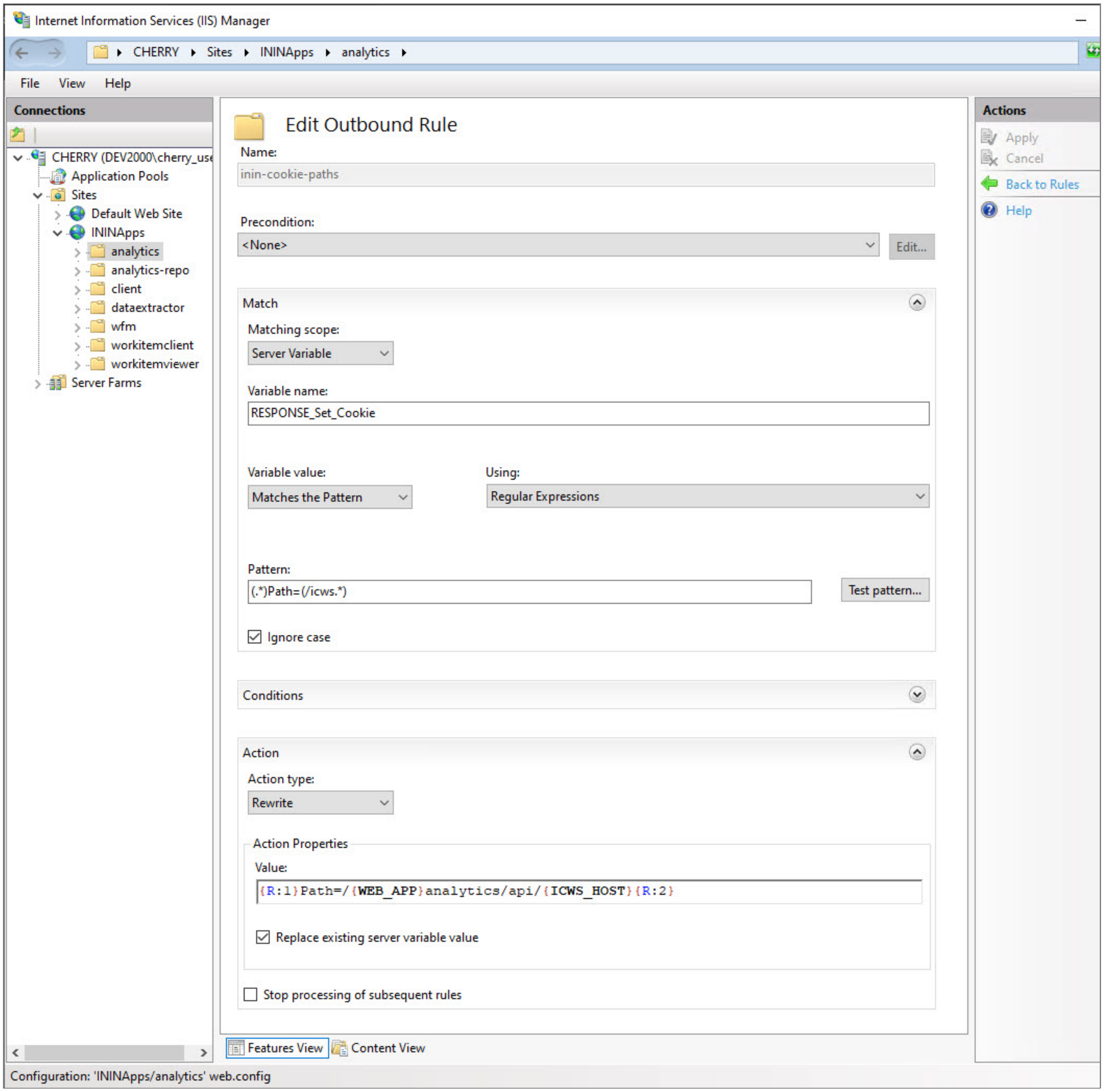
Value: /{WEB_APP}analytics/api/{ICWS_HOST}{R:0}

☑ Replace existing server variable value

☐ Stop processing of subsequent rules

Configuration: 'ININApps/analytics' web.config

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



Edit Outbound Rule

Name: inin-cookie-paths

Precondition: <None>

Match

Matching scope: Server Variable

Variable name: RESPONSE_Set_Cookie

Variable value: Matches the Pattern

Using: Regular Expressions

Pattern: (.*)Path=(/icws.*)

☑ Ignore case

Action

Action type: Rewrite

Action Properties

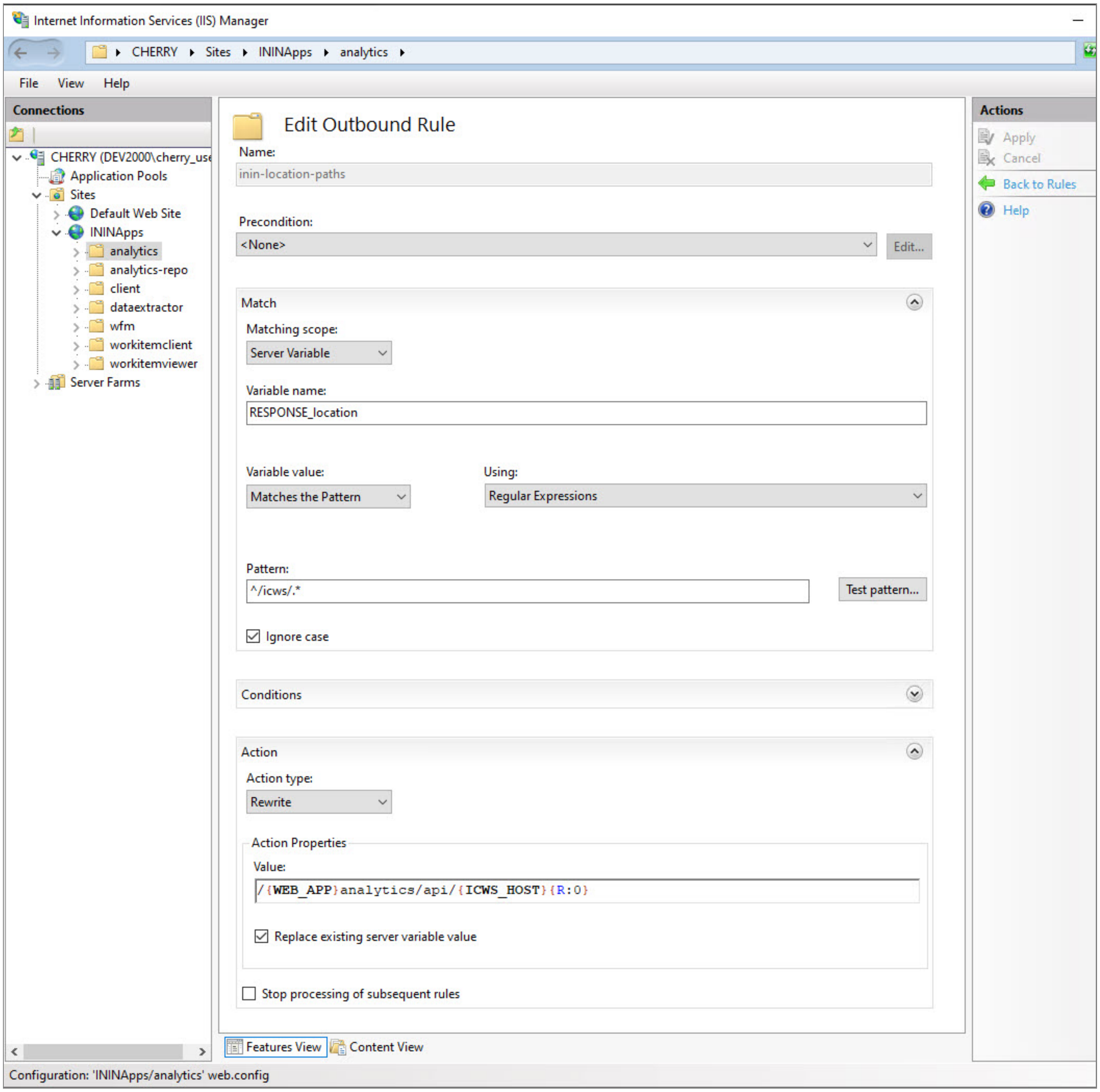
Value: {R:1}Path=/{WEB_APP}analytics/api/{ICWS_HOST}{R:2}

☑ Replace existing server variable value

☐ Stop processing of subsequent rules

Configuration: 'ININApps/analytics' web.config

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



Edit Outbound Rule

Name: inin-location-paths

Precondition: <None> Edit...

Match

Matching scope: Server Variable

Variable name: RESPONSE_location

Variable value: Matches the Pattern

Using: Regular Expressions

Pattern: ^/icws/.* Test pattern...

☑ Ignore case

Conditions

Action

Action type: Rewrite

Action Properties

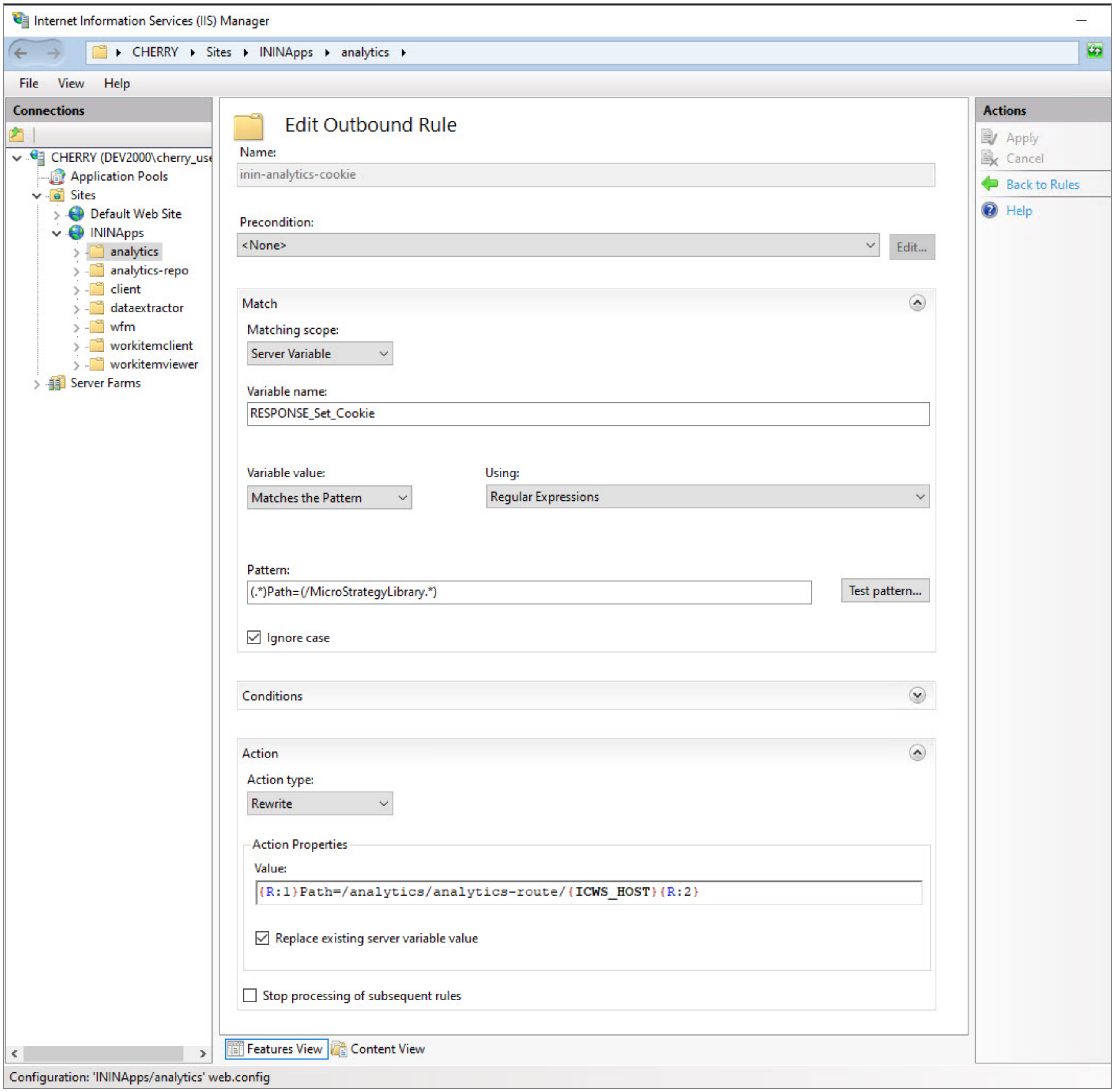
Value: /{WEB_APP}analytics/api/{ICWS_HOST}{R:0}

☑ Replace existing server variable value

☐ Stop processing of subsequent rules

Configuration: 'ININApps/analytics' web.config

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



Edit Outbound Rule

Name:
inin-analytics-cookie

Precondition:
<None>
Edit...

Match

Matching scope:
Server Variable

Variable name:
RESPONSE_Set_Cookie

Variable value:
Matches the Pattern

Using:
Regular Expressions

Pattern:
(.*)Path=(/MicroStrategyLibrary.*)
Test pattern...

☑ Ignore case

Conditions

Action

Action type:
Rewrite

Action Properties

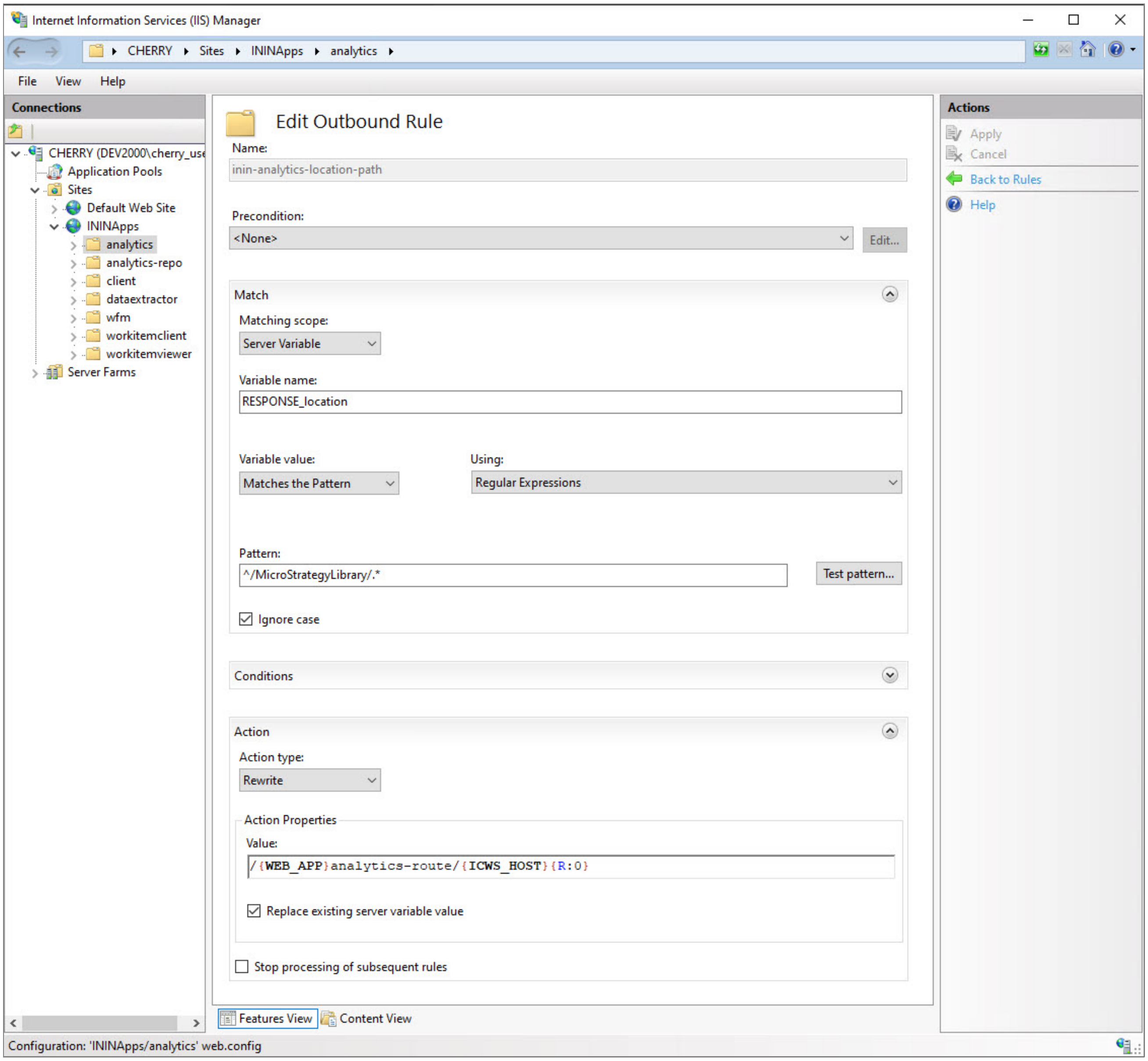
Value:
{R:1}Path=/analytics/analytics-route/{ICWS_HOST}{R:2}

☑ Replace existing server variable value

☐ Stop processing of subsequent rules

Configuration: 'ININApps/analytics' web.config

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



Edit Outbound Rule

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

Action type: Rewrite

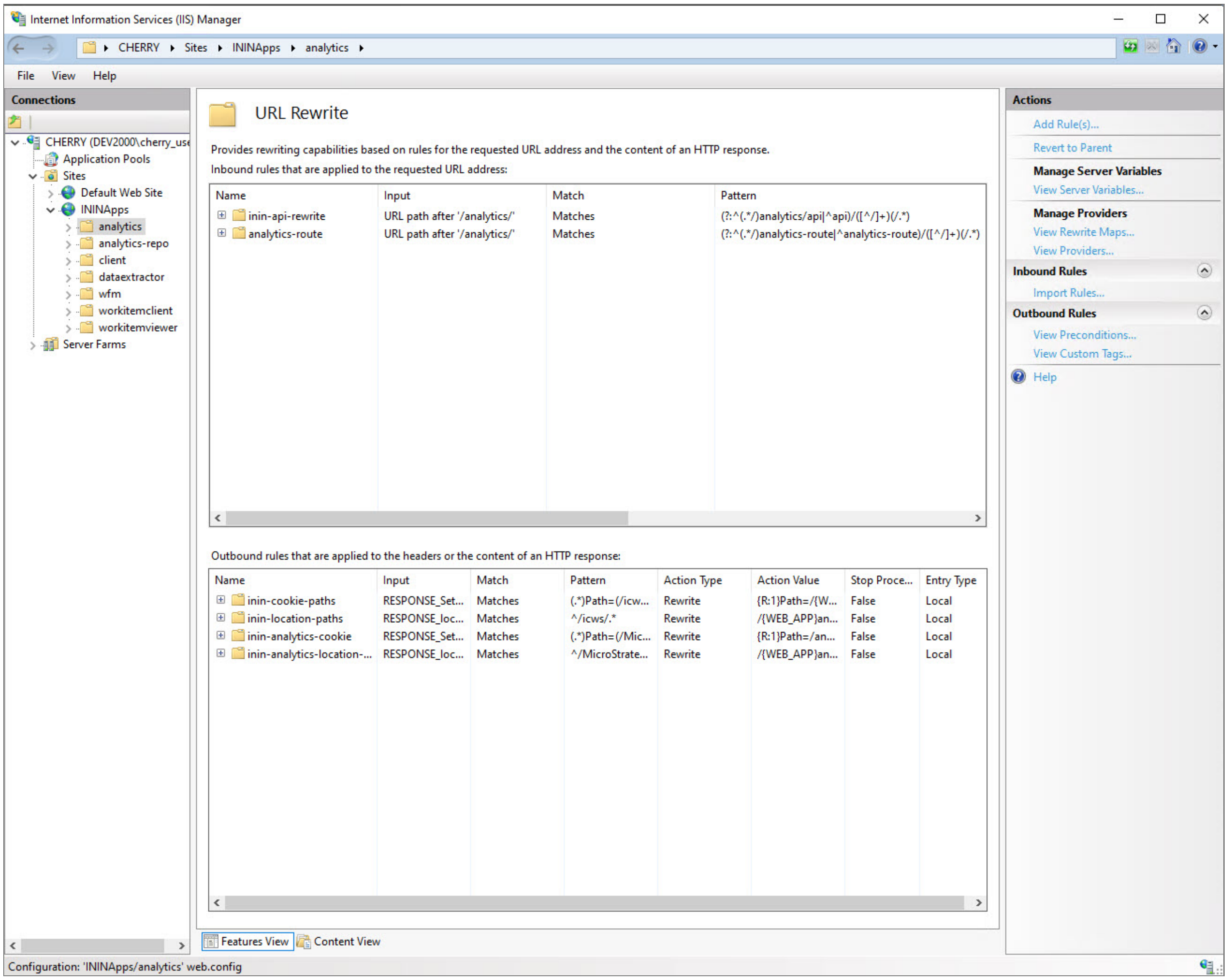
Value: /{WEB_APP}analytics-route/{ICWS_HOST}{R:0}

☑ Replace existing server variable value

☐ Stop processing of subsequent rules

Configuration: 'ININApps/analytics' web.config

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



URL Rewrite

Provides rewriting capabilities based on rules for the requested URL address and the content of an HTTP response.

Inbound rules that are applied to the requested URL address:

Name	Input	Match	Pattern
inin-api-rewrite	URL path after '/analytics/'	Matches	(?:^(.*/)analytics/api|^api)/([^/]+)(/.*)
analytics-route	URL path after '/analytics/'	Matches	(?:^(.*/)analytics-route|^analytics-route)/([^/]+)(/.*)

Outbound rules that are applied to the headers or the content of an HTTP response:

Name	Input	Match	Pattern	Action Type	Action Value	Stop Proce...	Entry Type
inin-cookie-paths	RESPONSE_Set...	Matches	(.*)Path=(/icw...	Rewrite	{R:1}Path=/{W...	False	Local
inin-location-paths	RESPONSE_loc...	Matches	^/icws/.*	Rewrite	/{WEB_APP}an...	False	Local
inin-analytics-cookie	RESPONSE_Set...	Matches	(.*)Path=(/Mic...	Rewrite	{R:1}Path=/an...	False	Local
inin-analytics-location-...	RESPONSE_loc...	Matches	^/MicroStrate...	Rewrite	/{WEB_APP}an...	False	Local

Configuration: 'ININApps/analytics' web.config

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`.

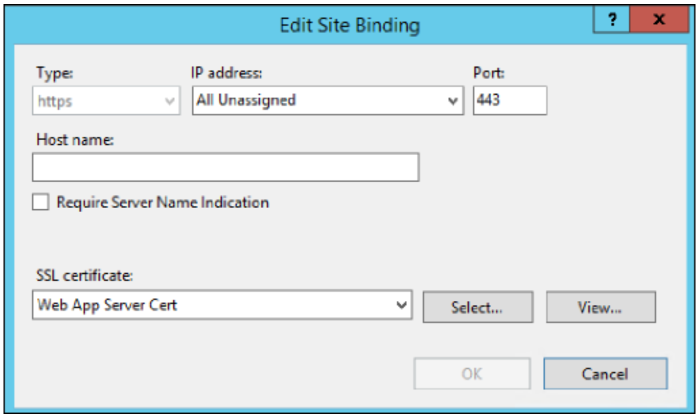
6. Click **Next**.
7. Choose the appropriate cryptographic service provider properties. Ask your third-party Certificate Authority (CA) which options to choose.
8. Click **Next**.
9. Enter a file name and location for the CSR.
10. Click **Finish**.
11. Send the generated CSR to your CA for signing.

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**.


Edit Site Binding

Type: https

IP address: All Unassigned

Port: 443

Host name:

□ Require Server Name Indication

SSL certificate: Web App Server Cert

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.

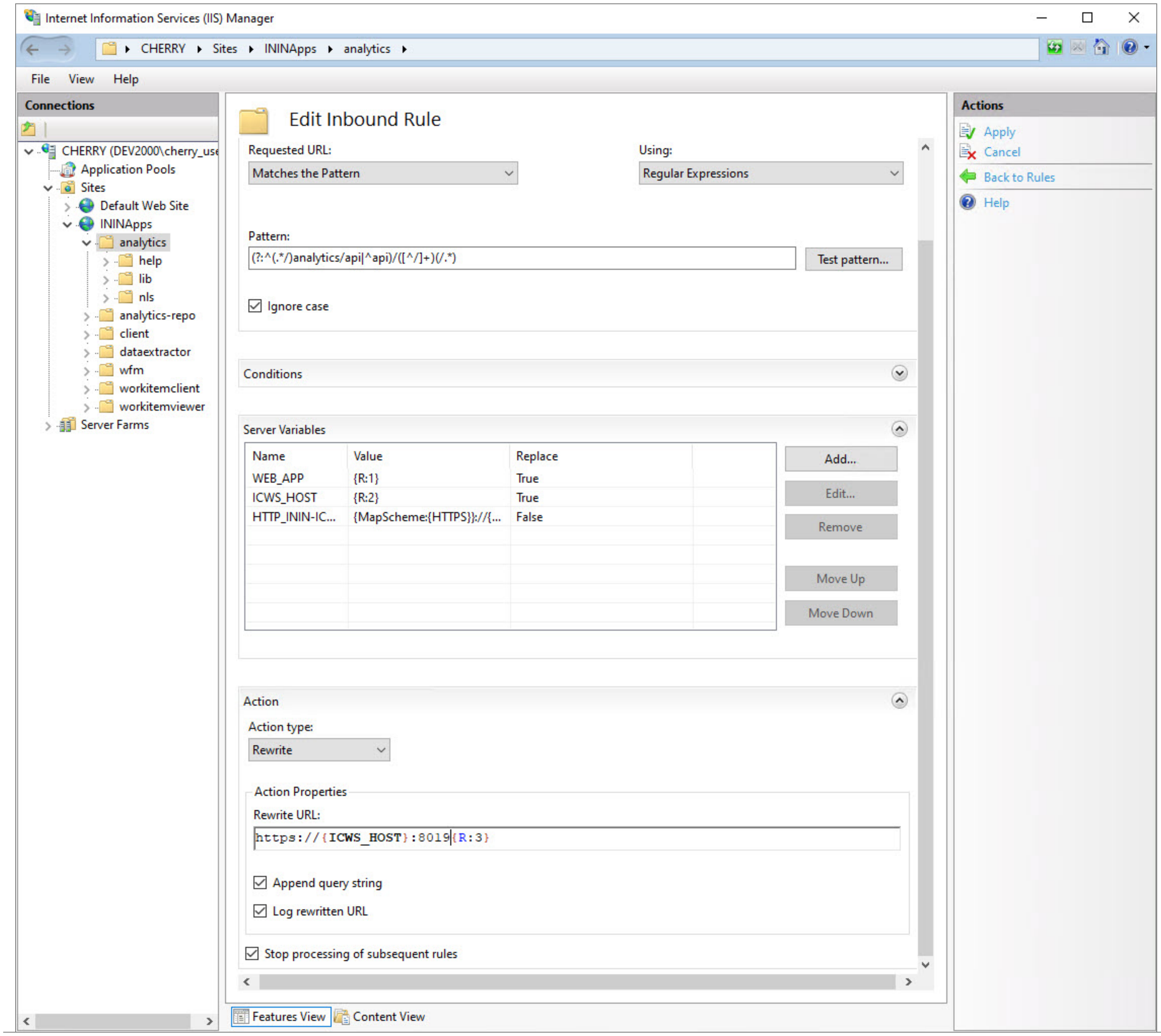
Enable HTTPS between IIS and CIC

> **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.

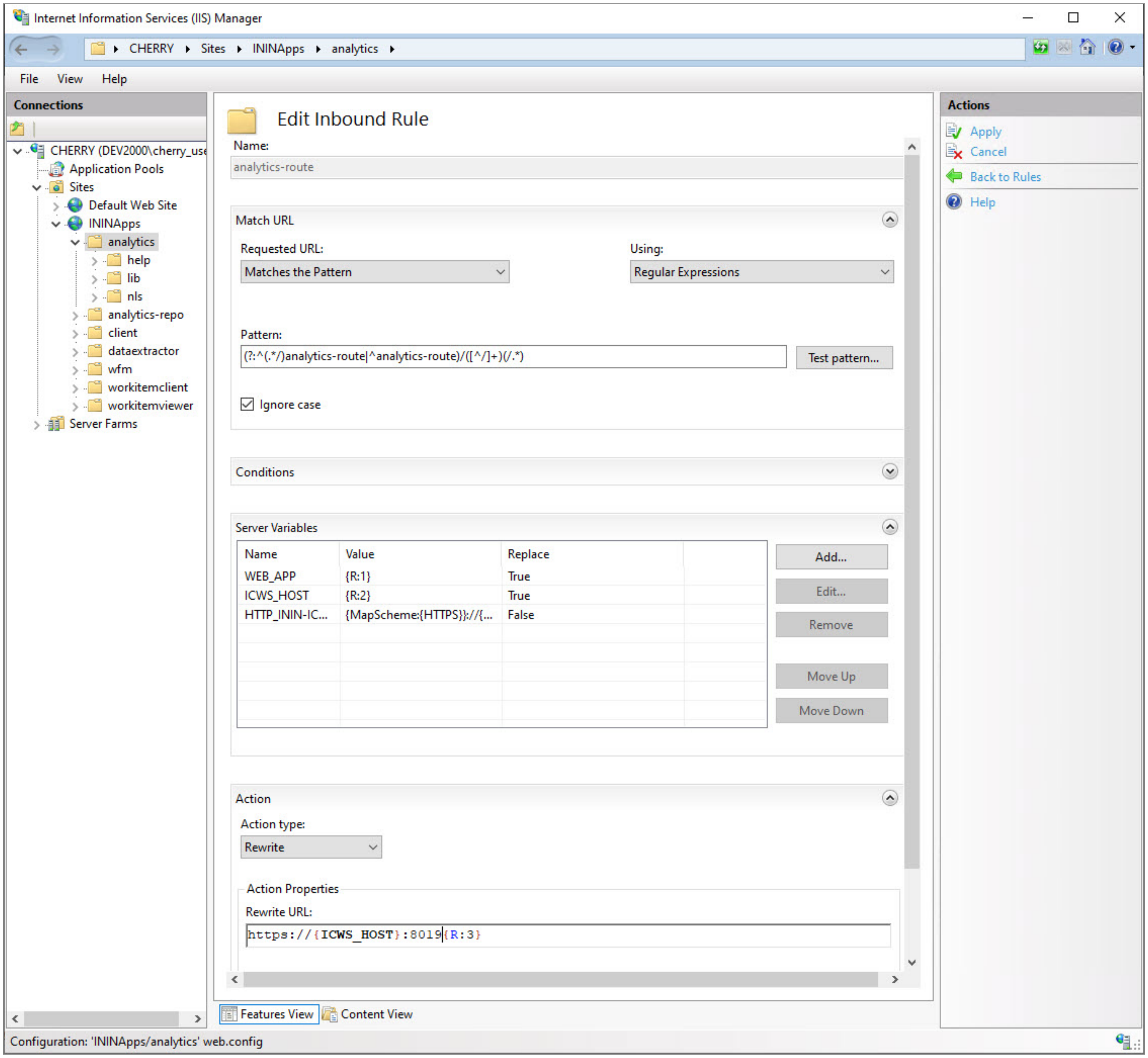
These directions encrypt the connection between the web server and the CIC server.

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**.



Configuration: 'ININApps/analytics' web.config



Edit Inbound Rule

Name:
analytics-route

Match URL

Requested URL: Matches the Pattern

Using: Regular Expressions

Pattern:
(?:^(.*/)analytics-route|^analytics-route)/([^/]+)(/.*)

☑ Ignore case

Conditions

Server Variables

Name	Value	Replace
WEB_APP	{R:1}	True
ICWS_HOST	{R:2}	True
HTTP_ININ-IC...	{MapScheme:{HTTPS}}://{...	False

Action

Action type: Rewrite

Action Properties

Rewrite URL:
`https://{ICWS_HOST}:8019{R:3}`

Configuration: 'ININApps/analytics' web.config

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;
}
}
```

Post deployment and installation verification

- To check if the deployment is successful or not. Use `docker ps` command to verify if all the containers are up and running.
- To check if mstrWeb dashboard page loading properly or not, use http://<host ip>:8080/MicroStrategy/servlet/mstrWeb URL.
- To check whether all the required ports are opened or not. Use `firewall-cmd --list-ports` command.
- To access container's log, use this `docker logs container_id -follow` command
- For an example: `docker logs 3bff --follow` where 3bff are the first characters of a container ID
- To check whether connector is up or not, wait till the below logs appear.

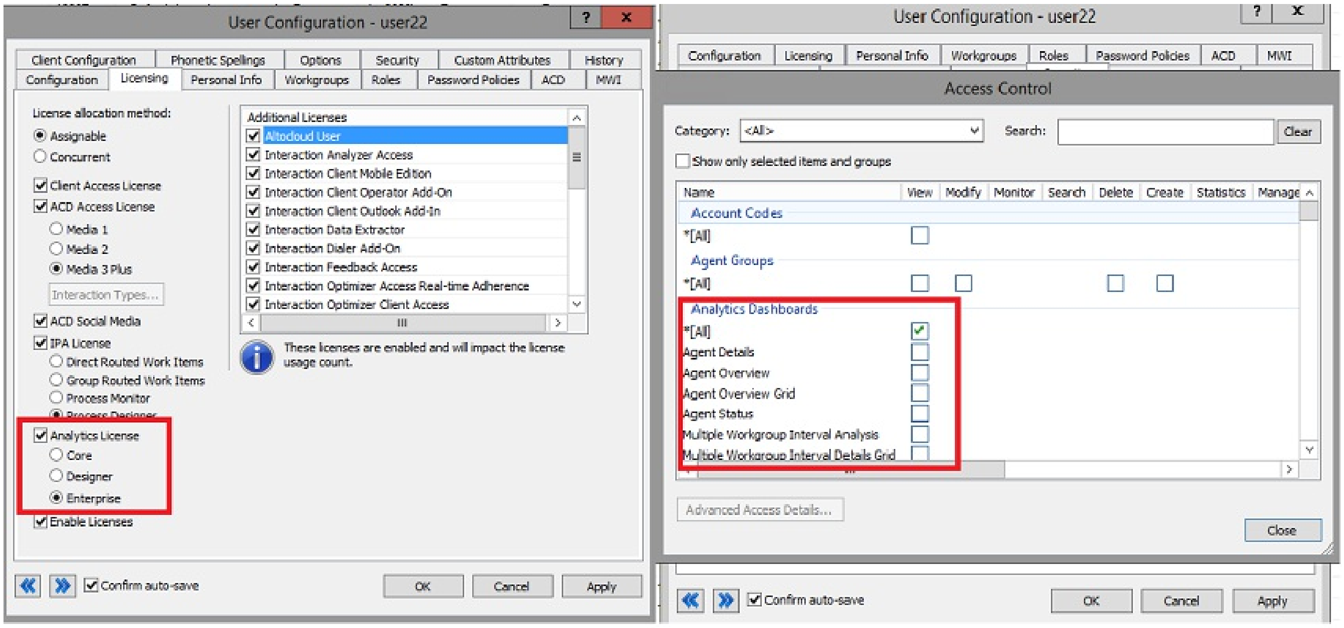
```
info [2020-02-06T11:16:03.185Z] - MicroStrategyConnector:Prometheus Prometheus started on port: 9090

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
```

- To check whether the dataadapter server is up or not, wait til the below logs appear.

```
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
```

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.



User Configuration - user22

Client Configuration | Phonetic Spellings | Options | Security | Custom Attributes | History | Configuration | Licensing | Personal Info | Workgroups | Roles | Password Policies | ACD | MWI

License allocation method:
● Assignable
○ Concurrent

☑ Client Access License
☑ ACD Access License
○ Media 1
○ Media 2
● Media 3 Plus
Interaction Types...
☑ ACD Social Media
☑ IPA License
○ Direct Routed Work Items
○ Group Routed Work Items
○ Process Monitor
● Process Designer
☑ Analytics License
○ Core
○ Designer
● Enterprise
☑ Enable Licenses

Additional Licenses
☑ Altocloud User
☑ Interaction Analyzer Access
☑ Interaction Client Mobile Edition
☑ Interaction Client Operator Add-On
☑ Interaction Client Outlook Add-In
☑ Interaction Data Extractor
☑ Interaction Dialer Add-On
☑ Interaction Feedback Access
☑ Interaction Optimizer Access Real-time Adherence
☑ Interaction Optimizer Client Access

These licenses are enabled and will impact the license usage count.

☑ Confirm auto-save

OK | Cancel | Apply

User Configuration - user22

Configuration | Licensing | Personal Info | Workgroups | Roles | Password Policies | ACD | MWI

Access Control

Category: <All>

Search:

Clear

□ Show only selected items and groups

Name	View	Modify	Monitor	Search	Delete	Create	Statistics	Manage
Account Codes								
*[All]	□							
Agent Groups								
*[All]	□	□			□	□		
Analytics Dashboards								
*[All]	☑							
Agent Details	□							
Agent Overview	□							
Agent Overview Grid	□							
Agent Status	□							
Multiple Workgroup Interval Analysis	□							
Multiple Workgroup Interval Details Grid	□							

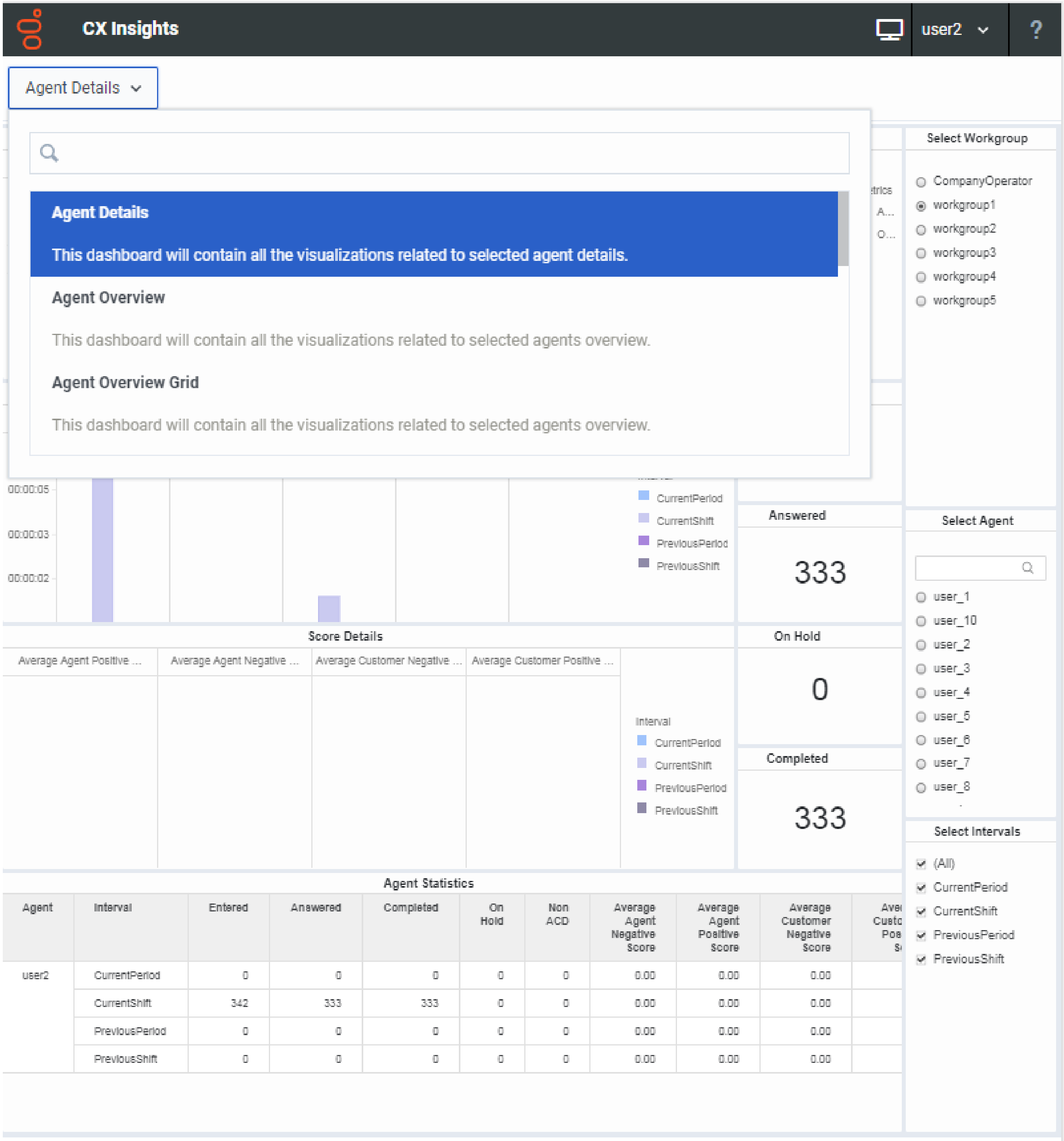
Advanced Access Details...

Close

☑ Confirm auto-save

OK | Cancel | Apply

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.



Agent Details

This dashboard will contain all the visualizations related to selected agent details.

Agent Overview

This dashboard will contain all the visualizations related to selected agents overview.

Agent Overview Grid

This dashboard will contain all the visualizations related to selected agents overview.

Answered

333

On Hold

0

Completed

333

Agent Statistics

Agent	Interval	Entered	Answered	Completed	On Hold	Non ACD	Average Agent Negative Score	Average Agent Positive Score	Average Customer Negative Score	Ave Custo Pos S
user2	CurrentPeriod	0	0	0	0	0	0.00	0.00	0.00	
	CurrentShift	342	333	333	0	0	0.00	0.00	0.00	
	PreviousPeriod	0	0	0	0	0	0.00	0.00	0.00	
	PreviousShift	0	0	0	0	0	0.00	0.00	0.00	

Troubleshooting

Errors	Description	Solution

No route to host	The container ports are blocked by firewall.	To resolve this error, run the following commands to allow the ports.
		`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 these changes, if the problem exists, ensure that there is no IP conflict with Linux host machine and docker container IP. In case if there is conflict then following the below steps
		Change docker daemon IP range. Add following line `{"bip":  in file etc/docker/daemon.json}`

Ansible host key checking error Example: `FAILED! =>{"msg": "Using a SSH password instead of a key is not possible because Host Key checking is enabled and sshpass does not support this. Please add this host's fingerprint to your known_hosts file to manage this host."}`	When you run the ansible for the first time, ansible host key check is not set.	To resolve this error, run this command `ANSIBLE_HOST_KEY_CHECKING=False`
Incorrect URL re-write rules error	This error may occur due to any of the following reasons: When re-write rules are configured incorrectly in IIS or when JSession and ISession cookie appear or when a microstrategy login screen appears.	To resolve this error, do the following: • Ensure that the URL rewrite rules are correct, refer to the file web.config of IIS. • Check both JSession and ISession cookie paths should be same. If not same, then change the IIS outbound rule inin-analytics-cookie path `{R:1}path=/{WEB_APP}analytics/analytics-route/{ICWS_HOST} {R:2}` for analytics project. For an example: JSession path: `/analytics/analytics-route/calvyn.dev2000.com/MicroStrategyLibrary` ISession path: `/analytics/analytics-route/calvyn.dev2000.com/MicroStrategyLibrary`
User dashboard access permission error *You do not have access to view dashboards, please contact your administrator* message appears on screen instead of dashboards.	If user do not have access permissions to the dashboards this error may occur	To resolve this error, permit access to the dashboard in Interaction Administrator, go to User Properties → Security→AccessControl screen and select the required dashboard check box to access the dashboards.
Connectivity problem between AnalyticsBridge and mstr connector	If a white blank screen without any content in that dashboard appears it indicates that an error occurred.	To resolve this error, check the connectivity between mstr connector and AnalyticBridge, and then restart AnalyticsBridge component.
Connectivity problem between AnalyticsBridge and dataadapter server	If the data is not updated in the dashboard, but there any valid statistics shown in ICBM it indicates that an error occurred.	To resolve this error, check the connectivity between dataadapter server and AnalyticBridge and then restart AnalyticsBridge component.

Known Errors

Note: The below mentioned known errors will not effect any environment

Errors	Description	Solution
Login Error	Sometimes login screen displays *user is not licensed* message even after user has a valid license	Ignore the error message displayed and try to login again.
Multiple user session error	If user login with one account and logged in again in another browser with the same account previous session will be deleted and even if user is idle for long time it shows this error	After closing all the existing user sessions then try to login again.

Appendix

MicroStrategy Server License Update Process

The MicroStrategy server instance that runs in the container has a pre-activated key, which is required for the operation of MicroStrategy. This pre-activated temporary key with limited life is to facilitate uninterrupted deployment and testing in the production environment. The following procedure describes the steps required to update the key.

Note: You need to request for a new license key, based on the MicroStrategy version and validity of license.

If you are a new CX Insights customer or an existing customer, renewing contract or upgrading CIC version, must check for the validity of your MicroStrategy container license and request a new license key using the prescribed license ordering process. The MicroStrategy version may or may not change for CIC release. If the MicroStrategy version change then you must raise an Activation File Request (AFR) for new MicroStrategy version license key. For CIC and CX Insights version mapping view the below table.

CX Insights Version	EIC Release	MicroStrategy Version
1.0	2019 R4	10.11
1.0	2020 R1	10.11
2.0	2020 R2	10.11
3.0	2020 R3	2020
4.0	2020 R4	2020
4.0	2021 R1	2020
4.0	2021 R2	2020

License Ordering Process

The license ordering process is taken care by the Sales Engineers for Customers, so the Customers must contact their account executives to initiate the process. There are two types of license key models available based on the requirements of Customer/Partners, you can select the best suited model. The following are the two types of license key models available.

For Perpetual model

If you have purchased the Stock Keeping Unit (SKU)/ Part Number, but was granted with the temporary file. Then, you need to submit the Activation File Request (AFR) and communicate to Genesys Licensing Team. For more information, see Request a License File.

For Subscription model

If you have the subscription file, then the file is always temporary with the end date locked on the subscription date. The requests for the subscription files should include the corresponded subscription Sales Order number or a copy of the software delivery notice that includes Sale Order number.

License Request Checklist

Scenario	Request for a License
New CX Insights Customer on boarded	Yes
Existing CX Insights Perpetual Customer	Yes
Existing Perpetual Customer, who is moving to a higher MicroStrategy version due to CIC version upgrade	Yes
Existing Perpetual Customer, who is upgrading their CIC version but has the identical MicroStrategy version in both the CIC versions	No
Existing CX Insights Subscription Customer, who is renewing the contract	Yes
Existing CX Insights Subscription Customer, who is upgrading to a higher CIC version within the contract tenure but the MicroStrategy version mapped to the future CIC version is different from the existing CIC version	Yes
Existing CX Insights Subscription Customer, who is upgrading to a higher CIC version within the contract tenure but the MicroStrategy version mapped to the future CIC version is identical as the existing CIC version	No

Process of Updating new License Key

To update a license key for a running container, you need to perform few commands inside the container by following the below instructions.

Prerequisites

- Contact your Genesys PureConnect representative to obtain a new license key.

Installing a new License Key

To enter into GCXI container instance, run the below command. Check for the name of the container, if the container name is according to **cxinsights_gcxi** change or not. To obtain the name of the container use **docker ps** command.

```
docker exec -it cxinsights_gcxi bash
```

Create update_license.scr file inside **/genesys/gcxi** folder with data below and save the file.

```
<your license key>
```

To update license, run the below command

```
cat /genesys/gcxi/update_license.scr | ${MSTR_INSTALL_HOME}/bin/mstrlicmgr -console
```

License Update Verification

After the license update is done, a log file is generated. To check the log file existence do the following:

1. Run the below command to enter into GCXI container instance. The name of the container you can get using **docker ps** command, if the name is not **cxinsights_gcxi** change accordingly before running the below command.

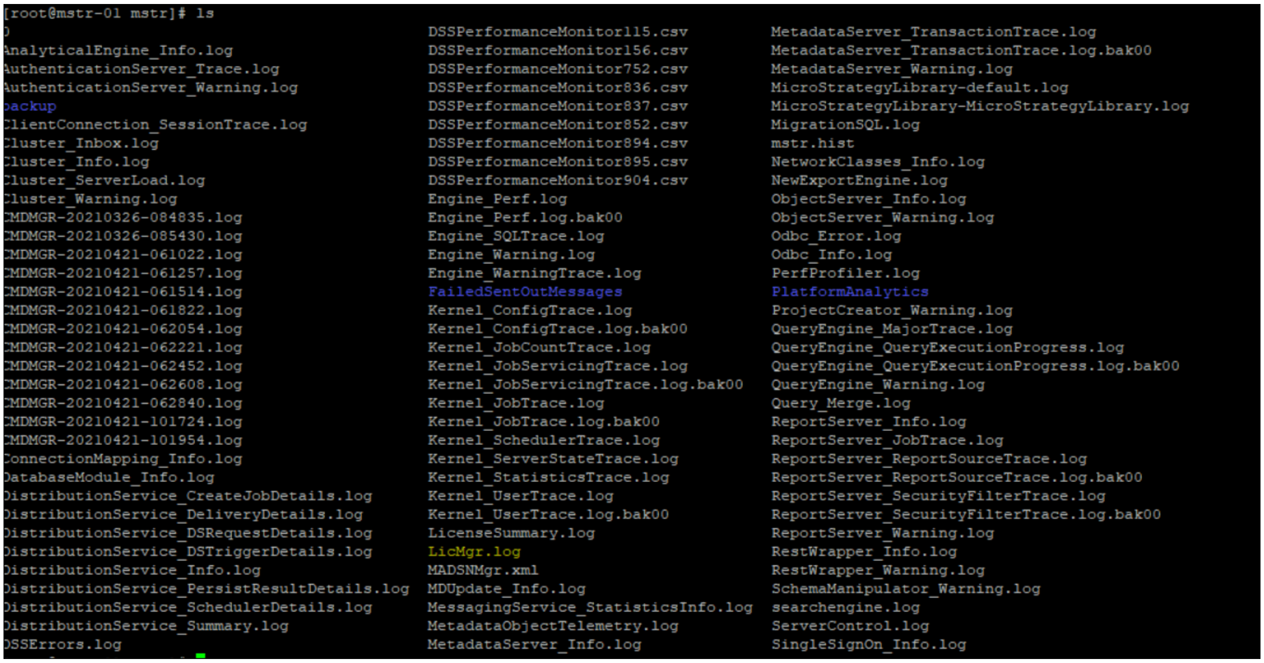
```
docker exec -it cxinsights_gcxi bash
```

2. It allows the user to go inside the GCXI container and navigate to the logging directory

```
cd /mnt/log/mstr
```

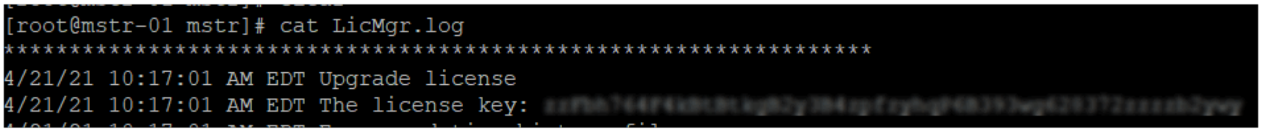
3. To get the list of files use the following command

```
ls
```



```
[root@mstr-01 mstr]# ls
0                                             DSSPerformanceMonitor115.csv         MetadataServer_TransactionTrace.log
AnalyticalEngine_Info.log                     DSSPerformanceMonitor156.csv         MetadataServer_TransactionTrace.log.bak00
AuthenticationServer_Trace.log                DSSPerformanceMonitor752.csv         MetadataServer_Warning.log
AuthenticationServer_Warning.log              DSSPerformanceMonitor836.csv         MicroStrategyLibrary-default.log
backup                                        DSSPerformanceMonitor837.csv         MicroStrategyLibrary-MicroStrategyLibrary.log
ClientConnection_SessionTrace.log             DSSPerformanceMonitor852.csv         MigrationSQL.log
Cluster_Inbox.log                             DSSPerformanceMonitor894.csv         mstr.hist
Cluster_Info.log                              DSSPerformanceMonitor895.csv         NetworkClasses_Info.log
Cluster_ServerLoad.log                        DSSPerformanceMonitor904.csv         NewExportEngine.log
Cluster_Warning.log                           Engine_Perf.log                      ObjectServer_Info.log
CMDMGR-20210326-084835.log                    Engine_Perf.log.bak00                ObjectServer_Warning.log
CMDMGR-20210326-085430.log                    Engine_SQLTrace.log                  Odbc_Error.log
CMDMGR-20210421-061022.log                    Engine_Warning.log                   Odbc_Info.log
CMDMGR-20210421-061257.log                    Engine_WarningTrace.log              PerfProfiler.log
CMDMGR-20210421-061514.log                    FailedSentOutMessages                PlatformAnalytics
CMDMGR-20210421-061822.log                    Kernel_ConfigTrace.log               ProjectCreator_Warning.log
CMDMGR-20210421-062054.log                    Kernel_ConfigTrace.log.bak00         QueryEngine_MajorTrace.log
CMDMGR-20210421-062221.log                    Kernel_JobCountTrace.log             QueryEngine_QueryExecutionProgress.log
CMDMGR-20210421-062452.log                    Kernel_JobServicingTrace.log         QueryEngine_QueryExecutionProgress.log.bak00
CMDMGR-20210421-062608.log                    Kernel_JobServicingTrace.log.bak00   QueryEngine_Warning.log
CMDMGR-20210421-062840.log                    Kernel_JobTrace.log                  Query_Merge.log
CMDMGR-20210421-101724.log                    Kernel_JobTrace.log.bak00            ReportServer_Info.log
CMDMGR-20210421-101954.log                    Kernel_SchedulerTrace.log            ReportServer_JobTrace.log
ConnectionMapping_Info.log                    Kernel_ServerStateTrace.log          ReportServer_ReportSourceTrace.log
DatabaseModule_Info.log                       Kernel_StatisticsTrace.log           ReportServer_ReportSourceTrace.log.bak00
DistributionService_CreateJobDetails.log      Kernel_UserTrace.log                 ReportServer_SecurityFilterTrace.log
DistributionService_DeliveryDetails.log       Kernel_UserTrace.log.bak00           ReportServer_SecurityFilterTrace.log.bak00
DistributionService_DSRequestDetails.log      LicenseSummary.log                   ReportServer_Warning.log
DistributionService_DSTriggerDetails.log      LicMgr.log                           RestWrapper_Info.log
DistributionService_Info.log                  MADSNMgr.xml                         RestWrapper_Warning.log
DistributionService_PersistResultDetails.log  MDUpdate_Info.log                    SchemaManipulator_Warning.log
DistributionService_SchedulerDetails.log      MessagingService_StatisticsInfo.log  searchengine.log
DistributionService_Summary.log               MetadataObjectTelemetry.log          ServerControl.log
DSSErrors.log                                 MetadataServer_Info.log              SingleSignOn_Info.log
```

4. Check for the log file with name (**LicMgr.log**). It is available only after the license key is updated.
5. Open the **LicMgr.log** file and check whether the newly upgraded License Key is displayed or not.



```
[root@mstr-01 mstr]# clear
[root@mstr-01 mstr]# cat LicMgr.log
******************************************************************
4/21/21 10:17:01 AM EDT Upgrade license
4/21/21 10:17:01 AM EDT The license key: [redacted]
```

Change Log

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

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
12-March-2021	Added a new topic MicroStrategy Server License Update Process
11-May-2021	Added License Update Verification Information
20-May-2021	Added additional steps to License Update Verification Information
01-September-2021	Added Internet connectivity info in prerequisites topic.