PureConnect®

2020 R2

Generated:

29-November-2021

Content last updated:

11-May-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/pureconnect.

Table of Contents

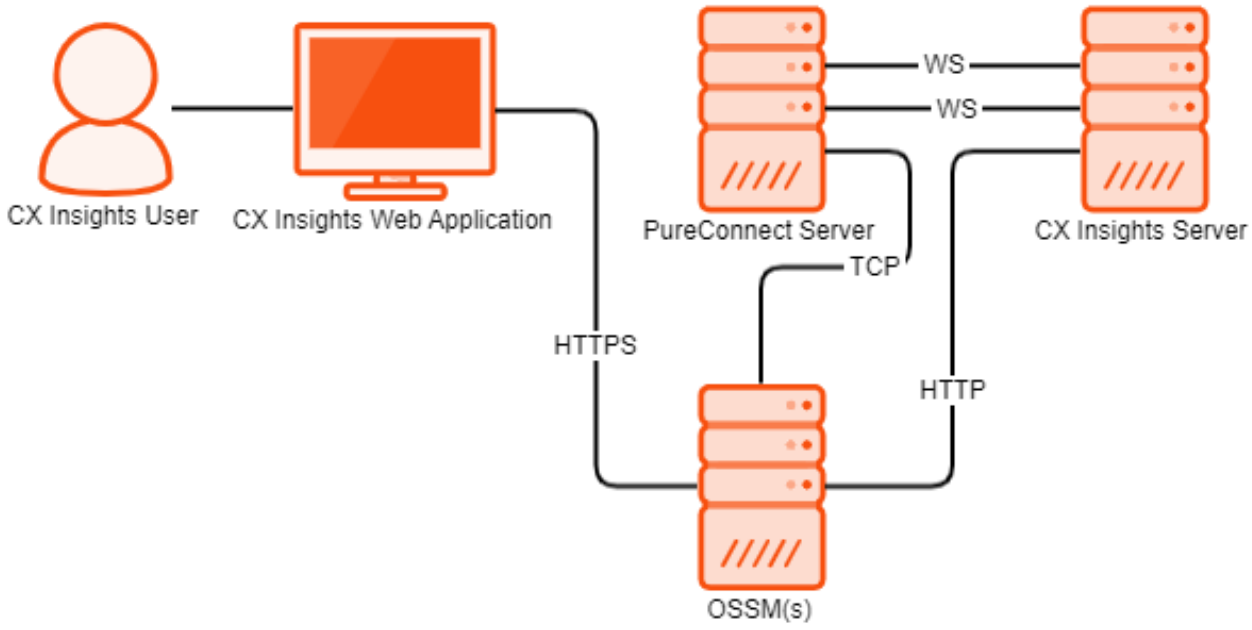
CX Insights overview

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 supervisor requires an assigned Analytics Core User license in order to log in, and 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 kubernetes through an Ansible playbook. CX Insights can be accessed on Google Chrome, Mozilla Firefox, Internet Explorer, and Safari.

CX Insights architecture

CX Insights deployment model



CX Insights server

The CX Insights server is a Linux server that uses Kubernetes 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 Kubernetes running.

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

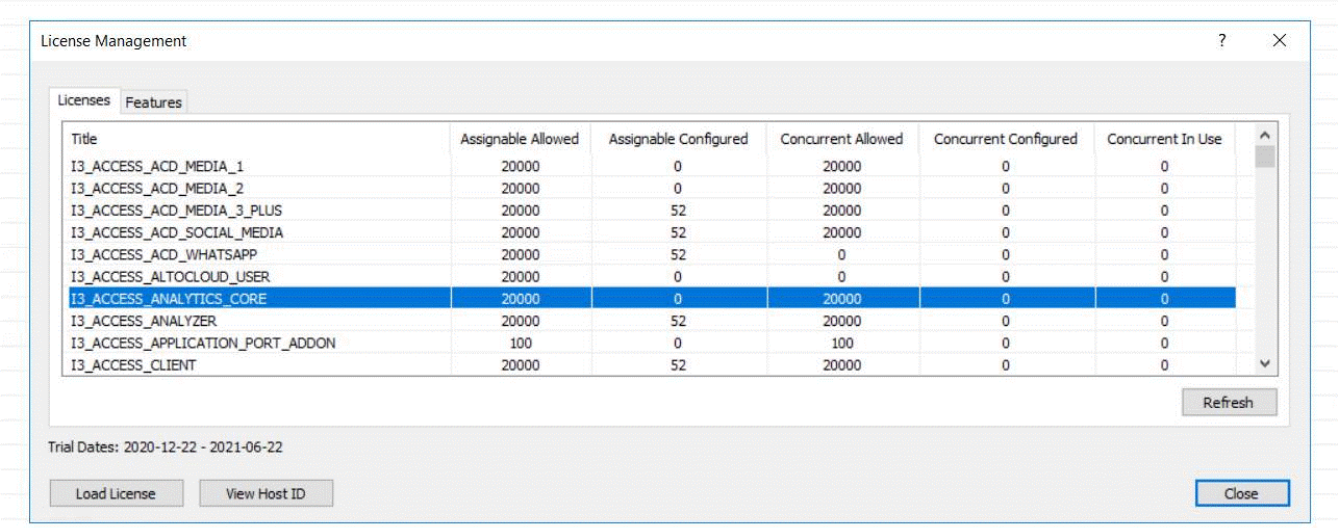
CX Insights licensing

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

Analytics access licenses

To confirm that you have the access licenses, go to the **License Management** form in Interaction Administrator and verify the following licenses on the **Licenses** tab.

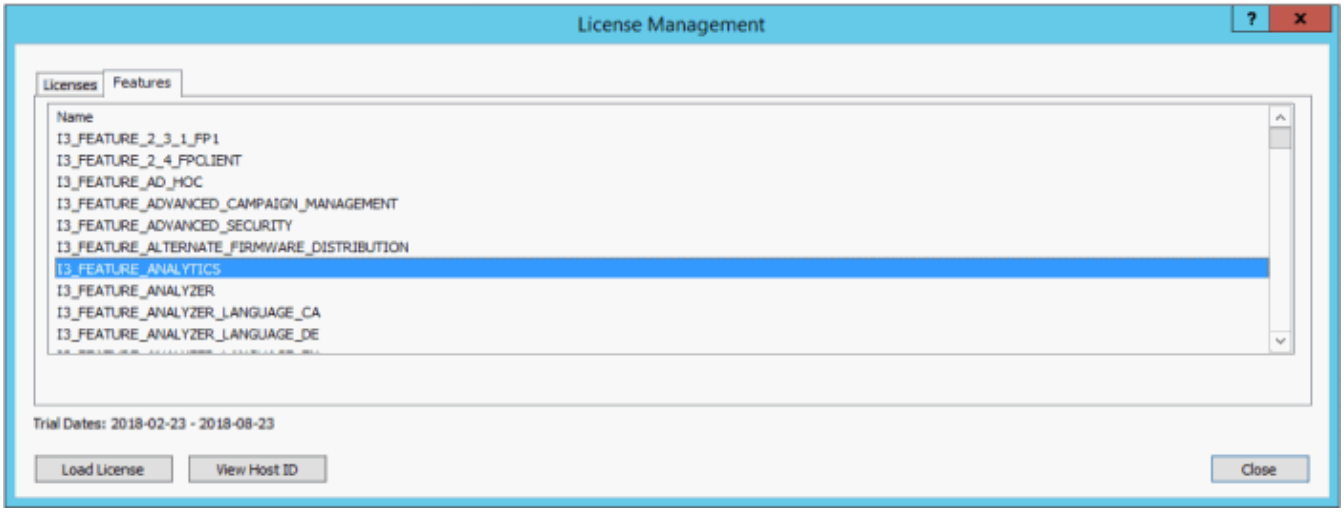
License	Description
I3_ACCESS_ANALYTICS_CORE	Basic dashboard license to view dashboards.



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

Analytics feature license

To confirm that you have the Analytics feature license, go to the **License Management** form in Interaction Administrator and on 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.

> Note: To set up a server and configure the following instructions a knowledgeable Linux administrator with Centos familiarity is required.

Prerequisite

- CIC version must be 2020R2 release.

> Important:
> - For versions older to 2020R2, CX Insights App does not work and also will not support kubernetes installation.
> - For version CIC 2020R1, require few settings after kubernetes installations.

Install CX Insights server

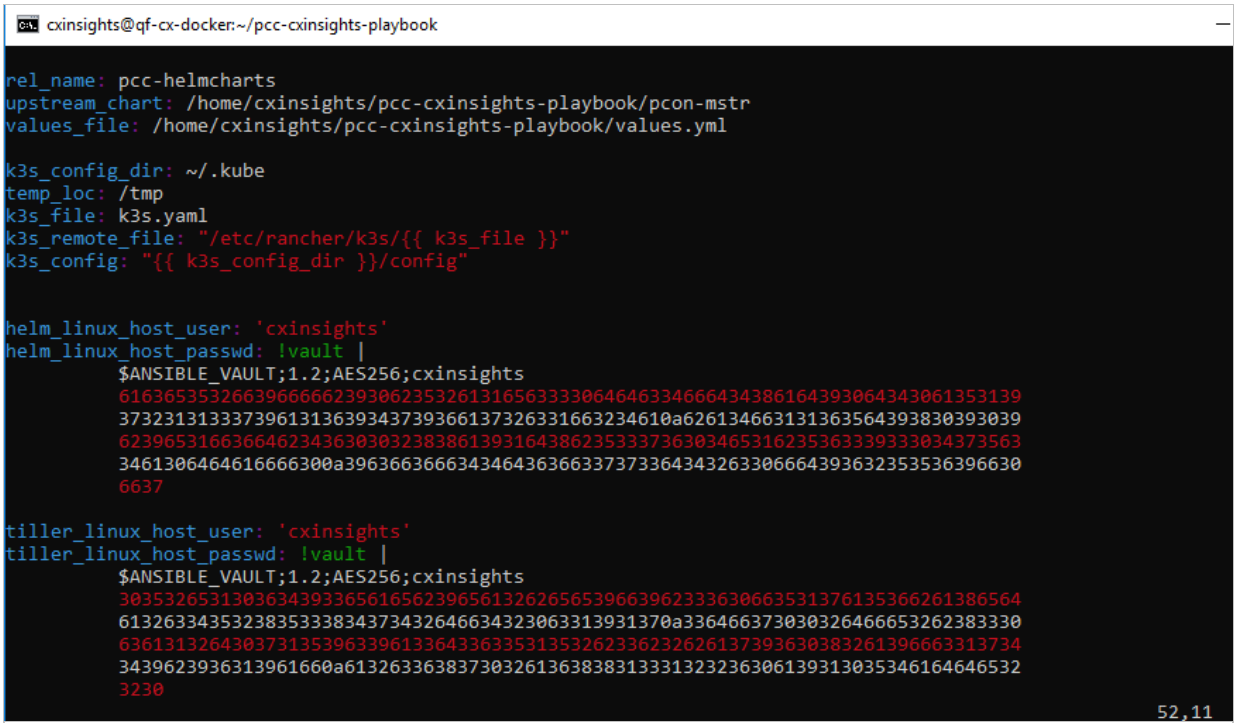
1. Install Centos7 on either a physical or virtual server that meets the minimum requirements for the production environment
   - 8+ vcpu
   - 32 GB RAM
   - 512 GB total storage space
   - 100+ GB root directory storage

   > Note: Make sure that the swap partition is at least 32 GB when installing Centos.
2. Download the CX Insights deployment containers from the following website:

   https://help.genesys.com/utilities-and-downloads.html
3. Extract the CX Insights artifacts archive that contains ansible_install, cxinsights-playbook-k3s.tgz, pcon-mstr.zip, and cx-insights.tgz.
4. Run the shell script ansible_install.sh to install the dependencies like the python and ansible packages using the root user account. The script also creates a CX Insights user account to perform all the ansible roles and tasks.

   > Notes:
   > - If the Centos already installed pip, then ensure that pip is version 8.1.2, which is compatible with python 2.7.5 version. Using a different version may cause the installation to fail.
   > - Verify whether ansible is installed by using the command `which ansible`. If ansible is installed, then the ansible version appears. If it is not installed, then re-run the ansible_install shell script.
   > - Verify whether the CX Insights account was created by using the command `cut -d: -f1 /etc/paswd`. Once created, log in to the CX Insights account.
   >   - su cxinsights
5. Complete the prerequisites for running ansible-playbook:

- Extract the cxinsights-playbook-k3s.zipfile to the CX Insights user home directory.
- Generate ansible vault for CX Insights user password, as it 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 *passwd* with CX Insights user account password. For vault usage, enter the password and make a note of it, so that you can enter the same password while running anisble-playbook command
  - `Ansible-vault encrypt_string 'passwd' --name 'tiller_linux_host_passwd' --vault-id cxinsights@prompt`, generate the password again only 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 the group_vars/all.yml file as shown below


cxinsights@qf-cx-docker:~/pcc-cxinsights-playbook
rel_name: pcc-helmcharts
upstream_chart: /home/cxinsights/pcc-cxinsights-playbook/pcon-mstr
values_file: /home/cxinsights/pcc-cxinsights-playbook/values.yml
k3s_config_dir: ~/.kube
temp_loc: /tmp
k3s_file: k3s.yaml
k3s_remote_file: "/etc/rancher/k3s/{{ k3s_file }}"
k3s_config: "{{ k3s_config_dir }}/config"
helm_linux_host_user: 'cxinsights'
helm_linux_host_passwd: !vault |
$ANSIBLE_VAULT;1.2;AES256;cxinsights
6163653532663966666239306235326131656333306464633466643438616439306434306135313937323131333739613136393437393661373263316632346104626134663131363564393830393039623965316636664623436303032383861393164386235333736303465316235363339333034373563346130646461666630a39636636663434643636633737336434326330666439363235353639663066376637
tiller_linux_host_user: 'cxinsights'
tiller_linux_host_passwd: !vault |
$ANSIBLE_VAULT;1.2;AES256;cxinsights
3035326531303634393365616562396561326265653966396233363066353137613536626138656461326334353238353338343734326466343230633139313704a3364663730303264666532623833306136131326430373135396339613364336335313532623623262613739363038326139666331373434396239363139613660a6132633638373032613638383133313232363061393130353461646465323230"
52,11

- Ensure host has proper hostname(FDQN), change the host details in *values.yml* file to the Linux host for the CX Insights install, and then ensure that the time zone details are mapped with the region where `gcxi` server is to be installed.

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

- Substitute the appropriate values in the *inventory.yml* file in the cxinsights-playbook-k3s directory. For example, assume ansible and K3S are running on the same machine. If the controller is different from the target machine, then `helm_linux_host` should be the controller host FQDN and `tiller-linux-host` should the FQDN of the 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 }}'
```

- After substituting the appropriate values, the local images can be loaded into the K3S registry if there is no access to the

root artifactory from which the images need to be pulled. Complete the following configuration steps in an ansible role:

- Navigate to the `../cxinsights-playbook-k3s/roles/load-images/vars` path. Right-click and open *main.yml* in edit mode.
- Change load_image *False* to load_image: *True*.
- Change `tar_file: 'cxinsights.tgz'` to `tar_file : <Name of the docker images .tgz file ( a big zip file, sizing around 12+ GB)>`

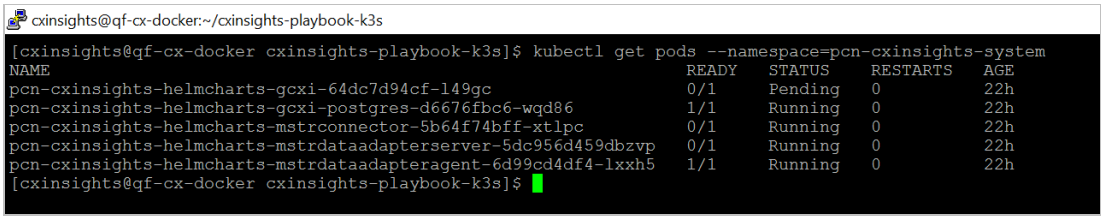
Note: Make sure the large *.zip* file is placed under the `../cxinsights-playbook-k3s` directory only.

6. Run the ansible playbook to start the services on the CX Insights server. The first time may take longer as dependencies are installed.

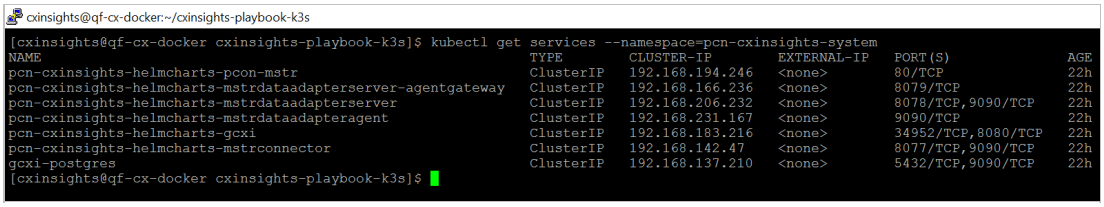
- sudo ansible-playbook –vault-id cxinsights@prompt -i inventory.yml site.yml -K

> Note: Make sure you enter the CX Insights password when the BECOME password is requested.

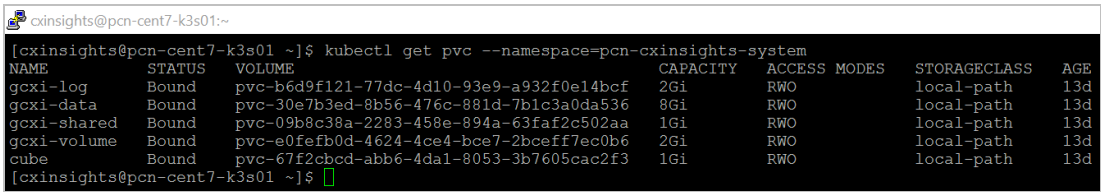
- Run the below mentioned commands to ensure that everything is up and running:
  - To verify that all the containers are up and running in all `namespaces`, use the command `kubectl get pods -A`
  - To verify that all the containers are up and running only in `pcn-cxinsights-system namespace`, use the command `kubectl get pods --namespace=pcn-cxinsights-system`



  - To verify that all the services are running in all `namespaces`, use the command `kubectl get services -A`
  - To verify that all the services are running only in the `pcn-cxinsights-system namespace`, use the command `kubectl get services --namespace=pcn-cxinsights-system`



  - To verify all ofl the persistent volumes in all `namespaces`, use the command `kubectl get pvc -A`
  - To verify the persistent volumes only in `pcn-cxinsights-system namespace`, use the command `kubectl get pvc --namespace=pcn-cxinsights-system`



> Note: If any of the above-mentioned commands fail to show the list, then run `helm delete --purge pcn-cxinsights-helmcharts --tiller-namespace pcn-tiller-system` command to delete the deployment and then run the ansible-playbook again.

Settings for CIC 2020R1 version

For CIC 2020R1 version, copy the below mentioned content in *traefik_custom.yaml* fileand then run the`kubectl apply -f traefik_custom.yaml` command.

```
apiVersion: helm.cattle.io/v1
kind: HelmChart
metadata:
name: traefik
namespace: kube-system
spec:
chart: https://%{KUBERNETES_API}%/static/charts/traefik-1.77.1.tgz
set:
rbac.enabled: "true"
ssl.enabled: "true"
metrics.prometheus.enabled: "true"
kubernetes.ingressEndpoint.useDefaultPublishedService: "true"
gzip.enabled: "false"
```

Upgrade containers

To upgrade the containers, use this command `sudo ansible-playbook -i inventory.yml site_upgrade.yml -K`. Ensure that proper tag names are updated in values.yml for the containers that need to be upgraded. If there is only one container that must be upgraded, then add a tag for the corresponding container and you can omit the rest of the properties.

```
gcxi:
  image:
    tag: 2.0
     tagcontrol: 2.0
gcxi-postgres:
  image:
    tag: 2.0
mstrconnector:
 image
   tag: 2.0
mstrdataadapteragent:
  image:
    tag: 2.0
mstrdataadapterserver:
  image:
    tag: 2.0
```

Rollback containers

To get the list of versions installed, use 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 rolled back 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 endpoints, and persistent volumes. This is equivalent to the `helm delete` command.

CX Insights monitoring and alerting

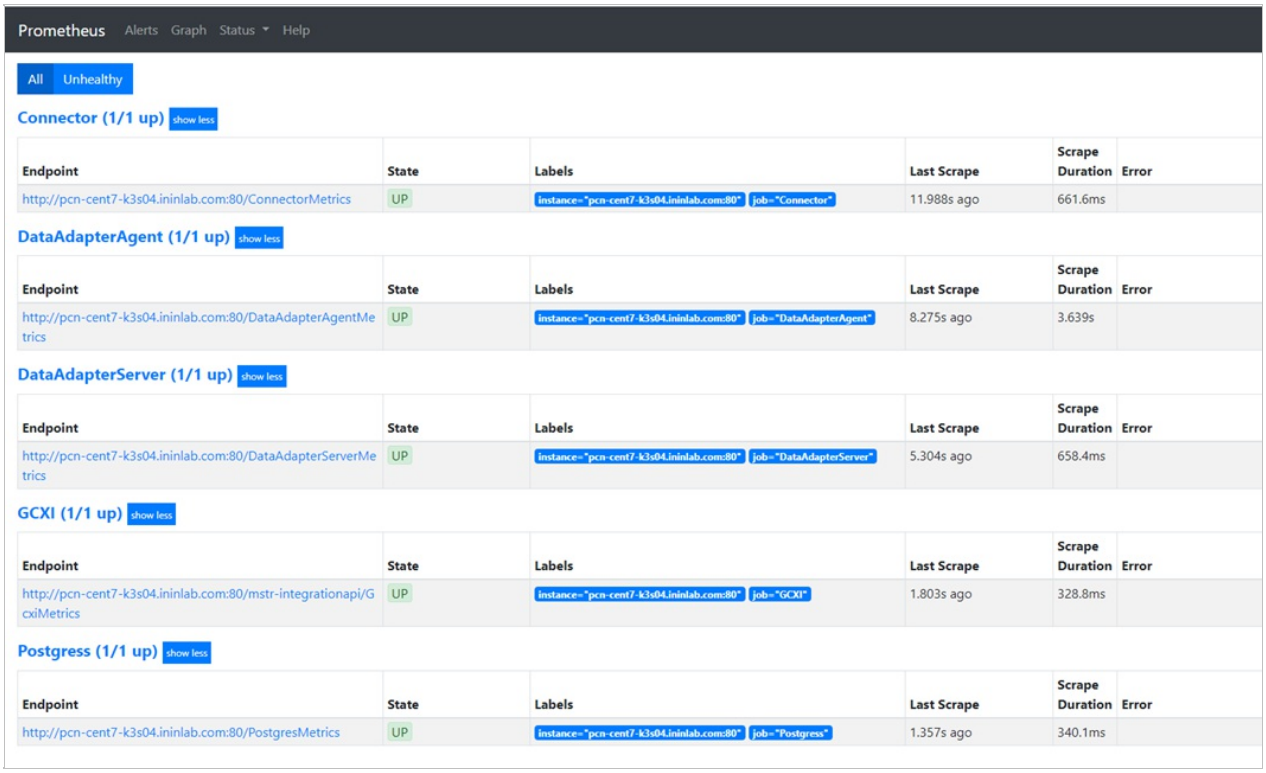
Prometheus is a third-party tool used to monitor health check of containers. It provides an alerting mechanism for critical scenarios.

Install Prometheus

1. Download Prometheus from https://prometheus.io/download/ and extract the files from the folder.
2. Copy alerts.yml to the Prometheus folder and update the prometheusus.yml *rule_files* property with*alerts.yml*.
3. Update Prometheus.yml with the below mentioned content and replace <SERVER> with Linux host (where all the containers are up and running). A reference to the alerts.yml file in the rules_files section contains all the alert scenarios. Scrape_interval is the interval in which data is pulled from all the services and evaluation_interval applies rules for the data.

```
# 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 the Prometheus executable file, verify that http://localhost:9090/rules is accessible and all rules are properly defined. Verify that warning and critical alerts are configured, and warning is of less priority, If there are any critical alerts raised, then file a ticket with proper logs.
5. The http://localhost:9090/targets shows the container state.



6. View alerts information in http://localhost:9090/alerts
7. To receive e-mail notifications/pagerduty, configure alertmanager. Refer to https://prometheus.io/docs/alerting/alertmanager/ to configure alertmanager. For download details, refer to https://prometheus.io/download/.
8. After downloading, configure the 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.

```
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"
```

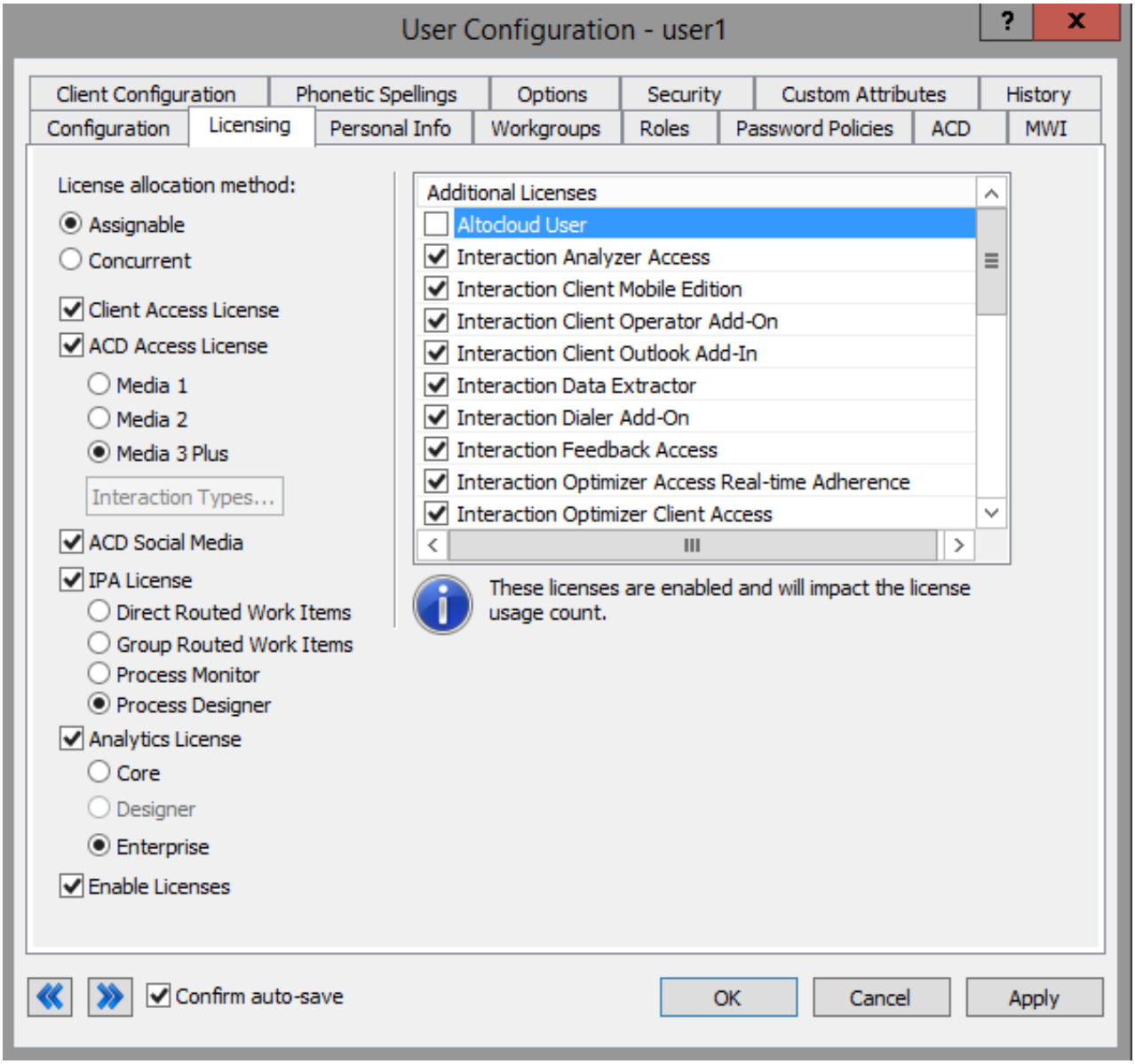
CX Insights server configuration

CX Insights server configuration

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

Allocate Access licenses

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



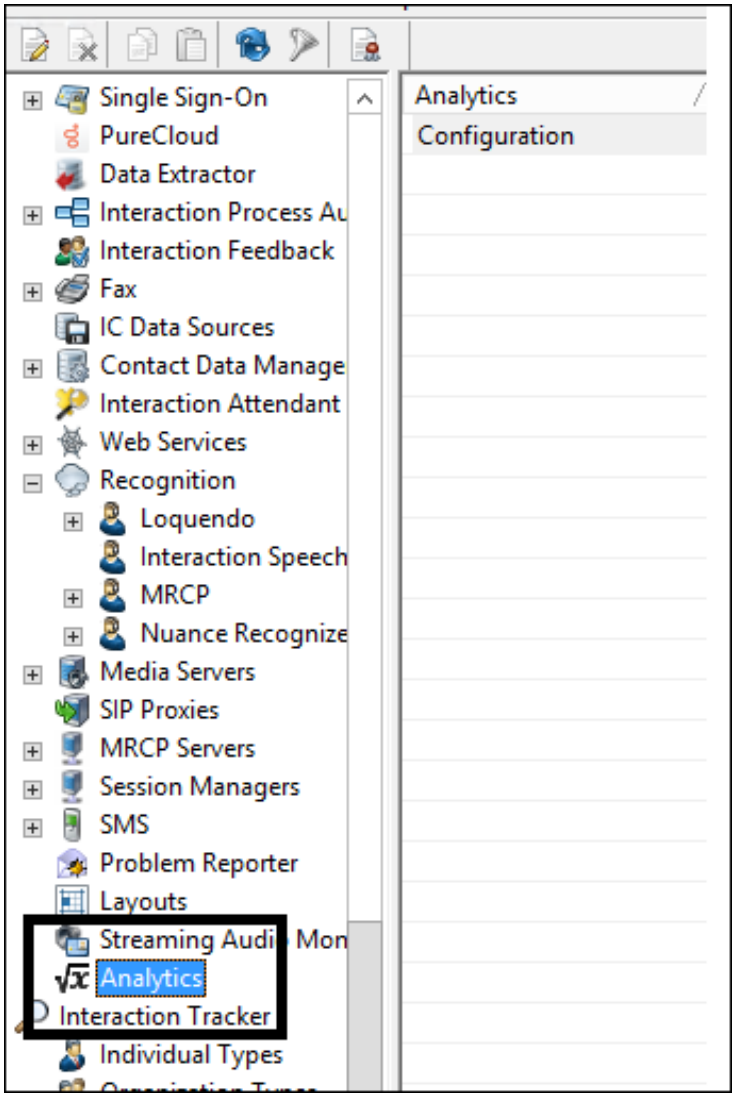
To assign an Analytics license to a user, select the **Analytics License** check box and select one of the following licenses.

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

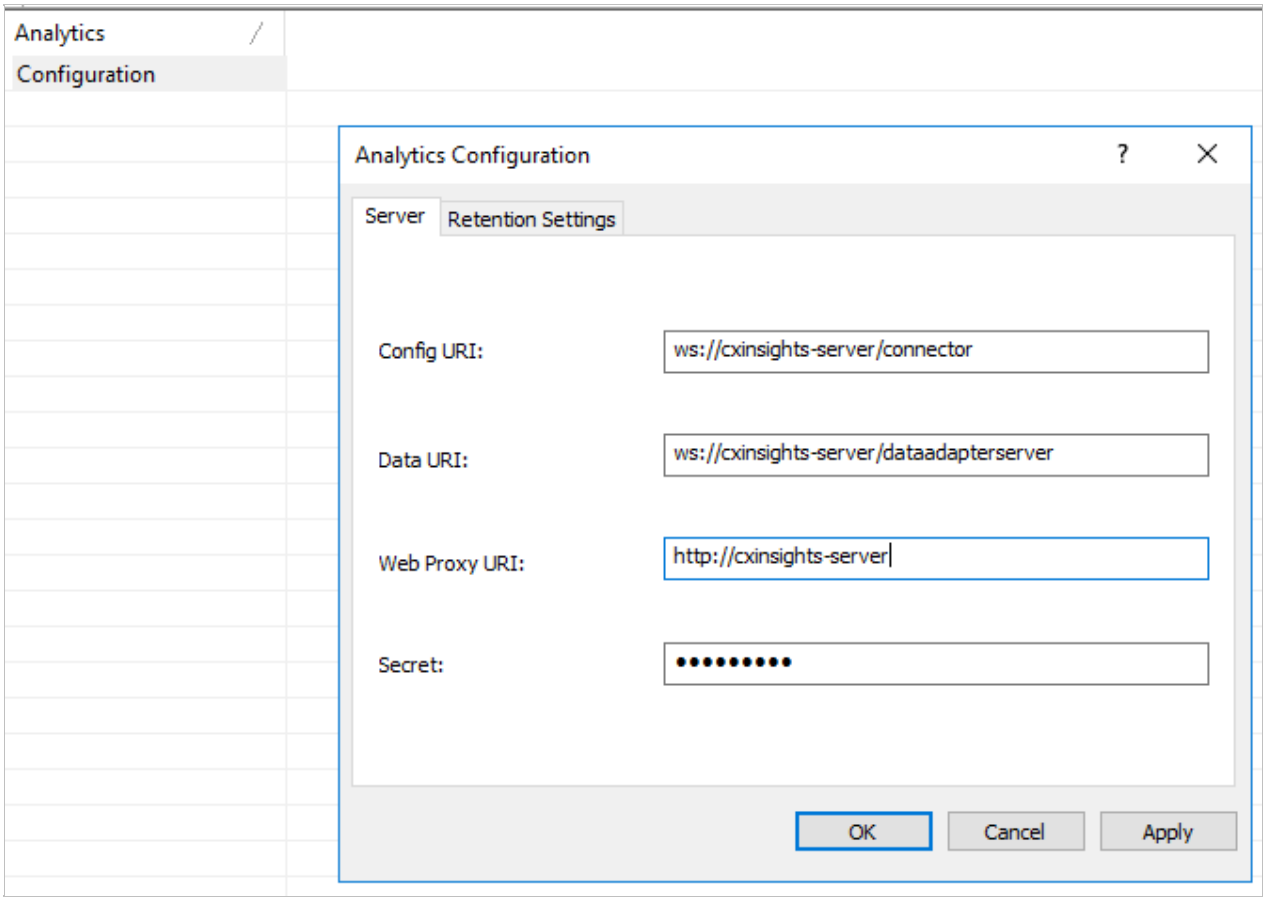
Configure CX Insights server in Interaction Administrator

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

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 **values.yml** file when deploying the CX Insights Server.

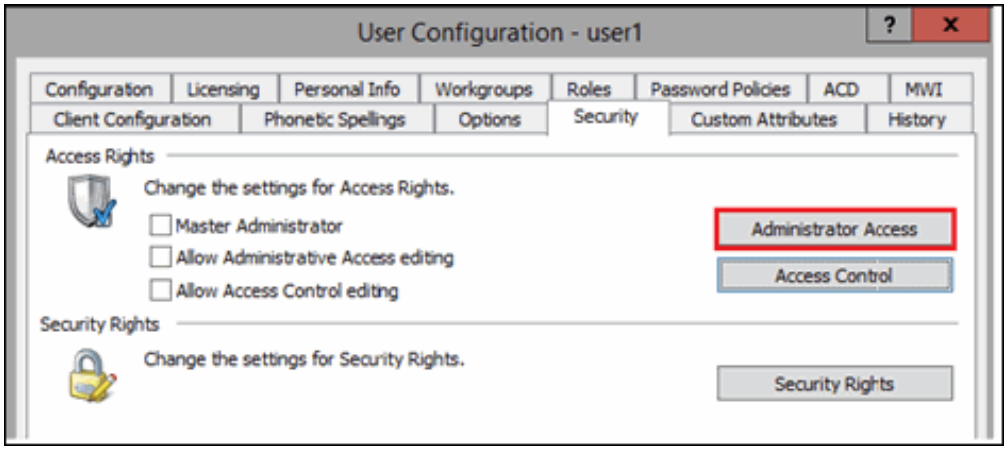
Once the configuration is complete, the AnalyticsBridge subsystem attempts to make the connections with the configured websocket. If the connection is successful, then the synchronization process begins. If there is a large number of users and workgroups to transfer, then it may take a few minutes to complete the process. Any additional changes to users, roles, workgroups, access controls, or memberships will trigger additional synchronization cycles. Once the servers are synchronized, the AnalyticsBridge subsystem begins streaming real-time statistics over the data websocket. Then, the users should be able to view the real-time dashboards.

Configure Administrator Access for CX Insights

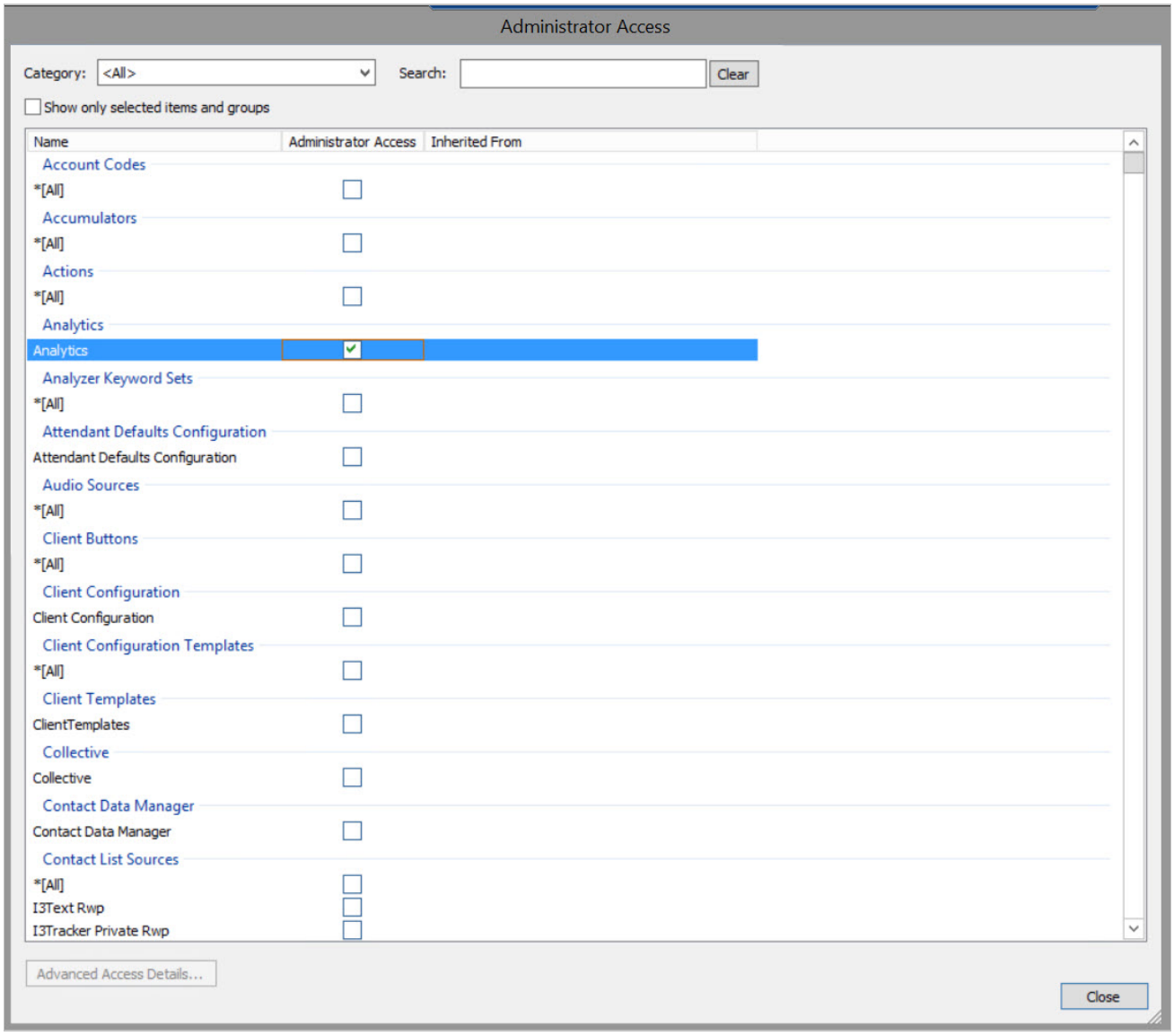
You can restrict which users, workgroups, or roles have access to configure the Analytics feature.

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.



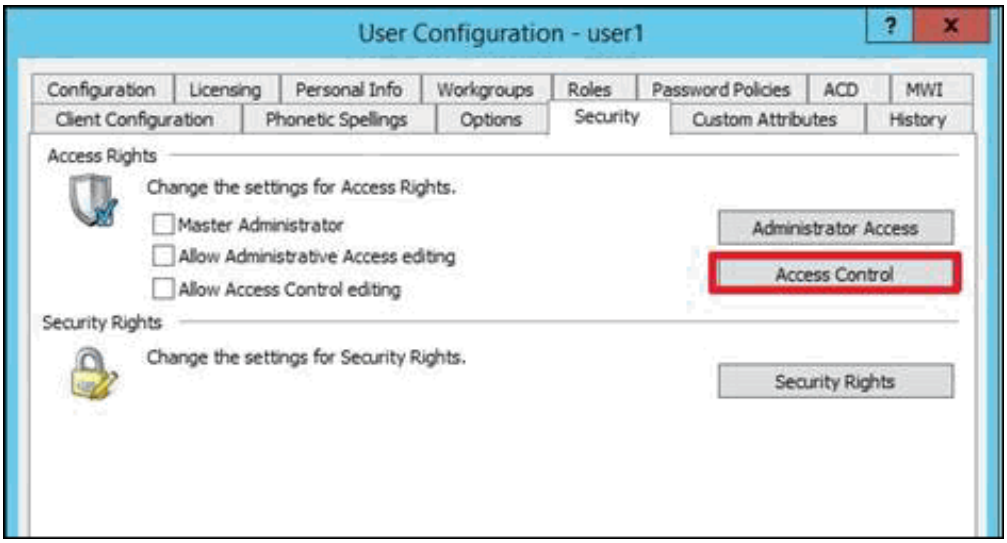
5. To give a user, workgroup, or role administrator rights to the Analytics feature, select the **Analytics** check box. You can clear the check box to remove the privilege.
6. Click **Close**.
7. To save the settings, click **OK** or **Apply**.

Configure Access Control for CX Insights dashboards

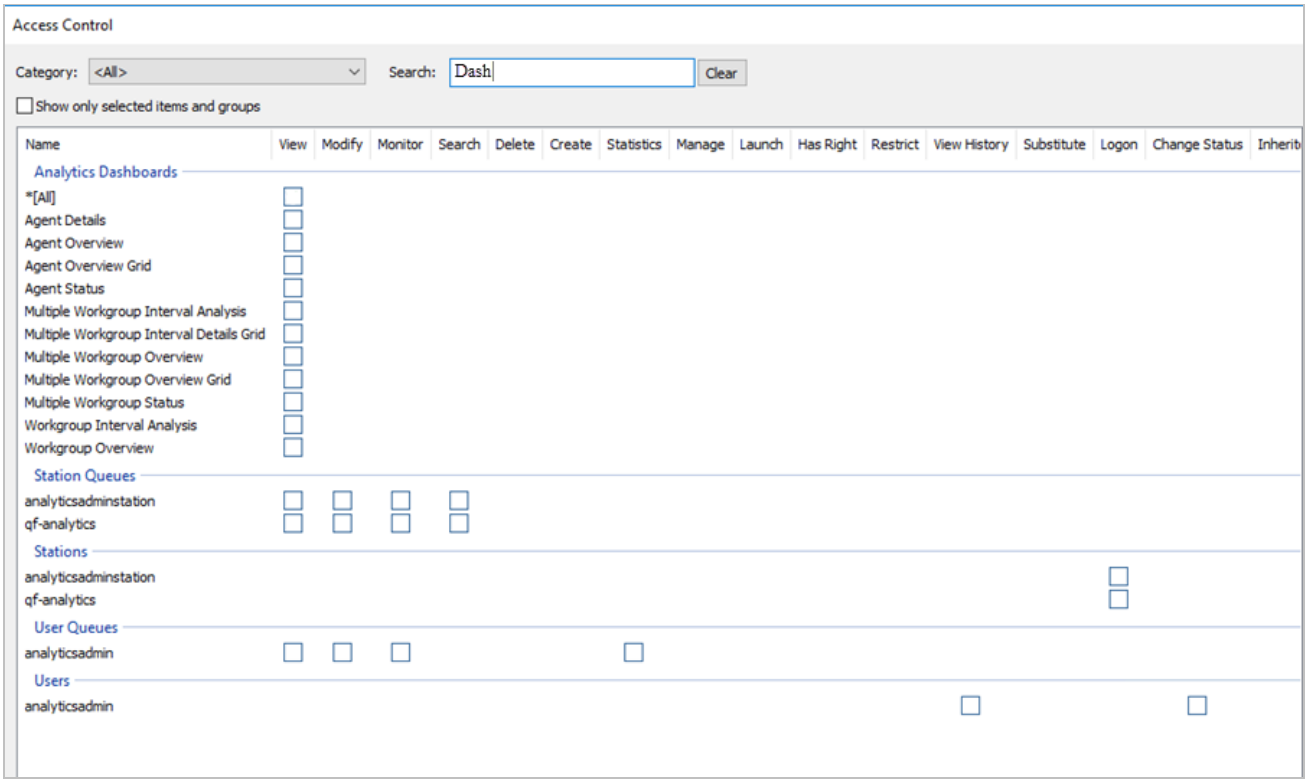
You can restrict which users, workgroups, or roles have access to specific dashboards.

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 the 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 PDF file. 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 is required.

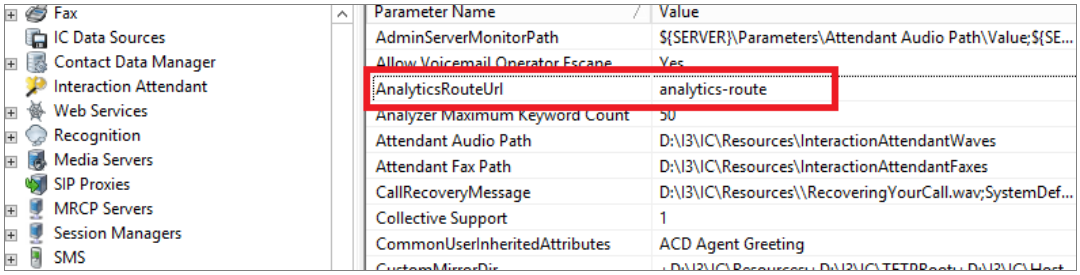
WebServer configuration

You can install and configure CX Insights on any 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:



Microsoft Internet Information Server

Install CX Insights web application for Microsoft IIS

For a basic working installation, such as for a test environment, follow these 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) was 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 the server as a proxy with response buffering enabled:
   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 the CIC web applications files (for analytics only)

1. In Windows Explorer, create a folder 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 folder.

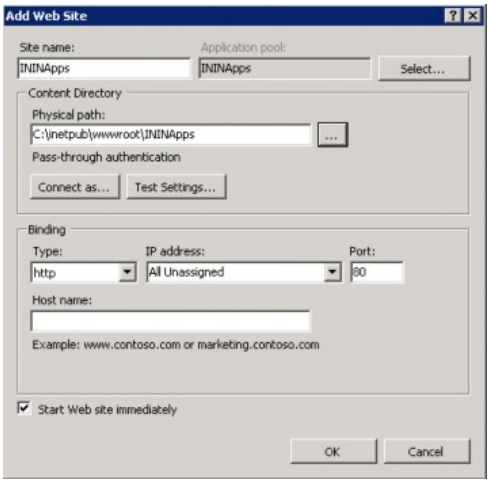
   > **Note:**
   > In this document, the directory is named **ININApps**.

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 the `analytics` folder in `web_files`.

6. Paste the folder you copied in the previous step into the directory you created in step 1. This 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 that 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 6 and 7 with the following information:

Original value	New value
off	http

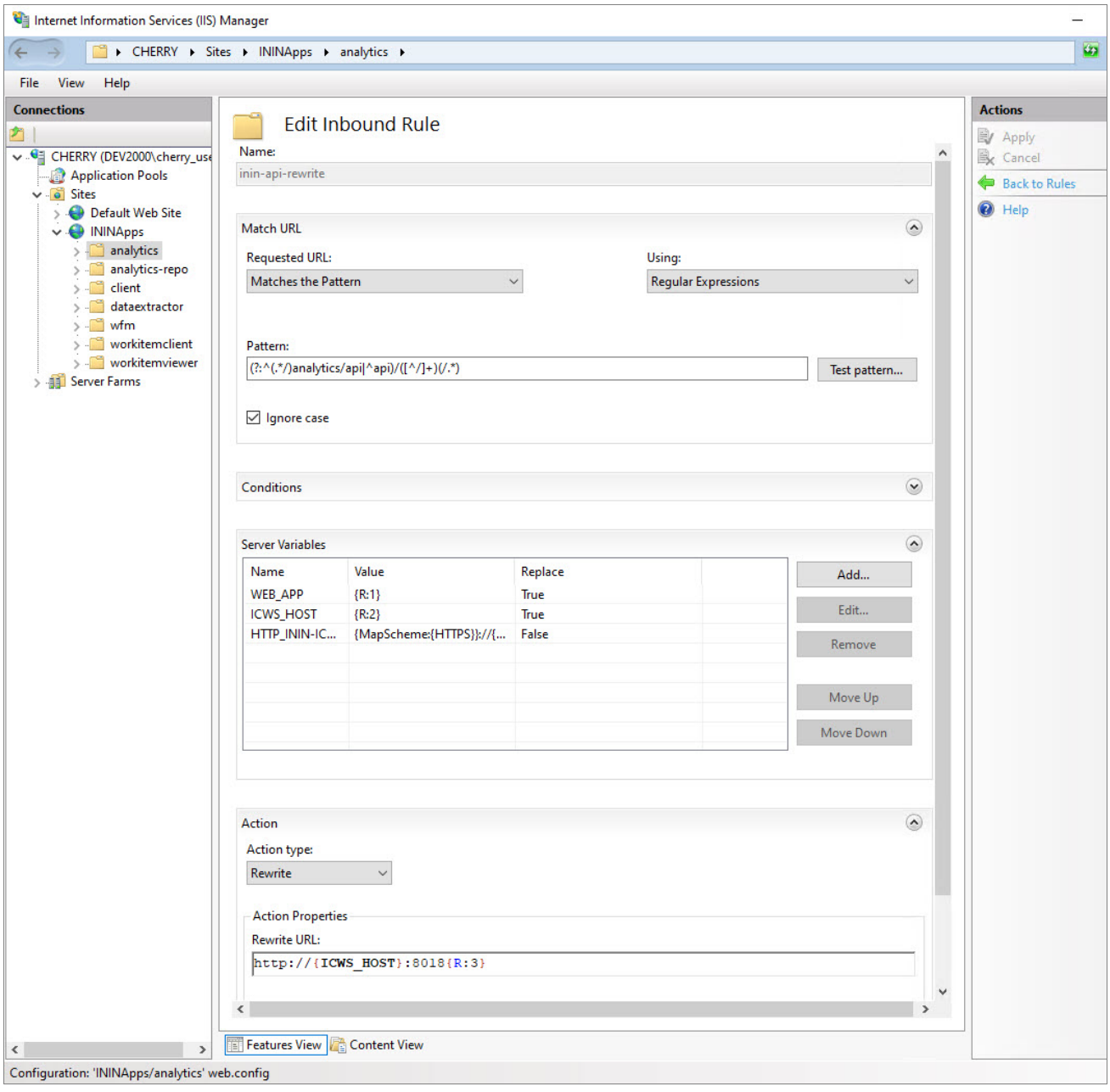
7. Create URL rewrite rules. You will create two inbound rules and four outbound rules.
   a. Click the site in the **IIS Manager**.
   b. Double-click the **URL Rewrite** module.
   c. Navigate to the **Actions** pane and select **Add Rule(s)**.
   d. For each rule, select **Blank rule** under the appropriate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

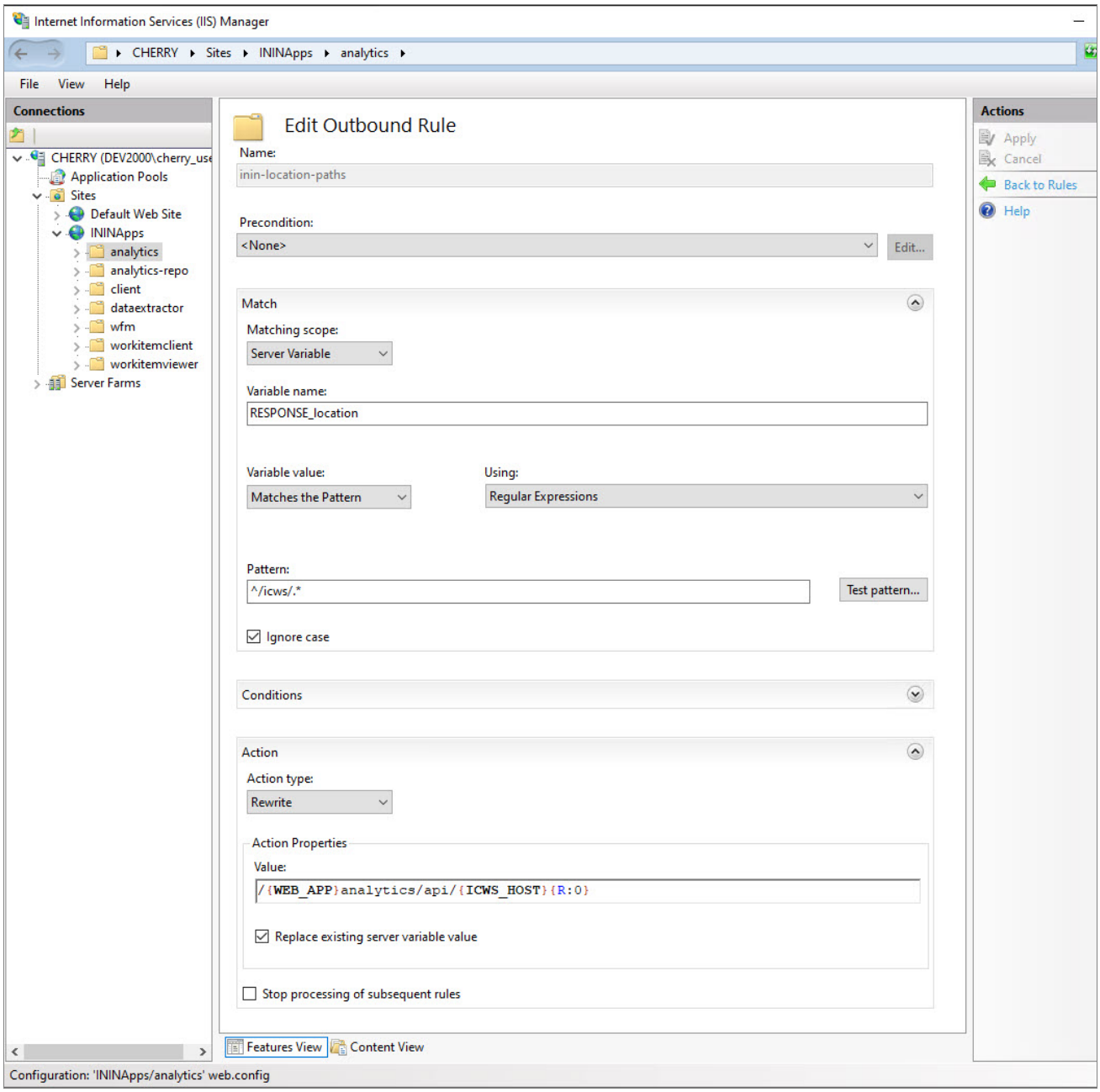
Name	Value	Replace
WEB_APP	{R:1}	True
ICWS_HOST	{R:2}	True
HTTP_ININ-ICWS-Original-URL	{MapScheme:{HTTPS}}://{HTTP_HOST}{UNENCODED_URL}	False



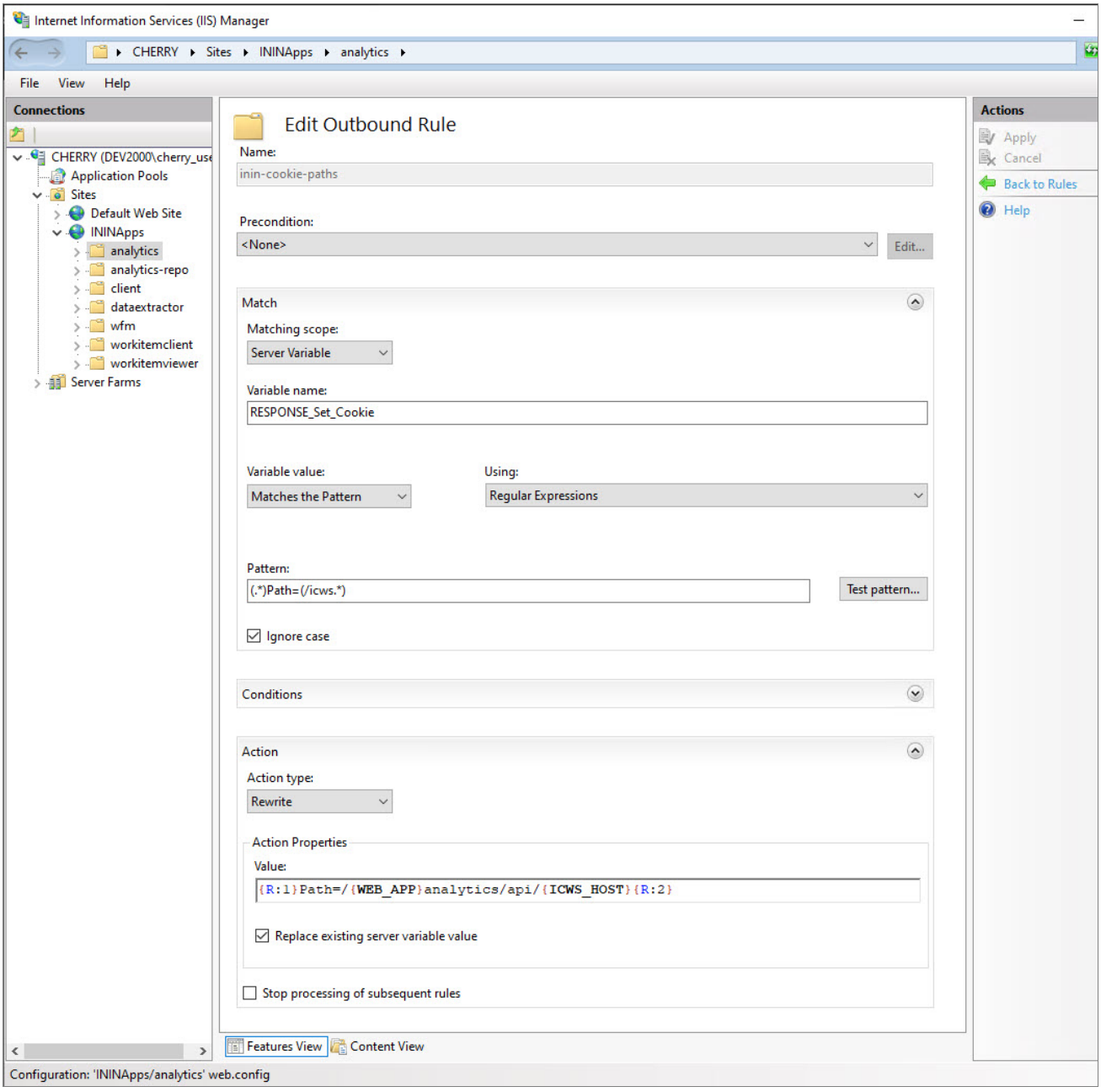
Inbound rule2 **This rule allows the client to reach the Session Manager host that Microstrategy calls is served from.**	
Name	analytics-route
Requested URL	Matches the Pattern
Using	Regular Expressions
Pattern	(?:^(.*/)analytics-route|^analytics- route)/([^/]+)(/.*)
Ignore case	Enabled
Server Variables	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

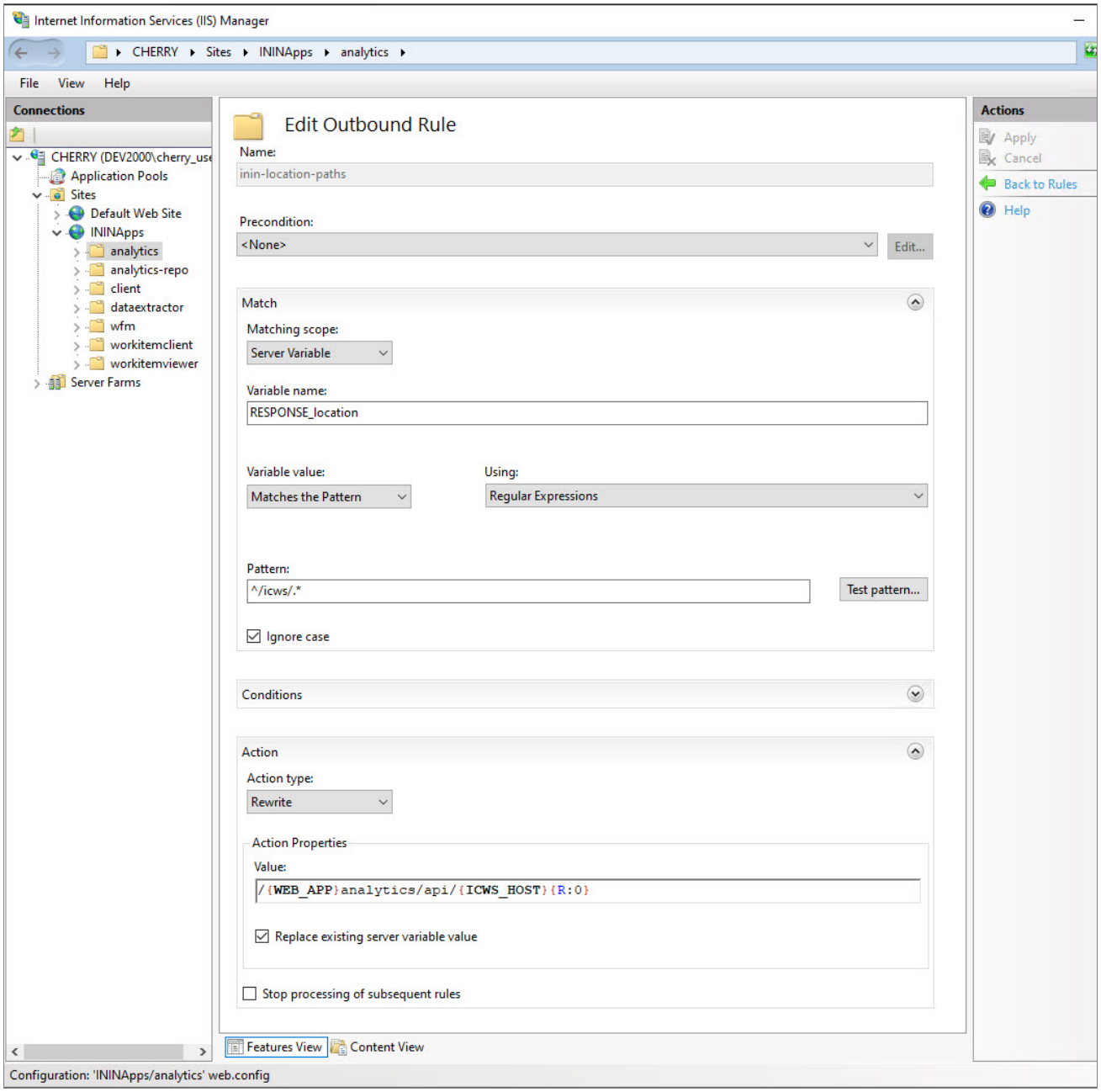
Name	Value	Replace
WEB_APP	{R:1}	True
ICWS_HOST	{R:2}	True
HTTP_ININ-ICWS-Original-URL	{MapScheme:{HTTPS}}://{HTTP_HOST}{UNENCODED_URL}	False



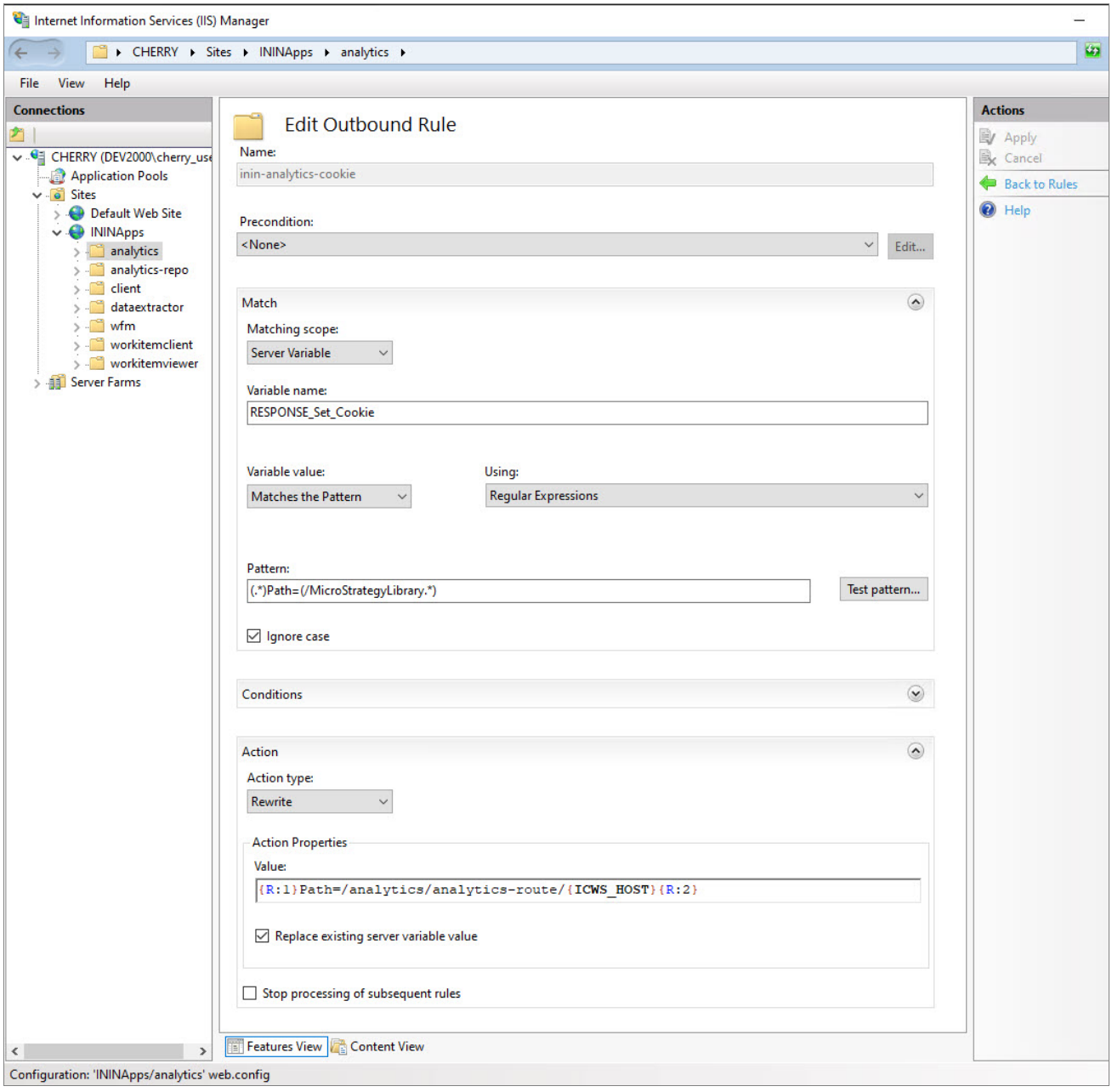
Outbound rule 1 This rule allows the cookies required by ICWS and the client to be located where the client needs them.	
Name	inin-cookie-paths
Precondition	<None>
Matching scope	Server Variable
Variable name	RESPONSE_Set_Cookie
Variable value	Matches the Pattern
Using	Regular Expressions
Pattern	(.*)Path=(/icws.*)
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



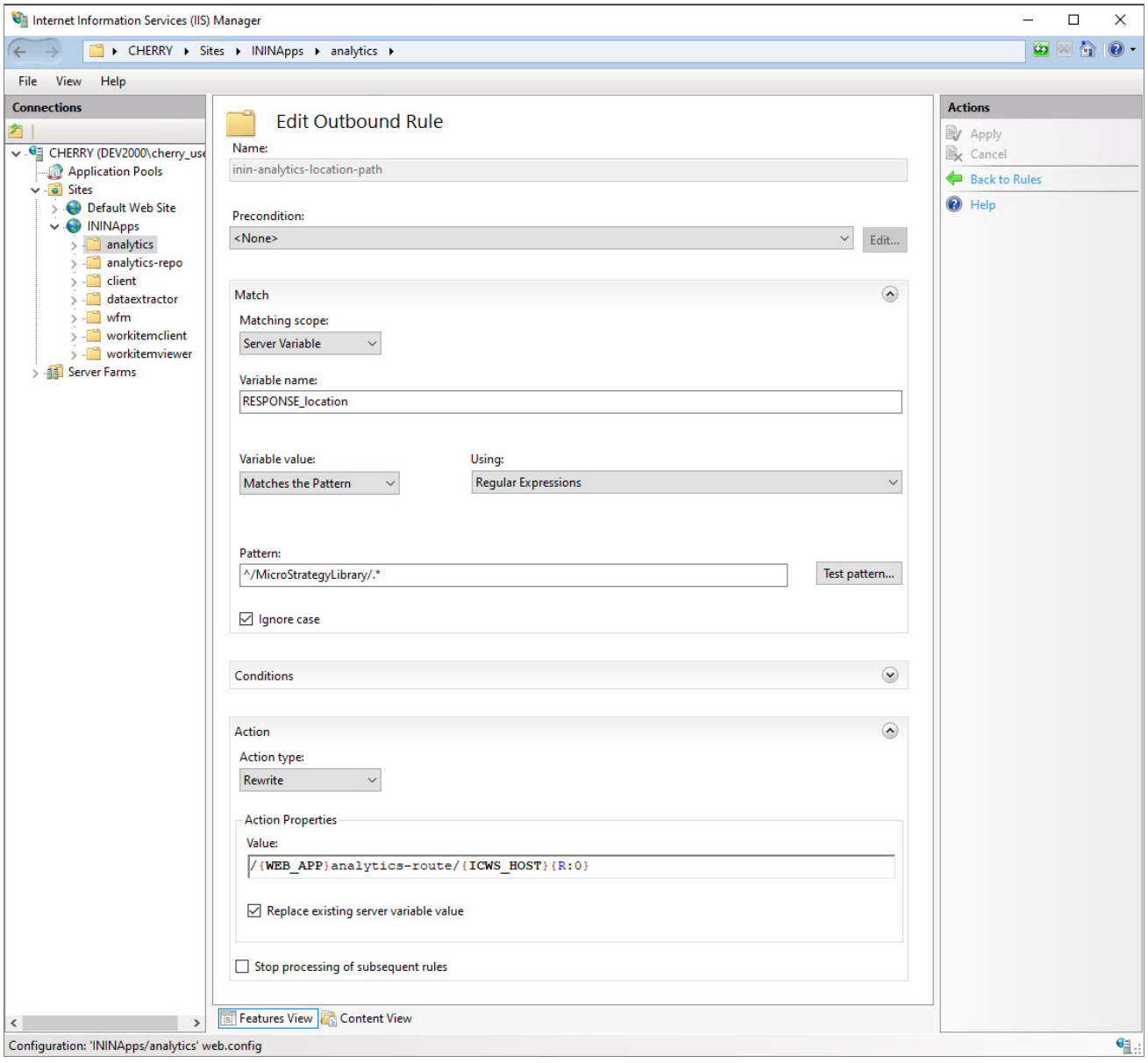
Outbound rule 2 This rule adjusts the location header	
Name	inin-location-paths
Precondition	<None>
Matching scope	Server Variable
Variable name	RESPONSE_location
Variable value	Matches the Pattern
Using	Regular Expressions
Pattern	^/icws/.*
Ignore case	Enabled
Action type	Rewrite
Value	/{WEB_APP}analytics/api/{ICWS_HOST}{R:0}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



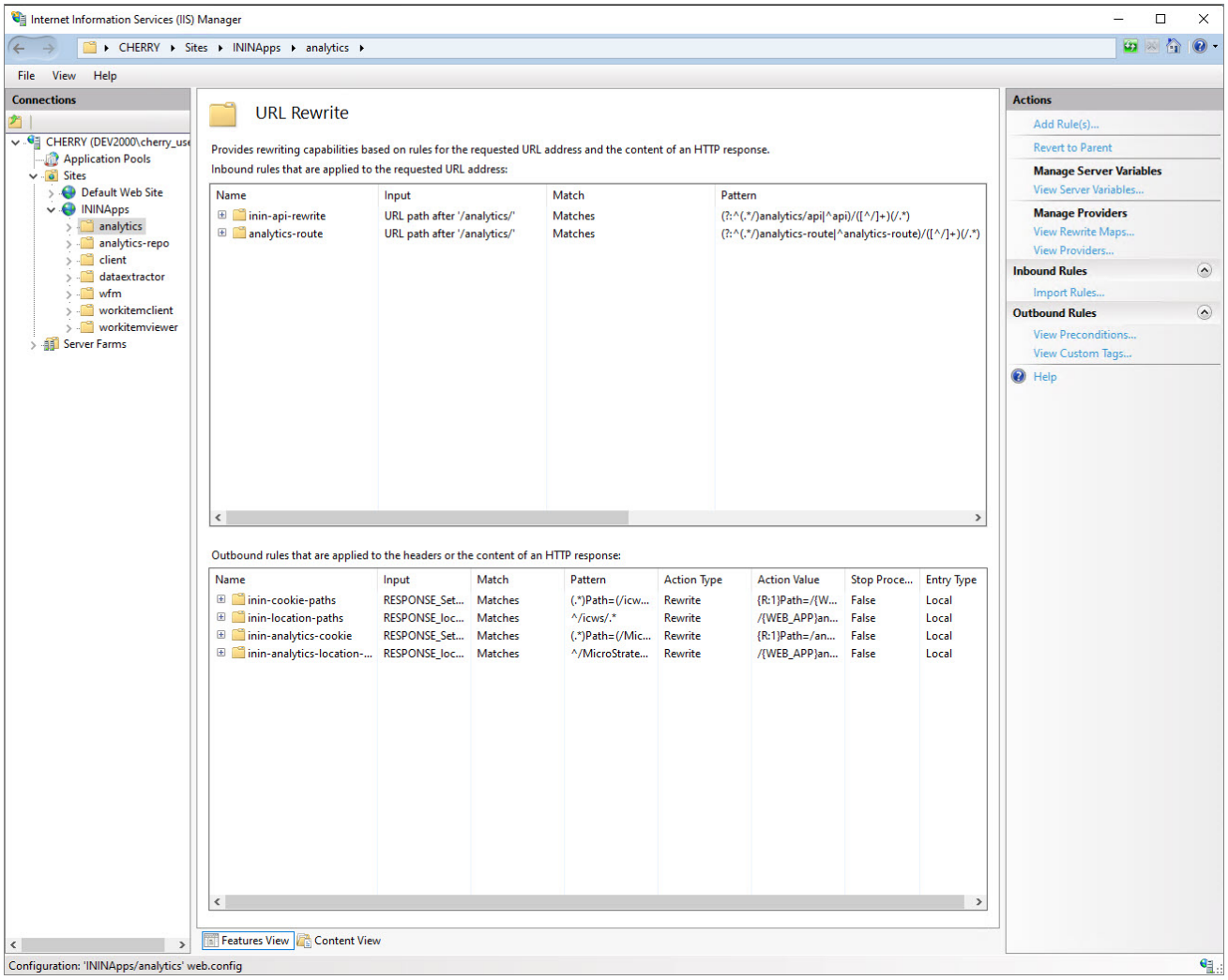
Outbound rule 3 **This rule allows the cookies required by MicroStrategyLibrary and the client to be located where the client needs them.**	
Name	inin-analytics-cookie
Precondition	<None>
Matching scope	Server Variable
Variable name	RESPONSE_Set_Cookie
Variable value	Matches the Pattern
Using	Regular 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. You must add a certificate to the web server, then bind the certificate to the HTTPS port, and then enable SSL on the site.

Step 1: Add a Certificate to the web server

You can use either a *self-signed certificate* or a *third-party certificate*. See the appropriate procedure below.

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`.

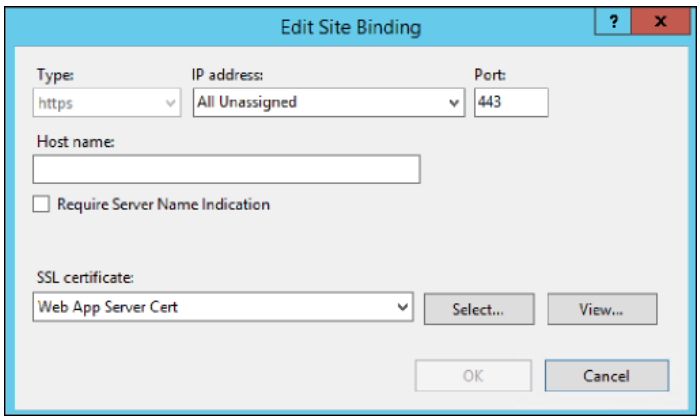
6. Click **Next**.
7. Choose the appropriate cryptographic service provider properties. Ask your third-party Certificate Authority (CA) which options to choose.
8. Click **Next**.
9. Enter a file name and location for the CSR.
10. Click **Finish**.
11. Send the generated CSR to your CA for signing.

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.

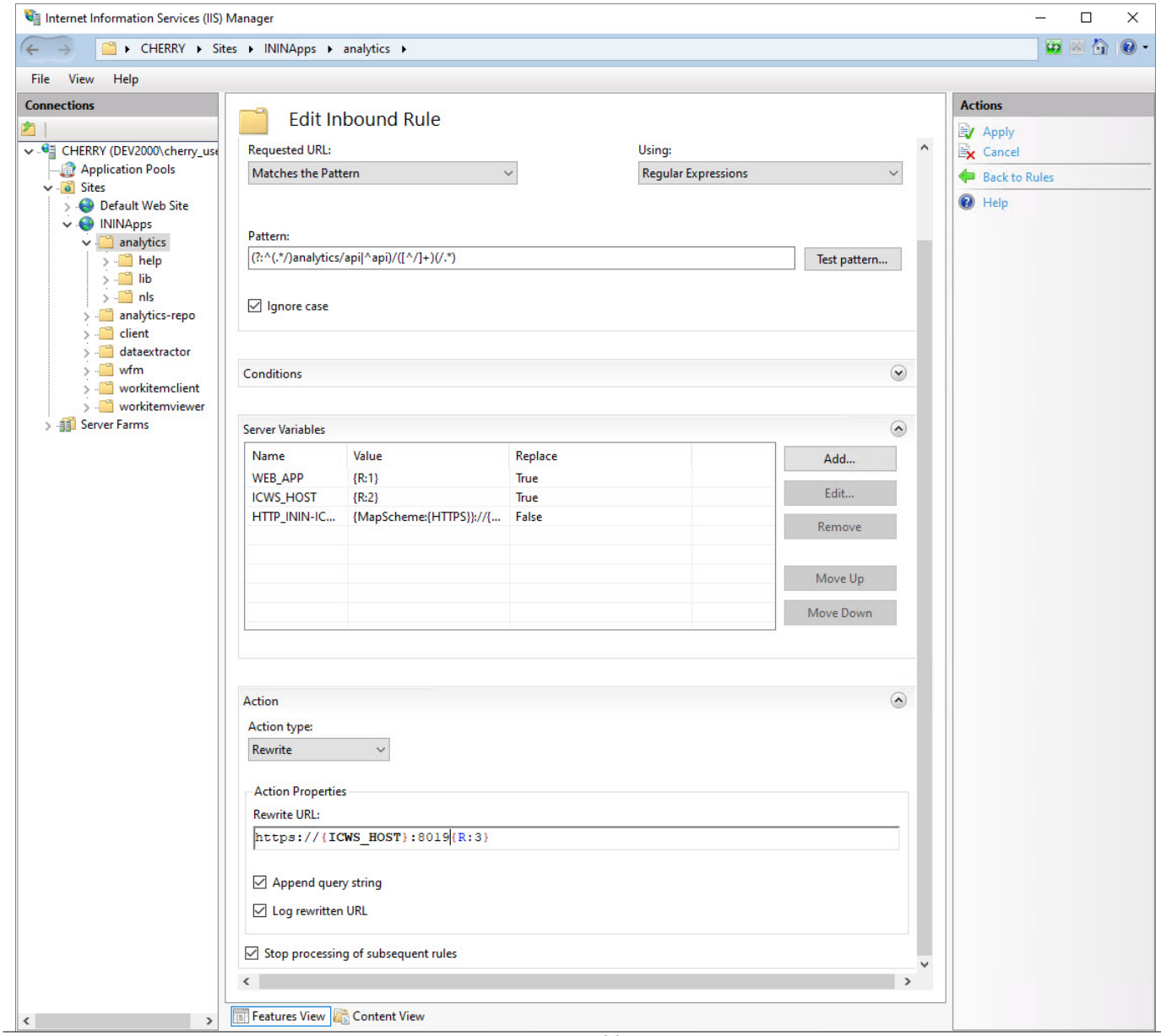
Enable HTTPS between IIS and CIC

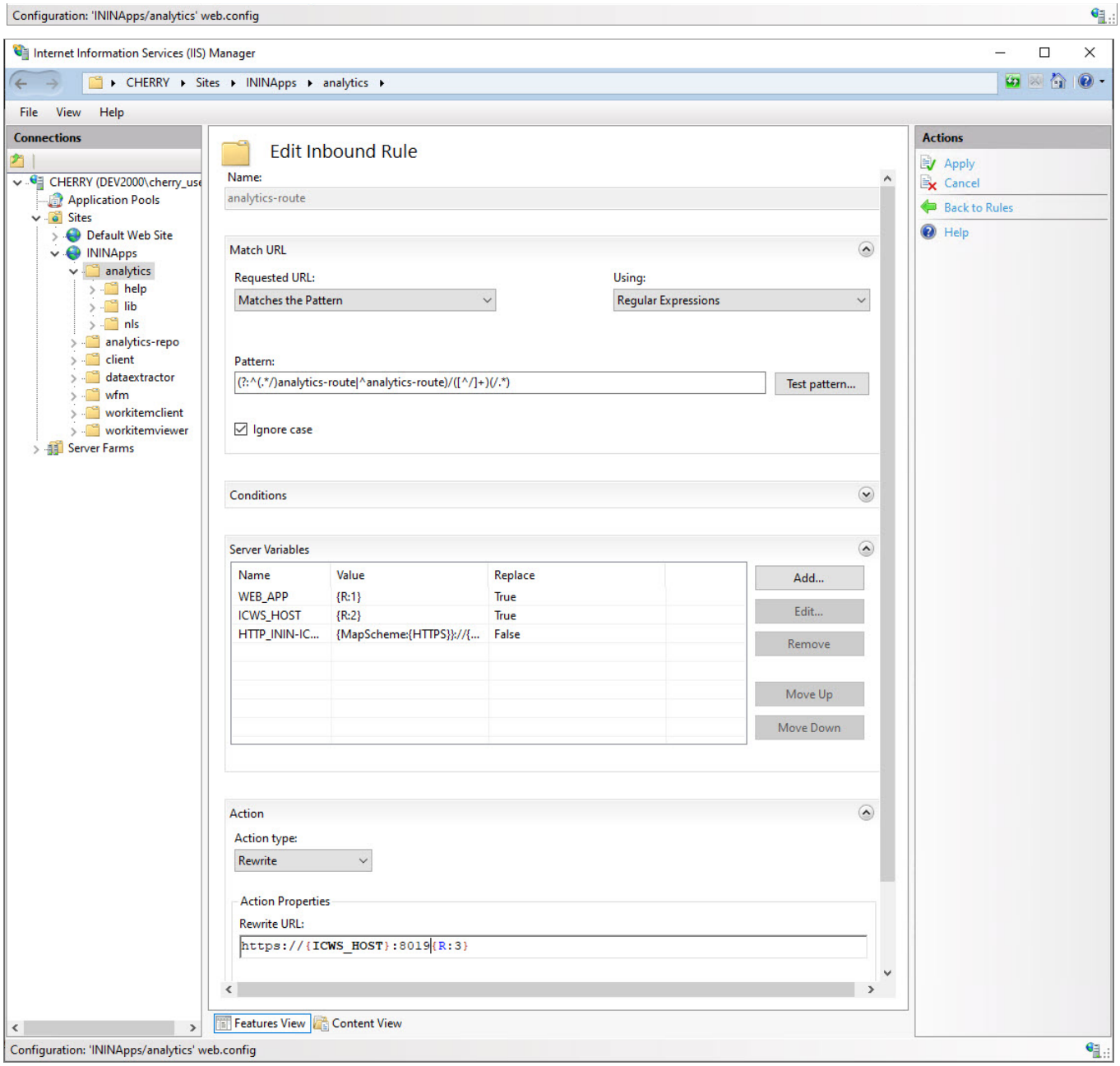
> **Tip:**
> The best practice is to use HTTPS from CIC to IIS and from IIS to the web browser, or from IIS to the web browser only. Securing traffic from IIS to CIC only can cause issues with Secure cookies.

These directions encrypt the connection between the web server and the CIC server. You must change the inbound rules to use HTTPS, and then trust the CIC server HTTPS certificate.

Step 1: Change inbound rules to use HTTPS

1. On your web server, open IIS Manager.
2. Expand **Sites**.
3. Select your website, that is.: ININApps.
4. Double-click the **URL Rewrite** module.
5. Open both inbound rules **inin-api-rewrite** and **analytics-route**.
6. In the **Rewrite URL** field, change the **Rewrite URL** to use **HTTPS** for the two inbound rules:
   - Change the protocol to **https**
   - 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 CIC Web Applications.
   Verify that your web server software has the appropriate permissions for the newly created folder.

   Note:
   In this document, the folder name is `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 the `Analytics` folder inside of `web_files`.
6. Paste the copied `Analytics` folder into the directory you created in step 1. Doing so places the appropriate directory structure and files for the `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 creates 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 listen.
   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

Part 1: To configure HTTPS, you need an SSL certificate, which you can generate using OpenSSL.

1. Download the OpenSSL Windows installer (`Win64OpenSSL-1_1_0k.exe`) from https://slproweb.com/products/Win32OpenSSL.html.
   You can use a more recent version, if available.
2. Create a directory where the SSL certificate will be generated (example: `C:\certs`).
3. Open a **Command Prompt** window in Administrator mode and navigate to the directory where the SSL certificate will be generated.
4. 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)
5. 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`
6. In the **Command Prompt** window, enter the following command:
   `"C:\OpenSSL-Win32\bin\openssl.exe" x509 -req -days 365 -in CSR.csr -signkey Private.Key -out server.crt`
7. Verify that the directory contains the following files:
   - `CSR.csr`
   - `PrivateKey.key`
   - `server.crt`

Part 2: The rest of the configuration is similar to the **HTTPS configuration**. Just modify the following steps of the **HTTPS configuration:**

- At step 2 in the HTTPS configuration, 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 to use 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://help.genesys.com/pureconnect/secure/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 the Analytics folder in `web_files`.
6. Paste the `Analytics` folder copied in the previous step into the folder you created in step 1. This 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 configure HTTPS for Nginx, you need to use OpenSSL to generate an SSL certificate.
   a. Download the OpenSSL Windows installer (Win64OpenSSL-1_1_0k.exe) from this link https://slproweb.com/products/Win32OpenSSL.html. You can use a later version if available.
   b. Create a directory (for example: C:\certs), where the SSL certificate will be generated.
   c. Open the command prompt in administrative mode and navigate to the directory where the SSL certificate will be generated.
   d. Set these configuration variables:
      `Set RANDFILE=C:\<directory name> \.rnd` (for example: C:\certs\.rnd). Modify your location accordingly.
      `Set OPENSSL_CONF=C:\OpenSSL-Win32\bin\openssl.cfg` (# as per installation)
   e. Enter "C:\OpenSSL-Win32\bin\openssl.exe" req -out CSR.csr -new -newkey rsa:2048 -nodes -keyout PrivateKey.key at the command prompt.
   f. Enter "C:\OpenSSL-Win32\bin\openssl.exe" x509 -req -days 365 -in CSR.csr -signkey PrivateKey.key -out server.crt at the command prompt.
   g. The directory should contain CSR.csr, PrivateKey.key and server.crt.
2. The following configuration is similar to HTTPS 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;
    #    }
    #}
}
```

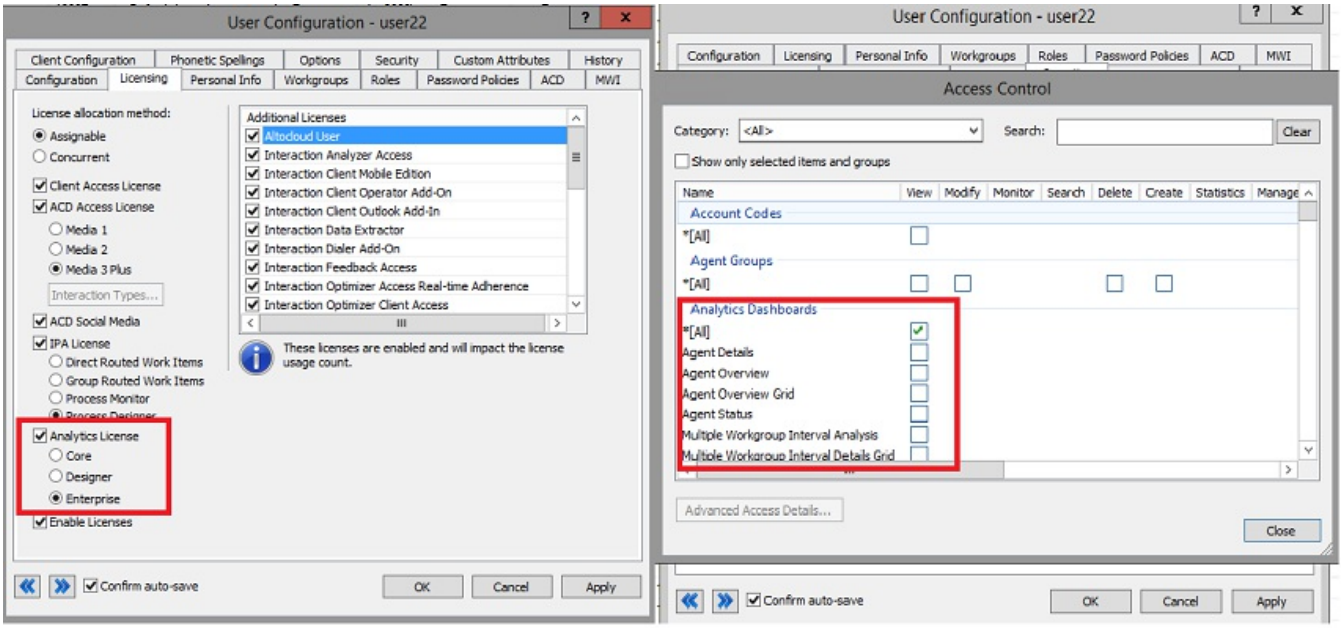
a. In the 'resolver' field , use the DNS server instead of instead of dl-hq1-dc01.ininlab.com.
b. Change port 8071 to custom port and provide 'SSL' binding beside the port number under server object.
c. In server object for 'server_name' field, use the proxy server name instead of eros.dev2000.com.
d. 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.

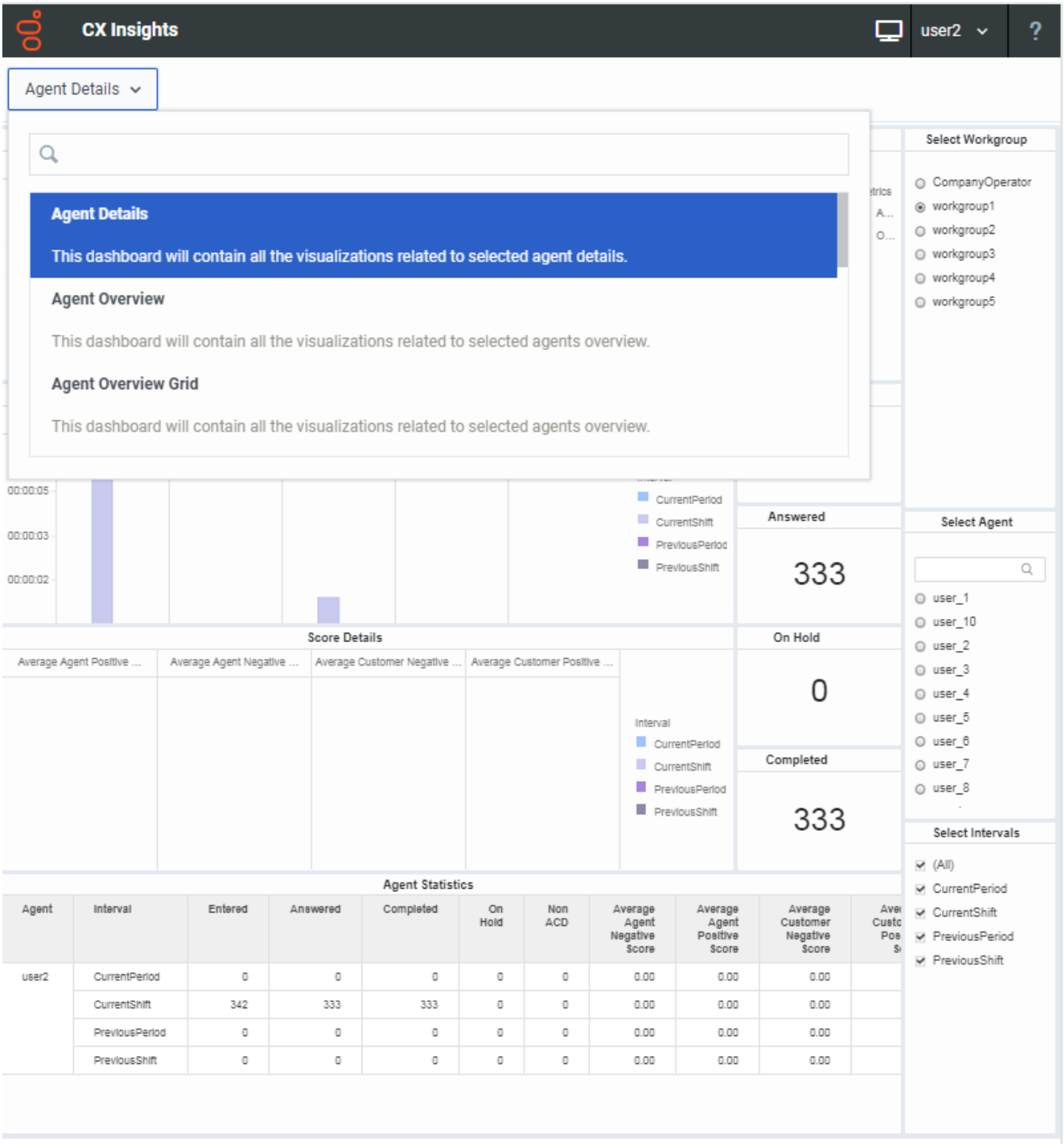
```
location ~ /client/ {
location ~ /client/help/ {
expires off;
}
location ~ /client/(?:addins|config)/ {
add_header Cache-Control "no-cache";
}
location ~ index.html?$ {
expires 15m;
}
location ~ .(?:js|css|jpe?g|ico|png|gif|svg|ttf|woff|otf|eot|mp3|wav|ogg)$
//eic/2019r2_systest/products/documentation/source/Technical_Reference_HTML/cic_web_applications_icg/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

View CX Insights dashboards

You can log in to the 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 on the CIC server. After successful loading, the dashboard refreshes every 30 seconds with real-time statistic values.


CX Insights
user2
Agent Details
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.
Select Workgroup
CompanyOperator
workgroup1
workgroup2
workgroup3
workgroup4
workgroup5
00:00:05
00:00:03
00:00:02
CurrentPeriod
CurrentShift
PreviousPeriod
PreviousShift
Answered
333
Select Agent
user_1
user_10
user_2
user_3
user_4
user_5
user_6
user_7
user_8
Score Details
Average Agent Positive ...
Average Agent Negative ...
Average Customer Negative ...
Average Customer Positive ...
Interval
CurrentPeriod
CurrentShift
PreviousPeriod
PreviousShift
On Hold
0
Completed
333
Select Intervals
(All)
CurrentPeriod
CurrentShift
PreviousPeriod
PreviousShift
Agent Statistics
Agent
Interval
Entered
Answered
Completed
On Hold
Non ACD
Average Agent Negative Score
Average Agent Positive Score
Average Customer Negative Score
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 CX Insights for Installation and Configuration Issues

Troubleshooting CX Insights installation and configuration issues requires administrator status (root permissions) and privileges, as well as access to the servers hosting CX Insights.

Error	Description	Solution
$'\r': command not found	While running the shell script, this error may occur because Windows uses '\r\n' as a new line character and Linux uses '\n'	To resolve this error, remove '\r' using the tr command. For example: `tr -d '\r' < ansible_install.sh > a.sh; mv a.sh ansible_install.sh ;`
Host FQDN error For example: `"Error: release pcc-helmcharts failed: Ingress.extensions \"pcc-helmchartsmstrdataadapterserver\" is invalid: sec.rules[0].host: Invalid value: \"172.26.20.55\": must be a DNS name, not an IP address"`	This error may occur when configuring and deploying CX Insights	To resolve this error, you must check for the host DNS. If the mentioned host is an IP address then change the host IP to host FQDN. For example: Instead of 123.45.67.890 IP address use pxx-kxx-cx.domainxxx.com (server.domain.com).
K3s server start error For example: `FAILED!=> {"changed": false, "msg": "Unable to restart service K3s: Failed to restart k3s.service: Connection timed out\nsee system logs and 'systemctl status k3s.service' for details.\n"}`	This error may occur when configuring and deploying CX Insights	To resolve this error, re-run the `sudo ansible-playbook --vault-id cxinsights@prompt -i inventory.yml ./site.yml -K` command.
Wrong pcon-mstr folder path error For example: `FAILED!=> {"changed": false, "cmd": ["helm", "install", "pcon-mstr", "--name", "pcc-hemcharts", "--namespace", "pcn-cxinsights-system", "--tiller-namespace", "pcn-tiller-system", "-f", "~/values.yml"], "delta": "0:0:00.166113", "end": "2020-02-21 06:47:47.533577", "failed_when_result": true, "msg": "non-sero return code", "rc": 1, "start": "2020-02-21 06:47:47.367464", "stderr": "Error: failed to download \"pcon-mstr\" (hint: running 'helm repo update' may help)", "stderr_lines": ["Error: failed to download \"pcon-mstr\" (hint: running 'helm repo update' may help)"], "stdout": "", "stdout_lines": []}`	This error may occur when configuring and deploying CX Insights	To resolve this error, check for the `pcon-mstr` folder path. It should be in `cxinsights-playbook-k3s/group_vars/all.yml upstream_chart` value path.
Pods evicted state error	This error may occur when configuring and deploying CX Insights	Sometimes many pods are in an evicted state. To remove all the evicted pods, use these commands. Prerequisites: yum install jq `kubectl get pods -A --all-namespaces -o json | jq '.items[] | select(.status.reason!=null) | select(.status.reason | contains("Evicted")) | "kubectl delete pod \ (.metadata.name) -n \ (.metadata.namespace)"' | xargs -n 1 bash -c`

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 a 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, 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 New 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

Prerequisites

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

Installing a new License Key

Edit the GCXI configmap using the command

kubectl edit configmap pcn-cxinsights-helmcharts-gcxi-config -n pcn-cxinsights-system .

Update the file with the below property with the license key under the data properties as shown below and save the file.

MSTR_LICENSE: <your new license>

```
# Please edit the object below. Lines beginning with a '#' will be ignored,
# and an empty file will abort the edit. If an error occurs while saving this file
# reopened with the relevant failures.
#
apiVersion: v1
data:
  CIC_BACKUP_SERVER_NAME: 10.145.0.252
  CIC_DB_HOST: qf-analyticsdb.qfun.com
  CIC_DB_LOGIN_ID: IC_ReadOnly
  CIC_DB_NAME: I3_IC_TITUS
  CIC_SERVER_NAME: 172.26.27.30
  CXINSIGHTS_VERSION: "3.0"
  ENABLE_SAML: "true"
  ENABLE_TLS: "true"
  GCXI_VERSION: 9.0.009.00
  GIM_DB: ""
  GIM_DB_TYPE: ""
  GIM_DB_TYPE_EX: ""
  GIM_HOST: ""
  GIM_LOGIN: ""
  GIM_PASSWORD: ""
  GIM_PORT: ""
  HOST_FQDN: pcn-cent7-k3s03.ininlab.com
  HOSTNAME: mstr-01
  LANGS: en-US,fr-FR,de-DE,ja-JP,pt-BR,es-ES,zh-CN,nl-NL,pl-PL
  LOG_LEVEL: INFO
  MAX_HTTP_CONNECTIONS: "16"
  MAX_POOL_SIZE: "200"
  MAX_USER_SESSIONS: "500"
  META_DB_ADMIN: ""
  META_DB_ADMINDB: ""
  META_DB_ADMINPWD: ""
  META_DB_HOST: ""
  META_DB_LOGIN: ""
  META_DB_PASSWORD: ""
  META_HIST_LOGIN: ""
  META_HIST_PASSWORD: ""
  MSTR_ADMIN_PASSWORD: Genesys_0
  MSTR_ADMIN_USER: Administrator
  MSTR_DATASET_CACHE_DIRECTORY: /var/opt/MicroStrategy/IntelligenceServer/Cube/mstr
  MSTR_DB_PORT: "1433"
  MSTR_DISABLE_REPORT_SERVER_CACHE: "true"
  MSTR_DSN_NAME: GCXI_CONNECT
  MSTR_ISERVER_TIMEZONE: America/Indiana/Indianapolis
  MSTR_LICENSE:
```

Delete the existing GCXI container using the below command.

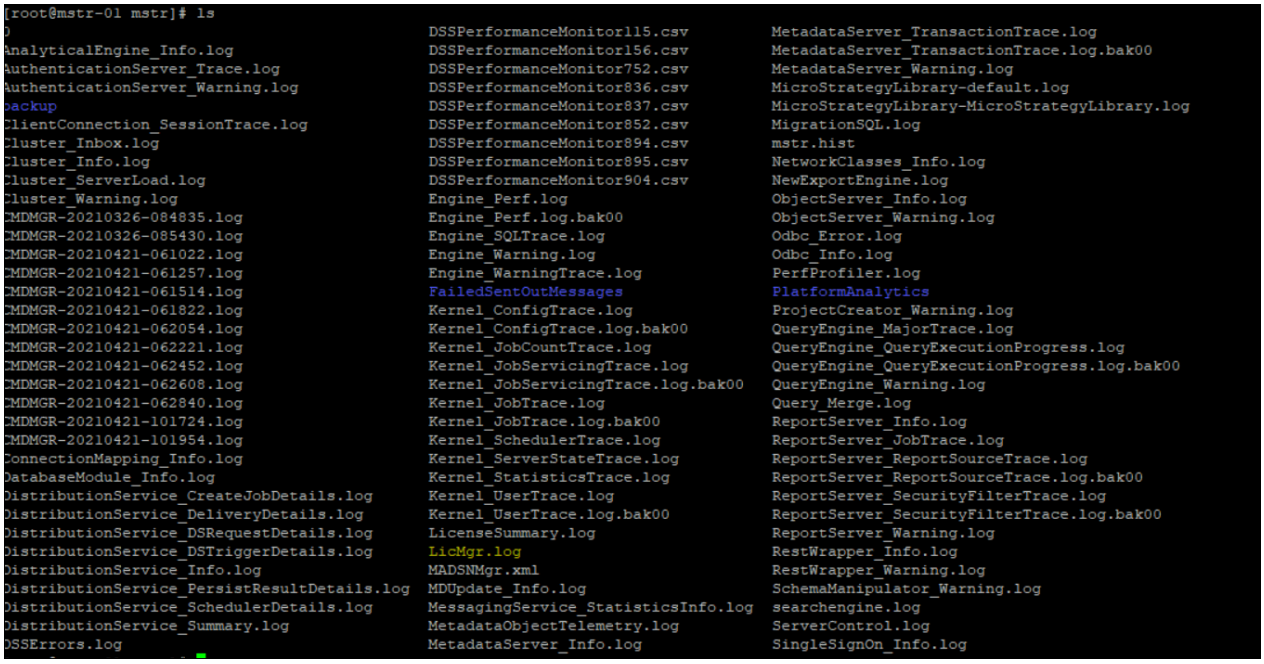
kubectl -n pcn-cxinsights-system scale --replicas=0 deployment/pcn-cxinsights-helmcharts-gcxi

Create new GCXI pod using the below commandand license key will be updated for newly created gcxi container. There is a down time of minimum 5-minutes for a new container to get up and running.

kubectl -n pcn-cxinsights-system scale --replicas=1 deployment/pcn-cxinsights-helmcharts-gcxi

License Update Verification

After the license update is done, a log file is generated. To check the log file existence do the following:
Go inside the GCXI pod and navigate to **/mnt/log/mstr** path.



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

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	Added Kubernetes Deployment Information
29-April-2020	Added Troubleshooting Information
04-May-2020	Updated Server Install and Upgrade Containers topics
11-June-2020	Updated Server Install and help.genesys.com links
21-July-2020	Updated CX Insights configuration in Interaction Administrator topic
20-January-2020	Removed Enterprise row and updated image of License Management
12-March -2021	Added a new topic MicroStrategy Server License Update Process
11-May-2021	Added License Update Verification information