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>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table of Contents</td>
<td>2</td>
</tr>
<tr>
<td>What’s New for CX Insights administrators?</td>
<td>3</td>
</tr>
<tr>
<td>CX Insights overview</td>
<td>4</td>
</tr>
<tr>
<td>CX Insights architecture</td>
<td>5</td>
</tr>
<tr>
<td>CX Insights deployment model</td>
<td>5</td>
</tr>
<tr>
<td>CX Insights server</td>
<td>5</td>
</tr>
<tr>
<td>CX Insights prerequisites</td>
<td>6</td>
</tr>
<tr>
<td>CX Insights requirements</td>
<td>6</td>
</tr>
<tr>
<td>Hardware</td>
<td>6</td>
</tr>
<tr>
<td>Software</td>
<td>6</td>
</tr>
<tr>
<td>CX Insights licensing</td>
<td>7</td>
</tr>
<tr>
<td>Analytics access licenses</td>
<td>7</td>
</tr>
<tr>
<td>Analytics feature license</td>
<td>7</td>
</tr>
<tr>
<td>CX Insights server installation</td>
<td>8</td>
</tr>
<tr>
<td>Prerequisite</td>
<td>8</td>
</tr>
<tr>
<td>Install CX Insights server</td>
<td>8</td>
</tr>
<tr>
<td>Install SSL certificate on CIC server</td>
<td>12</td>
</tr>
<tr>
<td>Upgrade containers</td>
<td>15</td>
</tr>
<tr>
<td>Roll back containers</td>
<td>15</td>
</tr>
<tr>
<td>Deleting deployment</td>
<td>15</td>
</tr>
<tr>
<td>CX Insights monitoring and alerting</td>
<td>15</td>
</tr>
<tr>
<td>Install Prometheus</td>
<td>15</td>
</tr>
<tr>
<td>Configure reverse proxy using nginx</td>
<td>17</td>
</tr>
<tr>
<td>Log file</td>
<td>18</td>
</tr>
<tr>
<td>Code to be copied to the nginx.conf file</td>
<td>18</td>
</tr>
<tr>
<td>Ports opened on CX Insights server</td>
<td>20</td>
</tr>
<tr>
<td>CX Insights server configuration</td>
<td>21</td>
</tr>
<tr>
<td>CX Insights server configuration</td>
<td>21</td>
</tr>
<tr>
<td>Allocate Analytics licenses</td>
<td>21</td>
</tr>
<tr>
<td>Configure CX Insights server in Interaction Administrator</td>
<td>22</td>
</tr>
<tr>
<td>Retention Settings</td>
<td>23</td>
</tr>
<tr>
<td>Configure Administrator Access for CX Insights</td>
<td>24</td>
</tr>
<tr>
<td>Configure Access Control for CX Insights dashboards</td>
<td>25</td>
</tr>
<tr>
<td>Test the CX Insights installation</td>
<td>27</td>
</tr>
<tr>
<td>Backup and restore configuration of CX Insights data</td>
<td>28</td>
</tr>
<tr>
<td>Backup CX Insights data</td>
<td>28</td>
</tr>
<tr>
<td>Configure CX Insights backup through Ansible</td>
<td>28</td>
</tr>
<tr>
<td>Configure CX Insights backup through script</td>
<td>28</td>
</tr>
<tr>
<td>Instant backup</td>
<td>29</td>
</tr>
<tr>
<td>Restore CX Insights data</td>
<td>29</td>
</tr>
<tr>
<td>Configure CX Insights data restore through Ansible</td>
<td>29</td>
</tr>
<tr>
<td>Configure CX Insights data restore through script</td>
<td>29</td>
</tr>
<tr>
<td>Backup log files</td>
<td>30</td>
</tr>
<tr>
<td>Troubleshooting CX Insights for Installation and Configuration Issues</td>
<td>31</td>
</tr>
<tr>
<td>Appendix</td>
<td>33</td>
</tr>
<tr>
<td>MicroStrategy Server License Update Process</td>
<td>33</td>
</tr>
<tr>
<td>License Ordering Process</td>
<td>33</td>
</tr>
<tr>
<td>License Request Checklist</td>
<td>33</td>
</tr>
<tr>
<td>Process of Updating new License Key</td>
<td>33</td>
</tr>
<tr>
<td>License Update Verification</td>
<td>35</td>
</tr>
<tr>
<td>Change Log</td>
<td>36</td>
</tr>
</tbody>
</table>
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. CX Insights can be accessed from Google Chrome, Mozilla Firefox, Internet Explorer, and Safari.
CX Insights architecture

CX Insights deployment model

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

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.

<table>
<thead>
<tr>
<th>Component</th>
<th>Large-size customers</th>
<th>Mid-size customers</th>
<th>Small-size customers</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of agents in Contact Center</td>
<td>Above 400</td>
<td>50-400</td>
<td>Less than 50</td>
</tr>
<tr>
<td>Platform</td>
<td>Virtual machine or physical server</td>
<td>Virtual machine or physical server</td>
<td>Virtual machine or physical server</td>
</tr>
<tr>
<td>CPU</td>
<td>8 cores</td>
<td>8 cores</td>
<td>4 cores</td>
</tr>
<tr>
<td></td>
<td>AMD-V or VT-X VM-extensions</td>
<td>AMD-V or VT-X VM-extensions</td>
<td>AMD-V or VT-X VM-extensions</td>
</tr>
<tr>
<td>RAM</td>
<td>32 GB</td>
<td>20 GB</td>
<td>16 GB</td>
</tr>
<tr>
<td>Primary partition</td>
<td>100+ GB (recommended)</td>
<td>50 GB (recommended)</td>
<td>40 GB (recommended)</td>
</tr>
<tr>
<td></td>
<td>50 GB (minimum)</td>
<td>35 GB (minimum)</td>
<td>30 GB (minimum)</td>
</tr>
<tr>
<td>Secondary partition</td>
<td>400+ GB</td>
<td>60 GB (recommended)</td>
<td>45 GB (recommended)</td>
</tr>
<tr>
<td></td>
<td>100 GB (minimum)</td>
<td>45 GB (minimum)</td>
<td>35 GB (minimum)</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>Component</th>
<th>Requirement</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operating system</td>
<td>CentOS 7, RHEL version 7.6, 7.7 and 7.8</td>
</tr>
<tr>
<td></td>
<td>The host supports RHEL versions mentioned above. However, the base image in the container still contains CentOS and Alpine Linux.</td>
</tr>
<tr>
<td>Software components</td>
<td>Virtualization Host:</td>
</tr>
<tr>
<td></td>
<td>• KVM</td>
</tr>
<tr>
<td></td>
<td>• QEMU</td>
</tr>
<tr>
<td></td>
<td>• QEMU+KVM</td>
</tr>
<tr>
<td></td>
<td>• Libvirt</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>License</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>I3_ACCESS_ANALYTICS_CORE</td>
<td>Basic dashboard license to view dashboards.</td>
</tr>
<tr>
<td>I3_ACCESS_ANALYTICS_DESIGNER</td>
<td>This license allows a user to create and modify dashboards.</td>
</tr>
</tbody>
</table>

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:


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/paswd 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
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 the **Backup and restore configuration** topic.

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

<table>
<thead>
<tr>
<th>Property name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>cicServerName</td>
<td>The IP address of the primary CIC server.</td>
</tr>
<tr>
<td>cicBackUpServerName</td>
<td>The IP address of the secondary CIC server.</td>
</tr>
<tr>
<td>cicDBName</td>
<td>The (SQL Server) CIC database name, specified in Setup Assistant.</td>
</tr>
<tr>
<td>cicDBHost</td>
<td>The (SQL Server) CIC database server name, specified in Setup Assistant.</td>
</tr>
<tr>
<td>cicDBLoginID</td>
<td>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 <strong>IC Report Logs</strong> user ID specified in Interaction Administrator.</td>
</tr>
<tr>
<td>cicDBLoginPwd</td>
<td>Specify the encoded password of CIC database user ID mentioned in cicDBLoginID. Encrypt password using <code>base64</code> encryption method only. <strong>Tip:</strong> You can use the following command to encrypt your password: `echo &quot;testpassword&quot;</td>
</tr>
<tr>
<td>langs (optional)</td>
<td>The localization language required for your organization. Configuring langs 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.</td>
</tr>
</tbody>
</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

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>tz</td>
<td>Specify the time zone of the region where gcxi server is installed.</td>
</tr>
<tr>
<td>hosts</td>
<td>The Linux host name of the CX Insights server. Note that the host name you specify here must be an FQDN.</td>
</tr>
<tr>
<td>maxPoolSize (optional)</td>
<td>The maximum number of concurrent web sessions allowed. This is an optional parameter and the default value is 200.</td>
</tr>
</tbody>
</table>

**tls (ingress)**

- 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.

*Note*: If you enable TLS, you must install an SSL certificate by following the [Install SSL certificate on CIC server](#) procedure.

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>secret name (ingress)</td>
<td>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</td>
</tr>
<tr>
<td>hosts (ingress)</td>
<td>Specify the FQDN of ingress host. Typically, this is the FQDN of the CX Insights server that you configure in the hosts setting.</td>
</tr>
</tbody>
</table>

**tls (prometheusIngress)**

- 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.

*Note*: If you enable TLS, you must install an SSL certificate by following the [Install SSL certificate on CIC server](#) procedure.

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>secret name (prometheusIngress)</td>
<td>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</td>
</tr>
<tr>
<td>hosts (prometheusIngress)</td>
<td>Specify the FQDN of Prometheus ingress host. Typically, this is the FQDN of the CX Insights server that you configure in the hosts setting.</td>
</tr>
</tbody>
</table>

### 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="
maxPoolSize: 250
certICSAML:
MIIDoTCCAomgAwIBAgIFQWCBgwkwDQYJKoZIhvcNAQEFBQAwRzEQMA4GA1UECgwH
U2VydmVyZ2VuZ2luZyBEb21haW5lciBEb24gaXMgdGhpc2UgQ29yZCIwIzAlMAIw
BgkqhkiG9w0BAQsFAAOCAQ8AMIGB
LmRldj1wMDAuY29t:
MIIDoTCCAomgAwIBAgIFQWCBgwkwDQYJKoZIhvcNAQEFBQAwRzEQMA4GA1UECgwH
U2VydmVyZ2VuZ2luZyBEb21haW5lciBEb24gaXMgdGhpc2UgQ29yZCIwIzAlMAIw
BgkqhkiG9w0BAQsFAAOCAQ8AMIGB
LmRldj1wMDAuY29t:
MIIDoTCCAomgAwIBAgIFQWCBgwkwDQYJKoZIhvcNAQEFBQAwRzEQMA4GA1UECgwH
U2VydmVyZ2VuZ2luZyBEb21haW5lciBEb24gaXMgdGhpc2UgQ29yZCIwIzAlMAIw
BgkqhkiG9w0BAQsFAAOCAQ8AMIGB
LmRldj1wMDAuY29t:
MIIDoTCCAomgAwIBAgIFQWCBgwkwDQYJKoZIhvcNAQEFBQAwRzEQMA4GA1UECgwH
U2VydmVyZ2VuZ2luZyBEb21haW5lciBEb24gaXMgdGhpc2UgQ29yZCIwIzAlMAIw
BgkqhkiG9w0BAQsFAAOCAQ8AMIGB
LmRldj1wMDAuY29t:
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.

```yaml
---
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 }}'
```

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.

```yaml
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.

```bash
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
To see all the services running in all namespaces, use the command `kubectl get services -A`

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

To see all the persistent volumes in all namespaces, use the command `kubectl get pvc -A`

To see all 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.

**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.
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.

   ```yaml
   gcxi:
     image:
       tag: 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

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.

```yaml
# 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__
              - __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.

7. To receive e-mail notifications/pagerduty configure alertmanager. More details about alert manager is found in [https://prometheus.io/docs/alerting/alertmanager/](https://prometheus.io/docs/alerting/alertmanager/) and download is available in the [https://prometheus.io/download/](https://prometheus.io/download/).

8. After downloading configure prometheus.yml with alert manager in the # Alertmanager configuration:

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

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

   ```yaml
   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: xxxxxxxxx@xxxx.com
         - from: xxxxxxxxx@xxxxxx.com
         - smarthost: xxxx
         - 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](https://nginx.org/en/docs/).

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.

3. 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.

   ```bash
   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.

   ```bash
   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**

```conf
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/ngx_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.

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

Related Topics:
Install CX Insights server
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.

<table>
<thead>
<tr>
<th>License</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CORE</td>
<td>Basic dashboard license to view dashboards.</td>
</tr>
<tr>
<td>DESIGNER</td>
<td>This license allows a user to create and modify dashboards.</td>
</tr>
</tbody>
</table>

In addition, you must select the Enable Licenses check box to activate the Analytics license.
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](#)
Allocate Analytics licenses

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.

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.
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.

<table>
<thead>
<tr>
<th>Error</th>
<th>Description</th>
<th>Solution</th>
</tr>
</thead>
<tbody>
<tr>
<td>User is unable to login to the CX Insights server.</td>
<td>This error may occur if the Endpoint update service is not running.</td>
<td>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 <code>endpoint.log</code> for more information which is located inside the <code>gcki-log persistent volume</code>. An example path looks like <code>/opt/local-path-provisioner/pvc-3f41dbeb-2649-40ad-9106-54228244ce77</code>.</td>
</tr>
<tr>
<td>Current user has no accessible project, or lacks privilege 'WebUser'. Please contact the administrator.</td>
<td>This error may occur when a user without an Analytics license logs in to the CX Insights application.</td>
<td>For the specific user, in <code>Interaction Administrator&gt;User configuration</code> 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.</td>
</tr>
<tr>
<td>Bad gateway</td>
<td>This error may occur when a user logs in with a different account, and proceed logging in by selecting Trusted Authentication Request.</td>
<td>Verify if IC-Secure Token is reachable in CIC Server.</td>
</tr>
<tr>
<td>Error in login Please contact your Administrator.</td>
<td>This error may occur when a user logs in with a different account, and proceed logging in by selecting Trusted Authentication Request.</td>
<td>Verify if IC Secure Token Certificate is properly placed in <code>vi /opt/tomcat/webapps/MicroStrategyWEB-INF/classes/resources/SAML/IDP/Metadata.xml</code> in the CX Insights server. You can also check SAML.log for more information. <strong>Tip:</strong> To get the path of SAML.log file, run the following command: <code>find / -name 'SAML.log'</code>.</td>
</tr>
<tr>
<td>&quot;$\backslash r&quot;**: command not found</td>
<td>While running the shell script, this error may occur because Windows uses '\n' as a new line character and Linux uses '\n'</td>
<td>To resolve this error, remove &quot;\r&quot; by using the <code>dos2Unix</code> tool or by using the <code>sed</code> command as shown below: <code>sed -i 's/\r//g' ansible_install.sh</code>.</td>
</tr>
<tr>
<td>Issue Description</td>
<td></td>
<td></td>
</tr>
<tr>
<td>----------------------------------------------------------------------------------</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Host FQDN error</strong> For example: &quot;Error: release pcc-helmcharts failed: Ingress.extensions &quot;pcc-helmchartsmstrdataadapterserver&quot; is invalid: sec.rules[0].host: Invalid value: &quot;172.26.20.55&quot;: must be a DNS name, not an IP address&quot;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>This error may occur when configuring and deploying CX Insights</td>
<td></td>
<td></td>
</tr>
<tr>
<td>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).</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>K3s server start error</strong> For example: FAILED!=&gt; (&quot;changed&quot;: false, &quot;msg&quot;: &quot;Unable to restart service K3s: Failed to restart k3s.service: Connection timed out\n\nssee system logs and 'systemctl status k3s.service' for details.&quot;)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>This error may occur when configuring and deploying CX Insights</td>
<td></td>
<td></td>
</tr>
<tr>
<td>To resolve this error, re-run the following command: sudo ansible-playbook --vault-id cxinsights@prompt -i inventory.yml site_upgrade.yml -K</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Wrong pcon-mstr folder path error</strong> For example: FAILED!=&gt; (&quot;changed&quot;: false, &quot;cmd&quot;: [&quot;helm&quot;, &quot;install&quot;, &quot;pcon-mstr&quot;, &quot;--name&quot;, &quot;pcc-helmcharts&quot;, &quot;--namespace&quot;, &quot;pcn-cxinsights-system&quot;, &quot;--tiller-namespace&quot;, &quot;pcn-tiller-system&quot;, &quot;-f&quot;, &quot;~/values.yml&quot;], &quot;delta&quot;: &quot;0:0:0.166113&quot;, &quot;end&quot;: &quot;2020-02-21 06:47:47.533577&quot;, &quot;failed_when_result&quot;: true, &quot;msg&quot;: &quot;non-sero return code&quot;, &quot;rc&quot;: 1, &quot;start&quot;: &quot;2020-02-21 06:47:47.367464&quot;, &quot;stderr&quot;: &quot;Error: failed to download &quot;pcon-mstr&quot; (hint: running 'helm repo update' may help)&quot;, &quot;stderr_lines&quot;: [&quot;Error: failed to download &quot;pcon-mstr&quot; (hint: running 'helm repo update' may help)&quot;], &quot;stdout&quot;: &quot;&quot;, &quot;stdout_lines&quot;: [&quot;]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>This error may occur when configuring and deploying CX Insights</td>
<td></td>
<td></td>
</tr>
<tr>
<td>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.</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Pods evicted state error</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>This error may occur when configuring and deploying CX Insights</td>
<td></td>
<td></td>
</tr>
<tr>
<td>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</td>
<td>jq '.items[]</td>
<td>select(.status.reason!=null)</td>
</tr>
<tr>
<td><strong>K3s server start error</strong> For example: FAILED!=&gt; (&quot;changed&quot;: false, &quot;msg&quot;: &quot;Unable to restart service K3s: Failed to restart k3s.service: Connection timed out\n\nssee system logs and 'systemctl status k3s.service' for details.&quot;)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>This error may occur when configuring and deploying CX Insights</td>
<td></td>
<td></td>
</tr>
<tr>
<td>To resolve this error, re-run the following command: sudo ansible-playbook --vault-id cxinsights@prompt -i inventory.yml site_upgrade.yml -K</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
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.

<table>
<thead>
<tr>
<th>CX Insights Version</th>
<th>EIC Release</th>
<th>MicroStrategy Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0</td>
<td>2019 R4</td>
<td>10.11</td>
</tr>
<tr>
<td>1.0</td>
<td>2020 R1</td>
<td>10.11</td>
</tr>
<tr>
<td>2.0</td>
<td>2020 R2</td>
<td>10.11</td>
</tr>
<tr>
<td>3.0</td>
<td>2020 R3</td>
<td>2020</td>
</tr>
<tr>
<td>4.0</td>
<td>2020 R4</td>
<td>2020</td>
</tr>
<tr>
<td>4.0</td>
<td>2021 R1</td>
<td>2020</td>
</tr>
<tr>
<td>4.0</td>
<td>2021 R2</td>
<td>2020</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Scenario</th>
<th>Request for New License</th>
</tr>
</thead>
<tbody>
<tr>
<td>New CX Insights Customer on boarded</td>
<td>Yes</td>
</tr>
<tr>
<td>Existing CX Insights Perpetual Customer</td>
<td>Yes</td>
</tr>
<tr>
<td>Existing Perpetual Customer, who is moving to a higher MicroStrategy version due to CIC version upgrade</td>
<td>Yes</td>
</tr>
<tr>
<td>Existing Perpetual Customer, who is upgrading their CIC version but has the identical MicroStrategy version in both the CIC versions</td>
<td>No</td>
</tr>
<tr>
<td>Existing CX Insights Subscription Customer, who is renewing the contract</td>
<td>Yes</td>
</tr>
<tr>
<td>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</td>
<td>Yes</td>
</tr>
<tr>
<td>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</td>
<td>No</td>
</tr>
</tbody>
</table>

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>
```

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 downtime 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.
   ```bash
   kubectl get pods -A
   ```

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`
   ```bash
   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.
   ```bash
   cd /mnt/log/mstr
   ```

4. To get the list of files use the following command
   ```bash
   ls
   ```

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.
## Change Log

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

<table>
<thead>
<tr>
<th>Date</th>
<th>Change</th>
</tr>
</thead>
<tbody>
<tr>
<td>28-June-2019</td>
<td>Initial release</td>
</tr>
<tr>
<td>21- November-2019</td>
<td>Updated architecture diagram</td>
</tr>
<tr>
<td>02-December-2019</td>
<td>Added Configure HTTPS For Nginx topic</td>
</tr>
<tr>
<td>04-December-2019</td>
<td>Updated Analytics Configuration description</td>
</tr>
<tr>
<td>06-April-2020</td>
<td>Added Kubernetes Deployment Information</td>
</tr>
<tr>
<td>29-April-2020</td>
<td>Added Troubleshooting Information</td>
</tr>
<tr>
<td>04-May-2020</td>
<td>Updated Server Install and Upgrade Containers topics</td>
</tr>
<tr>
<td>11-June-2020</td>
<td>Updated Server Install and help.genesys.com links</td>
</tr>
<tr>
<td>21-July-2020</td>
<td>Updated CX Insights configuration in Interaction Administrator topic</td>
</tr>
<tr>
<td>17-August-2020</td>
<td>Updated server installation procedure, included Switchover, and Backup and Restore features</td>
</tr>
<tr>
<td>10-November-2020</td>
<td>Included reverse proxy configuration procedure, added RHEL support</td>
</tr>
<tr>
<td>10-February-2021</td>
<td>Updated Install CX Insights Server topic.</td>
</tr>
<tr>
<td>12-March-2021</td>
<td>Added a new topic MicroStrategy Server License Update Process</td>
</tr>
<tr>
<td>11-May-2021</td>
<td>Added License Update Verification Information</td>
</tr>
<tr>
<td>20-May-2021</td>
<td>Added additional steps to License Update Verification Information</td>
</tr>
</tbody>
</table>