



PureConnect®

2021 R2

Generated:

22-September-2021

Content last updated:

01-September-2021

See [Change Log](#) for summary of changes.



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>.

For copyright and trademark information, see https://help.genesys.com/pureconnect/desktop/copyright_and_trademark_information.htm.

Table of Contents

Table of Contents	2
What's New for CX Insights administrators?	3
CX Insights overview	4
CX Insights architecture	5
CX Insights deployment model	5
CX Insights server	5
CX Insights prerequisites	6
CX Insights requirements	6
Hardware	6
Software	6
CX Insights licensing	8
Analytics access licenses	8
Analytics feature license	8
CX Insights server installation	9
Prerequisite	9
Install CX Insights server	9
Install SSL certificate on CIC server	13
Upgrade containers	16
Roll back containers	16
Deleting deployment	16
CX Insights monitoring and alerting	16
Install Prometheus	16
Configure reverse proxy using nginx	18
Log file	19
Code to be copied to the nginx.conf file	19
Ports opened on CX Insights server	21
CX Insights server configuration	22
CX Insights server configuration	22
Allocate Analytics licenses	22
Configure CX Insights server in Interaction Administrator	23
Retention Settings	24
Configure Administrator Access for CX Insights	25
Configure Access Control for CX Insights dashboards	26
Test the CX Insights installation	28
Backup and restore configuration of CX Insights data	29
Backup CX Insights data	29
Configure CX Insights backup through Ansible	29
Configure CX Insights backup through script	29
Instant backup	30
Restore CX Insights data	30
Configure CX Insights data restore through Ansible	30
Configure CX Insights data restore through script	30
Backup log files	31
Troubleshooting CX Insights for Installation and Configuration Issues	32
Appendix	34
MicroStrategy Server License Update Process	34
License Ordering Process	34
License Request Checklist	34
Process of Updating new License Key	34
License Update Verification	36
Change Log	37

What's New for CX Insights administrators?

For more information about the changes and enhancements in CX Insights for administrators, see the following:

2020 R4

PureConnect 2020 R4 introduced the following changes and enhancements in CX Insights for administrators.

Red Hat Enterprise Linux (RHEL) support

CX Insights now supports RHEL versions 7.6, 7.7, and 7.8 along with CentOS 7.0.

Automated Switchover

The CX Insights server now supports automatic switchover to the IC secondary server when the IC primary server fails.

Reverse proxy server using nginx

We have provided steps to configure nginx as a reverse proxy server.

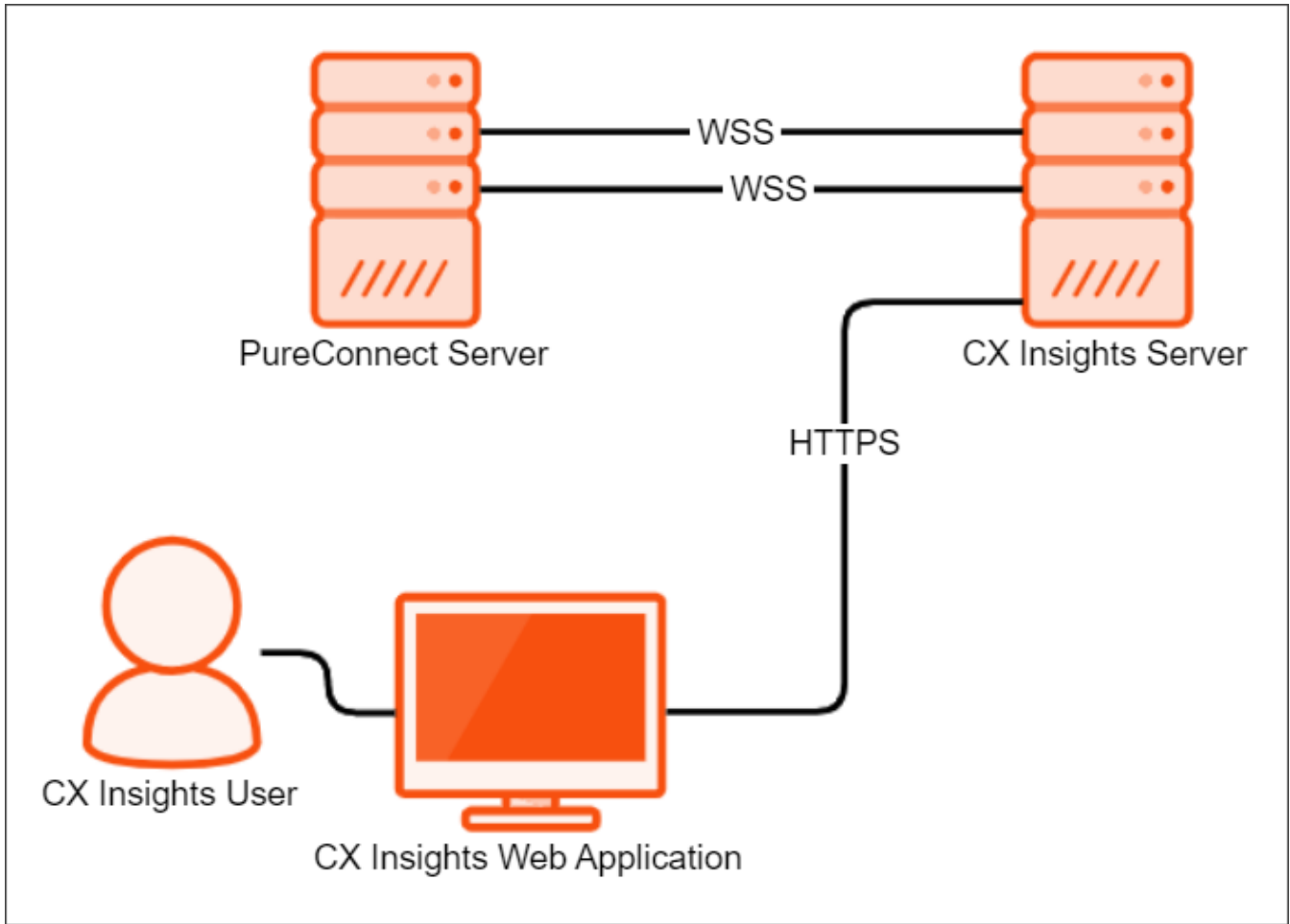
For more information about configuring nginx as a reverse proxy server, see [Configure reverse proxy using nginx](#).

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 view of available agents and their current states. Each agent or supervisor requires an assigned Analytics Core User license to log in, and an access permission to use the dashboards. In addition, you can configure a user with an Analytics Designer license who can create and modify the dashboards for agents and supervisors. 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. The current support is only for SQL and not Oracle database. CX Insights can be accessed from 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, and 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 prerequisites

CX Insights 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 Kubernetes.

Hardware

You can find the Genesys recommended hardware specifications in the following table. The sizing is arrived based on the number of active PureConnect users. Larger deployments may require more CPU and RAM to retain performance for the increased incoming traffic from the PureConnect Server.

Component	Large-size customers	Mid-size customers	Small-size customers
Number of agents in Contact Center	Above 400	50-400	Less than 50
Platform	Virtual machine or physical server	Virtual machine or physical server	Virtual machine or physical server
CPU	<ul style="list-style-type: none">8 coresAMD-V or VT-X VM-extensions	<ul style="list-style-type: none">8 coresAMD-V or VT-X VM-extensions	<ul style="list-style-type: none">4 coresAMD-V or VT-X VM-extensions
RAM	32 GB	20 GB	16 GB
Primary partition	100+ GB (recommended) 50 GB (minimum)	50 GB (recommended) 35 GB (minimum)	40 GB (recommended) 30 GB (minimum)
Secondary partition	400+ GB 100 GB (minimum)	60 GB (recommended) 45 GB (minimum)	45 GB (recommended) 35 GB (minimum)

Software

Important!

During installation of CentOS, you must include Virtualization Host to minimize the amount of extra configuration required to get Kubernetes running.

If Docker is already installed, ensure that you uninstall it.

Component	Requirement
Operating system	CentOS 7, RHEL version 7.6, 7.7 and 7.8 The host supports RHEL versions mentioned above. However, the base image in the container still contains CentOS and Alpine Linux.
Software components	Virtualization Host: <ul style="list-style-type: none"><li data-bbox="345 363 431 390">• KVM<li data-bbox="345 394 448 422">• QEMU<li data-bbox="345 426 513 453">• QEMU+KVM<li data-bbox="345 457 448 485">• Libvirt

Related Topics:

[Install CX Insights server](#)

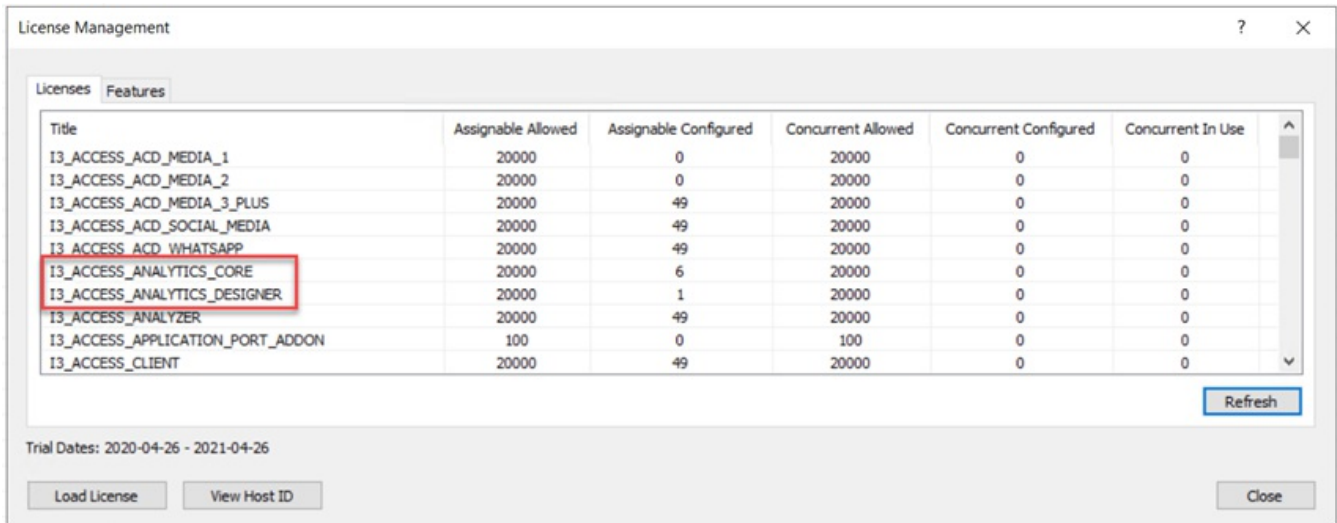
CX Insights licensing

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 availability of following licenses.

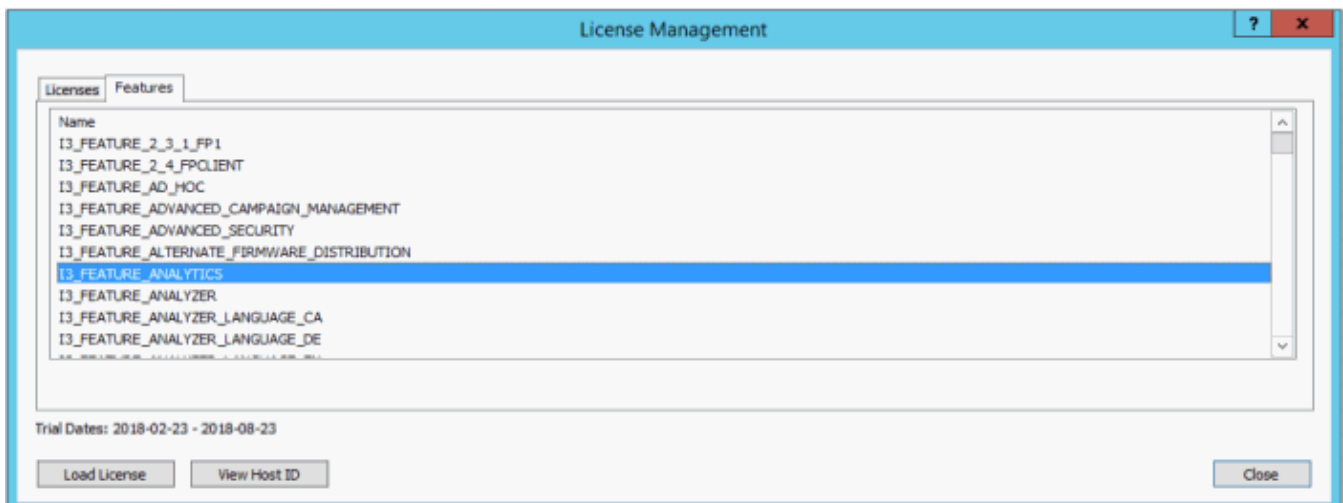
License	Description
I3_ACCESS_ANALYTICS_CORE	Basic dashboard license to view dashboards.
I3_ACCESS_ANALYTICS_DESIGNER	This license allows a user to create and modify dashboards.



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

Analytics feature license

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



If a license is not present or you do not have enough licenses, contact your sales representative. Also, see [Allocate Analytics licenses](#).

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, Red Hat Enterprise Linux (RHEL), Kubernetes, and Ansible.

Prerequisite

- CIC version must be 2020 R4.
- For CX Insights version 3.0 and above, the minimum required CIC version must be 2020 R3.
- If you are configuring the backup directory, then you must have the following:
 - A share path (for example, NFS share) of the remote computer where you are configuring the backup.
 - User installing the CX Insights server must have write access to the share path on the remote computer.

Install CX Insights server

1. Install CentOS 7 or RHEL version 7.6, 7.7 or 7.8 on either a physical or virtual server that meets the minimum requirements for the production environment. For more information about minimum requirements, see [CX Insights server requirements](#).
2. Download CX Insights Docker 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.tgz**, **pcon-mstr.zip**, and **cx-insights.tgz**.
4. Run the shell script **ansible_install.sh** to install the dependencies like Python, Ansible packages using the root user account. It also creates the CX Insights user account to perform all the Ansible roles and tasks.

Notes:

- 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 installation will fail.
 - By using the command **which ansible**, verify if Ansible is installed. If it is installed, you can see the Ansible version 2.9.10 and can also verify by running **ansible -version** command. Otherwise, rerun the **ansible_install** shell script.
 - By using the command **cut -d: -f1 /etc/passwd** and logging into CX Insights account, verify if CX Insights account is created.
 - **su cxinsights**
5. Prerequisite for running Ansible-playbook
 - Extract the **cxinsights-playbook-k3s.zip** file to the CX Insights user home directory. After extraction, move the **kube_archive_clean.py** file to the **/home/cxinsights** directory.
 - Generate Ansible vault for CX Insights user password. Ansible modules require this value 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. It asks for the password for vault usage, enter the password and make a note of it, so that the user can enter the same password while running **ansible-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 generated 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
    61636535326639666662393062353261316563333064646334666434386164393064343061353139
    3732313133373961313639343739366137326331663234610a626134663131363564393830393039
    62396531663664623436303032383861393164386235333736303465316235363339333034373563
    3461306464616666300a39636636663434643636637373364343263306664393632353536396630
    6637

tiller_linux_host_user: 'cxinsights'
tiller_linux_host_passwd: !vault |
    $ANSIBLE_VAULT;1.2;AES256;cxinsights
    30353265313036343933656165623965613262656539663962333630663531376135366261386564
    6132633435323835333834373432646634323063313931370a336466373030326466653262383330
    63613132643037313539633961336433633531353262336232626137393630383261396663313734
    3439623936313961660a613263363837303261363838313331323236306139313035346164646532
    3230

```

52,11

- Configure a backup directory and a cron job expression using the following parameters in the `group_vars/all.yml` file to backup CX Insights data.
 - **backup_dir** – specify the backup directory path. Configuring `backup_dir` is mandatory. For backup purposes, create the backup directory as a share path on a remote computer and mount the same on the local computer where you installed the CX Insights server. Example, `/mnt/nfs/share/gcxibackup`
 - **cron_schedule** – specify the cron expression that defines the backup frequency in which the backup activity runs. Configuring `cron_schedule` is optional. If you do not define any expression, the backup activity runs at the default time every day, that is at 12.00 am. An example cron expression to run the backup activity every day at 7.00 am and 12.00 pm looks like: `"0 7,12 * * *"`. Note that Cron job is added for the root user only.

You can also restore the backed-up data at a future date when there is a system failure. For more information about restore, see [Backup and restore configuration](#) topic.

- Specify the Genesys CX Insights (gcxi) properties in the `values.yml` file by referring to the following table:

Property name	Description
cicServerName	The IP address of the primary CIC server.
cicBackUpServerName	The IP address of the secondary CIC server.
cicDBName	The (SQL Server) CIC database name, specified in Setup Assistant.
cicDBHost	The (SQL Server) CIC database server name, specified in Setup Assistant.
cicDBLoginID	Specify the CIC database user ID of a user to read historical data from the database. The user ID you specify here is same as the IC Report Logs user ID specified in Interaction Administrator.
cicDBLoginPwd	Specify the encoded password of CIC database user ID mentioned in <code>cicDBLoginID</code> . Encrypt password using <i>base64</i> encryption method only. Tip: You can use the following command to encrypt your password: <code>echo "testpassword" base64</code>
langs (optional)	The localization language required for your organization. Configuring <code>langs</code> is optional. The US English (en_US) is mandatory. You can also specify other supported languages of your choice along with en_US. Currently, the supported language pack values are: en-US,fr-FR,de-DE,ja-JP,pt-BR,es-ES,zh-CN,nl-NL,pl-PL For more information about the language pack configuration, see the sample configuration given below this table.

certICSAML	Specify the certificate details required for SAML authentication. Copy the contents of the certificate details from the ICSecureTokenServerCertificate.cer file in the CIC Server IC-Token Service folder (I3\IC\Certificates\ICSecureTokenServer\Default\ICSecureTokenServerCertificate.cer) and paste it here.
proxyEndpoint	Specify the Fully Qualified Domain Name (FQDN) of a proxy server if the CX Insights server is accessed through a proxy server. If a proxy server is not configured in your environment, then you must specify the FQDN of the CX Insights server.
secret	Secret used for web socket authentication between the Analytics bridge and the microservices (mstrdataadapterserver and mstrtconnector). Ensure that the secret given here and the secret given in Interaction Administrator > System Configuration > Analytics > Configuration are same.
Global variables	
tz	Specify the time zone of the region where gcxi server is installed.
hosts	The Linux host name of the CX Insights server. Note that the host name you specify here must be an FQDN.
maxPoolSize (optional)	The maximum number of concurrent web sessions allowed. This is an optional parameter and the default value is 200.
tls (ingress)	<ul style="list-style-type: none"> If you do not want to enable TLS secured communication for ingress, keep the square brackets as given in the values.yml file, that is, []. If you want to enable TLS secured communication for ingress, remove the square brackets and specify the host name (ingress endpoint) and its secret. <p>Note: If you enable TLS, you must install an SSL certificate by following the Install SSL certificate on CIC server procedure.</p>
secret name (ingress)	Specify the Kubernetes cluster secret. We recommend that you keep the secret name value as given in the values.yml file, that is, pcn-cxinsights-tls
hosts (ingress)	Specify the FQDN of ingress host. Typically, this is the FQDN of the CX Insights server that you configure in the hosts setting.
tls (prometheusIngress)	<ul style="list-style-type: none"> If you do not want to enable TLS secured communication for Prometheus ingress, keep the square brackets as given in the values.yml file, that is, []. If you want to enable TLS secured communication for Prometheus ingress, remove the square brackets and specify the host name (Prometheus ingress endpoint) and its secret. <p>Note: If you enable TLS, you must install an SSL certificate by following the Install SSL certificate on CIC server procedure.</p>
secret name (prometheusIngress)	Specify the Kubernetes cluster secret. We recommend that you keep the secret name value as given in the values.yml file, that is, pcn-cxinsights-tls
hosts (prometheusIngress)	Specify the FQDN of Prometheus ingress host. Typically, this is the FQDN of the CX Insights server that you configure in the hosts setting.

Sample values.yml file configuration:

```
gcxi:
  gcxiproperties:
    cicDBName: I3_IC_MERCURY
    cicDBHost: qf-analyticstest.com
    cicServerName: 182.26.13.72
    cicBackupServerName: 182.26.13.72
    cicDBLoginID: "IC_ReadOnly"
    cicDBLoginPwd: "aTM="
    langs: en-US,fr-FR,de-DE,ja-JP,pt-BR,es-ES,zh-CN,nl-NL,pl-PL
    maxPoolSize: 250
    certICSAML:
      MIIDoTCCAomgAwIBAgIFQWCBgwkwDQYJKoZIhvcNAQEFBQAwRzEQMA4GA1UECgwH
      U2VydMVCyczEVMBMGA1UECwwMU2VydMVCyIEdyb3VwMRwwGgYDVQQDDDBNPbH1tcG1h
      LmRldjIwMDAuY29tMB4XDTEwMDMwNzIxNDQ0M1oXDTQwMDMwODIxNDQ0M1owRzEQ
      MA4GA1UECgwHU2VydMVCyczEVMBMGA1UECwwMU2VydMVCyIEdyb3VwMRwwGgYDVQQD
      DBNPbH1tcG1hLmRldjIwMDAuY29tMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIB
      CgKCAQEAs9WJ+2CqWRvQZs2Shc3kf/Ia+fOxW44SDgHxEMHKIqHx0rXwmuGbbqTTt
```

```

proxyEndpoint:
  - pcn-rhel7-rh8.testCXI.com
secret: analytics
global:
  tz: America/Indiana/Indianapolis
  hosts:
    - pcn-rhel7-rh8.testCXI.com
  ingress:
    tls:
      - secretName: pcn-cxinsights-tls
  hosts:
    - pcn-rhel7-rh8.testCXI.com
  prometheusIngress:
    tls:
      - secretName: pcn-cxinsights-tls
  hosts:
    - pcn-rhel7-rh8.testCXI.com

```

- Below is the **inventory.yml** file in the **cxinsights-playbook-k3s** directory, specify with appropriate values. For example: Assume Ansible and k3s are running on the same machine. If the controller is different from target machine, then **helm_linux_host** should be the controller host FQDN and **tiller-linux-host** should be the FQDN of the CX Insights server host.

```

---
helm_linux_host:
hosts:
xxx-xxxxxx-xxxxxx.xxxxxxxx.com
vars:
ansible_user: '{{ user }}'
ansible_ssh_pass: '{{ passwd }}'
tiller_linux_host:
hosts:
xxx-xxxxxx-xxxxxx.xxxxxxxx.com
vars:
ansible_user: '{{ user }}'
ansible_ssh_pass: '{{ passwd }}'

```

- If this is the fresh installation and you want to save the application data in secondary partition, keep the default value of **data_dir** as given in the **main.yml** file. The default value of **data_dir** is **/home/cxinsights/kube_data**. If you are already using the primary partition, modify the **data_dir** value in the **main.yml** file as shown below.

```
data_dir: ''
```

Note: If this is the fresh installation of CX Insights, we recommend that you deploy the software in secondary partition, provided you have the disk space as recommended in [step 1](#). Drive partitioning and using secondary drive to save CX Insights data is possible only for fresh installation. If CX Insights is already installed without partitioning the drive, you may not be able to use the secondary drive. In that case you must modify **data_dir** as ''.

6. Run the Ansible Playbook to start the services on the CX Insights server. For the first time, it is slow as dependencies get installed.

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

Note:

- Make sure you enter CX Insights password when BECOME password is asked.
- After the deployment is triggered, you must wait for some time until the state of GCXI pod is healthy.

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

```

cxinsights@qf-cx-docker:~/cxinsights-playbook-k3s
[cxinsights@qf-cx-docker cxinsights-playbook-k3s]$ kubectl get pods --namespace=pcn-cxinsights-system
NAME                                READY   STATUS    RESTARTS   AGE
pcn-cxinsights-helmcharts-gcxi-64dc7d94cf-149gc   0/1     Pending  0           22h
pcn-cxinsights-helmcharts-gcxi-postgres-d6676fbc6-wqd86   1/1     Running  0           22h
pcn-cxinsights-helmcharts-mstrconnector-5b64f74bff-xtlpc   0/1     Running  0           22h
pcn-cxinsights-helmcharts-mstrdataadapterserver-5dc956d459dbzvp   0/1     Running  0           22h
pcn-cxinsights-helmcharts-mstrdataadapteragent-6d99cd4df4-lxxh5   1/1     Running  0           22h
[cxinsights@qf-cx-docker cxinsights-playbook-k3s]$

```

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

```

cxinsights@qf-cx-docker:~/cxinsights-playbook-k3s
[cxinsights@qf-cx-docker cxinsights-playbook-k3s]$ kubectl get services --namespace=pcn-cxinsights-system
NAME                                TYPE           CLUSTER-IP   EXTERNAL-IP   PORT(S)          AGE
pcn-cxinsights-helmcharts-pcon-mstr   ClusterIP      192.168.194.246   <none>        80/TCP           22h
pcn-cxinsights-helmcharts-mstrdataadapterserver-agentgateway   ClusterIP      192.168.166.236   <none>        8079/TCP         22h
pcn-cxinsights-helmcharts-mstrdataadapterserver   ClusterIP      192.168.206.232   <none>        8078/TCP, 9090/TCP   22h
pcn-cxinsights-helmcharts-mstrdataadapteragent   ClusterIP      192.168.231.167   <none>        9090/TCP         22h
pcn-cxinsights-helmcharts-gcxi        ClusterIP      192.168.133.216   <none>        34952/TCP, 8080/TCP   22h
pcn-cxinsights-helmcharts-mstrconnector   ClusterIP      192.168.142.47    <none>        8077/TCP, 9090/TCP   22h
gcxi-postgres                          ClusterIP      192.168.137.210   <none>        5432/TCP, 9090/TCP   22h
[cxinsights@qf-cx-docker cxinsights-playbook-k3s]$

```

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

```

cxinsights@pcn-cent7-k3s01:~
[cxinsights@pcn-cent7-k3s01 ~]$ kubectl get pvc --namespace=pcn-cxinsights-system
NAME          STATUS   VOLUME                                     CAPACITY   ACCESS MODES   STORAGECLASS   AGE
gcxi-log      Bound   pvc-b6d9f121-77dc-4d10-93e9-a932f0e14bcf   2Gi        RWO             local-path     13d
gcxi-data     Bound   pvc-30e7b3ed-8b56-476c-881d-7b1c3a0da536   8Gi        RWO             local-path     13d
gcxi-shared   Bound   pvc-09b8c38a-2283-458e-894a-63faf2c502aa   1Gi        RWO             local-path     13d
gcxi-volume   Bound   pvc-e0fefb0d-4624-4ce4-bce7-2bceff7ec0b6   2Gi        RWO             local-path     13d
cube          Bound   pvc-67f2cbcd-abb6-4da1-8053-3b7605cac2f3   1Gi        RWO             local-path     13d
[cxinsights@pcn-cent7-k3s01 ~]$

```

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.

Related Topics:

- [Install SSL certificate on CIC server](#)
- [Ports exposed on CX](#)
- [Configure CX Insights in Interaction Administrator](#)
- [Backup and restore configuration](#)
- [Troubleshooting](#)
- [Upgrade containers](#)

Install SSL certificate on CIC server

The communication between the CIC server and Kubernetes is secured over the TLS protocol. This requires an installation of a valid SSL certificate signed by a third party or a self-signed SSL certificate which is auto generated in the file name `tls.crt` in the `/root` directory of the CX Insights server.

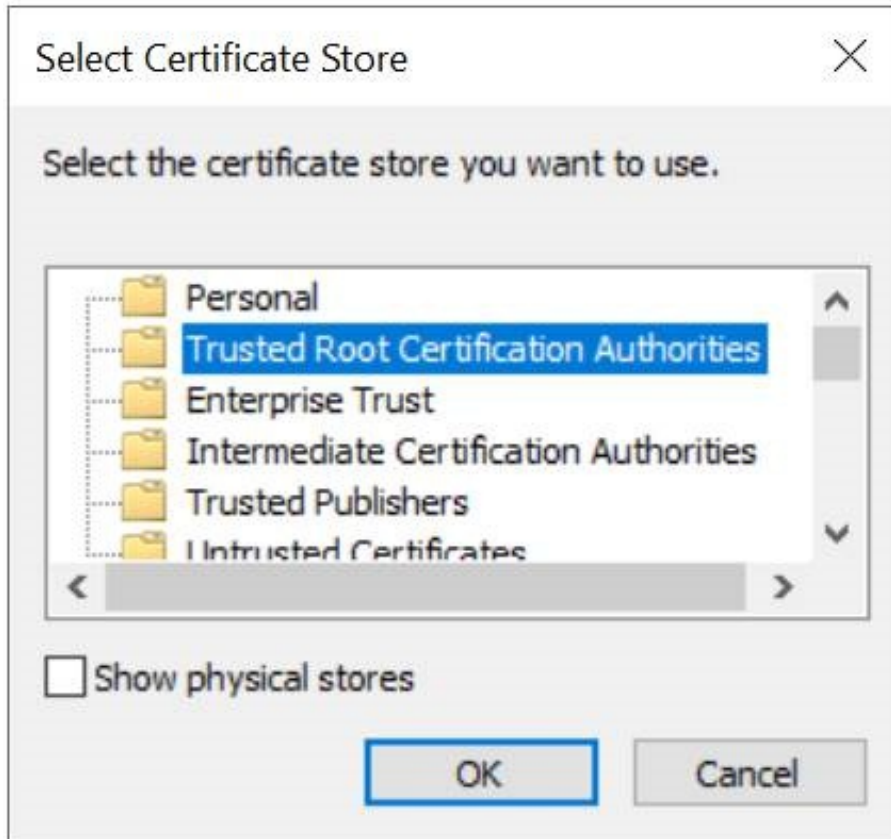
Note:

- If you enable TLS in `values.yml` file for ingress or Prometheus ingress, you must install a valid SSL certificate.
- Ensure that you install the SSL certificate in both the primary and secondary CIC servers.

To install the SSL certificate,

1. Copy the SSL certificate from the CX Insights server to a wanted location on the CIC server.

2. Right-click on the SSL certificate (tls.crt) from the CIC server and click **Install Certificate**.
3. On the **Certificate Import Wizard**, in the **Store Location** section, select **Local Machine**, and click **Next**.
4. Select **Place all certificates in the following store** option.
5. Click **Browse**. On the **Select Certificate Store** pop-up, select **Trusted Root Certification Authorities** as the certificate store and click **Ok**.



6. On the **Certificate Import Wizard**, verify the Certificate store selection and click **Next**.

Certificate Store

Certificate stores are system areas where certificates are kept.

Windows can automatically select a certificate store, or you can specify a location for the certificate.

- Automatically select the certificate store based on the type of certificate
- Place all certificates in the following store

Certificate store:

Trusted Root Certification Authorities

Browse...

Next

Cancel

7. Click **Finish**. A dialog showing the message **"The import was successful."** appears if the certification installation is correct.
8. Click **Ok**.

Related Topics:

[Install CX Insights server](#)

[Ports exposed on CX Insights server](#)

[Configure CX Insights in Interaction Administrator](#)

Upgrade containers

You can upgrade the CX Insights' containers whenever there is a new Analytics release with new features and critical updates.

To upgrade containers,

1. In the `values.yml` file, update proper tag name for containers that need upgrade, see example below. If you want to upgrade only one container, then add tag for the corresponding container and you can omit 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
```

2. Run the following command in the path `/home/cxinsights/cxinsights-playbook-k3s/`

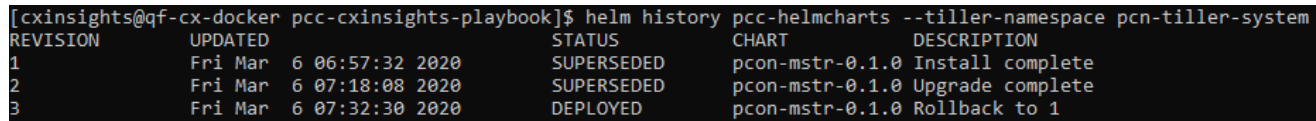
```
sudo ansible-playbook -i inventory.yml site_upgrade.yml -K
```

Roll back containers

To roll back containers, get the list of versions installed by running the following command.

```
helm history pcc-helmcharts --tiller-namespace pcn-tiller-system
```

Sample output shown in the following screenshot



```
[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 needs to be rolled back in `roles/helm-chart-rollback/vars/main.yml` file and run the following command:

```
sudo ansible-playbook --vault-id cxinsights@prompt -i inventory.yml site_rollback.yml -K
```

Deleting deployment

Use the following command to delete the entire deployment such as pods, services, ingress endpoints, and persistent volumes.

```
sudo ansible-playbook --vault-id cxinsights@prompt -i inventory.yml site_delete.yml -K
```

Running the above command is equivalent to `helm delete` command.

Related Topics:

[Install CX Insights server](#)

CX Insights monitoring and alerting

Install Prometheus

Prometheus is an open source software licensed under the Apache 2.0 license. When you install Prometheus, make sure that you install Prometheus in a private network.

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

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

```

4. After running Prometheus executable, ensure <http://localhost:9090/rules> is accessible and all rules are defined properly. Warning and critical alerts are configured, 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 container state.

Prometheus Alerts Graph Status Help

All Unhealthy

Connector (1/1 up) [show less](#)

Endpoint	State	Labels	Last Scrape	Scrape Duration	Error
http://pcn-cent7-k3s04.ininlab.com:80/ConnectorMetrics	UP	instance="pcn-cent7-k3s04.ininlab.com:80" job="Connector"	11.988s ago	661.6ms	

DataAdapterAgent (1/1 up) [show less](#)

Endpoint	State	Labels	Last Scrape	Scrape Duration	Error
http://pcn-cent7-k3s04.ininlab.com:80/DataAdapterAgentMetrics	UP	instance="pcn-cent7-k3s04.ininlab.com:80" job="DataAdapterAgent"	8.275s ago	3.639s	

DataAdapterServer (1/1 up) [show less](#)

Endpoint	State	Labels	Last Scrape	Scrape Duration	Error
http://pcn-cent7-k3s04.ininlab.com:80/DataAdapterServerMetrics	UP	instance="pcn-cent7-k3s04.ininlab.com:80" job="DataAdapterServer"	5.304s ago	658.4ms	

GCXi (1/1 up) [show less](#)

Endpoint	State	Labels	Last Scrape	Scrape Duration	Error
http://pcn-cent7-k3s04.ininlab.com:80/mstr-integrationapi/GcxiMetrics	UP	instance="pcn-cent7-k3s04.ininlab.com:80" job="GCXI"	1.803s ago	328.8ms	

Postgress (1/1 up) [show less](#)

Endpoint	State	Labels	Last Scrape	Scrape Duration	Error
http://pcn-cent7-k3s04.ininlab.com:80/PostgresMetrics	UP	instance="pcn-cent7-k3s04.ininlab.com:80" job="Postgress"	1.357s ago	340.1ms	

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

```

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

```

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

```

route:
  group_by: ['alertname']
  group_wait: 30s
  group_interval: 10s
  receiver: 'email-me'
  routes:
  - match:
    severity: warning
    repeat_interval: 1h
  - match:
    severity: critical
    repeat_interval: 15m
  receivers:
  - name: 'email-me'
  email_configs:
  - to: xxxxxxxx@xxxx.com
    from: xxxxxxxx@xxxx.com
    smarthost: xxxxx
    auth_username: ""
    auth_password: ""

```

Configure reverse proxy using nginx

You can install a public facing reverse proxy server and route all the incoming requests to the CX Insights server through proxy. Genesys verified the nginx reverse proxy server for the CX Insights server.

To install the nginx reverse proxy server, see [nginx documentation](#).

To configure a reverse proxy server,

1. Find the **nginx.conf** in the installed path and copy the code given [here](#) to the **nginx.conf** file.
2. Within the copied code, update the appropriate values for the following parameters:
 - **<dns_server_name>** - specify the dns server name of the server where nginx is installed.
 - **<proxy_server_name>** - specify the host name where nginx is installed.
 - **<cxinsight_server_name>** - specify the server name where the CX Insights server is installed.
- If you don't have a TLS certificate from a Certification Authority, generate a self-signed certificate by using the following command. Copy the generated certificate (**tls.crt**) and key file (**tls.key**) file under the nginx directory.

```
openssl req -x509 -newkey rsa:4096 -sha256 -nodes -keyout /etc/nginx/tls.key -out /etc/nginx/tls.crt -subj '/CN=<proxy_server_name>' -days 365
```

Note: Make sure that you configure TLS certificate and private key correctly, otherwise you cannot log in to CX Insights server.

4. Test the updated configuration in the **nginx.conf** file by running the following command. We recommend to test the configuration for any syntax errors whenever you make changes in the configuration file.

```
nginx -t
```

5. Restart the nginx service. Note that any changes in the **nginx.conf** file requires a restart of the nginx service.

Log file

You can view the error log file from the default path **/var/log/nginx/error.log**. If you want to set up a different path, you can do so in the **error_log** parameter in **nginx.conf**.

Code to be copied to the nginx.conf file

```
user nginx;
worker_processes auto;
error_log /var/log/nginx/error.log;
pid /run/nginx.pid;
# Load dynamic modules. See /usr/share/doc/nginx/README.dynamic.
include /usr/share/nginx/modules/*.conf;
events {
worker_connections 1024;
}
http {
resolver <dns_server_name> valid=90000000s;
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 /var/log/nginx/access.log main;
sendfile on;
tcp_nopush on;
tcp_nodelay on;
keepalive_timeout 65;
types_hash_max_size 2048;
include /etc/nginx/mime.types;
```

```

default_type application/octet-stream;
# Load modular configuration files from the /etc/nginx/conf.d directory.
# See http://nginx.org/en/docs/nginx_core_module.html#include
# for more information.
include /etc/nginx/conf.d/*.conf;
server {
listen 80;
listen [::]:80;
server_name _;
root /usr/share/nginx/html;
# Load configuration files for the default server block.
include /etc/nginx/default.d/*.conf;
location / {
}
error_page 404 /404.html;
location = /40x.html {
}
error_page 500 502 503 504 /50x.html;
location = /50x.html {
}
}
# Settings for a TLS enabled server.
#
server {
listen 443 ssl http2 default_server;
listen [::]:443 ssl http2 default_server;
server_name "<proxy_server_name>";
root /usr/share/nginx/html;
ssl_certificate "/etc/nginx/tls.crt";
ssl_certificate_key "/etc/nginx/tls.key";
ssl_session_cache shared:SSL:1m;
ssl_session_timeout 10m;
ssl_ciphers HIGH:!aNULL:!MD5;
ssl_prefer_server_ciphers on;
# Load configuration files for the default server block.
include /etc/nginx/default.d/*.conf;
location ~ ^/(MicroStrategy|cic|WindowsIDP|ICNotifierIDP)/ {
error_log /var/log/nginx/error.log debug;
proxy_pass $scheme://<cxinsight_server_name>$request_uri;
proxy_set_header HOST $host;
proxy_set_header X-Forwarded-Proto $scheme;
proxy_set_header X-Real-IP $remote_addr;
proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
}

```

```
error_page 404 /404.html;
location = /40x.html {
}
error_page 500 502 503 504 /50x.html;
location = /50x.html {
}
}
}
```

Related Topics:

[Install CX Insights server](#)

Ports opened on CX Insights server

At the end of installation, the following ports are opened on the CX Insights server.

Port Number	Description
80	Web server default port
8080	Tomcat server port
443	Https connection port
6443	Secured port for tiller communication
5432	PostgreSQL port
34952	Intelligence server port
8077	Mstr connector port
8078	Mstr data adapter server port
8079	Mstr agent server port
9090	Prometheus port
8008	Endpoint update service port

Related Topics:

[Install CX Insights server](#)

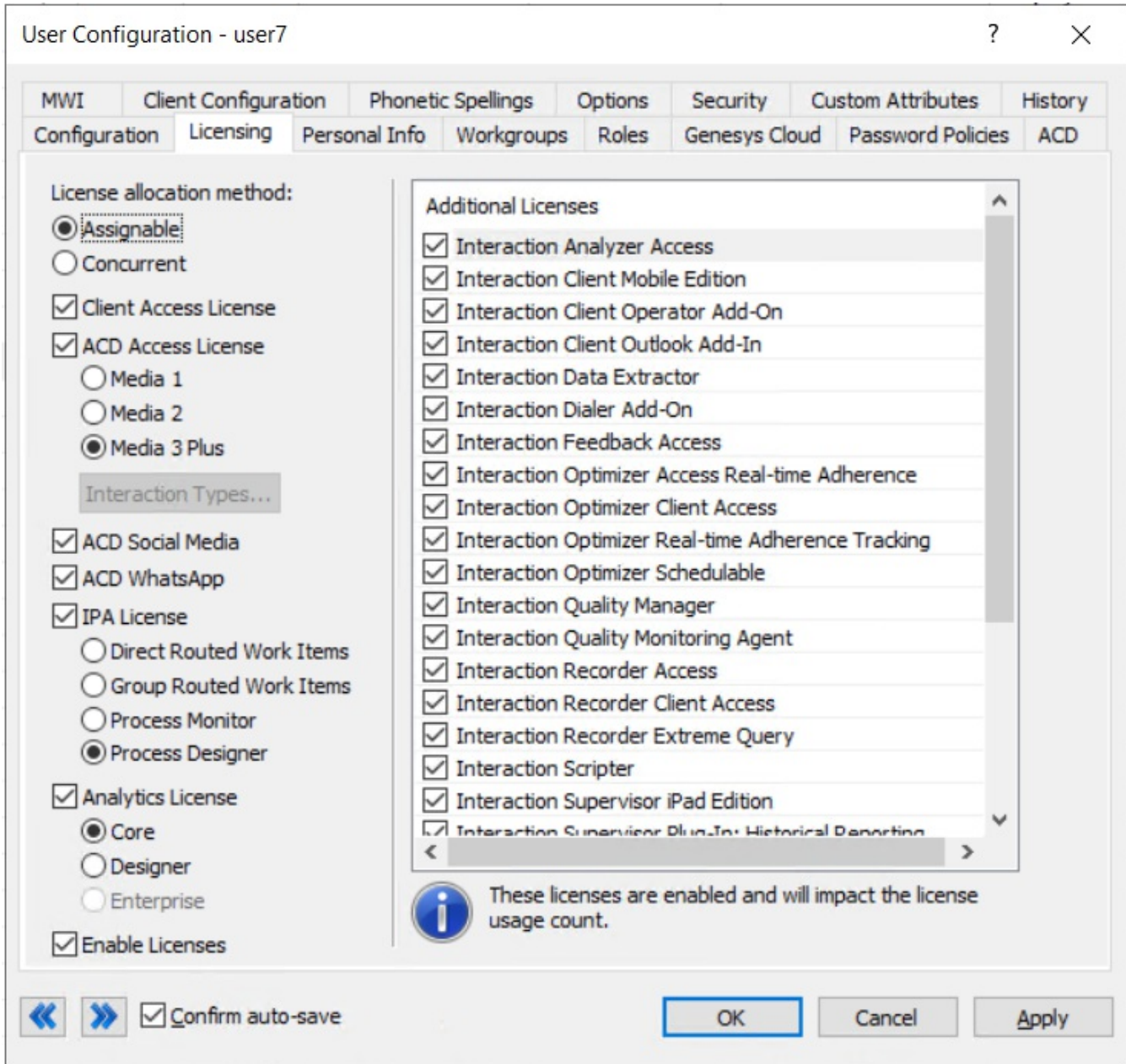
CX Insights server configuration

CX Insights server configuration

To configure the CX Insights server settings in Interaction Administrator, complete the following steps.

Allocate Analytics licenses

You can 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.
DESIGNER	This license allows a user to create and modify dashboards.

In addition, you must select the **Enable Licenses** check box to activate the Analytics license.

Related Topics:

[Install CX Insights server](#)

[Configure CX Insights in Interaction Administrator](#)

[Troubleshooting](#)

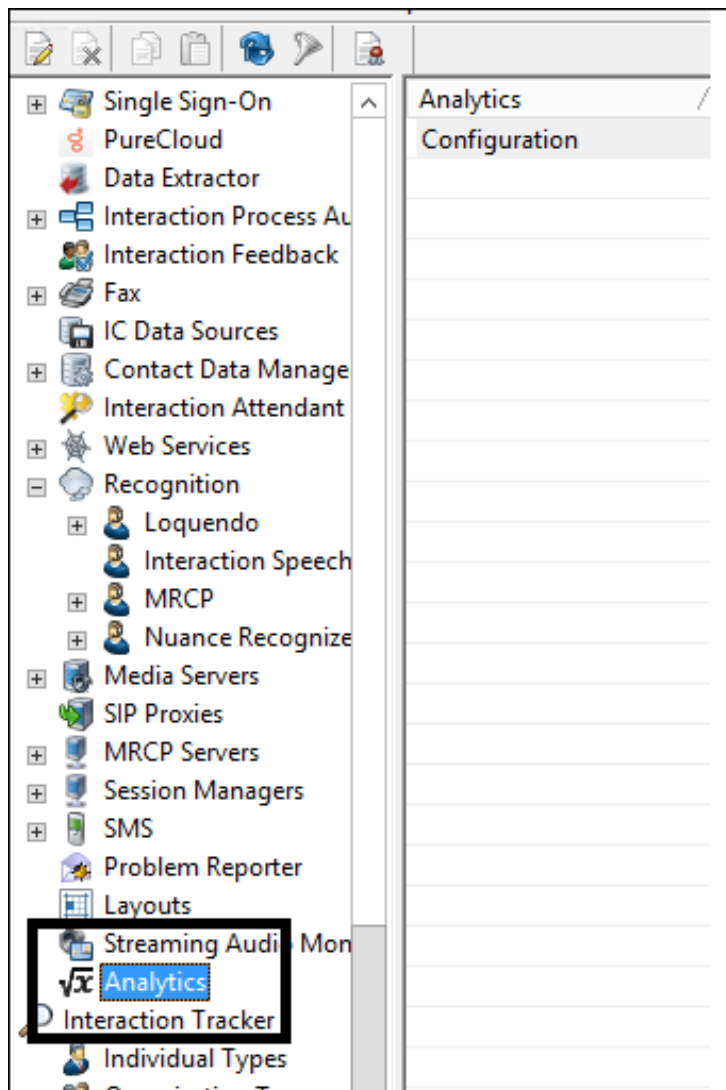
Configure CX Insights server in Interaction Administrator

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

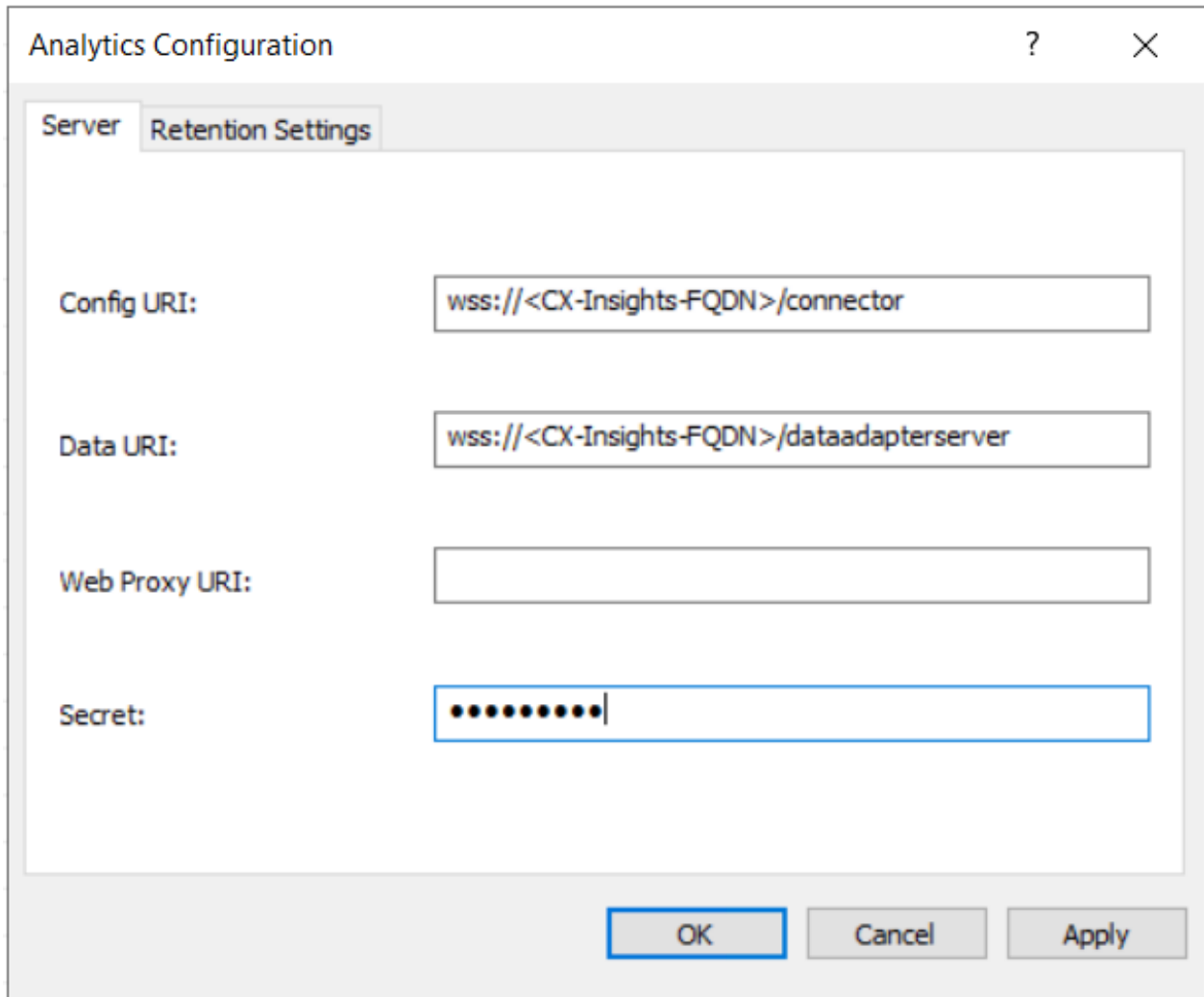
1. Apply the I3_FEATURE_ANALYTICS license to the PureConnect server.

To apply the I3_FEATURE_ANALYTICS license, open **Interaction Administrator** and go to **File > License Management > Features** tab, select the license, and click **Load License**.

2. Open Interaction Administrator and open the Analytics Node under System Configuration.



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



On the **Server** tab, configure the following values:

- **Config URI** - is the web socket address that PureConnect uses to synchronize configuration and security settings with the CX Insights server (default port shown). Configure the value as shown in the above screenshot and replace **<CX-Insights-FQDN>** value with the CX Insights' server name. **Note:** If you are using secured communication (enabled TLS), configure the URI value as **'wss'** else use **'ws'**.
- **Data URI** - is the web socket address through which PureConnect streams real-time statistics to the CX Insights server. Configure the value as shown in the above screenshot and replace **<CX-Insights-FQDN>** value with the CX Insights' server name. **Note:** If you are using secured communication (enabled TLS), configure the URI value as **'wss'** else use **'ws'**.
- **Web Proxy URI** - is the target URL used by HttpPluginHost to route web requests.
- **Secret** - is the secret that was entered in the **secret** field in the **values.yml** file when deploying the CX Insights Server.

Once Configuration is complete, the AnalyticsBridge subsystem will attempt to make the configured web socket connections. If the connections are established successfully, the synchronization process begins. Synchronization can take a few minutes to complete if there are large number of users and workgroups to transfer. Any additional changes to Users, Roles, Workgroups, Access Controls, or Memberships trigger extra synchronization cycles. Once the servers are synchronized, the AnalyticsBridge Subsystem begins streaming real-time statistics over the data web socket. At that point, users can view the real-time dashboards.

Retention Settings

Using retention settings, you can define how many days you want to retain the IVR data history. Based on the settings, the historical IVR data will be purged at the specified interval. For more information, see [Retention settings](#) in Interaction Administrator help.

Related Topics:

[Install CX Insights server](#)

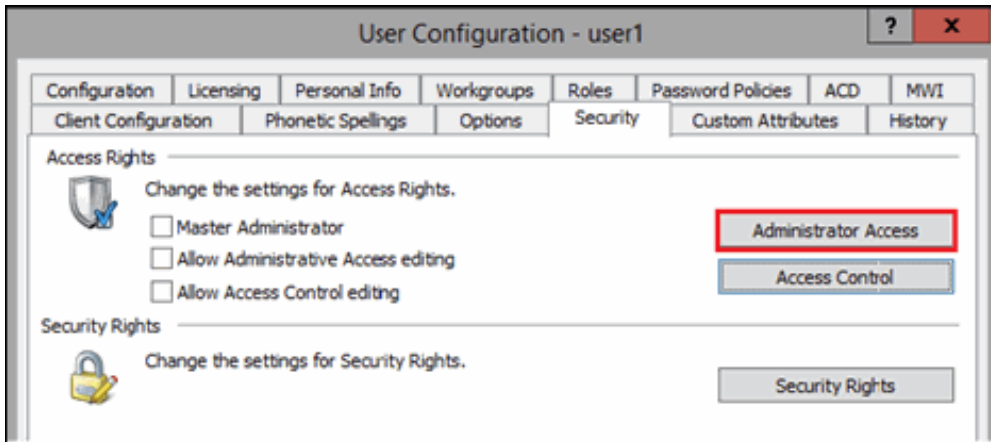
[CX Insights licensing](#)

Configure Administrator Access for CX Insights

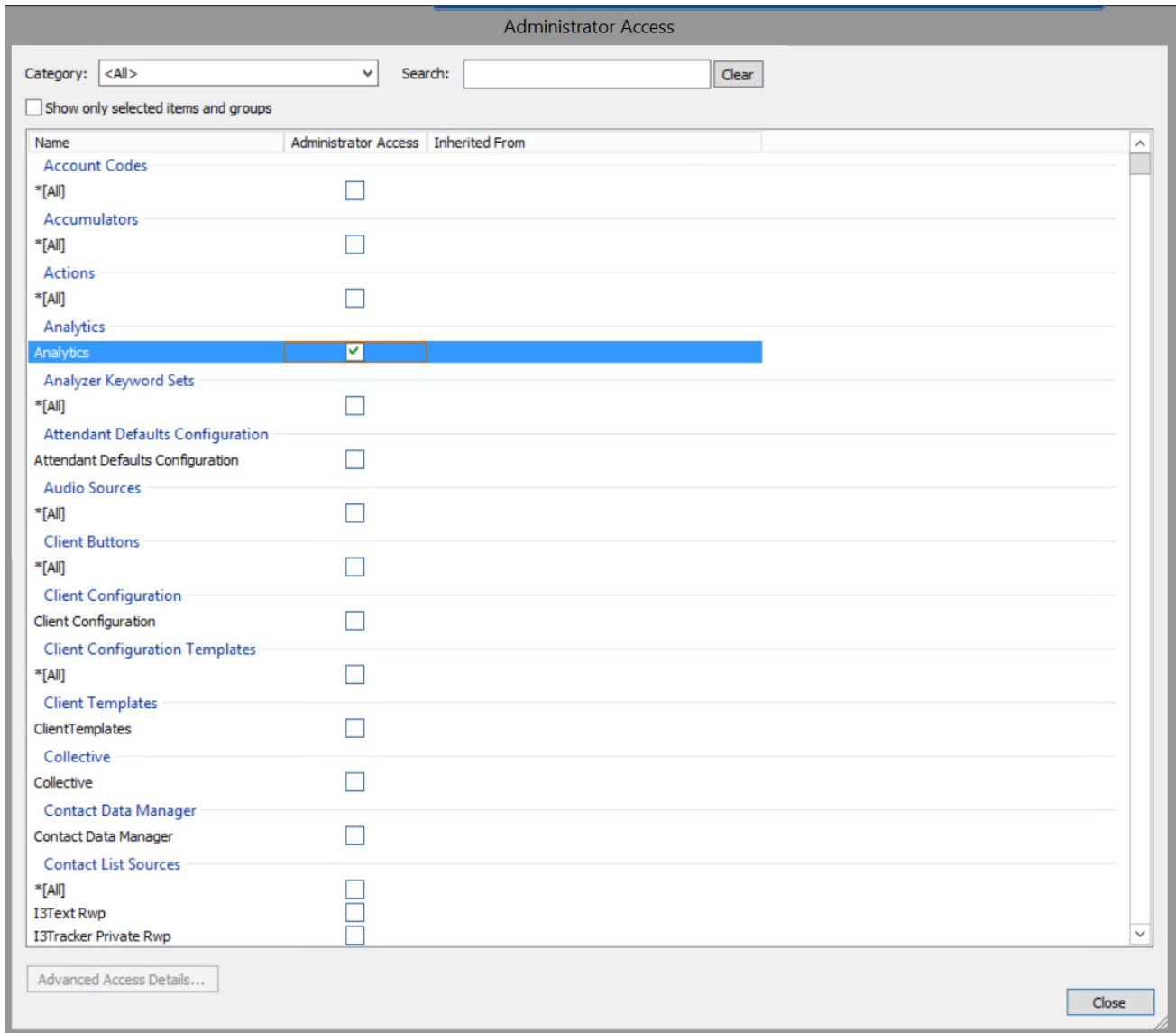
You can restrict which user, workgroup, or role has access to configure the Analytics feature.

To assign administrator access for Analytics:

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



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



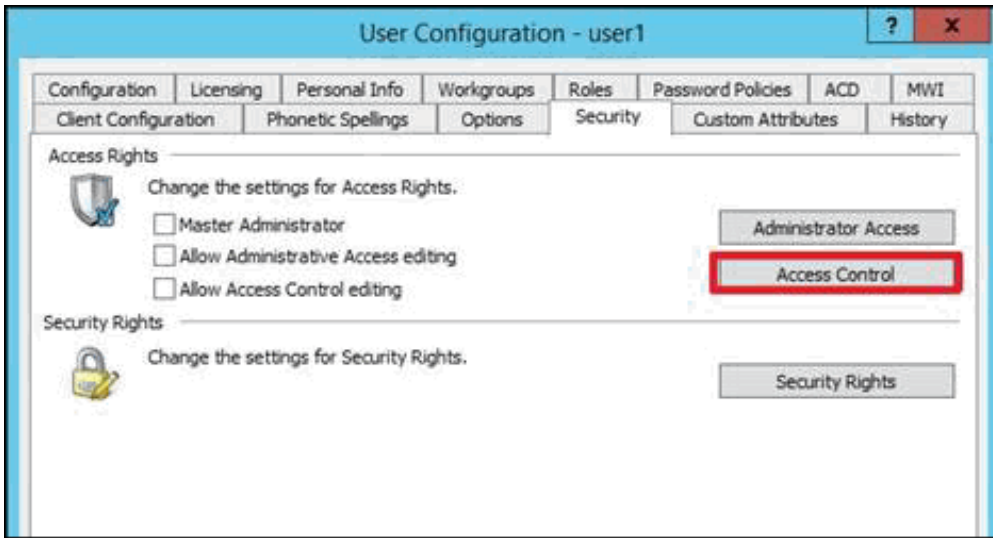
5. To give a user, workgroup, or role Administrator Rights to the Analytics feature, select the **Analytics** check box. You can clear the check box to remove the privilege.
6. Click **Close**.
7. To save the settings, click **OK** or **Apply**.

Configure Access Control for CX Insights dashboards

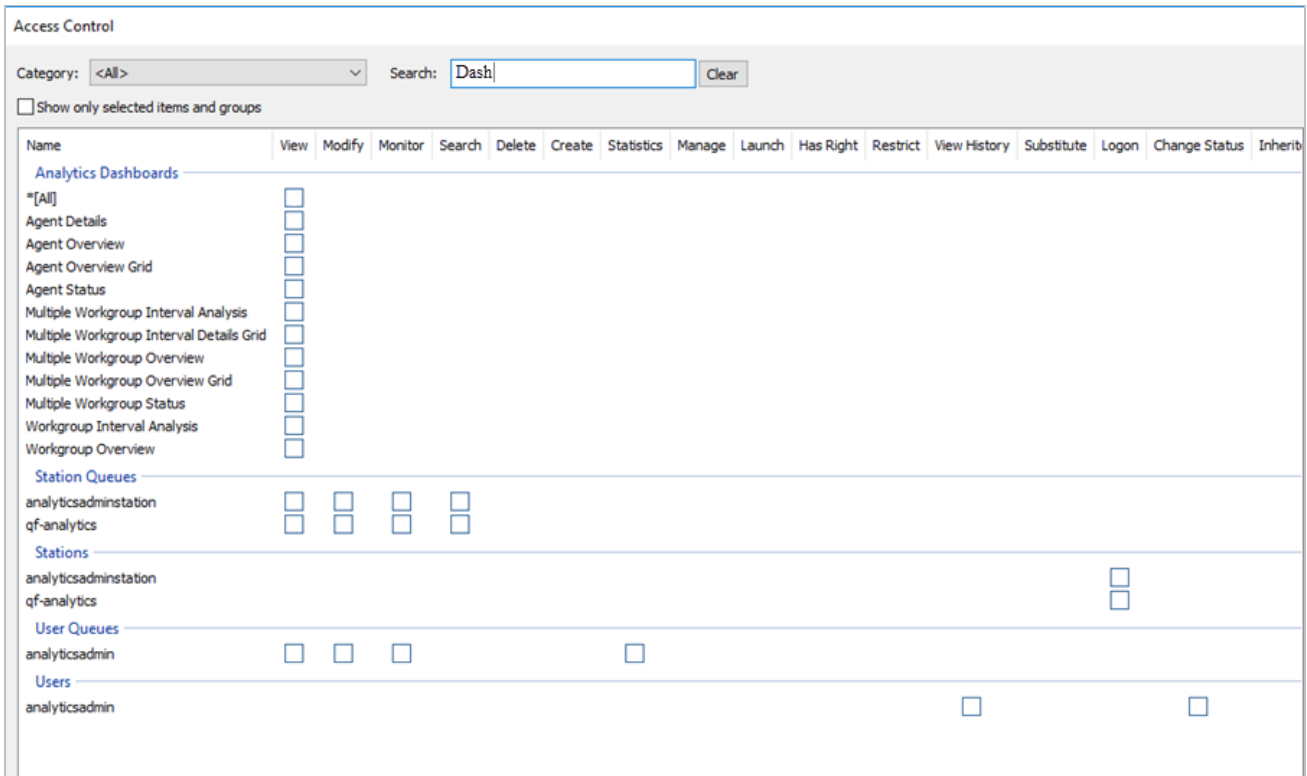
You can restrict which user, workgroup, or role has access to specific dashboards.

To assign dashboard access:

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



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



Note:

If the IC Server is in sync with the MicroStrategy server, then you can see the check boxes for all the dashboards.

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.

Test the CX Insights installation

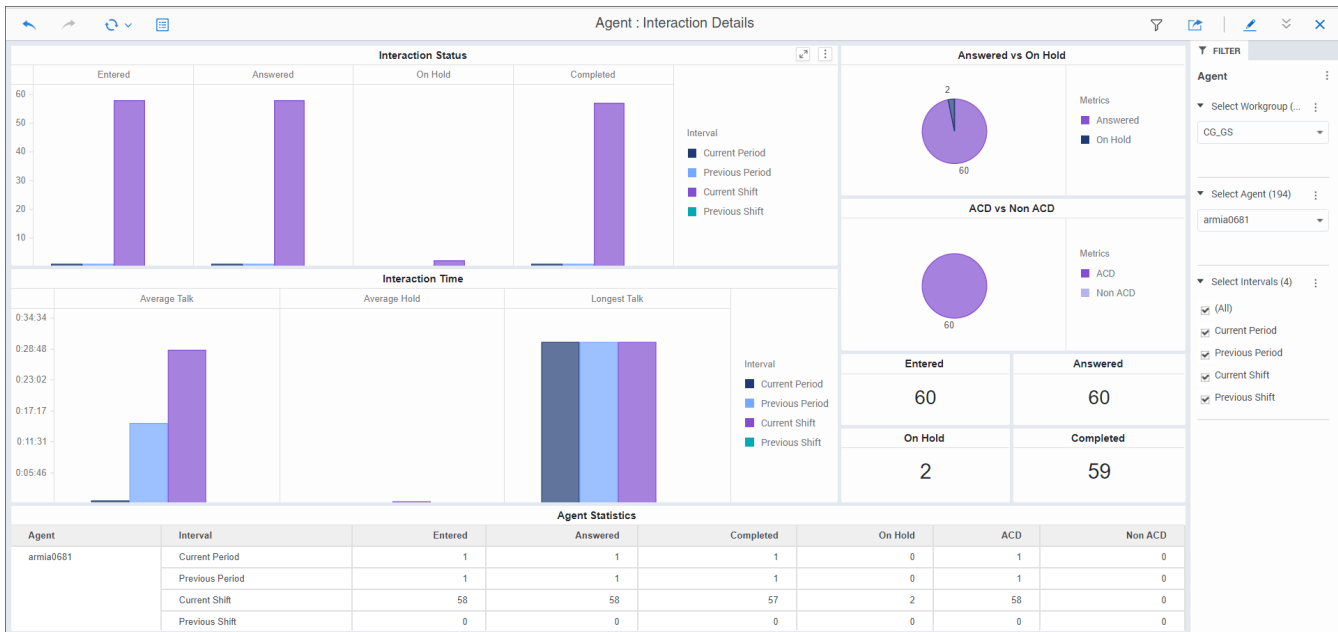
After you complete the initial configuration and user access, test the CX Insights installation by opening a CX Insights dashboard.

To access a dashboard,

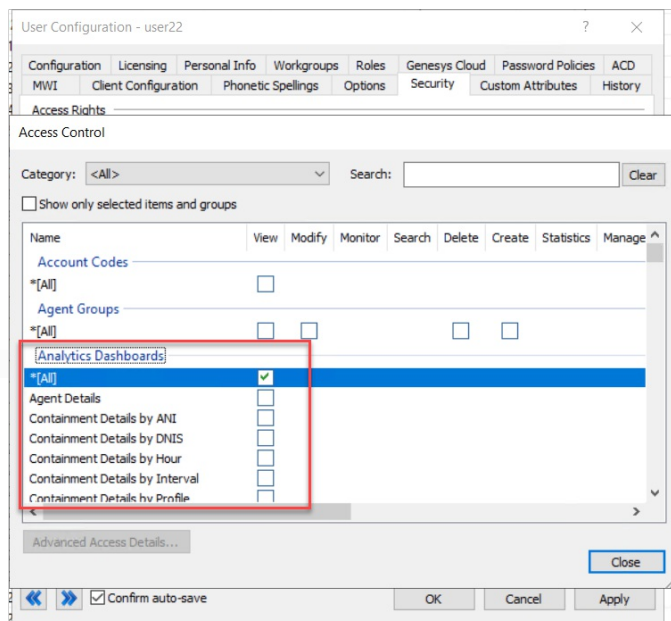
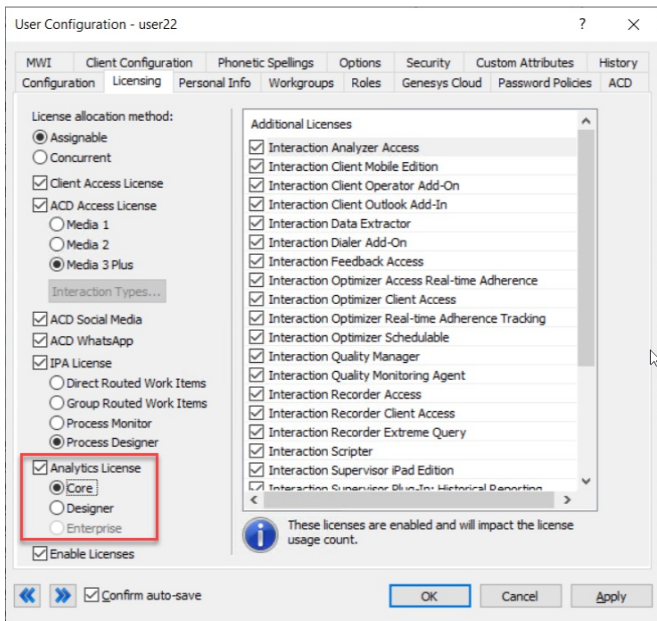
1. Log in to CX Insights. You can use the same login credentials that you use for PureConnect.
2. Click the **CX Insights** folder.
3. Select **IVR Dashboards** or **Real Time Dashboards**. Both these dashboards offer a range of metrics presented in different views.
4. Select the dashboard you want to explore. For example, the following image shows the **Agent Details** dashboard.

Note!

You can only view the dashboards for which you have access permissions defined in the CIC server. After successful loading, the Real Time dashboards refresh every 30 seconds with real-time statistic values.



The dashboards you can view depends on the Analytics license type (Designer/Core) you are assigned and the access permissions to view.



Backup and restore configuration of CX Insights data

CX Insights allows you to backup data at regular intervals. In case, there is a system failure, you can also [restore](#) the backed-up data to a new computer.

The procedures in this topic help you to configure data backup and restore settings for CX Insights.

Backup CX Insights data

You can configure the backup settings either in an `all.yml` file or run a script manually.

Configure CX Insights backup through Ansible

In this method, you can configure the backup criteria through Ansible installation. To start with, configure backup values even before running the Ansible installation. For more information about Ansible installation, see [CX Insights server installation](#) procedure.

Prerequisite

- A share path (for example, NFS share) on the computer where you are configuring the backup.
- User installing the CX Insights server must have write access to the share path.

To configure the backup settings

1. Mount the shared backup directory (example, NFS share) on the local computer where you installed the CX Insights server. For example, `/mnt/nfs/share`. The mounted directory is the backup path that maintains the CX Insights backup data. You can verify the mounted path using the `mount|grep` command as shown in the following example.

```
mount|grep "/mnt/nfs/share"
```

2. Configure the following values in the `group_vars/all.yml` file.

- **backup_dir** – specify the backup directory path. For example, `/mnt/nfs/share/gcxibackup`. Configuring `backup_dir` is mandatory.
- **cron_schedule** - specify the cron expression that defines the backup frequency in which the backup activity runs every day. Configuring `cron_schedule` is optional. However, if you do not define any expression, the backup activity runs at the default time every day, that is 12.00 am. An example cron expression to run the backup activity every day at 7.00 am and 12.00 pm looks like: `"0 7,12 * * *`

Note: Cron job is added for the root user only.

3. Convert the `cxinsight-backup-restore.sh` file to Unix format. You can do the conversion either by running the `dos2Unix` tool or by running the `sed` command as shown below.

```
sed -i 's/\r//g' cxinsight-backup-restore.sh
```

4. Log in as CX Insights user and run the [Ansible installation](#) using the following command.

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

Note:

- Running the above Ansible installation command installs the pods, and configures the backup settings which generate the `.gcxi_backup_cron.sh` file at `/home/cxinsights/.gcxi_backup_cron.sh`. However, it does not perform the actual backup. The actual backup is performed when the first scheduled backup activity runs or when the user backs up manually.
- If backup configuration causes any errors in Ansible installation, correct the errors, and configure the backup settings [manually](#).
- You can verify the backup activity logs from the path `/home/cxinsights/.gcxi_backup_trace.log`

Configure CX Insights backup through script

For some reasons, if Ansible installation fails to configure the backup settings, you can configure it manually by running a script.

To configure the backup settings

1. Mount the shared backup directory (example, NFS share) on the local computer where you installed the CX Insights server. For example, `/mnt/nfs/share`. The mounted directory is the backup path that maintains the CX Insights backup data. You can verify the mounted path using the `mount|grep` command as shown in the following example.

```
mount |grep "/mnt/nfs/share"
```

2. Run the script **cxinsight-backup-restore.sh** manually by providing the backup path and cron expression as shown below in the path `/home/cxinsights/cxinsights-playbook-k3s`

Syntax:

```
sudo cxinsight-backup-restore.sh backup <backup dir> ["Cron expression" (optional)]
```

Example:

```
sudo cxinsight-backup-restore.sh backup /mnt/nfs/share/gcxibackup "* */6 * * *"
```

Important:

- Run the **cxinsight-backup-restore.sh** script only once. Rerunning the script overwrites log file and backs up old data in the configured backup path.
- If you accidentally delete the volumes folder (for example, through helm delete), you must rerun the **cxinsight-backup-restore.sh** script to set up the backup path and the cron job schedule.

Instant backup

Run the following script if you want to backup CX Insights data instantly instead of waiting for the scheduled backup activity.

```
sudo /home/cxinsights/.gcxi_backup_cron.sh
```

Restore CX Insights data

You might want to restore old CX Insights data in case you replaced or upgraded your hardware. You can restore older data if you have a proper backup and you know the correct backup path.

You can provide the restore settings either in an `all.yml` file or run a script manually.

Prerequisite

A share path (for example, NFS share) of the computer where you are restoring the backup.

Configure CX Insights data restore through Ansible

To configure the restore settings,

1. Follow the steps 1-3 in configuring [CX Insight backup through Ansible](#) procedure.
2. Verify that the mounted directory has the following volume folders.

```
$ ls /mnt/nfs/share/gcxibackup
cube  gcxi-data  gcxi-volume
```

3. In the `group_vars/all.yml` file, configure `is_restore` as `true`.
4. Log in as CX Insights user and run the [Ansible installation](#) using the following command.

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

Note:

- Running the above command restores the CX Insights data and creates a new backup directory. You can find the restored data in an archive file created within the backup directory. The archive file is created with the date and time (example, `gcxi-backup_2020-08-06_01-55-36.tar.gz`) so that you can identify which file is relevant for you.
 - The Ansible installation requires several parameters to be configured as part of CX Insights server installation. For more information, see [Install CX Insights server](#).
5. Once restoration is successful and complete, change `is_restore` to its default value (`false`) in the `group_vars/all.yml` file. Changing `is_restore` back to its default value avoids unnecessary data restore during future upgrades.

Configure CX Insights data restore through script

If automatic restoration fails for any reason, you can restore the CX Insights data manually by using the following procedure.

1. Follow the steps 1-3 in configuring [CX Insight backup through Ansible](#) procedure.

2. Verify that the mounted directory has the following volume folders.

```
$ ls /mnt/nfs/share/gcxibackup
cube gcxi-data gcxi-volume
```

3. Run the restore script **cxinsight-backup-restore.sh** by providing restore directory as shown in the following example.

```
sudo cxinsight-backup-restore.sh restore /mnt/nfs/share/gcxibackup
```

Running the restore script automatically creates the new backup path and restores the old data.

Note:

- The time taken to restore the old data depends on its size. In test environment, the average duration to restore the old data is about 15 minutes approximately.
- You can restore the old data that is backed up until the last backup activity. The dashboard or metrics created after the backup activity is complete and before the system failure cannot be restored. For example, if the backup activity runs at 10.00 pm every day, and if the system stopped responding at 11.00 pm, then the data created between 10.00 pm and 11.00 pm is not restored.
- Do not use * in directory names.

Backup log files

You can find the archive of CX Insights log files such as application log, tomcat log, and so on, in the backup directory configured as part of backup settings. The log files are archived in the tar format with the archived date and time as its file name.

To backup log files, you do not need any specific configuration.

Related Topics:

[Install CX Insights server](#)

Troubleshooting CX Insights for Installation and Configuration Issues

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

Error	Description	Solution
User is unable to login to the CX Insights server.	This error may occur if the Endpoint update service is not running.	Verify if the Endpoint update service is up and running by using the following command: <code>systemctl status endpoint</code> If the status of the service shows that it is stopped, start the service by running the following command: <code>systemctl start endpoint</code> If the service doesn't start, verify if the endpoint service update file is available in the following location: <code>/usr/local/bin/endpoint-update-service.py</code> You can also verify the endpoint.log for more information which is located inside the gcxi-log persistent volume . An example path looks like <code>/opt/local-path-provisioner/pvc-3f41dbeb-2649-40ad-9106-54228244ce77</code>
Current user has no accessible project, or lacks privilege 'WebUser'. Please contact the administrator.	This error may occur when a user without an Analytics license logs in to the CX Insights application.	For the specific user, in Interaction Administrator>User configuration dialog, enable Analytics License (Core) and also select Enable Licenses check box. If the same error occurs even after enabling the licenses, clear the cookies and try logging in again.
Bad gateway	This error may occur when a user logs in with a different account, and proceed logging in by selecting Trusted Authentication Request .	Verify if IC-Secure Token is reachable in CIC Server.
Error in login Please contact your Administrator.	This error may occur when a user logs in with a different account, and proceed logging in by selecting Trusted Authentication Request .	Verify if IC Secure Token Certificate is properly placed in <code>/opt/tomcat/webapps/MicroStrategy/WEB-INF/classes/resources/SAML/IDPMetadata.xml</code> in the CX Insights server. You can also check SAML.log for more information. Tip: To get the path of SAML.log file, run the following command: <code>find / -name 'SAML.log'</code>
<code>\$(\r): command not found</code>	While running the shell script, this error may occur because Windows uses <code>\r\n</code> as a new line character and Linux uses <code>\n</code>	To resolve this error, remove <code>\r</code> by using the dos2Unix tool or by using the <code>sed</code> command as shown below: <pre>sed -i 's/\r//g' ansible_install.sh</pre>

<p>Host FQDN error For example: "Error: release pcc-helmcharts failed: Ingress.extensions \"pcc-helmcharts/mstrdataadapterserver\" is invalid: sec.rules[0].host: Invalid value: \"172.26.20.55\": must be a DNS name, not an IP address"</p>	<p>This error may occur when configuring and deploying CX Insights</p>	<p>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).</p>
<p>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"}</p>	<p>This error may occur when configuring and deploying CX Insights</p>	<p>To resolve this error, re-run the following command: <code>sudo ansible-playbook --vault-id cxinsights@prompt -i inventory.yml site_upgrade.yml -K</code></p>
<p>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-zero 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": []}</p>	<p>This error may occur when configuring and deploying CX Insights</p>	<p>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.</p>
<p>Pods evicted state error</p>	<p>This error may occur when configuring and deploying CX Insights</p>	<p>Sometimes many pods are in an evicted state. To remove all the evicted pods, use these commands. Prerequisites: <code>yum install jq</code> <code>kubect1 get pods -A --all-namespaces -o json jq '.items[] select(.status.reason!=null) select(.status.reason contains("Evicted")) "kubect1 delete pod \(.metadata.name) -n \(.metadata.namespace)" xargs -n 1 bash -c</code></p>

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: <your new 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 command and 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:

1. Type the following command to get the pods list.

Kubectl get pods -A

```
[root@pcn-cent7-k3s02 ~]# kubectl get pods -A
NAMESPACE      NAME                                                    READY   STATUS    RESTARTS   AGE
kube-system    coredns-66f496764-s8db4                               1/1     Running   0           55d
kube-system    helm-install-traefik-d4jr4                             0/1     Completed 0           55d
kube-system    svclb-traefik-mwt6h                                    3/3     Running   0           55d
local-path-storage local-path-provisioner-84f4c8b584-fsm8g                1/1     Running   0           55d
kube-system    traefik-785ffdcbbf-whngg                               1/1     Running   0           55d
pcn-tiller-system tiller-deploy-76cd8c74-24krp                           1/1     Running   0           55d
pcn-cxinsights-system pcn-cxinsights-helmcharts-gcxi-postgres-5659fbdcd6-7vkmd 1/1     Running   0           54d
pcn-cxinsights-system pcn-cxinsights-helmcharts-gcxi-7f5c78cb65-qtsn4         1/1     Running   0           28d
pcn-cxinsights-system pcn-cxinsights-helmcharts-mstrdataadapteragent-659f8ddf78-wftxp 1/1     Running   0           26d
pcn-cxinsights-system pcn-cxinsights-helmcharts-mstrdataadapterserver-744bf74f59mzdng 1/1     Running   0           26d
pcn-cxinsights-system pcn-cxinsights-helmcharts-mstrconnector-5c75cb6d66-lpb6x 1/1     Running   0           26d
[root@pcn-cent7-k3s02 ~]#
```

2. To go inside GCXI pod, we need to run the following command. For example, GCXI pod name is `pcn-cxinsights-helmcharts-gcxi-7f5c78cb65-qtsn4`

```
kubectl exec -it pcn-cxinsights-helmcharts-gcxi-7f5c78cb65-qtsn4 bash -n pcn-cxinsights-system
```

3. It allows you to go inside the GCXI pod and then navigate to the logging directory, using following command.

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

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

ls

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

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

```
[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: *****
4/21/21 10:17:01 AM EDT *****
```

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
17-August-2020	Updated server installation procedure, included Switchover, and Backup and Restore features
10-November-2020	Included reverse proxy configuration procedure, added RHEL support
10-February-2021	Updated Install CX Insights Server 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.