Contents

[Introduction: 2](#_Toc90915553)

[Components of Enterprise Connect: 2](#_Toc90915554)

[EC Service: 2](#_Toc90915555)

[EC Agent: 2](#_Toc90915556)

[Ensuring Security of the Enterprise Connect: (Security) 2](#_Toc90915557)

[How to subscribe to EC Service 2](#_Toc90915558)

[Prerequisites for Subscribing to UAA: 2](#_Toc90915559)

[Create EC Service: 3](#_Toc90915560)

[Add EC Scopes to UAA: 4](#_Toc90915561)

[Traditional EC Connection: 5](#_Toc90915562)

[Connecting to multiple targets (VLAN) 5](#_Toc90915563)

[Linux clients 5](#_Toc90915564)

[Windows clients 5](#_Toc90915565)

[Connecting to APIs (TLS) 5](#_Toc90915566)

[Fuse mode 5](#_Toc90915567)

[Deployment options 5](#_Toc90915568)

[Cloud Foundry 5](#_Toc90915569)

[EKS/AKS 5](#_Toc90915570)

[Binary process 5](#_Toc90915571)

[Troubleshooting tips: 6](#_Toc90915572)

[Common issues: 6](#_Toc90915573)

[Health page 6](#_Toc90915574)

Enterprise Connect (EC) Developer Guide:

Table of contents to EC developer guide (self-managed subscriptions)

# Introduction:

Enterprise Connect is a service that enables you to establish a secure and scalable connection between any cloud environment and your enterprise resources. As a service, it connects any two networks governed by different transmission protocols. Data passes through Enterprise Connect before getting routed to its destination.

# Components of Enterprise Connect:

Enterprise Connect service has two components

## EC Service:

The Enterprise Connect (EC) is a microservice that has multiple components. It constitutes a client, a server, and a gateway. The data passes through the EC service through TCP (Transmission Control Protocol).

## EC Agent:

EC Client: Both the EC client and EC server act as agents in Enterprise Connect (EC) service. The EC client and Server, though agents, have the ability to act as a gateway also. The EC Client appears on the side of the source system (a client application) and the EC Server appears on the side of the target system (a database).

# Ensuring Security of the Enterprise Connect: (Security)

The Enterprise Connect service ensures secure connectivity between any two independent networks. How to ensure secure connection??? Through TLS (Transport Layer Security) protocol???

# How to subscribe to EC Service

Enterprise Connect (EC) is a microservice offered as part of Predix environment. To use the service, you need to subscribe to the Predix User Account and Authentication (UAA) service as a trusted user. You can either use the UAA dashboard or the Cloud Foundry command line interface (CLI) to create and configure your service.

# Prerequisites for Subscribing to UAA:

In order to subscribe to the Enterprise Connect service, you need to meet some prerequisites. They are:

* An active account in predix.io
* Cloud Foundry CLI installed

## Create EC Service:

First, you need to create the Enterprise Connect service. Here are the steps involved in the creation of the EC service.

Configure your proxy settings if necessary.

If the traffic between your corporate network and the internet is monitored, access to some tools/applications may be blocked by the proxy. Therefore, you will need to configure your proxy settings to access remote resources such as Enterprise Connect. The values for your proxy settings vary based on your location and network configuration. Therefore, you may contact your IT administrator for the correct proxy settings.

Creating a UAA Service Instance

You can create multiple instances of the UAA service in your space.

However, it is a best practice to delete any older and unused UAA service instances before creating a new instance.

Steps to create a new service instance:

* Sign into your Predix account at <https://www.predix.io>
* Navigate to Catalog > Services, then click the User Account and Authentication tile
* Click **Subscribe** on the required plan
* Complete the fields on the New Service Instance page
* Click Create Service

**Note:** To know more about the creation of a UAA service instance, please visit the below-mentioned page:

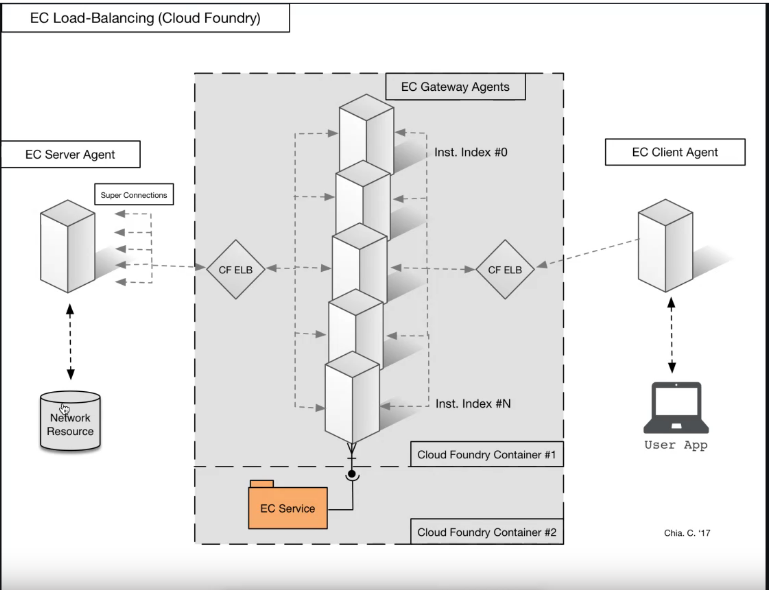
<https://www.ge.com/digital/documentation/edge-software/t_creating_predix_uaa_instance.html#task_y1l_vms_2s>

Binding an Application to the UAA Instance

Applications that run in the Cloud Foundry environment gain access to bound service (a server in a client-server interface) instances through the credentials stored in VCAP\_SERVICES. Therefore, the next step involves binding your application to your UAA instance through the VCAP\_SERVICES.

To know more about how to bind …

In the below diagram we have a client application running on the network and the application wants to access a network resource that could be a database, an SFDC server, or can even be an API service provider. If we want to access the resource from the client application, we need to install some agents, the Server Agent on the side of the network resource so that it can access the resource over a TCP protocol. Thereafter, it will translate the traffic over TCP into https and then via the EC gateway the data will be passed to the Client Agent and from them it will be accessed by the client application.



Add EC Scopes to UAA: (Please refer GE Digital documentation)

You need to add Enterprise Connect scopes to the User Account and Authentication (UAA).

# Traditional EC Connection:

In the traditional mode, you have three components namely, EC Client, EC Server, and EC Gateway that constitute the Enterprise Connect (EC) service. This arrangement ensures connectivity when you have multiple clients and servers and there is a need to be connected through a gateway.

What is the difference between a traditional EC connection and a Modern??? EC connection?

# Connecting to multiple targets (VLAN)

Here we will be writing something about how to use Enterprise Connect service to connect to multiple targets.

## Linux clients

How to connect multiple Linux clients?

## Windows clients

How to connect multiple Windows clients?

# Connecting to APIs (TLS)

How to connect the EC service to the APIs (Transport Layer Security) needs to be written over here.

# Fuse mode

In the traditional mode, you have three components namely, EC Client, EC Server, and EC Gateway that constitute the Enterprise Connect (EC) service. This arrangement ensures connectivity when you have multiple clients and servers and there is a need to be connected through a gateway.

However, if you have a requirement where you need only a single EC connection, then you can combine the EC Gateway with either EC Client or EC Server. In that case, you need only two EC components (instead of three in case of a traditional mode).

* Either a Gateway with server and a client (GW Server and Client)
* Or a Gateway with client and a Server (GW Client and Server)

Such an arrangement is termed as a Fuse Mode or direct connection.

# Deployment options

What are the various deployment options that are available with the Enterprise Connect service needs to be elucidated here.

## Cloud Foundry

Cloud Foundry is one of the deployment options.

## EKS/AKS

EKS/AKS is one of the deployment options.

## Binary process

Binary process is one of the deployment options.

# Troubleshooting tips:

Various troubleshooting tips that are available will be explained over here.

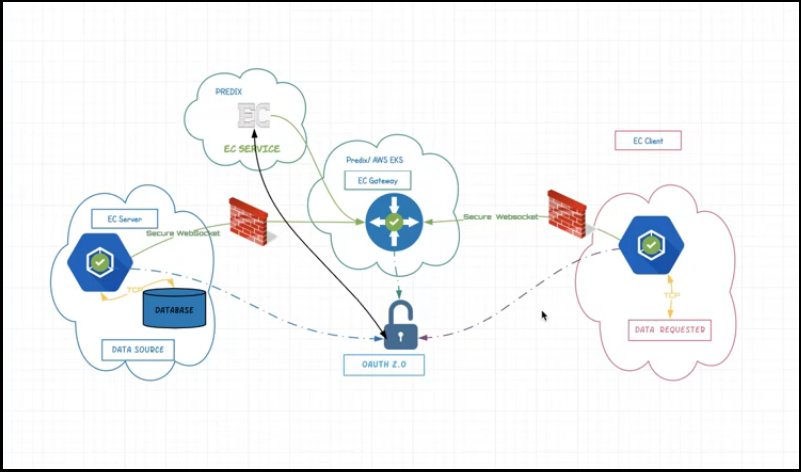
Any existing document on troubleshooting tips?

## Common issues:

Common issues that are faced by the Enterprise Connect service

## Health page

What is a health page?



Needs to know more about this diagram.

EC stands for "Enterprise Connect" is a professional assembly led by a group of researchers, and engineers who have common vision of the de-centralisation network, currently sponsored by General Electric. It has the following main components:

EC Service: Currently built in Cloud-Foundry, the EC service is a single-tenant, self-contained, async-ps microservice. EC service is endorsed by the GE security team and is available as a GA release. The service breaks down into the following functions.

* usage reporting
* system performance/health check
* account managing
* security validation
* two-way digital certs
* management UI

**EC Agent:** To make EC conformed to the design pattern of the TURN protocol (Traversal Using Relays around NAT, RFC5766) for the sake of its extendibility, sustainability and compatibility of mass-computing (cloud computing) it is essentially equipped with three explicitly functional modes- client, server, and Gateway, of each may operate independently without one another.

When in the 'client' node handles security handshakes, IP filtering, and seeks for the permission from a EC service instance by passing on the client/server credentials to authorising requests. Upon authenticated, the Gateway will then perform two-way binding (client/server), induce a session, and then signify requesters for the readiness.

Enterprise Connect is a connectivity service, it provides connectivity between two different networks over a secure channel. And behind the scenes it will use the web sockets

**Common Issues:**

**Functionality:** Gateway

When the Enterprise Connect gateway (either Predix or AWS) is down, all the EC connections from the gateway will be lost. This may occur if the Predix gateway application is mistakenly stopped, or the AWS gateway process is killed. When this situation occurs, the Operations Team has to restart the gateway manually.

**Potential Failure Mode(s):** Gateway (Predix or AWS) is down

**Effects of Failure:** All EC connections from the gateway will be lost.

**Potential causes/mechanisms of failure**: Predix Gateway application stopped by mistake or AWS Gateway process killed

**Current Design Controls:** manually start the gateway

**Recommendation:** Ops team have to start gateway manually

**Mandatory flags in gateways -**

mod - "gateway"

gpt - gateway port -

- Predix - ${port} to pick random port

- AWS - Should give the port where EC2 instance registered with target group

zon - EC service subscription id

sst - EC service URI

tkn - EC service admin token

hst - gateway URL

**Issue tracker:**

<https://github.com/Enterprise-connect/sdk/issues/54>

When the EC agent is down, the connectivity between the EC service and the agent gets snapped. The reasons behind the failure could be occasional heavy loads or processes getting killed. The Operations Team has already prepared a restage script and needs to deploy it along with agent deployment. The Operations team needs to check the connection details and should identify which agent went down. And, thereafter, deploy the restage script to bring the agent up whenever it is down.

**Functionality:** Client/Server

**Potential Failure Mode(s):** EC agent is down

**Effects of Failure:** EC connect with the agent will break

**Potential causes/mechanisms of failure**: Heavy loads but occasional OR process got killed.

**Current Design Controls:** Ops team already prepared restage script and need to deploy along with agent deployment.

**Recommendation:** Ops team can check the connection details and will identify which agent went down and have to bring it back. And deploy the restage job to bring agent up whenever it is down.

sst - EC service URL

zon - EC subscription id

rht - Target system IP/ Host name

rpt - Target system port to connect

dbg - Debug mode

**Flags in client agents -**

mod - "client"

grp - EC service group name

aid - agent id from EC group

tid - EC server agent id (make sure server and client agents from same EC group)

oa2 - UAA Oauth2 URL

cid - UAA client id

csc - UAA client secret

dur - Time in secs for UAA token validty

hst - Gateway URL - wss://{gatewayurl}/agent

sst - EC service URL

zon - EC subscription id

lpt - listen port

dbg - Debug mode

When the Client Agent fails to start or throws and error while making a connection, the gateway remains out of service. In that case the agent script needs to be corrected and a restage script needs to be deployed to bring the service back.

**Functionality:** GW-Client

**Potential Failure Mode(s):** Client agent will not start or throw error while making connection

**Effects of Failure:** Unable to make EC connection and gateway is down

**Potential causes/mechanisms of failure**: Check the agent script. And if agent is going down, have to deploy the restage script.

**Current Design Controls:** Correct the agent script

**Recommendation:** Correct the agent script

Flags for GW:Client -

mod: gw:client

gpt: "{gateway port}"

lpt: "{local listening port}"

zon: {EC Zone Id}

grp: {EC Group name}

sst: {EC Service URL}

dbg: true

tkn: {EC Admin token}

hst: wss://{gatewayurl}/agent

aid: {Client agent id}

tid: {Target agent id}

Note: In yml format, port numbers must given in double quotes

When the Server Agent fails to start or throws an error while making a connection, the gateway remains out of service. In that case, the agent script needs to be corrected and a restage script needs to be deployed to bring the service back.

**Functionality:** GW-Server

**Potential Failure Mode(s):** Server agent will not start or throw error while making EC connection

**Effects of Failure:** Unable to make EC connection and gateway is down

**Potential causes/mechanisms of failure**: Check the agent script. And if agent is going down, have to deploy the restage script.

**Current Design Controls:** Correct the agent script

**Recommendation:** Flags for gw:server -

mod: "gw:server"

gpt: gateway port

Predix: Use $port} to get ramndom port

AWS: Use port number where EC2 registered with target group

zon: EC Service subscription Id

grp: EC Service group id

sst: EC Service URL

dbg: enable debug

tkn: EC Service admin token

aid: agent id from EC service group

rpt: Port number were target system listen

rht: Target system host or IP

hst: wss://{gatewayurl}/agent

When you are unable to make EC connection with VLAN you need to check the configuration.

Carryout the following checks if EC connection is not established with VLAN:

* Check if -vln flag enabled in client script
* Check if plugins .yml file available next to client agent script
* Check the content format in plugins.yml

ec-plugin:

vlan:

command: ./vln\_linux\_sys

ips: {target ip}/32

status: active

Client application must use target ip and port to connect (not client host ip and lpt)

**Functionality:** VLAN plugin

**Potential Failure Mode(s):** Unable to make EC connection with VLAN

**Effects of Failure:** EC connection not established due to multiple issues

**Potential causes/mechanisms of failure**: Configuration check

**Current Design Controls:** Do following checks if EC connection not established with VLAN -

- Check if -vln flag enabled in client script

- Check if plugins. yml file available next to client agent script

- Check the content format in plugins.yml

---

ec-plugin:

vlan:

command: ./vln\_linux\_sys

ips: {target ip}/32

status: active

- Client application must use target ip and port to connect (not client host ip and lpt)

- 2

**Recommendation:** Follow the steps as in 'Current Design Controls'

Steps were documented in - https://github.com/Enterprise-connect/sdk/tree/v1/plugins/vln

Issue tracker:

https://github.com/Enterprise-connect/sdk/issues/74

https://github.com/Enterprise-connect/sdk/issues/43

https://github.com/Enterprise-connect/sdk/issues/45

https://github.com/Enterprise-connect/sdk/issues/32

https://github.com/Enterprise-connect/sdk/issues/24

<https://github.com/Enterprise-connect/sdk/issues/25>

If the TLS Plugin is not able to make a connection with TLS (Transport Layer Security) then you need to check the configuration.

Please carryout the following checks to fix the problem

pluins.yml file must be next to server script

plugins.yml fomart -

ec-plugin:

tls:

status: active

schema: https

hostname: github.build.ge.com

tlsport: "443"

proxy: ""

port: "9847"

command: ./tls\_darwin\_sys

**Functionality:** TLS pluigin

**Potential Failure Mode(s):** Unable to make connection with TLS

**Effects of Failure:**

**Potential causes/mechanisms of failure**: Configuration check

**Current Design Controls:** Do the following checks -

- pluins.yml file must be next to server script

- plugins.yml fomart -

ec-plugin:

tls:

status: active

schema: https

hostname: github.build.ge.com

tlsport: "443"

proxy: ""

port: "9847"

command: ./tls\_darwin\_sys

**Recommended Actions:** Check the formats as suggested

**Server flags -**

mod: server

aid: Target agent id

grp: {EC group name}

cid: {UAA client id}

csc: {UAA client secret}

dur: 3000

oa2: https://{OAuth URL}/oauth/token

hst: wss://{Gateway URL}/agent

zon: {EC Subscription Id}

sst: https://{EC Service URL}

rht localhost

rpt 9847 (Should match with port in plugins.yml)

Additional flags: -dbg -plg

Issue tracker:

https://github.com/Enterprise-connect/sdk/issues/9

<https://github.com/Enterprise-connect/sdk/issues/42>

If the CF (Cloud Foundry) service is down, the EC service for a subscription goes down. The failure will impact all the connections under the subscription and the Operations team has to restage the subscription application to bring the service back.

**Item/Function:** CF (Cloud Foundry) Service

**Potential Failure Mode(s):** EC service for a subscription is down

**Potential Effect(s) of Failure:** All connections under subscriptions will go down

**Recommended Action(s):** Ops team have to restage the subscription application to bring it up

Every Predix service has a broker application and if the broker app for Predix EC service is down, you will not be able to create new subscriptions. If the CF (Cloud Foundry) service broker fails you need to reach out to Predix support team for help.

**Item / Function:** CF Service Broker

**Potential Failure Mode(s):** Every Predix service will have a broker app and if the broker app for Predix EC service is down

**Potential Effect(s) of Failure:** Unable to create new subscriptions

**Potential Cause(s)/ Mechanism(s) of Failure:** Predix environment issues

**Recommended Action(s):** Follow-up with Predix support team

If the Predix Cloud Foundry is down, the entire Predix platform will be out of service and as a result all the EC services will get adversely affected. Therefore, to resolve the network issue you need to reach out to Predix support team to fix the issue. Once the platform is back on track, all EC connection will be back to normal.

**Item / Function:** CF (Predix)

**Potential Failure Mode(s):** Predix platform down

**Potential Effect(s) of Failure:** Effect all EC services

**Potential Cause(s)/ Mechanism(s) of Failure:** Network Issue/ Platform Issue

**Current Design Controls:** Once platform is back, all EC connections will back to normal

**Recommended Action(s):** Have to do follow-up with Predix support team to fix the issue

When the Predix UAA (User Account and Authentication) service is down, it impacts all new EC connections. Due to Predix environment issues, chances UAA service can go down. Subscribe at status.predix.io and will receive notification when any Predix service goes down. Since it is an automated alert, no action is needed by the admin team.

**Item / Function:** UAA

**Potential Failure Mode(s):** Predix UAA service is down

**Potential Effect(s) of Failure:** Effect all new EC connections

**Potential Cause(s)/ Mechanism(s) of Failure:** Due to Predix environment issues, chances UAA service can go down

**Current Design Controls:** Subscribe at status.predix.io and will receive notification when any Predix service went down

**Recommended Action(s):** Automated Alert. No action needed by Admin Team.

**Potential Failure Mode(s):** VM (Virtual Machine) where EC agent is running goes down

**Potential Effect(s) of Failure:** Effect EC connection for agents installed in that VM

**Potential Cause(s)/ Mechanism(s) of Failure:** VM's in cloud used to go down but will come back immediately.

**Current Design Controls:** Create bootstrap script to install EC agents and keep them running

**Recommended Action(s):** Best practice to have it during agent installation